IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

SANDPIPER CDN, LLC, *Plaintiff,* v. **MICROSOFT CORPORATION,** *Defendant*.	Case No. 2:25-cv-00664-JRG-RSP **JURY TRIAL DEMANDED**

SANDPIPER CDN'S DISCLOSURE OF ASSERTED CLAIMS AND INFRINGEMENT CONTENTIONS TO MICROSOFT

Under Local Patent Rules 3-1 and 3-2, Sandpiper CDN, LLC ("Sandpiper") makes the following Disclosure of Asserted Claims and Infringement Contentions and Accompanying Document Production as to Defendant Microsoft Corporation ("Microsoft"). Sandpiper waives no claim, contention, or argument regarding the factual and/or legal details of the following preliminary infringement contentions. The following should not be construed as Sandpiper's preliminary construction of any asserted claim or claim term. Sandpiper reserves the right to change, amend, supplement, or otherwise alter these contentions based on receipt of further information that will be provided by, produced by, or otherwise made available by Microsoft through discovery that is not available at the time of these disclosures. For example, but without limitation, because Microsoft's Accused Functionalities use software and/or firmware to perform certain operations, details regarding the operation of the Accused Functionalities are unavailable to Sandpiper. As such, Sandpiper does not have available to it, and Microsoft has not yet produced, internal Microsoft documents describing the Accused Functionalities, such as source code, specifications, schematics, flow charts, etc. Sandpiper thus reserves the right to amend its

contentions under the Court's Discovery Order. Sandpiper also reserves the right to contend that the claims do not require further construction, clarification, or interpretation. The claim charts in the attached Exhibits are not designed to represent the entire scope of infringement but are merely intended to illustrate examples of manners in which the Accused Functionalities infringe the identified claims of Sandpiper's Patents. Further, the division of claim elements in the attached claim charts is neither intended to be a modification of the claim language itself nor an admission that the claims should be construed by the division. Rather, the division of claim elements is provided for convenience and clarity. Sandpiper's contentions apply to Microsoft's products and any other similar past, present, or future products, as well as systems incorporating Microsoft's products or other products with the same or substantially similar features. Sandpiper reserves all rights to rely on additional evidence in its possession and evidence yet to be discovered.

The Accused Functionalities

As described in the Complaint, the Microsoft CDN technology and architecture, and the services Microsoft offers for sale and uses internally that utilize this technology and architecture, are collectively referred to herein as the "Accused Functionalities." The Accused Functionalities include Microsoft's products and services that satisfy each step of the method claims of the Sandpiper Patents, including Microsoft's Azure CDN and Azure Front Door (classic, standard, and premium) platforms that provide public CDN services to Microsoft customers and/or third parties, as well as Microsoft's internal use of these and similar CDN services, including those that support gaming (e.g., Xbox console and cloud gaming and downloads/updates; other cloud and mobile gaming and downloads/updates; PC gaming and downloads/updates; third-party gaming and downloads/updates using Microsoft's CDN infrastructure), Office 365 (e.g., Teams,

OneDrive, Outlook, Access, SharePoint, Copilot), Windows downloads and updates, and web resources (e.g., Edge, Linkedin, Bing, GitHub) (collectively, "Microsoft's Services").

Each step of the asserted claims of the Sandpiper Patents are either performed by or otherwise attributable to Microsoft. To the extent another actor performs any of these steps, it does so as an agent of Microsoft. Moreover, Microsoft directs and/or controls that performance, conditioning participation in the activity or the receipt of a benefit upon performance of the patented method steps, and establishing the manner or timing of that performance. Alternatively, Microsoft has formed a joint enterprise with others, to perform the patented method steps in an infringing manner, such as through (1) express or implied agreements, (2) with a common purpose to be carried out, (3) a common pecuniary interest, and (4) with an equal right to a voice in the direction of the enterprise, providing an equal right of control. Additionally, Microsoft profits from the infringement of third parties that perform one or more steps of the patented method and has the right and ability to stop or limit the infringement. For example, insofar as any third party is deemed to perform any step of the asserted claims, Microsoft directs and controls its infringement, profits from its infringement, and has the right and ability to stop or limit its infringement. As such, the performance of any or all steps of the methods claimed by the Sandpiper Patents are directly attributable to Microsoft such that Microsoft directly infringes each asserted claim of the Sandpiper Patents.

Disclosure of Asserted Claims and Infringement Contentions

1. Disclosure Under P.R. 3-1(a)

Sandpiper asserts that the Microsoft infringes at least:

- Claims 28–32, 34, 36, and 37–39 of U.S. Pat. No. 8,478,903 (the "'903 Patent");
- Claims 1–5 and 7–11 of U.S. Pat. No. 8,924,466 (the "'466 Patent");

- Claims 27–29, 31, 33–35, 37–42, 44–50 of U.S. Pat. No. 9,456,053 (the "'053 Patent");
- Claims 1, 2, 5–9, 11, 14, 15, 17–21, and 23 of U.S. Pat. No. 9,722,883 (the "'883 Patent");
- Claims 1, 2, 4, 5, 7, 8, 15, 17, and 18 of U.S. Pat. No. 9,762,692 (the "'692 Patent");
- Claims 1–7 of U.S. Pat. No. 10,701,173 (the "'173 Patent").

Sandpiper alleges that Microsoft infringes each of the asserted claims of the Sandpiper Patents directly (including, for example, through divided and/or joint-direct infringement).

2. Disclosure Under P.R. 3-1(b)

Sandpiper accuses Microsoft of infringement of:

- Each of the Asserted Claims in Exhibit A of the '903 Patent by its Azure CDN service, Azure Front Door service (classic, standard, and premium), and variants thereof serving Microsoft's Services.
- Each of the Asserted Claims in Exhibit B of the '466 Patent by its Azure CDN service, Azure Front Door service (classic, standard, and premium), and variants thereof serving Microsoft's Services.
- Each of the Asserted Claims in Exhibit C of the '053 Patent by its by its Azure CDN service, Azure Front Door service (classic, standard, and premium), and variants thereof serving Microsoft's Services.
- Each of the Asserted Claims in Exhibit D of the '883 Patent by its Azure Service Fabric service.
- Each of the Asserted Claims in Exhibit E of the '692 Patent by its Azure CDN service, Azure Front Door service (classic, standard, and premium), and variants thereof serving Microsoft's Services.

- Each of the Asserted Claims in Exhibit F of the '173 Patent by its Azure CDN service, Azure Front Door service (classic, standard, and premium), and variants thereof serving Microsoft's Services.

These disclosures are based at least in part on Sandpiper's present understanding of the meaning and scope of the claims of the patents-in-suit absent discovery. Discovery in this matter is ongoing and Sandpiper's investigation regarding the grounds contained herein and other grounds of infringement continues. Sandpiper has not yet obtained source code through discovery, which prevents Sandpiper from ascertaining certain details of the Accused Functionalities. Further, Sandpiper has not yet obtained 30(b)(6) deposition testimony on the operation of the Accused Functionalities. This disclosure is given without prejudice to Sandpiper's right to amend or supplement this disclosure as additional facts are ascertained, analysis is made, research is completed, and claims are construed.

3. Disclosure of Under P.R. 3-1(c)

Charts identifying where each element of each asserted claim is found within the Accused Functionalities are in the attached Exhibits. Sandpiper expressly reserves the right to augment and supplement its infringement contentions based upon additional information obtained, including but not limited to source code, through formal discovery.

4. Disclosure Under P.R. 3-1(d)

Sandpiper believes that each element of each asserted claim is literally present in the Accused Functionalities. To the extent any element of any claim is not literally present in any Accused Functionality, Sandpiper contends that each such element is present under the doctrine of equivalents. Sandpiper reserves the right to supplement and/or amend its infringement contentions

regarding the doctrine of equivalents in light of ongoing discovery and/or the Court's claim construction rulings.

5. Disclosure Under P.R. 3-1(e)

Each of the Asserted Claims of the '903 Patent is entitled to a priority date of at least February 10, 1998.

Each of the Asserted Claims of the '466 Patent is entitled to a priority date of at least February 14, 2002.

Each of the Asserted Claims of the '053 Patent is entitled to a priority date of at least December 14, 2011.

Each of the Asserted Claims of the '883 Patent is entitled to a priority date of at least December 13, 2012.

Each of the Asserted Claims of the '692 Patent is entitled to a priority date of at least April 4, 2008.

Each of the Asserted Claims of the '173 Patent is entitled to a priority date of at least December 15, 2014.

Sandpiper reserves its right to amend the priority date of the Patents based on discovery in this case, including any discovery from third parties.

6. **Disclosure Under P.R. 3-1(f)**

Sandpiper does not contend that its own apparatus, product, device, process, method, act, or other instrumentality practices the claimed inventions of the Asserted Patents. Previous assignees of the Asserted Patents may have practiced the claimed inventions.

Document Production Accompanying Disclosure

1. Disclosure Under P.R. 3-2(a)

Sandpiper is not aware of any documents in its possession, custody, or control that describe the subject matter set forth by this Rule.

2. Disclosure Under P.R. 3-2(b)

Sandpiper has produced contemporaneously herewith various documents related to the conception, reduction to practice, design, and development of the claimed invention at the following bates ranges: SANDPIPER-M-00000001 to SANDPIPER-M-00001726.

3. Disclosure Under P.R. 3-2(c)

Sandpiper has produced contemporaneously herewith a copy of the file histories of the patents-in-suit at the following bates ranges: SANDPIPER-M-00001727 TO SANDPIPER-M-00005200.

CONCLUSION

Consistent with the Federal Rules of Civil Procedure and the local rules of this Court, Sandpiper expressly reserves the right to supplement this disclosure, including the right to supplement its disclosure of asserted claims and infringement contentions, as discovery is ongoing and additional information regarding Microsoft's acts of infringement are likely to be uncovered through the course of discovery.

Dated: September 16, 2025

Respectfully Submitted,

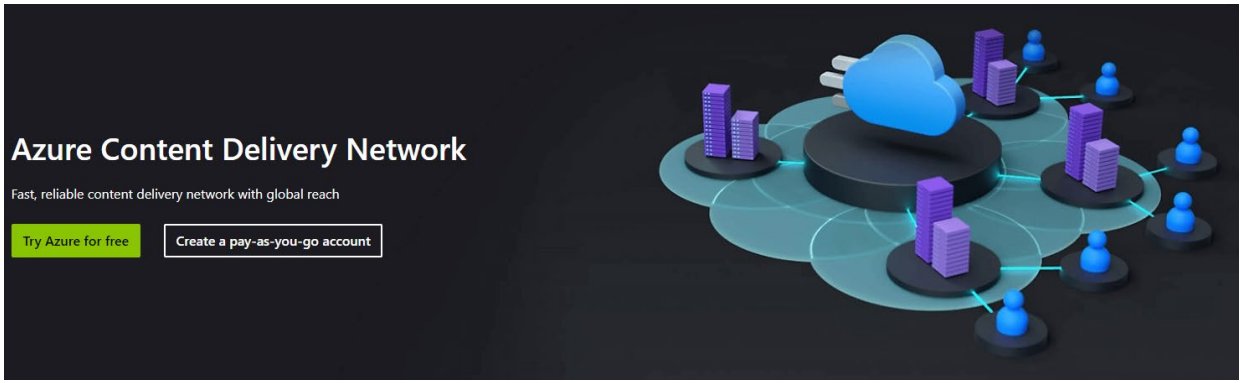
/s/ Ryan D. Dykal
Ryan D. Dykal (MO State Bar No. 60905)
Mark D. Schafer (*pro hac vice*)
Phillip Eckert (*pro hac vice*)
Carolyn Drell (*pro hac vice*)
BOIES SCHILLER FLEXNER LLP
1401 New York Ave, NW
Washington, DC 20005
(t): 202-274-1109
rdykal@bsfllp.com
mschafer@bsfllp.com
peckert@bsfllp.com
cdrell@bsfllp.com
Counsel for Plaintiff Sandpiper CDN, LLC

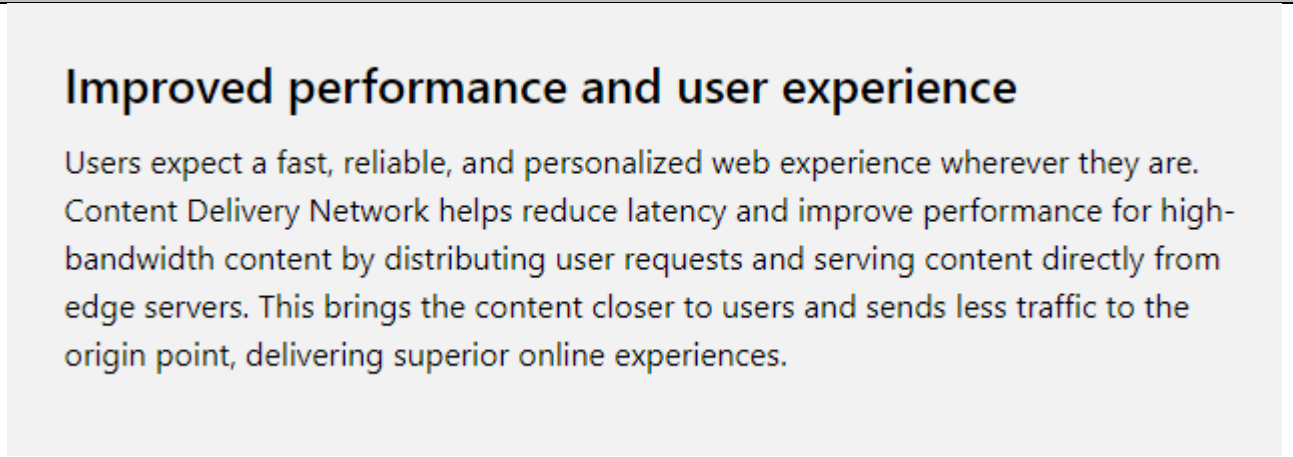
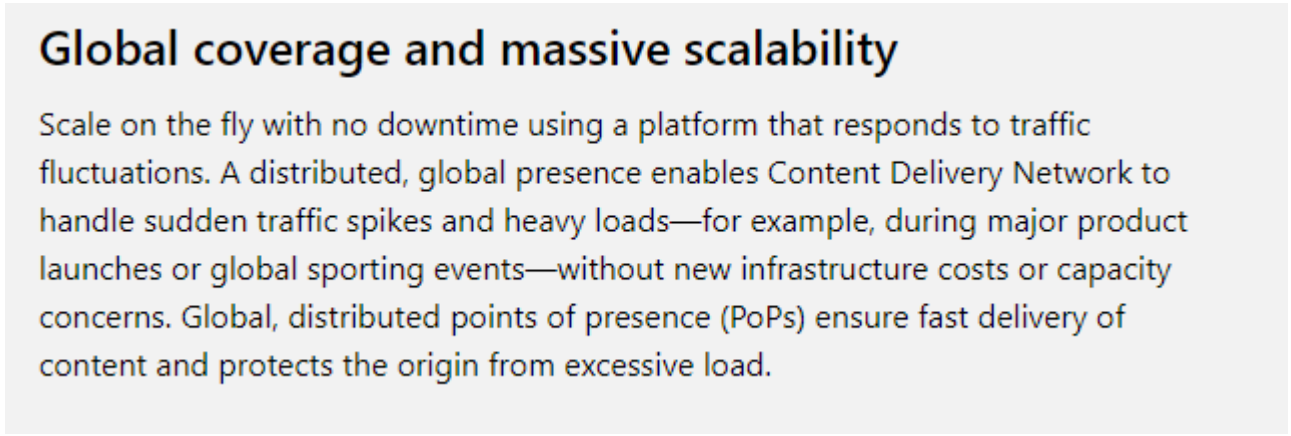
CERTIFICATE OF SERVICE

Pursuant to the Federal Rules of Civil Procedure and Local Rule CV-5, I hereby certify that on September 16, 2025 I electronically produced the foregoing with all counsel of record by electronic mail.

/s/ Ryan Dykal

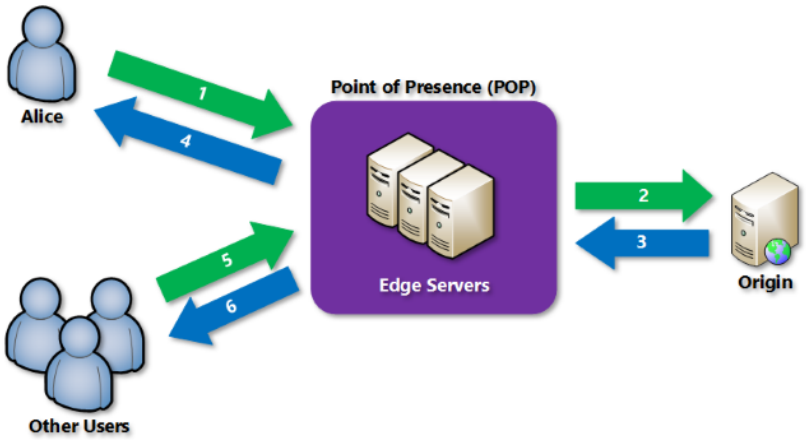
EXHIBIT A

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
28[Pre]. A method, in a content delivery system operative in a computer network for delivering content to client machines, the computer network comprising a plurality of origin servers, each of said origin servers having resources associated therewith, and the content delivery system comprising at least one shared repeater server operable to replicate resources associated with the plurality of origin servers, the method comprising:	Microsoft performs a method in a content delivery system operative in a computer network for delivering content to client machines, the computer network comprising a plurality of origin servers, each of said origin servers having resources associated therewith, and the content delivery system comprising at least one shared repeater server operable to replicate resources associated with the plurality of origin servers as set forth below. For instance, Microsoft Azure CDN and/or Azure Front Door comprises a plurality of origin servers hosting associated resources as well as shared repeater servers (e.g., Edge Nodes or Endpoints or Points of Presence) to replicate resources associated with the origin servers.  https://azure.microsoft.com/en-us/products/cdn **Increase speed and scalability for an optimal user experience** Azure Content Delivery Network offers a global solution for rapidly delivering content. Save bandwidth and improve responsiveness when encoding or distributing gaming software, firmware updates, and IoT endpoints. Reduce load times for websites, mobile apps, and streaming media to increase user satisfaction globally. https://azure.microsoft.com/en-us/products/cdn Stream media and download large files quickly with optimized delivery. Provide a scalable solution to handle worldwide traffic spikes and instantaneous high loads. Integrate seamlessly with your Azure services to activate within minutes. Protect content with custom domain HTTPS, DDoS, and WAF protection. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	 https://azure.microsoft.com/en-us/products/cdn  https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Protection at the edge Optimize security through features like custom domain HTTPS and DDoS mitigation. Enable Content Delivery Network to put edge servers on the front line serving and protect the origin server from malicious attacks. Set advanced rules to block requests or validate users and use geo-filtering to ensure the content is delivered only to specific regions. https://azure.microsoft.com/en-us/products/cdn **Automate processes and workflows with DevOps** Azure APIs and developer tools give you the performance, reliability, and security needed to build and automate global applications fast. A unified API allows you to deploy Content Delivery Network endpoints from each provider and still manage them the same way. Azure APIs also provide the flexibility to optimize features for availability and performance. https://azure.microsoft.com/en-us/products/cdn

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

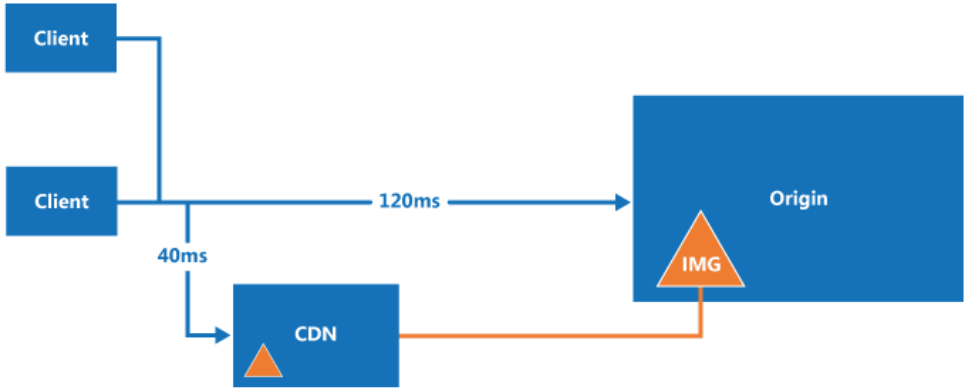
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	This article lists current metros containing point of presence (POP) locations, sorted by region, for Azure Content Delivery Network products. Each metro might contain more than one POP. For example, Azure Content Delivery Network from Microsoft has 192 POPs across 109 metro cities. **ⓘ Important** Each Azure Content Delivery Network product has a distinct way of building its content delivery network infrastructures, hence Microsoft recommends against using POP locations to decide which Azure Content Delivery Network product to use. Instead, you should consider its features and end-user performance. Test the performance with each Azure Content Delivery Network product to choose the right product for your users. https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

Region	Cities
North America	Etobicoke, Canada (2) Montreal, Canada Vancouver, Canada (2) Querétaro, Mexico (2) Atlanta, GA, USA (3) Boydton, VA, USA (2) Chaska, MN, USA (2) Cheyenne, WY, USA (2) Chicago, IL, USA (4) Dallas, TX, USA (4) Des Moines,IA, USA (3) Detroit, MI, USA Englewood, CO, USA (2) Honolulu, HI, USA Houston, TX, USA (3) Jacksonville, FL, USA (2) Las Vegas, NV, USA (3) Los Angeles, CA, USA (2) Manassas, VA, USA (3) Memphis, TN, USA Miami, FL, USA (4) Minneapolis, MN, USA (2) Needham Heights, MA, USA (2) Nashville, TN, USA Newark, NJ, USA New York, NY, USA (2) Philadelphia, PA, USA Phoenix, AZ, USA Plano, TX, USA Portland, OR, USA Quincy, WA, USA (3) San Antonio, TX, USA (4) San Jose, CA, USA (4) Salt Lake City, UT, USA (3) Seattle, WA, USA (2) Secaucus, NJ, USA (2) Southfield, MI, USA Tempe, AZ, USA

https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Edge Nodes - List

Reference

Feedback

Service: CDN

API Version: 2024-02-01

Edgenodes are the global Point of Presence (POP) locations used to deliver CDN content to end users.

HTTP Copy Try It

```
GET https://management.azure.com/providers/Microsoft.Cdn/edgenodes?api-version=2024-02-01
```

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

EdgeNode

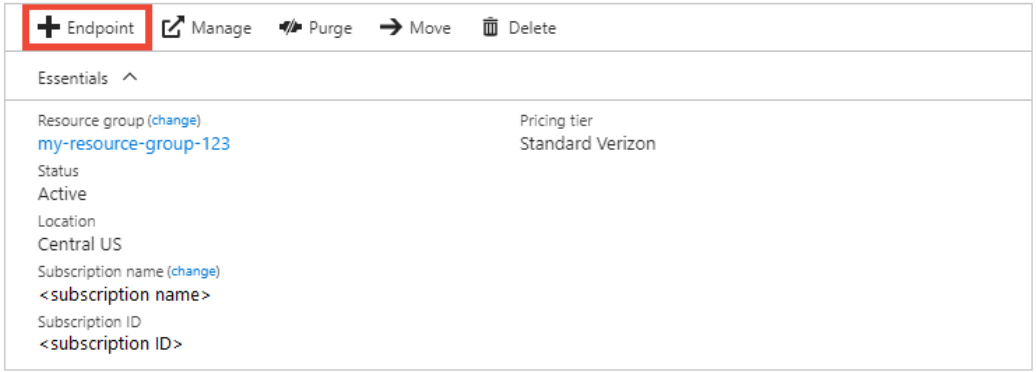
Edgenode is a global Point of Presence (POP) location used to deliver CDN content to end users.

Expand table

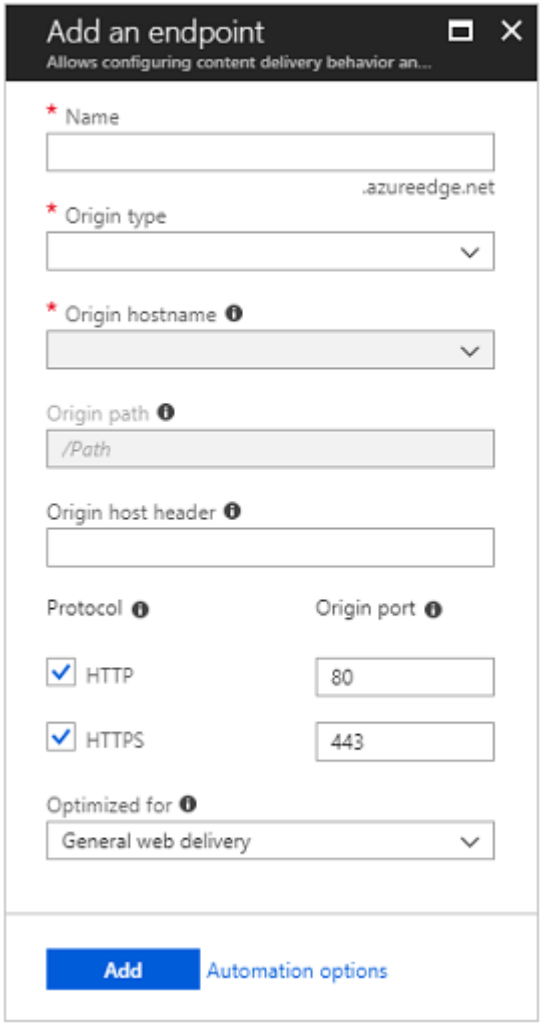
Name	Type	Description
id	string	Resource ID.
name	string	Resource name.
properties.ipAddressGroups	IpAddressGroup[]	List of ip address groups.
systemData	SystemData	Read only system data
type	string	Resource type.

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Create an Azure Content Delivery Network endpoint Article • 03/21/2024 • 6 contributors Feedback **In this article** Prerequisites Sign in to the Azure portal Create a new content delivery network endpoint Clean up resources Next steps This article describes all the settings for creating an Azure Content Delivery Network endpoint in an existing content delivery network profile. After you've created a profile and an endpoint, you can start delivering content to your customers. For a quickstart on creating a profile and endpoint, see Quickstart: Create an Azure Content Delivery Network profile and endpoint. https://learn.microsoft.com/en-us/azure/cdn/cdn-create-endpoint-how-to

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Create a new content delivery network endpoint 1. In the Azure portal, navigate to your content delivery network profile. You might have pinned it to the dashboard in the previous step. If not, you can find it by selecting **All services**, then selecting **CDN profiles**. In the **CDN profiles** pane, select the profile to which you plan to add your endpoint. The content delivery network profile pane appears. 2. Select **Endpoint**.  https://learn.microsoft.com/en-us/azure/cdn/cdn-create-endpoint-how-to

The **Add an endpoint** page appears.



https://learn.microsoft.com/en-us/azure/cdn/cdn-create-endpoint-how-to

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

3. For **Name**, enter a unique name for the new content delivery network endpoint. This name is used to access your cached resources at the domain <*endpointname*>.azureedge.net.

4. For **Origin type**, choose one of the following origin types:

   - **Storage** for Azure Storage
   - **Storage static website** for Azure Storage static websites
   - **Cloud service** for Azure Cloud Services
   - **Web App** for Azure Web Apps
   - **Custom origin** for any other publicly accessible origin web server (hosted in Azure or elsewhere)

5. For **Origin hostname**, select or enter your origin server domain. The dropdown lists all available origin servers of the type you specified in step 4. If you selected **Custom origin** as your origin type, enter the domain of your custom origin server.

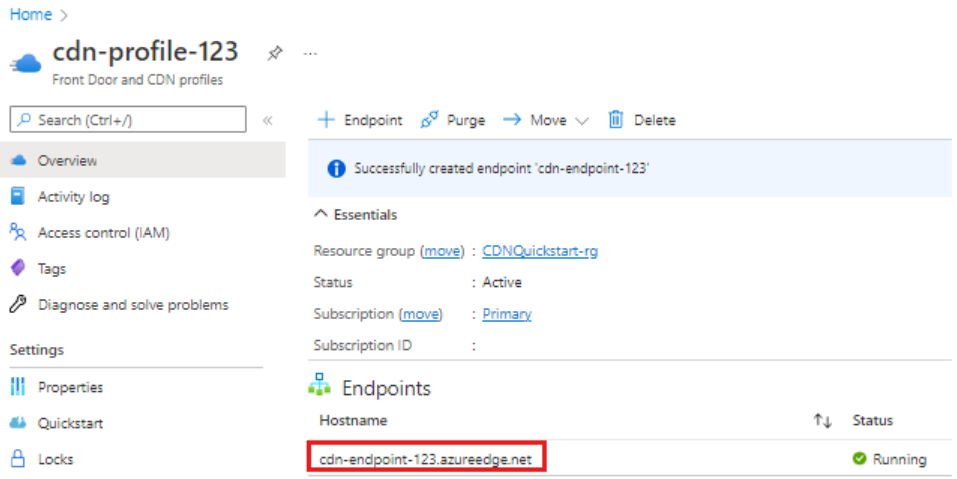
6. For **Origin path**, enter the path to the resources that you want to cache. To allow caching of any resource at the domain you specified in step 5, leave this setting blank.

7. For **Origin host header**, enter the host header you want Azure Content Delivery Network to send with each request, or leave the default.

   ⓘ Note

   Some types of origins, such as Azure Storage and Web Apps, require the host header to match the domain of the origin. Unless you have an origin that requires a host header different from its domain, you should leave the default value.

https://learn.microsoft.com/en-us/azure/cdn/cdn-create-endpoint-how-to

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	10. Select **Add** to create the new endpoint. After the endpoint is created, it appears in the list of endpoints for the profile.  Because it takes time for the registration to propagate, the endpoint isn't immediately available for use: • For **Azure CDN Standard from Microsoft** profiles, propagation usually completes in 10 minutes. • For **Azure CDN Standard from Edgio** and **Azure CDN Premium from Edgio** profiles, propagation usually completes within 30 minutes. https://learn.microsoft.com/en-us/azure/cdn/cdn-create-endpoint-how-to

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Microsoft Azure CDN additionally includes products such as Azure Front Door as well as Standard and Premium CDN services from Edgio (formerly Verizon). Azure Content Delivery Network includes three products: • Azure CDN Standard from Microsoft • Azure CDN Standard from Edgio (formerly Verizon) • Azure CDN Premium from Edgio (formerly Verizon). https://learn.microsoft.com/en-us/azure/cdn/cdn-features

Performance features and optimizations	Standard Microsoft	Standard Edgio	Premium Edgio
Dynamic site acceleration	Offered via Azure Front Door	✓	✓
General web delivery optimization	✓	✓	✓
Video streaming optimization	via General Web Delivery	via General Web Delivery	via General Web Delivery
Large file optimization	via General Web Delivery	via General Web Delivery	via General Web Delivery
Origin Port	All TCP ports	All TCP ports	All TCP ports
Global server load balancing (GSLB)	✓	✓	✓
Fast purge	✓	✓	✓
Asset pre-loading		✓	✓
Cache/header settings (using caching rules)	✓ using Standard rules engine	✓	
Customizable, rules based content delivery engine	✓ using Standard rules engine		✓ using rules engine
Cache/header settings	✓ using Standard rules engine		✓ using Premium rules engine
URL redirect/rewrite	✓ using Standard rules engine		✓ using Premium rules engine
Mobile device rules	✓ using Standard rules engine		✓ using Premium rules engine
Query string caching	✓	✓	✓
Internet Protocol version 4 (IPv4)/Internet Protocol version 6 (IPv6) dual-stack	✓	✓	✓
HTTP/2 support	✓	✓	✓

https://learn.microsoft.com/en-us/azure/cdn/cdn-features

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Microsoft also performs the methods of the asserted claims of the '903 Patent to serve Microsoft resources and services using the same or similar infrastructure. For example, Microsoft uses internal CDN technologies such as its Azure CDN to support at least Xbox gaming, Office 365, Windows updates, and web resources like Linkedin and Bing. • Built on Microsoft's massive-scale private global network, Azure Front Door is a proven platform used to power some of the largest and latency sensitive global services at Microsoft, such as Microsoft Office 365, Bing, LinkedIn, and Xbox. https://azure.microsoft.com/en-us/blog/azure-front-door-enhances-secure-cloud-cdn-with-intelligent-threat-protection/ Azure CDN provides a world class platform to let you reduce load times, save bandwidth, and speed responsiveness across your businesses diverse workflows. Azure CDN from Microsoft enables Azure customers to use and deliver content from the same global CDN network leveraged by Microsoft properties such as Office 365, Skype, Bing, OneDrive, Windows, and Xbox. Connectivity within Microsoft's network along with new Regional Caching capabilities enables more consistent, more predictable cache fill performance by providing multi-tier caching along direct, private access to content in Azure from each CDN edge point of presence (POP). Azure CDN from Microsoft entered public preview providing access to 54 global Edge POPs in 33 countries and 16 Regional Cache POPs. https://azure.microsoft.com/en-us/blog/announcing-microsoft-s-own-cdn-network/

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

What powers Windows 365?

Not surprising, Microsoft's own cloud computing platform, Azure. Specifically, **Azure Virtual Desktop**, which can be used to create Windows virtual machines on Redmond's servers, then stream those desktops to PC and non-PC devices.

https://www.computerworld.com/article/1614540/what-is-windows-365-microsoft-s-cloud-based-os-explained.html

Use the Office 365 Content Delivery Network (CDN) with SharePoint Online

02/28/2025

You can use the built-in Office 365 Content Delivery Network (CDN) to host static assets to provide better performance for your SharePoint Online pages. The Office 365 CDN improves performance by caching static assets closer to the browsers requesting them, which helps to speed up downloads and reduce latency. Also, the Office 365 CDN uses the HTTP/2 protocol for improved compression and HTTP pipelining. The Office 365 CDN service is included as part of your SharePoint Online subscription.

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	The Office 365 CDN is composed of multiple CDNs that allow you to host static assets in multiple locations, or *origins*, and serve them from global high-speed networks. Depending on the kind of content you want to host in the Office 365 CDN such as you can add **public** origins. If you're already familiar with the way that CDNs work, you only need to complete a few steps to enable the Office 365 CDN for your tenant. This article describes how. Read on for information about how to get started hosting your static assets. https://learn.microsoft.com/en-us/microsoft-365/enterprise/use-microsoft-365-cdn-with-spo?view=o365-worldwide On the Microsoft side, the company uses primarily Azure Front Door a cloud CDN platform and some third-party CDNs for some static delivery of content i.e Xbox downloads, Windows Updates. Microsoft does not https://www.streamingmediablog.com/2020/10/gaming-console-launch.html

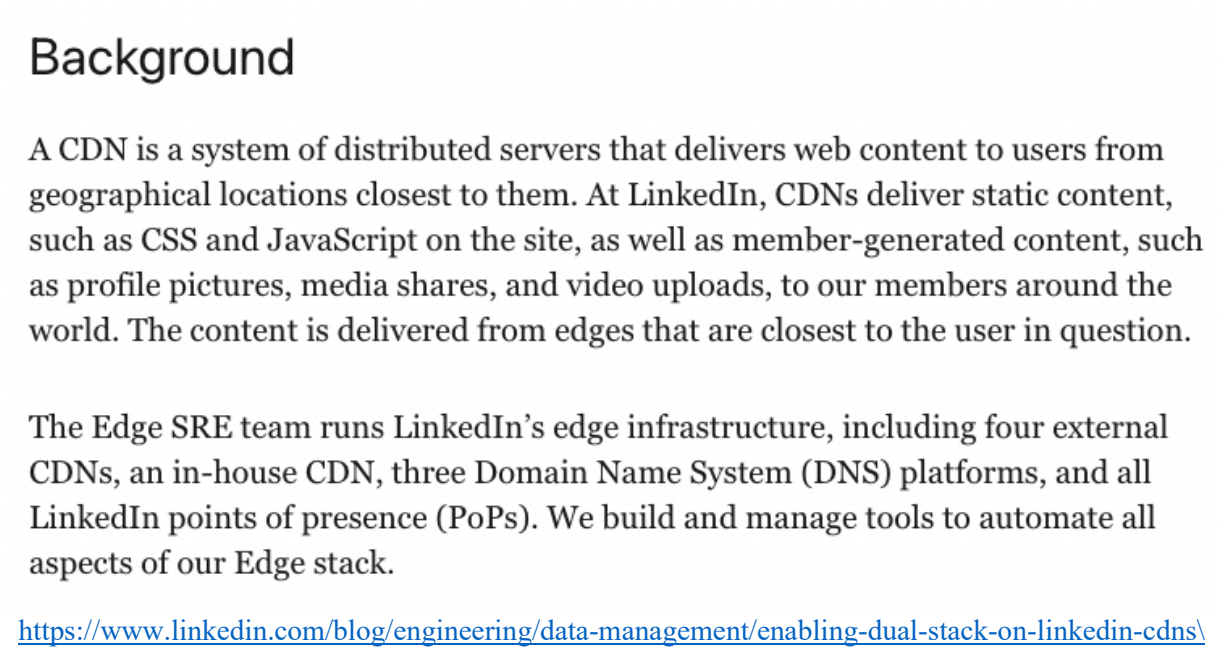
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Teams is built on a microservices architecture, with a few hundred microservices working cohesively to deliver our product's many features including messaging, meetings, files, calendar, and apps. Using microservices helps each of our component teams to work and release their changes independently.

Azure is the cloud platform that underpins all of Microsoft's cloud services, including Microsoft Teams. Our workloads run in Azure virtual machines (VMs), with our older services being deployed through Azure Cloud Services and our newer ones on Azure Service Fabric. Our primary storage stack is Azure Cosmos DB, with some services using Azure Blob Storage. We count on Azure Cache for Redis for increased throughput and resiliency. We leverage Traffic Manager and Azure Front Door to route traffic where we want it to be. We use Queue Storage and Event Hubs to communicate, and we depend on Azure Active Directory to manage our tenants and users.

- **Resiliency-optimized caches**: We leverage caches between our components extensively, for both performance and resiliency. Caches help reduce average latency and provide a source of data in case a downstream service is unavailable. Keeping data in caches for a long time introduces data freshness issues yet keeping data in caches for a long time is the best defense against downstream failures. We focus on Time to Refresh (TTR) to our cache data as well as Time to Live (TTL). By setting a long TTL and a shorter TTR value, we can fine-tune how fresh to keep our data versus how long we want data to stick around whenever a downstream dependency fails.

https://azure.microsoft.com/en-us/blog/advancing-microsoft-teams-on-azure-operating-at-pandemic-scale/

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	 https://www.linkedin.com/blog/engineering/data-management/enabling-dual-stack-on-linkedin-cdns\

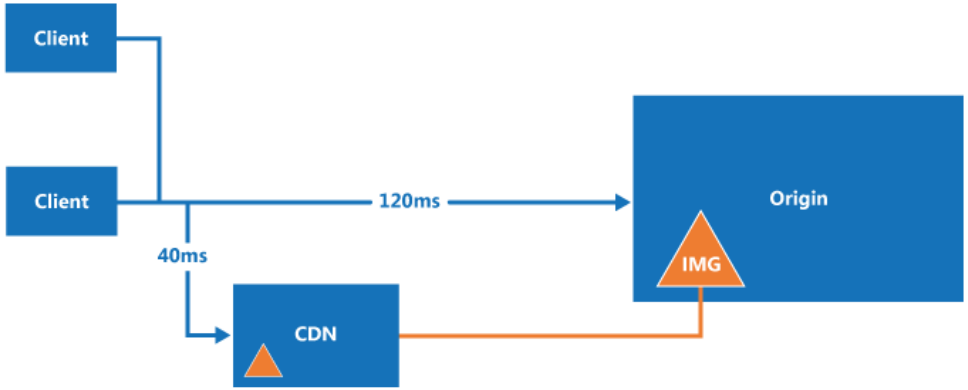
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	LinkedIn recently described how it uses the edge to support the popular professional networking web site. The engineering team wrote a blog post to talk about edge deployments and its migration away from an in-house solution to Azure Edge. LinkedIn is one of many large web properties that is owned by a hyperscaler cloud provider — Microsoft, in this case — but doesn't actually use the in-house cloud. YouTube is another example of this — it has yet to be migrated to GCP. LinkedIn actually had its own quite extensive in-house CDN, with nineteen PoPs (points of presence) in ten countries. This is a common practice for major content owners as it defrays transport costs and improves performance. The nodes can cache content and perform tasks such as SSL offloading. But as LinkedIn noted, Microsoft's CDN – Azure Front Door – gives access to 300 PoPs around the globe, which is a big step up in terms of reach. https://www.edgeir.com/how-linkedin-is-using-azure-front-door-for-edge-build-out-20220306
[28.a] associating the at least one repeater server with a first alias name, wherein requests for a first resource located on a first origin server are directed, based at least in part on said first alias name, to the at least one repeater server for delivery of the first resource from said at least one repeater server;	Microsoft associates the at least one repeater server with a first alias name, wherein requests for a first resource located on a first origin server are directed, based at least in part on said first alias name, to the at least one repeater server for delivery of the first resource from said at least one repeater server. For example, Microsoft provides Point of Presence Edge Servers associated with alias names, and directs requests for resources located on an origin server based at least in part on this alias name. These requests may be for resources of third-party customers of Azure CDN and/or Azure Front Door services, or Microsoft internal resources.

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

How and why a CDN is used

Typical uses for a CDN include:

- Delivering static resources for client applications, often from a website. These resources can be images, style sheets, documents, files, client-side scripts, HTML pages, HTML fragments, or any other content that the server does not need to modify for each request. The application can create items at runtime and make them available to the CDN (for example, by creating a list of current news headlines), but it does not do so for each request.
- Delivering public static and shared content to devices such as mobile phones and tablet computers. The application itself is a web service that offers an API to clients running on the various devices. The CDN can also deliver static datasets (via the web service) for the clients to use, perhaps to generate the client UI. For example, the CDN could be used to distribute JSON or XML documents.
- Serving entire websites that consist of only public static content to clients, without requiring any dedicated compute resources.
- Streaming video files to the client on demand. Video benefits from the low latency and reliable connectivity available from the globally located datacenters that offer CDN connections. Microsoft Azure Media Services integrates with Azure Content Delivery Network to deliver content directly to the CDN for further distribution. For more information, see Streaming endpoints overview.
- Generally improving the experience for users, especially those located far from the datacenter hosting the application. These users might otherwise suffer higher latency. A large proportion of the total size of the content in a web application is often static, and using the CDN can help to maintain performance and overall user experience while eliminating the requirement to deploy the application to multiple datacenters. For a list of Azure Content Delivery Network node locations, see Azure CDN POP Locations.
- Supporting IoT (Internet of Things) solutions. The huge numbers of devices and appliances involved in an IoT solution could easily overwhelm an application if it had to distribute firmware updates directly to each device.
- Coping with peaks and surges in demand without requiring the application to scale, avoiding the consequent increase in running costs. For example, when an update to an operating system is released for a hardware device such as a specific model of router, or for a consumer device such as a smart TV, there will be a huge peak in demand as it is downloaded by millions of users and devices over a short period.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Cache control Consider how to manage caching within the system. For example, in Azure Content Delivery Network, you can set global caching rules, and then set custom caching for particular origin endpoints. You can also control how caching is performed in a CDN by sending cache-directive headers at the origin. For more information, see How caching works. To prevent objects from being available on the CDN, you can delete them from the origin, remove or delete the CDN endpoint, or in the case of blob storage, make the container or blob private. However, items are not removed from the CDN until the time to live expires. You can also manually purge a CDN endpoint. https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Tutorial: Add a custom domain to your endpoint

Article • 03/21/2024 • 22 contributors

Feedback

In this article

- Prerequisites
- Create a CNAME DNS record
- Add a custom domain to your content delivery network endpoint
- Verify the custom domain

Show 2 more

This tutorial shows how to add a custom domain to an Azure Content Delivery Network endpoint.

The endpoint name in your content delivery network profile is a subdomain of azureedge.net. By default when delivering content, the content delivery network profile domain gets included in the URL.

For example, `https://contoso.azureedge.net/photo.png`.

Azure Content Delivery Network provides the option of associating a custom domain with a content delivery network endpoint. This option delivers content with a custom domain in your URL instead of the default domain.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Prerequisites

- Before you can complete the steps in this tutorial, create a content delivery network profile and at least one content delivery network endpoint.
  - For more information, see Quickstart: Create an Azure Content Delivery Network profile and endpoint.
- If you don't have a custom domain, purchase one with a domain provider.
  - For more information, see Buy a custom domain name.
- If you're using Azure to host your DNS domains, delegate your custom domain to Azure DNS.
  - For more information, see Delegate a domain to Azure DNS.
- If you're using a domain provider to handle your DNS domain, continue to Create a CNAME DNS record.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

Create a CNAME DNS record

Before you can use a custom domain with an Azure Content Delivery Network endpoint, you must first create a canonical name (CNAME) record with Azure DNS or your DNS provider to point to your content delivery network endpoint.

A CNAME record is a DNS record that maps a source domain name to a destination domain name.

For Azure Content Delivery Network, the source domain name is your custom domain name and the destination domain name is your content delivery network endpoint hostname.

Azure Content Delivery Network routes traffic addressed to the source custom domain to the destination content delivery network endpoint hostname after it verifies the CNAME record.

A custom domain and its subdomain can only get added to a single endpoint at a time.

Use multiple CNAME records for different subdomains from the same custom domain for different Azure services.

You can map a custom domain with different subdomains to the same content delivery network endpoint.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Microsoft associates alias records with Azure CDN endpoints for users of both Azure DNS and third-party DNS providers.

Azure DNS | DNS Provider

Azure DNS uses alias records for Azure resources within the same subscription.

To add an alias record for your Azure Content Delivery Network endpoint:

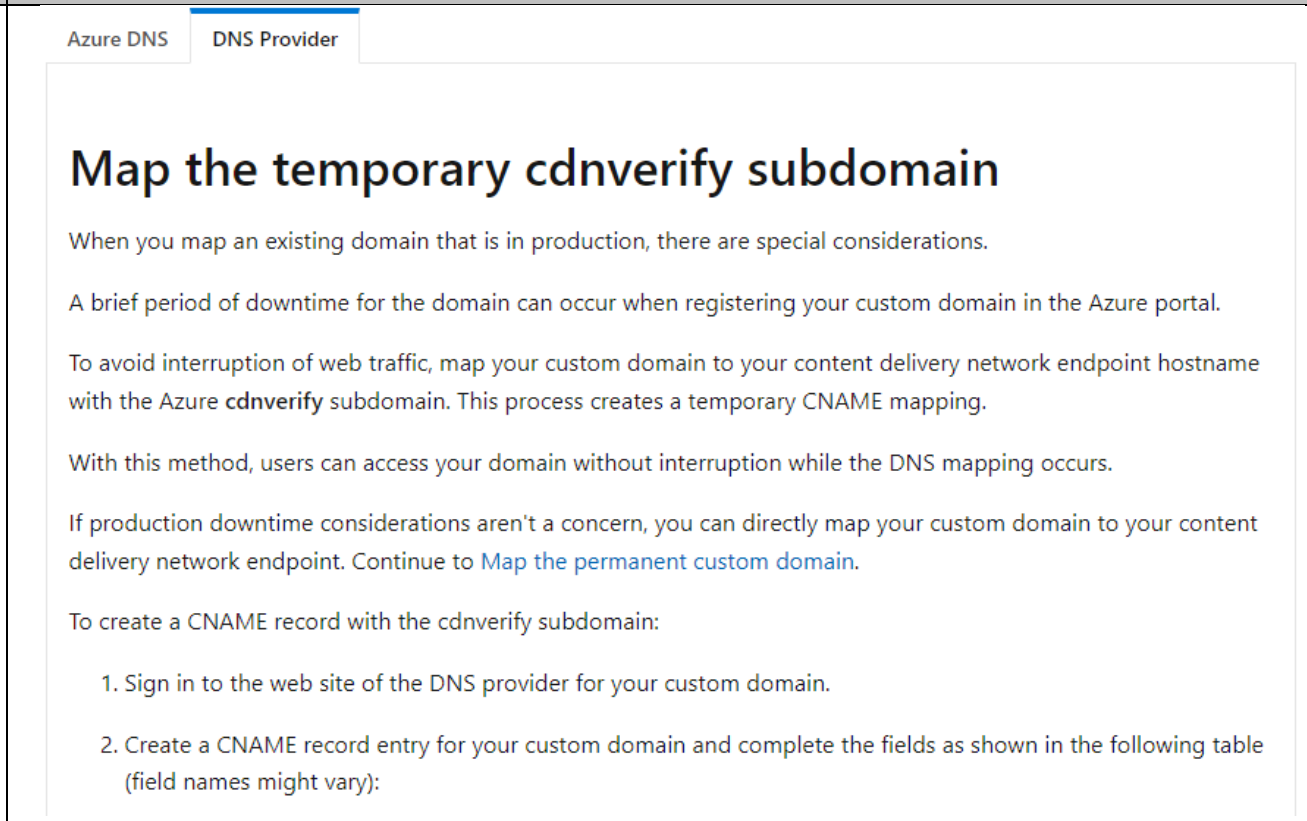
1. Sign in to the Azure portal.
2. In the left-hand menu, select **All resources** then the Azure DNS zone for your custom domain.
3. In the DNS zone for your custom domain, select + **Record set**.
4. In **Add record set** enter or select the following information:

Expand table

Setting	Value
Name	Enter alias you want to use for your content delivery network endpoint. For example, **www**.
Type	Select **CNAME**.
Alias record set	Select **Yes**.
Alias type	Select **Azure resource**.
Choose a subscription	Select your subscription.
Azure resource	Select your Azure Content Delivery Network endpoint.

5. Change the **TTL** for the record to your value.
6. Select **OK**.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	 https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Expand table

Source	Type	Destination
cdnverify.www.contoso.com	CNAME	cdnverify.contoso.azureedge.net

- Source: Enter your custom domain name, including the cdnverify subdomain, in the following format: **cdnverify.<custom-domain-name>**
  - For example: **cdnverify.www.contoso.com**
- Type: Enter or select **CNAME**.
- Destination: Enter your content delivery network endpoint hostname, including the cdnverify subdomain, in the following format: **cdnverify.<endpoint-name>.azureedge.net**.
  - For example: **cdnverify.contoso.azureedge.net**

3. Save your changes.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Map the permanent custom domain

In this section, you map the permanent custom domain to the content delivery network endpoint.

To create a CNAME record for your custom domain:

1. Sign in to the web site of the domain provider for your custom domain.
2. Create a CNAME record entry for your custom domain and complete the fields as shown in the following table (field names might vary):

Expand table

Source	Type	Destination
www.contoso.com	CNAME	contoso.azureedge.net

- Source: Enter your custom domain name.
  - For example: **www.contoso.com**
- Type: Enter or select **CNAME**.
- Destination: Enter your content delivery network endpoint hostname in the following format: **<endpoint-name>.azureedge.net**.
  - For example: **contoso.azureedge.net**

3. Save your changes.
4. If you previously created a temporary cdnverify subdomain CNAME record, delete it.
5. If you're using this custom domain in production for the first time, follow the steps for Add the custom domain with your content delivery network endpoint and Verify the custom domain.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

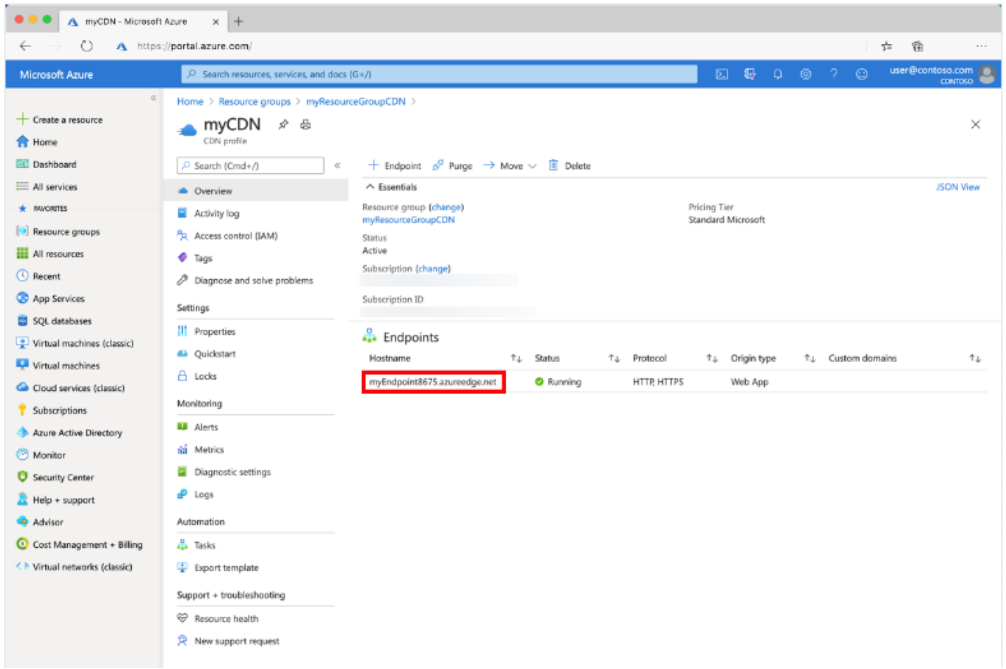
Add a custom domain to your content delivery network endpoint

After you've registered your custom domain, you can then add it to your content delivery network endpoint.

Azure portal | PowerShell

1. Sign in to the Azure portal and browse to the content delivery network profile containing the endpoint that you want to map to a custom domain.

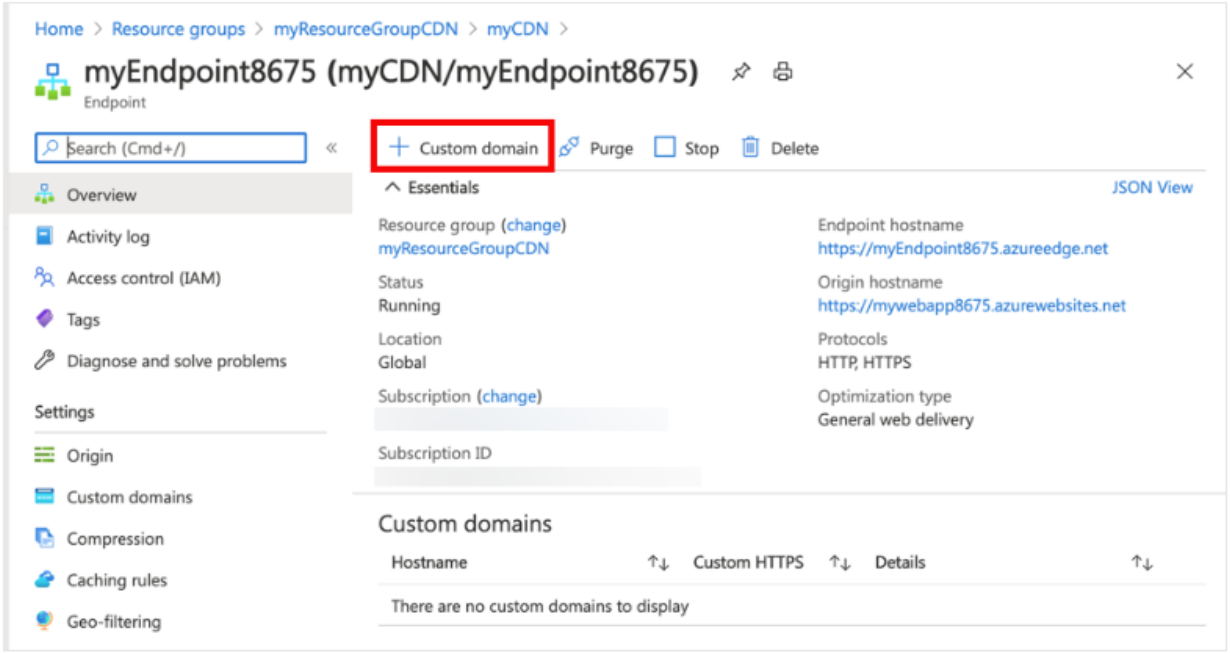
2. On the **CDN profile** page, select the content delivery network endpoint to add the custom domain.



https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

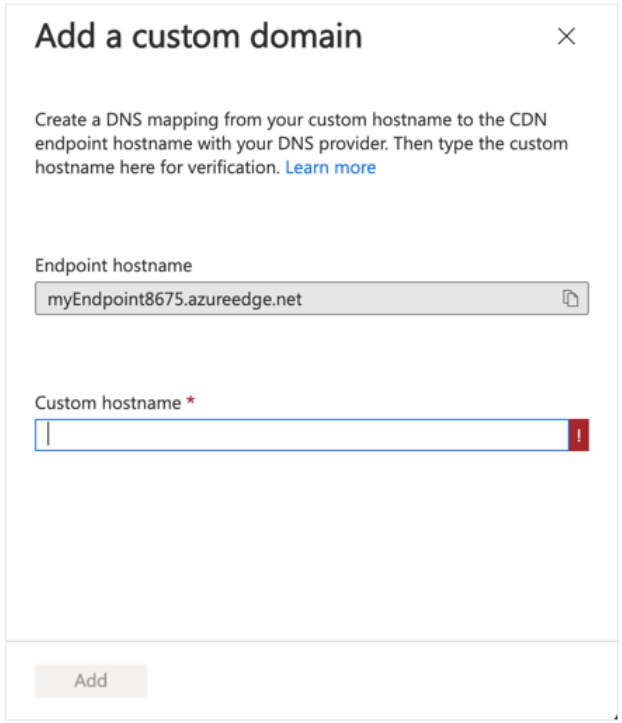
3. Select + **Custom domain**.



4. In **Add a custom domain**, **Endpoint hostname**, gets generated and pre-filled from your content delivery network endpoint URL: <**endpoint-hostname**>.azureedge.net. You can't change this value.

5. For **Custom hostname**, enter your custom domain, including the subdomain, to use as the source domain of your CNAME record.
   a. For example, www.contoso.com or **cdn.contoso.com**. **Don't use the cdnverify subdomain name.**

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain



6. Select **Add**.

Azure verifies that the CNAME record exists for the custom domain name you entered. If the CNAME is correct, your custom domain gets validated.

It can take some time for the new custom domain settings to propagate to all content delivery network edge nodes:

- For **Azure CDN Standard from Microsoft** profiles, propagation usually completes in 10 minutes.
- For **Azure CDN Standard from Edgio** and **Azure CDN Premium from Edgio** profiles, propagation usually completes in 10 minutes.

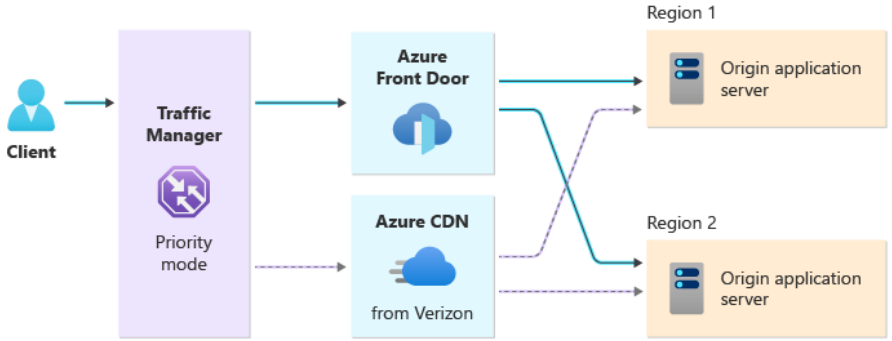
https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Requests for resources located on a first origin server are directed based at least in part on the alias name associated with the shared repeater server. For instance, Microsoft Azure CDN supports custom domain alias creation for multiple CDN hosting clients accessing content from various origin servers cached by Azure CDN endpoints. **Verify the custom domain** After you've completed the registration of your custom domain, verify that the custom domain references your content delivery network endpoint. 1. Ensure that you have public content that you want cached at the endpoint. For example, if your content delivery network endpoint is associated with a storage account, Azure Content Delivery Network caches the content in a public container. Set your container to allow public access and it contains at least one file to test the custom domain. 2. In your browser, navigate to the address of the file by using the custom domain. For example, if your custom domain is `www.contoso.com`, the URL to the cached file should be similar to the following URL: `http://www.contoso.com/my-public-container/my-file.jpg`. Verify that the result is that same as when you access the content delivery network endpoint directly at <**endpoint-hostname**>.azureedge.net. https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain Microsoft Azure additionally integrates Edgio (formerly Verizon) CDN features via Azure Front Door. Caching is a common way to reduce load on the backend services and optimize performance for users. Content delivery networks (CDNs), including Azure Front Door, provide caching at the network edge. Mission-critical workloads often use multiple CDNs to achieve a higher level of uptime. If one CDN experiences outage or degraded performance, your traffic is automatically diverted to another CDN. If you implement multiple CDNs, consider the implications of this approach. Each CDN provides a separate network path to your application servers, and you need to configure and test each CDN separately. This article describes an approach for using Azure Front Door with a partner CDN, Verizon. This approach is suitable for solutions that rely heavily on caching for delivering static content delivery, media, and high-scale eCommerce applications. https://learn.microsoft.com/en-us/azure/architecture/guide/networking/global-web-applications/mission-critical-content-delivery

Approach

Verizon's CDN and the CDN platform (Edgio) can be integrated into your Azure solution. You can configure it from the Azure portal and APIs. The platform is isolated from Microsoft's infrastructure.

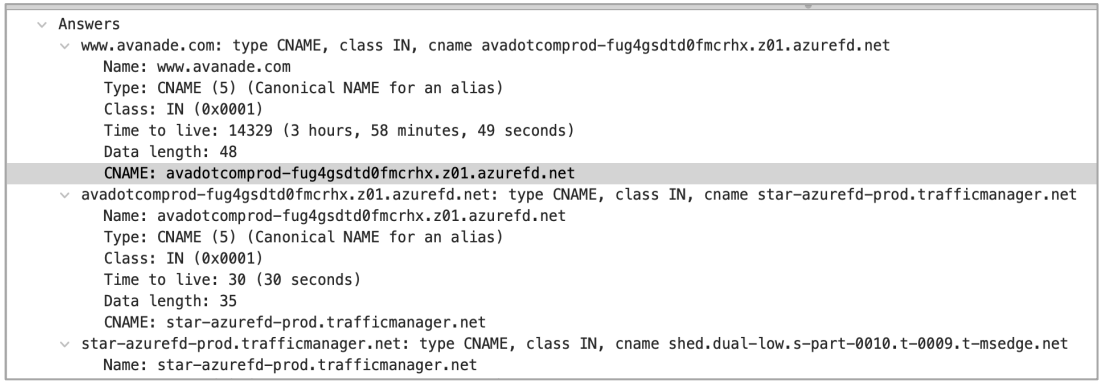
This isolation provides a high degree of resiliency from disaster scenarios. If an outage or disaster occurs, traffic is automatically shifted between Azure Front Door and Verizon's CDN. You can use Azure Traffic Manager to detect an outage and redirect traffic to the alternative CDN.



- **Traffic Manager using priority routing mode** has two endpoints. By default, Traffic Manager sends requests through Azure Front Door. If Azure Front Door is unavailable, Traffic Manager sends the request through the partner CDN instead.
- **Azure Front Door** processes and routes most of your application traffic. Azure Front Door routes traffic to the appropriate origin application server, and it provides the primary path to your application. If Azure Front Door is unavailable, traffic is automatically redirected through the secondary path.
- **Azure CDN from Verizon** is configured to send traffic to each origin server.
- **Your origin application servers** need to be ready to accept traffic from both Azure Front Door and Azure CDN from Verizon, at any time.

https://learn.microsoft.com/en-us/azure/architecture/guide/networking/global-web-applications/mission-critical-content-delivery

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities



DNS query response for avanade.com returns avadotcomprod-fug4gsdtd0fmcrhx.z01.azurefd.net

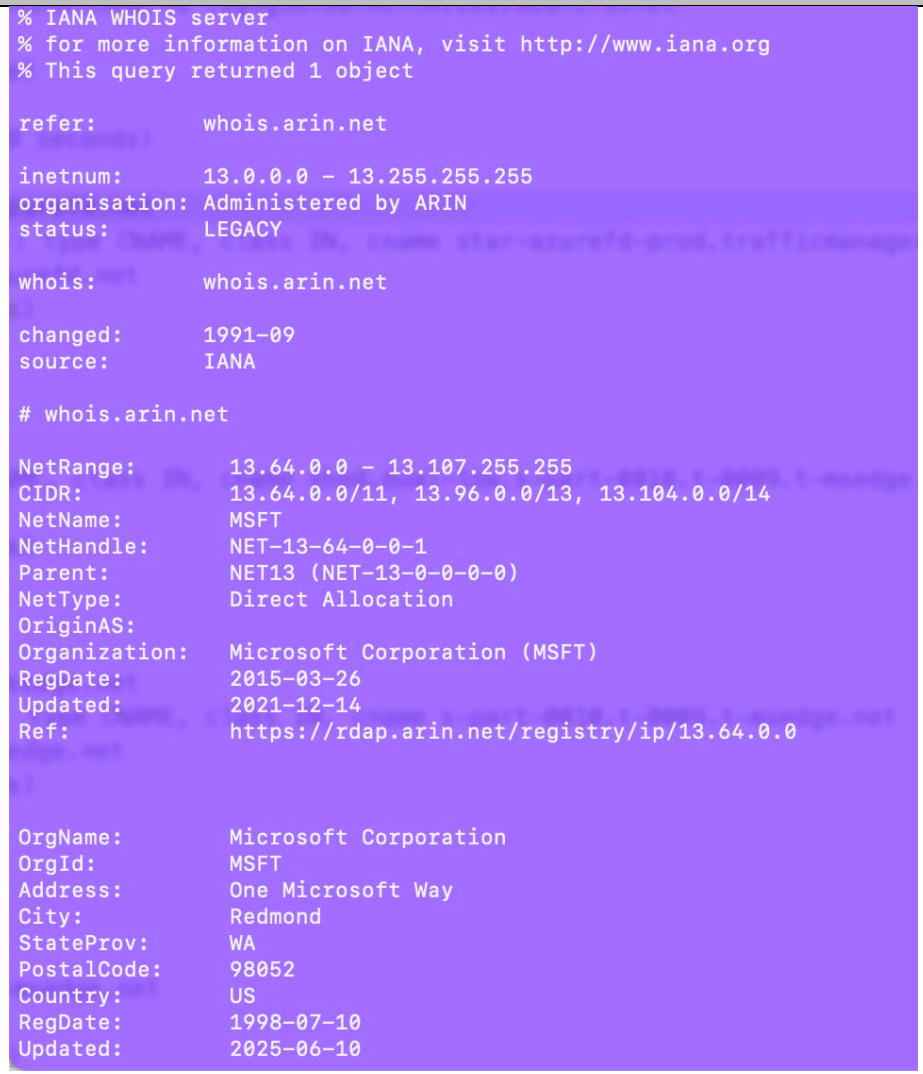
```
; <<>> DiG 9.10.6 <<>> avadotcomprod-fug4gsdtd0fmcrhx.z01.azurefd.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32757
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

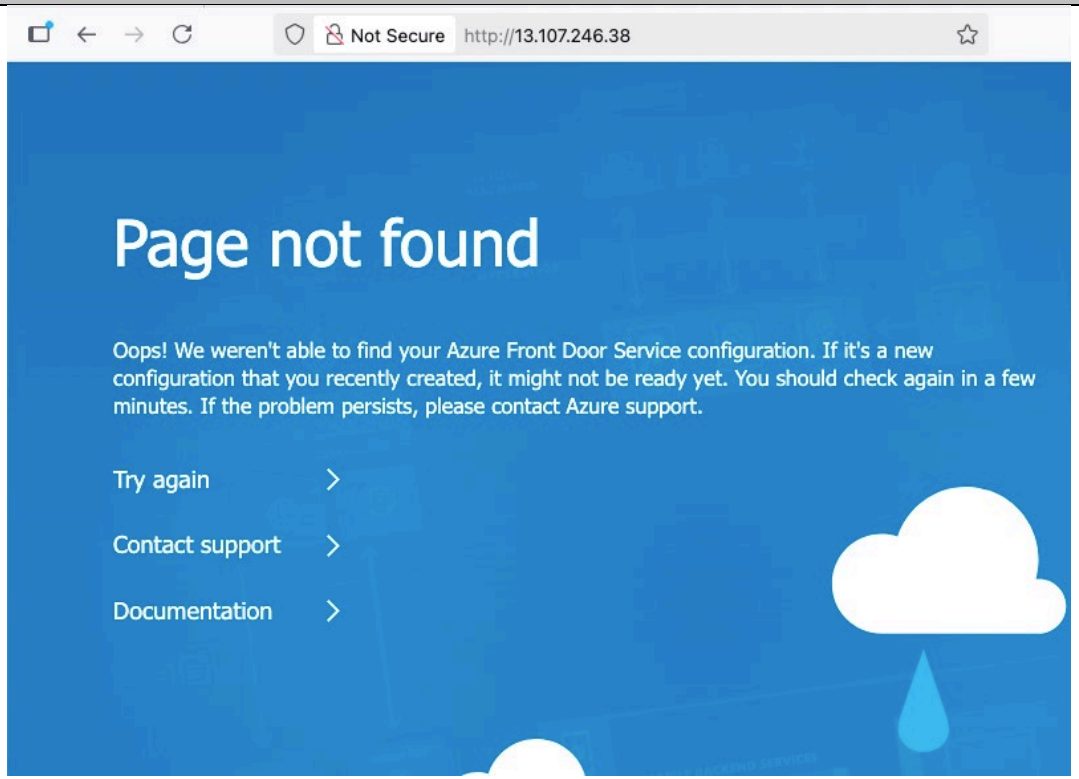
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;avadotcomprod-fug4gsdtd0fmcrhx.z01.azurefd.net.        IN A

;; ANSWER SECTION:
avadotcomprod-fug4gsdtd0fmcrhx.z01.azurefd.net. 30 IN CNAME star-azurefd-prod.trafficmanager.net.
star-azurefd-prod.trafficmanager.net. 60 IN CNAME shed.dual-low.s-part-0010.t-0009.t-msedge.net.
shed.dual-low.s-part-0010.t-0009.t-msedge.net. 22 IN CNAME s-part-0010.t-0009.t-msedge.net.
s-part-0010.t-0009.t-msedge.net. 22 IN  A       13.107.246.38

;; Query time: 79 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Mon Sep 15 10:51:23 CDT 2025
;; MSG SIZE  rcvd: 208
```

avadotcomprod-fug4gsdtd0fmcrhx.z01.azurefd.net directs traffic to 13.107.246.38

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	 13.107.246.38 belongs to Microsoft Corporation

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	 13.107.246.38 landing page indicates Azure Front Door server
[28.b] associating the at least one repeater server with a second alias name, wherein requests for a second resource located on a second origin server are directed, based at least in part on said second alias name, to the at least one repeater server for delivery of the second resource from said	Microsoft associates the at least one repeater server with a second alias name, wherein requests for a second resource located on a second origin server are directed, based at least in part on said second alias name, to the at least one repeater server for delivery of the second resource from said at least one repeater server, wherein the second origin server is distinct from the first origin server. For example, Azure CDN and/or Front Door provide services for many different third parties, each of which have different origin servers. Additionally, on information and belief, third-party content hosted on Azure CDN and/or Front Door and Microsoft-owned resources are hosted on different origin servers in Microsoft's network. As discussed above with respect to claim limitation [28.a], Microsoft provides instructions on creating alias records associated with Azure CDN endpoints for users of both Azure DNS and third-party DNS providers.

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
at least one repeater server, wherein the second origin server is distinct from the first origin server;	# Tutorial: Add a custom domain to your endpoint Article • 03/21/2024 • 22 contributors Feedback **In this article** Prerequisites Create a CNAME DNS record Add a custom domain to your content delivery network endpoint Verify the custom domain Show 2 more This tutorial shows how to add a custom domain to an Azure Content Delivery Network endpoint. The endpoint name in your content delivery network profile is a subdomain of azureedge.net. By default when delivering content, the content delivery network profile domain gets included in the URL. For example, `https://contoso.azureedge.net/photo.png`. Azure Content Delivery Network provides the option of associating a custom domain with a content delivery network endpoint. This option delivers content with a custom domain in your URL instead of the default domain. https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Azure DNS | DNS Provider

Azure DNS uses alias records for Azure resources within the same subscription.

To add an alias record for your Azure Content Delivery Network endpoint:

1. Sign in to the Azure portal.
2. In the left-hand menu, select **All resources** then the Azure DNS zone for your custom domain.
3. In the DNS zone for your custom domain, select + **Record set**.
4. In **Add record set** enter or select the following information:

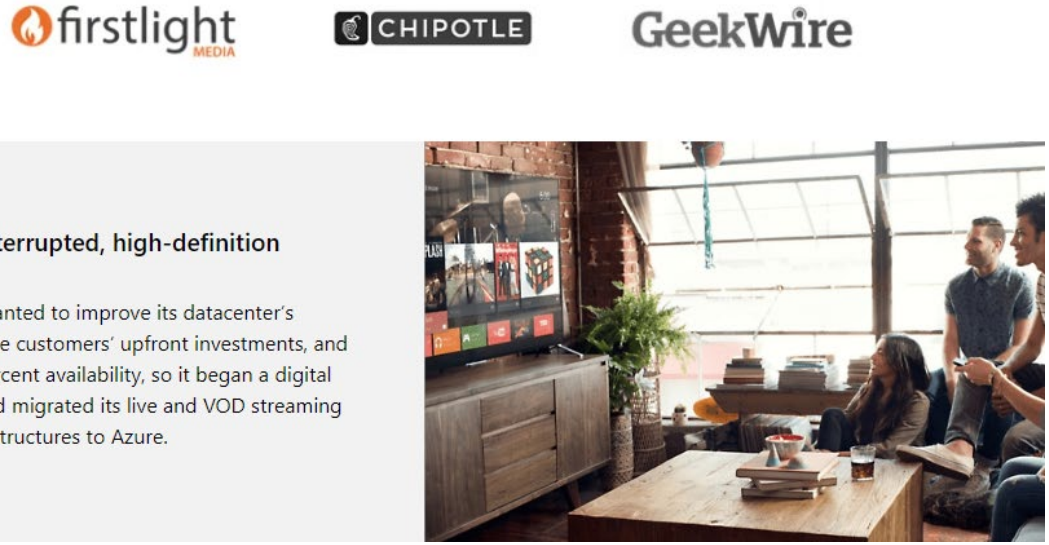
Expand table

Setting	Value
Name	Enter alias you want to use for your content delivery network endpoint. For example, **www**.
Type	Select **CNAME**.
Alias record set	Select **Yes**.
Alias type	Select **Azure resource**.
Choose a subscription	Select your subscription.
Azure resource	Select your Azure Content Delivery Network endpoint.

5. Change the **TTL** for the record to your value.
6. Select **OK**.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

See also Claim 28.a.

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Further, requests for second resources located on a second origin server are directed based at least in part on the second alias name associated with the shared repeater server. For instance, Microsoft Azure CDN and/or Azure Front Door support custom domain alias creation for multiple CDN hosting clients accessing content from various origin servers cached by CDN endpoints.  https://azure.microsoft.com/en-us/products/cdn#customer-stories

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

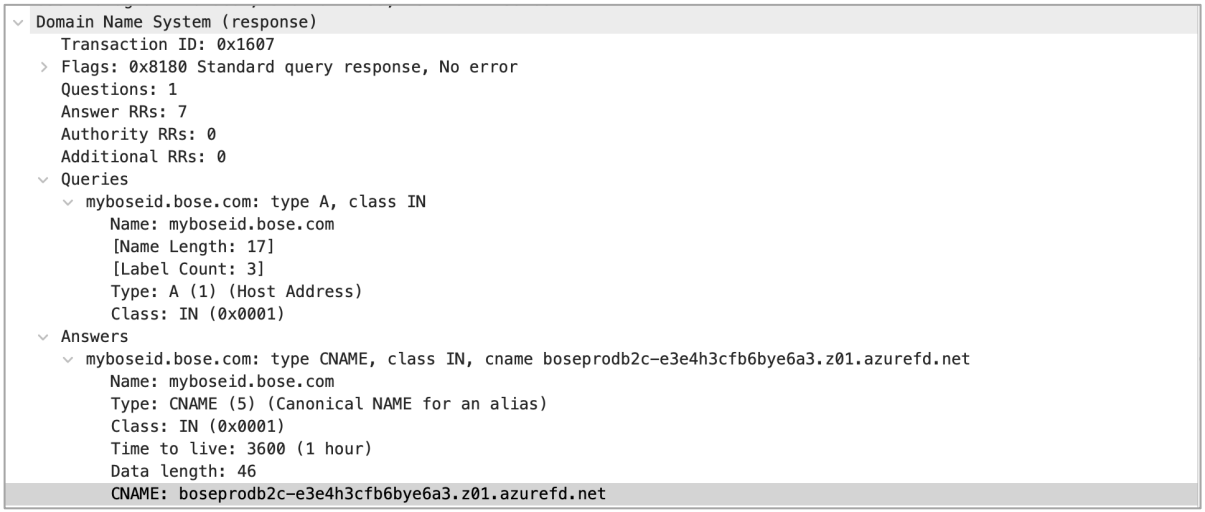
Verify the custom domain

After you've completed the registration of your custom domain, verify that the custom domain references your content delivery network endpoint.

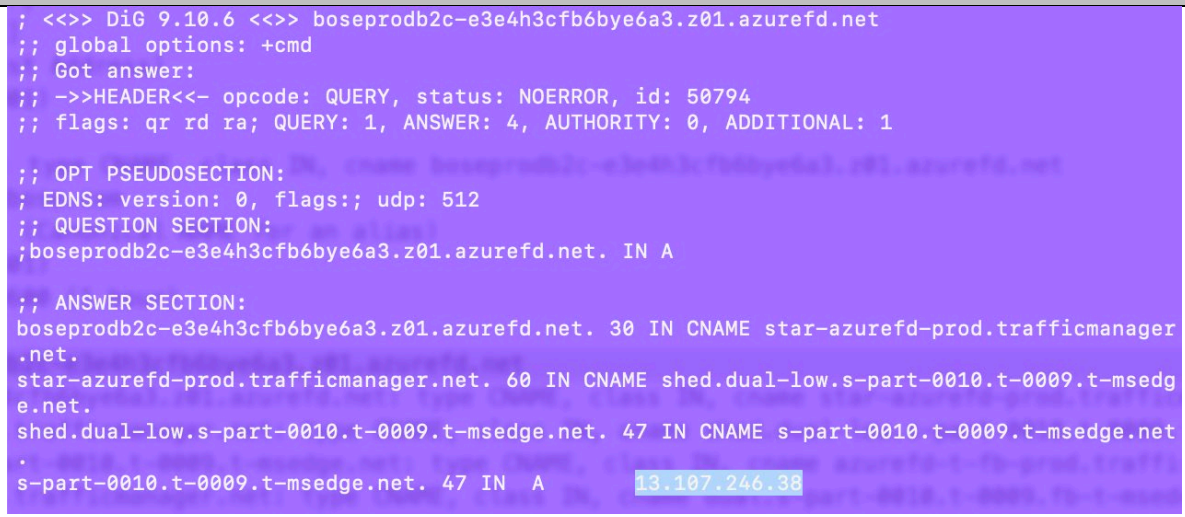
1. Ensure that you have public content that you want cached at the endpoint. For example, if your content delivery network endpoint is associated with a storage account, Azure Content Delivery Network caches the content in a public container. Set your container to allow public access and it contains at least one file to test the custom domain.

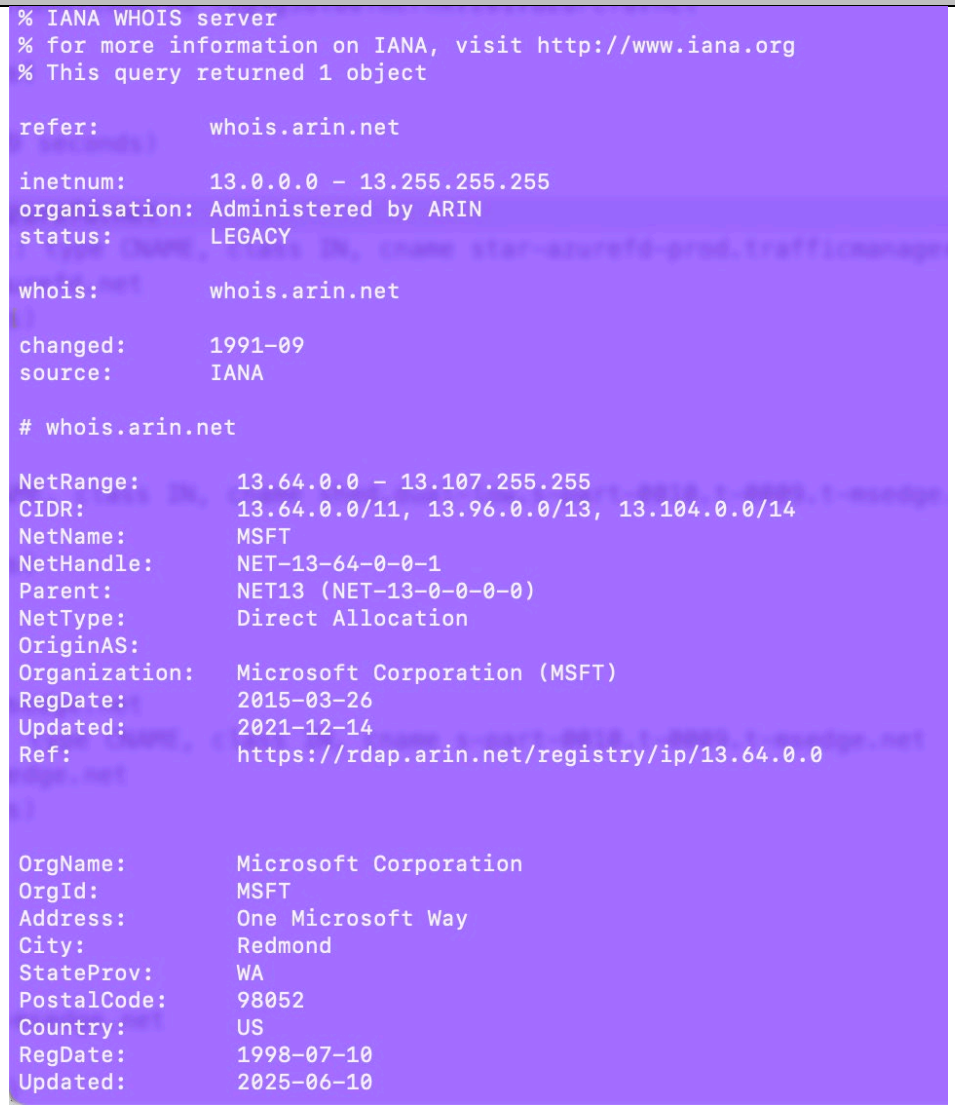
2. In your browser, navigate to the address of the file by using the custom domain. For example, if your custom domain is `www.contoso.com`, the URL to the cached file should be similar to the following URL: `http://www.contoso.com/my-public-container/my-file.jpg`. Verify that the result is that same as when you access the content delivery network endpoint directly at <**endpoint-hostname**>.azureedge.net.

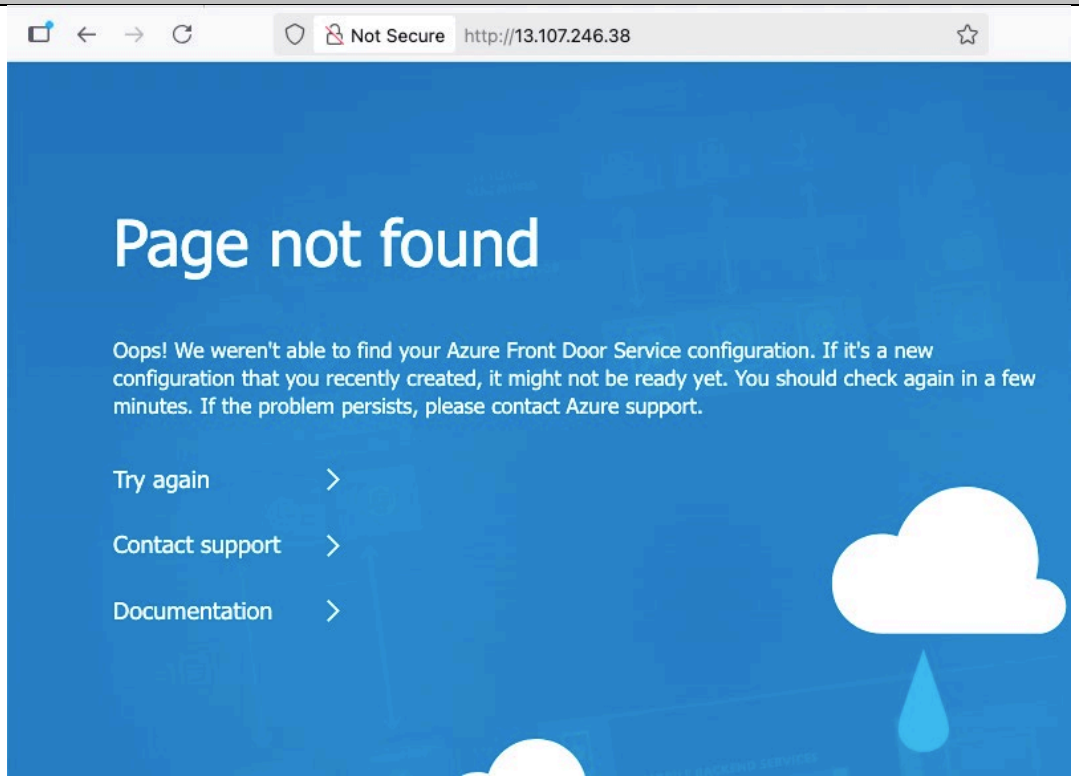
https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

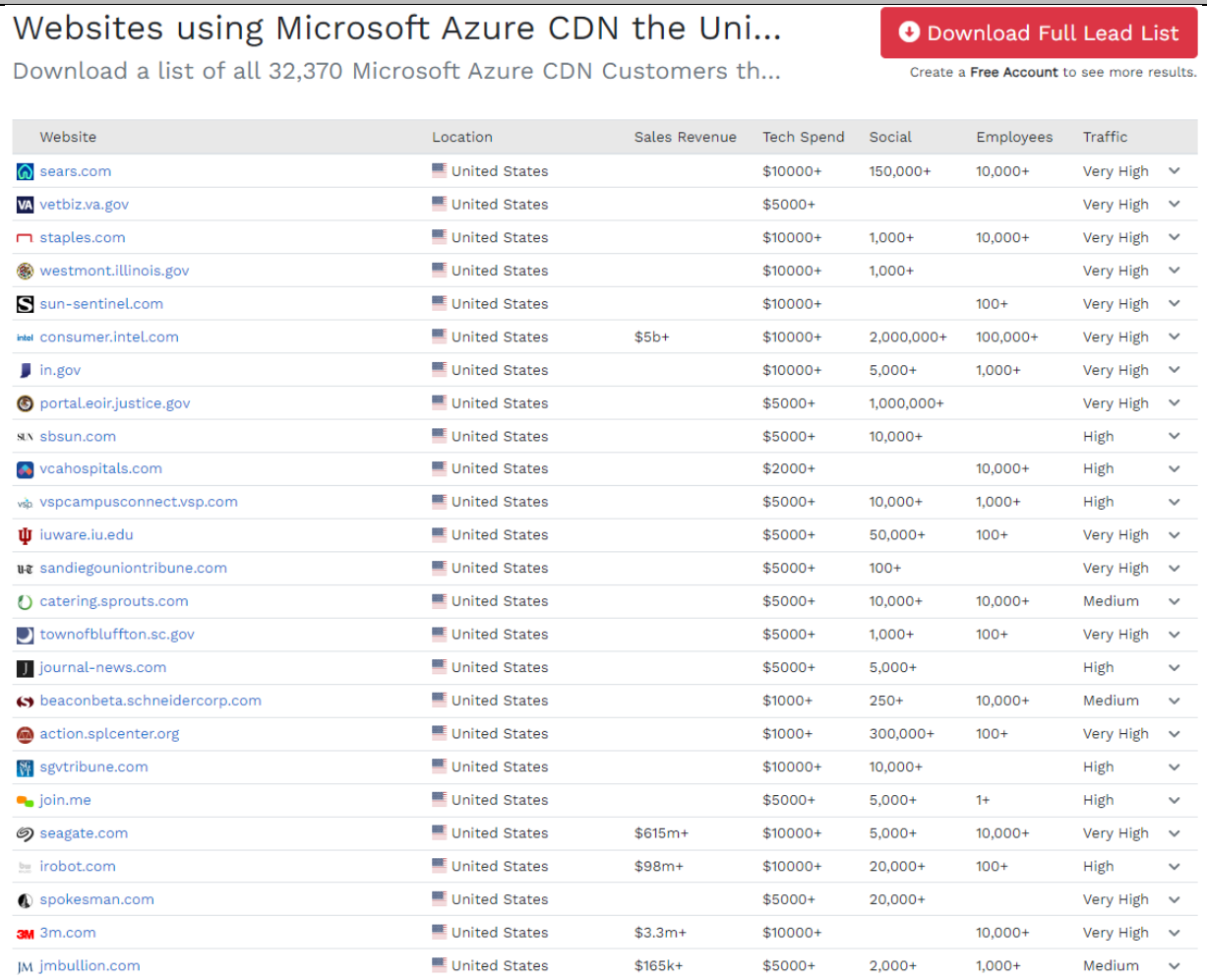


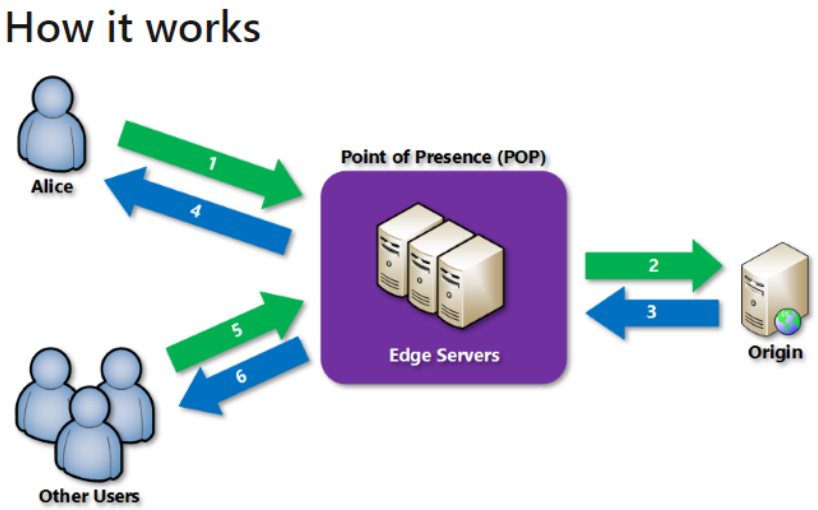
DNS query response for myboseid.bose.com returns boseprodb2c-e3e4h3cfb6bye6a3.z01.azurefd.net

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	 boseprodb2c-e3e4h3cfb6bye6a3.z01.azurefd.net directs traffic to 13.107.246.38

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	 13.107.246.38 belongs to Microsoft

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	 13.107.246.38 landing page indicates Azure Front Door server

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	 https://trends.builtwith.com/websitelist/Microsoft-Azure-CDN/United-States
[28.c] providing a table listing origin servers having content located thereon, wherein said content is authorized for	Microsoft provides a table listing origin servers having content located thereon, wherein said content is authorized for delivery to client machines via the at least one shared repeater server, and wherein the origin servers comprise the first origin server and the second origin server. For instance, Microsoft Azure CDN and/or Azure Front Door utilize a table to correlate requests to origin servers hosting the requested content.

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
delivery to client machines via the at least one shared repeater server, and wherein the origin servers comprise the first origin server and the second origin server; and	 1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user. 2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server. 3. The origin server returns the file to an edge server in the POP. 4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days. 5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP. 6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

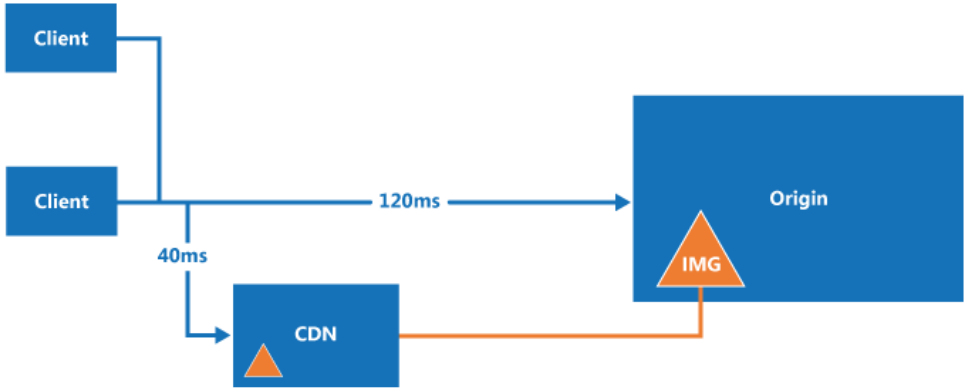
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

How and why a CDN is used

Typical uses for a CDN include:

- Delivering static resources for client applications, often from a website. These resources can be images, style sheets, documents, files, client-side scripts, HTML pages, HTML fragments, or any other content that the server does not need to modify for each request. The application can create items at runtime and make them available to the CDN (for example, by creating a list of current news headlines), but it does not do so for each request.
- Delivering public static and shared content to devices such as mobile phones and tablet computers. The application itself is a web service that offers an API to clients running on the various devices. The CDN can also deliver static datasets (via the web service) for the clients to use, perhaps to generate the client UI. For example, the CDN could be used to distribute JSON or XML documents.
- Serving entire websites that consist of only public static content to clients, without requiring any dedicated compute resources.
- Streaming video files to the client on demand. Video benefits from the low latency and reliable connectivity available from the globally located datacenters that offer CDN connections. Microsoft Azure Media Services integrates with Azure Content Delivery Network to deliver content directly to the CDN for further distribution. For more information, see Streaming endpoints overview.
- Generally improving the experience for users, especially those located far from the datacenter hosting the application. These users might otherwise suffer higher latency. A large proportion of the total size of the content in a web application is often static, and using the CDN can help to maintain performance and overall user experience while eliminating the requirement to deploy the application to multiple datacenters. For a list of Azure Content Delivery Network node locations, see Azure CDN POP Locations.
- Supporting IoT (Internet of Things) solutions. The huge numbers of devices and appliances involved in an IoT solution could easily overwhelm an application if it had to distribute firmware updates directly to each device.
- Coping with peaks and surges in demand without requiring the application to scale, avoiding the consequent increase in running costs. For example, when an update to an operating system is released for a hardware device such as a specific model of router, or for a consumer device such as a smart TV, there will be a huge peak in demand as it is downloaded by millions of users and devices over a short period.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Cache control Consider how to manage caching within the system. For example, in Azure Content Delivery Network, you can set global caching rules, and then set custom caching for particular origin endpoints. You can also control how caching is performed in a CDN by sending cache-directive headers at the origin. For more information, see How caching works. To prevent objects from being available on the CDN, you can delete them from the origin, remove or delete the CDN endpoint, or in the case of blob storage, make the container or blob private. However, items are not removed from the CDN until the time to live expires. You can also manually purge a CDN endpoint. https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Websites using Microsoft Azure CDN the Uni...

Download a list of all 32,370 Microsoft Azure CDN Customers th...

Download Full Lead List

Create a **Free Account** to see more results.

Website	Location	Sales Revenue	Tech Spend	Social	Employees	Traffic
sears.com	United States		$10000+	150,000+	10,000+	Very High
vetbiz.va.gov	United States		$5000+			Very High
staples.com	United States		$10000+	1,000+	10,000+	Very High
westmont.illinois.gov	United States		$10000+	1,000+		Very High
sun-sentinel.com	United States		$10000+		100+	Very High
consumer.intel.com	United States	$5b+	$10000+	2,000,000+	100,000+	Very High
in.gov	United States		$10000+	5,000+	1,000+	Very High
portal.eoir.justice.gov	United States		$5000+	1,000,000+		Very High
sbsun.com	United States		$5000+	10,000+		High
vcahospitals.com	United States		$2000+		10,000+	High
vspcampusconnect.vsp.com	United States		$5000+	10,000+	1,000+	High
iuware.iu.edu	United States		$5000+	50,000+	100+	Very High
sandiegouniontribune.com	United States		$5000+	100+		Very High
catering.sprouts.com	United States		$5000+	10,000+	10,000+	Medium
townofbluffton.sc.gov	United States		$5000+	1,000+	100+	Very High
journal-news.com	United States		$5000+	5,000+		High
beaconbeta.schneidercorp.com	United States		$1000+	250+	10,000+	Medium
action.splcenter.org	United States		$1000+	300,000+	100+	Very High
sgvtribune.com	United States		$10000+	10,000+		High
join.me	United States		$5000+	5,000+	1+	High
seagate.com	United States	$615m+	$10000+	5,000+	10,000+	Very High
irobot.com	United States	$98m+	$10000+	20,000+	100+	High
spokesman.com	United States		$5000+	20,000+		Very High
3m.com	United States	$3.3m+	$10000+		10,000+	Very High
jmbullion.com	United States	$165k+	$5000+	2,000+	1,000+	Medium

https://trends.builtwith.com/websitelist/Microsoft-Azure-CDN/United-States

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Azure DNS | DNS Provider

Azure DNS uses alias records for Azure resources within the same subscription.

To add an alias record for your Azure Content Delivery Network endpoint:

1. Sign in to the Azure portal.
2. In the left-hand menu, select **All resources** then the Azure DNS zone for your custom domain.
3. In the DNS zone for your custom domain, select + **Record set**.
4. In **Add record set** enter or select the following information:

Expand table

Setting	Value
Name	Enter alias you want to use for your content delivery network endpoint. For example, **www**.
Type	Select **CNAME**.
Alias record set	Select **Yes**.
Alias type	Select **Azure resource**.
Choose a subscription	Select your subscription.
Azure resource	Select your Azure Content Delivery Network endpoint.

5. Change the **TTL** for the record to your value.
6. Select **OK**.

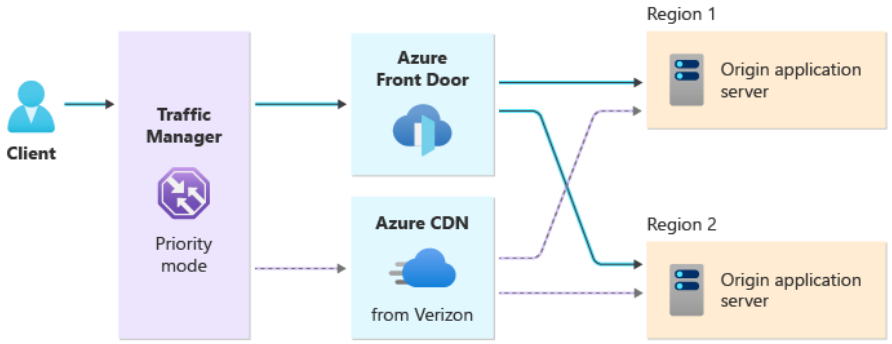
https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Microsoft Azure additionally integrates Edgio (formerly Verizon) CDN features via Azure Front Door. Caching is a common way to reduce load on the backend services and optimize performance for users. Content delivery networks (CDNs), including Azure Front Door, provide caching at the network edge. Mission-critical workloads often use multiple CDNs to achieve a higher level of uptime. If one CDN experiences outage or degraded performance, your traffic is automatically diverted to another CDN. If you implement multiple CDNs, consider the implications of this approach. Each CDN provides a separate network path to your application servers, and you need to configure and test each CDN separately. This article describes an approach for using Azure Front Door with a partner CDN, Verizon. This approach is suitable for solutions that rely heavily on caching for delivering static content delivery, media, and high-scale eCommerce applications. https://learn.microsoft.com/en-us/azure/architecture/guide/networking/global-web-applications/mission-critical-content-delivery

Approach

Verizon's CDN and the CDN platform (Edgio) can be integrated into your Azure solution. You can configure it from the Azure portal and APIs. The platform is isolated from Microsoft's infrastructure.

This isolation provides a high degree of resiliency from disaster scenarios. If an outage or disaster occurs, traffic is automatically shifted between Azure Front Door and Verizon's CDN. You can use Azure Traffic Manager to detect an outage and redirect traffic to the alternative CDN.

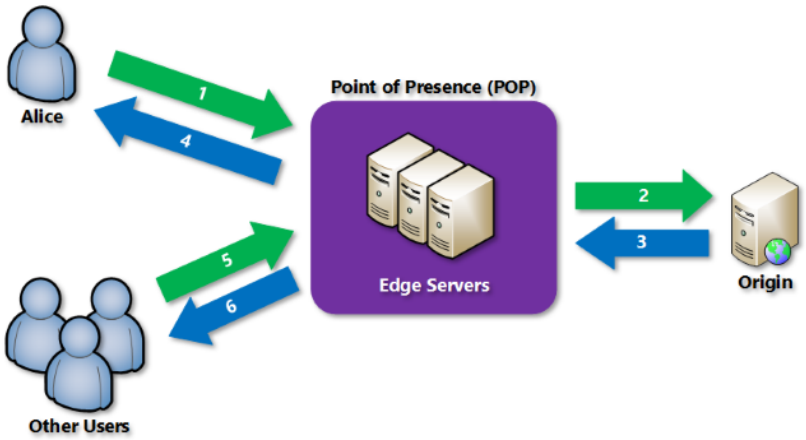


- **Traffic Manager using priority routing mode** has two endpoints. By default, Traffic Manager sends requests through Azure Front Door. If Azure Front Door is unavailable, Traffic Manager sends the request through the partner CDN instead.
- **Azure Front Door** processes and routes most of your application traffic. Azure Front Door routes traffic to the appropriate origin application server, and it provides the primary path to your application. If Azure Front Door is unavailable, traffic is automatically redirected through the secondary path.
- **Azure CDN from Verizon** is configured to send traffic to each origin server.
- **Your origin application servers** need to be ready to accept traffic from both Azure Front Door and Azure CDN from Verizon, at any time.

https://learn.microsoft.com/en-us/azure/architecture/guide/networking/global-web-applications/mission-critical-content-delivery

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	To the extent this limitation is not literally present, it is nonetheless present under the doctrine of equivalents because it performs substantially the same function, in substantially the same way, to achieve substantially the same result. Specifically, other data structures can perform the function of a table listing origin servers having content located thereon. Data structures in Microsoft servers, to the extent are not explicitly defined as a table, provide information sufficient for a repeater server to identify individual origins. This information is identified in substantially the same way—namely, as information stored in a data structure in Microsoft's network. The information contained in the data structures also achieve substantially the same result, by providing the requisite information for the repeater to analyze and determine an origin server associated with the particular resource.
[28.d] wherein the at least one repeater server is further constructed and adapted to analyze, using the table, an alias name received with a client request for a particular resource to determine an origin server associated with the particular resource.	Microsoft repeater servers are constructed and adapted to analyze, using the table, an alias name received with a client request for a particular resource to determine an origin server associated with the particular resource. For instance, Microsoft Azure CDN and/or Azure Front Door include servers in the edge infrastructure that cache content for multiple Azure CDN customers and/or Microsoft resources and which use a table to determine an origin server for a CDN customer.

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

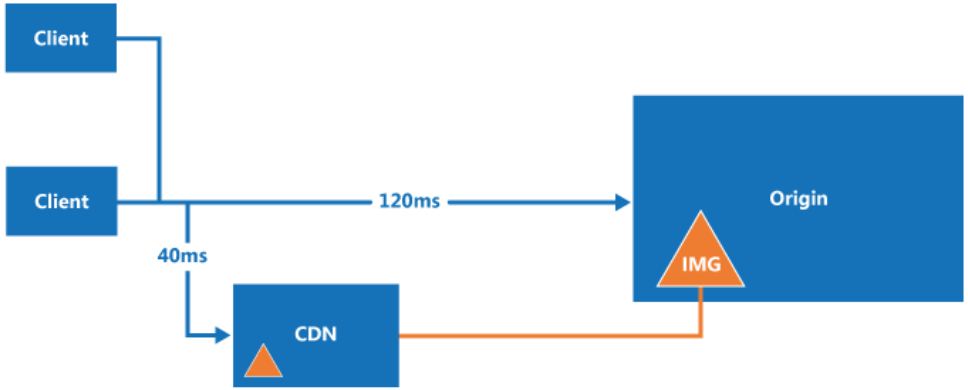
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

How and why a CDN is used

Typical uses for a CDN include:

- Delivering static resources for client applications, often from a website. These resources can be images, style sheets, documents, files, client-side scripts, HTML pages, HTML fragments, or any other content that the server does not need to modify for each request. The application can create items at runtime and make them available to the CDN (for example, by creating a list of current news headlines), but it does not do so for each request.
- Delivering public static and shared content to devices such as mobile phones and tablet computers. The application itself is a web service that offers an API to clients running on the various devices. The CDN can also deliver static datasets (via the web service) for the clients to use, perhaps to generate the client UI. For example, the CDN could be used to distribute JSON or XML documents.
- Serving entire websites that consist of only public static content to clients, without requiring any dedicated compute resources.
- Streaming video files to the client on demand. Video benefits from the low latency and reliable connectivity available from the globally located datacenters that offer CDN connections. Microsoft Azure Media Services integrates with Azure Content Delivery Network to deliver content directly to the CDN for further distribution. For more information, see Streaming endpoints overview.
- Generally improving the experience for users, especially those located far from the datacenter hosting the application. These users might otherwise suffer higher latency. A large proportion of the total size of the content in a web application is often static, and using the CDN can help to maintain performance and overall user experience while eliminating the requirement to deploy the application to multiple datacenters. For a list of Azure Content Delivery Network node locations, see Azure CDN POP Locations.
- Supporting IoT (Internet of Things) solutions. The huge numbers of devices and appliances involved in an IoT solution could easily overwhelm an application if it had to distribute firmware updates directly to each device.
- Coping with peaks and surges in demand without requiring the application to scale, avoiding the consequent increase in running costs. For example, when an update to an operating system is released for a hardware device such as a specific model of router, or for a consumer device such as a smart TV, there will be a huge peak in demand as it is downloaded by millions of users and devices over a short period.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	## Cache control Consider how to manage caching within the system. For example, in Azure Content Delivery Network, you can set global caching rules, and then set custom caching for particular origin endpoints. You can also control how caching is performed in a CDN by sending cache-directive headers at the origin. For more information, see How caching works. To prevent objects from being available on the CDN, you can delete them from the origin, remove or delete the CDN endpoint, or in the case of blob storage, make the container or blob private. However, items are not removed from the CDN until the time to live expires. You can also manually purge a CDN endpoint. https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Websites using Microsoft Azure CDN the Uni...

Download a list of all 32,370 Microsoft Azure CDN Customers th...

Download Full Lead List

Create a **Free Account** to see more results.

Website	Location	Sales Revenue	Tech Spend	Social	Employees	Traffic
sears.com	United States		$10000+	150,000+	10,000+	Very High
vetbiz.va.gov	United States		$5000+			Very High
staples.com	United States		$10000+	1,000+	10,000+	Very High
westmont.illinois.gov	United States		$10000+	1,000+		Very High
sun-sentinel.com	United States		$10000+		100+	Very High
consumer.intel.com	United States	$5b+	$10000+	2,000,000+	100,000+	Very High
in.gov	United States		$10000+	5,000+	1,000+	Very High
portal.eoir.justice.gov	United States		$5000+	1,000,000+		Very High
sbsun.com	United States		$5000+	10,000+		High
vcahospitals.com	United States		$2000+		10,000+	High
vspcampusconnect.vsp.com	United States		$5000+	10,000+	1,000+	High
iuware.iu.edu	United States		$5000+	50,000+	100+	Very High
sandiegouniontribune.com	United States		$5000+	100+		Very High
catering.sprouts.com	United States		$5000+	10,000+	10,000+	Medium
townofbluffton.sc.gov	United States		$5000+	1,000+	100+	Very High
journal-news.com	United States		$5000+	5,000+		High
beaconbeta.schneidercorp.com	United States		$1000+	250+	10,000+	Medium
action.splcenter.org	United States		$1000+	300,000+	100+	Very High
sgvtribune.com	United States		$10000+	10,000+		High
join.me	United States		$5000+	5,000+	1+	High
seagate.com	United States	$615m+	$10000+	5,000+	10,000+	Very High
irobot.com	United States	$98m+	$10000+	20,000+	100+	High
spokesman.com	United States		$5000+	20,000+		Very High
3m.com	United States	$3.3m+	$10000+		10,000+	Very High
jmbullion.com	United States	$165k+	$5000+	2,000+	1,000+	Medium

https://trends.builtwith.com/websitelist/Microsoft-Azure-CDN/United-States

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Azure DNS | DNS Provider

Azure DNS uses alias records for Azure resources within the same subscription.

To add an alias record for your Azure Content Delivery Network endpoint:

1. Sign in to the Azure portal.
2. In the left-hand menu, select **All resources** then the Azure DNS zone for your custom domain.
3. In the DNS zone for your custom domain, select + **Record set**.
4. In **Add record set** enter or select the following information:

Expand table

Setting	Value
Name	Enter alias you want to use for your content delivery network endpoint. For example, **www**.
Type	Select **CNAME**.
Alias record set	Select **Yes**.
Alias type	Select **Azure resource**.
Choose a subscription	Select your subscription.
Azure resource	Select your Azure Content Delivery Network endpoint.

5. Change the **TTL** for the record to your value.
6. Select **OK**.

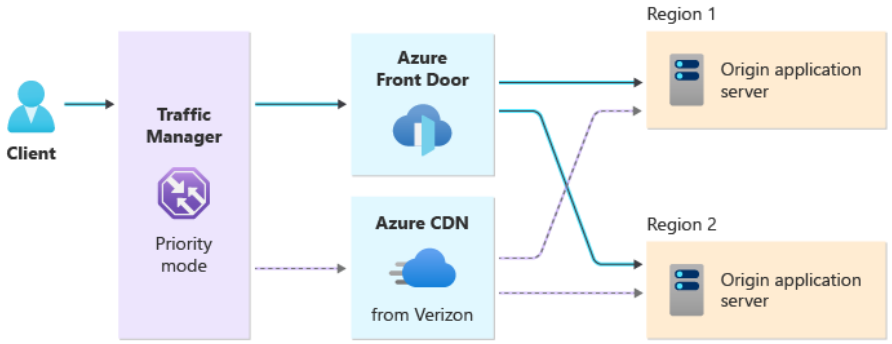
https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Microsoft Azure additionally integrates Edgio (formerly Verizon) CDN features via Azure Front Door. Caching is a common way to reduce load on the backend services and optimize performance for users. Content delivery networks (CDNs), including Azure Front Door, provide caching at the network edge. Mission-critical workloads often use multiple CDNs to achieve a higher level of uptime. If one CDN experiences outage or degraded performance, your traffic is automatically diverted to another CDN. If you implement multiple CDNs, consider the implications of this approach. Each CDN provides a separate network path to your application servers, and you need to configure and test each CDN separately. This article describes an approach for using Azure Front Door with a partner CDN, Verizon. This approach is suitable for solutions that rely heavily on caching for delivering static content delivery, media, and high-scale eCommerce applications. https://learn.microsoft.com/en-us/azure/architecture/guide/networking/global-web-applications/mission-critical-content-delivery

Approach

Verizon's CDN and the CDN platform (Edgio) can be integrated into your Azure solution. You can configure it from the Azure portal and APIs. The platform is isolated from Microsoft's infrastructure.

This isolation provides a high degree of resiliency from disaster scenarios. If an outage or disaster occurs, traffic is automatically shifted between Azure Front Door and Verizon's CDN. You can use Azure Traffic Manager to detect an outage and redirect traffic to the alternative CDN.

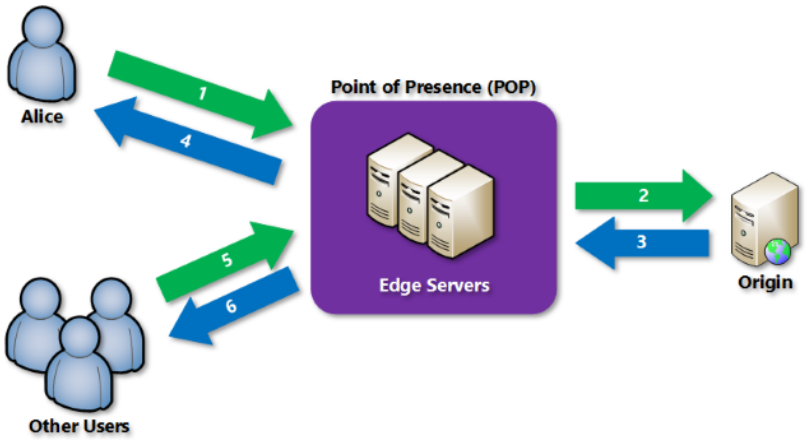


- **Traffic Manager using priority routing mode** has two endpoints. By default, Traffic Manager sends requests through Azure Front Door. If Azure Front Door is unavailable, Traffic Manager sends the request through the partner CDN instead.
- **Azure Front Door** processes and routes most of your application traffic. Azure Front Door routes traffic to the appropriate origin application server, and it provides the primary path to your application. If Azure Front Door is unavailable, traffic is automatically redirected through the secondary path.
- **Azure CDN from Verizon** is configured to send traffic to each origin server.
- **Your origin application servers** need to be ready to accept traffic from both Azure Front Door and Azure CDN from Verizon, at any time.

https://learn.microsoft.com/en-us/azure/architecture/guide/networking/global-web-applications/mission-critical-content-delivery

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
29. The method of claim 28, further comprising: [29.a] in response to a request for the first resource issued by a particular client machine, identifying the at least one shared repeater server from a set of shared repeater servers based at least in part on a network cost of delivering resources to the particular client machine; and	Microsoft performs the method of claim 28 as set forth above. Further, in response to a request for the first resource issued by a particular client machine, Microsoft identifies the at least one shared repeater server from a set of shared repeater servers based at least in part on a network cost of delivering resources to the particular client machine. For instance, Microsoft Azure CDN and/or Azure Front Door utilize traffic metering which factors in the cost of delivering network resources (e.g., requested content) to a particular client machine.

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

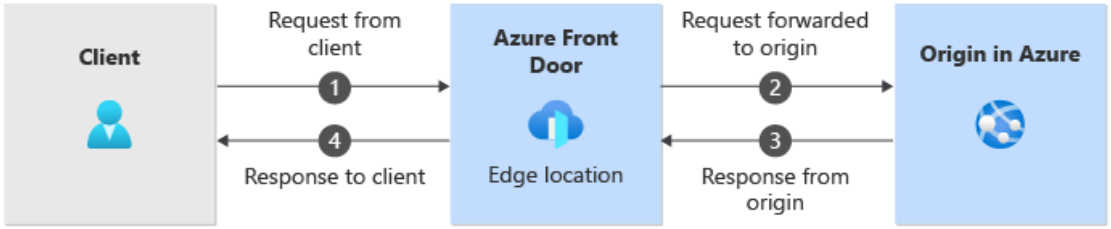
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Front Door's billing model includes several components. Front Door charges a base fee for each profile that you deploy. You're also charged for requests and data transfer based on your usage. *Billing meters* collect information about your Front Door usage. Your monthly Azure bill aggregates the billing information across the month and applies the pricing to determine the amount you need to pay.

https://learn.microsoft.com/en-us/azure/frontdoor/billing

Request processing and traffic fees

Each request that goes through Front Door incur request processing and traffic fees:



Each part of the request process is billed separately:

1. Number of requests from client to Front Door
2. Data transfer from Front Door edge to origin
3. Data transfer from origin to Front Door (nonbillable)
4. Data transfer from Front Door to client

https://learn.microsoft.com/en-us/azure/frontdoor/billing

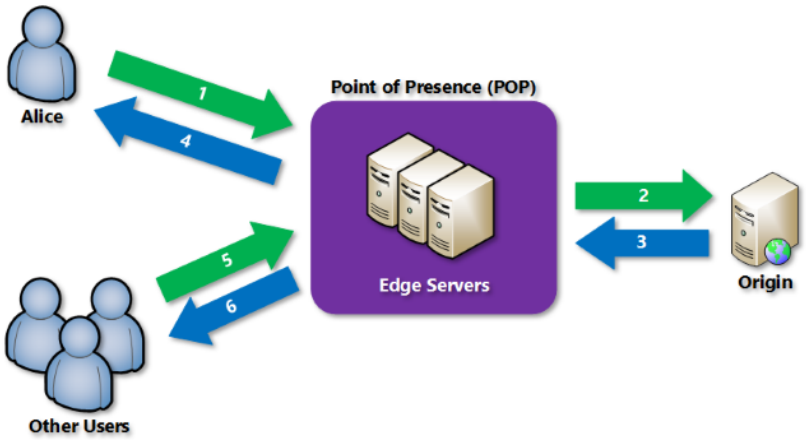
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Data transfer from Front Door to client Front Door charges for the bytes that are sent from the Front Door edge location back to the client. The price is different depending on the geographical region of the Front Door edge location that serves the request. If a response is compressed, Front Door only charges for the compressed data. https://learn.microsoft.com/en-us/azure/frontdoor/billing **Cross-region traffic** Some of the Front Door billing meters have different rates depending on the location of the Front Door edge location that processes a request. Usually, the Front Door edge location that processes a request is the one that's closest to the client, which helps to reduce latency and maximize performance. Front Door charges for traffic from the edge location to the origin. Traffic is charged at different rates depending on the location of the Front Door edge location. If your origin is in a different Azure region, you aren't billed extra for inter-region traffic. https://learn.microsoft.com/en-us/azure/frontdoor/billing
[29.b] associating the first resource with the first alias name.	Microsoft associates the first resource with the first alias name. For instance, Microsoft Azure CDN and/or Front Door associate requested resources with the alias name received with the client request.

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Content delivery network caching Caching is integral to the way a content delivery network operates to speed up delivery and reduce origin load for static assets such as images, fonts, and videos. In content delivery network caching, static resources are selectively stored on strategically placed servers that are more local to a user and offers the following advantages: • Because most web traffic is static (for example, images, fonts, and videos), content delivery network caching reduces network latency by moving content closer to the user, thus reducing the distance that data travels. • By offloading work to a content delivery network, caching can reduce network traffic and the load on the origin server. Doing so reduces cost and resource requirements for the application, even when there are large numbers of users. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Caching with Azure Front Door 08/26/2025 Azure Front Door is a modern content delivery network (CDN), with dynamic site acceleration and load balancing capabilities. When caching is configured on your route, the edge site that receives each request checks its cache for a valid response. Caching helps to reduce the amount of traffic sent to your origin server. If no cached response is available, the request is forwarded to the origin. Each Front Door edge site manages its own cache, and requests might get served by different edge sites. As a result, you might still see some traffic reach your origin, even if you served cached responses. Caching can significantly decrease latency and reduce the load on origin servers. However, not all types of traffic can benefit from caching. Static assets such as images, CSS, and JavaScript files are ideal for caching. While dynamic assets, such as authenticated API endpoints, shouldn't be cached to prevent the leakage of personal information. It's recommended to have separate routes for static and dynamic assets, with caching disabled for the latter. https://docs.azure.cn/en-us/frontdoor/front-door-caching

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	See also Claim elements 28.a and 28.b.
30. The method of claim 28, further comprising: [30.a] in response to a request for the first resource issued by a particular client machine, identifying the at least one repeater server from a set of shared repeater servers based at least in part on load characteristics of the set of shared repeater servers; and	Microsoft performs the method of claim 28 as set forth above. Further, in response to a request for the first resource issued by a particular client machine, Microsoft identifies the at least one repeater server from a set of shared repeater servers based at least in part on load characteristics of the set of shared repeater servers. For instance, Microsoft Azure CDN and/or Azure Front Door use load balancing in order to direct requests to shared repeater servers based on load characteristics. **Load balancing options** 07/31/2025 The term *load balancing* refers to the distribution of processing across multiple computing resources. You load balance to optimize resource usage, maximize throughput, minimize response time, and avoid overloading any single resource. Load balancing can also improve availability by sharing a workload across redundant computing resources. Azure provides various load balancing services that you can use to distribute your workloads across multiple computing resources. These services include Azure API Management, Azure Application Gateway, Azure Front Door, Azure Load Balancer, and Azure Traffic Manager. This article describes considerations to help you determine an appropriate load balancing solution for your workload's needs. https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/load-balancing-overview

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

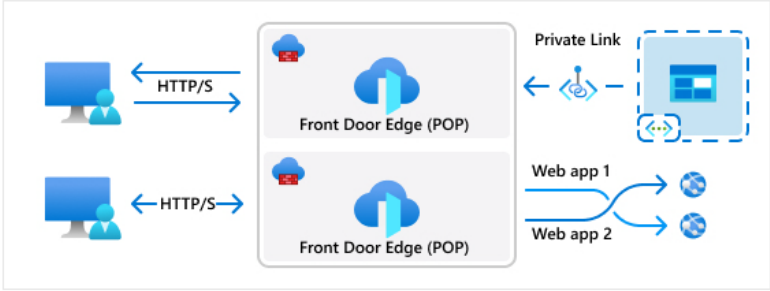
The following table summarizes the Azure load balancing services.

Expand table

Service	Global or regional	Recommended traffic
API Management	Regional or global	HTTP(S) APIs only
Application Gateway	Regional	HTTP(S)
Azure Front Door	Global	HTTP(S)
Load Balancer	Regional or global	Non-HTTP(S)
Traffic Manager	Global	Non-HTTP(S)

https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/load-balancing-overview

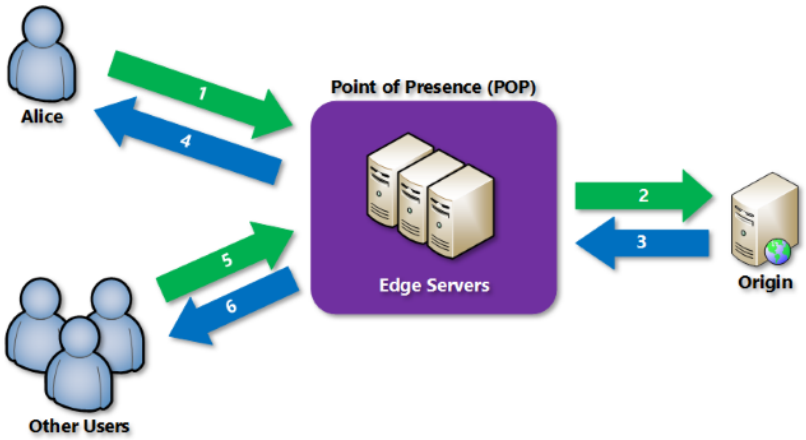
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# What is load balancing and content delivery? 06/26/2025 Load balancing and content delivery are critical components in optimizing the performance and reliability of web applications. Load balancing ensures that incoming network traffic is distributed evenly across multiple servers or services, preventing any single server from becoming overwhelmed with requests. And content delivery optimizes the delivery of content to users by caching and distributing it across multiple locations, reducing latency and improving performance. Together, these two concepts help ensure that applications are highly available, responsive, and capable of handling varying levels of traffic. This article provides an overview of load balancing and content delivery in the context of Azure's services - Azure Application Gateway, Azure Load Balancer, and Azure Front Door. You learn about the key services and categories to help you choose the right solution for your needs. https://learn.microsoft.com/en-us/azure/networking/load-balancer-content-delivery/load-balancing-content-delivery-overview

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Azure Front Door Azure Front Door is an application delivery network that provides global load balancing and site acceleration service for web applications. It offers Layer 7 capabilities for your application like SSL offload, path-based routing, fast failover, and caching to improve performance and high availability of your applications.  ## Use cases - **Global content delivery**: Delivers content and applications globally with low latency by using Microsoft's global edge network. - **Application acceleration**: Improves application performance by using features like split TCP connections and anycast network. - **Security**: Provides platform-level protection against DDoS attacks and integrates with web application firewalls for enhanced security. - **Modern Internet-first architectures**: Supports modern architectures with dynamic, high-quality digital experiences, and automated, secure platforms. https://learn.microsoft.com/en-us/azure/networking/load-balancer-content-delivery/load-balancing-content-delivery-overview
[30.b] associating the first resource with the first alias name.	Microsoft associates the first resource with the first alias name. For instance, Microsoft Azure CDN and/or Azure Front Door associate requested resources with the alias name received with the client request.

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Content delivery network caching Caching is integral to the way a content delivery network operates to speed up delivery and reduce origin load for static assets such as images, fonts, and videos. In content delivery network caching, static resources are selectively stored on strategically placed servers that are more local to a user and offers the following advantages: • Because most web traffic is static (for example, images, fonts, and videos), content delivery network caching reduces network latency by moving content closer to the user, thus reducing the distance that data travels. • By offloading work to a content delivery network, caching can reduce network traffic and the load on the origin server. Doing so reduces cost and resource requirements for the application, even when there are large numbers of users. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Caching with Azure Front Door 08/26/2025 Azure Front Door is a modern content delivery network (CDN), with dynamic site acceleration and load balancing capabilities. When caching is configured on your route, the edge site that receives each request checks its cache for a valid response. Caching helps to reduce the amount of traffic sent to your origin server. If no cached response is available, the request is forwarded to the origin. Each Front Door edge site manages its own cache, and requests might get served by different edge sites. As a result, you might still see some traffic reach your origin, even if you served cached responses. Caching can significantly decrease latency and reduce the load on origin servers. However, not all types of traffic can benefit from caching. Static assets such as images, CSS, and JavaScript files are ideal for caching. While dynamic assets, such as authenticated API endpoints, shouldn't be cached to prevent the leakage of personal information. It's recommended to have separate routes for static and dynamic assets, with caching disabled for the latter. https://docs.azure.cn/en-us/frontdoor/front-door-caching

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

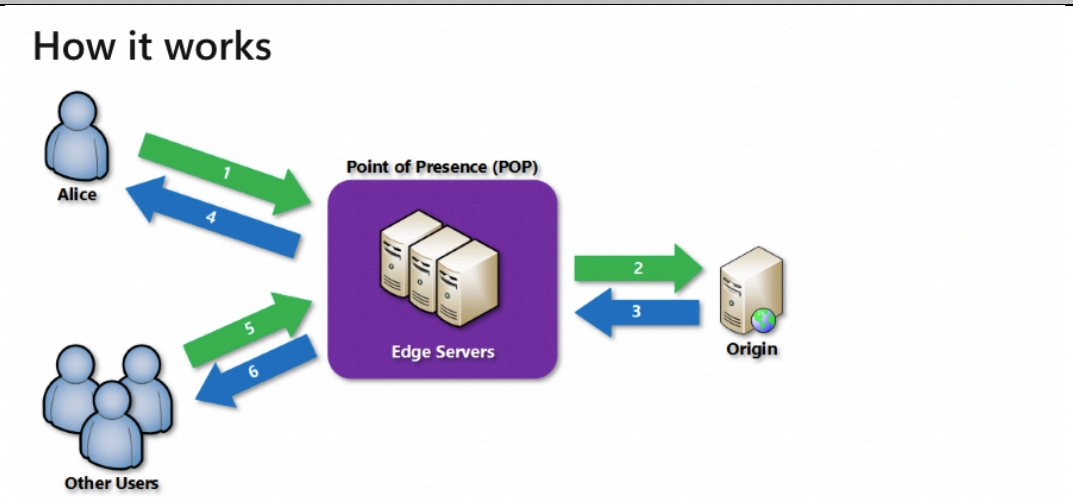
4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	See also Claim elements 28.a and 28.b.
31. The method of claim 28, further comprising: [31.a] in response to a request for the first resource issued by a particular client machine, identifying the at least one repeater server from a set of shared repeater servers based at least in part on a location of the particular client machine relative to the at least one shared repeater server; and	Microsoft performs the method of claim 28 as set forth above. Further, in response to a request for the first resource issued by a particular client machine, Microsoft identifies the at least one repeater server from a set of shared repeater servers based at least in part on a location of the particular client machine relative to the at least one shared repeater server. For instance, Microsoft Azure CDN and/or Front Door identify repeater servers at least partly based on the location of the client machine relative to the shared repeater server (e.g., in order to reduce latency).

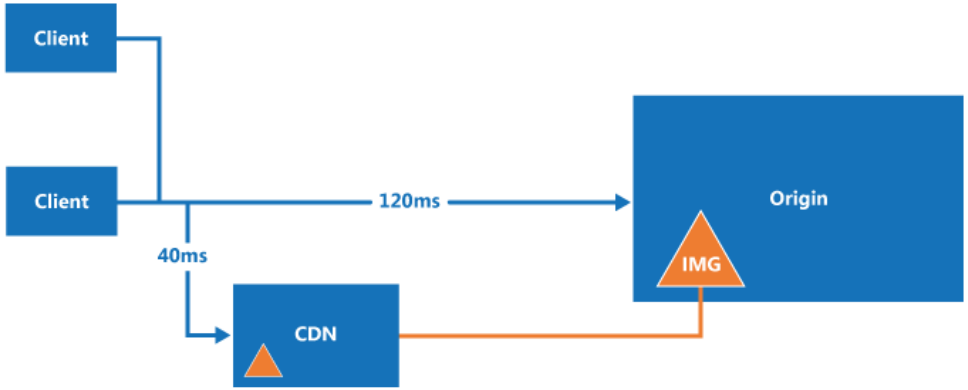
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	 1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as *<endpoint name>*.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user. 2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server. 3. The origin server returns the file to an edge server in the POP. 4. An edge server in the POP caches the file and returns the file to the original requester (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days. 5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP. 6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	This article lists current metros containing point of presence (POP) locations, sorted by region, for Azure Content Delivery Network products. Each metro might contain more than one POP. For example, Azure Content Delivery Network from Microsoft has 192 POPs across 109 metro cities. **ⓘ Important** Each Azure Content Delivery Network product has a distinct way of building its content delivery network infrastructures, hence Microsoft recommends against using POP locations to decide which Azure Content Delivery Network product to use. Instead, you should consider its features and end-user performance. Test the performance with each Azure Content Delivery Network product to choose the right product for your users. https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

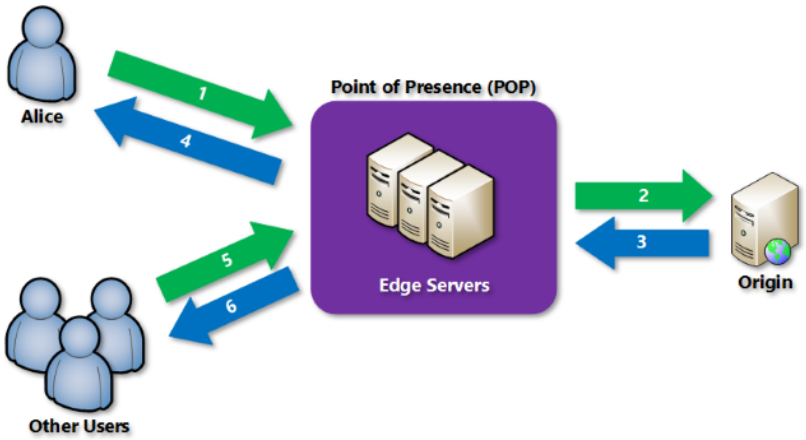
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Region	Cities
North America	Etobicoke, Canada (2) Montreal, Canada Vancouver, Canada (2) Querétaro, Mexico (2) Atlanta, GA, USA (3) Boydton, VA, USA (2) Chaska, MN, USA (2) Cheyenne, WY, USA (2) Chicago, IL, USA (4) Dallas, TX, USA (4) Des Moines,IA, USA (3) Detroit, MI, USA Englewood, CO, USA (2) Honolulu, HI, USA Houston, TX, USA (3) Jacksonville, FL, USA (2) Las Vegas, NV, USA (3) Los Angeles, CA, USA (2) Manassas, VA, USA (3) Memphis, TN, USA Miami, FL, USA (4) Minneapolis, MN, USA (2) Needham Heights, MA, USA (2) Nashville, TN, USA Newark, NJ, USA New York, NY, USA (2) Philadelphia, PA, USA Phoenix, AZ, USA Plano, TX, USA Portland, OR, USA Quincy, WA, USA (3) San Antonio, TX, USA (4) San Jose, CA, USA (4) Salt Lake City, UT, USA (3) Seattle, WA, USA (2) Secaucus, NJ, USA (2) Southfield, MI, USA Tempe, AZ, USA

https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
[31.b] associating the first resource with the first alias name.	Microsoft associates the first resource with the first alias name. For instance, Microsoft Azure CDN and/or Azure Front Door associate requested resources with the alias name received with the client request. **Content delivery network caching** Caching is integral to the way a content delivery network operates to speed up delivery and reduce origin load for static assets such as images, fonts, and videos. In content delivery network caching, static resources are selectively stored on strategically placed servers that are more local to a user and offers the following advantages: • Because most web traffic is static (for example, images, fonts, and videos), content delivery network caching reduces network latency by moving content closer to the user, thus reducing the distance that data travels. • By offloading work to a content delivery network, caching can reduce network traffic and the load on the origin server. Doing so reduces cost and resource requirements for the application, even when there are large numbers of users. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Caching with Azure Front Door 08/26/2025 Azure Front Door is a modern content delivery network (CDN), with dynamic site acceleration and load balancing capabilities. When caching is configured on your route, the edge site that receives each request checks its cache for a valid response. Caching helps to reduce the amount of traffic sent to your origin server. If no cached response is available, the request is forwarded to the origin. Each Front Door edge site manages its own cache, and requests might get served by different edge sites. As a result, you might still see some traffic reach your origin, even if you served cached responses. Caching can significantly decrease latency and reduce the load on origin servers. However, not all types of traffic can benefit from caching. Static assets such as images, CSS, and JavaScript files are ideal for caching. While dynamic assets, such as authenticated API endpoints, shouldn't be cached to prevent the leakage of personal information. It's recommended to have separate routes for static and dynamic assets, with caching disabled for the latter. https://docs.azure.cn/en-us/frontdoor/front-door-caching

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	## How it works  1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user. 2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server. 3. The origin server returns the file to an edge server in the POP. 4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days. 5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP. 6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	See also Claim elements 28.a and 28.b.
32. The method as recited in claim 28, wherein the resources comprise an HTTP-based resource.	Microsoft performs the method of claim 28, wherein the resources comprise an HTTP-based resource. For example, Microsoft hosts the web content of numerous third-party websites through its Azure services, as well as hosting Microsoft's own internet content, which each comprise an HTTP-based resource. **Azure Content Delivery Network features** Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

Features and optimizations	Front Door Standard	Front Door Premium	Front Door (classic)	CDN Standard from Microsoft (classic)
Delivery and acceleration				
Static file delivery	✓	✓	✓	✓
Dynamic site delivery	✓	✓	✓	
WebSockets	Preview	Preview		
Domains and certs				
Custom domains	✓ (DNS TXT record based domain validation)	✓ (DNS TXT record based domain validation)	✓ (CNAME based validation)	✓ (CNAME based validation)
Prevalidated domain integration with Azure PaaS service	✓	✓		
HTTPS support	✓	✓	✓	✓
Custom domain HTTPS	✓	✓	✓	✓

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-cdn-comparison

HTTP/2 support	✓	✓	✓	✓

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-cdn-comparison

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Security The CDN can deliver content over HTTPS (SSL), by using the certificate provided by the CDN, as well as over standard HTTP. To avoid browser warnings about mixed content, you might need to use HTTPS to request static content that is displayed in pages loaded through HTTPS. https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn
34. A method as recited in claim 28, wherein at least the first alias name is a domain name and the computer network further comprises at least one name server that provides domain name service (DNS) resolution, the method further comprising:	Microsoft performs the method of claim 28, wherein at least the first alias name is a domain name and the computer network further comprises at least one name server that provides domain name service (DNS) resolution. For instance, an alias name may be a domain name such as a custom domain. **Create a CNAME DNS record** Before you can use a custom domain with an Azure Content Delivery Network endpoint, you must first create a canonical name (CNAME) record with Azure DNS or your DNS provider to point to your content delivery network endpoint. A CNAME record is a DNS record that maps a source domain name to a destination domain name. For Azure Content Delivery Network, the source domain name is your custom domain name and the destination domain name is your content delivery network endpoint hostname. Azure Content Delivery Network routes traffic addressed to the source custom domain to the destination content delivery network endpoint hostname after it verifies the CNAME record. A custom domain and its subdomain can only get added to a single endpoint at a time. Use multiple CNAME records for different subdomains from the same custom domain for different Azure services. You can map a custom domain with different subdomains to the same content delivery network endpoint. https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain Microsoft provides instructions on creating alias records associated with Azure CDN endpoints for users of both Azure DNS and third-party DNS providers.

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Azure DNS | DNS Provider

Azure DNS uses alias records for Azure resources within the same subscription.

To add an alias record for your Azure Content Delivery Network endpoint:

1. Sign in to the Azure portal.
2. In the left-hand menu, select **All resources** then the Azure DNS zone for your custom domain.
3. In the DNS zone for your custom domain, select + **Record set**.
4. In **Add record set** enter or select the following information:

Expand table

Setting	Value
Name	Enter alias you want to use for your content delivery network endpoint. For example, **www**.
Type	Select **CNAME**.
Alias record set	Select **Yes**.
Alias type	Select **Azure resource**.
Choose a subscription	Select your subscription.
Azure resource	Select your Azure Content Delivery Network endpoint.

5. Change the **TTL** for the record to your value.
6. Select **OK**.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Azure DNS DNS Provider **Map the temporary cdnverify subdomain** When you map an existing domain that is in production, there are special considerations. A brief period of downtime for the domain can occur when registering your custom domain in the Azure portal. To avoid interruption of web traffic, map your custom domain to your content delivery network endpoint hostname with the Azure **cdnverify** subdomain. This process creates a temporary CNAME mapping. With this method, users can access your domain without interruption while the DNS mapping occurs. If production downtime considerations aren't a concern, you can directly map your custom domain to your content delivery network endpoint. Continue to Map the permanent custom domain. To create a CNAME record with the cdnverify subdomain: 1. Sign in to the web site of the DNS provider for your custom domain. 2. Create a CNAME record entry for your custom domain and complete the fields as shown in the following table (field names might vary): https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Expand table

Source	Type	Destination
cdnverify.www.contoso.com	CNAME	cdnverify.contoso.azureedge.net

- Source: Enter your custom domain name, including the cdnverify subdomain, in the following format: **cdnverify.<custom-domain-name>**
  - For example: **cdnverify.www.contoso.com**
- Type: Enter or select **CNAME**.
- Destination: Enter your content delivery network endpoint hostname, including the cdnverify subdomain, in the following format: **cdnverify.<endpoint-name>.azureedge.net**.
  - For example: **cdnverify.contoso.azureedge.net**

3. Save your changes.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

Map the permanent custom domain

In this section, you map the permanent custom domain to the content delivery network endpoint.

To create a CNAME record for your custom domain:

1. Sign in to the web site of the domain provider for your custom domain.
2. Create a CNAME record entry for your custom domain and complete the fields as shown in the following table (field names might vary):

Expand table

Source	Type	Destination
www.contoso.com	CNAME	contoso.azureedge.net

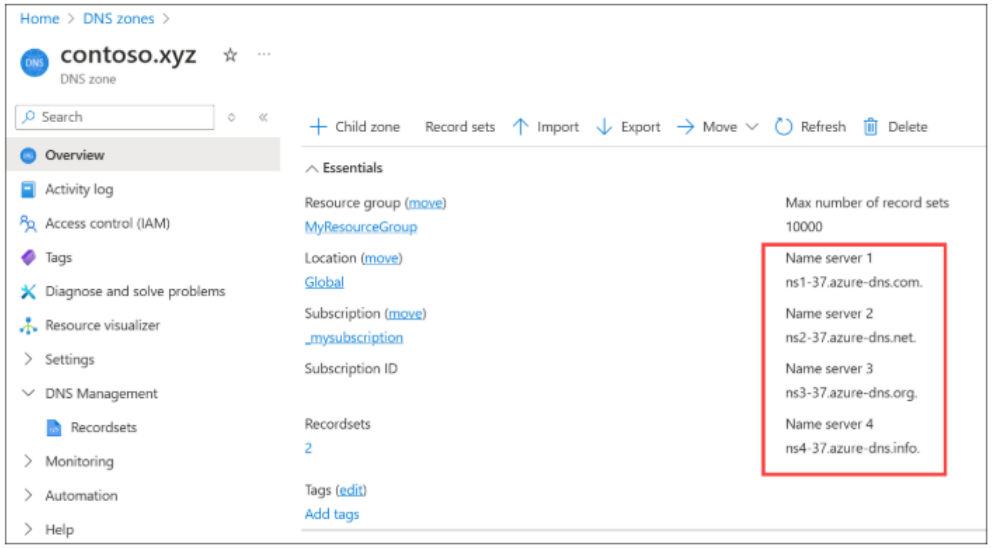
- Source: Enter your custom domain name.
  - For example: **www.contoso.com**
- Type: Enter or select **CNAME**.
- Destination: Enter your content delivery network endpoint hostname in the following format: **<endpoint-name>.azureedge.net**.
  - For example: **contoso.azureedge.net**

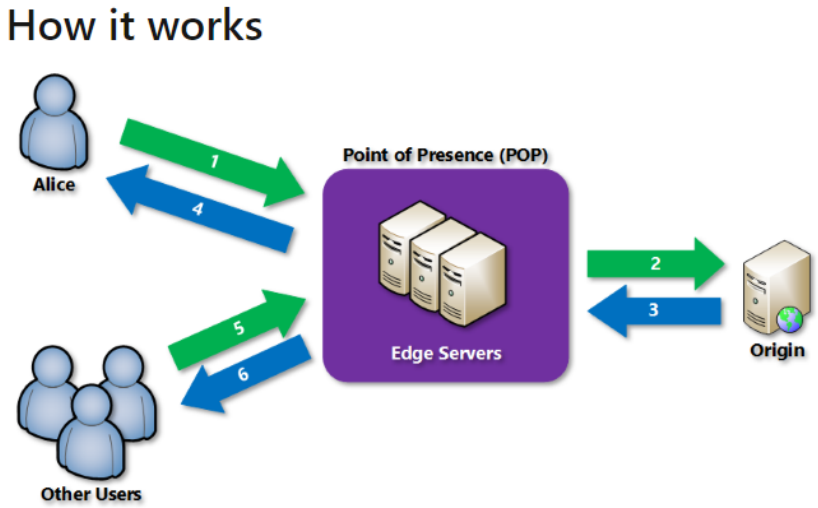
3. Save your changes.
4. If you previously created a temporary cdnverify subdomain CNAME record, delete it.
5. If you're using this custom domain in production for the first time, follow the steps for Add the custom domain with your content delivery network endpoint and Verify the custom domain.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

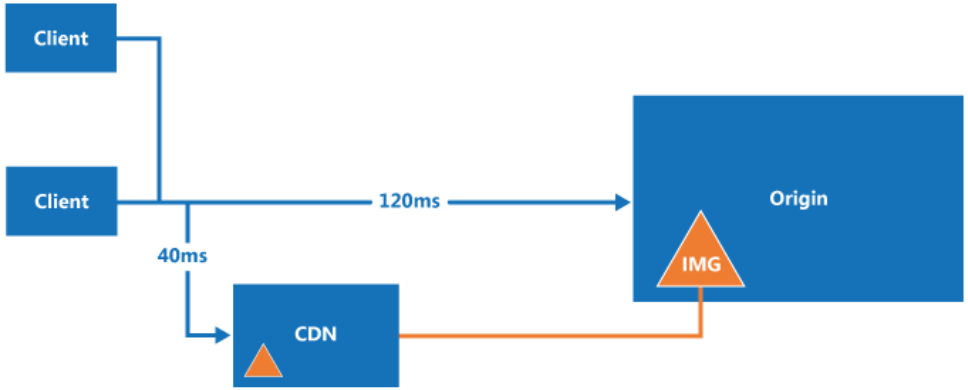
See also Claim limitation 28.a.

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	Microsoft additionally provides name servers (e.g., Azure DNS servers) that provide DNS resolution. **Tutorial: Host your domain in Azure DNS** 06/11/2024 You can use Azure DNS to host your DNS domain and manage your DNS records. By hosting your domains in Azure, you can manage your DNS records using the same credentials, APIs, tools, and billing as your other Azure services. Suppose you buy the domain `contoso.com` from a domain name registrar and then create a zone with the name `contoso.com` in Azure DNS. Since you're the owner of the domain, your registrar offers you the option to configure the name server (NS) records for your domain. The registrar stores the NS records in the `.com` parent zone. Internet users around the world are then directed to your domain in your Azure DNS zone when they try to resolve DNS records in `contoso.com`. https://learn.microsoft.com/en-us/azure/dns/dns-delegate-domain-azure-dns

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Retrieve name servers Before you can delegate your DNS zone to Azure DNS, you need to know the name servers for your zone. Azure DNS assigns name servers from a pool each time a zone is created. 1. In the Azure portal, enter **DNS zones** in the search box at the top of the portal and then select **DNS zones** from the search results. 2. In **DNS zones**, select the name of your zone. For example: **contoso.xyz**. 3. On the **Overview** page, retrieve the name servers. In this example, the DNS zone **contoso.xyz** has been assigned name servers `ns1-37.azure-dns.com`, `ns2-37.azure-dns.net`, `ns3-37.azure-dns.org`, and `ns4-37.azure-dns.info`:  Azure DNS automatically creates authoritative NS records in your zone for the assigned name servers. https://learn.microsoft.com/en-us/azure/dns/dns-delegate-domain-azure-dns
[34.a] receiving, at the at least one name server, the first alias	Microsoft receives, at the at least one name server, the first alias name as part of a client request. For instance, a request includes the domain name as part of the URL.

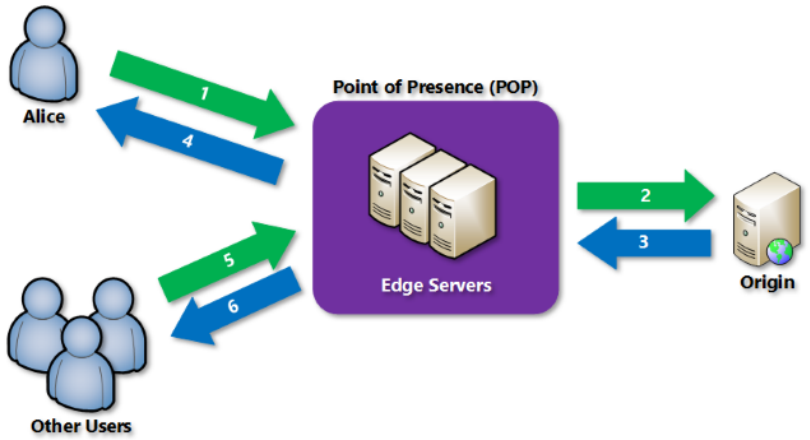
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
name as part of a client request; and	 1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user. 2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server. 3. The origin server returns the file to an edge server in the POP. 4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days. 5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP. 6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# CDN guidance Azure Storage Azure Blob Storage A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency. CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.  In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server. https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn
[34.b] resolving, at the at least one name server, the received first alias name to identify the at least one shared repeater	Microsoft resolves, at the at least one name server, the received first alias name to identify the at least one shared repeater server to deliver the first resource in response to the client request. For instance, Microsoft Azure DNS resolves domain names to identify shared repeater servers to deliver requested resources. The Azure DNS does this in response to client requests (e.g. a resource request).

<table>
<tr><th>U.S. Patent No. 8,478,903</th><th>Microsoft Accused Functionalities</th></tr>
<tr><td>server to deliver the first resource in response to the client request.</td><td>

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.
2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.
3. The origin server returns the file to an edge server in the POP.
4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.
5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.
6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview</td></tr>
</table>

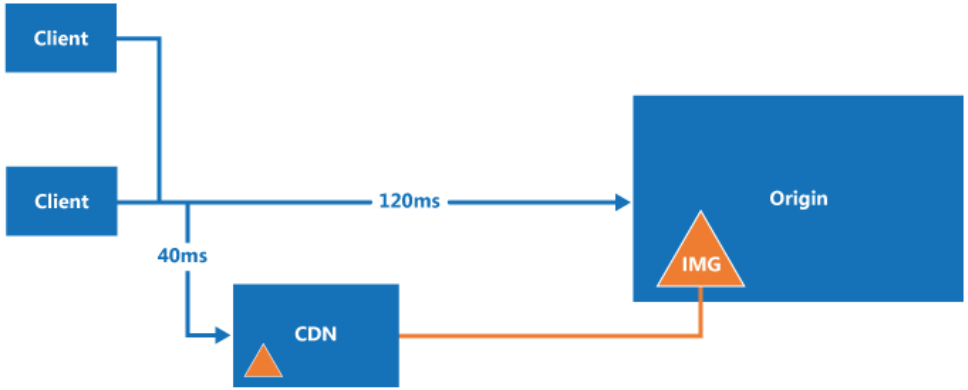
U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
	# Tutorial: Host your domain in Azure DNS 06/11/2024 You can use Azure DNS to host your DNS domain and manage your DNS records. By hosting your domains in Azure, you can manage your DNS records using the same credentials, APIs, tools, and billing as your other Azure services. Suppose you buy the domain `contoso.com` from a domain name registrar and then create a zone with the name `contoso.com` in Azure DNS. Since you're the owner of the domain, your registrar offers you the option to configure the name server (NS) records for your domain. The registrar stores the NS records in the `.com` parent zone. Internet users around the world are then directed to your domain in your Azure DNS zone when they try to resolve DNS records in `contoso.com`. https://learn.microsoft.com/en-us/azure/dns/dns-delegate-domain-azure-dns
36. A method as recited in claim 28, further comprising: [36.a] rejecting the client request for the particular resource if the alias name received with the client request is not included the table.	Microsoft performs the method of claim 28 as set forth above. Further, Microsoft rejects the client request for the particular resource if the alias name received with the client request is not included in the table. For instance, Microsoft Azure CDN and/or Azure Front Door reject client requests including alias names that are not included in the table (e.g., non-validated domains).

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities

- This tutorial uses the CNAME record type for multiple purposes:
  - *Traffic routing* can be accomplished with a CNAME record as well as A or AAAA record types in Azure DNS. To apply, use the following steps to replace the CNAME record with the record type of your choice.
  - A CNAME record is **required** for custom domain *ownership validation* and must be available when adding the custom domain to a content delivery network endpoint. More details in the following section.

https://learn.microsoft.com/en-us/azure/cdn/cdn-map-content-to-custom-domain

Endpoints - Validate Custom Domain

Service: CDN

API Version: 2025-04-15

Validates the custom domain mapping to ensure it maps to the correct CDN endpoint in DNS.



https://learn.microsoft.com/en-us/rest/api/cdn/endpoints/validate-custom-domain?view=rest-cdn-2025-04-15&tabs=HTTP

<table>
<tr><th>U.S. Patent No. 8,478,903</th><th>Microsoft Accused Functionalities</th></tr>
<tr><td></td><td>

Request Body

Expand table

Name	Required	Type	Description
hostName	True	string	The host name of the custom domain. Must be a domain name.

Responses

Expand table

Name	Type	Description
200 OK	Validate Custom Domain Output	OK. The request has succeeded.
Other Status Codes	ErrorResponse	CDN error response describing why the operation failed.

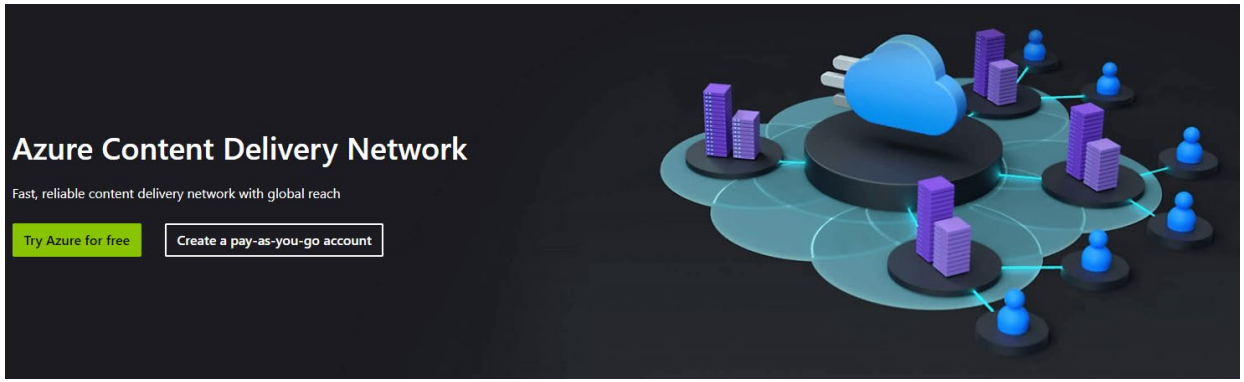
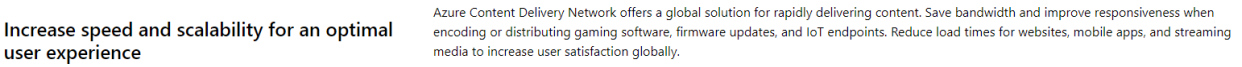
https://learn.microsoft.com/en-us/rest/api/cdn/endpoints/validate-custom-domain?view=rest-cdn-2025-04-15&tabs=HTTP
</td></tr>
<tr><td>37[Pre]. A method, in a content delivery system operative in a computer network for delivering content to client machines and comprising at least one shared repeater server operable to replicate resources stored on a plurality of origin</td><td>Microsoft performs a method, in a content delivery system operative in a computer network for delivering content to client machines and comprising at least one shared repeater server operable to replicate resources stored on a plurality of origin servers. For instance, Microsoft Azure CDN and/or Front Door comprise a plurality of origin servers hosting associated resources as well as shared repeater servers (e.g., Edge Nodes, Endpoints, or Points of Presence) to replicate resources associated with the origin servers.

See Claim limitation 28[Pre].</td></tr>
</table>

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
servers, the method comprising:	
[37.a] associating a plurality of alias names with the at least one shared repeater server, each of said plurality of alias names being associated with an origin server, wherein a first alias name of said plurality of alias names is associated with a first origin server, and at least a second alias name of said plurality of alias names is associated with a second origin server distinct from said first origin server,	Microsoft associates a plurality of alias names with the at least one shared repeater server, each of said plurality of alias names being associated with an origin server, wherein a first alias name of said plurality of alias names is associated with a first origin server, and at least a second alias name of said plurality of alias names is associated with a second origin server distinct from said first origin server. See Claim limitations 28.a and 28.b.
[37.b] providing at least one table associating alias names with origin servers having content located thereon, wherein said content is authorized for delivery to client machines via the at least one shared repeater server, wherein the origin servers comprise the first origin server and the second origin server;	Microsoft provides at least one table associating alias names with origin servers having content located thereon, wherein said content is authorized for delivery to client machines via the at least one shared repeater server, wherein the origin servers comprise the first origin server and the second origin server. For instance, Microsoft Azure CDN and/or Azure Front Door utilize a table to correlate shared repeater servers to origin servers hosting the requested content. See Claim limitation 28.c.
[37.c] wherein requests for a first resource originating on the first origin server are directed, based at least in part on said first alias name, to the at least one shared repeater server for delivery of the first resource	Requests for a first resource originating on the first origin server are directed, based at least in part on said first alias name, to the at least one shared repeater server for delivery of the first resource from said at least one repeater server. Requests for resources located on a first origin server are directed based at least in part on the alias name associated with the shared repeater server. For instance, Microsoft Azure CDN and/or Azure Front Door support custom domain alias creation for multiple CDN hosting clients accessing content from various origin servers cached by Azure CDN endpoints. See Claim limitation 28.a.

U.S. Patent No. 8,478,903	Microsoft Accused Functionalities
from said at least one repeater server; and	
[37.d] wherein requests for a second resource originating on the second origin server are directed, based at least in part on said second alias name, to the at least one shared repeater server for delivery of the second resource from said at least one repeater server; and	Requests for a second resource originating on the second origin server are directed, based at least in part on said second alias name, to the at least one shared repeater server for delivery of the second resource from said at least one repeater server. Requests for second resources located on a second origin server are directed based at least in part on the second alias name associated with the shared repeater server. For instance, Microsoft Azure CDN and/or Azure Front Door support custom domain alias creation for multiple CDN hosting clients accessing content from various origin servers cached by Azure CDN endpoints. See Claim limitation 28.b.
[37.e] wherein the at least one repeater server uses the at least one table and an alias name received with a client request for a resource to determine an origin server associated with the requested resource.	The at least one repeater server uses the at least one table and an alias name received with a client request for a resource to determine an origin server associated with the requested resource. For instance, Microsoft Azure CDN and/or Azure Front Door include servers in the edge infrastructure that cache content for multiple Azure CDN customers and which use a table to determine an origin server for a CDN customer. See Claim limitation 28.d.
38. The method as in claim 37, wherein a copy of the at least one table is located on each of a plurality of repeater servers.	Microsoft performs the method of claim 37, wherein a copy of the at least one table is located on each of a plurality of repeater servers. For instance, Microsoft Azure CDN and/or Azure Front Door repeater servers (e.g., Edge Nodes, Endpoints, or Points of Presence) store a copy of the table listing origin servers in order to correlate requests for content to origin servers for a CDN customer. See Claim limitation 28.d.
39. A method as recited in claim 37, further comprising: rejecting the client request for the resource if the alias name received with the client request is not included the table.	Microsoft performs the method of claim 37, further comprising rejecting the client request for the resource if the alias name received with the client request is not included the table. For instance, Microsoft Azure CDN and/or Azure Front Door reject client requests including alias names that are not included in the table (e.g., non-validated domains). See Claim 36.

EXHIBIT B

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
1[Pre]. A content delivery method for distributing objects maintained on at least one origin server via a content delivery network (CDN), the method comprising:	Microsoft performs a content delivery method for distributing objects maintained on at least one origin server via a content delivery network (CDN) as set forth below. For instance, Microsoft Azure CDN and/or Azure Front Door distribute objects maintained on origin servers via content delivery networks in the manner of claim 1.  https://azure.microsoft.com/en-us/products/cdn  https://azure.microsoft.com/en-us/products/cdn  https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

Improved performance and user experience

Users expect a fast, reliable, and personalized web experience wherever they are. Content Delivery Network helps reduce latency and improve performance for high-bandwidth content by distributing user requests and serving content directly from edge servers. This brings the content closer to users and sends less traffic to the origin point, delivering superior online experiences.

https://azure.microsoft.com/en-us/products/cdn

Global coverage and massive scalability

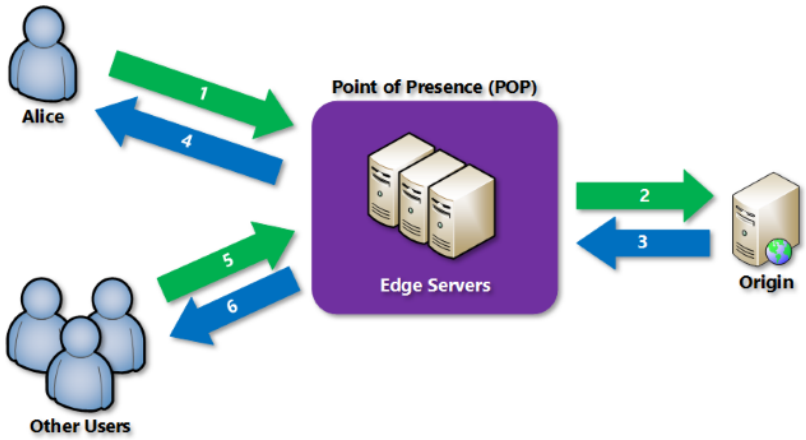
Scale on the fly with no downtime using a platform that responds to traffic fluctuations. A distributed, global presence enables Content Delivery Network to handle sudden traffic spikes and heavy loads—for example, during major product launches or global sporting events—without new infrastructure costs or capacity concerns. Global, distributed points of presence (PoPs) ensure fast delivery of content and protects the origin from excessive load.

https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	Protection at the edge Optimize security through features like custom domain HTTPS and DDoS mitigation. Enable Content Delivery Network to put edge servers on the front line serving and protect the origin server from malicious attacks. Set advanced rules to block requests or validate users and use geo-filtering to ensure the content is delivered only to specific regions. https://azure.microsoft.com/en-us/products/cdn **Automate processes and workflows with DevOps** Azure APIs and developer tools give you the performance, reliability, and security needed to build and automate global applications fast. A unified API allows you to deploy Content Delivery Network endpoints from each provider and still manage them the same way. Azure APIs also provide the flexibility to optimize features for availability and performance. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

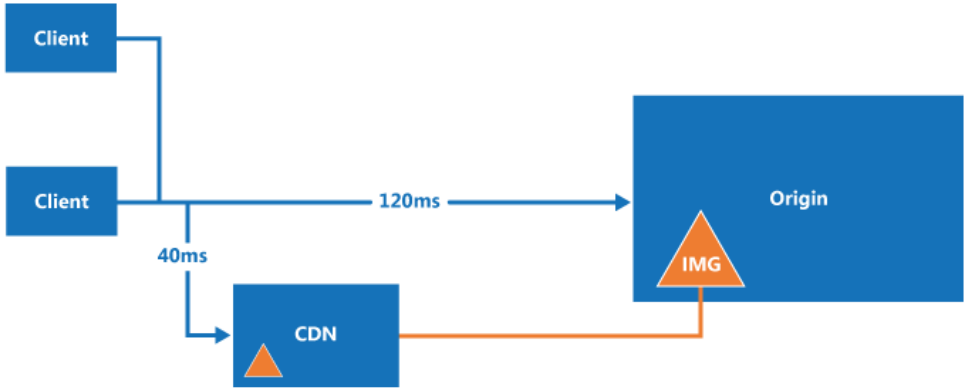
U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	This article lists current metros containing point of presence (POP) locations, sorted by region, for Azure Content Delivery Network products. Each metro might contain more than one POP. For example, Azure Content Delivery Network from Microsoft has 192 POPs across 109 metro cities. **ⓘ Important** Each Azure Content Delivery Network product has a distinct way of building its content delivery network infrastructures, hence Microsoft recommends against using POP locations to decide which Azure Content Delivery Network product to use. Instead, you should consider its features and end-user performance. Test the performance with each Azure Content Delivery Network product to choose the right product for your users. https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

Region	Cities
North America	Etobicoke, Canada (2) Montreal, Canada Vancouver, Canada (2) Querétaro, Mexico (2) Atlanta, GA, USA (3) Boydton, VA, USA (2) Chaska, MN, USA (2) Cheyenne, WY, USA (2) Chicago, IL, USA (4) Dallas, TX, USA (4) Des Moines,IA, USA (3) Detroit, MI, USA Englewood, CO, USA (2) Honolulu, HI, USA Houston, TX, USA (3) Jacksonville, FL, USA (2) Las Vegas, NV, USA (3) Los Angeles, CA, USA (2) Manassas, VA, USA (3) Memphis, TN, USA Miami, FL, USA (4) Minneapolis, MN, USA (2) Needham Heights, MA, USA (2) Nashville, TN, USA Newark, NJ, USA New York, NY, USA (2) Philadelphia, PA, USA Phoenix, AZ, USA Plano, TX, USA Portland, OR, USA Quincy, WA, USA (3) San Antonio, TX, USA (4) San Jose, CA, USA (4) Salt Lake City, UT, USA (3) Seattle, WA, USA (2) Secaucus, NJ, USA (2) Southfield, MI, USA Tempe, AZ, USA

https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

Microsoft Azure CDN additionally includes products such as Azure Front Door as well as Standard and Premium CDN services from Edgio (formerly Verizon).

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

What differentiates Azure CDN from Azure Front Door?

CDN Standard from Microsoft (classic) is on retirement path for September 30, 2027, and Azure Front Door is the modern CDN replacement product offering. Azure Front Door is a globally distributed network of servers that can efficiently deliver web content to consumers. It stores cached content on the edge servers' PoP locations that are close to users, thereby minimizing network latency. In addition, Azure Front Door is an application delivery network service that offers Layer 7 load-balancing capabilities for your applications. It delivers content while protecting apps, APIs, and websites from cyberthreats, and works seamlessly with Azure services and modern content delivery network (CDN) technology.

https://azure.microsoft.com/en-us/products/cdn

Azure Content Delivery Network includes three products:

- Azure CDN Standard from Microsoft
- Azure CDN Standard from Edgio (formerly Verizon)
- Azure CDN Premium from Edgio (formerly Verizon).

https://learn.microsoft.com/en-us/azure/cdn/cdn-features

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	Microsoft also performs the methods of the asserted claims of the '466 Patent to serve Microsoft resources and services using the same or similar infrastructure. For example, Microsoft uses internal CDN technologies such as its Azure CDN to support at least Xbox gaming, Office 365, Windows updates, and web resources like Linkedin and Bing. • Built on Microsoft's massive-scale private global network, Azure Front Door is a proven platform used to power some of the largest and latency sensitive global services at Microsoft, such as Microsoft Office 365, Bing, LinkedIn, and Xbox. https://azure.microsoft.com/en-us/blog/azure-front-door-enhances-secure-cloud-cdn-with-intelligent-threat-protection/ Azure CDN provides a world class platform to let you reduce load times, save bandwidth, and speed responsiveness across your businesses diverse workflows. Azure CDN from Microsoft enables Azure customers to use and deliver content from the same global CDN network leveraged by Microsoft properties such as Office 365, Skype, Bing, OneDrive, Windows, and Xbox. Connectivity within Microsoft's network along with new Regional Caching capabilities enables more consistent, more predictable cache fill performance by providing multi-tier caching along direct, private access to content in Azure from each CDN edge point of presence (POP). Azure CDN from Microsoft entered public preview providing access to 54 global Edge POPs in 33 countries and 16 Regional Cache POPs. https://azure.microsoft.com/en-us/blog/announcing-microsoft-s-own-cdn-network/

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

What powers Windows 365?

Not surprising, Microsoft's own cloud computing platform, Azure. Specifically, **Azure Virtual Desktop**, which can be used to create Windows virtual machines on Redmond's servers, then stream those desktops to PC and non-PC devices.

https://www.computerworld.com/article/1614540/what-is-windows-365-microsoft-s-cloud-based-os-explained.html

Use the Office 365 Content Delivery Network (CDN) with SharePoint Online

02/28/2025

You can use the built-in Office 365 Content Delivery Network (CDN) to host static assets to provide better performance for your SharePoint Online pages. The Office 365 CDN improves performance by caching static assets closer to the browsers requesting them, which helps to speed up downloads and reduce latency. Also, the Office 365 CDN uses the HTTP/2 protocol for improved compression and HTTP pipelining. The Office 365 CDN service is included as part of your SharePoint Online subscription.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	The Office 365 CDN is composed of multiple CDNs that allow you to host static assets in multiple locations, or *origins*, and serve them from global high-speed networks. Depending on the kind of content you want to host in the Office 365 CDN such as you can add **public** origins. If you're already familiar with the way that CDNs work, you only need to complete a few steps to enable the Office 365 CDN for your tenant. This article describes how. Read on for information about how to get started hosting your static assets. https://learn.microsoft.com/en-us/microsoft-365/enterprise/use-microsoft-365-cdn-with-spo?view=o365-worldwide On the Microsoft side, the company uses primarily Azure Front Door a cloud CDN platform and some third-party CDNs for some static delivery of content i.e Xbox downloads, Windows Updates. Microsoft does not https://www.streamingmediablog.com/2020/10/gaming-console-launch.html

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	Teams is built on a microservices architecture, with a few hundred microservices working cohesively to deliver our product's many features including messaging, meetings, files, calendar, and apps. Using microservices helps each of our component teams to work and release their changes independently. Azure is the cloud platform that underpins all of Microsoft's cloud services, including Microsoft Teams. Our workloads run in Azure virtual machines (VMs), with our older services being deployed through Azure Cloud Services and our newer ones on Azure Service Fabric. Our primary storage stack is Azure Cosmos DB, with some services using Azure Blob Storage. We count on Azure Cache for Redis for increased throughput and resiliency. We leverage Traffic Manager and Azure Front Door to route traffic where we want it to be. We use Queue Storage and Event Hubs to communicate, and we depend on Azure Active Directory to manage our tenants and users. • **Resiliency-optimized caches**: We leverage caches between our components extensively, for both performance and resiliency. Caches help reduce average latency and provide a source of data in case a downstream service is unavailable. Keeping data in caches for a long time introduces data freshness issues yet keeping data in caches for a long time is the best defense against downstream failures. We focus on Time to Refresh (TTR) to our cache data as well as Time to Live (TTL). By setting a long TTL and a shorter TTR value, we can fine-tune how fresh to keep our data versus how long we want data to stick around whenever a downstream dependency fails. https://azure.microsoft.com/en-us/blog/advancing-microsoft-teams-on-azure-operating-at-pandemic-scale/

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

Background

A CDN is a system of distributed servers that delivers web content to users from geographical locations closest to them. At LinkedIn, CDNs deliver static content, such as CSS and JavaScript on the site, as well as member-generated content, such as profile pictures, media shares, and video uploads, to our members around the world. The content is delivered from edges that are closest to the user in question.

The Edge SRE team runs LinkedIn's edge infrastructure, including four external CDNs, an in-house CDN, three Domain Name System (DNS) platforms, and all LinkedIn points of presence (PoPs). We build and manage tools to automate all aspects of our Edge stack.

https://www.linkedin.com/blog/engineering/data-management/enabling-dual-stack-on-linkedin-cdns\

LinkedIn recently described how it uses the edge to support the popular professional networking web site. The engineering team wrote a blog post to talk about edge deployments and its migration away from an in-house solution to Azure Edge.

LinkedIn is one of many large web properties that is owned by a hyperscaler cloud provider — Microsoft, in this case — but doesn't actually use the in-house cloud. YouTube is another example of this — it has yet to be migrated to GCP.

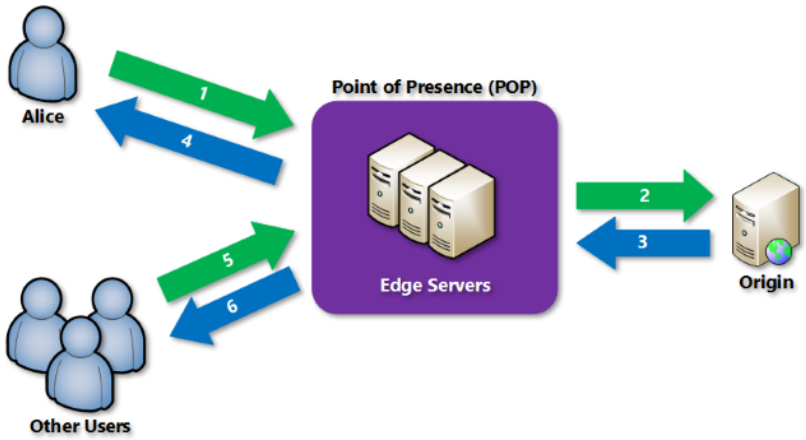
LinkedIn actually had its own quite extensive in-house CDN, with nineteen PoPs (points of presence) in ten countries. This is a common practice for major content owners as it defrays transport costs and improves performance. The nodes can cache content and perform tasks such as SSL offloading. But as LinkedIn noted, Microsoft's CDN – Azure Front Door – gives access to 300 PoPs around the globe, which is a big step up in terms of reach.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	https://www.edgeir.com/how-linkedin-is-using-azure-front-door-for-edge-build-out-20220306
[1.a] providing a plurality of servers in the CDN, wherein each of the plurality of servers is distinct from the at least one origin server, the plurality of servers comprising a plurality of edge servers and a plurality of parent servers;	Microsoft provides a plurality of servers in the CDN, wherein each of the plurality of servers is distinct from the at least one origin server, the plurality of servers comprising a plurality of edge servers and a plurality of parent servers. For instance, Microsoft Azure CDN and/or Azure Front Door provide servers including edge servers (e.g., EdgeNodes, Edge POPs) and parent servers (e.g., regional cache POPs or origin shield servers) distinct from the origin server. **Improved performance and user experience** Users expect a fast, reliable, and personalized web experience wherever they are. Content Delivery Network helps reduce latency and improve performance for high-bandwidth content by distributing user requests and serving content directly from edge servers. This brings the content closer to users and sends less traffic to the origin point, delivering superior online experiences. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	Global coverage and massive scalability Scale on the fly with no downtime using a platform that responds to traffic fluctuations. A distributed, global presence enables Content Delivery Network to handle sudden traffic spikes and heavy loads—for example, during major product launches or global sporting events—without new infrastructure costs or capacity concerns. Global, distributed points of presence (PoPs) ensure fast delivery of content and protects the origin from excessive load. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

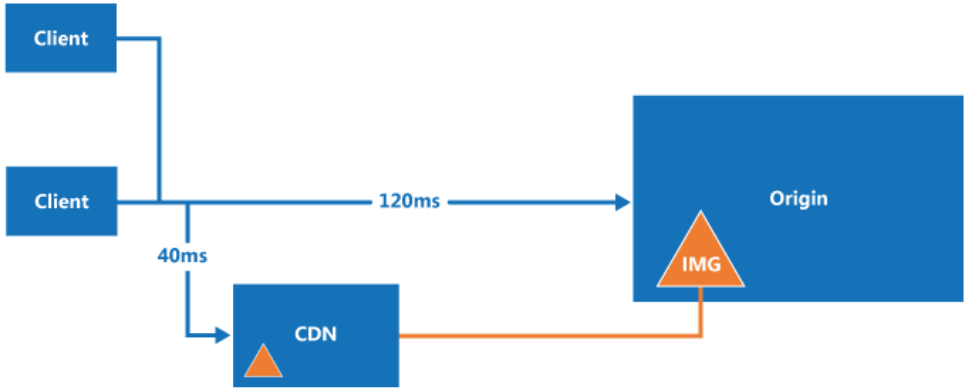
U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

Edge Nodes - List

Reference

Feedback

Service: CDN

API Version: 2024-02-01

Edgenodes are the global Point of Presence (POP) locations used to deliver CDN content to end users.

HTTP Copy Try It

```
GET https://management.azure.com/providers/Microsoft.Cdn/edgenodes?api-version=2024-02-01
```

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

EdgeNode

Edgenode is a global Point of Presence (POP) location used to deliver CDN content to end users.

Expand table

Name	Type	Description
id	string	Resource ID.
name	string	Resource name.
properties.ipAddressGroups	IpAddressGroup[]	List of ip address groups.
systemData	SystemData	Read only system data
type	string	Resource type.

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

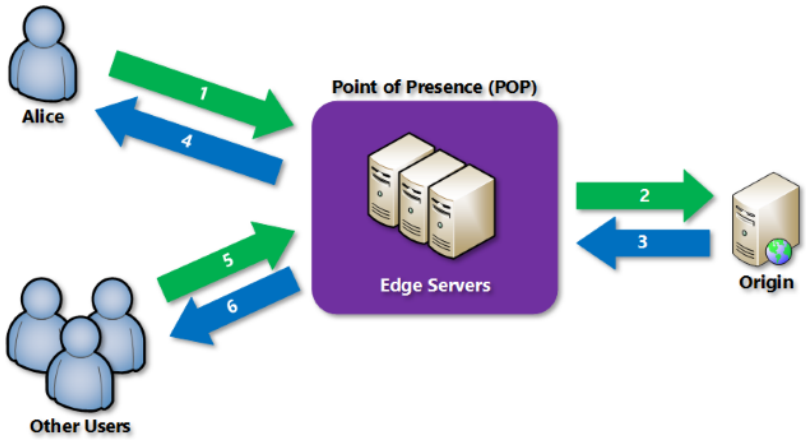
U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	Connectivity within Microsoft's network along with new Regional Caching capabilities enables more consistent, more predictable cache fill performance by providing multi-tier caching along direct, private access to content in Azure from each CDN edge point of presence (POP). Azure CDN from Microsoft entered public preview providing access to 54 global Edge POPs in 33 countries and 16 Regional Cache POPs.  https://azure.microsoft.com/es-es/blog/new-locations-for-azure-cdn-now-available/
[1.b] directing a request by a client for an object to a first edge server in the CDN, regardless of whether the first edge server has the requested object, wherein said first	Microsoft directs a request by a client for an object to a first edge server in the CDN, regardless of whether the first edge server has the requested object, wherein said first edge server was selected based at least in part on network traffic conditions. For instance, in Azure CDN and/or Azure Front Door, client requests are directed to edge nodes selected at least in part based on network traffic conditions (e.g. a performance traffic-routing method, a Traffic Manager policy, a load balancing policy, etc.).

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
edge server was selected based at least in part on network traffic conditions;	**Improved performance and user experience** Users expect a fast, reliable, and personalized web experience wherever they are. Content Delivery Network helps reduce latency and improve performance for high-bandwidth content by distributing user requests and serving content directly from edge servers. This brings the content closer to users and sends less traffic to the origin point, delivering superior online experiences. https://azure.microsoft.com/en-us/products/cdn **Global coverage and massive scalability** Scale on the fly with no downtime using a platform that responds to traffic fluctuations. A distributed, global presence enables Content Delivery Network to handle sudden traffic spikes and heavy loads—for example, during major product launches or global sporting events—without new infrastructure costs or capacity concerns. Global, distributed points of presence (PoPs) ensure fast delivery of content and protects the origin from excessive load. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	Protection at the edge Optimize security through features like custom domain HTTPS and DDoS mitigation. Enable Content Delivery Network to put edge servers on the front line serving and protect the origin server from malicious attacks. Set advanced rules to block requests or validate users and use geo-filtering to ensure the content is delivered only to specific regions. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

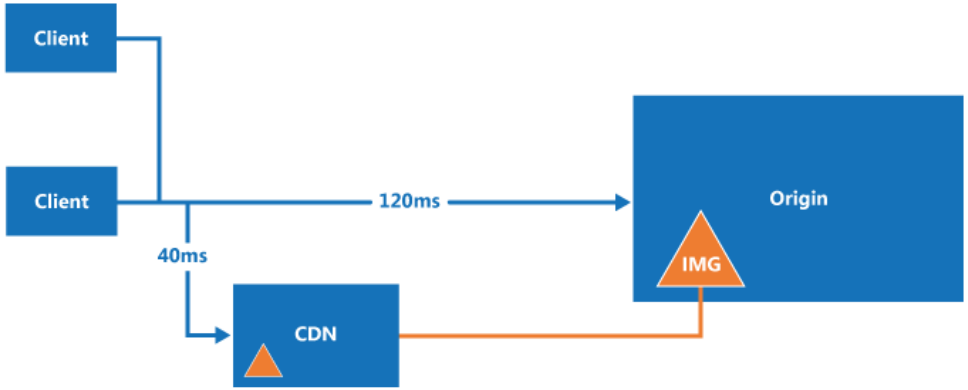
U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



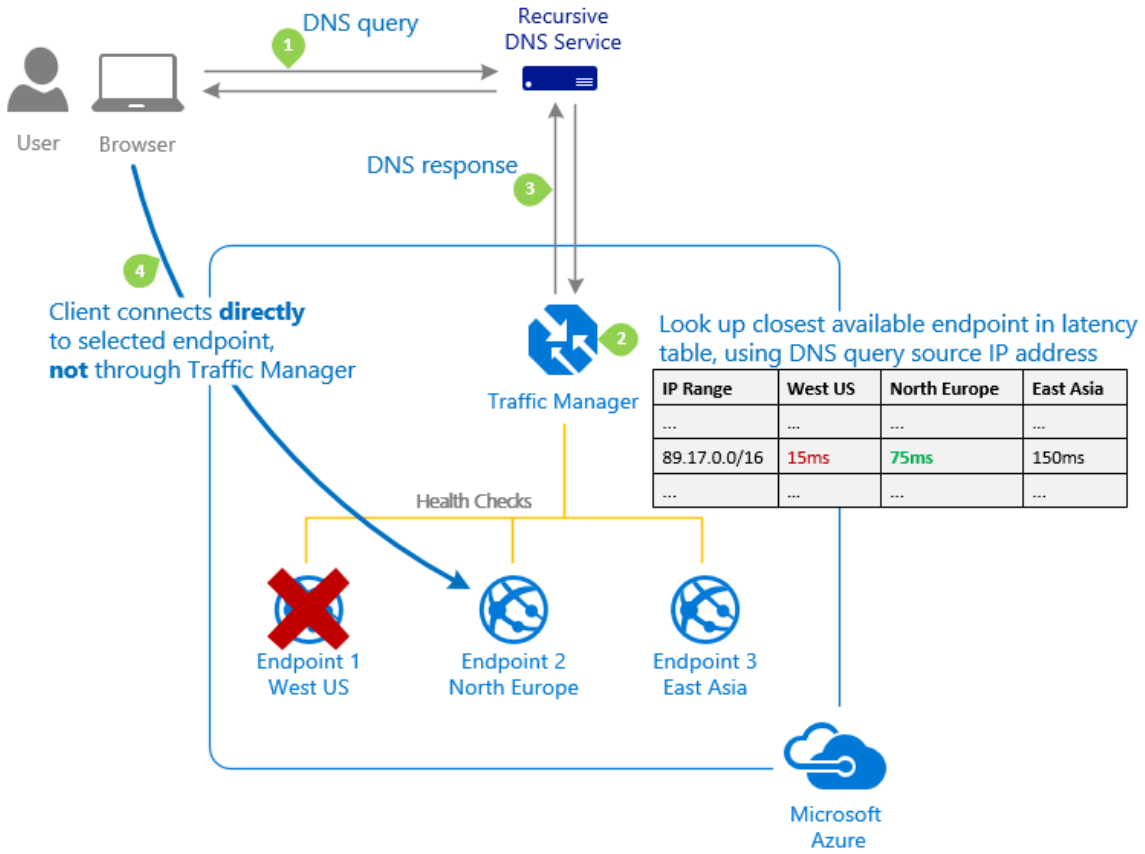
In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

Performance traffic-routing method

Deploying endpoints in two or more locations across the globe can improve the responsiveness of your applications. With the 'Performance' traffic-routing method, you can route traffic to the location that is 'closest' to you.



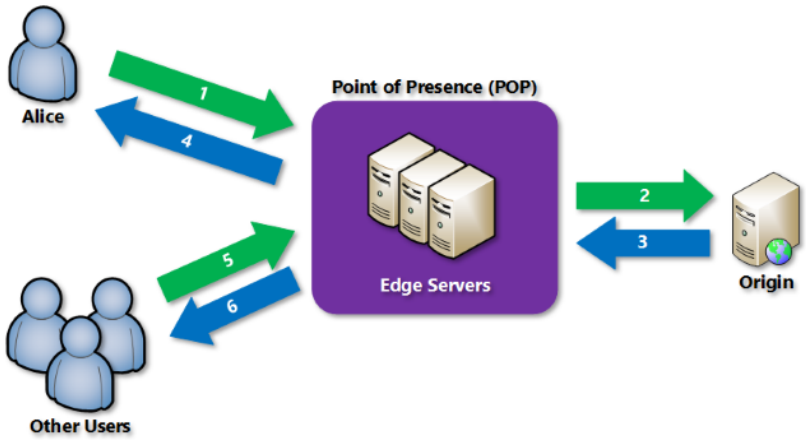
https://learn.microsoft.com/en-us/azure/traffic-manager/traffic-manager-routing-methods

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	The 'closest' endpoint isn't necessarily closest as measured by geographic distance. Instead, the 'Performance' traffic-routing method determines the closest endpoint by measuring network latency. Traffic Manager maintains an Internet Latency Table to track the round-trip time between IP address ranges and each Azure datacenter. Traffic Manager looks up the source IP address of the incoming DNS request in the Internet Latency Table. Traffic Manager then chooses an available endpoint in the Azure datacenter that has the lowest latency for that IP address range. Then Traffic Manager returns that endpoint in the DNS response. As explained in How Traffic Manager Works, Traffic Manager doesn't receive DNS queries directly from clients. Instead, DNS queries come from the recursive DNS service that the clients are configured to use. As such, the IP address used to determine the 'closest' endpoint isn't the client's IP address, but it's the IP address of the recursive DNS service. This IP address is a good proxy for the client. Traffic Manager regularly updates the Internet Latency Table to account for changes in the global Internet and new Azure regions. However, application performance varies based on real-time variations in load across the Internet. Performance traffic-routing doesn't monitor load on a given service endpoint. If an endpoint becomes unavailable, Traffic Manager won't include it in the DNS query responses. https://learn.microsoft.com/en-us/azure/traffic-manager/traffic-manager-routing-methods
[1.c] if the first edge server has the requested object, serving the requested object to the client from the first edge server;	If the first edge server (e.g. edge node or POP server) has the requested object, Microsoft serves the requested object to the client from the first edge server. **Improved performance and user experience** Users expect a fast, reliable, and personalized web experience wherever they are. Content Delivery Network helps reduce latency and improve performance for high-bandwidth content by distributing user requests and serving content directly from edge servers. This brings the content closer to users and sends less traffic to the origin point, delivering superior online experiences.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	https://azure.microsoft.com/en-us/products/cdn **Global coverage and massive scalability** Scale on the fly with no downtime using a platform that responds to traffic fluctuations. A distributed, global presence enables Content Delivery Network to handle sudden traffic spikes and heavy loads—for example, during major product launches or global sporting events—without new infrastructure costs or capacity concerns. Global, distributed points of presence (PoPs) ensure fast delivery of content and protects the origin from excessive load. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

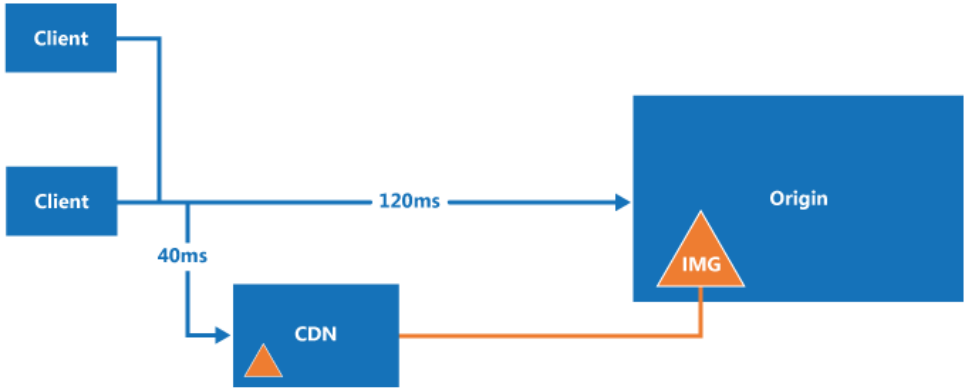
U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

Edge Nodes - List

Reference

Feedback

Service: CDN

API Version: 2024-02-01

Edgenodes are the global Point of Presence (POP) locations used to deliver CDN content to end users.

HTTP Copy Try It

```
GET https://management.azure.com/providers/Microsoft.Cdn/edgenodes?api-version=2024-02-01
```

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

EdgeNode

Edgenode is a global Point of Presence (POP) location used to deliver CDN content to end users.

Expand table

Name	Type	Description
id	string	Resource ID.
name	string	Resource name.
properties.ipAddressGroups	IpAddressGroup[]	List of ip address groups.
systemData	SystemData	Read only system data
type	string	Resource type.

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

Introduction to caching

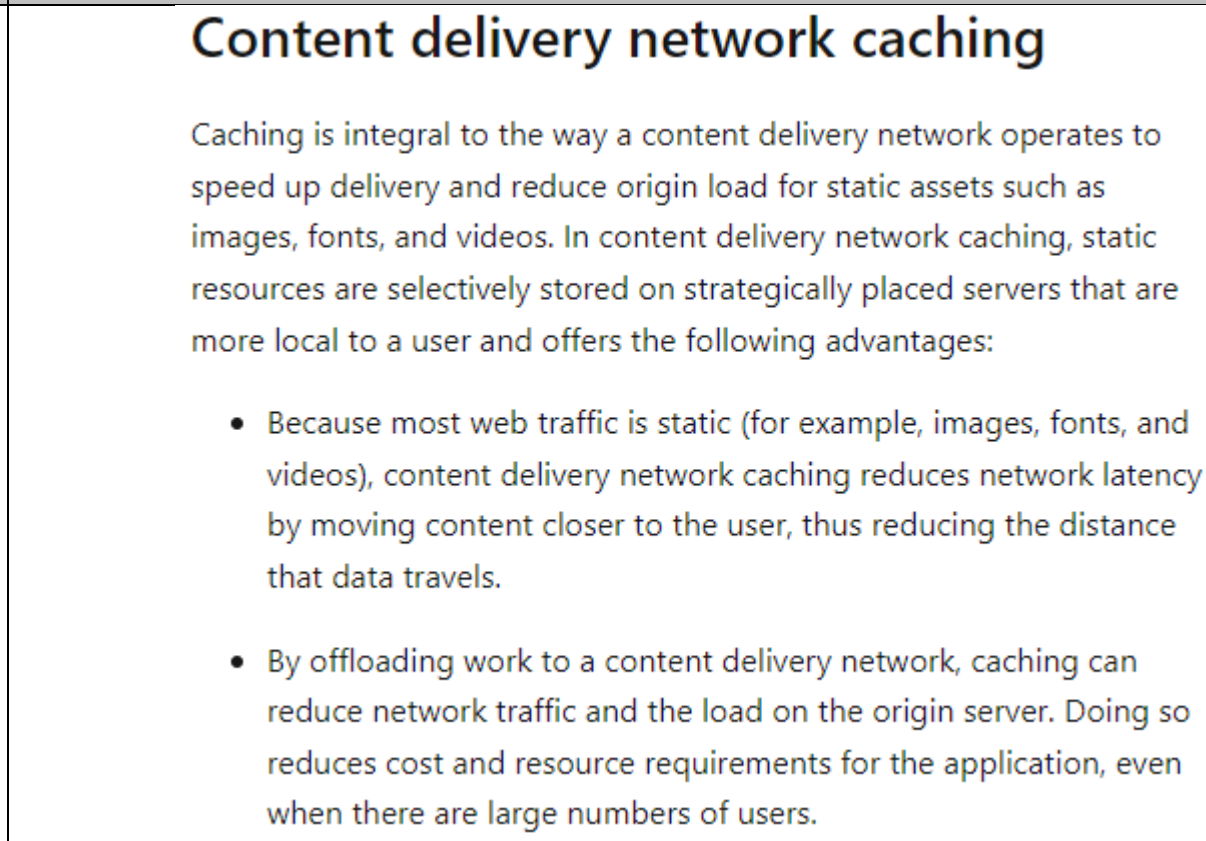
Caching is the process of storing data locally so that future requests for that data can be accessed more quickly. In the most common type of caching, web browser caching, a web browser stores copies of static data locally on a local hard drive. By using caching, the web browser can avoid making multiple round-trips to the server and instead access the same data locally, thus saving time and resources. Caching is well-suited for locally managing small, static data such as static images, CSS files, and JavaScript files.

Similarly, caching is used by a content delivery network on edge servers close to the user to avoid requests traveling back to the origin and reducing end-user latency. Unlike a web browser cache, which is used only for a single user, the content delivery network has a shared cache. In a content delivery network shared cache, a file request by a user can be used by another user, which greatly decreases the number of requests to the origin server.

Dynamic resources that change frequently or are unique to an individual user can't be cached. Those types of resources, however, can take advantage of dynamic site acceleration (DSA) optimization on the Azure content delivery network for performance improvements.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	Caching can occur at multiple levels between the origin server and the end user: • Web server: Uses a shared cache (for multiple users). • Content delivery network: Uses a shared cache (for multiple users). • Internet service provider (ISP): Uses a shared cache (for multiple users). • Web browser: Uses a private cache (for one user). Each cache typically manages its own resource freshness and performs validation when a file is stale. This behavior is defined in the HTTP caching specification, RFC 7234 . https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	 https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works
[1.d] otherwise, sending the client a response that causes the client to be redirected to a second server in the CDN to handle the request, wherein the second server	If the first edge server does not have the requested object, Microsoft sends the client a response that causes the client to be redirected to a second server in the CDN to handle the request, wherein the second server comprises a parent server of the first edge server. For instance, if a first edge server (e.g. edge node or POP server) does not have a requested object, the client is sent a response to be redirected to a second server (e.g., a parent, origin shield, or regional POP) in the CDN to handle the request.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
comprises a parent server of the first edge server; and	

SentToOriginShield (deprecated) * **See notes on deprecation in the following section.**	If true, it means that request was answered from origin shield cache instead of the edge pop. Origin shield is a parent cache used to improve cache hit ratio.
isReceivedFromClient	If true, it means that the request came from the client. If false, the request is a miss in the edge (child POP) and is responded from origin shield (parent POP).

https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log

Sent to origin shield deprecation

The raw log property **isSentToOriginShield** is deprecated and replaced by a new field **isReceivedFromClient**. Use the new field if you're already using the deprecated field.

Raw logs include logs generated from both CDN edge (child POP) and origin shield. Origin shield refers to parent nodes that are strategically located across the globe. These nodes communicate with origin servers and reduce the traffic load on origin.

For every request that goes to origin shield, there are two log entries:

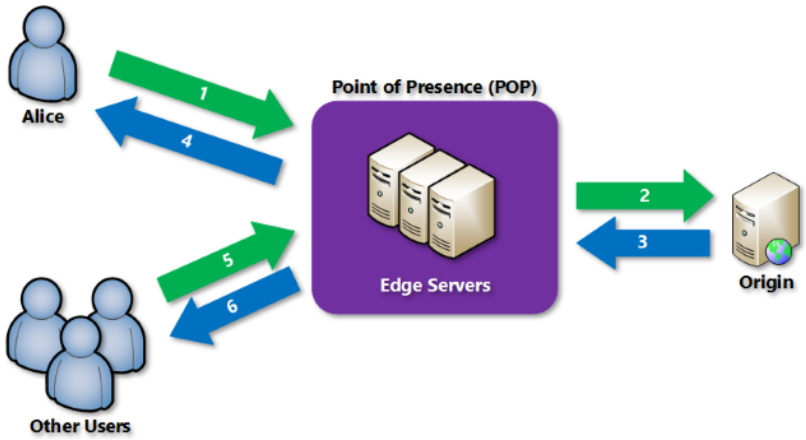
- One for edge nodes
- One for origin shield.

https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	Availability zones Many Azure regions provide availability zones. Availability zones are independent sets of datacenters that contain isolated power, cooling, and network connections. Availability zones are physically located close enough together to provide a low-latency network, but far enough apart to provide fault isolation from such things as storms and isolated power outages. Most Azure services provide built-in support for availability zones and you can decide how to use them to meet your needs. When you design an Azure solution, you should use availability zones to provide redundancy and fault isolation. To learn more about availability zones, see What are availability zones?. https://learn.microsoft.com/en-us/azure/reliability/regions-overview
[1.e] receiving, at the second server, a second request for the object from the client and serving the requested object to the client from the second server, wherein if a copy of the requested object is not stored on the second server, directing the client to another server in the CDN.	Microsoft receives, at the second server, a second request for the object from the client and serving the requested object to the client from the second server, wherein if a copy of the requested object is not stored on the second server, directing the client to another server in the CDN. For instance, if the requested object is not stored on the second server (e.g., a parent, shield, or regional POP), the client is directed to another server in the CDN (e.g., a peer second server, a third-tier parent server, or an origin server) to serve the requested object.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

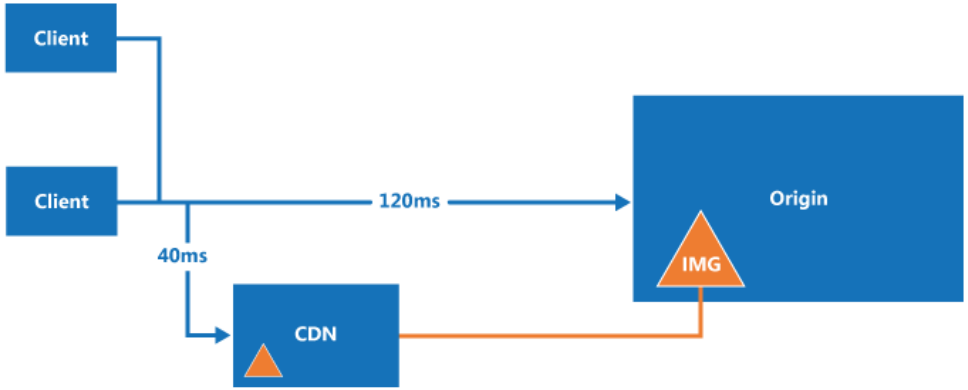
U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



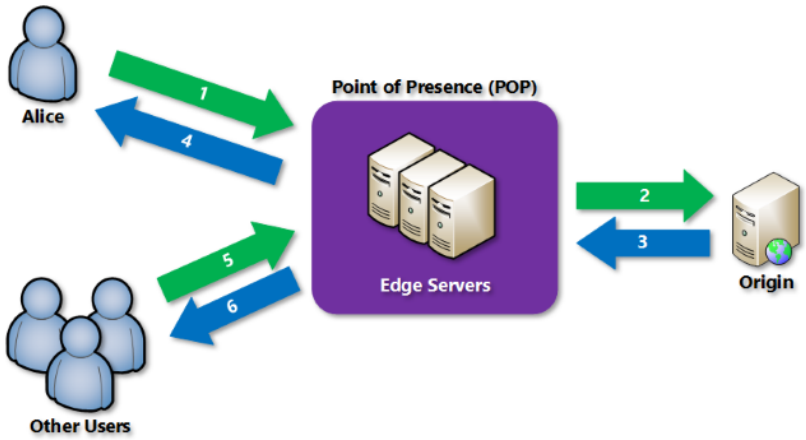
In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	SentToOriginShield (deprecated) * **See notes on deprecation in the following section.** — If true, it means that request was answered from origin shield cache instead of the edge pop. Origin shield is a parent cache used to improve cache hit ratio. isReceivedFromClient — If true, it means that the request came from the client. If false, the request is a miss in the edge (child POP) and is responded from origin shield (parent POP). https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log
2. A method as recited in claim 1, wherein said serving the requested object to the client from the second server is responsive to said second request.	Microsoft performs the method of claim 1, wherein said serving the requested object to the client from the second server is responsive to said second request. For instance, the second server (e.g., a parent or regional POP) serves the requested object to the client responsive to the second request from the client.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.
2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.
3. The origin server returns the file to an edge server in the POP.
4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.
5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.
6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

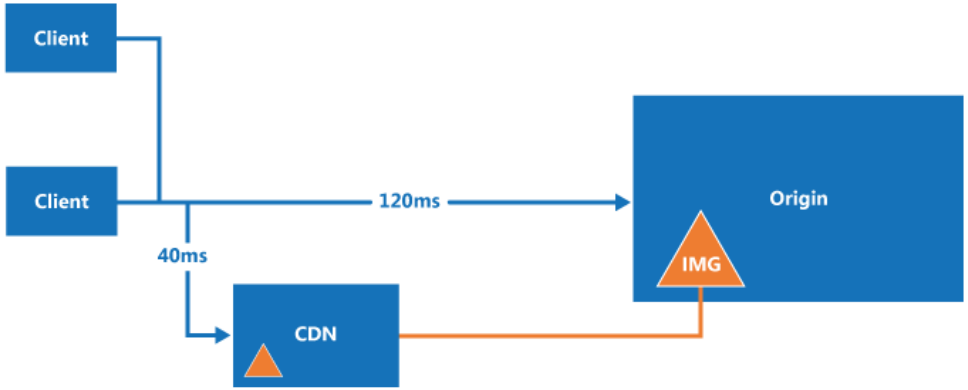
U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

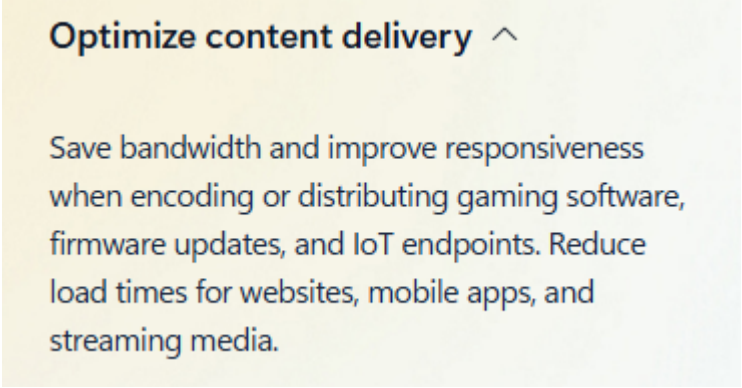
A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

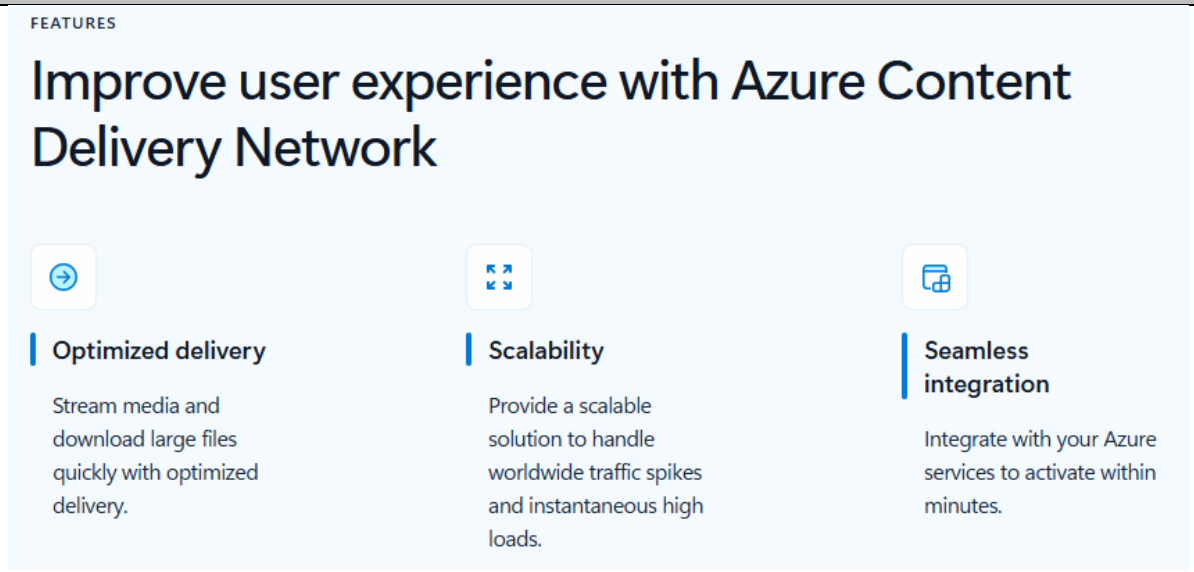
CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

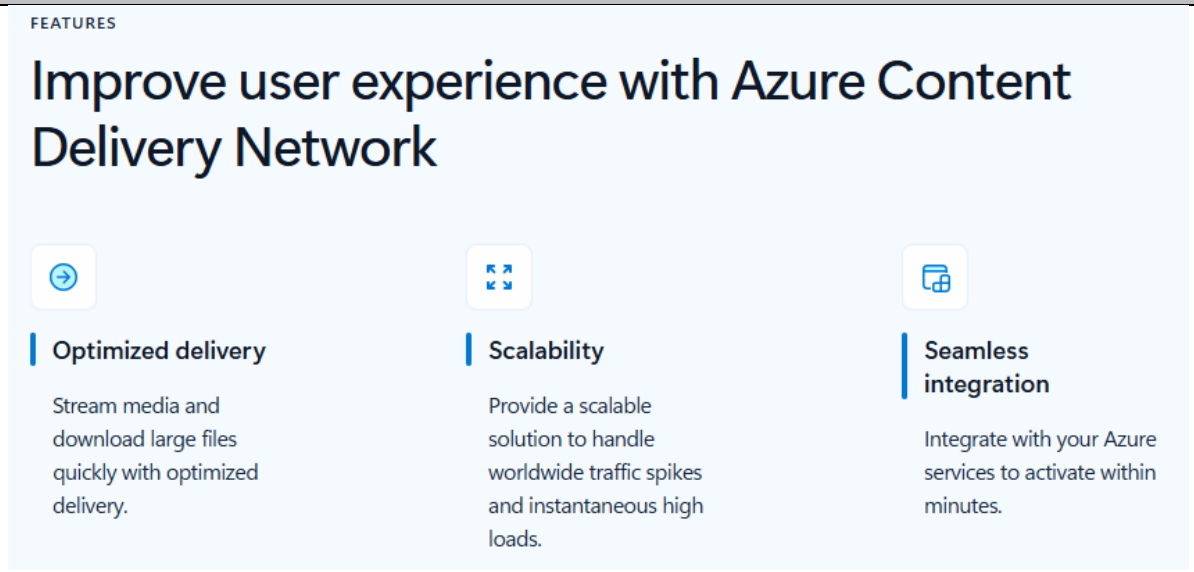
https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	SentToOriginShield (deprecated) * **See notes on deprecation in the following section.** — If true, it means that request was answered from origin shield cache instead of the edge pop. Origin shield is a parent cache used to improve cache hit ratio. isReceivedFromClient — If true, it means that the request came from the client. If false, the request is a miss in the edge (child POP) and is responded from origin shield (parent POP). https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log
3. A method as recited in claim 1, wherein the object is a streaming object.	Microsoft performs the method of claim 1, wherein the object is a streaming object. For instance, Microsoft Azure CDN and/or Azure Front Door support delivery of streaming objects (e.g. media streaming).  https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	 https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	Streaming media acceleration Customers are very enthusiastic about the speed and real-time characteristics of live streaming are popular with customers, but the real-time nature of streaming means that huge numbers of users access services concurrently. This use puts an enormous strain on source station and bandwidth resources. Such services also are subject to Chinese restrictions on cross-region or cross-carrier traffic, which impose significant requirements on high-quality, high-speed live streams. The Azure Content Delivery Network streaming media acceleration service acquires the source station video stream in real time. It then delivers the video stream to the Content Delivery Network edge node closest to the user. It uses intelligent caching and scheduling strategies to calculate and provide the optimal node for the user. This process reduces the lag and bandwidth pressures caused by link transmissions. The user is charged for data usage. The service provides users with a high-speed, smooth, high-quality direct broadcast viewing experience. See Streaming Media Acceleration for details. https://docs.azure.cn/en-us/cdn/cdn-overview
4. A method as recited in claim 1, wherein said first edge server was selected based at least in part on server load.	Microsoft performs the method of claim 1, wherein said first edge server was selected based at least in part on server load. For instance, Microsoft Azure CDN supports scalability features that select edge servers based at least in part on server load such as by using load-balancing or other redirect policies.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities



https://azure.microsoft.com/en-us/products/cdn

Microsoft Azure also features load balancing services for CDNs including, e.g., Azure Front Door, Azure Load Balancer, and Azure Traffic Manager.

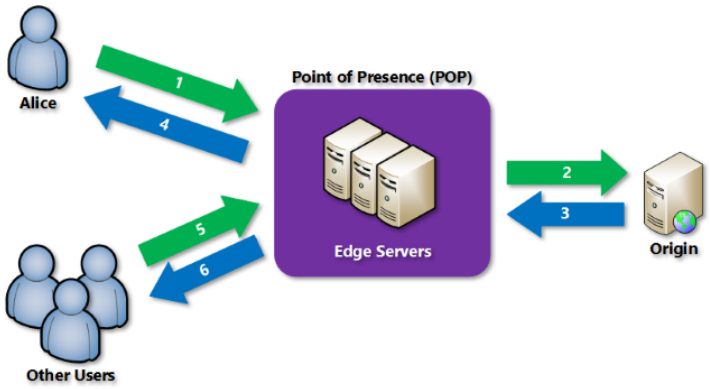
Load balancing options

07/31/2025

The term *load balancing* refers to the distribution of processing across multiple computing resources. You load balance to optimize resource usage, maximize throughput, minimize response time, and avoid overloading any single resource. Load balancing can also improve availability by sharing a workload across redundant computing resources.

Azure provides various load balancing services that you can use to distribute your workloads across multiple computing resources. These services include Azure API Management, Azure Application Gateway, Azure Front Door, Azure Load Balancer, and Azure Traffic Manager.

https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/load-balancing-overview

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
5. A method as recited in claim 1, further comprising: attempting to serve the requested object to the client from the second server in the CDN.	Microsoft performs the method of claim 1, further comprising: attempting to serve the requested object to the client from the second server in the CDN. For instance, the second server (e.g., a parent or regional POP) attempts to serve the requested object to the client. How it works  1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user. 2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server. 3. The origin server returns the file to an edge server in the POP. 4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days. 5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP. 6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

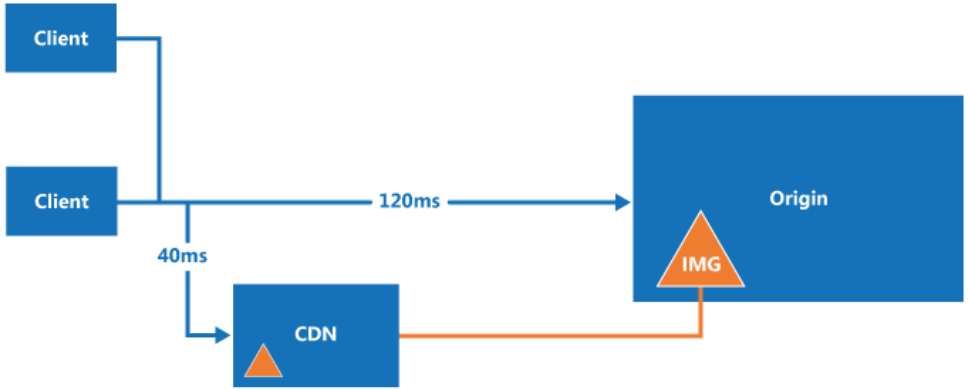
U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	SentToOriginShield (deprecated) * **See notes on deprecation in the following section.** — If true, it means that request was answered from origin shield cache instead of the edge pop. Origin shield is a parent cache used to improve cache hit ratio. isReceivedFromClient — If true, it means that the request came from the client. If false, the request is a miss in the edge (child POP) and is responded from origin shield (parent POP). https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log
7[Pre]. A method, in a framework in which resources maintained on one or more origin servers of content providers are delivered to end user clients via a shared content delivery network (CDN), the method comprising:	Microsoft performs a method, in a framework in which resources maintained on one or more origin servers of content providers are delivered to end user clients via a shared content delivery network (CDN) as set forth below. See Claim limitation 1.Pre.
[7.a] providing a plurality of server sites in the CDN, wherein each of the plurality of server sites comprises one or more servers distinct from the one or more origin servers, the plurality of server sites comprising a plurality of edge server sites and a plurality of parent server sites;	Microsoft provides a plurality of server sites in the CDN, wherein each of the plurality of server sites comprises one or more servers distinct from the one or more origin servers, the plurality of server sites comprising a plurality of edge server sites and a plurality of parent server sites. See Claim limitation 1.a.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
[7.b] causing a client request for a resource to be directed to a first edge server site in said CDN, without consideration as to whether the first edge server site has a copy of the requested resource, wherein said first edge server site was selected based at least in part on network traffic conditions;	Microsoft causes a client request for a resource to be directed to a first edge server site in said CDN, without consideration as to whether the first edge server site has a copy of the requested resource, wherein said first edge server site was selected based at least in part on network traffic conditions. See Claim limitation 1.b.
[7.c] if the first edge server site has a copy of the requested resource, then serving the requested resource to the client from the first edge server site; otherwise,	If the first edge server site has a copy of the requested resource, then serving the requested resource to the client from the first edge server site. See Claim limitation 1.c.
[7.d] if the first edge server site does not have a copy of the requested resource, sending the client a response that causes the client to be directed to a second server site in the CDN, wherein the second server site comprises a parent server site of the first edge server site; and	If the first edge server site does not have a copy of the requested resource, sending the client a response that causes the client to be directed to a second server site in the CDN, wherein the second server site comprises a parent server site of the first edge server site. See Claim limitation 1.d.
[7.e] receiving, at the second server site, a second request for the resource from the client and serving the requested resource to	Microsoft receives, at the second server site, a second request for the resource from the client and serving the requested resource to the client from the second server site, wherein if a copy of the requested resource is not stored on the second server site, directing the client to another server site in the CDN. See Claim limitation 1.e.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
the client from the second server site, wherein if a copy of the requested resource is not stored on the second server site, directing the client to another server site in the CDN.	
8. A method as recited in claim 7, further comprising: attempting to serve the requested resource to the client from the second server site in the CDN.	Microsoft performs the method of claim 7, further comprising: attempting to serve the requested resource to the client from the second server site in the CDN. See Claim 5.
9. A method as recited in claim 7, wherein a copy of the requested resource is stored at the second server site prior to the issuance of the resource request by the client.	Microsoft performs the method of claim 7, wherein a copy of the requested resource is stored at the second server site prior to the issuance of the resource request by the client. For instance, a copy of the requested resource may be stored (e.g., cached) at the second server site (e.g., a parent or regional POP) prior to the issuance of the resource request by the client.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities

Introduction to caching

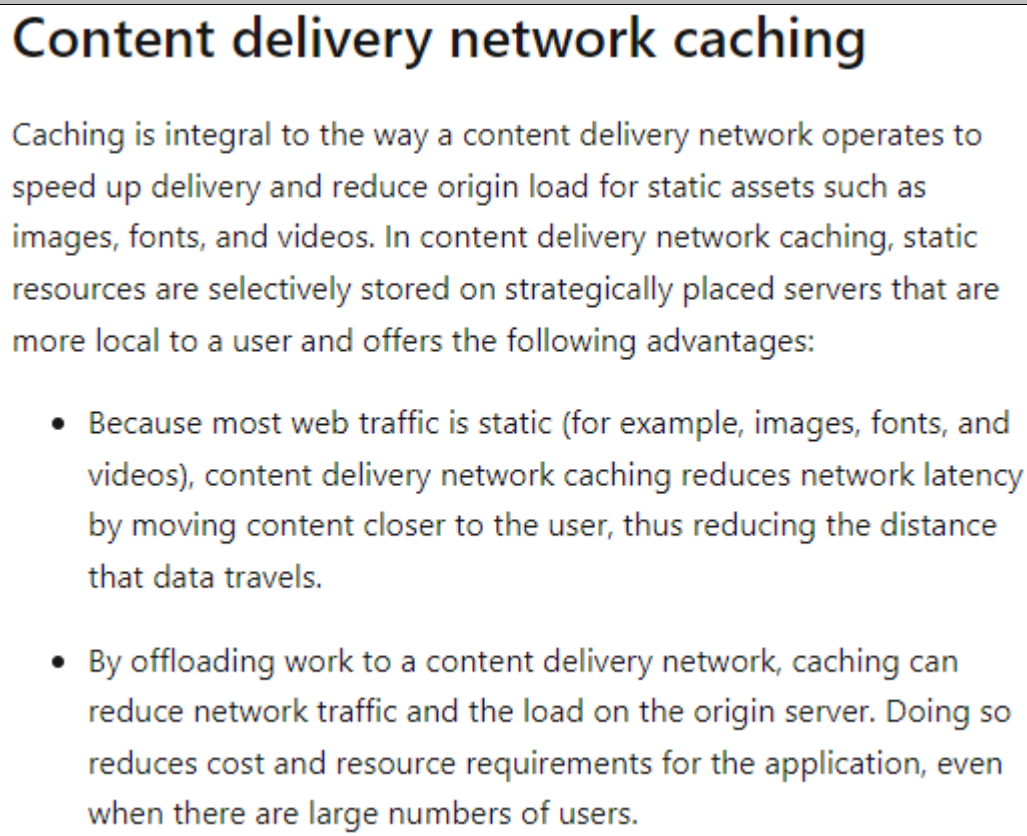
Caching is the process of storing data locally so that future requests for that data can be accessed more quickly. In the most common type of caching, web browser caching, a web browser stores copies of static data locally on a local hard drive. By using caching, the web browser can avoid making multiple round-trips to the server and instead access the same data locally, thus saving time and resources. Caching is well-suited for locally managing small, static data such as static images, CSS files, and JavaScript files.

Similarly, caching is used by a content delivery network on edge servers close to the user to avoid requests traveling back to the origin and reducing end-user latency. Unlike a web browser cache, which is used only for a single user, the content delivery network has a shared cache. In a content delivery network shared cache, a file request by a user can be used by another user, which greatly decreases the number of requests to the origin server.

Dynamic resources that change frequently or are unique to an individual user can't be cached. Those types of resources, however, can take advantage of dynamic site acceleration (DSA) optimization on the Azure content delivery network for performance improvements.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

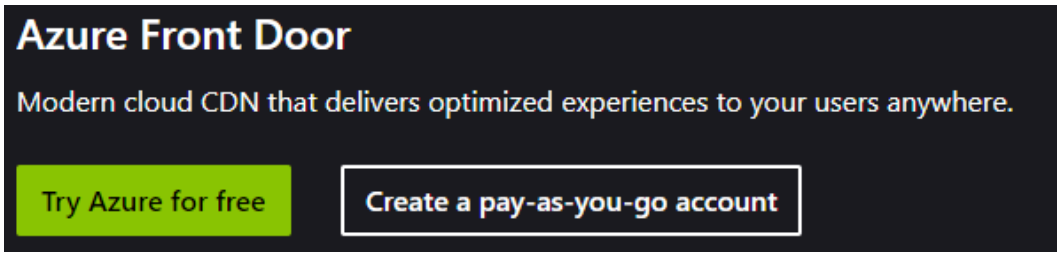
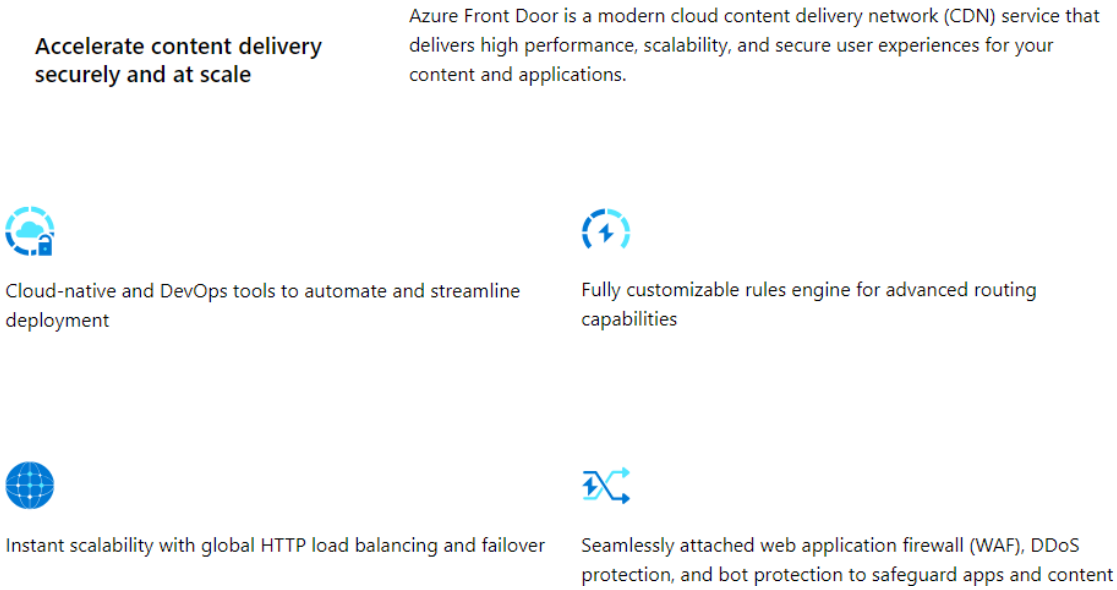
U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	Caching can occur at multiple levels between the origin server and the end user: • Web server: Uses a shared cache (for multiple users). • Content delivery network: Uses a shared cache (for multiple users). • Internet service provider (ISP): Uses a shared cache (for multiple users). • Web browser: Uses a private cache (for one user). Each cache typically manages its own resource freshness and performs validation when a file is stale. This behavior is defined in the HTTP caching specification, RFC 7234. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
	 https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works
10[Pre]. A method, in a framework in which resources maintained on one or more origin servers of at least one content provider are delivered to end user clients via a content delivery network	Microsoft performs a method, in a framework in which resources maintained on one or more origin servers of at least one content provider are delivered to end user clients via a content delivery network (CDN) as set forth below. See Claim limitation 1.Pre.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
(CDN), the method comprising:	
[10.a] providing a plurality of server sites in CDN, wherein each of the plurality of server sites comprises one or more servers distinct from the one or more origin servers, the plurality of server sites comprising a plurality of edge server sites and a plurality of parent server sites;	Microsoft provides a plurality of server sites in CDN, wherein each of the plurality of server sites comprises one or more servers distinct from the one or more origin servers, the plurality of server sites comprising a plurality of edge server sites and a plurality of parent server sites. See Claim limitation 1.a.
[10.b] receiving a client request for a resource at a first edge server site in said CDN, wherein said first edge server site was selected based at least in part on network traffic conditions;	Microsoft receives a client request for a resource at a first edge server site in said CDN, wherein said first edge server site was selected based at least in part on network traffic conditions. See Claim limitation 1.b.
[10.c] if the first edge server site has a copy of the requested resource, then serving the requested resource to the client from the first edge server site; otherwise,	If the first edge server site has a copy of the requested resource, then Microsoft serves the requested resource to the client from the first edge server site. See Claim limitation 1.c.
[10.d] if the first edge server site does not have a copy of the requested resource, then sending the client a response that causes the client to retrieve	If the first edge server site does not have a copy of the requested resource, then Microsoft sends the client a response that causes the client to retrieve the requested resource from a second server site in the CDN, wherein the second server site comprises a parent server site of the first edge server site. See Claim limitation 1.d.

U.S. Patent No. 8,924,466	Microsoft Accused Functionalities
the requested resource from a second server site in the CDN, wherein the second server site comprises a parent server site of the first edge server site; and	
[10.e] receiving, at the second server site, a second request for the resource from the client and serving the requested resource to the client from the second server site, wherein if a copy of the requested resource is not stored on the second server site, directing the client to another server site in the CDN.	Microsoft receives, at the second server site, a second request for the resource from the client and serving the requested resource to the client from the second server site, wherein if a copy of the requested resource is not stored on the second server site, directing the client to another server site in the CDN. See Claim limitation 1.e.
11. A method as recited in claim 10, a copy of the requested resource is stored at the second server site prior to receipt of the client request by the first server site.	Microsoft performs the method of claim 10, wherein a copy of the requested resource is stored at the second server site prior to receipt of the client request by the first server site. See Claim 9.

EXHIBIT C

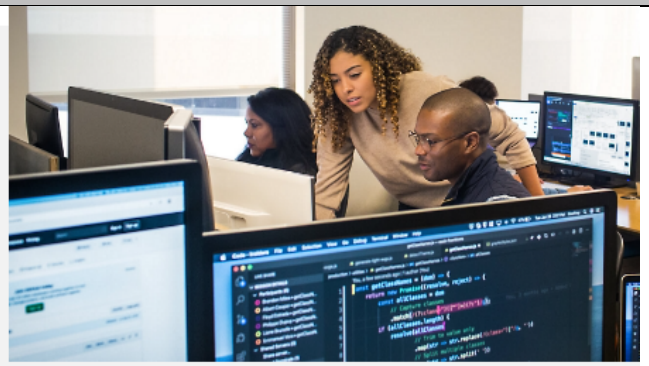
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
27[Pre]. A computer-implemented method, operable in a content delivery network (CDN) comprising: (a) a control core; and (b) a plurality of cache servers, the CDN serving resources on behalf of one or more customers of the CDN, the method comprising, by hardware and software on a particular cache server of said plurality of cache servers:	Microsoft performs a computer-implemented method operable in a content delivery network (CDN) comprising a control core and a plurality of cache servers, the CDN serving resources on behalf of one or more customers of the CDN as set forth below. For instance, Azure CDN and/or Azure Front Door include a control core and a plurality of cache servers with hardware and software components serving resources on behalf of customers of the CDN in the manner of claim 27.  https://azure.microsoft.com/en-us/products/frontdoor/  https://azure.microsoft.com/en-us/products/frontdoor/

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Build apps from any origin on modern architecture Easily deploy apps on a modern network architecture to build dynamic, high-quality digital experiences. Simplify cloud-native development with a fully programmable and API-driven service. Azure Front Door is deeply integrated with a wide array of Azure services including Azure DNS and Azure Web Apps. https://azure.microsoft.com/en-us/products/frontdoor/ **Accelerate dynamic and static content delivery** Deliver low-latency, high-throughput content at scale from the cloud or on-premises infrastructure to users anywhere. Accelerate static and dynamic content delivery with a unified platform built on the massively scalable Microsoft private global network. https://azure.microsoft.com/en-us/products/frontdoor/

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Protect your apps with seamlessly attached security

Defend your digital estate against threats with advanced security services including web application firewall (WAF), bot protection, and DDoS protection. Privately access your app origin and embrace a Zero Trust model with support for Azure Private Link. Monitor and troubleshoot traffic and security with real-time analytics.



https://azure.microsoft.com/en-us/products/frontdoor/

What is a secure, modern cloud CDN?

A secure, modern cloud CDN provides a distributed platform of servers. This helps minimize latency when users are accessing webpages. Historically, IT staff might have used a CDN and a web-application firewall to control HTTP and HTTPS traffic flowing to and from target applications.

If an organization uses Azure, they might achieve these goals by implementing the products described in the following table:

Product	Description
Azure Front Door	Enables an entry point to your apps positioned in the Microsoft global edge network. Provides faster, more secure, and scalable access to your web applications.
Azure Content Delivery Network	Delivers high-bandwidth content to your users by caching their content at strategically placed physical nodes around the world.
Azure Web Application Firewall	Helps provide centralized, greater protection for web applications from common exploits and vulnerabilities.

https://learn.microsoft.com/en-us/training/modules/intro-to-azure-front-door/2-what-is-azure-front-door

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Azure Front Door definition

Azure Front Door Standard/Premium provides the capabilities of these three products. It offers a fast, reliable, and more secure modern cloud CDN by using the Microsoft global edge network to integrate with intelligent threat protection. Azure Front Door resides in the edge locations and manages user requests to your hosted applications. Users connect to your application through the Microsoft global network. Azure Front Door then routes user requests to the fastest and most available application backend.

> ⓘ Note
>
> An *application backend* is any internet-facing service that you host, either inside or outside Azure.

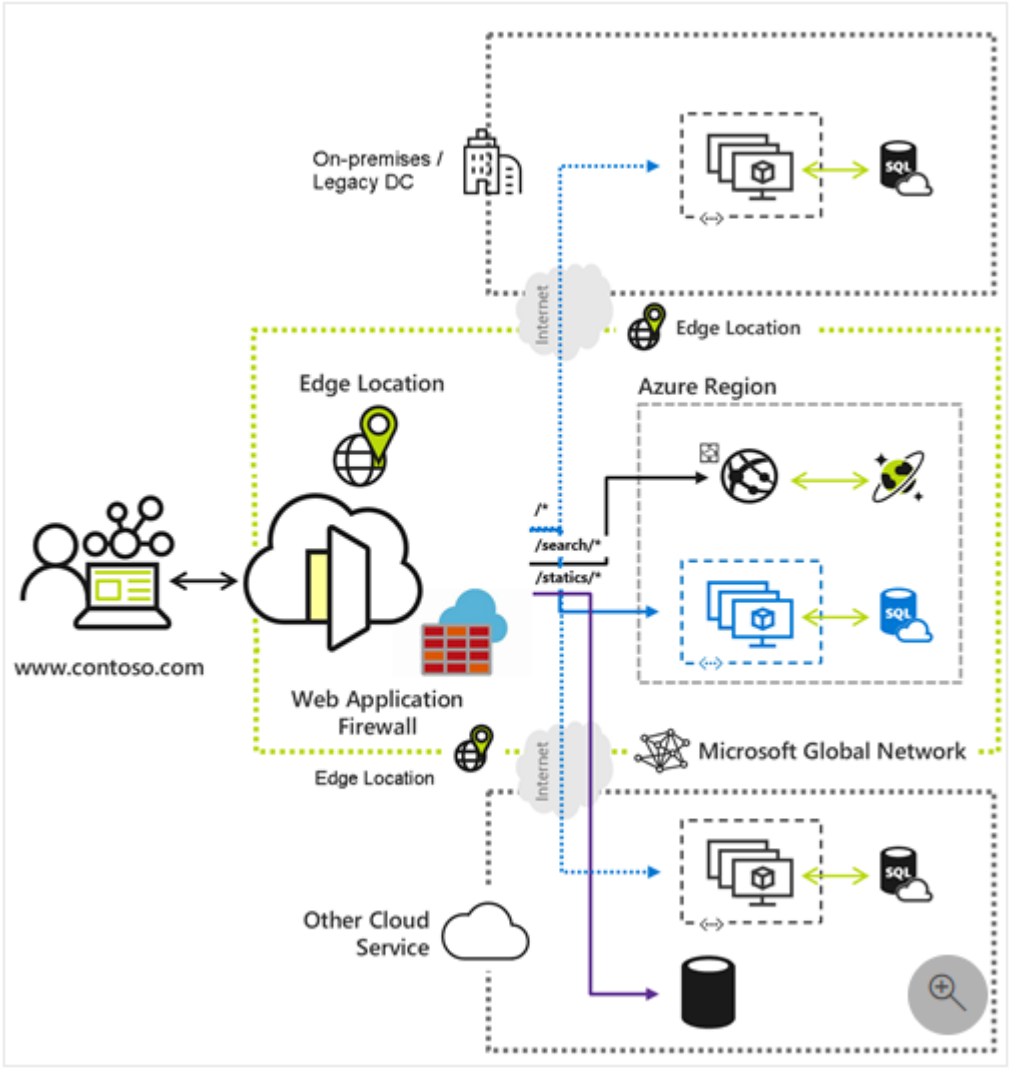
The following Azure Front door tiers are available:

- **Azure Front Door (classic)**, which is the entry level. Existing Azure customers often bolster these features with Azure Content Delivery Network, and Azure Web Application Firewall.
- **Azure Front Door Standard**, which is optimized for seamless content delivery.
- **Azure Front Door Premium**, which is optimized for improved security.

https://learn.microsoft.com/en-us/training/modules/intro-to-azure-front-door/2-what-is-azure-front-door

> Azure Front Door is Microsoft's modern cloud Content Delivery Network (CDN) that provides fast, reliable, and secure access between your users and your applications' static and dynamic web content across the globe. Azure Front Door delivers your content using Microsoft's global edge network with hundreds of global and local points of presence (PoPs) distributed around the world close to both your enterprise and consumer end users.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	 https://learn.microsoft.com/en-us/azure/frontdoor/front-door-overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Deliver modern apps and architectures Modernize your internet first applications on Azure with Cloud Native experiences • Integrate with DevOps friendly command line tools across SDKs of different languages, Bicep, ARM templates, CLI and PowerShell. • Define your own custom domain with flexible domain validation. • Load balance and route traffic across origins and use intelligent health probe monitoring across apps or content hosted in Azure or anywhere. • Integrate with other Azure services such as DNS, Web Apps, Storage and many more for domain and origin management. • Move your routing business logic to the edge with enhanced rules engine capabilities including regular expressions and server variables. • Analyze built-in reports with an all-in-one dashboard for both Front Door and security patterns. • Monitoring your Front Door traffic in real time, and configure alerts that integrate with Azure Monitor. • Log each Front Door request and failed health probes. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Microsoft also performs the methods of the asserted claims of the '053 Patent to serve Microsoft resources and services using the same or similar infrastructure. For example, Microsoft uses internal CDN technologies such as its Azure Front Door to support at least Xbox gaming, Office 365, Windows updates, and web resources like Linkedin and Bing. • Built on Microsoft's massive-scale private global network, Azure Front Door is a proven platform used to power some of the largest and latency sensitive global services at Microsoft, such as Microsoft Office 365, Bing, LinkedIn, and Xbox. https://azure.microsoft.com/en-us/blog/azure-front-door-enhances-secure-cloud-cdn-with-intelligent-threat-protection/ Azure CDN provides a world class platform to let you reduce load times, save bandwidth, and speed responsiveness across your businesses diverse workflows. Azure CDN from Microsoft enables Azure customers to use and deliver content from the same global CDN network leveraged by Microsoft properties such as Office 365, Skype, Bing, OneDrive, Windows, and Xbox. Connectivity within Microsoft's network along with new Regional Caching capabilities enables more consistent, more predictable cache fill performance by providing multi-tier caching along direct, private access to content in Azure from each CDN edge point of presence (POP). Azure CDN from Microsoft entered public preview providing access to 54 global Edge POPs in 33 countries and 16 Regional Cache POPs. https://azure.microsoft.com/en-us/blog/announcing-microsoft-s-own-cdn-network/

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

What powers Windows 365?

Not surprising, Microsoft's own cloud computing platform, Azure. Specifically, **Azure Virtual Desktop**, which can be used to create Windows virtual machines on Redmond's servers, then stream those desktops to PC and non-PC devices.

https://www.computerworld.com/article/1614540/what-is-windows-365-microsoft-s-cloud-based-os-explained.html

Use the Office 365 Content Delivery Network (CDN) with SharePoint Online

02/28/2025

You can use the built-in Office 365 Content Delivery Network (CDN) to host static assets to provide better performance for your SharePoint Online pages. The Office 365 CDN improves performance by caching static assets closer to the browsers requesting them, which helps to speed up downloads and reduce latency. Also, the Office 365 CDN uses the HTTP/2 protocol for improved compression and HTTP pipelining. The Office 365 CDN service is included as part of your SharePoint Online subscription.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	The Office 365 CDN is composed of multiple CDNs that allow you to host static assets in multiple locations, or *origins*, and serve them from global high-speed networks. Depending on the kind of content you want to host in the Office 365 CDN such as you can add **public** origins. If you're already familiar with the way that CDNs work, you only need to complete a few steps to enable the Office 365 CDN for your tenant. This article describes how. Read on for information about how to get started hosting your static assets. https://learn.microsoft.com/en-us/microsoft-365/enterprise/use-microsoft-365-cdn-with-spo?view=o365-worldwide On the Microsoft side, the company uses primarily Azure Front Door a cloud CDN platform and some third-party CDNs for some static delivery of content i.e Xbox downloads, Windows Updates. Microsoft does not https://www.streamingmediablog.com/2020/10/gaming-console-launch.html

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Teams is built on a microservices architecture, with a few hundred microservices working cohesively to deliver our product's many features including messaging, meetings, files, calendar, and apps. Using microservices helps each of our component teams to work and release their changes independently.

Azure is the cloud platform that underpins all of Microsoft's cloud services, including Microsoft Teams. Our workloads run in Azure virtual machines (VMs), with our older services being deployed through Azure Cloud Services and our newer ones on Azure Service Fabric. Our primary storage stack is Azure Cosmos DB, with some services using Azure Blob Storage. We count on Azure Cache for Redis for increased throughput and resiliency. We leverage Traffic Manager and Azure Front Door to route traffic where we want it to be. We use Queue Storage and Event Hubs to communicate, and we depend on Azure Active Directory to manage our tenants and users.

- **Resiliency-optimized caches**: We leverage caches between our components extensively, for both performance and resiliency. Caches help reduce average latency and provide a source of data in case a downstream service is unavailable. Keeping data in caches for a long time introduces data freshness issues yet keeping data in caches for a long time is the best defense against downstream failures. We focus on Time to Refresh (TTR) to our cache data as well as Time to Live (TTL). By setting a long TTL and a shorter TTR value, we can fine-tune how fresh to keep our data versus how long we want data to stick around whenever a downstream dependency fails.

https://azure.microsoft.com/en-us/blog/advancing-microsoft-teams-on-azure-operating-at-pandemic-scale/

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Background

A CDN is a system of distributed servers that delivers web content to users from geographical locations closest to them. At LinkedIn, CDNs deliver static content, such as CSS and JavaScript on the site, as well as member-generated content, such as profile pictures, media shares, and video uploads, to our members around the world. The content is delivered from edges that are closest to the user in question.

The Edge SRE team runs LinkedIn's edge infrastructure, including four external CDNs, an in-house CDN, three Domain Name System (DNS) platforms, and all LinkedIn points of presence (PoPs). We build and manage tools to automate all aspects of our Edge stack.

https://www.linkedin.com/blog/engineering/data-management/enabling-dual-stack-on-linkedin-cdns\

LinkedIn recently described how it uses the edge to support the popular professional networking web site. The engineering team wrote a blog post to talk about edge deployments and its migration away from an in-house solution to Azure Edge.

LinkedIn is one of many large web properties that is owned by a hyperscaler cloud provider — Microsoft, in this case — but doesn't actually use the in-house cloud. YouTube is another example of this — it has yet to be migrated to GCP.

LinkedIn actually had its own quite extensive in-house CDN, with nineteen PoPs (points of presence) in ten countries. This is a common practice for major content owners as it defrays transport costs and improves performance. The nodes can cache content and perform tasks such as SSL offloading. But as LinkedIn noted, Microsoft's CDN – Azure Front Door – gives access to 300 PoPs around the globe, which is a big step up in terms of reach.

https://www.edgeir.com/how-linkedin-is-using-azure-front-door-for-edge-build-out-20220306

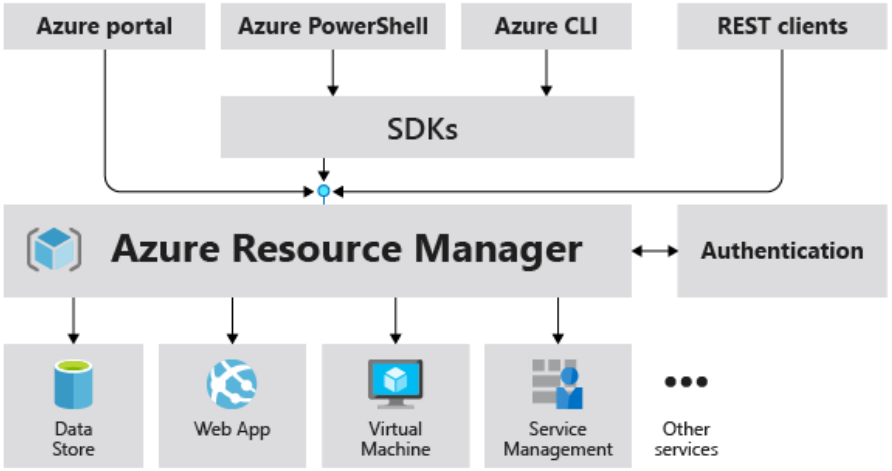
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
[27.a] (A) obtaining global configuration data from the control core;	Microsoft obtains global configuration data from the control core. For instance, Azure CDN and/or Azure Front Door cache servers obtain global configuration data (e.g., routing policies, caching rules, content updates, and security settings) from a control core such as Azure Resource Manager.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Consistent management layer

When you send a request through any of the Azure APIs, tools, or SDKs, Resource Manager receives the request. It authenticates and authorizes the request before forwarding it to the appropriate Azure service. Because all requests are handled through the same API, you see consistent results and capabilities in all the different tools.

The following image shows the role Azure Resource Manager plays in handling Azure requests.



All capabilities that are available in the portal are also available through PowerShell, Azure CLI, REST APIs, and client SDKs. Functionality initially released through APIs are represented in the portal within 180 days of initial release.

https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Terminology

If you're new to Azure Resource Manager, there are some terms you might not be familiar with.

- **resource** - A manageable item that is available through Azure. Virtual machines, storage accounts, web apps, databases, and virtual networks are examples of resources. Resource groups, subscriptions, management groups, and tags are also examples of resources.
- **resource group** - A container that holds related resources for an Azure solution. The resource group includes those resources that you want to manage as a group. You decide which resources belong in a resource group based on what makes the most sense for your organization. See What is a resource group?.
- **resource provider** - A service that supplies Azure resources. For example, a common resource provider is `Microsoft.Compute`, which supplies the virtual machine resource. `Microsoft.Storage` is another common resource provider. See Resource providers and types.
- **declarative syntax** - Syntax that lets you state "Here's what I intend to create" without having to write the sequence of programming commands to create it. ARM templates and Bicep files are examples of declarative syntax. In those files, you define the properties for the infrastructure to deploy to Azure.
- **ARM template** - A JavaScript Object Notation (JSON) file that defines one or more resources to deploy to a resource group, subscription, management group, or tenant. The template can be used to deploy the resources consistently and repeatedly. See Template deployment overview.
- **Bicep file** - A file for declaratively deploying Azure resources. Bicep is a language that was designed to provide the best authoring experience for infrastructure as code solutions in Azure. See Bicep overview.
- **extension resource** - A resource that adds to another resource's capabilities. For example, a role assignment is an extension resource. You apply a role assignment to any other resource to specify access. See Extension resources.

https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

The benefits of using Resource Manager

With Resource Manager, you can:

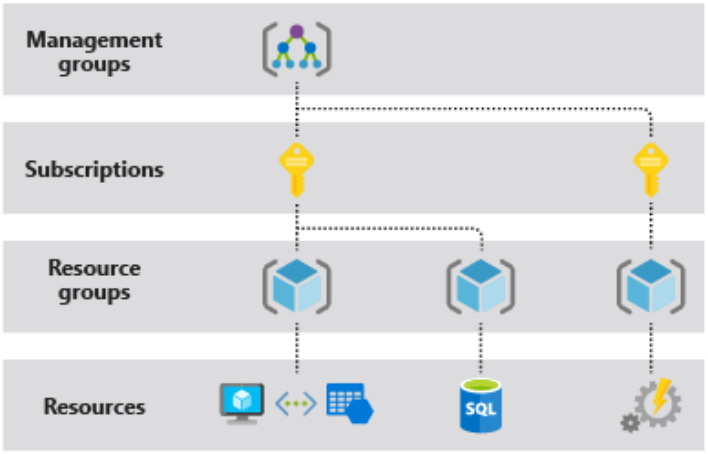
- Manage your infrastructure through declarative templates rather than scripts.
- Deploy, manage, and monitor all the resources for your solution as a group, rather than handling these resources individually.
- Redeploy your solution throughout the development lifecycle and have confidence your resources are deployed in a consistent state.
- Define the dependencies between resources so they're deployed in the correct order.
- Apply access control to all services because Azure role-based access control (Azure RBAC) is natively integrated into the management platform.
- Apply tags to resources to logically organize all the resources in your subscription.
- Clarify your organization's billing by viewing costs for a group of resources sharing the same tag.

https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

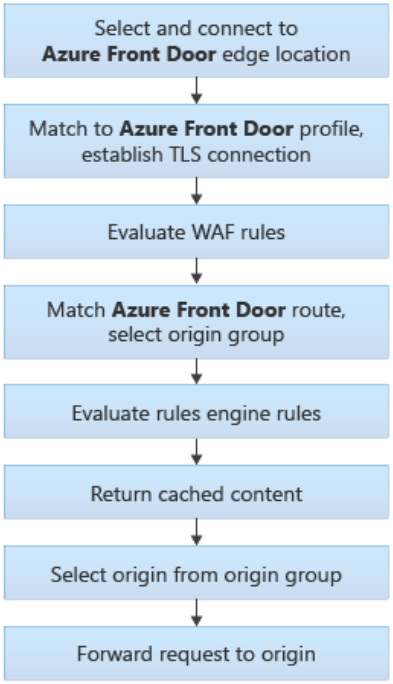
Understand scope

Azure provides four levels of management scope: management groups, subscriptions, resource groups, and resources. The following image shows an example of these layers.



You apply management settings at any of these levels of scope. The level you select determines how widely the setting is applied. Lower levels inherit settings from higher levels. For example, when you apply a policy to the subscription, the policy is applied to all resource groups and resources in your subscription. When you apply a policy on the resource group, that policy is applied to the resource group and all its resources. However, another resource group doesn't have that policy assignment.

https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Azure Front Door traffic routing takes place over multiple stages. First, traffic is routed from the client to the Front Door. Then, Front Door uses your configuration to determine the origin to send the traffic to. The Front Door web application firewall, routing rules, rules engine, and caching configuration can all affect the routing process. The following diagram illustrates the routing architecture:  https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Select and connect to the Front Door edge location The user or client application initiates a connection to the Front Door. The connection terminates at an edge location closest to the end user. Front Door's edge location processes the request. For more information about how requests are made to Front Door, see Front Door traffic acceleration. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture
[27.b] (B) selectively obtaining updated global configuration data; and	Microsoft selectively obtains updated global configuration data. For instance, Azure CDN and/or Azure Front Door cache servers selectively obtain updated global configuration data. As an example, an edge node may obtain updated global routing policies or caching rules as needed, such as from Azure Resource Manager. See Claim limitation 27.a. ⓘ Note Even when a region provides availability zones, it's possible that some services might not support them in that region. Refer to the Azure service reliability guides to learn about region support for availability zone-enabled services.

<table>
<tr><th>U.S. Patent No. 9,456,053</th><th>Microsoft Accused Functionalities</th></tr>
<tr><td></td><td>https://learn.microsoft.com/en-us/azure/reliability/regions-list?source=recommendations

Azure Government edge locations

Expand table

<table><tr><th>Region</th><th>Location</th></tr><tr><td>North America (US Gov)</td><td>Arizona, AZ, USA
Texas, TX, USA
Virginia, VA, USA</td></tr></table>
https://learn.microsoft.com/en-us/azure/frontdoor/edge-locations-by-region</td></tr>
<tr><td>[27.c] (C) obtaining customer configuration information associated with a particular customer of the CDN;</td><td>Microsoft obtains customer configuration information associated with a particular customer of the CDN. For instance, Azure CDN and/or Azure Front Door provide rule sets configurable specific to CDN customers.

A rule set is a customized rules engine that groups a combination of rules into a single set. You can associate a rule set with multiple routes. A Rule set allows you to customize how requests get processed and handled at the Azure Front Door edge.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine</td></tr>
</table>

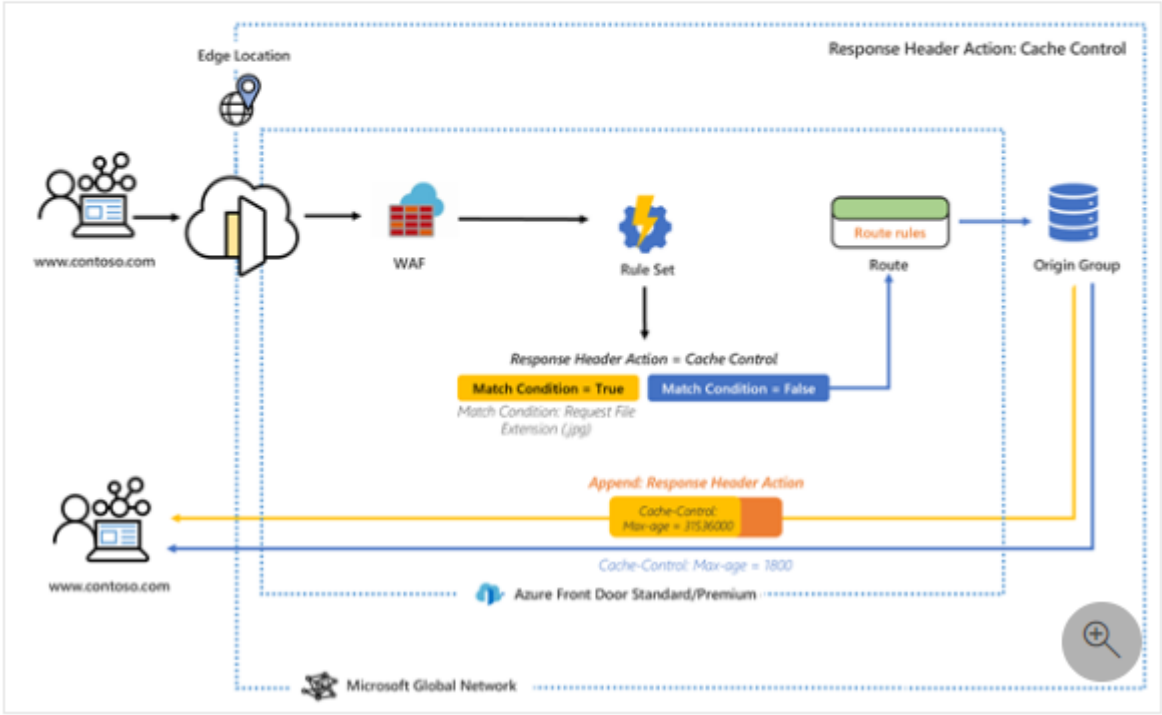
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Common supported scenarios • Implementing security headers to prevent browser-based vulnerabilities like HTTP Strict-Transport-Security (HSTS), X-XSS-Protection, Content-Security-Policy, X-Frame-Options, and Access-Control-Allow-Origin headers for Cross-Origin Resource Sharing (CORS) scenarios. Security-based attributes can also be defined with cookies. • Route requests to mobile or desktop versions of your application based on the client device type. • Using redirect capabilities to return 301, 302, 307, and 308 redirects to the client to direct them to new hostnames, paths, query strings, or protocols. • Dynamically modify the caching configuration of your route based on the incoming requests. • Rewrite the request URL path and forwards the request to the appropriate origin in your configured origin group. • Add, modify, or remove request/response header to hide sensitive information or capture important information through headers. • Support server variables to dynamically change the request header, response headers, or URL rewrite paths/query strings. For example, when a new page load or when a form gets posted. Server variable is currently supported in **rule set actions** only. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	# Architecture Rule sets handle requests at the Front Door edge. When a request arrives at your Front Door endpoint, WAF (Web Application Firewall) is processed first, followed by the settings configured in route. Those settings include the rule set associated to the route. Rule sets are processed in the order they appear under the routing configuration. Rules in a rule set also get processed in the order they appear. In order for all the actions in each rule to run, all the match conditions within a rule have to be met. If a request doesn't match any of the conditions in your rule set configuration, then only the default route settings get applied. If the **Stop evaluating remaining rules** is selected, then any remaining rule sets associated with the route don't get ran. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Example

In the following diagram, WAF policies get processed first. Then the rule set configuration appends a response header. The header changes the max-age of the cache control if the match condition is true.



https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

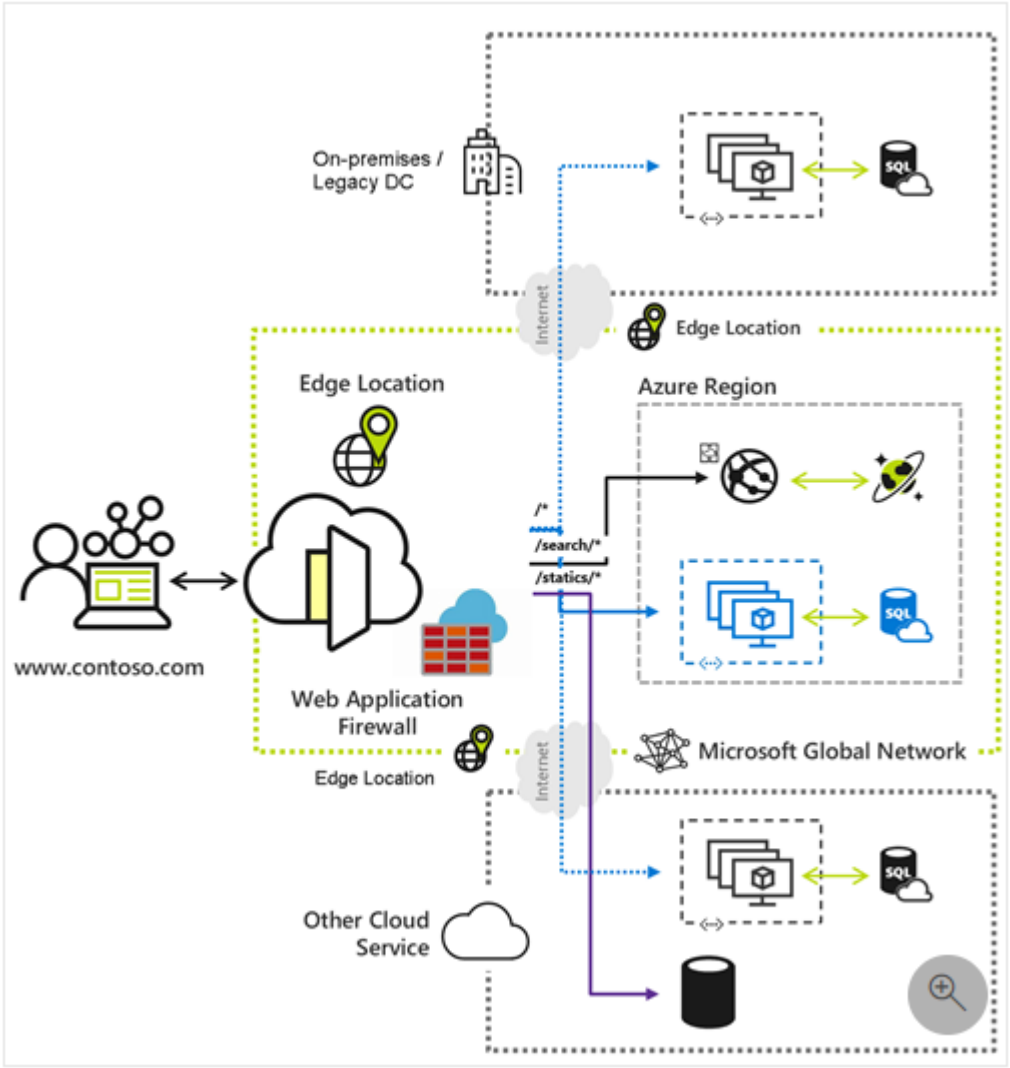
Terminology

With a Front Door rule set, you can create any combination of configurations, each composed of a set of rules. The following out lines some helpful terminologies you come across when configuring your rule set.

- *Rule set*: A set of rules that gets associated to one or multiple routes.
- *Rule set rule*: A rule composed of up to 10 match conditions and 5 actions. Rules are local to a rule set and can't be exported to use across other rule sets. You can create the same rule in different rule sets.
- *Match condition*: There are many match conditions that you can configure to parse an incoming request. A rule can contain up to 10 match conditions. Match conditions are evaluated with an **AND** operator. *Regular expression is supported in conditions*. A full list of match conditions can be found in Rule set match conditions.
- *Action*: An action dictates how Front Door handles the incoming requests based on the matching conditions. You can modify caching behaviors, modify request headers, response headers, set URL rewrite, and URL redirection. *Server variables are supported with Action*. A rule can contain up to five actions. A full list of actions can be found in Rule set actions.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
[27.d] (D) serving, to a client, a particular resource associated with the particular customer in accordance with (i) the global configuration data; and (ii) the customer configuration information associated with the particular customer.	Microsoft serves to a client a particular resource associated with the particular customer in accordance with the global configuration data and the customer configuration information associated with the particular customer. For instance, Azure CDN and/or Azure Front Door serves resources according to global configuration data (e.g., routing policies, caching rules, content updates, and security settings) and customer configuration data (e.g., rule sets) associated with the particular customer. Azure Front Door is Microsoft's modern cloud Content Delivery Network (CDN) that provides fast, reliable, and secure access between your users and your applications' static and dynamic web content across the globe. Azure Front Door delivers your content using Microsoft's global edge network with hundreds of global and local points of presence (PoPs) distributed around the world close to both your enterprise and consumer end users. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	 https://learn.microsoft.com/en-us/azure/frontdoor/front-door-overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Deliver modern apps and architectures Modernize your internet first applications on Azure with Cloud Native experiences • Integrate with DevOps friendly command line tools across SDKs of different languages, Bicep, ARM templates, CLI and PowerShell. • Define your own custom domain with flexible domain validation. • Load balance and route traffic across origins and use intelligent health probe monitoring across apps or content hosted in Azure or anywhere. • Integrate with other Azure services such as DNS, Web Apps, Storage and many more for domain and origin management. • Move your routing business logic to the edge with enhanced rules engine capabilities including regular expressions and server variables. • Analyze built-in reports with an all-in-one dashboard for both Front Door and security patterns. • Monitoring your Front Door traffic in real time, and configure alerts that integrate with Azure Monitor. • Log each Front Door request and failed health probes. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	A rule set is a customized rules engine that groups a combination of rules into a single set. You can associate a rule set with multiple routes. A Rule set allows you to customize how requests get processed and handled at the Azure Front Door edge. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

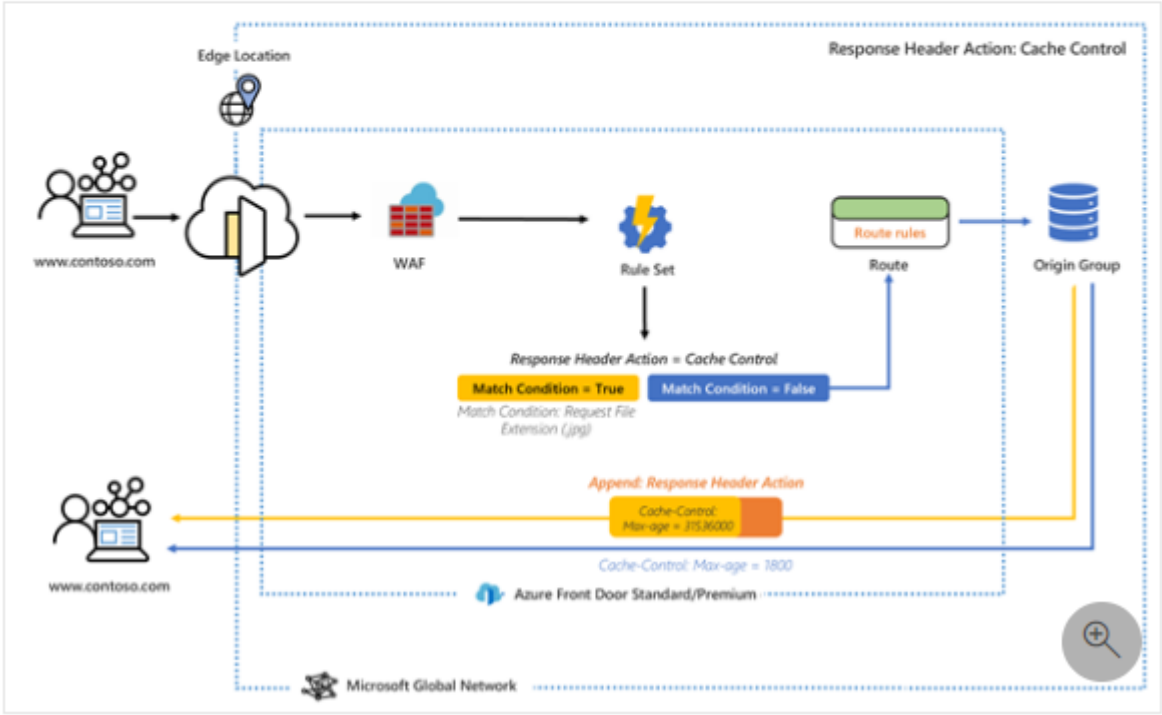
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Common supported scenarios • Implementing security headers to prevent browser-based vulnerabilities like HTTP Strict-Transport-Security (HSTS), X-XSS-Protection, Content-Security-Policy, X-Frame-Options, and Access-Control-Allow-Origin headers for Cross-Origin Resource Sharing (CORS) scenarios. Security-based attributes can also be defined with cookies. • Route requests to mobile or desktop versions of your application based on the client device type. • Using redirect capabilities to return 301, 302, 307, and 308 redirects to the client to direct them to new hostnames, paths, query strings, or protocols. • Dynamically modify the caching configuration of your route based on the incoming requests. • Rewrite the request URL path and forwards the request to the appropriate origin in your configured origin group. • Add, modify, or remove request/response header to hide sensitive information or capture important information through headers. • Support server variables to dynamically change the request header, response headers, or URL rewrite paths/query strings. For example, when a new page load or when a form gets posted. Server variable is currently supported in **rule set actions** only. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	# Architecture Rule sets handle requests at the Front Door edge. When a request arrives at your Front Door endpoint, WAF (Web Application Firewall) is processed first, followed by the settings configured in route. Those settings include the rule set associated to the route. Rule sets are processed in the order they appear under the routing configuration. Rules in a rule set also get processed in the order they appear. In order for all the actions in each rule to run, all the match conditions within a rule have to be met. If a request doesn't match any of the conditions in your rule set configuration, then only the default route settings get applied. If the **Stop evaluating remaining rules** is selected, then any remaining rule sets associated with the route don't get ran. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Example

In the following diagram, WAF policies get processed first. Then the rule set configuration appends a response header. The header changes the max-age of the cache control if the match condition is true.



https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

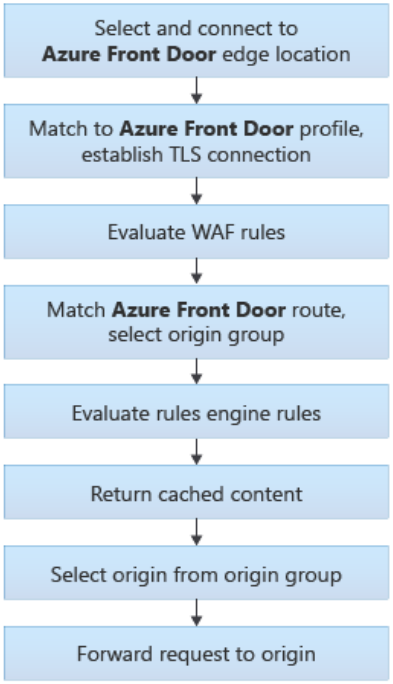
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Terminology

With a Front Door rule set, you can create any combination of configurations, each composed of a set of rules. The following out lines some helpful terminologies you come across when configuring your rule set.

- *Rule set*: A set of rules that gets associated to one or multiple routes.
- *Rule set rule*: A rule composed of up to 10 match conditions and 5 actions. Rules are local to a rule set and can't be exported to use across other rule sets. You can create the same rule in different rule sets.
- *Match condition*: There are many match conditions that you can configure to parse an incoming request. A rule can contain up to 10 match conditions. Match conditions are evaluated with an **AND** operator. *Regular expression is supported in conditions*. A full list of match conditions can be found in Rule set match conditions.
- *Action*: An action dictates how Front Door handles the incoming requests based on the matching conditions. You can modify caching behaviors, modify request headers, response headers, set URL rewrite, and URL redirection. *Server variables are supported with Action*. A rule can contain up to five actions. A full list of actions can be found in Rule set actions.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Azure Front Door traffic routing takes place over multiple stages. First, traffic is routed from the client to the Front Door. Then, Front Door uses your configuration to determine the origin to send the traffic to. The Front Door web application firewall, routing rules, rules engine, and caching configuration can all affect the routing process. The following diagram illustrates the routing architecture:  https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Select and connect to the Front Door edge location

The user or client application initiates a connection to the Front Door. The connection terminates at an edge location closest to the end user. Front Door's edge location processes the request.

For more information about how requests are made to Front Door, see Front Door traffic acceleration.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture

Match request to a Front Door profile

When Front Door receives an HTTP request, it uses the request's `Host` header to match the request to the correct customer's Front Door profile. If the request is using a custom domain name, the domain name must be registered with Front Door to enable requests to get matched to your profile.

The client and server perform a TLS handshake using the TLS certificate you configured for your custom domain name, or by using the Front Door certificate when the `Host` header ends with `*.azurefd.net`.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	## Evaluate WAF rules If your domain has Web Application Firewall enabled, WAF rules are evaluated. If a rule gets violated, Front Door returns an error to the client and the request processing stops. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture ## Match a route Front Door matches the request to a route. Learn more about the route matching process. The route specifies the origin group that the request should be sent to. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture ## Evaluate rule sets If you define rule sets for the route, they get process in the order configured. Rule sets can override the origin group specified in a route. Rule sets can also trigger a redirection response to the request instead of forwarding it to an origin. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Return cached response If the Front Door routing rule has caching enabled, and the Front Door edge location's cache includes a valid response for the request, then Front Door returns the cached response. If caching is disabled or no response is available, the request is forwarded to the origin. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture **Select origin** Front Door selects an origin to use within the origin group. Origin selection is based on several factors, including: • Health of each origin, which Front Door monitors by using health probes. • Routing method for your origin group. • If you enable session affinity https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture
28. The method of claim 27 further comprising: (A)(2) obtaining data from one or more other locations in the CDN.	Microsoft performs the method of claim 27 further comprising obtaining data from one or more other locations in the CDN. For instance, Azure CDN and/or Azure Front Door obtains data from locations in the CDN including from Azure Resource Manager. See Claim limitation 27.a.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
29. The method of claim 27 wherein the obtaining of said customer configuration information in (C) occurs upon receipt of a request of said particular cache server for a resource associated with the particular customer.	Microsoft performs the method of claim 27 wherein the obtaining of said customer configuration information in (C) occurs upon receipt of a request of said particular cache server for a resource associated with the particular customer. For instance, Azure CDN and/or Azure Front Door provide rule sets configurable specific to CDN customers, which may be processed upon receipt of a request. A rule set is a customized rules engine that groups a combination of rules into a single set. You can associate a rule set with multiple routes. A Rule set allows you to customize how requests get processed and handled at the Azure Front Door edge. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

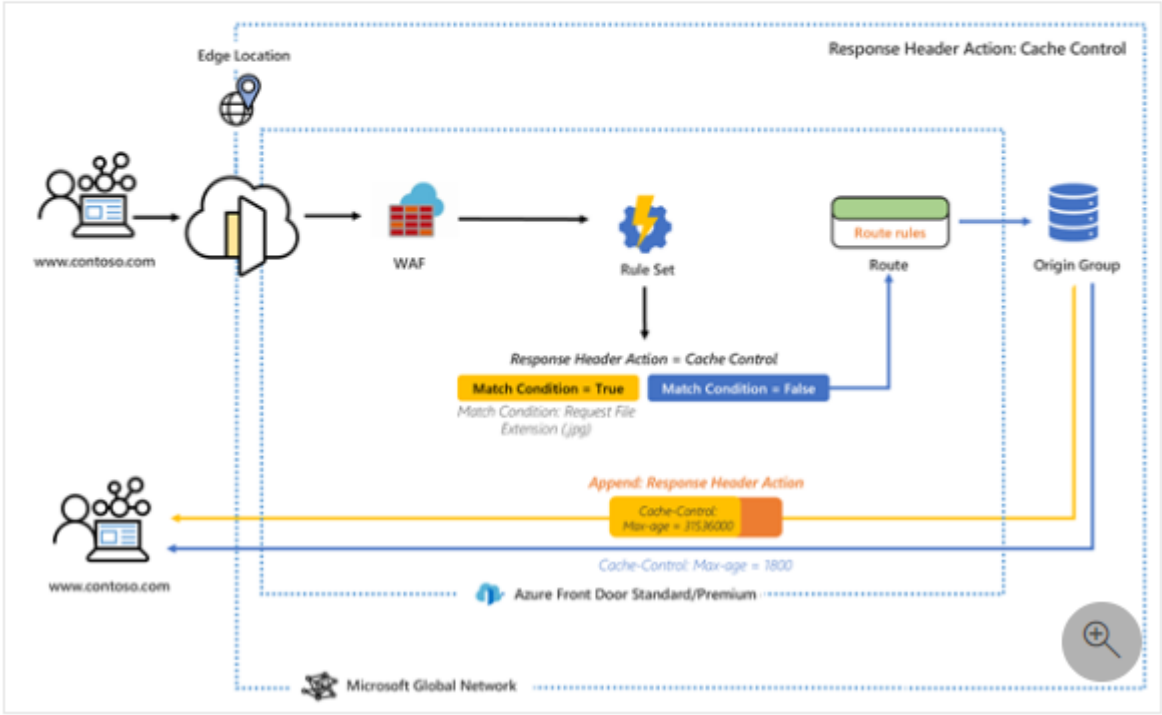
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Common supported scenarios • Implementing security headers to prevent browser-based vulnerabilities like HTTP Strict-Transport-Security (HSTS), X-XSS-Protection, Content-Security-Policy, X-Frame-Options, and Access-Control-Allow-Origin headers for Cross-Origin Resource Sharing (CORS) scenarios. Security-based attributes can also be defined with cookies. • Route requests to mobile or desktop versions of your application based on the client device type. • Using redirect capabilities to return 301, 302, 307, and 308 redirects to the client to direct them to new hostnames, paths, query strings, or protocols. • Dynamically modify the caching configuration of your route based on the incoming requests. • Rewrite the request URL path and forwards the request to the appropriate origin in your configured origin group. • Add, modify, or remove request/response header to hide sensitive information or capture important information through headers. • Support server variables to dynamically change the request header, response headers, or URL rewrite paths/query strings. For example, when a new page load or when a form gets posted. Server variable is currently supported in **rule set actions** only. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	# Architecture Rule sets handle requests at the Front Door edge. When a request arrives at your Front Door endpoint, WAF (Web Application Firewall) is processed first, followed by the settings configured in route. Those settings include the rule set associated to the route. Rule sets are processed in the order they appear under the routing configuration. Rules in a rule set also get processed in the order they appear. In order for all the actions in each rule to run, all the match conditions within a rule have to be met. If a request doesn't match any of the conditions in your rule set configuration, then only the default route settings get applied. If the **Stop evaluating remaining rules** is selected, then any remaining rule sets associated with the route don't get ran. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Example

In the following diagram, WAF policies get processed first. Then the rule set configuration appends a response header. The header changes the max-age of the cache control if the match condition is true.



https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Terminology

With a Front Door rule set, you can create any combination of configurations, each composed of a set of rules. The following out lines some helpful terminologies you come across when configuring your rule set.

- *Rule set*: A set of rules that gets associated to one or multiple routes.
- *Rule set rule*: A rule composed of up to 10 match conditions and 5 actions. Rules are local to a rule set and can't be exported to use across other rule sets. You can create the same rule in different rule sets.
- *Match condition*: There are many match conditions that you can configure to parse an incoming request. A rule can contain up to 10 match conditions. Match conditions are evaluated with an **AND** operator. *Regular expression is supported in conditions*. A full list of match conditions can be found in Rule set match conditions.
- *Action*: An action dictates how Front Door handles the incoming requests based on the matching conditions. You can modify caching behaviors, modify request headers, response headers, set URL rewrite, and URL redirection. *Server variables are supported with Action*. A rule can contain up to five actions. A full list of actions can be found in Rule set actions.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
31. The method of claim 27 wherein the customer configuration information comprises one or more scripts to be used by the particular cache server to process requests for resources associated with the particular customer.	Microsoft performs the method of claim 27 wherein the customer configuration information comprises one or more scripts to be used by the particular cache server to process requests for resources associated with the particular customer. For instance, on information and belief, Azure Front Door rule sets comprise scripts (e.g., a series of executable instructions) to process requests for resources associated with a particular customer. See also Claim limitation 27.c
33. The method of claim 27 wherein the customer configuration information comprises at least one customer configuration script (CCS) associated with the particular customer.	Microsoft performs the method of claim 27 wherein the customer configuration information comprises at least one customer configuration script (CCS) associated with the particular customer. For instance, Azure CDN and/or Azure Front Door rule sets comprise customer configuration scripts (e.g. series of executable instructions) to process requests for resources associated with a particular customer. See also Claim limitation 27.c.
34. The method of claim 27 wherein the customer configuration information associated with the particular customer specifies customer-specific processing for resources served on behalf of that customer.	Microsoft performs the method of claim 27 wherein the customer configuration information associated with the particular customer specifies customer-specific processing for resources served on behalf of that customer. For instance, Azure CDN and/or Azure Front Door provide rule sets to configure request processing specific to CDN customers. A rule set is a customized rules engine that groups a combination of rules into a single set. You can associate a rule set with multiple routes. A Rule set allows you to customize how requests get processed and handled at the Azure Front Door edge. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Common supported scenarios • Implementing security headers to prevent browser-based vulnerabilities like HTTP Strict-Transport-Security (HSTS), X-XSS-Protection, Content-Security-Policy, X-Frame-Options, and Access-Control-Allow-Origin headers for Cross-Origin Resource Sharing (CORS) scenarios. Security-based attributes can also be defined with cookies. • Route requests to mobile or desktop versions of your application based on the client device type. • Using redirect capabilities to return 301, 302, 307, and 308 redirects to the client to direct them to new hostnames, paths, query strings, or protocols. • Dynamically modify the caching configuration of your route based on the incoming requests. • Rewrite the request URL path and forwards the request to the appropriate origin in your configured origin group. • Add, modify, or remove request/response header to hide sensitive information or capture important information through headers. • Support server variables to dynamically change the request header, response headers, or URL rewrite paths/query strings. For example, when a new page load or when a form gets posted. Server variable is currently supported in **rule set actions** only. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

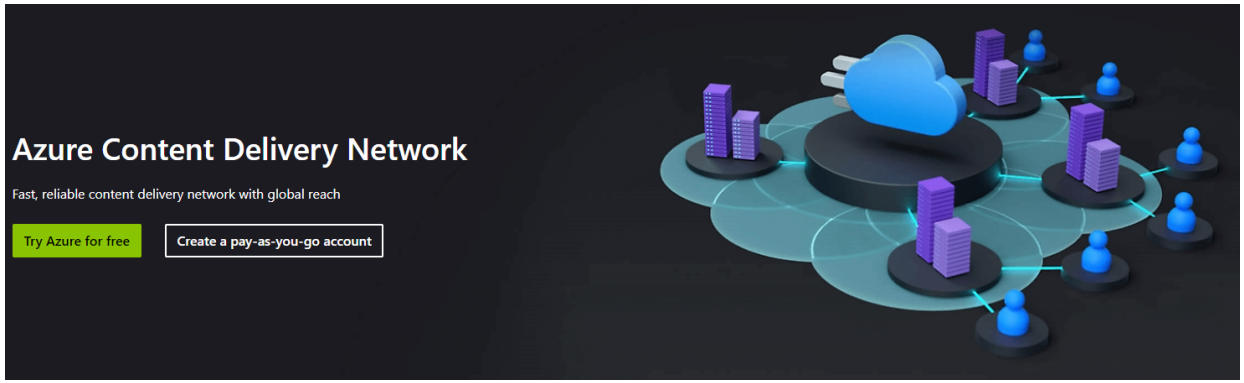
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Terminology

With a Front Door rule set, you can create any combination of configurations, each composed of a set of rules. The following out lines some helpful terminologies you come across when configuring your rule set.

- *Rule set*: A set of rules that gets associated to one or multiple routes.
- *Rule set rule*: A rule composed of up to 10 match conditions and 5 actions. Rules are local to a rule set and can't be exported to use across other rule sets. You can create the same rule in different rule sets.
- *Match condition*: There are many match conditions that you can configure to parse an incoming request. A rule can contain up to 10 match conditions. Match conditions are evaluated with an **AND** operator. *Regular expression is supported in conditions*. A full list of match conditions can be found in Rule set match conditions.
- *Action*: An action dictates how Front Door handles the incoming requests based on the matching conditions. You can modify caching behaviors, modify request headers, response headers, set URL rewrite, and URL redirection. *Server variables are supported with Action*. A rule can contain up to five actions. A full list of actions can be found in Rule set actions.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
35. The method of claim 27 wherein the global configuration data comprises at least one CDN resource.	Microsoft performs the method of claim 27 wherein the global configuration data comprises at least one CDN resource. For instance, Azure CDN and/or Azure Front Door global configuration data (e.g., routing policies, caching rules, content updates, and security settings) comprise CDN resources. See Claim limitation 27.a.
37. The method of claim 28 wherein the one or more other locations in the CDN comprise at least one location selected from: a cache server, a cache server site, a region of cache servers, a cache cluster, and a cache cluster site.	Microsoft performs the method of claim 28 wherein the one or more other locations in the CDN comprise at least one location selected from: a cache server, a cache server site, a region of cache servers, a cache cluster, and a cache cluster site. For instance, Azure CDN and/or Azure Front Door locations comprise at least cache servers, cache server sites, and regions of cache servers from which data may be obtained.  https://azure.microsoft.com/en-us/products/cdn **Increase speed and scalability for an optimal user experience** Azure Content Delivery Network offers a global solution for rapidly delivering content. Save bandwidth and improve responsiveness when encoding or distributing gaming software, firmware updates, and IoT endpoints. Reduce load times for websites, mobile apps, and streaming media to increase user satisfaction globally. https://azure.microsoft.com/en-us/products/cdn Stream media and download large files quickly with optimized delivery. Provide a scalable solution to handle worldwide traffic spikes and instantaneous high loads. Integrate seamlessly with your Azure services to activate within minutes. Protect content with custom domain HTTPS, DDoS, and WAF protection.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	https://azure.microsoft.com/en-us/products/cdn **Improved performance and user experience** Users expect a fast, reliable, and personalized web experience wherever they are. Content Delivery Network helps reduce latency and improve performance for high-bandwidth content by distributing user requests and serving content directly from edge servers. This brings the content closer to users and sends less traffic to the origin point, delivering superior online experiences. https://azure.microsoft.com/en-us/products/cdn **Global coverage and massive scalability** Scale on the fly with no downtime using a platform that responds to traffic fluctuations. A distributed, global presence enables Content Delivery Network to handle sudden traffic spikes and heavy loads—for example, during major product launches or global sporting events—without new infrastructure costs or capacity concerns. Global, distributed points of presence (PoPs) ensure fast delivery of content and protects the origin from excessive load. https://azure.microsoft.com/en-us/products/cdn

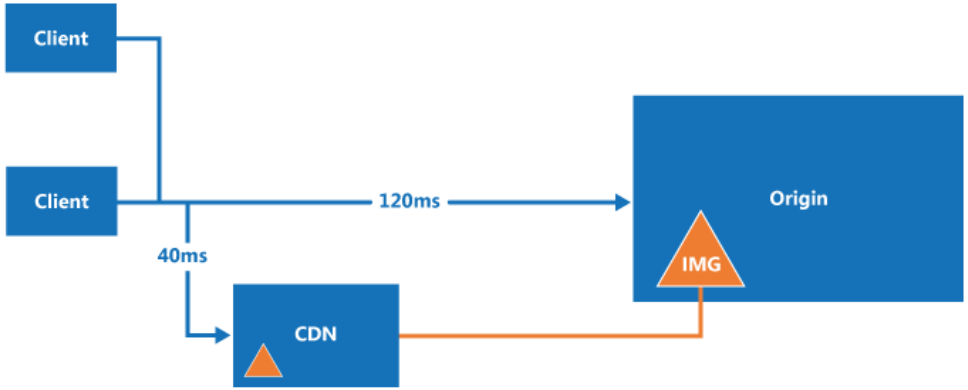
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

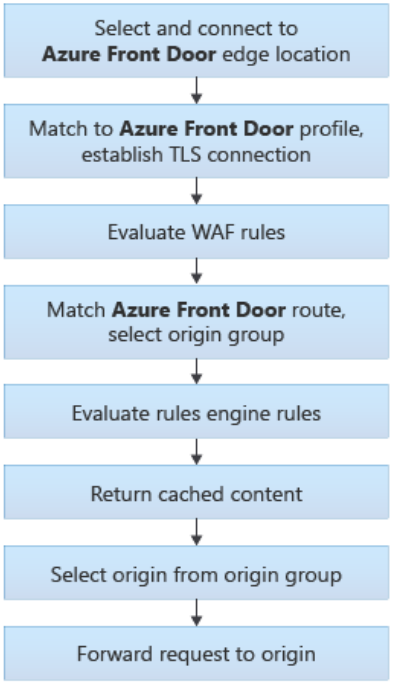
https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

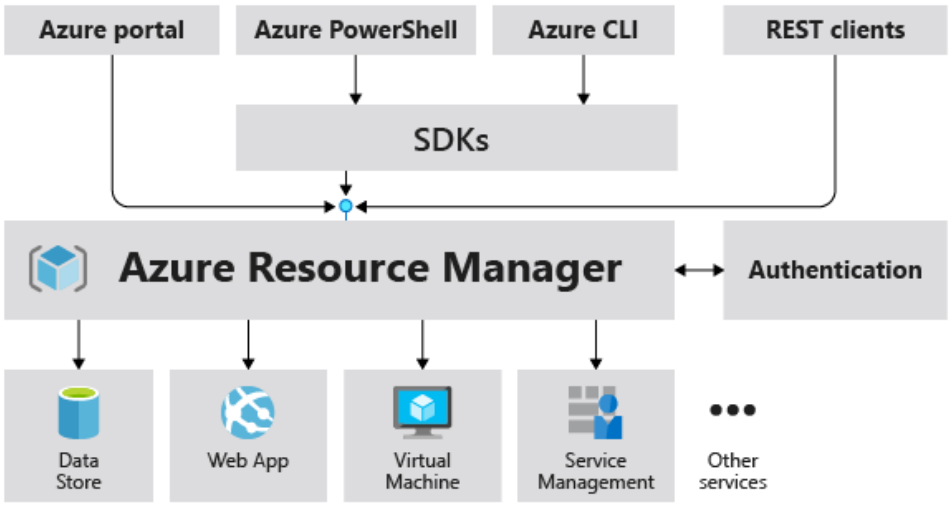
Region	Cities
North America	Etobicoke, Canada (2) Montreal, Canada Vancouver, Canada (2) Querétaro, Mexico (2) Atlanta, GA, USA (3) Boydton, VA, USA (2) Chaska, MN, USA (2) Cheyenne, WY, USA (2) Chicago, IL, USA (4) Dallas, TX, USA (4) Des Moines,IA, USA (3) Detroit, MI, USA Englewood, CO, USA (2) Honolulu, HI, USA Houston, TX, USA (3) Jacksonville, FL, USA (2) Las Vegas, NV, USA (3) Los Angeles, CA, USA (2) Manassas, VA, USA (3) Memphis, TN, USA Miami, FL, USA (4) Minneapolis, MN, USA (2) Needham Heights, MA, USA (2) Nashville, TN, USA Newark, NJ, USA New York, NY, USA (2) Philadelphia, PA, USA Phoenix, AZ, USA Plano, TX, USA Portland, OR, USA Quincy, WA, USA (3) San Antonio, TX, USA (4) San Jose, CA, USA (4) Salt Lake City, UT, USA (3) Seattle, WA, USA (2) Secaucus, NJ, USA (2) Southfield, MI, USA Tempe, AZ, USA

https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Connectivity within Microsoft's network along with new Regional Caching capabilities enables more consistent, more predictable cache fill performance by providing multi-tier caching along direct, private access to content in Azure from each CDN edge point of presence (POP). Azure CDN from Microsoft entered public preview providing access to 54 global Edge POPs in 33 countries and 16 Regional Cache POPs.  https://azure.microsoft.com/es-es/blog/new-locations-for-azure-cdn-now-available/ See also Claim 28.
38. The method of claim 37 wherein the one or more other locations in the CDN comprise at least one peer location.	Microsoft performs the method of claim 37 wherein the one or more other locations in the CDN comprise at least one peer location. For instance, other Azure CDN locations comprise at least one peer location (e.g. peer edge POPs). See also Claim 37.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
39. The method of claim 28 wherein the particular cache server determines said one or more other locations using said information in said global configuration data.	Microsoft performs the method of claim 28 wherein the particular cache server determines said one or more other locations using said information in said global configuration data. For instance, Azure CDN and/or Azure Front Door cache servers determine other CDN locations for obtaining data based on information in the global configuration data (e.g., routing policies, caching rules, content updates, and security settings). Azure Front Door traffic routing takes place over multiple stages. First, traffic is routed from the client to the Front Door. Then, Front Door uses your configuration to determine the origin to send the traffic to. The Front Door web application firewall, routing rules, rules engine, and caching configuration can all affect the routing process. The following diagram illustrates the routing architecture: 

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture  https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture See also Claim limitation 27.a.
40. The method of claim 27 wherein said particular cache server obtains said updated global configuration data in (B) from one or more locations in the CDN.	Microsoft performs the method of claim 27 wherein said particular cache server obtains said updated global configuration data in (B) from one or more locations in the CDN. For instance, an edge node may obtain updated global routing policies or caching rules from one or more locations in the CDN, such as Azure Resource Manager and/or an intermediate node in the CDN between Azure Resource Manager and the cache server. See Claim limitation 27.a and 28.
41. The method of claim 40 wherein the one or more locations in the CDN comprise one or more locations selected from: (i) the control core; and (ii) one or more locations selected from: a cache	Microsoft performs the method of claim 40 wherein the one or more locations in the CDN comprise one or more locations selected from: (i) the control core; and (ii) one or more locations selected from: a cache server, a cache server site, a region of cache servers, a cache cluster, and a cache cluster site. For instance, Azure CDN locations comprise one or more of at least Azure Resource Manager, cache servers, cache server sites, and regions of cache servers from which data may be obtained.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
server, a cache server site, a region of cache servers, a cache cluster, and a cache cluster site.	# Consistent management layer When you send a request through any of the Azure APIs, tools, or SDKs, Resource Manager receives the request. It authenticates and authorizes the request before forwarding it to the appropriate Azure service. Because all requests are handled through the same API, you see consistent results and capabilities in all the different tools. The following image shows the role Azure Resource Manager plays in handling Azure requests.  All capabilities that are available in the portal are also available through PowerShell, Azure CLI, REST APIs, and client SDKs. Functionality initially released through APIs are represented in the portal within 180 days of initial release. https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/overview See also Claim 37 and 40.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
42. The method of claim 27 wherein the control core comprises a distributed system consisting of a plurality of machines.	Microsoft performs the method of claim 27 wherein the control core comprises a distributed system consisting of a plurality of machines. For instance, Azure Resource Manager comprises a distributed system consisting of a plurality of machines (e.g., virtual machine servers).

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Resiliency of Resource Manager

Resource Manager is designed for resiliency and continuous availability. Resource Manager and control plane operations (requests sent to `management.azure.com`) in the REST API:

- Distribute across regions. Resource Manager has a separate instance in each Azure region, meaning that if a Resource Manager instance fails in one region, this doesn't affect the service's availability in another region; the same applies to other Azure services. Although Resource Manager is distributed across regions, some services are regional. This distinction means that while the initial handling of the control plane operation is resilient, the request might be susceptible to regional outages when forwarded to the service.
- Distribute across availability zones (and regions) in locations that have multiple availability zones. This distribution ensures that when a region loses one or more zones, Resource Manager can fail over to another zone or region. It continues to provide control plane capability for the resources.
- Do not depend on one logical datacenter.
- Are not taken down for maintenance activities.

This resiliency applies to services that receive requests through Resource Manager. Azure Key Vault is one service that benefits from this consistency.

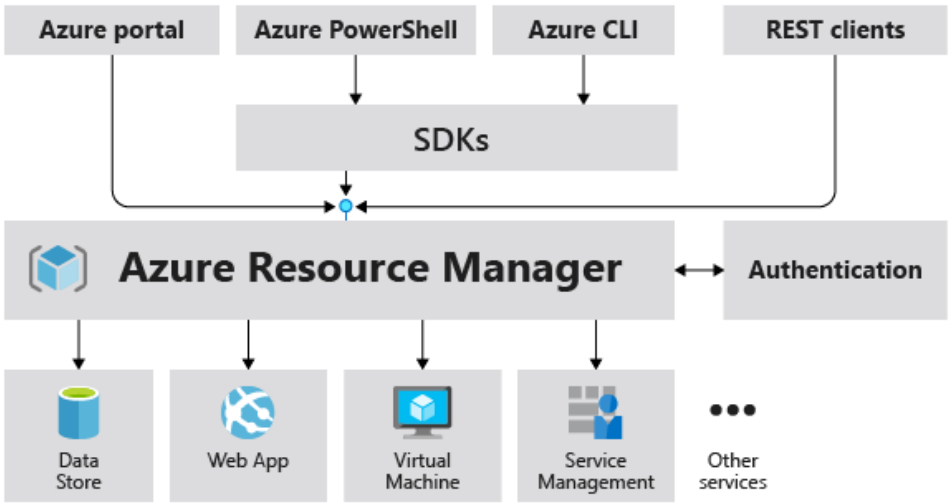
https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Consistent management layer

When you send a request through any of the Azure APIs, tools, or SDKs, Resource Manager receives the request. It authenticates and authorizes the request before forwarding it to the appropriate Azure service. Because all requests are handled through the same API, you see consistent results and capabilities in all the different tools.

The following image shows the role Azure Resource Manager plays in handling Azure requests.



All capabilities that are available in the portal are also available through PowerShell, Azure CLI, REST APIs, and client SDKs. Functionality initially released through APIs are represented in the portal within 180 days of initial release.

https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Workload resources

- **Azure Virtual Machines** serves as the compute resource for the application and is distributed across availability zones. For illustrative purposes, a combination of both Windows and Linux VMs is used.

  Azure Virtual Machine Scale Sets in Flexible orchestration mode is used to provision and manage the VMs.

  The sample application can be represented in two tiers, each requiring its own compute.
  - The front end runs the web server and receives user requests.
  - The back end runs another web server acting as a web API that exposes a single endpoint where the business logic is executed.

  The front-end VMs have data disks (Premium_LRS) attached, which could be used to deploy a stateless application. The back-end VMs persist data to Premium_ZRS local disks as part of its operation. This layout can be extended to include a database tier for storing state from the front-end and back-end compute. That tier is outside the scope of this architecture.

- **Azure Virtual Network** provides a private network for all workload resources. The network is segmented into subnets, which serve as isolation boundaries.

- **Azure Application Gateway** is the single point of ingress that routes requests to the front-end servers. The selected SKU includes integrated Azure Web Application Firewall (WAF) for added security.

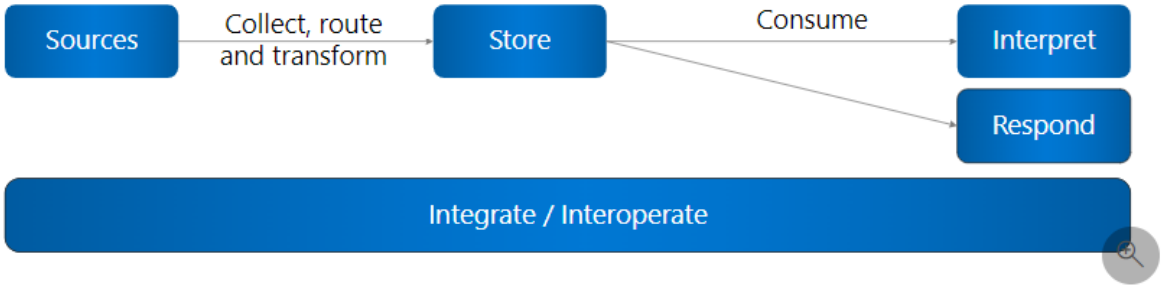
- **Internal Azure Load Balancer** routes traffic from the front-end tier to the back-end servers.

- **Azure Load Balancer** Standard SKU provides outbound internet access to the VMs using three public IP addresses.

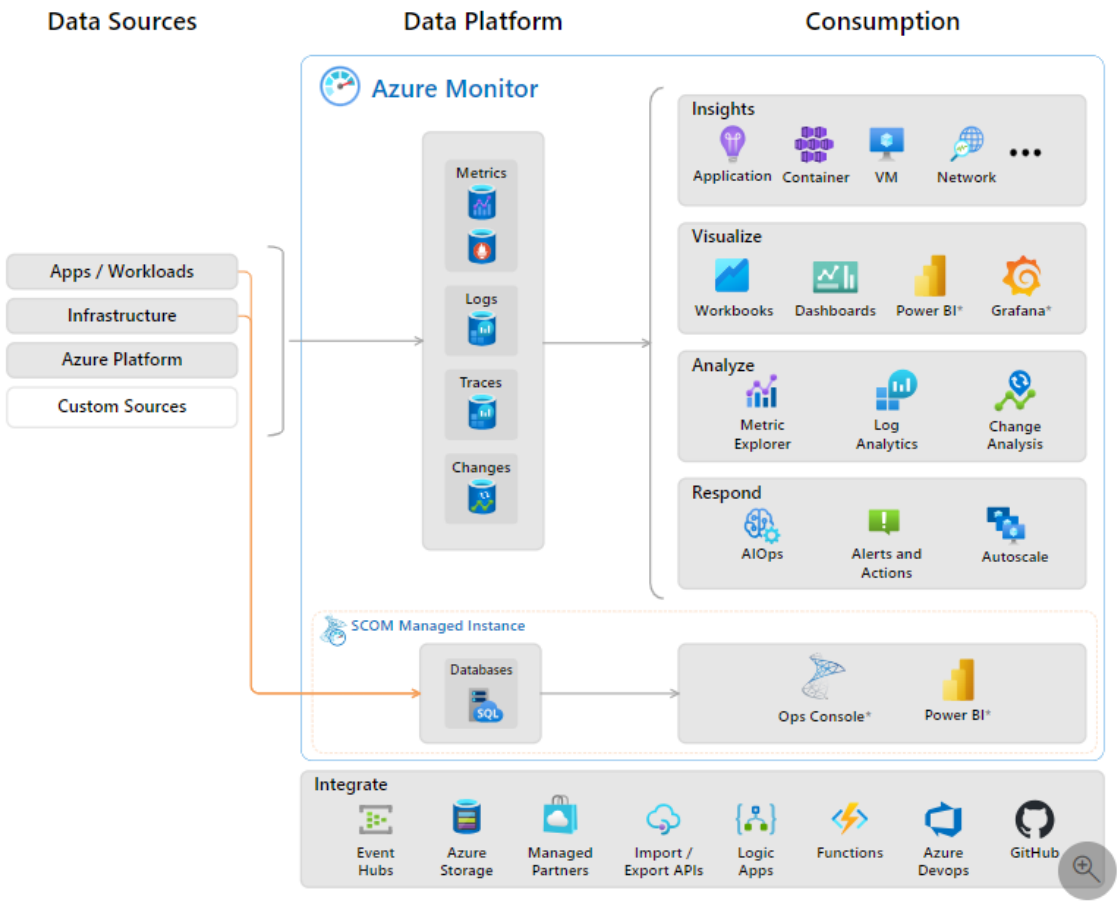
https://learn.microsoft.com/en-us/azure/architecture/virtual-machines/baseline

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	See also Claim limitation 27.a.
44. The method of claim 27 further comprising, by said particular cache server: **[44.a]** (F) determining validity of a version of global configuration data stored on said particular cache server, and	Microsoft performs the method of claim 27 further comprising, by said particular cache server, determining validity of a version of global configuration data stored on said particular cache server. For instance, the validity of a version of the global configuration data (e.g., routing policies, caching rules, content updates, and security settings) is determined, e.g., before applying during request processing. See Claim limitation 27.a.
[44.b] wherein said particular cache server obtains updated global configuration data in (B) when said particular cache server determines that said version of said global configuration data stored on said particular cache server is invalid.	Said particular cache server obtains updated global configuration data in (B) when said particular cache server determines that said version of said global configuration data stored on said particular cache server is invalid. For instance, an edge node may obtain updated global routing policies or caching rules when the version of configuration data is determined to be invalid. See Claim limitation 27.a.
45. The method of claim 27 further comprising, by said particular cache server: **[45.a]** (G) determining validity of a version of particular customer configuration information associated with the particular customer and stored on said particular cache server, and	Microsoft performs the method of claim 27 further comprising, by said particular cache server, determining validity of a version of particular customer configuration information associated with the particular customer and stored on said particular cache server. For instance, the validity of a version of the customer configuration information (e.g., whether the Azure Front Door rule sets are stale or updated) is determined, e.g., before applying during request processing. See Claim limitation 27.c.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
[45.b] wherein said particular cache server obtains customer configuration information in (C) when said particular cache server determines that said version of said particular customer configuration information stored on said particular cache server is invalid.	Said particular cache server obtains customer configuration information in (C) when said particular cache server determines that said version of said particular customer configuration information stored on said particular cache server is invalid. For instance, an edge node may obtain updated rule sets when the version of customer configuration information is determined to be invalid. See Claim limitation 27.c.
46. The method of claim 45 wherein said particular cache server determines validity of said version of particular customer configuration information in (G) in response to a request to serve content associated with said particular customer.	Microsoft performs the method of claim 45 wherein said particular cache server determines validity of said version of particular customer configuration information in (G) in response to a request to serve content associated with said particular customer. For instance, the validity of a version of the customer configuration information (e.g., Azure Front Door rule sets) is determined in response to a request to serve content associated with a particular customer. See Claim 45.
47. The method of claim 27 wherein the CDN further comprises: **[47.a]** (d) a collector system comprising one or more collector mechanisms, each of said collector mechanisms being constructed and adapted to obtain information from at least one cache server, the method further comprising,	Microsoft performs the method of claim 27 wherein the CDN further comprises: a collector system comprising one or more collector mechanisms, each of said collector mechanisms being constructed and adapted to obtain information from at least one cache server. For instance, Azure Monitor collects information from nodes.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Azure Monitor is a comprehensive monitoring solution for collecting, analyzing, and responding to monitoring data from your cloud and on-premises environments. You can use Azure Monitor to maximize the availability and performance of your applications and services. It helps you understand how your applications are performing and allows you to manually and programmatically respond to system events. Azure Monitor collects and aggregates the data from every layer and component of your system across multiple Azure and non-Azure subscriptions and tenants. It stores it in a common data platform for consumption by a common set of tools that can correlate, analyze, visualize, and/or respond to the data. You can also integrate other Microsoft and non-Microsoft tools.  The diagram above shows an abstracted view of the monitoring process. A more detailed breakdown of the Azure Monitor architecture is shown in the High level architecture section below. https://learn.microsoft.com/en-us/azure/azure-monitor/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	# High level architecture Azure Monitor can monitor these types of resources in Azure, other clouds, or on-premises: • Applications • Virtual machines • Guest operating systems • Containers including Prometheus metrics • Databases • Security events in combination with Azure Sentinel • Networking events and health in combination with Network Watcher • Custom sources that use the APIs to get data into Azure Monitor You can also export monitoring data from Azure Monitor into other systems so you can: • Integrate with other third-party and open-source monitoring and visualization tools • Integrate with ticketing and other ITSM systems If you're a System Center Operations Manager (SCOM) user, Azure Monitor now includes Azure Monitor SCOM Managed Instance (SCOM MI). Operations Manager MI is a cloud-hosted version of Operations Manager and allows you to move your on-premises Operations Manager installation to Azure. https://learn.microsoft.com/en-us/azure/azure-monitor/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	The following diagram shows a high-level architecture view of Azure Monitor.  https://learn.microsoft.com/en-us/azure/azure-monitor/overview

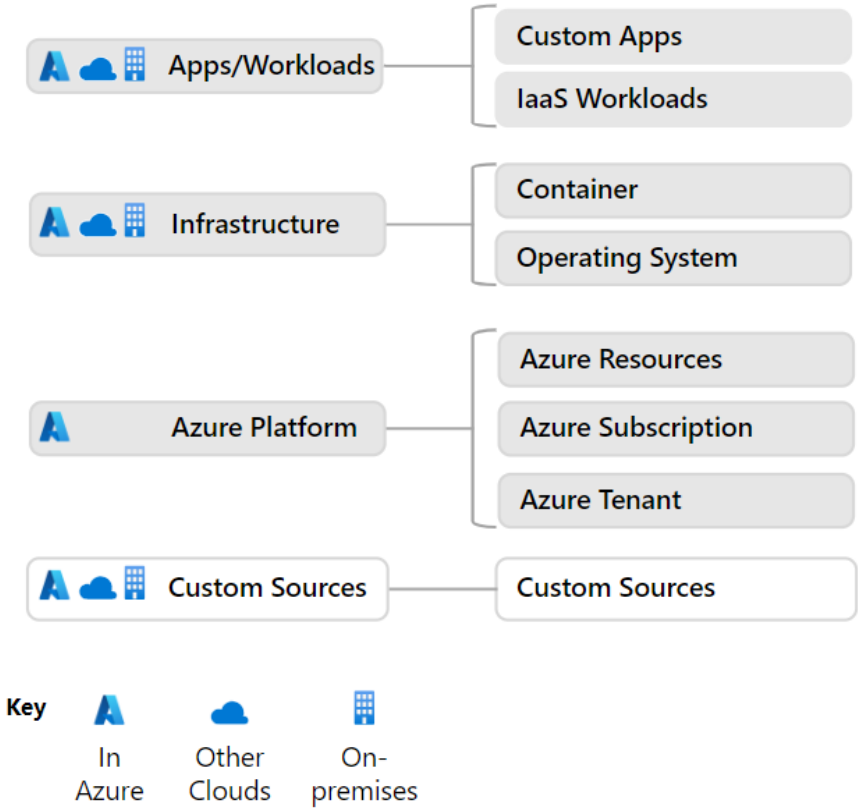
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	The diagram depicts the Azure Monitor system components: • Data sources are the types of resources being monitored. • The data is **collected and routed** to the data platform. Clicking on the diagram shows these options, which are also called out in detail later in this article. • The data platform stores the collected monitoring data. Azure Monitor's core data platform has stores for metrics, logs, traces, and changes. System Center Operations Manager MI uses its own database hosted in SQL Managed Instance. • The **consumption** section shows the components that use data from the data platform. ○ Azure Monitor's core consumption methods include tools to provide **insights**, **visualize**, and **analyze** data. The visualization tools build on the analysis tools and the insights build on top of both the visualization and analysis tools. ○ There are additional mechanisms to help you **respond** to incoming monitoring data. • The **SCOM MI** path uses the traditional Operations Manager console that SCOM customers are already familiar with. • Interoperability options are shown in the **integrate** section. Not all services integrate at all levels. SCOM MI only integrates with Power BI. https://learn.microsoft.com/en-us/azure/azure-monitor/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Data sources

Azure Monitor can collect data from multiple sources.

The diagram below shows an expanded version of the data source types that Azure Monitor can gather monitoring data from.



https://learn.microsoft.com/en-us/azure/azure-monitor/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Azure Monitor collects these types of data:

Expand table

Data Type	Description and subtypes
App/Workloads	**App**- Application performance, health, and activity data. **Workloads** - IaaS workloads such as SQL server, Oracle or SAP running on a hosted Virtual Machine.
Infrastructure	**Container** - Data about containers, such as Azure Kubernetes Service, Prometheus, and the applications running inside containers. **Operating system** - Data about the guest operating system on which your application is running.
Azure Platform	**Azure resource** - Data about the operation of an Azure resource from inside the resource, including changes. Resource Logs are one example. **Azure subscription** - The operation and management of an Azure subscription, and data about the health and operation of Azure itself. The activity log is one example. **Azure tenant** - Data about the operation of tenant-level Azure services, such as Microsoft Entra ID.
Custom Sources	Data that gets into the system using the - Azure Monitor REST API - Data Collection API

https://learn.microsoft.com/en-us/azure/azure-monitor/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Collection method	Description
Application instrumentation	Application Insights is enabled through either Auto-Instrumentation (agent) or by adding the Application Insights SDK to your application code. In addition, Application Insights is in process of implementing Open Telemetry. For more information, reference How do I instrument an application?.
Agents	Agents can collect monitoring data from the guest operating system of Azure and hybrid virtual machines.
Data collection rules	Use data collection rules to specify what data should be collected, how to transform it, and where to send it.
Zero Config	Data is automatically sent to a destination without user configuration. Platform metrics are the most common example.
Diagnostic settings	Use diagnostic settings to determine where to send resource log and activity log data on the data platform.
Azure Monitor REST API	The Logs Ingestion API in Azure Monitor lets you send data to a Log Analytics workspace in Azure Monitor Logs. You can also send metrics into the Azure Monitor Metrics store using the custom metrics API.

https://learn.microsoft.com/en-us/azure/azure-monitor/overview

Pillar of Observability/ Data Store	Description
Azure Monitor Metrics	Metrics are numerical values that describe an aspect of a system at a particular point in time. Azure Monitor Metrics is a time-series database, optimized for analyzing time-stamped data. Azure Monitor collects metrics at regular intervals. Metrics are identified with a timestamp, a name, a value, and one or more defining labels. They can be aggregated using algorithms, compared to other metrics, and analyzed for trends over time. It supports native Azure Monitor metrics and Prometheus metrics.
Azure Monitor Logs	Logs are recorded system events. Logs can contain different types of data, be structured or free-form text, and they contain a timestamp. Azure Monitor stores structured and unstructured log data of all types in Azure Monitor Logs. You can route data to Log Analytics workspaces for querying and analysis.
Traces	Distributed tracing allows you to see the path of a request as it travels through different services and components. Azure Monitor gets distributed trace data from instrumented applications. The trace data is stored in a separate workspace in Azure Monitor Logs.
Changes	Changes are a series of events in your application and resources. They're tracked and stored when you use the Change Analysis service, which uses Azure Resource Graph as its store. Change Analysis helps you understand which changes, such as deploying updated code, may have caused issues in your systems.

https://learn.microsoft.com/en-us/azure/azure-monitor/overview

Consumption

The following sections outline methods and services that consume monitoring data from the Azure Monitor data platform.

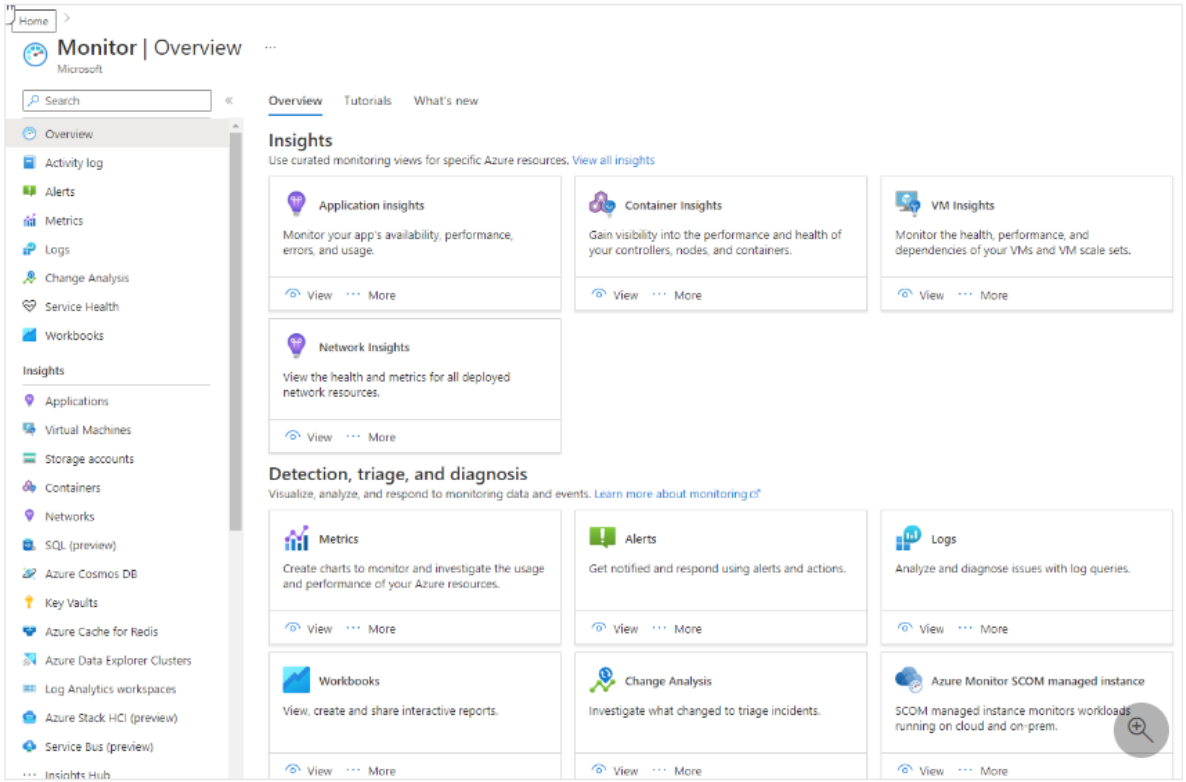
All areas in the *consumption* section of the diagram have a user interface that appears in the Azure portal.

The top part of the consumption section applies to Azure Monitor core only. SCOM MI uses the traditional Ops Console running in the cloud. It can also send monitoring data to Power BI for visualization.

https://learn.microsoft.com/en-us/azure/azure-monitor/overview

The Azure portal

The Azure portal is a web-based, unified console that provides an alternative to command-line tools. With the Azure portal, you can manage your Azure subscription using a graphical user interface. You can build, manage, and monitor everything from simple web apps to complex cloud deployments in the portal. The *Monitor* section of the Azure portal provides a visual interface that gives you access to the data collected for Azure resources and an easy way to access the tools, insights, and visualizations in Azure Monitor.



https://learn.microsoft.com/en-us/azure/azure-monitor/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Insight	Description
Application Insights	Application Insights monitors the availability, performance, and usage of your web applications.
Container Insights	Container Insights gives you performance visibility into container workloads that are deployed to managed Kubernetes clusters hosted on Azure Kubernetes Service. Container Insights collects container logs and metrics from controllers, nodes, and containers that are available in Kubernetes through the Metrics API. After you enable monitoring from Kubernetes clusters, these metrics and logs are automatically collected for you through a containerized version of the Log Analytics agent for Linux.
VM Insights	VM Insights monitors your Azure VMs. It analyzes the performance and health of your Windows and Linux VMs and identifies their different processes and interconnected dependencies on external processes. The solution includes support for monitoring performance and application dependencies for VMs hosted on-premises or another cloud provider.
Network Insights	Network Insights provides a comprehensive and visual representation through topologies, of health and metrics for all deployed network resources, without requiring any configuration. It also provides access to network monitoring capabilities like Connection Monitor, flow logging for network security groups (NSGs), and Traffic Analytics and other diagnostic features.

https://learn.microsoft.com/en-us/azure/azure-monitor/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Visualization	Description
Dashboards	Azure dashboards allow you to combine different kinds of data into a single pane in the Azure portal. You can optionally share the dashboard with other Azure users. You can add the output of any log query or metrics chart to an Azure dashboard. For example, you could create a dashboard that combines tiles that show a graph of metrics, a table of activity logs, a usage chart from Application Insights, and the output of a log query.
Workbooks	Workbooks provide a flexible canvas for data analysis and the creation of rich visual reports in the Azure portal. You can use them to query data from multiple data sources. Workbooks can combine and correlate data from multiple data sets in one visualization giving you easy visual representation of your system. Workbooks are interactive and can be shared across teams with data updating in real time. Use workbooks provided with Insights, utilize the library of templates, or create your own.
Power BI	Power BI is a business analytics service that provides interactive visualizations across various data sources. It's an effective means of making data available to others within and outside your organization. You can configure Power BI to automatically import log data from Azure Monitor to take advantage of these visualizations.
Grafana	Grafana is an open platform that excels in operational dashboards. All versions of Grafana include the Azure Monitor data source plug-in to visualize your Azure Monitor metrics and logs. Azure Managed Grafana also optimizes this experience for Azure-native data stores such as Azure Monitor and Azure Data Explorer. In this way, you can easily connect to any resource in your subscription and view all resulting monitoring data in a familiar Grafana dashboard. It also supports pinning charts from Azure Monitor metrics and logs to Grafana dashboards. Grafana has popular plug-ins and dashboard templates for non-Microsoft APM tools such as Dynatrace, New Relic, and AppDynamics as well. You can use these resources to visualize Azure platform data alongside other metrics from higher in the stack collected by these other tools. It also has AWS CloudWatch and GCP BigQuery plug-ins for multicloud monitoring in a single pane of glass.

https://learn.microsoft.com/en-us/azure/azure-monitor/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Tool	Description
Metrics explorer	Use the Azure Monitor metrics explorer user interface in the Azure portal to investigate the health and utilization of your resources. Metrics explorer helps you plot charts, visually correlate trends, and investigate spikes and dips in metric values. Metrics explorer contains features for applying dimensions and filtering, and for customizing charts. These features help you analyze exactly the data you need in a visually intuitive way.
Log Analytics	The Log Analytics user interface in the Azure portal helps you query the log data collected by Azure Monitor so that you can quickly retrieve, consolidate, and analyze collected data. After creating test queries, you can then directly analyze the data with Azure Monitor tools, or you can save the queries for use with visualizations or alert rules. Log Analytics workspaces are based on Azure Data Explorer, using a powerful analysis engine and the rich Kusto query language (KQL).Azure Monitor Logs uses a version of the Kusto Query Language suitable for simple log queries, and advanced functionality such as aggregations, joins, and smart analytics. You can get started with KQL quickly and easily. NOTE: The term "Log Analytics" is sometimes used to mean both the Azure Monitor Logs data platform store and the UI that accesses that store. Previous to 2019, the term "Log Analytics" did refer to both. It's still common to find content using that framing in various blogs and documentation on the internet.
Change Analysis	Change Analysis is a subscription-level Azure resource provider that checks resource changes in the subscription and provides data for diagnostic tools to help users understand what changes might have caused issues. The Change Analysis user interface in the Azure portal gives you insight into the cause of live site issues, outages, or component failures. Change Analysis uses the Azure Resource Graph to detect various types of changes, from the infrastructure layer through application deployment.

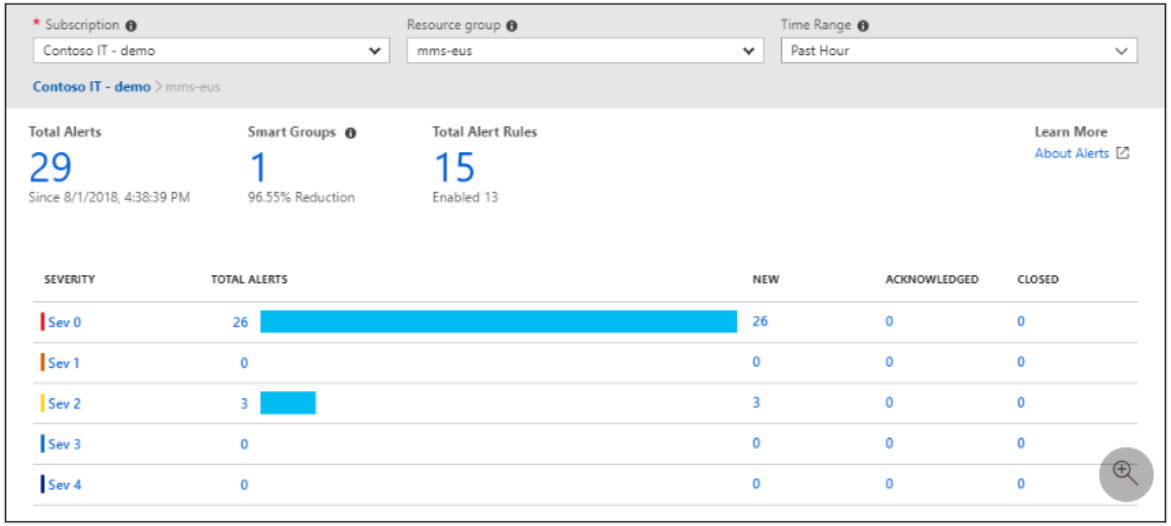
https://learn.microsoft.com/en-us/azure/azure-monitor/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Azure Monitor Alerts notify you of critical conditions and can take corrective action. Alert rules can be based on metric or log data.

- Metric alert rules provide near-real-time alerts based on collected metrics.
- Log search alert rules based on logs allow for complex logic across data from multiple sources.

Alert rules use action groups, which can perform actions such as sending email or SMS notifications. Action groups can send notifications using webhooks to trigger external processes or to integrate with your IT service management tools. Action groups, actions, and sets of recipients can be shared across multiple rules.



https://learn.microsoft.com/en-us/azure/azure-monitor/overview

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	<table><tr><th>Azure service</th><th>Description</th></tr><tr><td>Event Hubs</td><td>Azure Event Hubs is a streaming platform and event ingestion service. It can transform and store data by using any real-time analytics provider or batching/storage adapters. Use Event Hubs to stream Azure Monitor data to partner SIEM and monitoring tools.</td></tr><tr><td>Azure Storage</td><td>Export data to Azure storage for less expensive, long-term archival of monitoring data for auditing or compliance purposes.</td></tr><tr><td>Hosted and Managed Partners</td><td>Many external partners integrate with Azure Monitor. Azure Monitor has also partnered with a few monitoring providers to provide an Azure-hosted version of their products to make interoperability easier. Examples include Elastic, Datadog, Logz.io, and Dynatrace.</td></tr><tr><td>API</td><td>Multiple APIs are available to read and write metrics and logs to and from Azure Monitor in addition to accessing generated alerts. You can also configure and retrieve alerts. With APIs, you have unlimited possibilities to build custom solutions that integrate with Azure Monitor.</td></tr><tr><td>Azure Logic Apps</td><td>Azure Logic Apps is a service you can use to automate tasks and business processes by using workflows that integrate with different systems and services with little or no code. Activities are available that read and write metrics and logs in Azure Monitor. You can use Logic Apps to customize responses and perform other actions in response to Azure Monitor alerts. You can also perform other more complex actions when the Azure Monitor infrastructure doesn't already supply a built-it method.</td></tr><tr><td>Azure Functions</td><td>Similar to Azure Logic Apps, Azure Functions give you the ability to preprocess and post process monitoring data and perform complex action beyond the scope of typical Azure Monitor alerts. Azure Functions uses code however providing additional flexibility over Logic Apps.</td></tr><tr><td>Azure DevOps and GitHub</td><td>Azure Monitor Application Insights gives you the ability to create Work Item Integration with monitoring data embedding in it. Additional options include release annotations and continuous monitoring.</td></tr></table> https://learn.microsoft.com/en-us/azure/azure-monitor/overview
by said particular cache server: **[47.b]** (E) generating information; and	Microsoft generates information by said particular cache server. For instance, Azure CDN and/or Azure Front Door cache servers generate information related to performance. See Claim limitation 47.a.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
[47.c] (F) providing at least some of said generated information to said collector system.	Microsoft provides at least some of said generated information to said collector system. For instance, Azure CDN and/or Azure Front Door cache servers provide generated information to Azure Monitor. See Claim limitation 47.a.
48. The method of claim 47 wherein said providing in (F) comprises: (F)(1) streaming said at least some of said generated information to one or more collector mechanisms in said collector system.	Microsoft performs the method of claim 47 wherein said providing in (F) comprises: streaming said at least some of said generated information to one or more collector mechanisms in said collector system. For instance, Azure CDN and/or Azure Front Door cache servers stream generated information to Azure Monitor. See Claim limitation 47.a.
49. The method of claim 27 further comprising, by said particular cache server of said plurality of cache servers: **[49.a]** (E) obtaining second customer configuration information associated with a second customer of the CDN, said second customer being distinct from said particular customer; and	Microsoft performs the method of claim 27 further comprising, by said particular cache server of said plurality of cache servers, obtaining second customer configuration information associated with a second customer of the CDN, said second customer being distinct from said particular customer. For instance, Azure CDN and/or Azure Front Door cache servers obtain configuration information associated with different CDN customers in a similar manner as previously described. See Claim limitation 27.c.
[49.b] (F) serving a second particular resource associated with the second customer in accordance with (i) the global configuration data, and (ii) the second customer configuration information	Microsoft serves a second particular resource associated with the second customer in accordance with (i) the global configuration data, and (ii) the second customer configuration information associated with the second customer. For instance, Azure CDN and/or Azure Front Door cache servers serve resources associated with different CDN customers in accordance with the global configuration data and customer configuration information in a similar manner as previously described. See Claim limitation 27.d.

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
associated with the second customer.	
50. The method of claim 27 wherein the CDN has customer configuration information associated therewith, and wherein the particular cache server serves CDN resources in accordance with the global configuration data and the customer configuration information associated with the CDN.	Microsoft performs the method of claim 27 wherein the CDN has customer configuration information associated therewith, and wherein the particular cache server serves CDN resources in accordance with the global configuration data and the customer configuration information associated with the CDN. For instance, Azure CDN and/or Azure Front Door provide rule sets configurable specific to CDN customers, which are processed along with global configuration data (e.g., routing policies, caching rules, content updates, and security settings) to serve CDN resources. A rule set is a customized rules engine that groups a combination of rules into a single set. You can associate a rule set with multiple routes. A Rule set allows you to customize how requests get processed and handled at the Azure Front Door edge. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

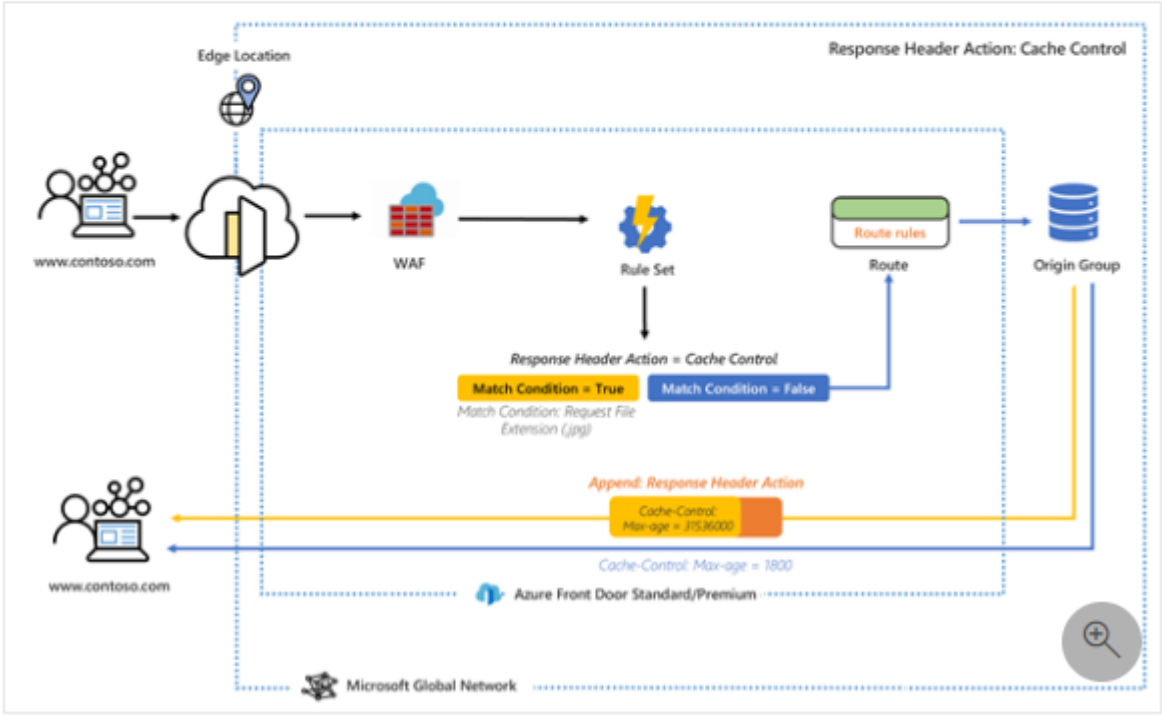
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Common supported scenarios • Implementing security headers to prevent browser-based vulnerabilities like HTTP Strict-Transport-Security (HSTS), X-XSS-Protection, Content-Security-Policy, X-Frame-Options, and Access-Control-Allow-Origin headers for Cross-Origin Resource Sharing (CORS) scenarios. Security-based attributes can also be defined with cookies. • Route requests to mobile or desktop versions of your application based on the client device type. • Using redirect capabilities to return 301, 302, 307, and 308 redirects to the client to direct them to new hostnames, paths, query strings, or protocols. • Dynamically modify the caching configuration of your route based on the incoming requests. • Rewrite the request URL path and forwards the request to the appropriate origin in your configured origin group. • Add, modify, or remove request/response header to hide sensitive information or capture important information through headers. • Support server variables to dynamically change the request header, response headers, or URL rewrite paths/query strings. For example, when a new page load or when a form gets posted. Server variable is currently supported in **rule set actions** only. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	# Architecture Rule sets handle requests at the Front Door edge. When a request arrives at your Front Door endpoint, WAF (Web Application Firewall) is processed first, followed by the settings configured in route. Those settings include the rule set associated to the route. Rule sets are processed in the order they appear under the routing configuration. Rules in a rule set also get processed in the order they appear. In order for all the actions in each rule to run, all the match conditions within a rule have to be met. If a request doesn't match any of the conditions in your rule set configuration, then only the default route settings get applied. If the **Stop evaluating remaining rules** is selected, then any remaining rule sets associated with the route don't get ran. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Example

In the following diagram, WAF policies get processed first. Then the rule set configuration appends a response header. The header changes the max-age of the cache control if the match condition is true.



https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Terminology

With a Front Door rule set, you can create any combination of configurations, each composed of a set of rules. The following out lines some helpful terminologies you come across when configuring your rule set.

- *Rule set*: A set of rules that gets associated to one or multiple routes.
- *Rule set rule*: A rule composed of up to 10 match conditions and 5 actions. Rules are local to a rule set and can't be exported to use across other rule sets. You can create the same rule in different rule sets.
- *Match condition*: There are many match conditions that you can configure to parse an incoming request. A rule can contain up to 10 match conditions. Match conditions are evaluated with an **AND** operator. *Regular expression is supported in conditions*. A full list of match conditions can be found in Rule set match conditions.
- *Action*: An action dictates how Front Door handles the incoming requests based on the matching conditions. You can modify caching behaviors, modify request headers, response headers, set URL rewrite, and URL redirection. *Server variables are supported with Action*. A rule can contain up to five actions. A full list of actions can be found in Rule set actions.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

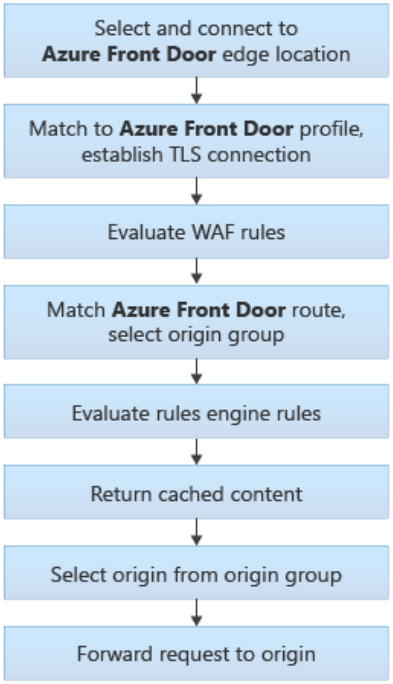
U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Terminology

With a Front Door rule set, you can create any combination of configurations, each composed of a set of rules. The following out lines some helpful terminologies you come across when configuring your rule set.

- *Rule set*: A set of rules that gets associated to one or multiple routes.
- *Rule set rule*: A rule composed of up to 10 match conditions and 5 actions. Rules are local to a rule set and can't be exported to use across other rule sets. You can create the same rule in different rule sets.
- *Match condition*: There are many match conditions that you can configure to parse an incoming request. A rule can contain up to 10 match conditions. Match conditions are evaluated with an **AND** operator. *Regular expression is supported in conditions*. A full list of match conditions can be found in Rule set match conditions.
- *Action*: An action dictates how Front Door handles the incoming requests based on the matching conditions. You can modify caching behaviors, modify request headers, response headers, set URL rewrite, and URL redirection. *Server variables are supported with Action*. A rule can contain up to five actions. A full list of actions can be found in Rule set actions.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Azure Front Door traffic routing takes place over multiple stages. First, traffic is routed from the client to the Front Door. Then, Front Door uses your configuration to determine the origin to send the traffic to. The Front Door web application firewall, routing rules, rules engine, and caching configuration can all affect the routing process. The following diagram illustrates the routing architecture:  https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	## Select and connect to the Front Door edge location The user or client application initiates a connection to the Front Door. The connection terminates at an edge location closest to the end user. Front Door's edge location processes the request. For more information about how requests are made to Front Door, see Front Door traffic acceleration. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture ## Match request to a Front Door profile When Front Door receives an HTTP request, it uses the request's `Host` header to match the request to the correct customer's Front Door profile. If the request is using a custom domain name, the domain name must be registered with Front Door to enable requests to get matched to your profile. The client and server perform a TLS handshake using the TLS certificate you configured for your custom domain name, or by using the Front Door certificate when the `Host` header ends with `*.azurefd.net`. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities

Evaluate WAF rules

If your domain has Web Application Firewall enabled, WAF rules are evaluated.

If a rule gets violated, Front Door returns an error to the client and the request processing stops.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture

Match a route

Front Door matches the request to a route. Learn more about the route matching process.

The route specifies the origin group that the request should be sent to.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture

Evaluate rule sets

If you define rule sets for the route, they get process in the order configured. Rule sets can override the origin group specified in a route. Rule sets can also trigger a redirection response to the request instead of forwarding it to an origin.

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture

U.S. Patent No. 9,456,053	Microsoft Accused Functionalities
	Return cached response If the Front Door routing rule has caching enabled, and the Front Door edge location's cache includes a valid response for the request, then Front Door returns the cached response. If caching is disabled or no response is available, the request is forwarded to the origin. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture **Select origin** Front Door selects an origin to use within the origin group. Origin selection is based on several factors, including: • Health of each origin, which Front Door monitors by using health probes. • Routing method for your origin group. • If you enable session affinity https://learn.microsoft.com/en-us/azure/frontdoor/front-door-routing-architecture See also Claim limitation 27.d.

EXHIBIT D

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
1[Pre]. A computer-implemented method, operable in a network comprising multiple service endpoints running on a plurality of devices, the method operable on a device comprising hardware including memory and at least one processor, the method comprising,	Microsoft performs a computer-implemented method, operable in a network comprising multiple service endpoints running on a plurality of devices, the method operable on a device comprising hardware including memory and at least one processor as set forth below. For instance, Azure Service Fabric provides a network comprising multiple service endpoints (e.g., nodes in a cluster) running on a plurality of devices (e.g., virtual machine servers) comprising hardware including memory and at least one processor in the manner of claim 1. Azure Service Fabric is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices. Service Fabric is a container and process orchestrator that allows you to host your clusters anywhere: on Azure, in an on-premises datacenter, or on any cloud provider. You can use any framework to write your services and choose where to run the application from multiple environment choices. This article details the terminology used by Service Fabric to understand the terms used in the documentation. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	# Environments Service Fabric is an open-source platform technology that several different services and products are based on. Microsoft provides the following options: • **Azure Service Fabric**: The Azure hosted Service Fabric cluster offering. It provides integration between Service Fabric and the Azure infrastructure, along with upgrade and configuration management of Service Fabric clusters. • **Service Fabric standalone**: A set of installation and configuration tools to deploy Service Fabric clusters anywhere (on-premises or on any cloud provider). Not managed by Azure. • **Service Fabric development cluster**: Provides a local development experience on Windows, Linux, or Mac for development of Service Fabric applications. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Focus on building applications and business logic, and let Azure solve the hard distributed systems problems such as reliability, scalability, management, and latency. Service Fabric is an open source project and it powers core Azure infrastructure as well as other Microsoft services such as Skype for Business, Intune, Azure Event Hubs, Azure Data Factory, Azure Cosmos DB, Azure SQL Database, Dynamics 365, and Cortana. Designed to deliver highly available and durable services at cloud-scale, Azure Service Fabric intrinsically understands the available infrastructure and resource needs of applications, enabling automatic scale, rolling upgrades, and self-healing from faults when they occur.

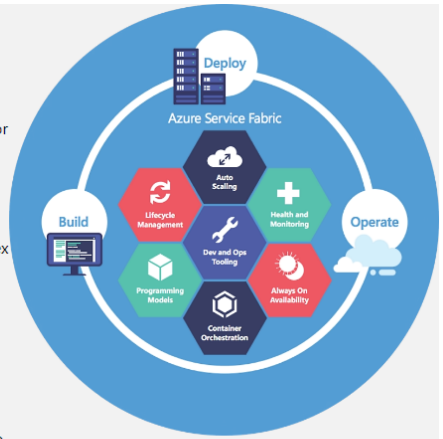
Choose from a variety of productive programming models and languages including .NET Core 2.0, C#, and Java to build your microservice and container-based applications. Deploy Service Fabric cluster in Azure or use Azure Service Fabric mesh - a fully managed microservices platform, currently in preview. Azure Service Fabric is also available as a free download for Windows Server, enabling you to create Service Fabric clusters on premises or in other clouds.

- ✓ Simplify microservices development and application lifecycle management
- ✓ Data-aware platform for low-latency, high-throughput workloads with stateful containers or microservices
- ✓ Run anywhere – supports Windows/Linux in Azure, on-premises, or other clouds
- ✓ Reliably scale and orchestrate containers and microservices
- ✓ Run anything – your choice of languages and programming models
- ✓ Scales up to thousands of machines

https://azure.microsoft.com/en-us/products/service-fabric

Simplify building and managing microservice applications

- **Fast time to market:** Focus on building features that add business value to your application, without the overhead of designing and writing additional code to deal with issues of reliability, scalability, management, or latency in the underlying infrastructure.
- **Container and service orchestration in the same environment:** Bring your Docker containers on Linux or Windows and run them reliably at scale alongside other workloads and services. Take advantage of Azure Service Fabric's container hosting, cluster resource management, and workload orchestration capabilities.
- **Choose your architecture:** Build stateless or stateful microservices—an architectural approach where complex applications are composed of small, independently versioned services—to power the most complex, low-latency, data-intensive scenarios and scale them into or across the cloud with Azure Service Fabric. Mix and match the programming models and languages that work for you, from containers and guest-executables to microservices and actors.
- **Microservice agility:** Enable continuous integration and development practices and accelerate delivery of new features by architecting fine-grained microservice applications.
- **IDE integration:** Quickly and easily build, test, debug, deploy, and upgrade Service Fabric applications in single-box, test, and production environments. Do this using the IDE tool of your choice, such as Visual Studio, Eclipse, or command-line support.
- **Run anywhere:** Flexibly deploy the same application code on public, hosted, or private clouds using consistent platform services and the same application programming models, with a choice of Windows Server or Linux (Ubuntu or RHEL) as the host operating system.



U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

https://azure.microsoft.com/en-us/products/service-fabric



Deliver low latency and improved efficiency at massive scale

Deliver fast yet safe upgrades with zero downtime, automate scale operations, integrate health monitoring, and provide automatic recovery from failures. Orchestrate microservices and container-based apps, gain insight into application health and performance, and enable development of low-latency, resilient services that scale.

Solve difficult distributed systems problems such as reliable failure detection and failover, leader election, state management, service discovery, rolling upgrades, application portability, environment abstraction, resource management and governance, and configuration management, and provide application lifecycle management capabilities so developers don't have to re-architect applications as usage grows. Develop and deliver many kinds of applications and workloads, including multi-tenant SaaS applications, mission-critical line-of-business applications, IoT data gathering and processing workloads, and gaming.

https://azure.microsoft.com/en-us/products/service-fabric

Select the OS for the cluster

Service Fabric allows for the creation of Service Fabric clusters on any VMs or computers running Windows Server or Linux. Before deploying your cluster, you must choose the OS: Windows or Linux. Every node (virtual machine) in the cluster runs the same OS, you cannot mix Windows and Linux VMs in the same cluster.

Capacity planning

For any production deployment, capacity planning is an important step. Here are some things to consider as a part of that process.

- The initial number of node types for your cluster
- The properties of each of node type (size, number of instances, primary, internet facing, number of VMs, etc.)
- The reliability and durability characteristics of the cluster

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-cluster-azure-deployment-preparation **Select the initial number of node types** First, you need to figure out what the cluster you are creating is going to be used for. What kinds of applications you are planning to deploy into this cluster? Does your application have multiple services, and do any of them need to be public or internet facing? Do your services (that make up your application) have different infrastructure needs such as greater RAM or higher CPU cycles? A Service Fabric cluster can consist of more than one node type: a primary node type and one or more non-primary node types. Each node type is mapped to a virtual machine scale set. Each node type can then be scaled up or down independently, have different sets of ports open, and can have different capacity metrics. Node properties and placement constraints can be set up to constrain specific services to specific node types. For more information, see Service Fabric cluster capacity planning. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-cluster-azure-deployment-preparation
[1.Pre.1] by a node in a group of said multiple service endpoints, wherein each node in the group assumes one or more discrete responsibilities involved in collaborative processing of a request across the group, and	Microsoft performs the method steps below by a node in a group of said multiple service endpoints, wherein each node in the group assumes one or more discrete responsibilities involved in collaborative processing of a request across the group. For instance, Azure Service Fabric clusters assume one or more discrete responsibilities (e.g., services or microservices) in order to collaboratively process requests across the group of service endpoints (e.g., nodes in a cluster).

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	# Infrastructure concepts **Cluster**: A network-connected set of virtual or physical machines into which your microservices are deployed and managed. Clusters can scale to thousands of machines. **Node**: A machine or VM that's part of a cluster is called a *node*. Each node is assigned a node name (string). Nodes have characteristics, such as placement properties. Each machine or VM has an auto-start Windows service, `FabricHost.exe`, that starts running upon boot and then starts two executables: `Fabric.exe` and `FabricGateway.exe`. These two executables make up the node. For testing scenarios, you can host multiple nodes on a single machine or VM by running multiple instances of `Fabric.exe` and `FabricGateway.exe`. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Application and service concepts

Service Fabric Native Application: Service Fabric Native Applications are described by the Native Application Model (XML-based application and service manifests).

Service Fabric Native Application concepts

Application: An application is a collection of constituent services that perform a certain function or functions. The lifecycle of each application instance can be managed independently.

Service: A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service.

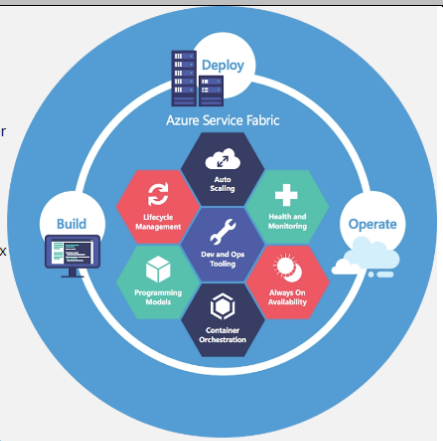
Application type: The name/version assigned to a collection of service types. It is defined in an `ApplicationManifest.xml` file and embedded in an application package directory. The directory is then copied to the Service Fabric cluster's image store. You can then create a named application from this application type within the cluster.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Simplify building and managing microservice applications

- **Fast time to market:** Focus on building features that add business value to your application, without the overhead of designing and writing additional code to deal with issues of reliability, scalability, management, or latency in the underlying infrastructure.
- **Container and service orchestration in the same environment:** Bring your Docker containers on Linux or Windows and run them reliably at scale alongside other workloads and services. Take advantage of Azure Service Fabric's container hosting, cluster resource management, and workload orchestration capabilities.
- **Choose your architecture:** Build stateless or stateful microservices—an architectural approach where complex applications are composed of small, independently versioned services—to power the most complex, low-latency, data-intensive scenarios and scale them into or across the cloud with Azure Service Fabric. Mix and match the programming models and languages that work for you, from containers and guest-executables to microservices and actors.
- **Microservice agility:** Enable continuous integration and development practices and accelerate delivery of new features by architecting fine-grained microservice applications.
- **IDE integration:** Quickly and easily build, test, debug, deploy, and upgrade Service Fabric applications in single-box, test, and production environments. Do this using the IDE tool of your choice, such as Visual Studio, Eclipse, or command-line support.
- **Run anywhere:** Flexibly deploy the same application code on public, hosted, or private clouds using consistent platform services and the same application programming models, with a choice of Windows Server or Linux (Ubuntu or RHEL) as the host operating system.



https://azure.microsoft.com/en-us/products/service-fabric



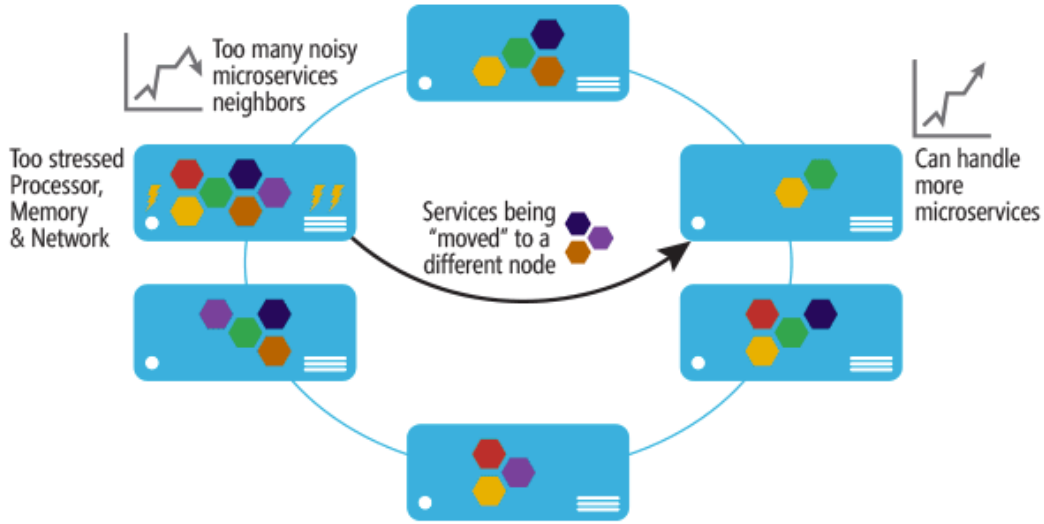
Deliver low latency and improved efficiency at massive scale

Deliver fast yet safe upgrades with zero downtime, automate scale operations, integrate health monitoring, and provide automatic recovery from failures. Orchestrate microservices and container-based apps, gain insight into application health and performance, and enable development of low-latency, resilient services that scale.

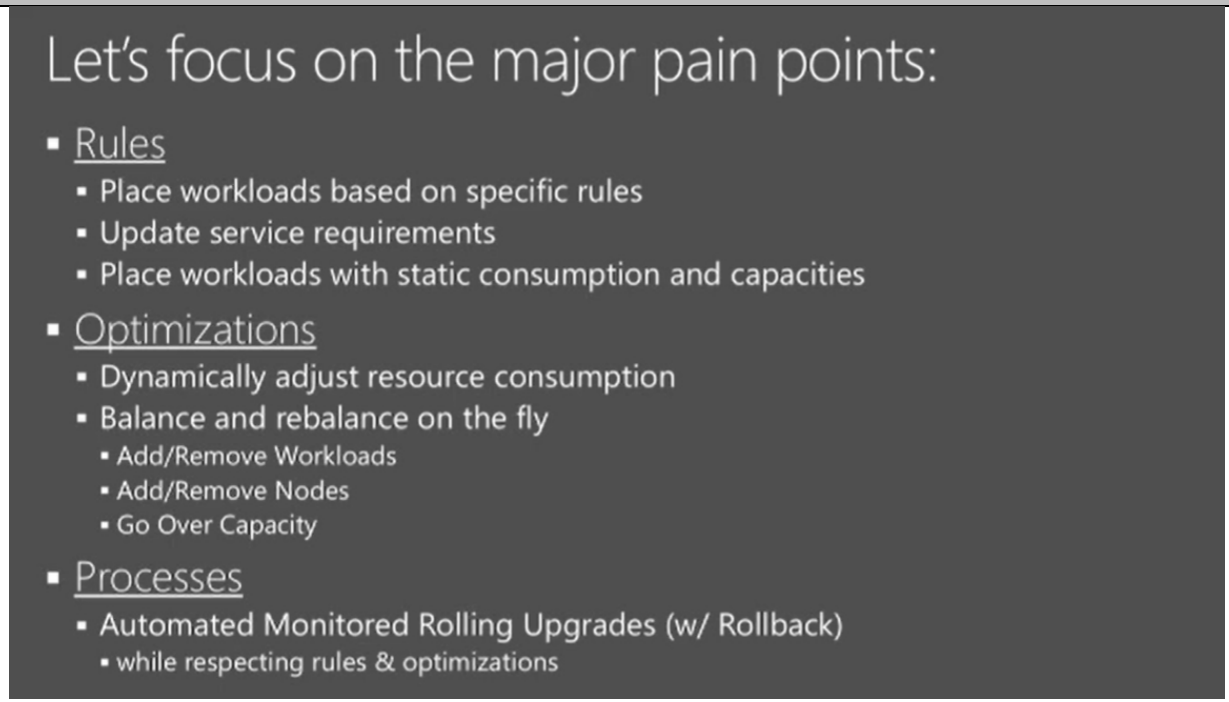
Solve difficult distributed systems problems such as reliable failure detection and failover, leader election, state management, service discovery, rolling upgrades, application portability, environment abstraction, resource management and governance, and configuration management, and provide application lifecycle management capabilities so developers don't have to re-architect applications as usage grows. Develop and deliver many kinds of applications and workloads, including multi-tenant SaaS applications, mission-critical line-of-business applications, IoT data gathering and processing workloads, and gaming.

https://azure.microsoft.com/en-us/products/service-fabric

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	# Connect and communicate with services in Service Fabric Article • 08/22/2024 • 18 contributors Feedback **In this article** Bring your own protocol Service discovery and resolution Connecting to other services Connections from external clients Show 4 more In Service Fabric, a service runs somewhere in a Service Fabric cluster, typically distributed across multiple VMs. It can be moved from one place to another, either by the service owner, or automatically by Service Fabric. Services are not statically tied to a particular machine or address. A Service Fabric application is generally composed of many different services, where each service performs a specialized task. These services may communicate with each other to form a complete function, such as rendering different parts of a web application. There are also client applications that connect to and communicate with services. This document discusses how to set up communication with and between your services in Service Fabric. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services
[1.Pre.2] wherein a configurable policy defines a mapping from a number of discrete resource types to a request peering type with a capacity allocation	Microsoft Azure Service Fabric features a configurable policy which defines a mapping from a number of discrete resource types to a request peering type with a capacity allocation for each responsibility in the request peering type. For instance, Azure Service fabric supports defined policies for resource balancing (or service orchestration) that map services based on a node's capacity (e.g., processor, memory, and network availability).

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
for each responsibility in the request peering type:	**Automatic Resource Balancing** Service Fabric provides service resource balancing (or service orchestration) that understands the total available resources in the cluster (like the compute from the VMs). It automatically moves the microservices depending on defined policies and constraints to best optimize how the created services are placed across the Vms (see **Figure 11**). This is focused on optimizing costs and performance.  **Figure 11 Cluster Showing Services Distribution Across the Nodes and Automatic Resource Balancing** https://learn.microsoft.com/en-us/archive/msdn-magazine/2015/december/microsoft-azure-azure-service-fabric-and-the-microservices-architecture

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	 https://learn.microsoft.com/en-us/shows/ignite-2015/brk3485 at 5:15.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	 https://learn.microsoft.com/en-us/shows/ignite-2015/brk3485 at 5:37.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Azure Service Fabric

Expand table

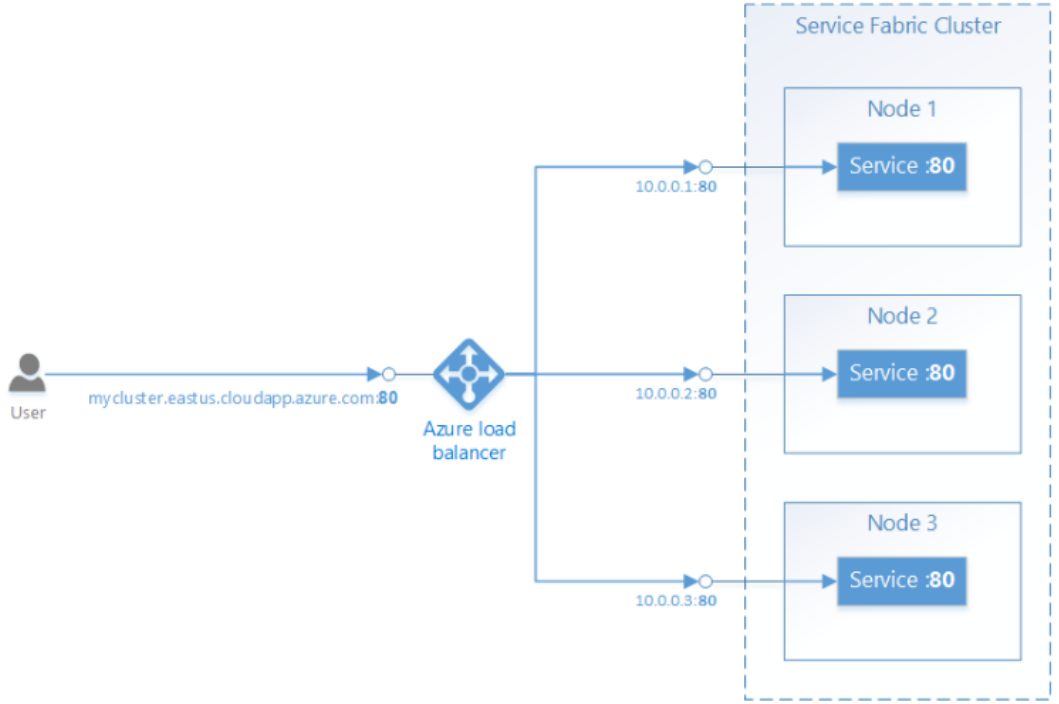
Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Service Fabric Clusters should be Zone Redundant	Service Fabric Clusters can be configured to be Zone Redundant or not. Servicefabric Clusters whose nodeType do not have the multipleAvailabilityZones set to true are not Zone Redundant. This policy identifies Servicefabric Clusters lacking the redundancy needed to withstand a zone outage.	Audit, Deny, Disabled	1.0.0-preview
Service Fabric clusters should have the ClusterProtectionLevel property set to EncryptAndSign	Service Fabric provides three levels of protection (None, Sign and EncryptAndSign) for node-to-node communication using a primary cluster certificate. Set the protection level to ensure that all node-to-node messages are encrypted and digitally signed	Audit, Deny, Disabled	1.1.0
Service Fabric clusters should only use Azure Active Directory for client authentication	Audit usage of client authentication only via Azure Active Directory in Service Fabric	Audit, Deny, Disabled	1.1.0

https://learn.microsoft.com/en-us/azure/service-fabric/policy-reference

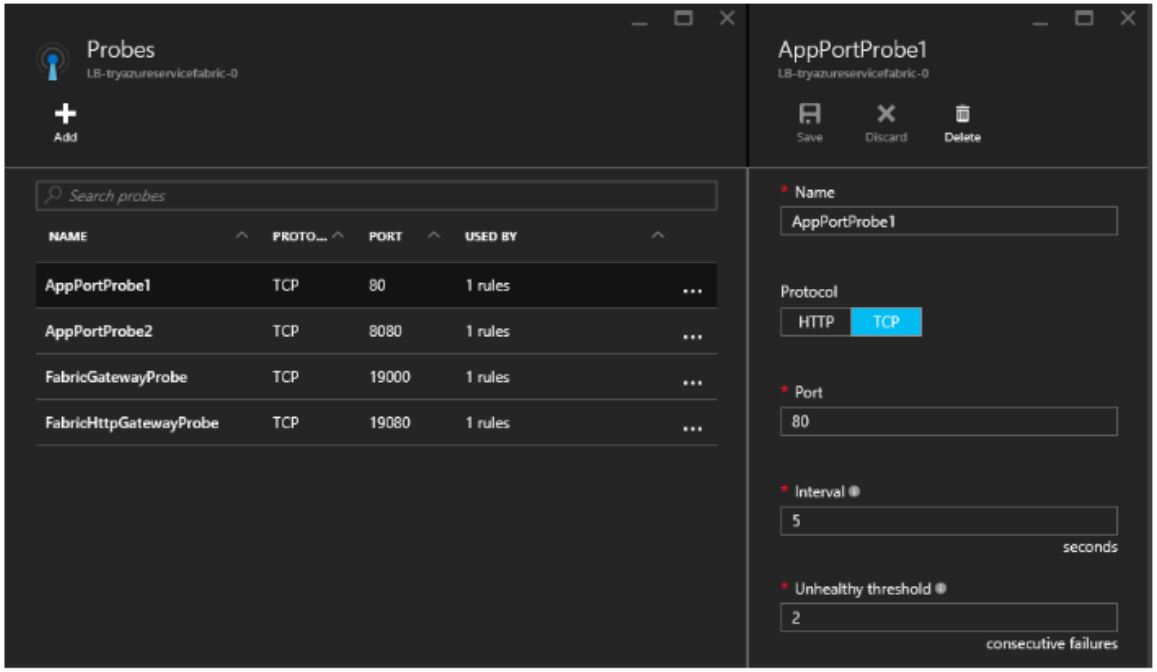
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service Fabric in Azure

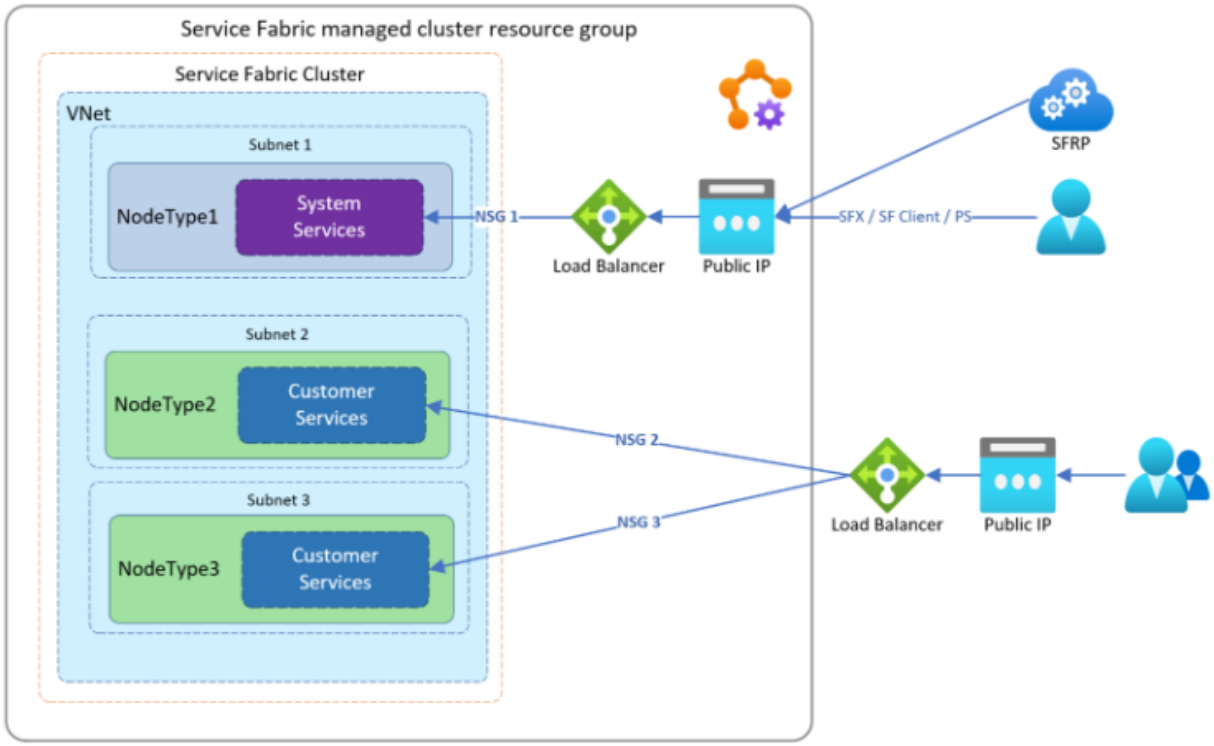
A Service Fabric cluster in Azure is placed behind an Azure Load Balancer. All external traffic to the cluster must pass through the load balancer. The load balancer will automatically forward traffic inbound on a given port to a random *node* that has the same port open. The Azure Load Balancer only knows about ports open on the *nodes*, it does not know about ports open by individual *services*.

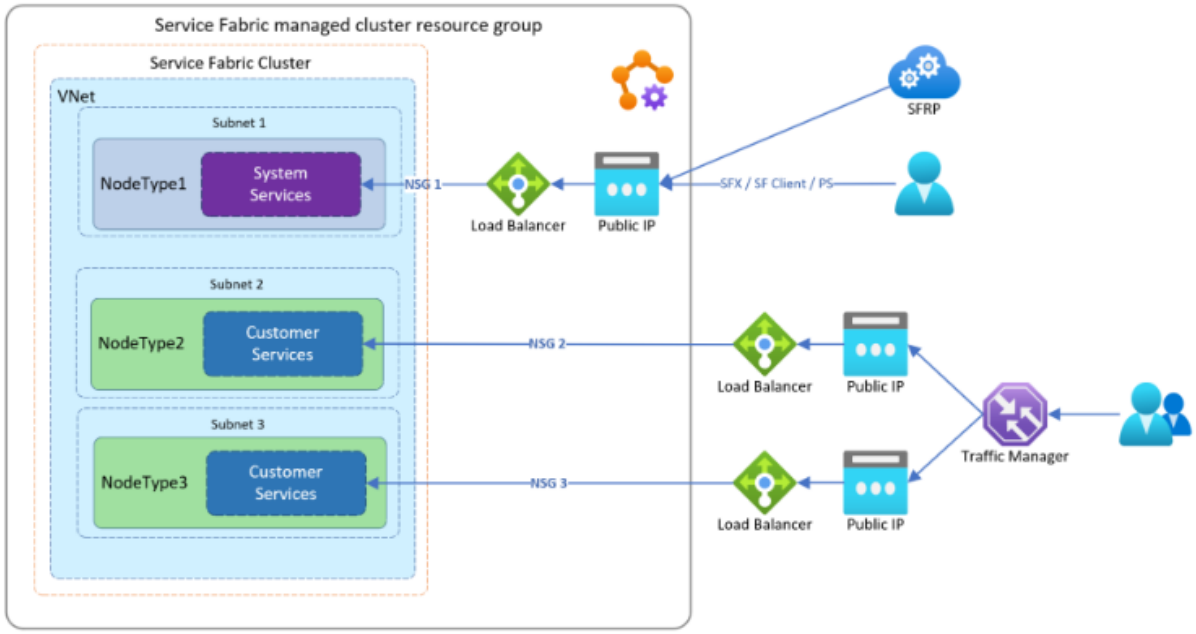


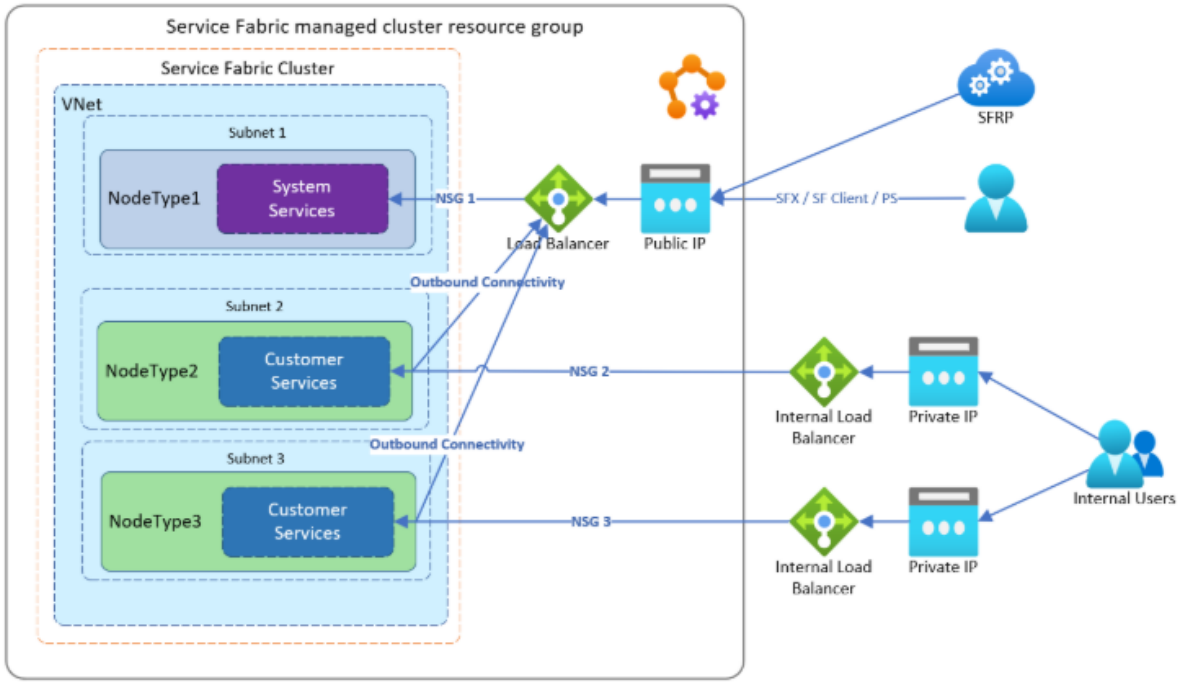
https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	4. The Azure Load Balancer uses a probe to determine whether or not to send traffic to a particular node. The probe periodically checks an endpoint on each node to determine whether or not the node is responding. If the probe fails to receive a response after a configured number of times, the load balancer stops sending traffic to that node. When creating a cluster through the Azure portal, a probe is automatically set up for each custom endpoint port that was configured.  https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	Bring your own Azure Load Balancer Managed clusters create an Azure public Standard Load Balancer and fully qualified domain name with a static public IP for both the primary and secondary node types. Bring your own load balancer allows you to use an existing Azure Load Balancer for secondary node types for both inbound and outbound traffic. When you bring your own Azure Load Balancer, you can: • Use a preconfigured Load Balancer static IP address for either private or public traffic • Map a Load Balancer to a specific node type • Configure network security group rules per node type because each node type is deployed in its own subnet • Maintain existing policies and controls you may have in place • Configure an internal-only load balancer and use the default load balancer for external traffic https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through an existing Azure Load Balancer configured with an existing static IP address to two node types.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through existing Azure Load Balancers to help them manage traffic flow to their applications independently that live on separate node types. When set up like this example, each node type will be behind its own managed NSG.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through existing internal Azure Load Balancers. This helps them manage traffic flow to their applications independently that live on separate node types. When set up like this example, each node type will be behind its own managed NSG and use the default load balancer for external traffic.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking
[21.a] (A) receiving, at said node, a request relating to a particular resource;	Microsoft receives, at said node, a request relating to a particular resource. For instance, a request for a resource (e.g., a service provided by an application) is received at an Azure Service Fabric cluster node.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Application and service concepts

Service Fabric Native Application: Service Fabric Native Applications are described by the Native Application Model (XML-based application and service manifests).

Service Fabric Native Application concepts

Application: An application is a collection of constituent services that perform a certain function or functions. The lifecycle of each application instance can be managed independently.

Service: A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service.

Application type: The name/version assigned to a collection of service types. It is defined in an `ApplicationManifest.xml` file and embedded in an application package directory. The directory is then copied to the Service Fabric cluster's image store. You can then create a named application from this application type within the cluster.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Connect and communicate with services in Service Fabric

Article • 08/22/2024 • 17 contributors

Feedback

In this article

Bring your own protocol
Service discovery and resolution
Connecting to other services
Connections from external clients
Show 4 more

In Service Fabric, a service runs somewhere in a Service Fabric cluster, typically distributed across multiple VMs. It can be moved from one place to another, either by the service owner, or automatically by Service Fabric. Services are not statically tied to a particular machine or address.

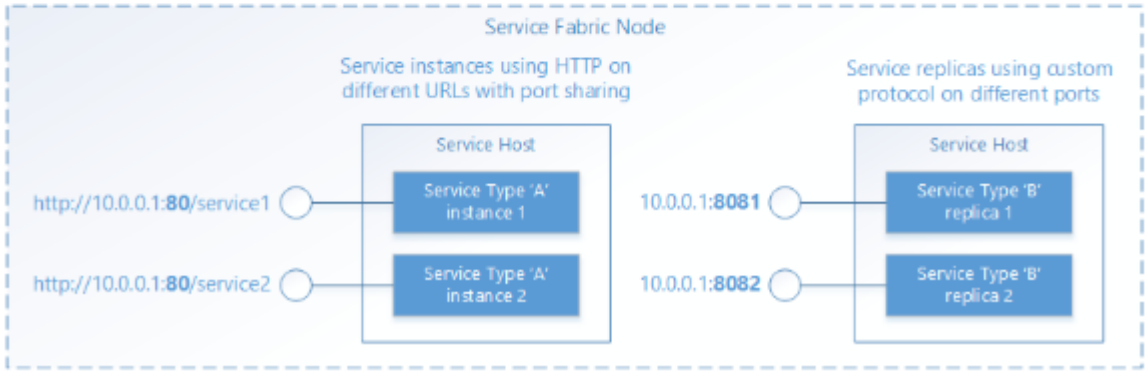
A Service Fabric application is generally composed of many different services, where each service performs a specialized task. These services may communicate with each other to form a complete function, such as rendering different parts of a web application. There are also client applications that connect to and communicate with services. This document discusses how to set up communication with and between your services in Service Fabric.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Bring your own protocol

Service Fabric helps manage the lifecycle of your services but it does not make decisions about what your services do. This includes communication. When your service is opened by Service Fabric, that's your service's opportunity to set up an endpoint for incoming requests, using whatever protocol or communication stack you want. Your service will listen on a normal **IP:port** address using any addressing scheme, such as a URI. Multiple service instances or replicas may share a host process, in which case they will either need to use different ports or use a port-sharing mechanism, such as the http.sys kernel driver in Windows. In either case, each service instance or replica in a host process must be uniquely addressable.

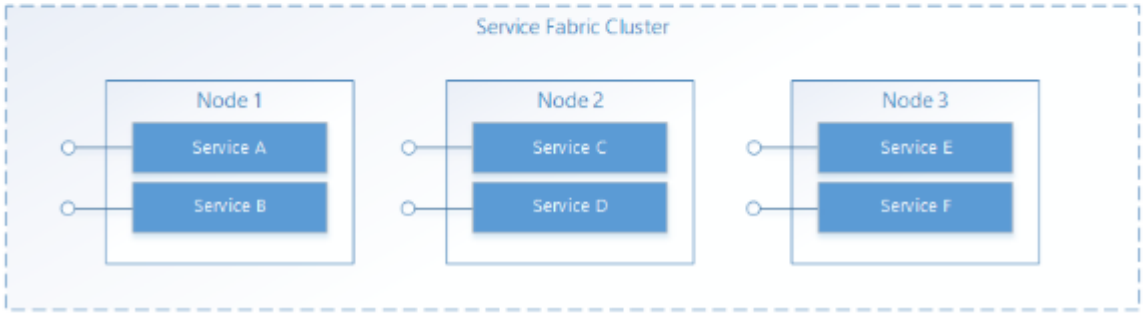


https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service discovery and resolution

In a distributed system, services may move from one machine to another over time. This can happen for various reasons, including resource balancing, upgrades, failovers, or scale-out. This means service endpoint addresses change as the service moves to nodes with different IP addresses, and may open on different ports if the service uses a dynamically selected port.



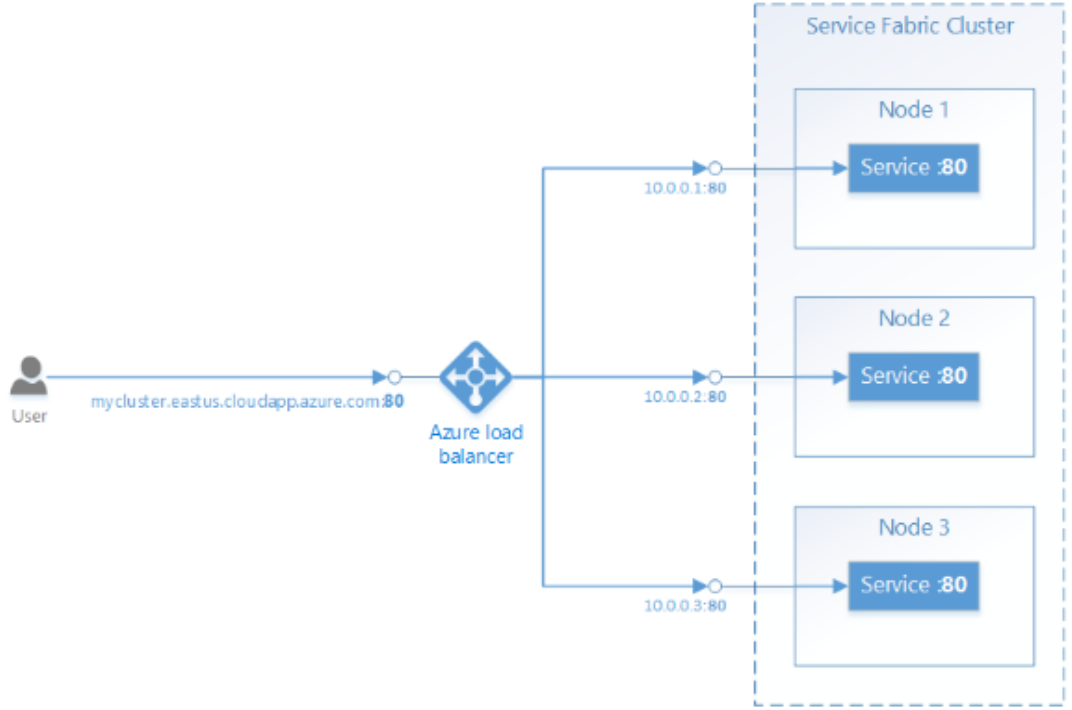
https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	Service Fabric provides a discovery and resolution service called the Naming Service. The Naming Service maintains a table that maps named service instances to the endpoint addresses they listen on. All named service instances in Service Fabric have unique names represented as URIs, for example, `"fabric:/MyApplication/MyService"`. The name of the service does not change over the lifetime of the service, it's only the endpoint addresses that can change when services move. This is analogous to websites that have constant URLs but where the IP address may change. And similar to DNS on the web, which resolves website URLs to IP addresses, Service Fabric has a registrar that maps service names to their endpoint address.  https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service Fabric in Azure

A Service Fabric cluster in Azure is placed behind an Azure Load Balancer. All external traffic to the cluster must pass through the load balancer. The load balancer will automatically forward traffic inbound on a given port to a random *node* that has the same port open. The Azure Load Balancer only knows about ports open on the *nodes*, it does not know about ports open by individual *services*.



https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Reliable Services: Built-in communication API options

The Reliable Services framework ships with several pre-built communication options. The decision about which one will work best for you depends on the choice of the programming model, the communication framework, and the programming language that your services are written in.

- **No specific protocol**: If you don't have a particular choice of communication framework, but you want to get something up and running quickly, then the ideal option for you is service remoting, which allows strongly-typed remote procedure calls for Reliable Services and Reliable Actors. This is the easiest and fastest way to get started with service communication. Service remoting handles resolution of service addresses, connection, retry, and error handling. This is available for both C# and Java applications.
- **HTTP**: For language-agnostic communication, HTTP provides an industry-standard choice with tools and HTTP servers available in many different languages, all supported by Service Fabric. Services can use any HTTP stack available, including ASP.NET Web API for C# applications. Clients written in C# can leverage the `ICommunicationClient` and `ServicePartitionClient` classes, whereas for Java, use the `CommunicationClient` and `FabricServicePartitionClient` classes, for service resolution, HTTP connections, and retry loops.
- **WCF**: If you have existing code that uses WCF as your communication framework, then you can use the `WcfCommunicationListener` for the server side and `WcfCommunicationClient` and `ServicePartitionClient` classes for the client. This however is only available for C# applications on Windows based clusters. For more details, see this article about WCF-based implementation of the communication stack.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Azure Service Fabric

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Service Fabric Clusters should be Zone Redundant	Service Fabric Clusters can be configured to be Zone Redundant or not. Servicefabric Clusters whose nodeType do not have the multipleAvailabilityZones set to true are not Zone Redundant. This policy identifies Servicefabric Clusters lacking the redundancy needed to withstand a zone outage.	Audit, Deny, Disabled	1.0.0-preview
Service Fabric clusters should have the ClusterProtectionLevel property set to EncryptAndSign	Service Fabric provides three levels of protection (None, Sign and EncryptAndSign) for node-to-node communication using a primary cluster certificate. Set the protection level to ensure that all node-to-node messages are encrypted and digitally signed	Audit, Deny, Disabled	1.1.0
Service Fabric clusters should only use Azure Active Directory for client authentication	Audit usage of client authentication only via Azure Active Directory in Service Fabric	Audit, Deny, Disabled	1.1.0

https://learn.microsoft.com/en-us/azure/service-fabric/policy-reference

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
[1.b] (B) determining what kind of responsibility said node has for said request based on at least one kind of responsibility assigned to said node for requests relating to said particular resource, and (ii) said mapping; and	Microsoft determines what kind of responsibility said node has for said request based on at least one kind of responsibility assigned to said node for requests relating to said particular resource, and (ii) said mapping. For instance, Azure Service Fabric nodes may be responsible for system services (e.g., name resolution, service discovery), application hosting (e.g., handling client requests), primary and non-primary workloads, resource management (e.g., scaling, load balancing), or health monitoring. Determining what kind of responsibility the node in the cluster assumes is based on the kind of responsibility assigned to a node for requests relating to a particular resource and the mapping in 1.Pre.2 above.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Application and service concepts

Service Fabric Native Application: Service Fabric Native Applications are described by the Native Application Model (XML-based application and service manifests).

Service Fabric Native Application concepts

Application: An application is a collection of constituent services that perform a certain function or functions. The lifecycle of each application instance can be managed independently.

Service: A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service.

Application type: The name/version assigned to a collection of service types. It is defined in an `ApplicationManifest.xml` file and embedded in an application package directory. The directory is then copied to the Service Fabric cluster's image store. You can then create a named application from this application type within the cluster.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service type: The name/version assigned to a service's code packages, data packages, and configuration packages. The service type is defined in the `ServiceManifest.xml` file and embedded in a service package directory. The service package directory is then referenced by an application package's `ApplicationManifest.xml` file. Within the cluster, after creating a named application, you can create a named service from one of the application type's service types. The service type's `ServiceManifest.xml` file describes the service.

Read the Application model article for more information.

There are two types of services:

- **Stateless**: Use a stateless service when the service's persistent state is stored in an external storage service, such as Azure Storage, Azure SQL Database, or Azure Cosmos DB. Use a stateless service when the service has no persistent storage. For example, for a calculator service where values are passed to the service, a computation is performed that uses these values, and then a result is returned.
- **Stateful**: Use a stateful service when you want Service Fabric to manage your service's state via its Reliable Collections or Reliable Actors programming models. When you create a named service, specify how many partitions you want to spread your state over for scalability. Also specify how many times to replicate your state across nodes, for reliability. Each named service has a single primary replica and multiple secondary replicas. You modify your named service's state when you write to the primary replica. Service Fabric then replicates this state to all the secondary replicas to keep your state in sync. Service Fabric automatically detects when a primary replica fails and promotes an existing secondary replica to a primary replica. Service Fabric then creates a new secondary replica.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

System services

There are system services that are created in every cluster that provide the platform capabilities of Service Fabric.

Naming Service: Each Service Fabric cluster has a Naming Service, which resolves service names to a location in the cluster. You manage the service names and properties, like an internet Domain Name System (DNS) for the cluster. Clients securely communicate with any node in the cluster by using the Naming Service to resolve a service name and its location. Applications move within the cluster. For example, this can be due to failures, resource balancing, or the resizing of the cluster. You can develop services and clients that resolve the current network location. Clients obtain the actual machine IP address and port where it's currently running.

Read Communicate with services for more information on the client and service communication APIs that work with the Naming Service.

Image Store service: Each Service Fabric cluster has an Image Store service where deployed, versioned application packages are kept. Copy an application package to the Image Store and then register the application type contained within that application package. After the application type is provisioned, you create a named application from it. You can unregister an application type from the Image Store service after all its named applications have been deleted.

Read Understand the ImageStoreConnectionString setting for more information about the Image Store service.

Read the Deploy an application article for more information on deploying applications to the Image Store service.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	Failover Manager service: Each Service Fabric cluster has a Failover Manager service that is responsible for the following actions: • Performs functions related to high availability and consistency of services. • Orchestrates application and cluster upgrades. • Interacts with other system components. **Repair Manager service**: This is an optional system service that allows repair actions to be performed on a cluster in a way that is safe, automatable, and transparent. Repair manager is used in: • Performing Azure maintenance repairs on Silver and Gold durability Azure Service Fabric clusters. • Carrying out repair actions for Patch Orchestration Application https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Deployment and application models

To deploy your services, you need to describe how they should run. Service Fabric supports three different deployment models:

Native model

The native application model provides your applications with full low-level access to Service Fabric. Applications and services are defined as registered types in XML manifest files.

The native model supports the Reliable Services and Reliable Actors frameworks, which provides access to the Service Fabric runtime APIs and cluster management APIs in C# and Java. The native model also supports arbitrary containers and executables.

Reliable Services: An API to build stateless and stateful services. Stateful services store their state in Reliable Collections, such as a dictionary or a queue. You can also plug in various communication stacks, such as Web API and Windows Communication Foundation (WCF).

Reliable Actors: An API to build stateless and stateful objects through the virtual Actor programming model. This model is useful when you have lots of independent units of computation or state. This model uses a turn-based threading model, so it's best to avoid code that calls out to other actors or services because an individual actor can't process other incoming requests until all its outbound requests are finished.

You can also run your existing applications on Service Fabric:

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	Containers: Service Fabric supports the deployment of Docker containers on Linux and Windows Server containers on Windows Server 2016, along with support for Hyper-V isolation mode. In the Service Fabric application model, a container represents an application host in which multiple service replicas are placed. Service Fabric can run any containers, and the scenario is similar to the guest executable scenario, where you package an existing application inside a container. In addition, you can run Service Fabric services inside containers as well. **Guest executables**: You can run any type of code, such as Node.js, Python, Java, or C++ in Azure Service Fabric as a service. Service Fabric refers to these types of services as guest executables, which are treated as stateless services. The advantages to running a guest executable in a Service Fabric cluster include high availability, health monitoring, application lifecycle management, high density, and discoverability. Read the Choose a programming model for your service article for more information. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Overview of Service Fabric Standalone clusters

Article • 08/22/2024 • 7 contributors

Feedback

In this article

Cluster security
Scaling
Upgrading
Supported operating systems
Next steps

A Service Fabric cluster is a network-connected set of virtual or physical machines into which your microservices are deployed and managed. A machine or VM that is part of a cluster is called a cluster node. Clusters can scale to thousands of nodes. If you add new nodes to the cluster, Service Fabric rebalances the service partition replicas and instances across the increased number of nodes. Overall application performance improves and contention for access to memory decreases. If the nodes in the cluster are not being used efficiently, you can decrease the number of nodes in the cluster. Service Fabric again rebalances the partition replicas and instances across the decreased number of nodes to make better use of the hardware on each node.

A node type defines the size, number, and properties for a set of nodes in the cluster. Each node type can then be scaled up or down independently, have different sets of ports open, and can have different capacity metrics. Node types are used to define roles for a set of cluster nodes, such as "front end" or "back end". Your cluster can have more than one node type, but the primary node type must have at least five VMs for production clusters (or at least three VMs for test clusters). Service Fabric system services are placed on the nodes of the primary node type.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Azure Service Fabric

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Service Fabric Clusters should be Zone Redundant	Service Fabric Clusters can be configured to be Zone Redundant or not. Servicefabric Clusters whose nodeType do not have the multipleAvailabilityZones set to true are not Zone Redundant. This policy identifies Servicefabric Clusters lacking the redundancy needed to withstand a zone outage.	Audit, Deny, Disabled	1.0.0-preview
Service Fabric clusters should have the ClusterProtectionLevel property set to EncryptAndSign	Service Fabric provides three levels of protection (None, Sign and EncryptAndSign) for node-to-node communication using a primary cluster certificate. Set the protection level to ensure that all node-to-node messages are encrypted and digitally signed	Audit, Deny, Disabled	1.1.0
Service Fabric clusters should only use Azure Active Directory for client authentication	Audit usage of client authentication only via Azure Active Directory in Service Fabric	Audit, Deny, Disabled	1.1.0

https://learn.microsoft.com/en-us/azure/service-fabric/policy-reference

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

To configure with your own load balancer:

1. Get the service `Id` from your subscription for Service Fabric Resource Provider application:

PowerShell Copy

```
Login-AzAccount
Select-AzSubscription -SubscriptionId <SubId>
Get-AzADServicePrincipal -DisplayName "Azure Service Fabric Resource Provider"
```

> **Note**
>
> Make sure you are in the correct subscription, the principal ID will change if the subscription is in a different tenant.

PowerShell Copy

```
ServicePrincipalNames : {00001111-aaaa-2222-bbbb-3333cccc4444}
ApplicationId         : 00001111-aaaa-2222-bbbb-3333cccc4444
ObjectType            : ServicePrincipal
DisplayName           : Azure Service Fabric Resource Provider
Id                    : 00000000-0000-0000-0000-000000000000
```

Note the **Id** of the previous output as **principalId** for use in a later step

Expand table

Role definition name	Role definition ID
Network Contributor	4d97b98b-1d4f-4787-a291-c67834d212e7

Note the `Role definition name` and `Role definition ID` property values for use in a later step

https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

	2. Add a role assignment to the Service Fabric Resource Provider application. Adding a role assignment is a one time action. You add the role by running the following PowerShell commands or by configuring an Azure Resource Manager (ARM) template as detailed below.
	
	In the following steps, we start with an existing load balancer named Existing-LoadBalancer1, in the Existing-RG resource group.
	
	Obtain the required `Id` property info from the existing Azure Load Balancer.
	
	PowerShell Copy
	`Login-AzAccount`
	`Select-AzSubscription -SubscriptionId <SubId>`
	`Get-AzLoadBalancer -Name "Existing-LoadBalancer1" -ResourceGroupName "Existing-RG"`
	
	https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

3. Configure required outbound connectivity for the node type. You must configure a public load balancer to provide outbound connectivity or use the default public load balancer.

   Configure `outboundRules` to configure a public load balancer to provide outbound connectivity See the create load balancer and assign role sample Azure Resource Manager (ARM) template

   OR

   To configure the node type to use the default load balancer set the following in your template:

   - The Service Fabric managed cluster resource apiVersion should be **2022-01-01** or later.

JSON Copy

```
"resources": [
  {
  "apiVersion": "[variables('sfApiVersion')]",
  "type": "Microsoft.ServiceFabric/managedclusters/nodetypes",
  "properties": {
      "isPrimary": false,
      "useDefaultPublicLoadBalancer": true
      }
  }
]
```

https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

4. Optionally configure an inbound application port and related probe on your existing Azure Load Balancer. See the bring your own load balancer sample Azure Resource Manager (ARM) template for an example

5. Optionally configure the managed cluster NSG rules applied to the node type to allow any required traffic that you've configured on the Azure Load Balancer or traffic will be blocked. See the bring your own load balancer sample Azure Resource Manager (ARM) template for an example inbound NSG rule configuration. In the template, look for the `networkSecurityRules` property.

6. Deploy the configured managed cluster ARM Template For this step we'll use the bring your own load balancer sample Azure Resource Manager (ARM) template

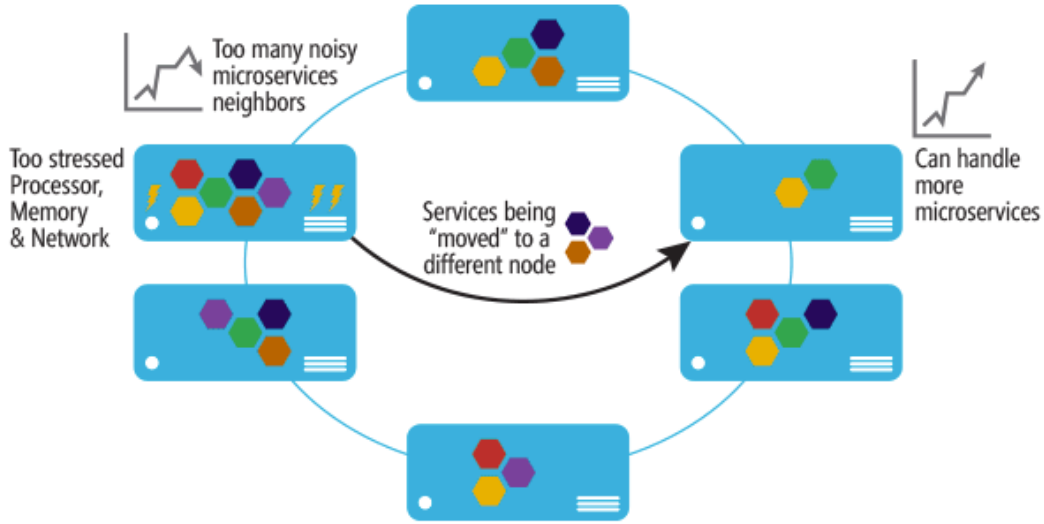
   The following will create a resource group called `MyResourceGroup` in `westus` and deploy a cluster using an existing load balancer.

```
PowerShell                                                    Copy

New-AzResourceGroup -Name MyResourceGroup -Location westus
New-AzResourceGroupDeployment -Name deployment -ResourceGroupName MyResourceGroup -Templ
```

   After deployment, the secondary node type is configured to use the specified load balancer for inbound and outbound traffic. The Service Fabric client connection and gateway endpoints will still point to the public DNS of the managed cluster primary node type static IP address.

https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	Automatic Resource Balancing Service Fabric provides service resource balancing (or service orchestration) that understands the total available resources in the cluster (like the compute from the VMs). It automatically moves the microservices depending on defined policies and constraints to best optimize how the created services are placed across the Vms (see **Figure 11**). This is focused on optimizing costs and performance.  **Figure 11 Cluster Showing Services Distribution Across the Nodes and Automatic Resource Balancing** https://learn.microsoft.com/en-us/archive/msdn-magazine/2015/december/microsoft-azure-azure-service-fabric-and-the-microservices-architecture
[1.c] (C) based on said determining in (B), attempting to process said request in accordance with the type of said particular resource and the kind of	Based on said determining in (B), Microsoft attempts to process said request in accordance with the type of said particular resource and the kind of responsibility said node has for said particular resource. For instance, Azure Service Fabric nodes may process requests in accordance with the type of resource and kind of responsibility the node has for said resource. For instance, Azure Service Fabric nodes may be responsible for system services (e.g., name resolution, service discovery), application hosting (e.g., handling client requests), primary and non-primary workloads, resource management (e.g., scaling, load balancing), or health monitoring.

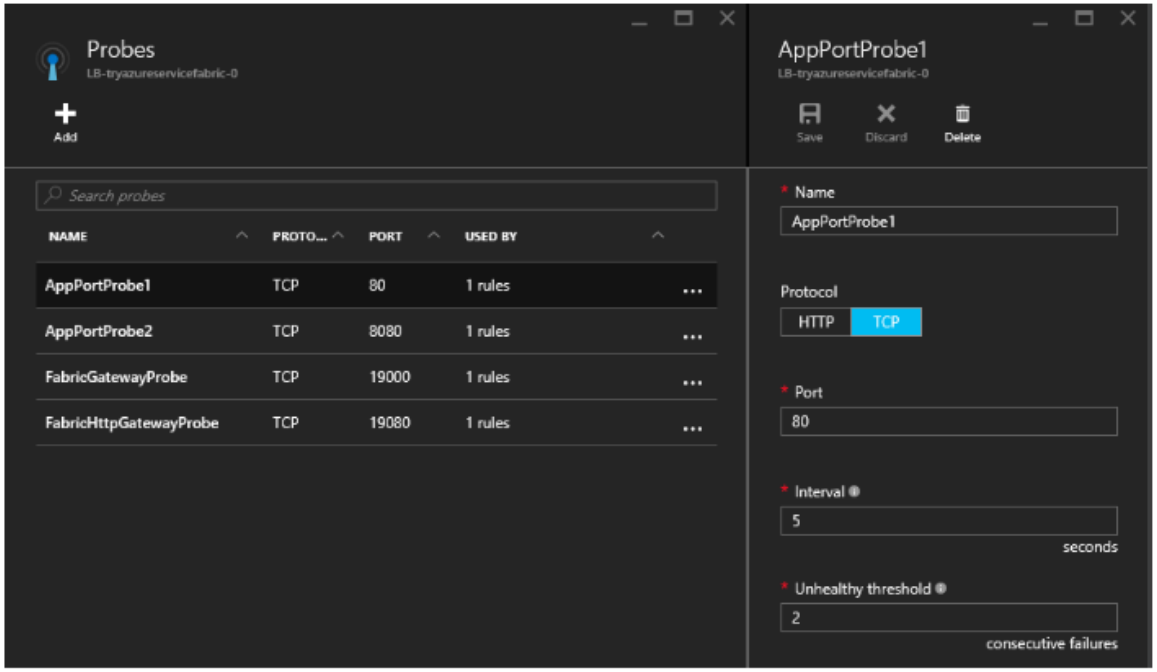
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
responsibility said node has for said particular resource.	

Azure Service Fabric

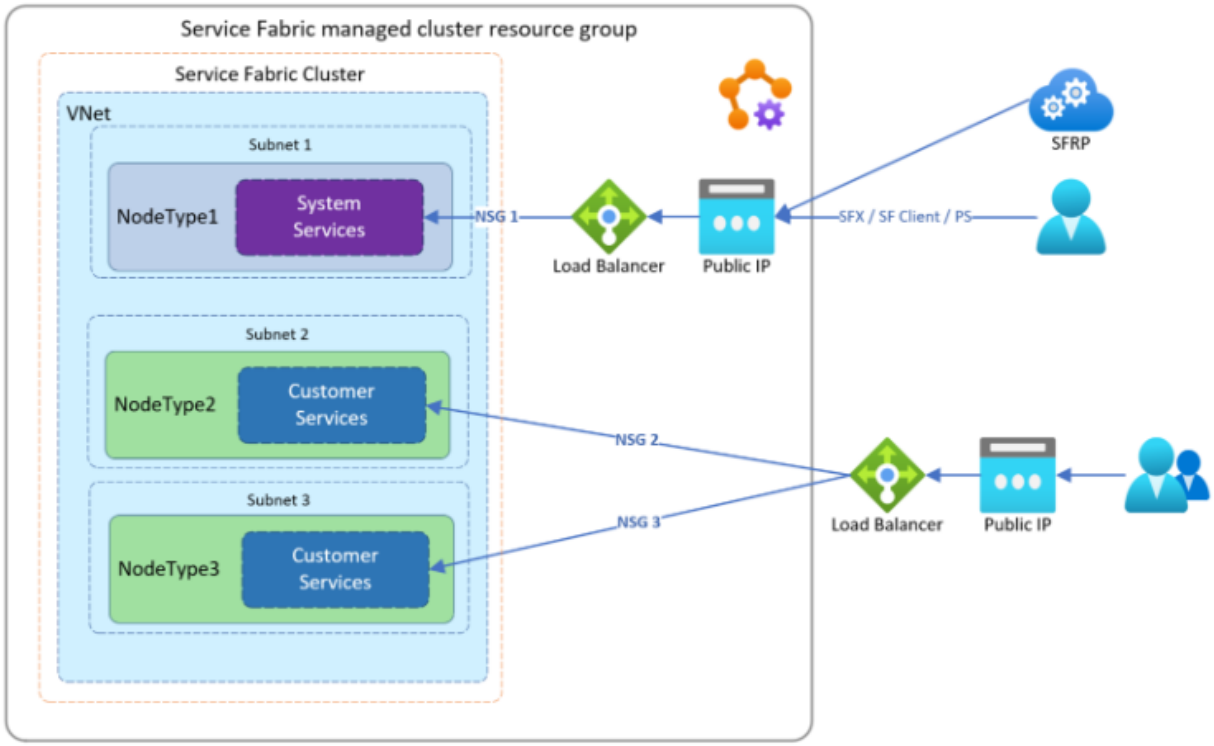
Expand table

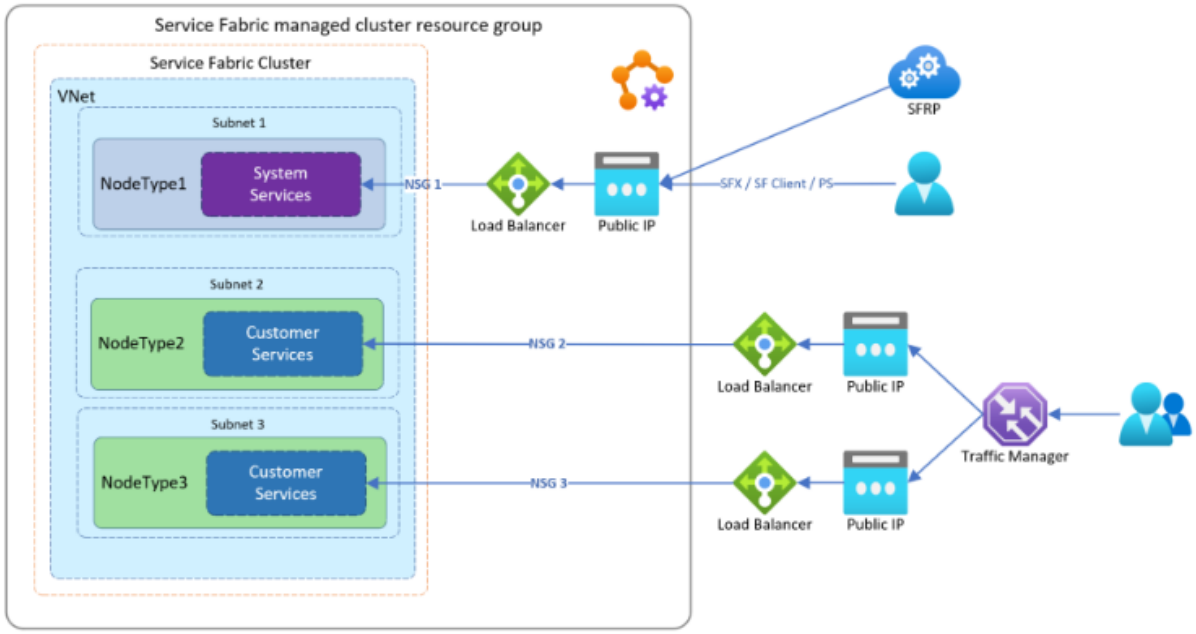
Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Service Fabric Clusters should be Zone Redundant	Service Fabric Clusters can be configured to be Zone Redundant or not. Servicefabric Clusters whose nodeType do not have the multipleAvailabilityZones set to true are not Zone Redundant. This policy identifies Servicefabric Clusters lacking the redundancy needed to withstand a zone outage.	Audit, Deny, Disabled	1.0.0-preview
Service Fabric clusters should have the ClusterProtectionLevel property set to EncryptAndSign	Service Fabric provides three levels of protection (None, Sign and EncryptAndSign) for node-to-node communication using a primary cluster certificate. Set the protection level to ensure that all node-to-node messages are encrypted and digitally signed	Audit, Deny, Disabled	1.1.0
Service Fabric clusters should only use Azure Active Directory for client authentication	Audit usage of client authentication only via Azure Active Directory in Service Fabric	Audit, Deny, Disabled	1.1.0

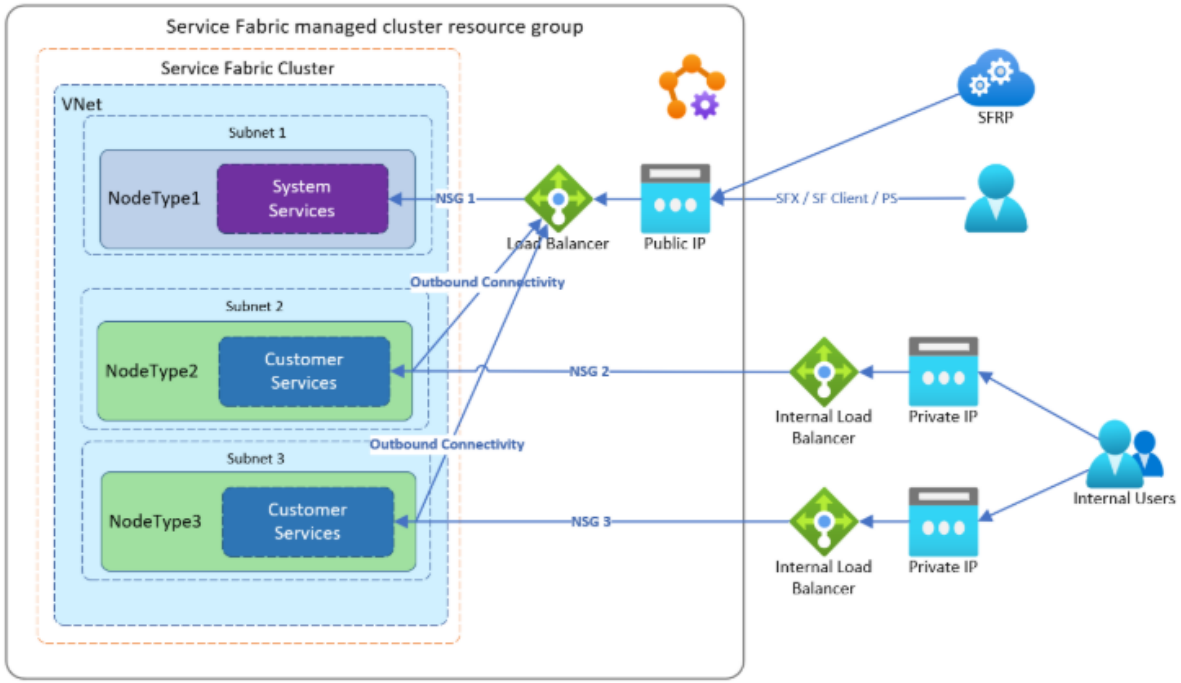
https://learn.microsoft.com/en-us/azure/service-fabric/policy-reference

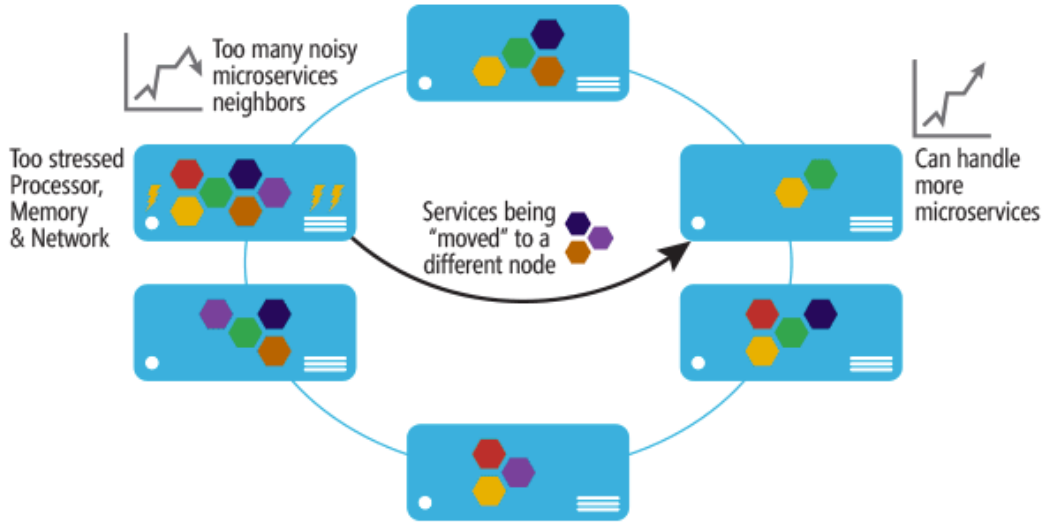
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	4. The Azure Load Balancer uses a probe to determine whether or not to send traffic to a particular node. The probe periodically checks an endpoint on each node to determine whether or not the node is responding. If the probe fails to receive a response after a configured number of times, the load balancer stops sending traffic to that node. When creating a cluster through the Azure portal, a probe is automatically set up for each custom endpoint port that was configured.  https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	# Bring your own Azure Load Balancer Managed clusters create an Azure public Standard Load Balancer and fully qualified domain name with a static public IP for both the primary and secondary node types. Bring your own load balancer allows you to use an existing Azure Load Balancer for secondary node types for both inbound and outbound traffic. When you bring your own Azure Load Balancer, you can: • Use a preconfigured Load Balancer static IP address for either private or public traffic • Map a Load Balancer to a specific node type • Configure network security group rules per node type because each node type is deployed in its own subnet • Maintain existing policies and controls you may have in place • Configure an internal-only load balancer and use the default load balancer for external traffic https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through an existing Azure Load Balancer configured with an existing static IP address to two node types.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through existing Azure Load Balancers to help them manage traffic flow to their applications independently that live on separate node types. When set up like this example, each node type will be behind its own managed NSG.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

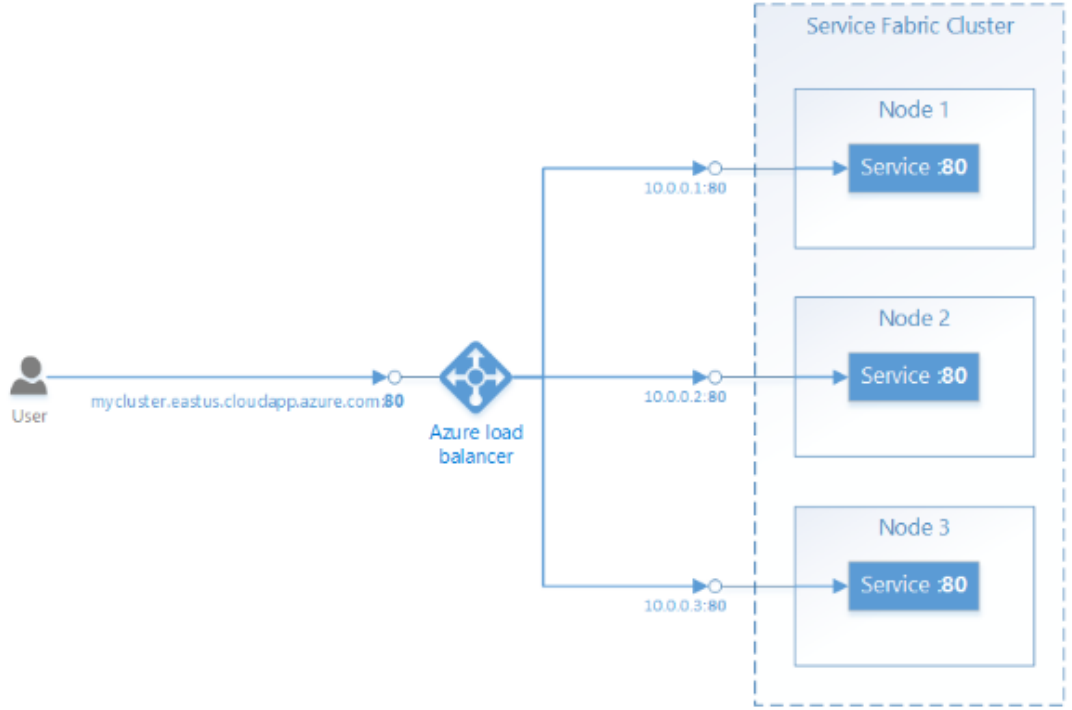
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through existing internal Azure Load Balancers. This helps them manage traffic flow to their applications independently that live on separate node types. When set up like this example, each node type will be behind its own managed NSG and use the default load balancer for external traffic.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	Automatic Resource Balancing Service Fabric provides service resource balancing (or service orchestration) that understands the total available resources in the cluster (like the compute from the VMs). It automatically moves the microservices depending on defined policies and constraints to best optimize how the created services are placed across the Vms (see **Figure 11**). This is focused on optimizing costs and performance.  **Figure 11 Cluster Showing Services Distribution Across the Nodes and Automatic Resource Balancing** https://learn.microsoft.com/en-us/archive/msdn-magazine/2015/december/microsoft-azure-azure-service-fabric-and-the-microservices-architecture
2. The method of claim 1 wherein the group of said multiple service endpoints comprise a peer group.	Microsoft performs the method of claim 1 wherein the group of said multiple service endpoints comprise a peer group. For instance, Microsoft Azure Service Fabric clusters comprise groups of peer nodes.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service Fabric in Azure

A Service Fabric cluster in Azure is placed behind an Azure Load Balancer. All external traffic to the cluster must pass through the load balancer. The load balancer will automatically forward traffic inbound on a given port to a random *node* that has the same port open. The Azure Load Balancer only knows about ports open on the *nodes*, it does not know about ports open by individual *services*.



https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	As previously described, the Service Fabric runtime distinguishes between its role as the peer of other nodes in the cluster, and as the server for cluster management operations. In principle, these settings can be configured distinctly, but in practice they tend to align. For the remainder of this article, we'll assume the settings match for simplicity. https://learn.microsoft.com/en-us/azure/service-fabric/cluster-security-certificates As another example, Service Cluster nodes may belong to network security groups (NSGs) in which virtual machines may communicate to one another as peers. **Network security groups** Network security groups (NSGs) control inbound and outbound traffic of a subnet, VM, or specific NIC. By default, when multiple VMs are put on the same virtual network they can communicate with each other through any port. If you want to constrain communications among the machines you can define NSGs to segment the network or isolate VMs from each other. If you have multiple node types in a cluster, you can apply NSGs to subnets to prevent machines belonging to different node types from communicating with each other. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-azure-clusters-overview
5. The method of claim 1 wherein each resource has a resource type selected from a discrete set of resource types.	Microsoft performs the method of claim 1 wherein each resource has a resource type selected from a discrete set of resource types. For instance, Microsoft Azure Service Fabric resources (e.g., services) have defined service types as well as application types.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Application and service concepts

Service Fabric Native Application: Service Fabric Native Applications are described by the Native Application Model (XML-based application and service manifests).

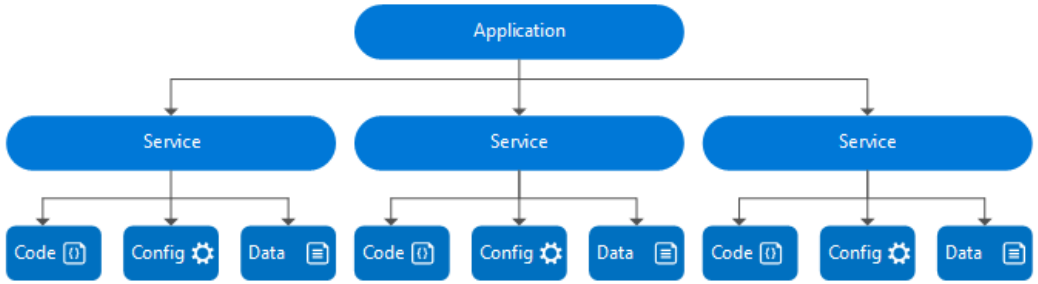
Service Fabric Native Application concepts

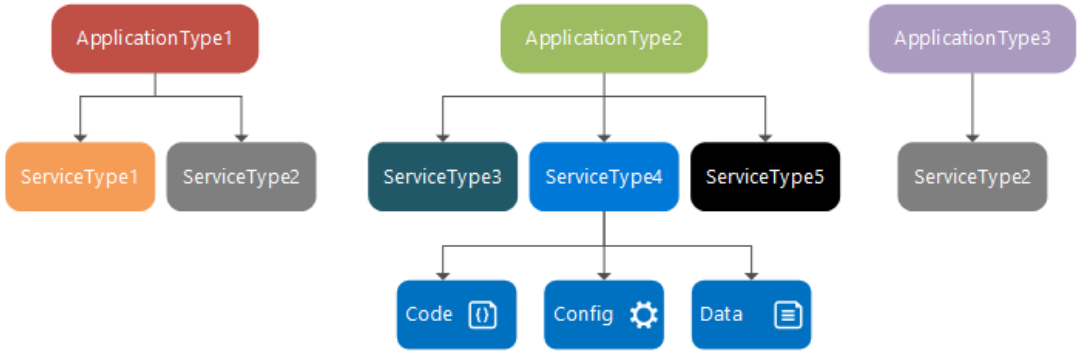
Application: An application is a collection of constituent services that perform a certain function or functions. The lifecycle of each application instance can be managed independently.

Service: A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service.

Application type: The name/version assigned to a collection of service types. It is defined in an `ApplicationManifest.xml` file and embedded in an application package directory. The directory is then copied to the Service Fabric cluster's image store. You can then create a named application from this application type within the cluster.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	An application is a collection of constituent services that perform a certain function or functions. A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service. Each component in this hierarchical application model can be versioned and upgraded independently.  An application type is a categorization of an application and consists of a bundle of service types. A service type is a categorization of a service. The categorization can have different settings and configurations, but the core functionality remains the same. The instances of a service are the different service configuration variations of the same service type. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-application-model

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	The code for different application instances runs as separate processes even when hosted by the same Service Fabric node. Furthermore, the lifecycle of each application instance can be managed (for example, upgraded) independently. The following diagram shows how application types are composed of service types, which in turn are composed of code, configuration, and data packages. To simplify the diagram, only the code/config/data packages for `ServiceType4` are shown, though each service type would include some or all those package types.  https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-application-model
6. The method of claim 1 wherein each node of said group is assigned at least one kind of responsibility for certain resources.	Microsoft performs the method of claim 1 wherein each node of said group is assigned at least one kind of responsibility for certain resources. For instance, Microsoft Service Fabric nodes are assigned to at least one kind of responsibility for certain resources (e.g., name resolution, service discovery), application hosting (e.g., handling client requests), primary and non-primary workloads, resource management (e.g., scaling, load balancing), or health monitoring.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Application and service concepts

Service Fabric Native Application: Service Fabric Native Applications are described by the Native Application Model (XML-based application and service manifests).

Service Fabric Native Application concepts

Application: An application is a collection of constituent services that perform a certain function or functions. The lifecycle of each application instance can be managed independently.

Service: A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service.

Application type: The name/version assigned to a collection of service types. It is defined in an `ApplicationManifest.xml` file and embedded in an application package directory. The directory is then copied to the Service Fabric cluster's image store. You can then create a named application from this application type within the cluster.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service type: The name/version assigned to a service's code packages, data packages, and configuration packages. The service type is defined in the `ServiceManifest.xml` file and embedded in a service package directory. The service package directory is then referenced by an application package's `ApplicationManifest.xml` file. Within the cluster, after creating a named application, you can create a named service from one of the application type's service types. The service type's `ServiceManifest.xml` file describes the service.

Read the Application model article for more information.

There are two types of services:

- **Stateless**: Use a stateless service when the service's persistent state is stored in an external storage service, such as Azure Storage, Azure SQL Database, or Azure Cosmos DB. Use a stateless service when the service has no persistent storage. For example, for a calculator service where values are passed to the service, a computation is performed that uses these values, and then a result is returned.
- **Stateful**: Use a stateful service when you want Service Fabric to manage your service's state via its Reliable Collections or Reliable Actors programming models. When you create a named service, specify how many partitions you want to spread your state over for scalability. Also specify how many times to replicate your state across nodes, for reliability. Each named service has a single primary replica and multiple secondary replicas. You modify your named service's state when you write to the primary replica. Service Fabric then replicates this state to all the secondary replicas to keep your state in sync. Service Fabric automatically detects when a primary replica fails and promotes an existing secondary replica to a primary replica. Service Fabric then creates a new secondary replica.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

System services

There are system services that are created in every cluster that provide the platform capabilities of Service Fabric.

Naming Service: Each Service Fabric cluster has a Naming Service, which resolves service names to a location in the cluster. You manage the service names and properties, like an internet Domain Name System (DNS) for the cluster. Clients securely communicate with any node in the cluster by using the Naming Service to resolve a service name and its location. Applications move within the cluster. For example, this can be due to failures, resource balancing, or the resizing of the cluster. You can develop services and clients that resolve the current network location. Clients obtain the actual machine IP address and port where it's currently running.

Read Communicate with services for more information on the client and service communication APIs that work with the Naming Service.

Image Store service: Each Service Fabric cluster has an Image Store service where deployed, versioned application packages are kept. Copy an application package to the Image Store and then register the application type contained within that application package. After the application type is provisioned, you create a named application from it. You can unregister an application type from the Image Store service after all its named applications have been deleted.

Read Understand the ImageStoreConnectionString setting for more information about the Image Store service.

Read the Deploy an application article for more information on deploying applications to the Image Store service.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	See also Claim limitation 1.b.
7. The method of claim 6 wherein the at least one kind of responsibility is selected from a discrete set of kinds of responsibilities.	Microsoft performs the method of claim 6 wherein the at least one kind of responsibility is selected from a discrete set of kinds of responsibilities. For instance, Azure Service Fabric nodes may be responsible for system services (e.g., name resolution, service discovery), application hosting (e.g., handling client requests), primary and non-primary workloads, resource management (e.g., scaling, load balancing), or health monitoring.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Application and service concepts

Service Fabric Native Application: Service Fabric Native Applications are described by the Native Application Model (XML-based application and service manifests).

Service Fabric Native Application concepts

Application: An application is a collection of constituent services that perform a certain function or functions. The lifecycle of each application instance can be managed independently.

Service: A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service.

Application type: The name/version assigned to a collection of service types. It is defined in an `ApplicationManifest.xml` file and embedded in an application package directory. The directory is then copied to the Service Fabric cluster's image store. You can then create a named application from this application type within the cluster.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service type: The name/version assigned to a service's code packages, data packages, and configuration packages. The service type is defined in the `ServiceManifest.xml` file and embedded in a service package directory. The service package directory is then referenced by an application package's `ApplicationManifest.xml` file. Within the cluster, after creating a named application, you can create a named service from one of the application type's service types. The service type's `ServiceManifest.xml` file describes the service.

Read the Application model article for more information.

There are two types of services:

- **Stateless**: Use a stateless service when the service's persistent state is stored in an external storage service, such as Azure Storage, Azure SQL Database, or Azure Cosmos DB. Use a stateless service when the service has no persistent storage. For example, for a calculator service where values are passed to the service, a computation is performed that uses these values, and then a result is returned.
- **Stateful**: Use a stateful service when you want Service Fabric to manage your service's state via its Reliable Collections or Reliable Actors programming models. When you create a named service, specify how many partitions you want to spread your state over for scalability. Also specify how many times to replicate your state across nodes, for reliability. Each named service has a single primary replica and multiple secondary replicas. You modify your named service's state when you write to the primary replica. Service Fabric then replicates this state to all the secondary replicas to keep your state in sync. Service Fabric automatically detects when a primary replica fails and promotes an existing secondary replica to a primary replica. Service Fabric then creates a new secondary replica.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

System services

There are system services that are created in every cluster that provide the platform capabilities of Service Fabric.

Naming Service: Each Service Fabric cluster has a Naming Service, which resolves service names to a location in the cluster. You manage the service names and properties, like an internet Domain Name System (DNS) for the cluster. Clients securely communicate with any node in the cluster by using the Naming Service to resolve a service name and its location. Applications move within the cluster. For example, this can be due to failures, resource balancing, or the resizing of the cluster. You can develop services and clients that resolve the current network location. Clients obtain the actual machine IP address and port where it's currently running.

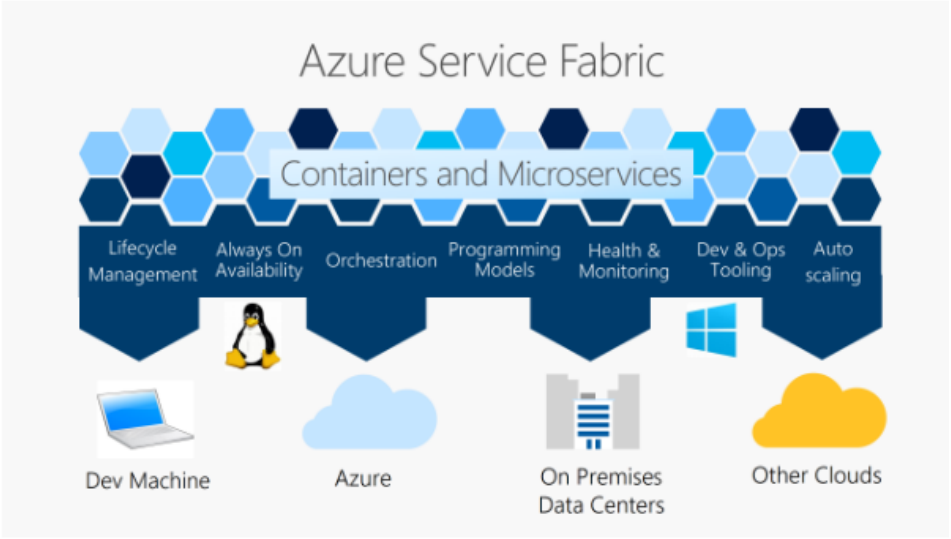
Read Communicate with services for more information on the client and service communication APIs that work with the Naming Service.

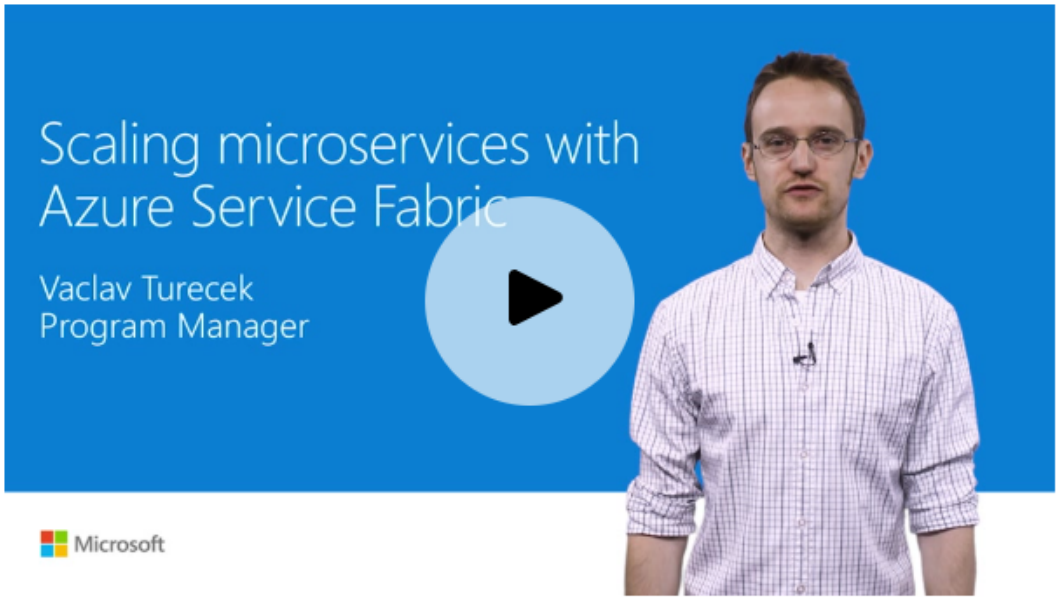
Image Store service: Each Service Fabric cluster has an Image Store service where deployed, versioned application packages are kept. Copy an application package to the Image Store and then register the application type contained within that application package. After the application type is provisioned, you create a named application from it. You can unregister an application type from the Image Store service after all its named applications have been deleted.

Read Understand the ImageStoreConnectionString setting for more information about the Image Store service.

Read the Deploy an application article for more information on deploying applications to the Image Store service.

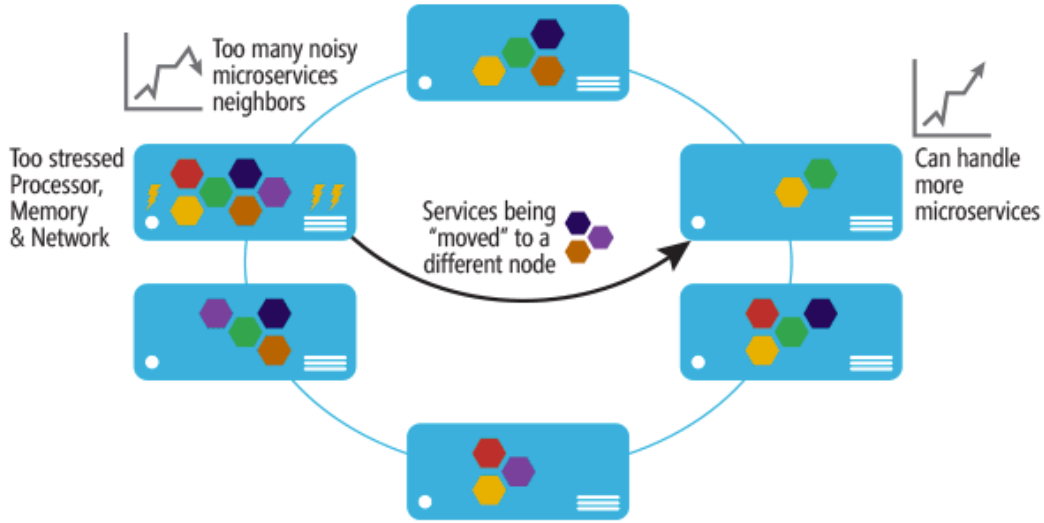
https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	See also Claim 6.
8. The method of claim 6 wherein at least one kind of responsibility is dynamically assigned to each node of the group.	Microsoft performs the method of claim 6 wherein at least one kind of responsibility is dynamically assigned to each node of the group. For instance, Microsoft Azure Service Fabric allows dynamic assignment (e.g., scalability) of services to nodes. Azure Service Fabric is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices and containers. Service Fabric also addresses the significant challenges in developing and managing cloud native applications. A key differentiator of Service Fabric is its strong focus on building stateful services. You can use the Service Fabric programming model or run containerized stateful services written in any language or code. You can create Service Fabric clusters anywhere, including Windows Server and Linux on premises and other public clouds, in addition to Azure.  https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-overview

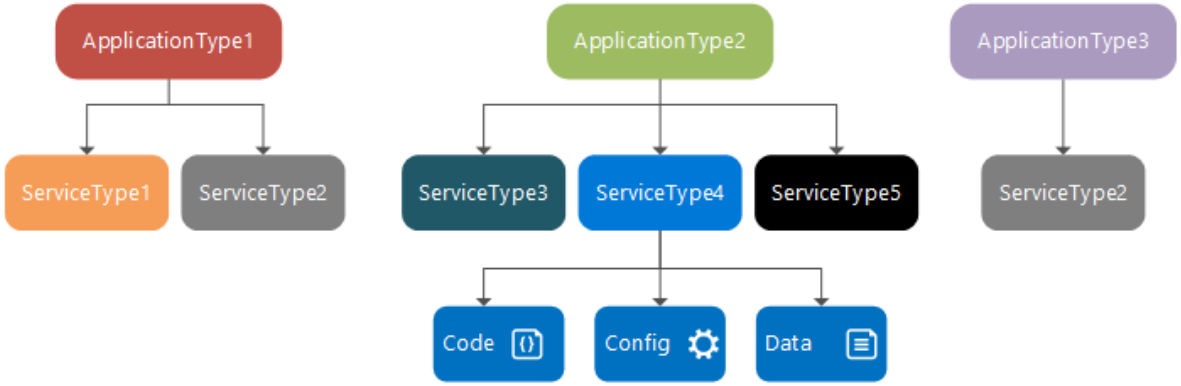
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	# Scaling in Service Fabric 08/22/2024 Azure Service Fabric makes it easy to build scalable applications by managing the services, partitions, and replicas on the nodes of a cluster. Running many workloads on the same hardware enables maximum resource utilization, but also provides flexibility in terms of how you choose to scale your workloads. This Channel 9 video describes how you can build scalable microservices applications:  https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-concepts-scalability

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	Using Dynamic Instance Count Specifically for stateless services, Service Fabric offers an automatic way to change the instance count. This allows the service to scale dynamically with the number of nodes that are available. The way to opt into this behavior is to set the instance count = -1. InstanceCount = -1 is an instruction to Service Fabric that says "Run this stateless service on every node." If the number of nodes changes, Service Fabric automatically changes the instance count to match, ensuring that the service is running on all valid nodes. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-concepts-scalability

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	# Scaling by creating or removing new named services A named service instance is a specific instance of a service type (see Service Fabric application life cycle) within some named application instance in the cluster. New named service instances can be created (or removed) as services become more or less busy. This allows requests to be spread across more service instances, usually allowing load on existing services to decrease. When creating services, the Service Fabric Cluster Resource Manager places the services in the cluster in a distributed fashion. The exact decisions are governed by the metrics in the cluster and other placement rules. Services can be created several different ways, but the most common are either through administrative actions like someone calling New-ServiceFabricService, or by code calling CreateServiceAsync. `CreateServiceAsync` can even be called from within other services running in the cluster. Creating services dynamically can be used in all sorts of scenarios, and is a common pattern. For example, consider a stateful service that represents a particular workflow. Calls representing work are going to show up to this service, and this service is going to execute the steps to that workflow and record progress. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-concepts-scalability Azure Service fabric additionally supports resource balancing (or service orchestration) that dynamically assign services based on a node's capacity (e.g., processor, memory, and network availability).

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	Automatic Resource Balancing Service Fabric provides service resource balancing (or service orchestration) that understands the total available resources in the cluster (like the compute from the VMs). It automatically moves the microservices depending on defined policies and constraints to best optimize how the created services are placed across the Vms (see **Figure 11**). This is focused on optimizing costs and performance.  **Figure 11 Cluster Showing Services Distribution Across the Nodes and Automatic Resource Balancing** https://learn.microsoft.com/en-us/archive/msdn-magazine/2015/december/microsoft-azure-azure-service-fabric-and-the-microservices-architecture
9. The method of claim 6 wherein there is a responsibility set corresponding to each kind	Microsoft performs the method of claim 6 wherein there is a responsibility set corresponding to each kind of responsibility, and wherein each node in the group is assigned to at least one responsibility set. For instance, Azure Service Fabric supports categorization (e.g., classes or types) of applications and services. Azure Service Fabric includes responsibility sets (e.g., categorizations or bundles or service types) corresponding to each kind of

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
of responsibility, and wherein each node in the group is assigned to at least one responsibility set.	responsibility (e.g., service or microservice) such as system services (e.g., name resolution, service discovery), application hosting (e.g., handling client requests), primary and non-primary workloads, resource management (e.g., scaling, load balancing), or health monitoring. An application type is a categorization of an application and consists of a bundle of service types. A service type is a categorization of a service. The categorization can have different settings and configurations, but the core functionality remains the same. The instances of a service are the different service configuration variations of the same service type. Classes (or "types") of applications and services are described through XML files (application manifests and service manifests). The manifests describe applications and services and are the templates against which applications can be instantiated from the cluster's image store. Manifests are covered in detail in Application and service manifests. The schema definition for the ServiceManifest.xml and ApplicationManifest.xml file is installed with the Service Fabric SDK and tools to *C:\Program Files\Microsoft SDKs\Service Fabric\schemas\ServiceFabricServiceModel.xsd*. The XML schema is documented in ServiceFabricServiceModel.xsd schema documentation. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-application-model

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	 There can be one or more instances of a service type active in the cluster. For example, stateful service instances, or replicas, achieve high reliability by replicating state between replicas located on different nodes in the cluster. Replication essentially provides redundancy for the service to be available even if one node in a cluster fails. A partitioned service further divides its state (and access patterns to that state) across nodes in the cluster. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-application-model

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Application and service concepts

Service Fabric Native Application: Service Fabric Native Applications are described by the Native Application Model (XML-based application and service manifests).

Service Fabric Native Application concepts

Application: An application is a collection of constituent services that perform a certain function or functions. The lifecycle of each application instance can be managed independently.

Service: A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service.

Application type: The name/version assigned to a collection of service types. It is defined in an `ApplicationManifest.xml` file and embedded in an application package directory. The directory is then copied to the Service Fabric cluster's image store. You can then create a named application from this application type within the cluster.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service type: The name/version assigned to a service's code packages, data packages, and configuration packages. The service type is defined in the `ServiceManifest.xml` file and embedded in a service package directory. The service package directory is then referenced by an application package's `ApplicationManifest.xml` file. Within the cluster, after creating a named application, you can create a named service from one of the application type's service types. The service type's `ServiceManifest.xml` file describes the service.

Read the Application model article for more information.

There are two types of services:

- **Stateless**: Use a stateless service when the service's persistent state is stored in an external storage service, such as Azure Storage, Azure SQL Database, or Azure Cosmos DB. Use a stateless service when the service has no persistent storage. For example, for a calculator service where values are passed to the service, a computation is performed that uses these values, and then a result is returned.
- **Stateful**: Use a stateful service when you want Service Fabric to manage your service's state via its Reliable Collections or Reliable Actors programming models. When you create a named service, specify how many partitions you want to spread your state over for scalability. Also specify how many times to replicate your state across nodes, for reliability. Each named service has a single primary replica and multiple secondary replicas. You modify your named service's state when you write to the primary replica. Service Fabric then replicates this state to all the secondary replicas to keep your state in sync. Service Fabric automatically detects when a primary replica fails and promotes an existing secondary replica to a primary replica. Service Fabric then creates a new secondary replica.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

System services

There are system services that are created in every cluster that provide the platform capabilities of Service Fabric.

Naming Service: Each Service Fabric cluster has a Naming Service, which resolves service names to a location in the cluster. You manage the service names and properties, like an internet Domain Name System (DNS) for the cluster. Clients securely communicate with any node in the cluster by using the Naming Service to resolve a service name and its location. Applications move within the cluster. For example, this can be due to failures, resource balancing, or the resizing of the cluster. You can develop services and clients that resolve the current network location. Clients obtain the actual machine IP address and port where it's currently running.

Read Communicate with services for more information on the client and service communication APIs that work with the Naming Service.

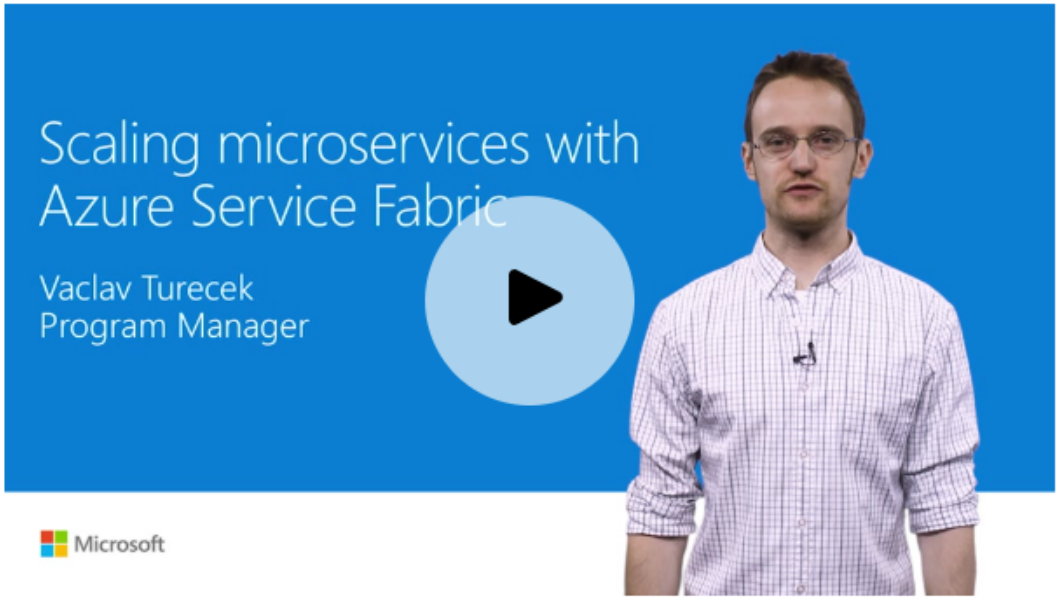
Image Store service: Each Service Fabric cluster has an Image Store service where deployed, versioned application packages are kept. Copy an application package to the Image Store and then register the application type contained within that application package. After the application type is provisioned, you create a named application from it. You can unregister an application type from the Image Store service after all its named applications have been deleted.

Read Understand the ImageStoreConnectionString setting for more information about the Image Store service.

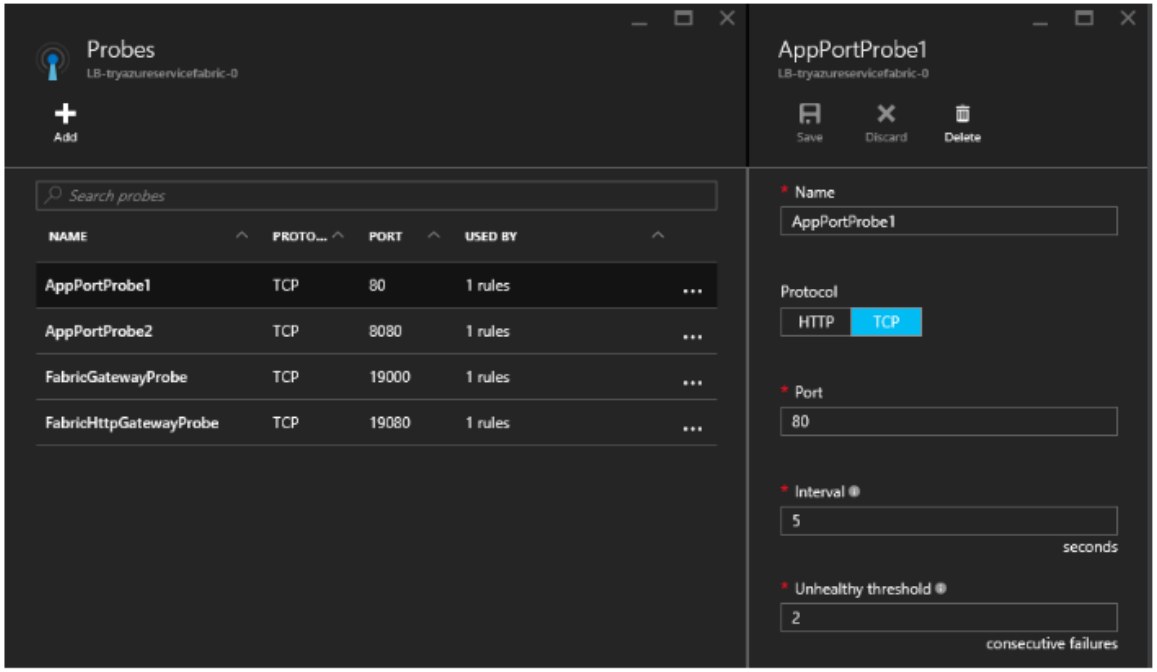
Read the Deploy an application article for more information on deploying applications to the Image Store service.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

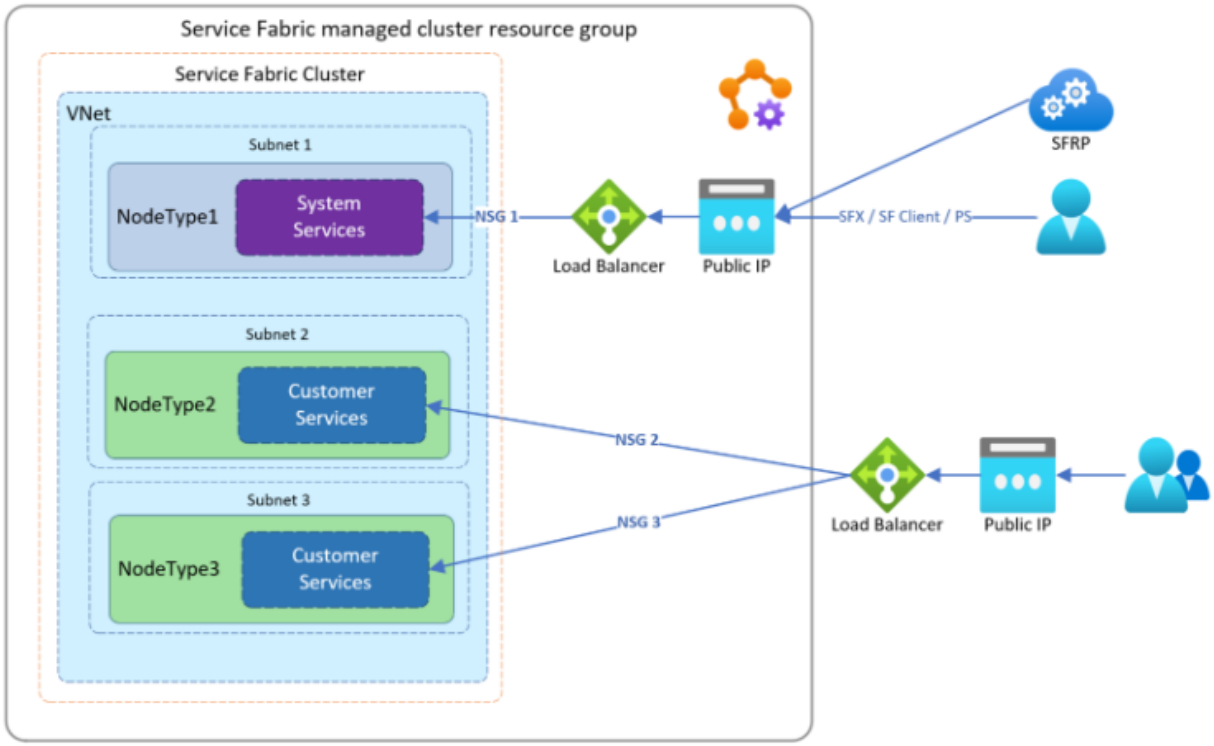
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	See also Claim 6.
11. The method of claim 9 wherein each responsibility set has a corresponding size, and wherein a policy determines the size of each responsibility set based on a measured load on at least a part of the group.	Microsoft performs the method of claim 9 wherein each responsibility set has a corresponding size, and wherein a policy determines the size of each responsibility set based on a measured load on at least a part of the group. For instance, Microsoft Service Fabric clusters have a defined size and responsibility sets (e.g., services) may scale across instances according to a measured load.

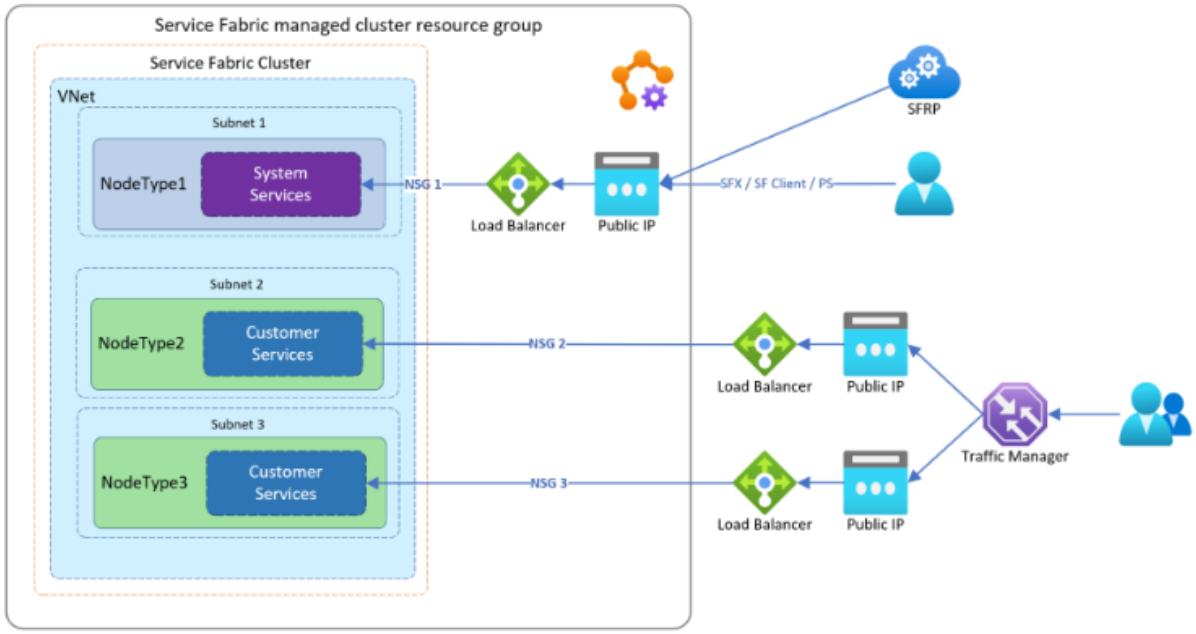
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	# Scaling in Service Fabric 08/22/2024 Azure Service Fabric makes it easy to build scalable applications by managing the services, partitions, and replicas on the nodes of a cluster. Running many workloads on the same hardware enables maximum resource utilization, but also provides flexibility in terms of how you choose to scale your workloads. This Channel 9 video describes how you can build scalable microservices applications:  https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-concepts-scalability

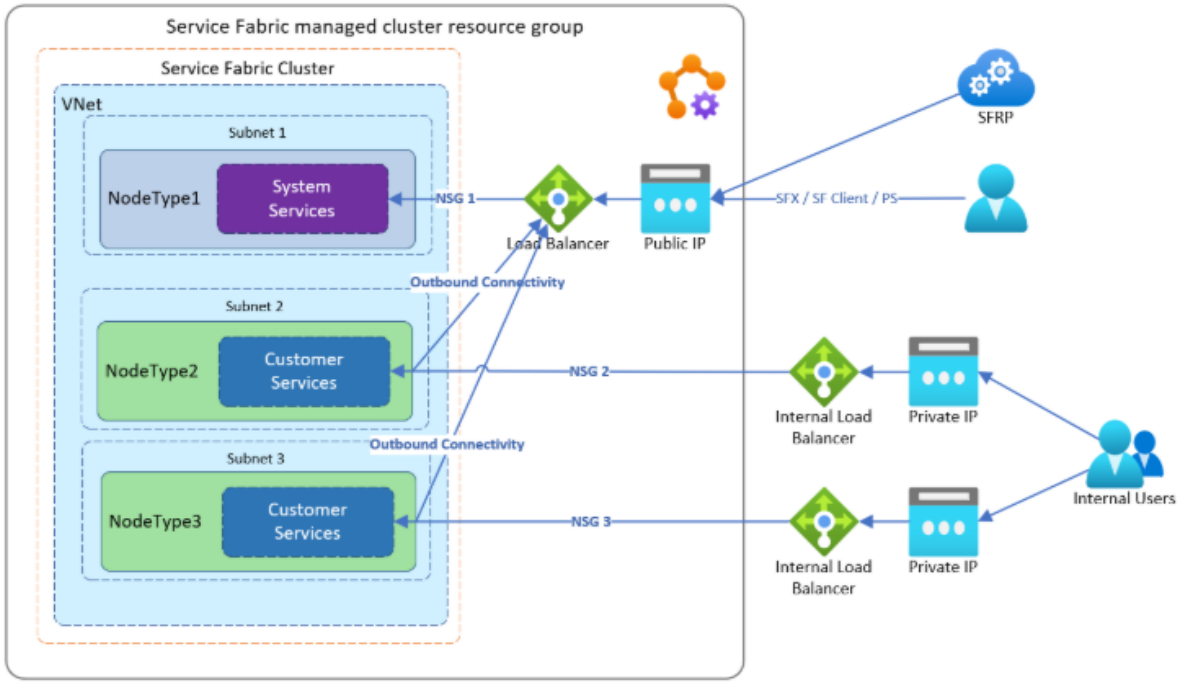
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	Scaling by creating or removing stateless service instances One of the simplest ways to scale within Service Fabric works with stateless services. When you create a stateless service, you get a chance to define an `InstanceCount`. `InstanceCount` defines how many running copies of that service's code are created when the service starts up. Let's say, for example, that there are 100 nodes in the cluster. Let's also say that a service is created with an `InstanceCount` of 10. During runtime, those 10 running copies of the code could all become too busy (or could be not busy enough). One way to scale that workload is to change the number of instances. For example, some piece of monitoring or management code can change the existing number of instances to 50, or to 5, depending on whether the workload needs to scale in or out based on the load. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-concepts-scalability
14. The method of claim 1 wherein attempting to process said request in (C) comprises: **[14.a]** (C)(1) attempting to pass at least some processing of the request to another node in the group.	Microsoft performs the method of claim 1 wherein attempting to process said request in (C) comprises (C)(1) attempting to pass at least some processing of the request to another node in the group. For instance, Microsoft Azure Service Fabric and Azure Load Balancer attempt to send traffic to a particular node, but may send traffic to another node for processing based on capacity constraints or other factors.

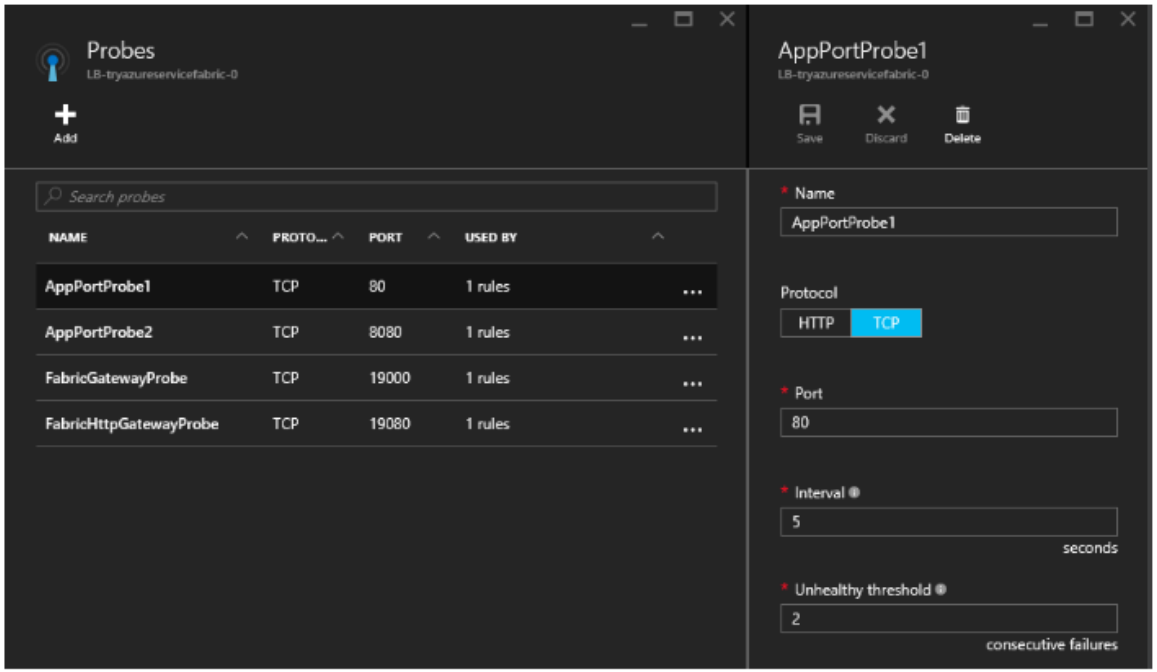
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	4. The Azure Load Balancer uses a probe to determine whether or not to send traffic to a particular node. The probe periodically checks an endpoint on each node to determine whether or not the node is responding. If the probe fails to receive a response after a configured number of times, the load balancer stops sending traffic to that node. When creating a cluster through the Azure portal, a probe is automatically set up for each custom endpoint port that was configured.  https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	# Bring your own Azure Load Balancer Managed clusters create an Azure public Standard Load Balancer and fully qualified domain name with a static public IP for both the primary and secondary node types. Bring your own load balancer allows you to use an existing Azure Load Balancer for secondary node types for both inbound and outbound traffic. When you bring your own Azure Load Balancer, you can: • Use a preconfigured Load Balancer static IP address for either private or public traffic • Map a Load Balancer to a specific node type • Configure network security group rules per node type because each node type is deployed in its own subnet • Maintain existing policies and controls you may have in place • Configure an internal-only load balancer and use the default load balancer for external traffic https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

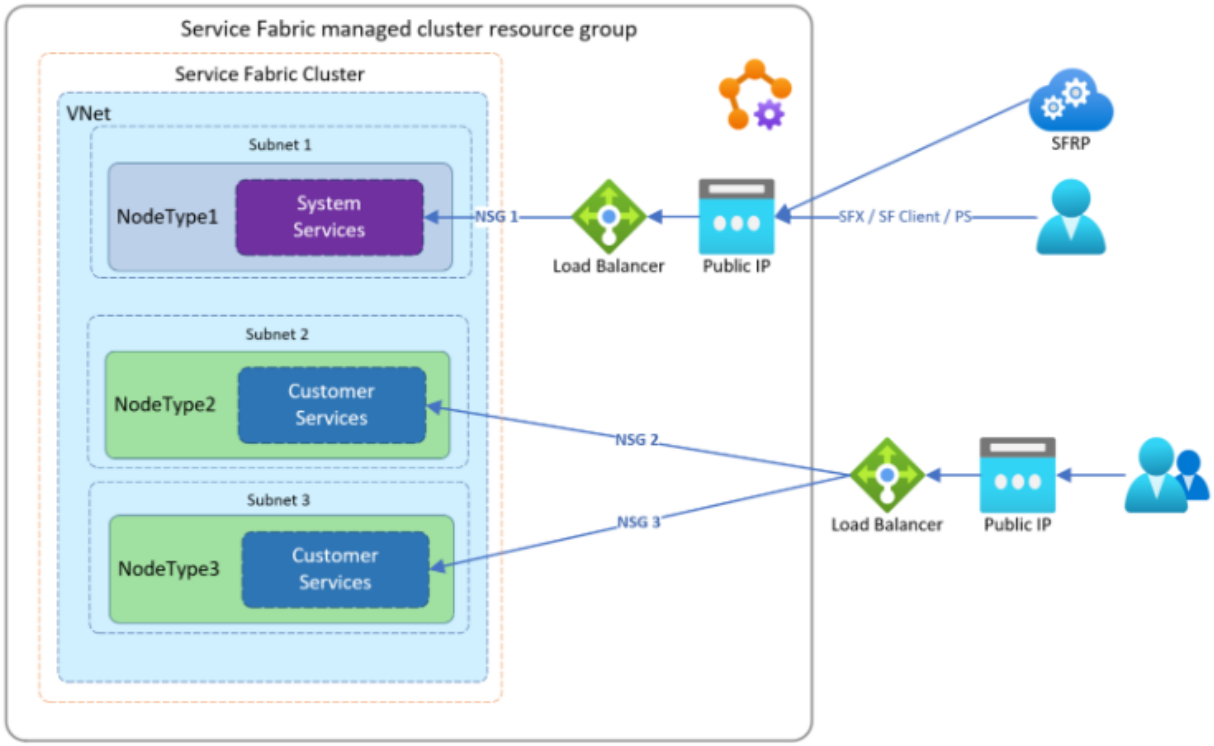
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through an existing Azure Load Balancer configured with an existing static IP address to two node types.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

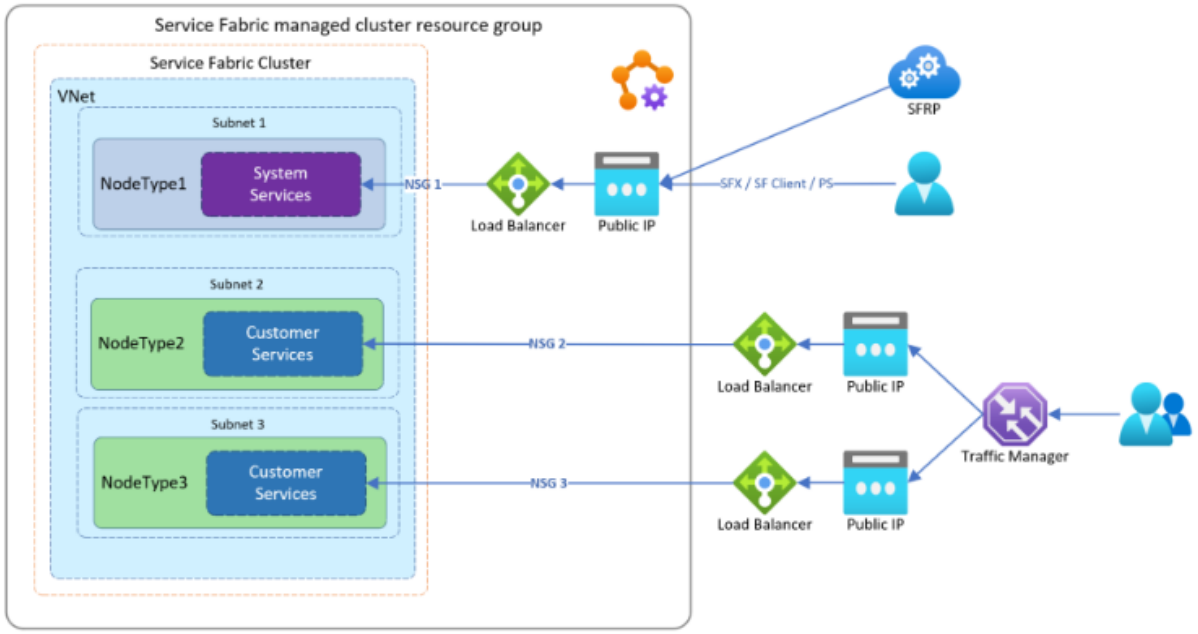
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through existing Azure Load Balancers to help them manage traffic flow to their applications independently that live on separate node types. When set up like this example, each node type will be behind its own managed NSG.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

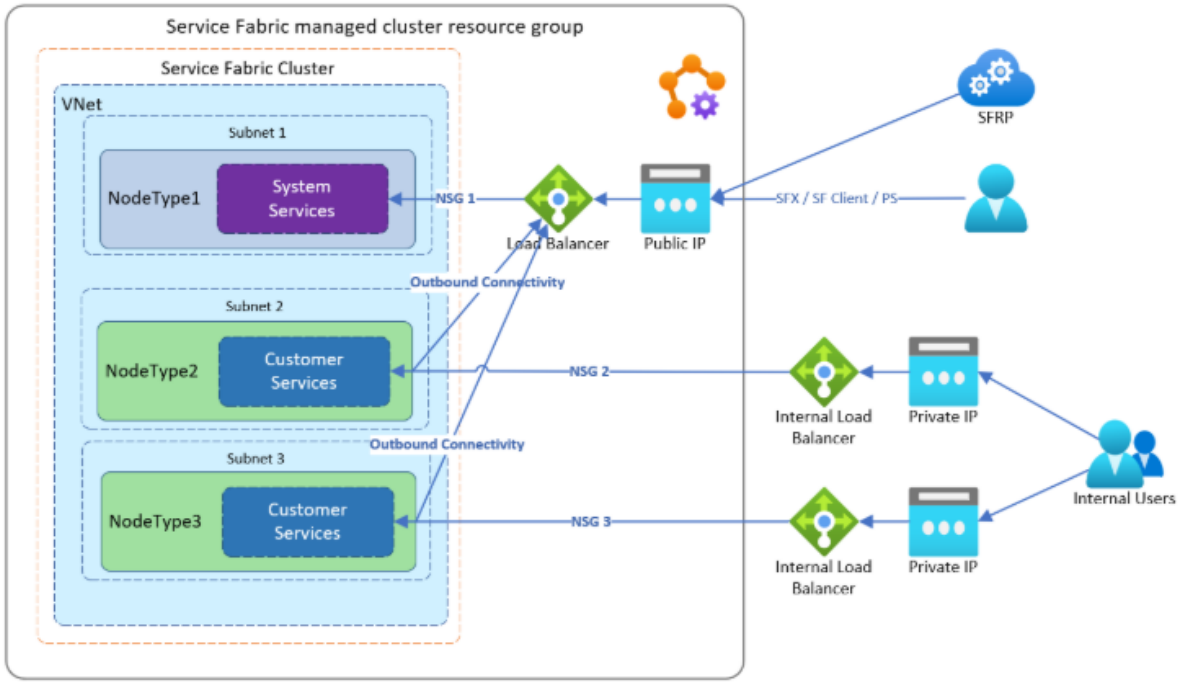
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through existing internal Azure Load Balancers. This helps them manage traffic flow to their applications independently that live on separate node types. When set up like this example, each node type will be behind its own managed NSG and use the default load balancer for external traffic.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking
15. The method of claim 14, wherein said attempting to pass in (C)(1) comprises at least one of:	Microsoft performs the method of claim 14 wherein said attempting to pass in (C)(1) comprises at least one of the following.
[15.a] (i) attempting to transfer at least some processing of the request to	Microsoft attempts to redirect at least some processing of the request to said other node in the group. For instance, Microsoft Azure Service Fabric and Azure Load Balancer attempt to redirect at least some processing of requests to an available node in the group.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
said other node in the group; (ii) attempting to proxy at least some processing of the request to said other node in the group; (iii) attempting to redirect at least some processing of the request to said other node in the group.	4. The Azure Load Balancer uses a probe to determine whether or not to send traffic to a particular node. The probe periodically checks an endpoint on each node to determine whether or not the node is responding. If the probe fails to receive a response after a configured number of times, the load balancer stops sending traffic to that node. When creating a cluster through the Azure portal, a probe is automatically set up for each custom endpoint port that was configured.  https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	# Bring your own Azure Load Balancer Managed clusters create an Azure public Standard Load Balancer and fully qualified domain name with a static public IP for both the primary and secondary node types. Bring your own load balancer allows you to use an existing Azure Load Balancer for secondary node types for both inbound and outbound traffic. When you bring your own Azure Load Balancer, you can: • Use a preconfigured Load Balancer static IP address for either private or public traffic • Map a Load Balancer to a specific node type • Configure network security group rules per node type because each node type is deployed in its own subnet • Maintain existing policies and controls you may have in place • Configure an internal-only load balancer and use the default load balancer for external traffic https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through an existing Azure Load Balancer configured with an existing static IP address to two node types.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through existing Azure Load Balancers to help them manage traffic flow to their applications independently that live on separate node types. When set up like this example, each node type will be behind its own managed NSG.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through existing internal Azure Load Balancers. This helps them manage traffic flow to their applications independently that live on separate node types. When set up like this example, each node type will be behind its own managed NSG and use the default load balancer for external traffic.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking
17. The method of claim 14 wherein said other node in the group is a node with responsibility for the request.	Microsoft performs the method of claim 14 wherein said other node in the group is a node with responsibility for the request. For instance, Azure Service Fabric nodes may be responsible for system services (e.g., name resolution, service discovery), application hosting (e.g., handling client requests), primary and non-primary workloads, resource management (e.g., scaling, load balancing), or health monitoring.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Application and service concepts

Service Fabric Native Application: Service Fabric Native Applications are described by the Native Application Model (XML-based application and service manifests).

Service Fabric Native Application concepts

Application: An application is a collection of constituent services that perform a certain function or functions. The lifecycle of each application instance can be managed independently.

Service: A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service.

Application type: The name/version assigned to a collection of service types. It is defined in an `ApplicationManifest.xml` file and embedded in an application package directory. The directory is then copied to the Service Fabric cluster's image store. You can then create a named application from this application type within the cluster.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service type: The name/version assigned to a service's code packages, data packages, and configuration packages. The service type is defined in the `ServiceManifest.xml` file and embedded in a service package directory. The service package directory is then referenced by an application package's `ApplicationManifest.xml` file. Within the cluster, after creating a named application, you can create a named service from one of the application type's service types. The service type's `ServiceManifest.xml` file describes the service.

Read the Application model article for more information.

There are two types of services:

- **Stateless**: Use a stateless service when the service's persistent state is stored in an external storage service, such as Azure Storage, Azure SQL Database, or Azure Cosmos DB. Use a stateless service when the service has no persistent storage. For example, for a calculator service where values are passed to the service, a computation is performed that uses these values, and then a result is returned.
- **Stateful**: Use a stateful service when you want Service Fabric to manage your service's state via its Reliable Collections or Reliable Actors programming models. When you create a named service, specify how many partitions you want to spread your state over for scalability. Also specify how many times to replicate your state across nodes, for reliability. Each named service has a single primary replica and multiple secondary replicas. You modify your named service's state when you write to the primary replica. Service Fabric then replicates this state to all the secondary replicas to keep your state in sync. Service Fabric automatically detects when a primary replica fails and promotes an existing secondary replica to a primary replica. Service Fabric then creates a new secondary replica.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

System services

There are system services that are created in every cluster that provide the platform capabilities of Service Fabric.

Naming Service: Each Service Fabric cluster has a Naming Service, which resolves service names to a location in the cluster. You manage the service names and properties, like an internet Domain Name System (DNS) for the cluster. Clients securely communicate with any node in the cluster by using the Naming Service to resolve a service name and its location. Applications move within the cluster. For example, this can be due to failures, resource balancing, or the resizing of the cluster. You can develop services and clients that resolve the current network location. Clients obtain the actual machine IP address and port where it's currently running.

Read Communicate with services for more information on the client and service communication APIs that work with the Naming Service.

Image Store service: Each Service Fabric cluster has an Image Store service where deployed, versioned application packages are kept. Copy an application package to the Image Store and then register the application type contained within that application package. After the application type is provisioned, you create a named application from it. You can unregister an application type from the Image Store service after all its named applications have been deleted.

Read Understand the ImageStoreConnectionString setting for more information about the Image Store service.

Read the Deploy an application article for more information on deploying applications to the Image Store service.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
18. The method of claim 14 wherein processing of said request requires a certain kind of responsibility, and wherein the other node has said certain kind of responsibility.	Microsoft performs the method of claim 14 wherein processing of said request requires a certain kind of responsibility, and wherein the other node has said certain kind of responsibility. For instance, Azure Service Fabric nodes may be responsible for system services (e.g., name resolution, service discovery), application hosting (e.g., handling client requests), primary and non-primary workloads, resource management (e.g., scaling, load balancing), or health monitoring.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Application and service concepts

Service Fabric Native Application: Service Fabric Native Applications are described by the Native Application Model (XML-based application and service manifests).

Service Fabric Native Application concepts

Application: An application is a collection of constituent services that perform a certain function or functions. The lifecycle of each application instance can be managed independently.

Service: A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service.

Application type: The name/version assigned to a collection of service types. It is defined in an `ApplicationManifest.xml` file and embedded in an application package directory. The directory is then copied to the Service Fabric cluster's image store. You can then create a named application from this application type within the cluster.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service type: The name/version assigned to a service's code packages, data packages, and configuration packages. The service type is defined in the `ServiceManifest.xml` file and embedded in a service package directory. The service package directory is then referenced by an application package's `ApplicationManifest.xml` file. Within the cluster, after creating a named application, you can create a named service from one of the application type's service types. The service type's `ServiceManifest.xml` file describes the service.

Read the Application model article for more information.

There are two types of services:

- **Stateless**: Use a stateless service when the service's persistent state is stored in an external storage service, such as Azure Storage, Azure SQL Database, or Azure Cosmos DB. Use a stateless service when the service has no persistent storage. For example, for a calculator service where values are passed to the service, a computation is performed that uses these values, and then a result is returned.
- **Stateful**: Use a stateful service when you want Service Fabric to manage your service's state via its Reliable Collections or Reliable Actors programming models. When you create a named service, specify how many partitions you want to spread your state over for scalability. Also specify how many times to replicate your state across nodes, for reliability. Each named service has a single primary replica and multiple secondary replicas. You modify your named service's state when you write to the primary replica. Service Fabric then replicates this state to all the secondary replicas to keep your state in sync. Service Fabric automatically detects when a primary replica fails and promotes an existing secondary replica to a primary replica. Service Fabric then creates a new secondary replica.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

System services

There are system services that are created in every cluster that provide the platform capabilities of Service Fabric.

Naming Service: Each Service Fabric cluster has a Naming Service, which resolves service names to a location in the cluster. You manage the service names and properties, like an internet Domain Name System (DNS) for the cluster. Clients securely communicate with any node in the cluster by using the Naming Service to resolve a service name and its location. Applications move within the cluster. For example, this can be due to failures, resource balancing, or the resizing of the cluster. You can develop services and clients that resolve the current network location. Clients obtain the actual machine IP address and port where it's currently running.

Read Communicate with services for more information on the client and service communication APIs that work with the Naming Service.

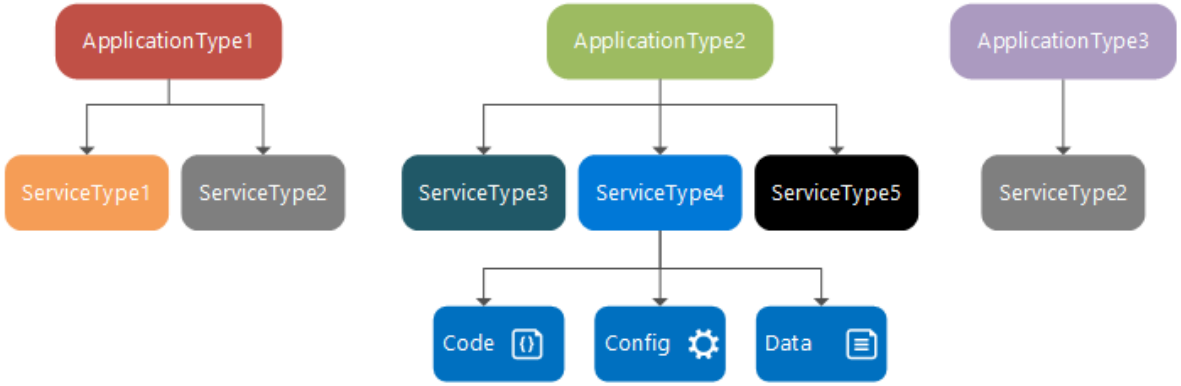
Image Store service: Each Service Fabric cluster has an Image Store service where deployed, versioned application packages are kept. Copy an application package to the Image Store and then register the application type contained within that application package. After the application type is provisioned, you create a named application from it. You can unregister an application type from the Image Store service after all its named applications have been deleted.

Read Understand the ImageStoreConnectionString setting for more information about the Image Store service.

Read the Deploy an application article for more information on deploying applications to the Image Store service.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
19. The method of claim 18 wherein there is a responsibility set corresponding to each kind of responsibility, and wherein each node in the group is assigned to at least one responsibility set, and wherein said other node is in the set of nodes having said certain kind of responsibility.	Microsoft performs the method of claim 18 wherein there is a responsibility set corresponding to each kind of responsibility, and wherein each node in the group is assigned to at least one responsibility set, and wherein said other node is in the set of nodes having said certain kind of responsibility. For instance, Azure Service Fabric supports categorization (e.g., classes or types) of applications and services. Azure Service Fabric includes responsibility sets (e.g., categorizations or bundles or service types) corresponding to each kind of responsibility (e.g., service or microservice) such as system services (e.g., name resolution, service discovery), application hosting (e.g., handling client requests), primary and non-primary workloads, resource management (e.g., scaling, load balancing), or health monitoring. An application type is a categorization of an application and consists of a bundle of service types. A service type is a categorization of a service. The categorization can have different settings and configurations, but the core functionality remains the same. The instances of a service are the different service configuration variations of the same service type. Classes (or "types") of applications and services are described through XML files (application manifests and service manifests). The manifests describe applications and services and are the templates against which applications can be instantiated from the cluster's image store. Manifests are covered in detail in Application and service manifests. The schema definition for the ServiceManifest.xml and ApplicationManifest.xml file is installed with the Service Fabric SDK and tools to *C:\Program Files\Microsoft SDKs\Service Fabric\schemas\ServiceFabricServiceModel.xsd*. The XML schema is documented in ServiceFabricServiceModel.xsd schema documentation. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-application-model

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	 There can be one or more instances of a service type active in the cluster. For example, stateful service instances, or replicas, achieve high reliability by replicating state between replicas located on different nodes in the cluster. Replication essentially provides redundancy for the service to be available even if one node in a cluster fails. A partitioned service further divides its state (and access patterns to that state) across nodes in the cluster. https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-application-model

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Application and service concepts

Service Fabric Native Application: Service Fabric Native Applications are described by the Native Application Model (XML-based application and service manifests).

Service Fabric Native Application concepts

Application: An application is a collection of constituent services that perform a certain function or functions. The lifecycle of each application instance can be managed independently.

Service: A service performs a complete and standalone function and can start and run independently of other services. A service is composed of code, configuration, and data. For each service, code consists of the executable binaries, configuration consists of service settings that can be loaded at run time, and data consists of arbitrary static data to be consumed by the service.

Application type: The name/version assigned to a collection of service types. It is defined in an `ApplicationManifest.xml` file and embedded in an application package directory. The directory is then copied to the Service Fabric cluster's image store. You can then create a named application from this application type within the cluster.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities

Service type: The name/version assigned to a service's code packages, data packages, and configuration packages. The service type is defined in the `ServiceManifest.xml` file and embedded in a service package directory. The service package directory is then referenced by an application package's `ApplicationManifest.xml` file. Within the cluster, after creating a named application, you can create a named service from one of the application type's service types. The service type's `ServiceManifest.xml` file describes the service.

Read the Application model article for more information.

There are two types of services:

- **Stateless**: Use a stateless service when the service's persistent state is stored in an external storage service, such as Azure Storage, Azure SQL Database, or Azure Cosmos DB. Use a stateless service when the service has no persistent storage. For example, for a calculator service where values are passed to the service, a computation is performed that uses these values, and then a result is returned.
- **Stateful**: Use a stateful service when you want Service Fabric to manage your service's state via its Reliable Collections or Reliable Actors programming models. When you create a named service, specify how many partitions you want to spread your state over for scalability. Also specify how many times to replicate your state across nodes, for reliability. Each named service has a single primary replica and multiple secondary replicas. You modify your named service's state when you write to the primary replica. Service Fabric then replicates this state to all the secondary replicas to keep your state in sync. Service Fabric automatically detects when a primary replica fails and promotes an existing secondary replica to a primary replica. Service Fabric then creates a new secondary replica.

https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

System services

There are system services that are created in every cluster that provide the platform capabilities of Service Fabric.

Naming Service: Each Service Fabric cluster has a Naming Service, which resolves service names to a location in the cluster. You manage the service names and properties, like an internet Domain Name System (DNS) for the cluster. Clients securely communicate with any node in the cluster by using the Naming Service to resolve a service name and its location. Applications move within the cluster. For example, this can be due to failures, resource balancing, or the resizing of the cluster. You can develop services and clients that resolve the current network location. Clients obtain the actual machine IP address and port where it's currently running.

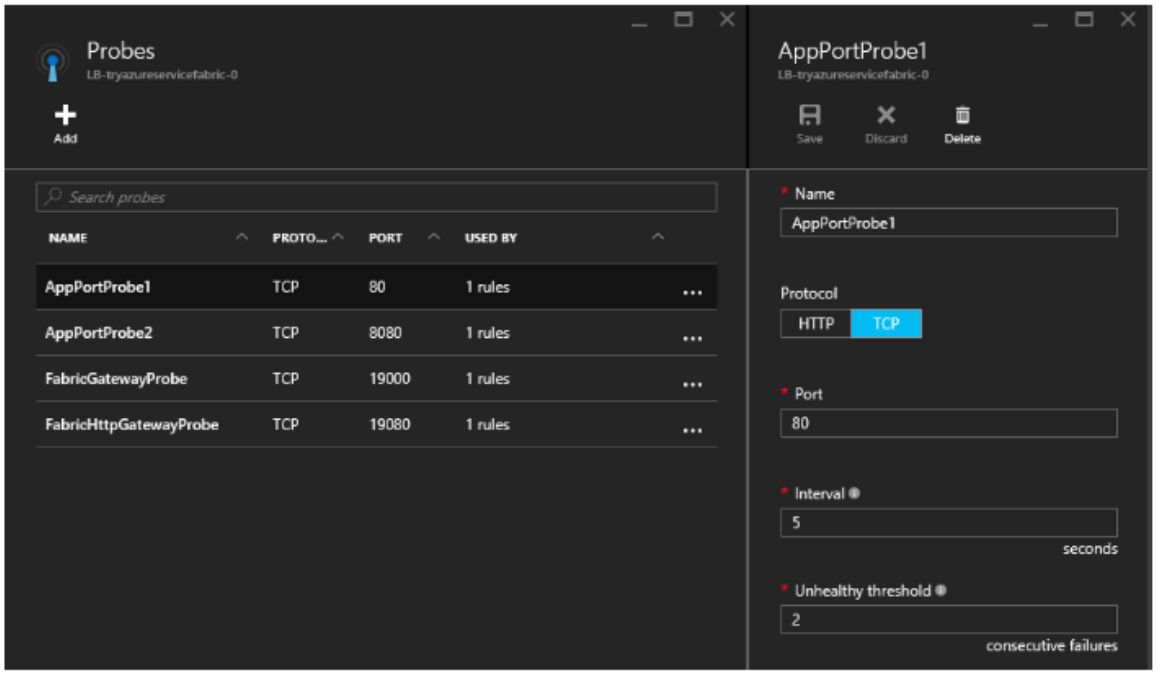
Read Communicate with services for more information on the client and service communication APIs that work with the Naming Service.

Image Store service: Each Service Fabric cluster has an Image Store service where deployed, versioned application packages are kept. Copy an application package to the Image Store and then register the application type contained within that application package. After the application type is provisioned, you create a named application from it. You can unregister an application type from the Image Store service after all its named applications have been deleted.

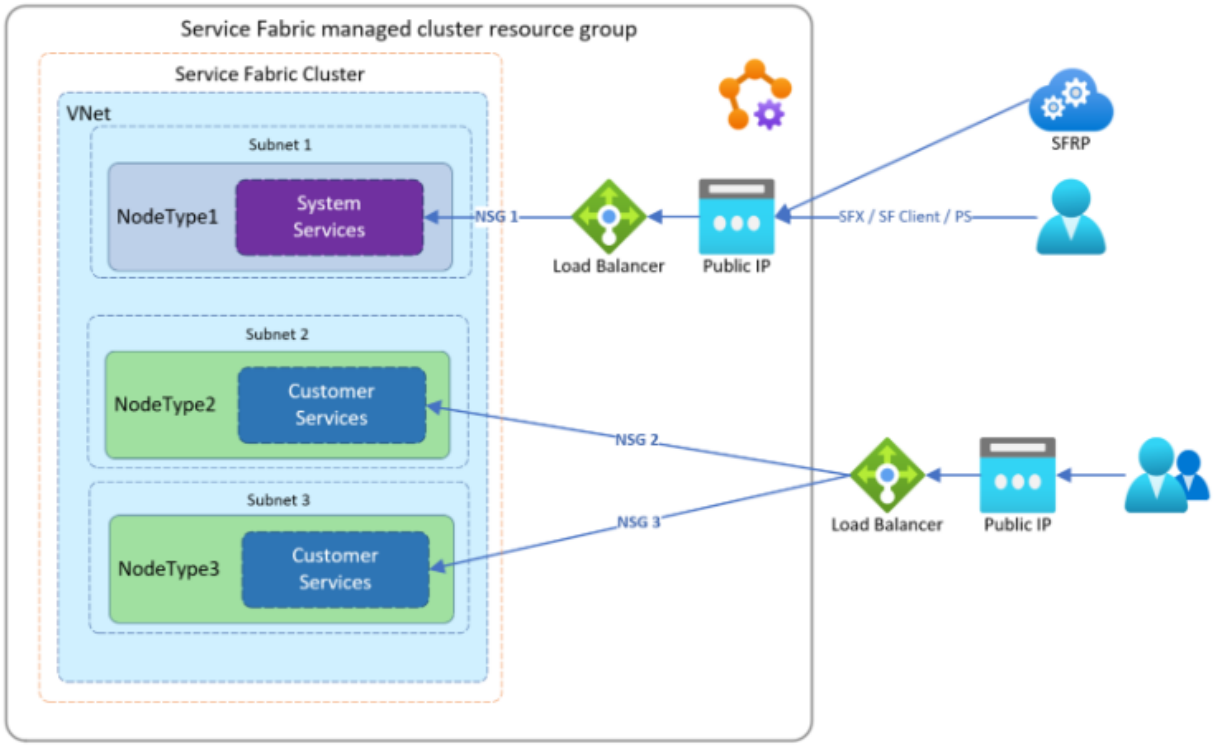
Read Understand the ImageStoreConnectionString setting for more information about the Image Store service.

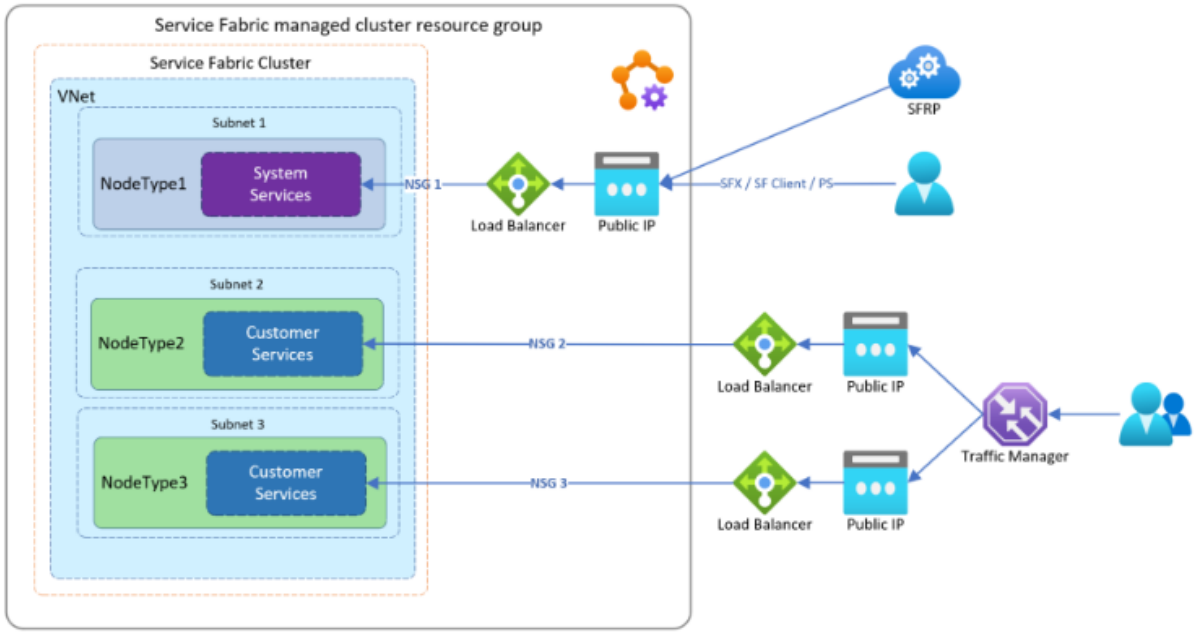
Read the Deploy an application article for more information on deploying applications to the Image Store service.

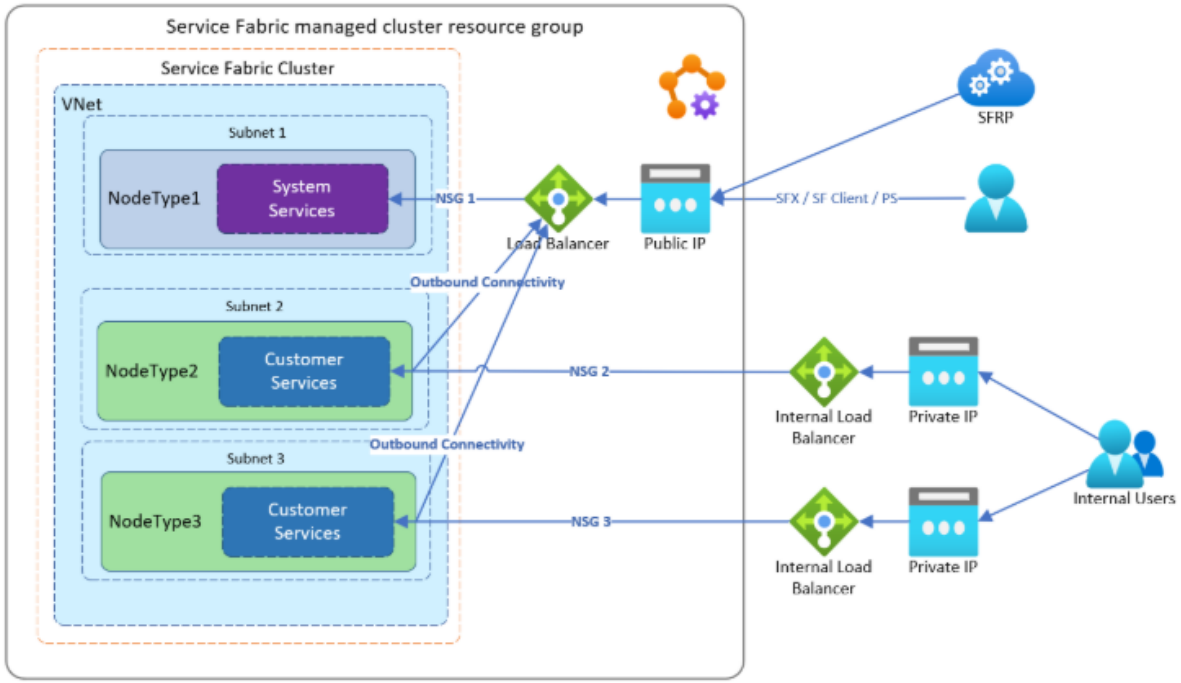
https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-technical-overview

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
20. The method of claim 14 further comprising: **[20.a]** (C)(2) in response to said attempting to pass, when said other node does not respond, attempting to pass said at least some processing of the request to at least one second other node having the kind of responsibility to process the request.	Microsoft performs the method of claim 14 further comprising (C)(2) in response to said attempting to pass, when said other node does not respond, attempting to pass said at least some processing of the request to at least one second other node having the kind of responsibility to process the request. For instance, Microsoft Azure Service Fabric and Azure Load Balancer attempt to redirect at least some processing of requests to an available node in the group or cluster. 4. The Azure Load Balancer uses a probe to determine whether or not to send traffic to a particular node. The probe periodically checks an endpoint on each node to determine whether or not the node is responding. If the probe fails to receive a response after a configured number of times, the load balancer stops sending traffic to that node. When creating a cluster through the Azure portal, a probe is automatically set up for each custom endpoint port that was configured.  https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-connect-and-communicate-with-services

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	# Bring your own Azure Load Balancer Managed clusters create an Azure public Standard Load Balancer and fully qualified domain name with a static public IP for both the primary and secondary node types. Bring your own load balancer allows you to use an existing Azure Load Balancer for secondary node types for both inbound and outbound traffic. When you bring your own Azure Load Balancer, you can: • Use a preconfigured Load Balancer static IP address for either private or public traffic • Map a Load Balancer to a specific node type • Configure network security group rules per node type because each node type is deployed in its own subnet • Maintain existing policies and controls you may have in place • Configure an internal-only load balancer and use the default load balancer for external traffic https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

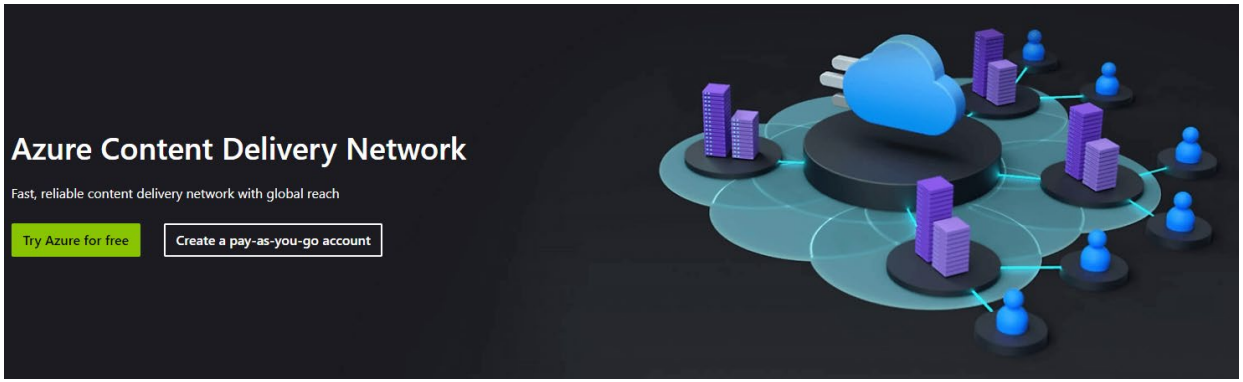
U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through an existing Azure Load Balancer configured with an existing static IP address to two node types.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through existing Azure Load Balancers to help them manage traffic flow to their applications independently that live on separate node types. When set up like this example, each node type will be behind its own managed NSG.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
	In this example, a customer wants to route traffic through existing internal Azure Load Balancers. This helps them manage traffic flow to their applications independently that live on separate node types. When set up like this example, each node type will be behind its own managed NSG and use the default load balancer for external traffic.  https://learn.microsoft.com/en-us/azure/service-fabric/how-to-managed-cluster-networking
21. The method of claim 20 further comprising: **[21.a]** (C)(3) repeating act (C)(2) until either **[21.b]** (i) another node accepts	Microsoft performs the method of claim 20 further comprising (C)(3) repeating act (C)(2) until another node accepts responsibility for said request. For instance, on information and belief, Azure Load Balancer repeats the act of attempting to pass processing of a request until a node accepts responsibility. See Claim 20.

U.S. Patent No. 9,722,883	Microsoft Accused Functionalities
responsibility for said request, or	
[21.c] (ii) the attempting to pass in(C)(1) has been tried for all other nodes having the kind of responsibility to process the request.	Alternatively, Microsoft repeats act (C)(2) until the attempting to pass in (C)(1) has been tried for all other nodes having the kind of responsibility to process the request. For instance, on information and belief, Azure Load Balancer repeats the act of attempting to pass processing of a request until all available nodes have been tried. See Claim 20.
23. The method of claim 14 wherein, in (C)(1), said node attempts to pass all processing of the request to another node in the group.	Microsoft performs the method of claim 14 wherein, in (C)(1), said node attempts to pass all processing of the request to another node in the group. For instance, on information and belief, Azure Load Balancer attempts to pass all processing of a request to a node in the group. See Claim 14.

EXHIBIT E

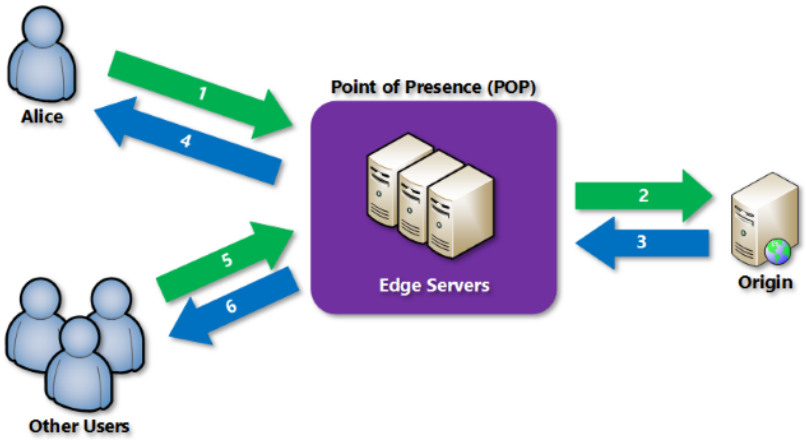
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
1[Pre]. A method of content delivery in a content delivery network comprising at least a first tier of servers, the method comprising:	Microsoft performs a method of content delivery in a content delivery network comprising at least a first tier of servers as set forth below. For instance, Microsoft Azure CDN and/or Azure Front Door comprise at least a first tier of servers (e.g., edge nodes or edge POPs).  https://azure.microsoft.com/en-us/products/cdn **Increase speed and scalability for an optimal user experience** Azure Content Delivery Network offers a global solution for rapidly delivering content. Save bandwidth and improve responsiveness when encoding or distributing gaming software, firmware updates, and IoT endpoints. Reduce load times for websites, mobile apps, and streaming media to increase user satisfaction globally. https://azure.microsoft.com/en-us/products/cdn Stream media and download large files quickly with optimized delivery. Provide a scalable solution to handle worldwide traffic spikes and instantaneous high loads. Integrate seamlessly with your Azure services to activate within minutes. Protect content with custom domain HTTPS, DDoS, and WAF protection. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Improved performance and user experience Users expect a fast, reliable, and personalized web experience wherever they are. Content Delivery Network helps reduce latency and improve performance for high-bandwidth content by distributing user requests and serving content directly from edge servers. This brings the content closer to users and sends less traffic to the origin point, delivering superior online experiences. https://azure.microsoft.com/en-us/products/cdn **Global coverage and massive scalability** Scale on the fly with no downtime using a platform that responds to traffic fluctuations. A distributed, global presence enables Content Delivery Network to handle sudden traffic spikes and heavy loads—for example, during major product launches or global sporting events—without new infrastructure costs or capacity concerns. Global, distributed points of presence (PoPs) ensure fast delivery of content and protects the origin from excessive load. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Protection at the edge Optimize security through features like custom domain HTTPS and DDoS mitigation. Enable Content Delivery Network to put edge servers on the front line serving and protect the origin server from malicious attacks. Set advanced rules to block requests or validate users and use geo-filtering to ensure the content is delivered only to specific regions. https://azure.microsoft.com/en-us/products/cdn **Automate processes and workflows with DevOps** Azure APIs and developer tools give you the performance, reliability, and security needed to build and automate global applications fast. A unified API allows you to deploy Content Delivery Network endpoints from each provider and still manage them the same way. Azure APIs also provide the flexibility to optimize features for availability and performance. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

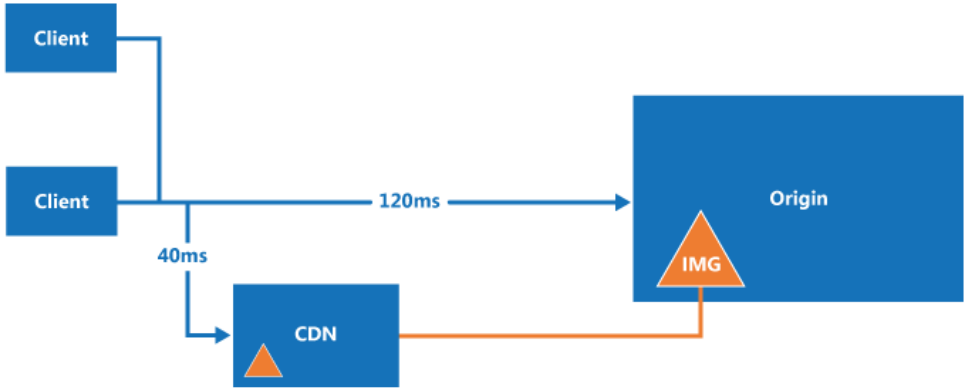
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	This article lists current metros containing point of presence (POP) locations, sorted by region, for Azure Content Delivery Network products. Each metro might contain more than one POP. For example, Azure Content Delivery Network from Microsoft has 192 POPs across 109 metro cities. **ⓘ Important** Each Azure Content Delivery Network product has a distinct way of building its content delivery network infrastructures, hence Microsoft recommends against using POP locations to decide which Azure Content Delivery Network product to use. Instead, you should consider its features and end-user performance. Test the performance with each Azure Content Delivery Network product to choose the right product for your users. https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Region	Cities
North America	Etobicoke, Canada (2) Montreal, Canada Vancouver, Canada (2) Querétaro, Mexico (2) Atlanta, GA, USA (3) Boydton, VA, USA (2) Chaska, MN, USA (2) Cheyenne, WY, USA (2) Chicago, IL, USA (4) Dallas, TX, USA (4) Des Moines,IA, USA (3) Detroit, MI, USA Englewood, CO, USA (2) Honolulu, HI, USA Houston, TX, USA (3) Jacksonville, FL, USA (2) Las Vegas, NV, USA (3) Los Angeles, CA, USA (2) Manassas, VA, USA (3) Memphis, TN, USA Miami, FL, USA (4) Minneapolis, MN, USA (2) Needham Heights, MA, USA (2) Nashville, TN, USA Newark, NJ, USA New York, NY, USA (2) Philadelphia, PA, USA Phoenix, AZ, USA Plano, TX, USA Portland, OR, USA Quincy, WA, USA (3) San Antonio, TX, USA (4) San Jose, CA, USA (4) Salt Lake City, UT, USA (3) Seattle, WA, USA (2) Secaucus, NJ, USA (2) Southfield, MI, USA Tempe, AZ, USA

https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Edge Nodes - List

Reference

Feedback

Service: CDN

API Version: 2024-02-01

Edgenodes are the global Point of Presence (POP) locations used to deliver CDN content to end users.

HTTP Copy Try It

```
GET https://management.azure.com/providers/Microsoft.Cdn/edgenodes?api-version=2024-02-01
```

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

EdgeNode

Edgenode is a global Point of Presence (POP) location used to deliver CDN content to end users.

Expand table

Name	Type	Description
id	string	Resource ID.
name	string	Resource name.
properties.ipAddressGroups	IpAddressGroup[]	List of ip address groups.
systemData	SystemData	Read only system data
type	string	Resource type.

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Connectivity within Microsoft's network along with new Regional Caching capabilities enables more consistent, more predictable cache fill performance by providing multi-tier caching along direct, private access to content in Azure from each CDN edge point of presence (POP). Azure CDN from Microsoft entered public preview providing access to 54 global Edge POPs in 33 countries and 16 Regional Cache POPs.  https://azure.microsoft.com/es-es/blog/new-locations-for-azure-cdn-now-available/ Microsoft Azure CDN additionally includes products such as Azure Front Door as well as Standard and Premium CDN services from Edgio (formerly Verizon).

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Azure Content Delivery Network includes three products: • Azure CDN Standard from Microsoft • Azure CDN Standard from Edgio (formerly Verizon) • Azure CDN Premium from Edgio (formerly Verizon). https://learn.microsoft.com/en-us/azure/cdn/cdn-features

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Performance features and optimizations	Standard Microsoft	Standard Edgio	Premium Edgio
Dynamic site acceleration	Offered via Azure Front Door	✓	✓
General web delivery optimization	✓	✓	✓
Video streaming optimization	via General Web Delivery	via General Web Delivery	via General Web Delivery
Large file optimization	via General Web Delivery	via General Web Delivery	via General Web Delivery
Origin Port	All TCP ports	All TCP ports	All TCP ports
Global server load balancing (GSLB)	✓	✓	✓
Fast purge	✓	✓	✓
Asset pre-loading		✓	✓
Cache/header settings (using caching rules)	✓ using Standard rules engine	✓	
Customizable, rules based content delivery engine	✓ using Standard rules engine		✓ using rules engine
Cache/header settings	✓ using Standard rules engine		✓ using Premium rules engine
URL redirect/rewrite	✓ using Standard rules engine		✓ using Premium rules engine
Mobile device rules	✓ using Standard rules engine		✓ using Premium rules engine
Query string caching	✓	✓	✓
Internet Protocol version 4 (IPv4)/Internet Protocol version 6 (IPv6) dual-stack	✓	✓	✓
HTTP/2 support	✓	✓	✓

https://learn.microsoft.com/en-us/azure/cdn/cdn-features

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Microsoft also performs the methods of the asserted claims of the '692 Patent to serve Microsoft resources and services using the same or similar infrastructure. For example, Microsoft uses internal CDN technologies such as its Azure CDN to support at least Xbox gaming, Office 365, Windows updates, and web resources like Linkedin and Bing. • Built on Microsoft's massive-scale private global network, Azure Front Door is a proven platform used to power some of the largest and latency sensitive global services at Microsoft, such as Microsoft Office 365, Bing, LinkedIn, and Xbox. https://azure.microsoft.com/en-us/blog/azure-front-door-enhances-secure-cloud-cdn-with-intelligent-threat-protection/ Azure CDN provides a world class platform to let you reduce load times, save bandwidth, and speed responsiveness across your businesses diverse workflows. Azure CDN from Microsoft enables Azure customers to use and deliver content from the same global CDN network leveraged by Microsoft properties such as Office 365, Skype, Bing, OneDrive, Windows, and Xbox. Connectivity within Microsoft's network along with new Regional Caching capabilities enables more consistent, more predictable cache fill performance by providing multi-tier caching along direct, private access to content in Azure from each CDN edge point of presence (POP). Azure CDN from Microsoft entered public preview providing access to 54 global Edge POPs in 33 countries and 16 Regional Cache POPs. https://azure.microsoft.com/en-us/blog/announcing-microsoft-s-own-cdn-network/

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	What powers Windows 365? Not surprising, Microsoft's own cloud computing platform, Azure. Specifically, **Azure Virtual Desktop**, which can be used to create Windows virtual machines on Redmond's servers, then stream those desktops to PC and non-PC devices. https://www.computerworld.com/article/1614540/what-is-windows-365-microsoft-s-cloud-based-os-explained.html **Use the Office 365 Content Delivery Network (CDN) with SharePoint Online** 02/28/2025 You can use the built-in Office 365 Content Delivery Network (CDN) to host static assets to provide better performance for your SharePoint Online pages. The Office 365 CDN improves performance by caching static assets closer to the browsers requesting them, which helps to speed up downloads and reduce latency. Also, the Office 365 CDN uses the HTTP/2 protocol for improved compression and HTTP pipelining. The Office 365 CDN service is included as part of your SharePoint Online subscription.

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	The Office 365 CDN is composed of multiple CDNs that allow you to host static assets in multiple locations, or *origins*, and serve them from global high-speed networks. Depending on the kind of content you want to host in the Office 365 CDN such as you can add **public** origins. If you're already familiar with the way that CDNs work, you only need to complete a few steps to enable the Office 365 CDN for your tenant. This article describes how. Read on for information about how to get started hosting your static assets. https://learn.microsoft.com/en-us/microsoft-365/enterprise/use-microsoft-365-cdn-with-spo?view=o365-worldwide On the Microsoft side, the company uses primarily Azure Front Door a cloud CDN platform and some third-party CDNs for some static delivery of content i.e Xbox downloads, Windows Updates. Microsoft does not https://www.streamingmediablog.com/2020/10/gaming-console-launch.html

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Teams is built on a microservices architecture, with a few hundred microservices working cohesively to deliver our product's many features including messaging, meetings, files, calendar, and apps. Using microservices helps each of our component teams to work and release their changes independently. Azure is the cloud platform that underpins all of Microsoft's cloud services, including Microsoft Teams. Our workloads run in Azure virtual machines (VMs), with our older services being deployed through Azure Cloud Services and our newer ones on Azure Service Fabric. Our primary storage stack is Azure Cosmos DB, with some services using Azure Blob Storage. We count on Azure Cache for Redis for increased throughput and resiliency. We leverage Traffic Manager and Azure Front Door to route traffic where we want it to be. We use Queue Storage and Event Hubs to communicate, and we depend on Azure Active Directory to manage our tenants and users. • **Resiliency-optimized caches**: We leverage caches between our components extensively, for both performance and resiliency. Caches help reduce average latency and provide a source of data in case a downstream service is unavailable. Keeping data in caches for a long time introduces data freshness issues yet keeping data in caches for a long time is the best defense against downstream failures. We focus on Time to Refresh (TTR) to our cache data as well as Time to Live (TTL). By setting a long TTL and a shorter TTR value, we can fine-tune how fresh to keep our data versus how long we want data to stick around whenever a downstream dependency fails. https://azure.microsoft.com/en-us/blog/advancing-microsoft-teams-on-azure-operating-at-pandemic-scale/

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Background

A CDN is a system of distributed servers that delivers web content to users from geographical locations closest to them. At LinkedIn, CDNs deliver static content, such as CSS and JavaScript on the site, as well as member-generated content, such as profile pictures, media shares, and video uploads, to our members around the world. The content is delivered from edges that are closest to the user in question.

The Edge SRE team runs LinkedIn's edge infrastructure, including four external CDNs, an in-house CDN, three Domain Name System (DNS) platforms, and all LinkedIn points of presence (PoPs). We build and manage tools to automate all aspects of our Edge stack.

https://www.linkedin.com/blog/engineering/data-management/enabling-dual-stack-on-linkedin-cdns\

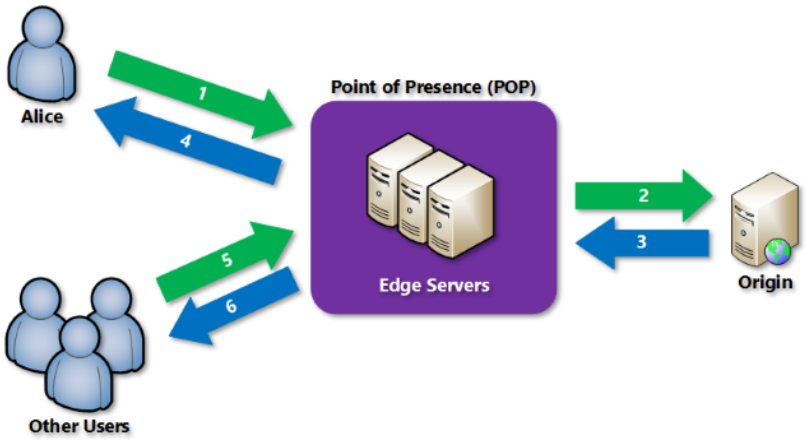
> LinkedIn recently described how it uses the edge to support the popular professional networking web site. The engineering team wrote a blog post to talk about edge deployments and its migration away from an in-house solution to Azure Edge.
>
> LinkedIn is one of many large web properties that is owned by a hyperscaler cloud provider — Microsoft, in this case — but doesn't actually use the in-house cloud. YouTube is another example of this — it has yet to be migrated to GCP.
>
> LinkedIn actually had its own quite extensive in-house CDN, with nineteen PoPs (points of presence) in ten countries. This is a common practice for major content owners as it defrays transport costs and improves performance. The nodes can cache content and perform tasks such as SSL offloading. But as LinkedIn noted, Microsoft's CDN – Azure Front Door – gives access to 300 PoPs around the globe, which is a big step up in terms of reach.

https://www.edgeir.com/how-linkedin-is-using-azure-front-door-for-edge-build-out-20220306

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
[1.a] at a first server in the first tier of servers, obtaining a request from a client for a resource, wherein the resource is available as part of a content provider's library;	Microsoft obtains, at a first server in the first tier of servers, a request from a client for a resource, wherein the resource is available as part of a content provider's library. For instance, Microsoft Azure CDN and/or Azure Front Door contain edge servers like POP servers that obtain a client resource request, wherein the requested resource is part of a content library provided by Microsoft or an Azure CDN customer. **Improved performance and user experience** Users expect a fast, reliable, and personalized web experience wherever they are. Content Delivery Network helps reduce latency and improve performance for high-bandwidth content by distributing user requests and serving content directly from edge servers. This brings the content closer to users and sends less traffic to the origin point, delivering superior online experiences. https://azure.microsoft.com/en-us/products/cdn **Global coverage and massive scalability** Scale on the fly with no downtime using a platform that responds to traffic fluctuations. A distributed, global presence enables Content Delivery Network to handle sudden traffic spikes and heavy loads—for example, during major product launches or global sporting events—without new infrastructure costs or capacity concerns. Global, distributed points of presence (PoPs) ensure fast delivery of content and protects the origin from excessive load. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

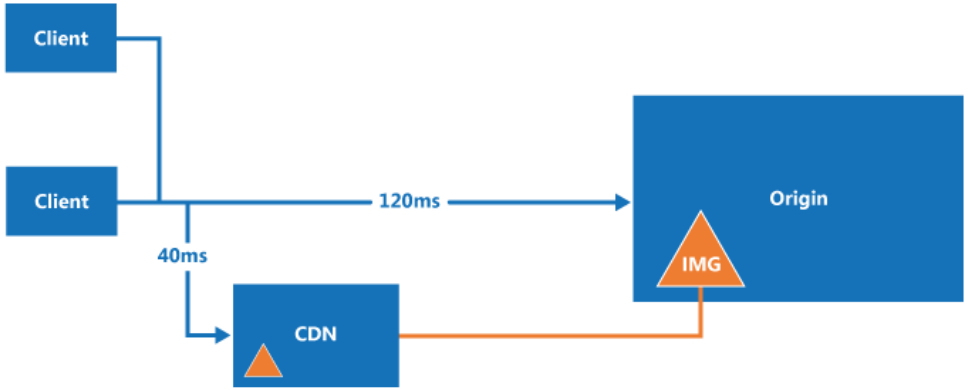
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Edge Nodes - List

Reference

Feedback

Service: CDN

API Version: 2024-02-01

Edgenodes are the global Point of Presence (POP) locations used to deliver CDN content to end users.

HTTP Copy Try It

```
GET https://management.azure.com/providers/Microsoft.Cdn/edgenodes?api-version=2024-02-01
```

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

EdgeNode

Edgenode is a global Point of Presence (POP) location used to deliver CDN content to end users.

Expand table

Name	Type	Description
id	string	Resource ID.
name	string	Resource name.
properties.ipAddressGroups	IpAddressGroup[]	List of ip address groups.
systemData	SystemData	Read only system data
type	string	Resource type.

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	# Introduction to caching Caching is the process of storing data locally so that future requests for that data can be accessed more quickly. In the most common type of caching, web browser caching, a web browser stores copies of static data locally on a local hard drive. By using caching, the web browser can avoid making multiple round-trips to the server and instead access the same data locally, thus saving time and resources. Caching is well-suited for locally managing small, static data such as static images, CSS files, and JavaScript files. Similarly, caching is used by a content delivery network on edge servers close to the user to avoid requests traveling back to the origin and reducing end-user latency. Unlike a web browser cache, which is used only for a single user, the content delivery network has a shared cache. In a content delivery network shared cache, a file request by a user can be used by another user, which greatly decreases the number of requests to the origin server. Dynamic resources that change frequently or are unique to an individual user can't be cached. Those types of resources, however, can take advantage of dynamic site acceleration (DSA) optimization on the Azure content delivery network for performance improvements. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Caching can occur at multiple levels between the origin server and the end user: • Web server: Uses a shared cache (for multiple users). • Content delivery network: Uses a shared cache (for multiple users). • Internet service provider (ISP): Uses a shared cache (for multiple users). • Web browser: Uses a private cache (for one user). Each cache typically manages its own resource freshness and performs validation when a file is stale. This behavior is defined in the HTTP caching specification, RFC 7234. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Resource freshness

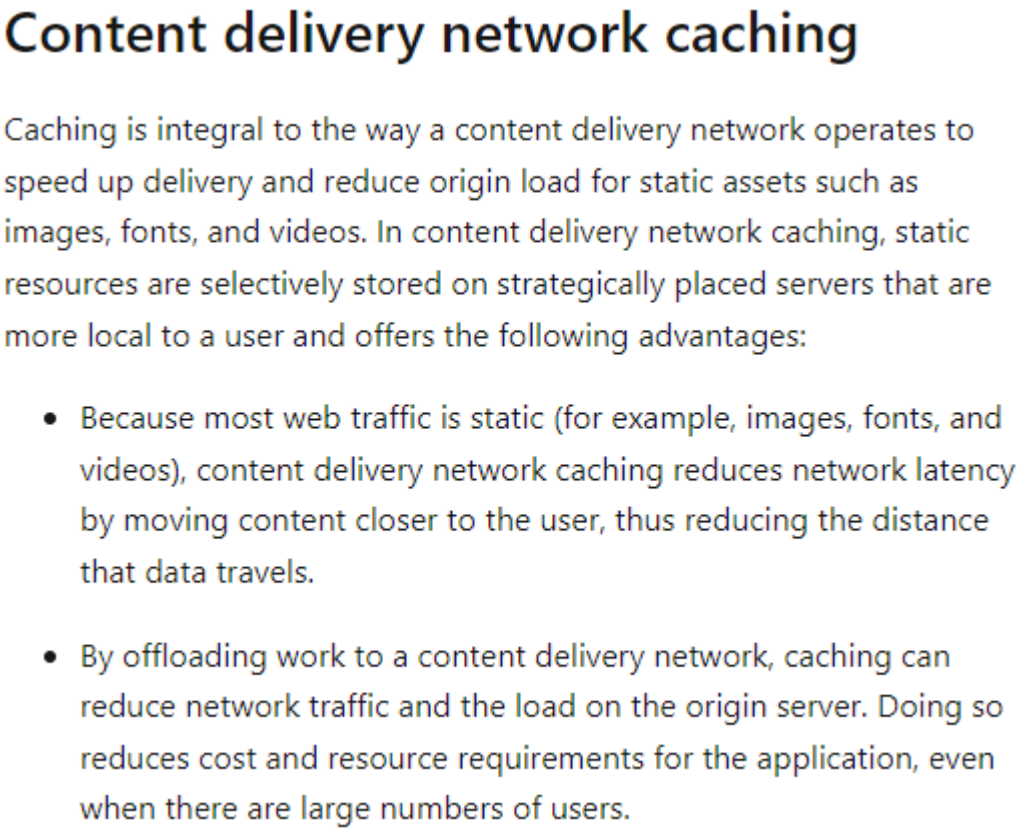
Since a cached resource can potentially be out-of-date, or stale (as compared to the corresponding resource on the origin server), it's important for any caching mechanism to control when content gets a refresh. To save time and bandwidth consumption, a cached resource isn't compared to the version on the origin server every time it's accessed. Instead, as long as a cached resource is considered to be fresh, it's assumed to be the most current version and is sent directly to the client. A cached resource is considered to be fresh when its age is less than the age or period defined by a cache setting. For example, when a browser reloads a web page, it verifies that each cached resource on your hard drive is fresh and loads it. If the resource isn't fresh (stale), an up-to-date copy is loaded from the server.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

Validation

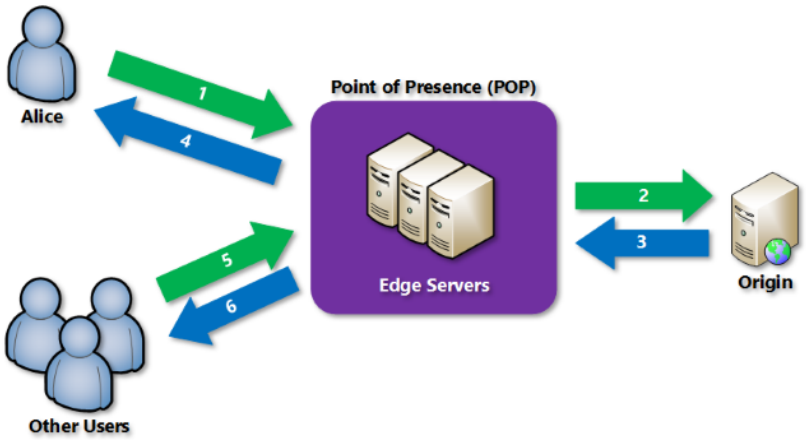
If a resource is considered stale, the origin server gets asked to validate it to determine whether the data in the cache still matches what's on the origin server. If the file has been modified on the origin server, the cache updates its version of the resource. Otherwise, if the resource is fresh, the data is delivered directly from the cache without validating it first.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	 https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works
[1.b] if the resource is not available at the first server or at a peer of the first server, determining if the resource is popular;	If the resource is not available at the first server or at a peer of the first server, Microsoft determines if the resource is popular, for example based on cache eviction policies or other popularity services.

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	ⓘ Note Azure Content Delivery Network makes no guarantees about minimum amount of time that the object will be stored in the cache. Cached contents might be evicted from the content delivery network cache before they are expired if the contents are not requested as frequently to make room for more frequently requested contents. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works ⓘ Note • Azure Front Door makes no guarantees about the amount of time that the content is stored in the cache. Cached content may be removed from the edge cache before the content expiration if the content isn't frequently used. Front Door might be able to serve data from the cache even if the cached data has expired. This behavior can help your site to remain partially available when your origins are offline. • Origins may specify not to cache specific responses using the Cache-Control header with a value of no-cache, private, or no-store. When used in an HTTP response from the origin server to the Azure Front Door POPs, Azure Front Door supports Cache-control directives and honors caching behaviors for Cache-Control directives in RFC 7234 - Hypertext Transfer Protocol (HTTP/1.1): Caching (ietf.org) . https://docs.azure.cn/en-us/frontdoor/front-door-caching

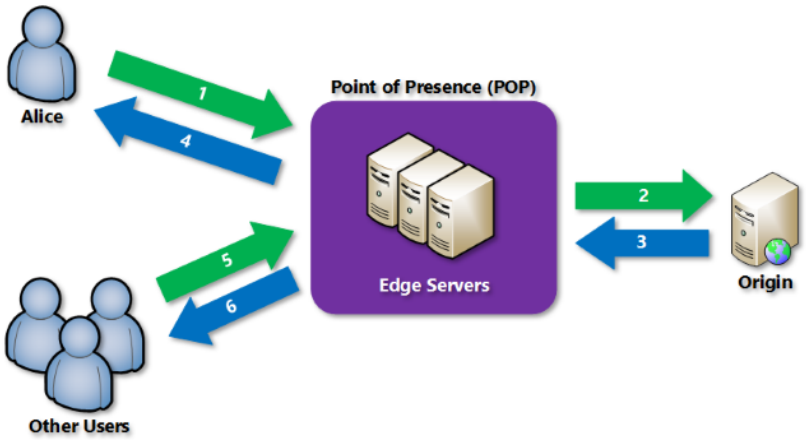
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	# Cache behavior and duration Cache behavior and duration can be configured in Rules Engine. Rules Engine caching configuration always overrides the route configuration. • **When caching is disabled**, Azure Front Door doesn't cache the response contents, irrespective of the origin response directives. • **When caching is enabled**, the cache behavior is different depending on the cache behavior value applied by the Rules Engine: ○ **Honor origin**: Azure Front Door always honors origin response header directive. If the origin directive is missing, Azure Front Door caches contents anywhere from one to three days. ○ **Override always**: Azure Front Door always overrides with the cache duration, meaning that it caches the contents for the cache duration ignoring the values from origin response directives. This behavior only applies if the response is cacheable. ○ **Override if origin missing**: If the origin doesn't return caching TTL values, Azure Front Door uses the specified cache duration. This behavior only applies if the response is cacheable. https://docs.azure.cn/en-us/frontdoor/front-door-caching

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Manage data expiration in a cache In most cases, data that's held in a cache is a copy of data that's held in the original data store. The data in the original data store might change after it was cached, causing the cached data to become stale. Many caching systems enable you to configure the cache to expire data and reduce the period for which data may be out of date. When cached data expires, it's removed from the cache, and the application must retrieve the data from the original data store (it can put the newly fetched information back into cache). You can set a default expiration policy when you configure the cache. In many cache services, you can also stipulate the expiration period for individual objects when you store them programmatically in the cache. Some caches enable you to specify the expiration period as an absolute value, or as a sliding value that causes the item to be removed from the cache if it isn't accessed within the specified time. This setting overrides any cache-wide expiration policy, but only for the specified objects. ⓘ Note Consider the expiration period for the cache and the objects that it contains carefully. If you make it too short, objects will expire too quickly and you will reduce the benefits of using the cache. If you make the period too long, you risk the data becoming stale. It's also possible that the cache might fill up if data is allowed to remain resident for a long time. In this case, any requests to add new items to the cache might cause some items to be forcibly removed in a process known as eviction. Cache services typically evict data on a least-recently-used (LRU) basis, but you can usually override this policy and prevent items from being evicted. However, if you adopt this approach, you risk exceeding the memory that's available in the cache. An application that attempts to add an item to the cache will fail with an exception. Some caching implementations might provide additional eviction policies. There are several types of eviction policies. These include: • A most-recently-used policy (in the expectation that the data won't be required again). • A first-in-first-out policy (oldest data is evicted first). • An explicit removal policy based on a triggered event (such as the data being modified). https://learn.microsoft.com/en-us/azure/architecture/best-practices/caching
[1.c] if the resource is determined to be popular, then the first server obtaining the resource and the first server	If the resource is determined to be popular, for example, based on the cache eviction policy, then Microsoft obtains the resource at the first server and serves the resource to the client via the first server.

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
serving the resource to the client, otherwise,	ⓘ Note Azure Content Delivery Network makes no guarantees about minimum amount of time that the object will be stored in the cache. Cached contents might be evicted from the content delivery network cache before they are expired if the contents are not requested as frequently to make room for more frequently requested contents. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works ⓘ Note • Azure Front Door makes no guarantees about the amount of time that the content is stored in the cache. Cached content may be removed from the edge cache before the content expiration if the content isn't frequently used. Front Door might be able to serve data from the cache even if the cached data has expired. This behavior can help your site to remain partially available when your origins are offline. • Origins may specify not to cache specific responses using the Cache-Control header with a value of no-cache, private, or no-store. When used in an HTTP response from the origin server to the Azure Front Door POPs, Azure Front Door supports Cache-control directives and honors caching behaviors for Cache-Control directives in RFC 7234 - Hypertext Transfer Protocol (HTTP/1.1): Caching (ietf.org) . https://docs.azure.cn/en-us/frontdoor/front-door-caching

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Introduction to caching

Caching is the process of storing data locally so that future requests for that data can be accessed more quickly. In the most common type of caching, web browser caching, a web browser stores copies of static data locally on a local hard drive. By using caching, the web browser can avoid making multiple round-trips to the server and instead access the same data locally, thus saving time and resources. Caching is well-suited for locally managing small, static data such as static images, CSS files, and JavaScript files.

Similarly, caching is used by a content delivery network on edge servers close to the user to avoid requests traveling back to the origin and reducing end-user latency. Unlike a web browser cache, which is used only for a single user, the content delivery network has a shared cache. In a content delivery network shared cache, a file request by a user can be used by another user, which greatly decreases the number of requests to the origin server.

Dynamic resources that change frequently or are unique to an individual user can't be cached. Those types of resources, however, can take advantage of dynamic site acceleration (DSA) optimization on the Azure content delivery network for performance improvements.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Caching can occur at multiple levels between the origin server and the end user: • Web server: Uses a shared cache (for multiple users). • Content delivery network: Uses a shared cache (for multiple users). • Internet service provider (ISP): Uses a shared cache (for multiple users). • Web browser: Uses a private cache (for one user). Each cache typically manages its own resource freshness and performs validation when a file is stale. This behavior is defined in the HTTP caching specification, RFC 7234. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Resource freshness

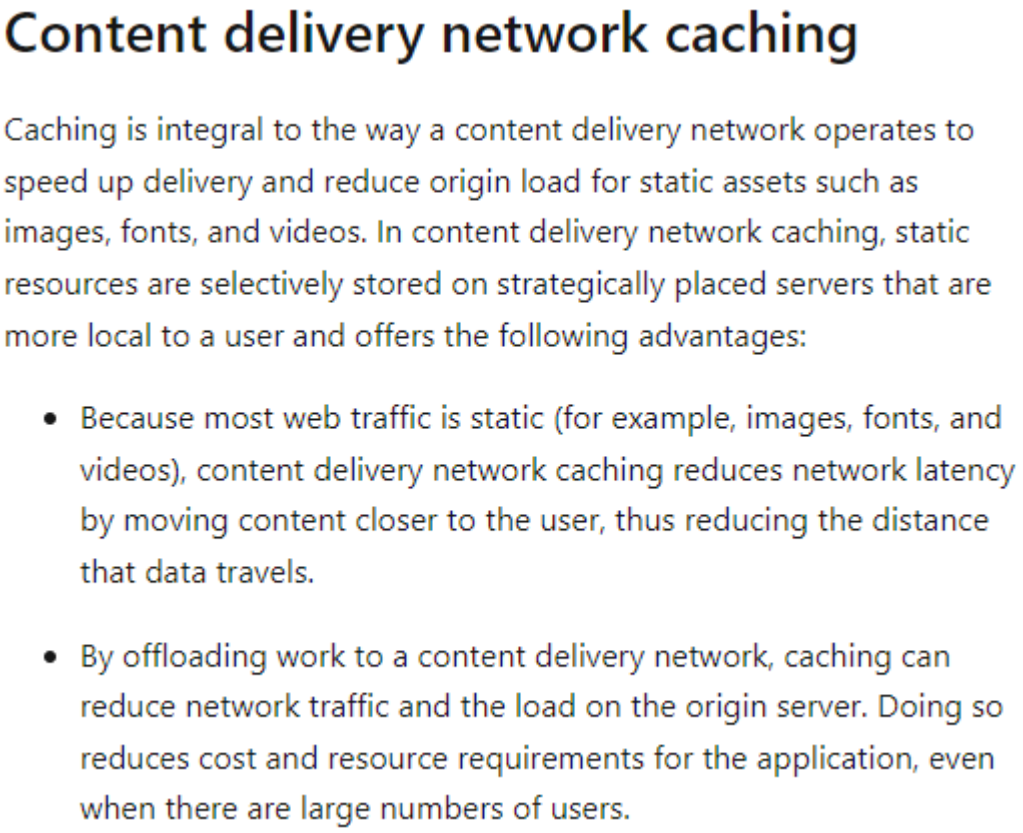
Since a cached resource can potentially be out-of-date, or stale (as compared to the corresponding resource on the origin server), it's important for any caching mechanism to control when content gets a refresh. To save time and bandwidth consumption, a cached resource isn't compared to the version on the origin server every time it's accessed. Instead, as long as a cached resource is considered to be fresh, it's assumed to be the most current version and is sent directly to the client. A cached resource is considered to be fresh when its age is less than the age or period defined by a cache setting. For example, when a browser reloads a web page, it verifies that each cached resource on your hard drive is fresh and loads it. If the resource isn't fresh (stale), an up-to-date copy is loaded from the server.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

Validation

If a resource is considered stale, the origin server gets asked to validate it to determine whether the data in the cache still matches what's on the origin server. If the file has been modified on the origin server, the cache updates its version of the resource. Otherwise, if the resource is fresh, the data is delivered directly from the cache without validating it first.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	 https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works
[1.d] if the resource is determined not to be popular, directing the client to a second server in a second tier of servers distinct from the first tier of servers,	If the resource is determined to not be popular, for example, based on the cache eviction policy, Microsoft directs the client to a second server in a second tier of servers in a second tier of servers distinct from the first tier of servers. If the resource is determined not to be popular, the client is directed to a second server in a second tier of servers (e.g., a parent, origin shield, origin tier, or regional POP) distinct from the first tier of servers (e.g., an edge server).

SentToOriginShield (deprecated) * **See notes on deprecation in the following section.**	If true, it means that request was answered from origin shield cache instead of the edge pop. Origin shield is a parent cache used to improve cache hit ratio.
isReceivedFromClient	If true, it means that the request came from the client. If false, the request is a miss in the edge (child POP) and is responded from origin shield (parent POP).

https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log

Sent to origin shield deprecation

The raw log property **isSentToOriginShield** is deprecated and replaced by a new field **isReceivedFromClient**. Use the new field if you're already using the deprecated field.

Raw logs include logs generated from both CDN edge (child POP) and origin shield. Origin shield refers to parent nodes that are strategically located across the globe. These nodes communicate with origin servers and reduce the traffic load on origin.

For every request that goes to origin shield, there are two log entries:

- One for edge nodes
- One for origin shield.

https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

> ⓘ Note
>
> Azure Content Delivery Network makes no guarantees about minimum amount of time that the object will be stored in the cache. Cached contents might be evicted from the content delivery network cache before they are expired if the contents are not requested as frequently to make room for more frequently requested contents.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

Availability zones

Many Azure regions provide availability zones. Availability zones are independent sets of datacenters that contain isolated power, cooling, and network connections. Availability zones are physically located close enough together to provide a low-latency network, but far enough apart to provide fault isolation from such things as storms and isolated power outages. Most Azure services provide built-in support for availability zones and you can decide how to use them to meet your needs. When you design an Azure solution, you should use availability zones to provide redundancy and fault isolation.

To learn more about availability zones, see What are availability zones?.

https://learn.microsoft.com/en-us/azure/reliability/regions-overview

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	ⓘ Note • Azure Front Door makes no guarantees about the amount of time that the content is stored in the cache. Cached content may be removed from the edge cache before the content expiration if the content isn't frequently used. Front Door might be able to serve data from the cache even if the cached data has expired. This behavior can help your site to remain partially available when your origins are offline. • Origins may specify not to cache specific responses using the Cache-Control header with a value of no-cache, private, or no-store. When used in an HTTP response from the origin server to the Azure Front Door POPs, Azure Front Door supports Cache-control directives and honors caching behaviors for Cache-Control directives in RFC 7234 - Hypertext Transfer Protocol (HTTP/1.1): Caching (ietf.org). https://docs.azure.cn/en-us/frontdoor/front-door-caching
[1.e] wherein the second server comprises a first portion of the content provider's library, the first portion comprising at least the resource, wherein at least one other server in the second tier comprises a second portion of the content provider's library,	Microsoft's second server comprises a first portion of the content provider's library, the first portion comprising at least the resource, wherein at least one other server in the second tier comprises a second portion of the content provider's library. For instance, a content provider's library may be stored across multiple servers in the second tier (e.g., parent or regional POPs or origin tier or origin shield servers). **Sent to origin shield deprecation** The raw log property **isSentToOriginShield** is deprecated and replaced by a new field **isReceivedFromClient**. Use the new field if you're already using the deprecated field. Raw logs include logs generated from both CDN edge (child POP) and origin shield. Origin shield refers to parent nodes that are strategically located across the globe. These nodes communicate with origin servers and reduce the traffic load on origin. For every request that goes to an origin shield, there are two log entries: • One for edge nodes • One for origin shield To differentiate the egress or responses from the edge nodes vs. origin shield, you can use the field **isReceivedFromClient** to get the correct data.

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	https://learn.microsoft.com/en-us/azure/frontdoor/monitor-front-door?pivots=front-door-classic
[1.e.1] wherein the first portion of the content provider's library is distinct from the second portion, and	The first portion of the content provider's library is distinct from the second portion. For instance, a content provider's library may be stored across multiple servers in the second tier (e.g., parent or regional POPs or origin tier or origin shield servers). **Sent to origin shield deprecation** The raw log property **isSentToOriginShield** is deprecated and replaced by a new field **isReceivedFromClient**. Use the new field if you're already using the deprecated field. Raw logs include logs generated from both CDN edge (child POP) and origin shield. Origin shield refers to parent nodes that are strategically located across the globe. These nodes communicate with origin servers and reduce the traffic load on origin. For every request that goes to an origin shield, there are two log entries: • One for edge nodes • One for origin shield To differentiate the egress or responses from the edge nodes vs. origin shield, you can use the field **isReceivedFromClient** to get the correct data. https://learn.microsoft.com/en-us/azure/frontdoor/monitor-front-door?pivots=front-door-classic
[1.e.2] wherein the second tier is any intermediate tier between the first tier and an origin server that stores resources associated with the content provider's library, and	The second tier is any intermediate tier between the first tier and an origin server that stores resources associated with the content provider's library. For instance, Microsoft Azure CDN and/or Azure Front Door provide an intermediate tier of servers (e.g., parent or regional POPs or origin tier or origin shield) between the first tier (e.g., edge nodes or edge POPs) and an origin server.

SentToOriginShield (deprecated) * **See notes on deprecation in the following section.**	If true, it means that request was answered from origin shield cache instead of the edge pop. Origin shield is a parent cache used to improve cache hit ratio.
isReceivedFromClient	If true, it means that the request came from the client. If false, the request is a miss in the edge (child POP) and is responded from origin shield (parent POP).

https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log

Sent to origin shield deprecation

The raw log property **isSentToOriginShield** is deprecated and replaced by a new field **isReceivedFromClient**. Use the new field if you're already using the deprecated field.

Raw logs include logs generated from both CDN edge (child POP) and origin shield. Origin shield refers to parent nodes that are strategically located across the globe. These nodes communicate with origin servers and reduce the traffic load on origin.

For every request that goes to origin shield, there are two log entries:

- One for edge nodes
- One for origin shield.

https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log

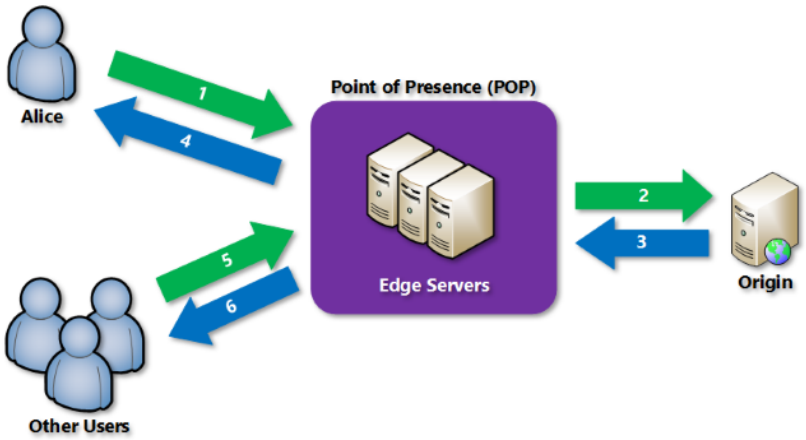
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	# Introduction to caching Caching is the process of storing data locally so that future requests for that data can be accessed more quickly. In the most common type of caching, web browser caching, a web browser stores copies of static data locally on a local hard drive. By using caching, the web browser can avoid making multiple round-trips to the server and instead access the same data locally, thus saving time and resources. Caching is well-suited for locally managing small, static data such as static images, CSS files, and JavaScript files. Similarly, caching is used by a content delivery network on edge servers close to the user to avoid requests traveling back to the origin and reducing end-user latency. Unlike a web browser cache, which is used only for a single user, the content delivery network has a shared cache. In a content delivery network shared cache, a file request by a user can be used by another user, which greatly decreases the number of requests to the origin server. Dynamic resources that change frequently or are unique to an individual user can't be cached. Those types of resources, however, can take advantage of dynamic site acceleration (DSA) optimization on the Azure content delivery network for performance improvements. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Caching can occur at multiple levels between the origin server and the end user: • Web server: Uses a shared cache (for multiple users). • Content delivery network: Uses a shared cache (for multiple users). • Internet service provider (ISP): Uses a shared cache (for multiple users). • Web browser: Uses a private cache (for one user). Each cache typically manages its own resource freshness and performs validation when a file is stale. This behavior is defined in the HTTP caching specification, RFC 7234. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Content delivery network caching Caching is integral to the way a content delivery network operates to speed up delivery and reduce origin load for static assets such as images, fonts, and videos. In content delivery network caching, static resources are selectively stored on strategically placed servers that are more local to a user and offers the following advantages: • Because most web traffic is static (for example, images, fonts, and videos), content delivery network caching reduces network latency by moving content closer to the user, thus reducing the distance that data travels. • By offloading work to a content delivery network, caching can reduce network traffic and the load on the origin server. Doing so reduces cost and resource requirements for the application, even when there are large numbers of users. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works
[1.f] the second server serving the resource to the client.	Microsoft's second server serves the resource to the client. For instance, the second server (e.g., a parent or regional POP or origin tier or origin shield) serves the requested object to the client.

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

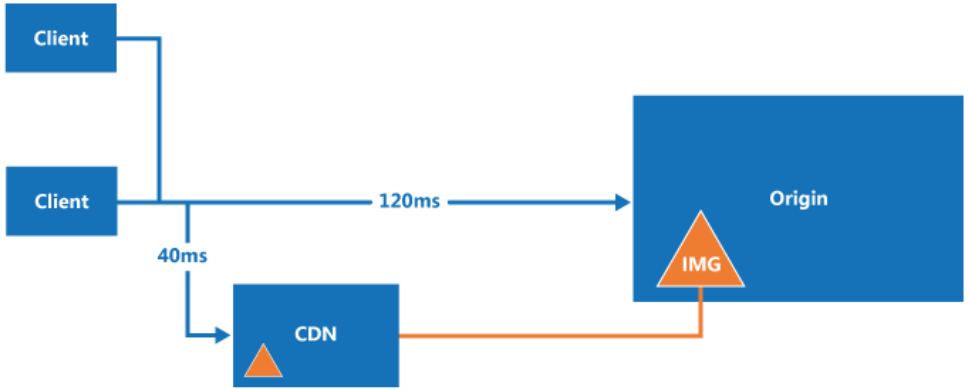
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



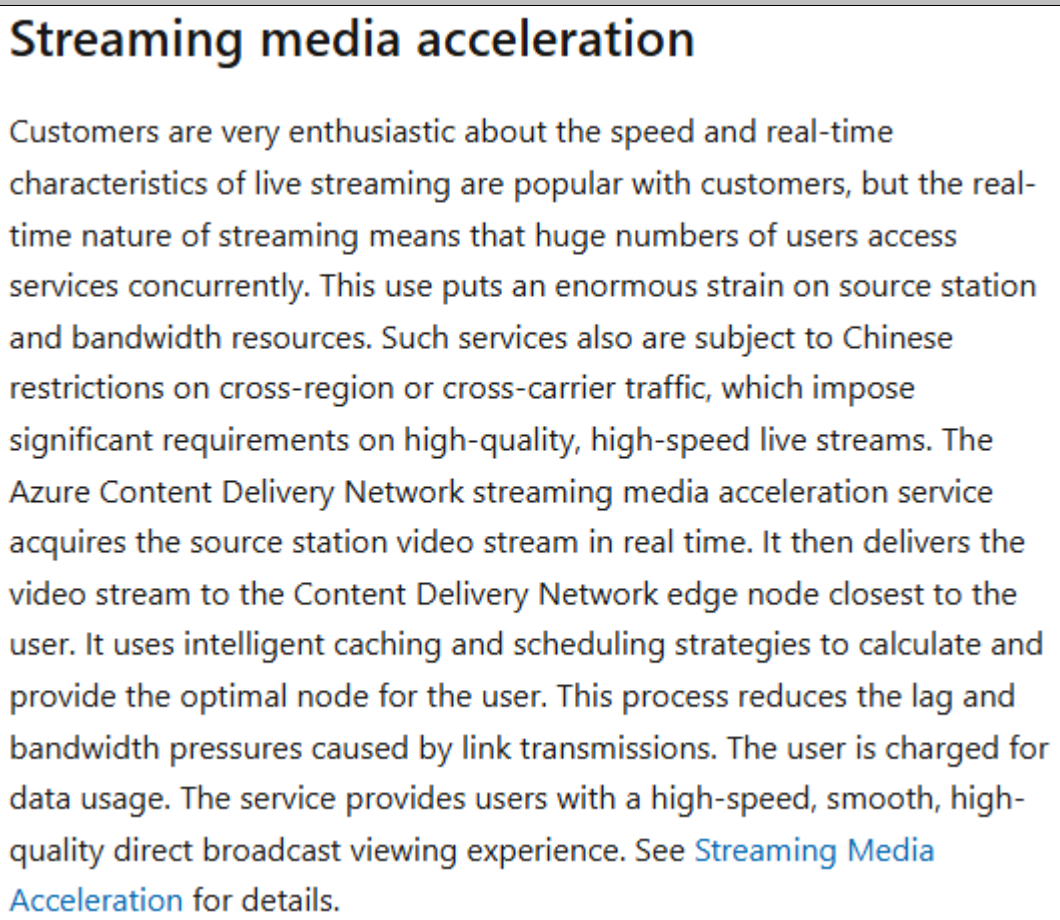
In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	SentToOriginShield (deprecated) * **See notes on deprecation in the following section.** — If true, it means that request was answered from origin shield cache instead of the edge pop. Origin shield is a parent cache used to improve cache hit ratio. isReceivedFromClient — If true, it means that the request came from the client. If false, the request is a miss in the edge (child POP) and is responded from origin shield (parent POP). https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log
2. A method as recited in claim 1, wherein portions of the content provider's library are logically partitioned across servers in the second tier.	Microsoft performs the method of claim 1, wherein portions of the content provider's library are logically partitioned across servers in the second tier (e.g. based on content type, file size, traffic behavior, etc.). See Claim limitation 1[e].
4. A method as recited in claim 1, wherein the content provider's library comprises a plurality of content-types, each resource in the library being associated with one of the plurality of content-types.	Microsoft performs the method of claim 1, wherein the content provider's library comprises a plurality of content-types, each resource in the library being associated with one of the plurality of content-types. For instance, Azure CDN and/or Azure Front Door support a plurality of content types including "audio, video, apps, photos, and other files" of a content provider's library. The Azure Content Delivery Network is designed to send audio, video, apps, photos, and other files to your customers faster and more reliably, using the servers closest to each user. Acceleration Data Transfer provides dynamic site acceleration of non-cacheable, dynamic content generated by your web applications. https://azure.microsoft.com/en-us/pricing/details/cdn/

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Website and small file acceleration One classic usage scenario for Content Delivery Network is providing acceleration for so-called "small files" (for example, HTML webpage files, image files, JavaScript files, or CSS files) used by many websites. Acceleration allows the website to deliver a better user experience and thereby increase user visits and ultimately drive up revenues for the entire business. The typical user group is small, medium, and large enterprises that provide website services for Internet users. See Web Acceleration for details. https://docs.azure.cn/en-us/cdn/cdn-overview **Download delivery for large files** Another typical usage scenario for Content Delivery Network is to facilitate multinode delivery of large file downloads, which ultimately ensures that the download experience proceeds smoothly. Without the use of Content Delivery Network services, typical user scenarios, such as operating system and firmware upgrades, publishing new games online (which require downloading client installation packages), and mobile app updates, would have a huge impact on source station bandwidth usage. Such heavy use could even cause the source station to stop working. See Download Acceleration for details. https://docs.azure.cn/en-us/cdn/cdn-overview

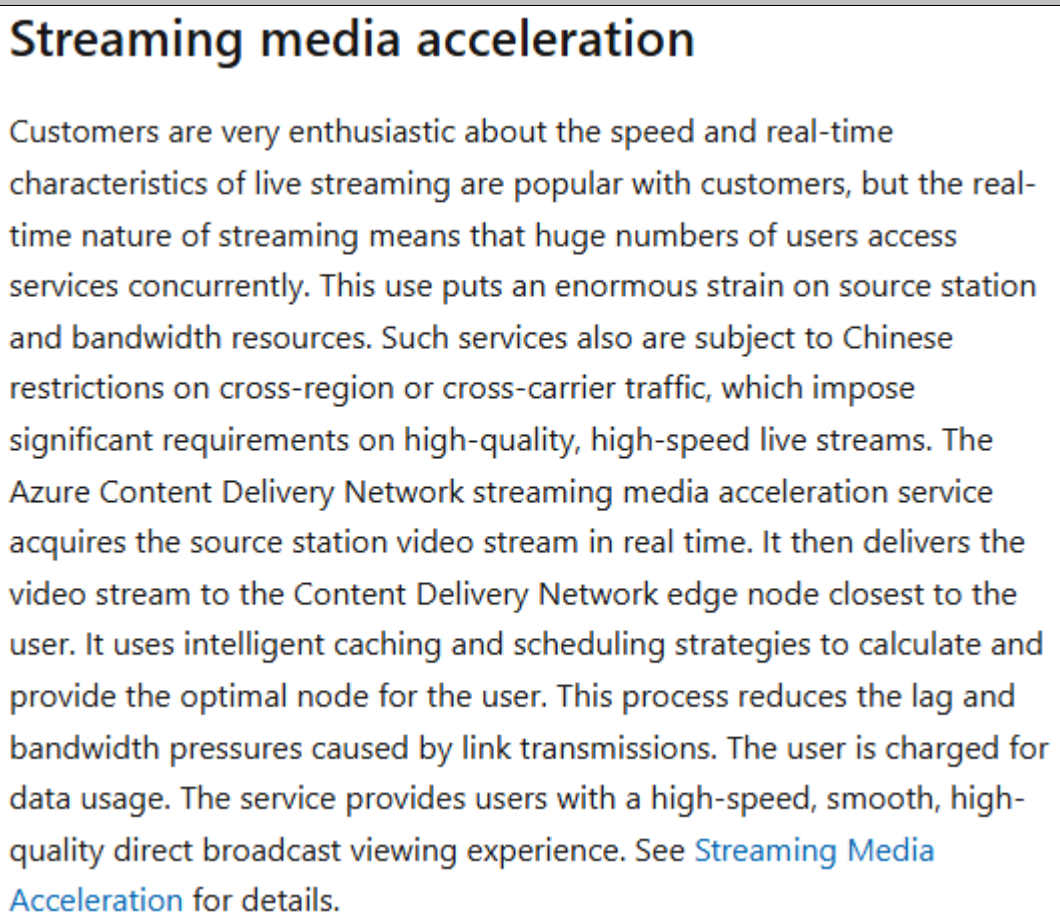
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	## VOD Acceleration As the range of online video and media services has grown over the last few years, increasing numbers of people have become used to using Internet platforms to watch videos and listen to audio content. The limitations on the Internet environment in China places huge demands on the final delivery of audio and video content. The typical user group is all types of media website and mobile app clients that provide services to Internet users. See VOD Acceleration for details. https://docs.azure.cn/en-us/cdn/cdn-overview ## Image Acceleration With Content Delivery Network Image Processing, you can use Content Delivery Network services to obtain processed versions of images from any location and any device at any time. See Image Acceleration for details. https://docs.azure.cn/en-us/cdn/cdn-overview

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	 https://docs.azure.cn/en-us/cdn/cdn-overview
5. A method as recited in claim 4 further comprising storing distinct portions of the content provider's library on two or more servers in the second tier, wherein the content-types on	Microsoft performs the method of claim further comprising storing distinct portions of the content provider's library on two or more servers in the second tier, wherein the content-types on each of the two or more servers are proportionally distributed, the content-types comprising at least one of audio content, video content, gaming content, and software content. For instance, Azure CDN and/or Front Door support proportional distribution of content types including audio content, video content, gaming content, and software content.

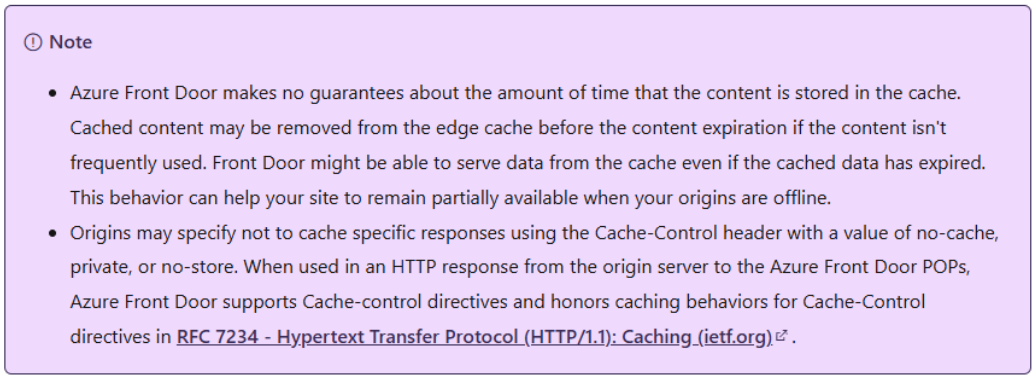
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
each of the two or more servers are proportionally distributed, the content-types comprising at least one of audio content, video content, gaming content, and software content.	The Azure Content Delivery Network is designed to send audio, video, apps, photos, and other files to your customers faster and more reliably, using the servers closest to each user. Acceleration Data Transfer provides dynamic site acceleration of non-cacheable, dynamic content generated by your web applications. https://azure.microsoft.com/en-us/pricing/details/cdn/ **Website and small file acceleration** One classic usage scenario for Content Delivery Network is providing acceleration for so-called "small files" (for example, HTML webpage files, image files, JavaScript files, or CSS files) used by many websites. Acceleration allows the website to deliver a better user experience and thereby increase user visits and ultimately drive up revenues for the entire business. The typical user group is small, medium, and large enterprises that provide website services for Internet users. See Web Acceleration for details. https://docs.azure.cn/en-us/cdn/cdn-overview

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	## Download delivery for large files Another typical usage scenario for Content Delivery Network is to facilitate multinode delivery of large file downloads, which ultimately ensures that the download experience proceeds smoothly. Without the use of Content Delivery Network services, typical user scenarios, such as operating system and firmware upgrades, publishing new games online (which require downloading client installation packages), and mobile app updates, would have a huge impact on source station bandwidth usage. Such heavy use could even cause the source station to stop working. See Download Acceleration for details. https://docs.azure.cn/en-us/cdn/cdn-overview ## VOD Acceleration As the range of online video and media services has grown over the last few years, increasing numbers of people have become used to using Internet platforms to watch videos and listen to audio content. The limitations on the Internet environment in China places huge demands on the final delivery of audio and video content. The typical user group is all types of media website and mobile app clients that provide services to Internet users. See VOD Acceleration for details. https://docs.azure.cn/en-us/cdn/cdn-overview

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Image Acceleration With Content Delivery Network Image Processing, you can use Content Delivery Network services to obtain processed versions of images from any location and any device at any time. See Image Acceleration for details. https://docs.azure.cn/en-us/cdn/cdn-overview

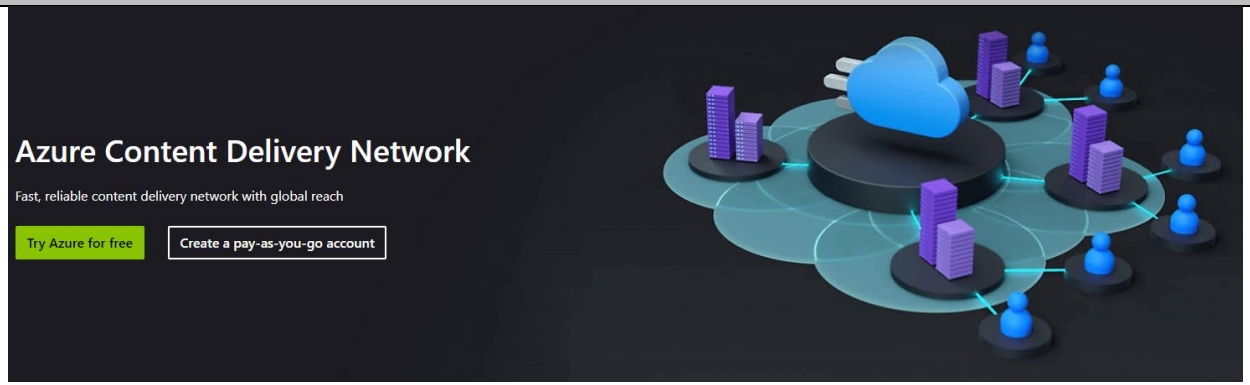
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	 https://docs.azure.cn/en-us/cdn/cdn-overview See also Claim 4.
7. A method as recited in claim 1, wherein the step of directing the client to a second server in a second tier is performed in	Microsoft performs the method of claim 1, wherein the step of directing the client to a second server in a second tier is performed in response to a step of using a hash to determine which second server in the second tier stores the first portion of the content provider's library. For instance, on information and belief, Azure CDN and/or

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
response to a step of using a hash to determine which second server in the second tier stores the first portion of the content provider's library.	Azure Front Door use a hash of the request URL to compute a cache key used to determine whether a particular object is stored in the cache. **Property** — **Supported values** Query string caching behavior: • **Ignore Query String:** Query strings aren't considered when the cache key gets generated. In ARM templates, set the `queryStringCachingBehavior` property to `IgnoreQueryString`. • **Use query string:** Each unique URL has its own cache key. In ARM templates, use the `queryStringCachingBehavior` of `UseQueryString`. • **Ignore specified query string:** Query strings specified in the parameters get excluded when the cache key gets generated. In ARM templates, set the `queryStringCachingBehavior` property to `IgnoreSpecifiedQueryStrings`. • **Include specified query string:** Query strings specified in the parameters get included when the cache key gets generated. In ARM templates, set the `queryStringCachingBehavior` property to `IncludeSpecifiedQueryStrings`. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-rules-engine-actions?utm_source=chatgpt.com&tabs=portal&pivots=front-door-standard-premium
8. A method as recited in claim 1 wherein the step of determining whether the resource is popular comprises determining whether a current popularity value for the resource exceeds a popularity threshold.	Microsoft performs the method of claim 1 wherein the step of determining whether the resource is popular comprises determining whether a current popularity value for the resource exceeds a popularity threshold. For instance, Microsoft determines if a popularity value for the resource exceeds a popularity threshold (e.g. via a cache eviction policy). ⓘ Note Azure Content Delivery Network makes no guarantees about minimum amount of time that the object will be stored in the cache. Cached contents might be evicted from the content delivery network cache before they are expired if the contents are not requested as frequently to make room for more frequently requested contents.

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works  Note • Azure Front Door makes no guarantees about the amount of time that the content is stored in the cache. Cached content may be removed from the edge cache before the content expiration if the content isn't frequently used. Front Door might be able to serve data from the cache even if the cached data has expired. This behavior can help your site to remain partially available when your origins are offline. • Origins may specify not to cache specific responses using the Cache-Control header with a value of no-cache, private, or no-store. When used in an HTTP response from the origin server to the Azure Front Door POPs, Azure Front Door supports Cache-control directives and honors caching behaviors for Cache-Control directives in RFC 7234 - Hypertext Transfer Protocol (HTTP/1.1): Caching (ietf.org). https://docs.azure.cn/en-us/frontdoor/front-door-caching
15[Pre]. A method of content delivery in a content delivery network comprising a plurality of tiers of servers, including at least a first tier of servers and a second tier of servers, the method comprising:	Microsoft performs a method of content delivery in a content delivery network comprising a plurality of tiers of servers, including at least a first tier of servers and a second tier of servers as set forth below. See Claim limitation 1.Pre.
[15.a] at a first server in one tier of the tiers of servers, obtaining a request from a client for a resource, wherein the resource is available as part of a content provider's library;	At a first server in one tier of the tiers of servers, Microsoft obtains a request from a client for a resource, wherein the resource is available as part of a content provider's library. See Claim limitation 1.a.
[15.b] if the resource is not available at the first server or at a peer of the first server, then selectively redirecting the request from the client to a second tier server,	If the resource is not available at the first server or at a peer of the first server, then Microsoft selectively redirects the request from the client to a second tier server. See Claim limitations 1.b, 1.c, and 1.d.

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
[15.b.1] the second tier server being in a tier distinct from the tier containing the first server,	The second tier server is in a tier distinct from the tier containing the first server. See Claim limitation 1.d.
[15.b.2] the redirecting being based, at least in part, on a current popularity value for the resource;	The redirecting is based at least in part on a current popularity value for the resource. Azure CDN and/or Azure Front Door redirect requests based at least in part on a current popularity value for the resource. ⓘ Note Azure Content Delivery Network makes no guarantees about minimum amount of time that the object will be stored in the cache. Cached contents might be evicted from the content delivery network cache before they are expired if the contents are not requested as frequently to make room for more frequently requested contents. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works ⓘ Note • Azure Front Door makes no guarantees about the amount of time that the content is stored in the cache. Cached content may be removed from the edge cache before the content expiration if the content isn't frequently used. Front Door might be able to serve data from the cache even if the cached data has expired. This behavior can help your site to remain partially available when your origins are offline. • Origins may specify not to cache specific responses using the Cache-Control header with a value of no-cache, private, or no-store. When used in an HTTP response from the origin server to the Azure Front Door POPs, Azure Front Door supports Cache-control directives and honors caching behaviors for Cache-Control directives in RFC 7234 - Hypertext Transfer Protocol (HTTP/1.1): Caching (ietf.org) . https://docs.azure.cn/en-us/frontdoor/front-door-caching
[15.c] wherein the second tier server comprises a first portion of the content provider's	The second tier server comprises a first portion of the content provider's library and the first portion comprises at least the resource.

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
library, the first portion comprising at least the resource,	See Claim limitation 1.e.
[15.c.1] wherein at least one other second tier server comprises a second portion of the content provider's library distinct from the first portion, and	At least one other second tier server comprises a second portion of the content provider's library distinct from the first portion. See Claim limitations 1.e and 1.e.1.
[15.c.2] wherein the second tier of servers is any intermediate tier between the first tier and an origin server that stores the content provider's library, and	The second tier of servers is any intermediate tier between the first tier and an origin server that stores the content provider's library. See Claim limitation 1.e.2.
[15.d] wherein the step of selectively redirecting the request from the client to a second tier server comprises determining to which second tier server the request should be directed,	The step of selectively redirecting the request from the client to a second tier server comprises determining to which second tier server the request should be directed. For instance, Azure CDN and/or Azure Front Door determine to which second tier server the request should be directed. See Claim limitation 1.d.
[15.d.1] the determining step using a hash function to identify a second tier server that stores the first portion of the content provider's library.	The determining step uses a hash function to identify a second tier server that stores the first portion of the content provider's library. For instance, Azure CDN and/or Azure Front Door use a hash of the request URL to compute a cache key used to determine whether a particular object is stored in the cache. See Claim 7.
17. A method as recited in claim 15 wherein the first tier of servers comprises a plurality of edge servers and the second tier of servers comprises a plurality of parent servers, and wherein the first server is an edge server in the first tier and	Microsoft performs the method of claim 15 wherein the first tier of servers comprises a plurality of edge servers and the second tier of servers comprises a plurality of parent servers, and wherein the first server is an edge server in the first tier and the second server is a parent server in the second tier. For instance, Microsoft Azure CDN and/or Azure Front Door comprise at least a first tier of servers (e.g., edge nodes or edge POPs) and a second tier of servers (e.g., parent or regional POPs or origin shield servers).

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
the second server is a parent server in the second tier.	 https://azure.microsoft.com/en-us/products/cdn **Increase speed and scalability for an optimal user experience** Azure Content Delivery Network offers a global solution for rapidly delivering content. Save bandwidth and improve responsiveness when encoding or distributing gaming software, firmware updates, and IoT endpoints. Reduce load times for websites, mobile apps, and streaming media to increase user satisfaction globally. https://azure.microsoft.com/en-us/products/cdn Stream media and download large files quickly with optimized delivery. Provide a scalable solution to handle worldwide traffic spikes and instantaneous high loads. Integrate seamlessly with your Azure services to activate within minutes. Protect content with custom domain HTTPS, DDoS, and WAF protection. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Improved performance and user experience

Users expect a fast, reliable, and personalized web experience wherever they are. Content Delivery Network helps reduce latency and improve performance for high-bandwidth content by distributing user requests and serving content directly from edge servers. This brings the content closer to users and sends less traffic to the origin point, delivering superior online experiences.

https://azure.microsoft.com/en-us/products/cdn

Global coverage and massive scalability

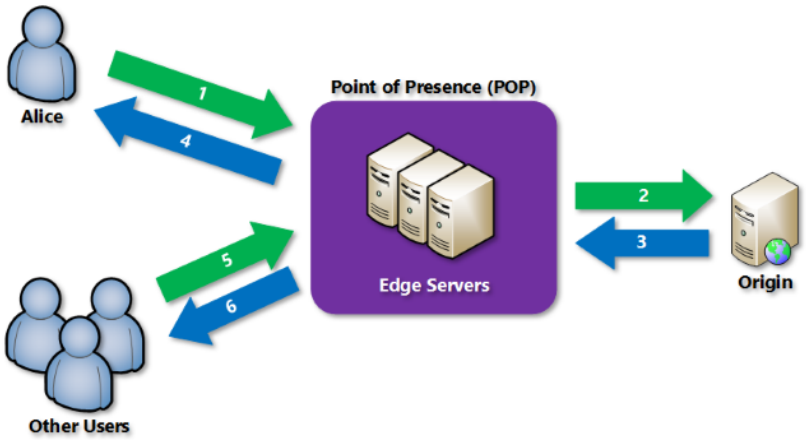
Scale on the fly with no downtime using a platform that responds to traffic fluctuations. A distributed, global presence enables Content Delivery Network to handle sudden traffic spikes and heavy loads—for example, during major product launches or global sporting events—without new infrastructure costs or capacity concerns. Global, distributed points of presence (PoPs) ensure fast delivery of content and protects the origin from excessive load.

https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Protection at the edge Optimize security through features like custom domain HTTPS and DDoS mitigation. Enable Content Delivery Network to put edge servers on the front line serving and protect the origin server from malicious attacks. Set advanced rules to block requests or validate users and use geo-filtering to ensure the content is delivered only to specific regions. https://azure.microsoft.com/en-us/products/cdn **Automate processes and workflows with DevOps** Azure APIs and developer tools give you the performance, reliability, and security needed to build and automate global applications fast. A unified API allows you to deploy Content Delivery Network endpoints from each provider and still manage them the same way. Azure APIs also provide the flexibility to optimize features for availability and performance. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

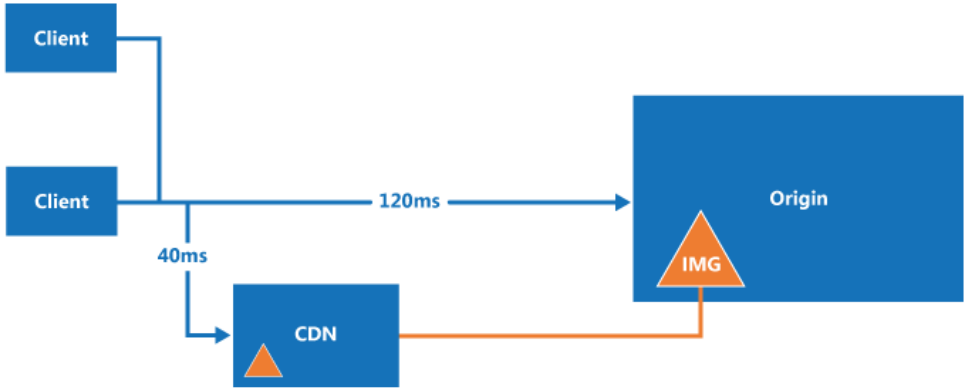
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	This article lists current metros containing point of presence (POP) locations, sorted by region, for Azure Content Delivery Network products. Each metro might contain more than one POP. For example, Azure Content Delivery Network from Microsoft has 192 POPs across 109 metro cities. ⓘ **Important** Each Azure Content Delivery Network product has a distinct way of building its content delivery network infrastructures, hence Microsoft recommends against using POP locations to decide which Azure Content Delivery Network product to use. Instead, you should consider its features and end-user performance. Test the performance with each Azure Content Delivery Network product to choose the right product for your users. https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Region	Cities
North America	Etobicoke, Canada (2) Montreal, Canada Vancouver, Canada (2) Querétaro, Mexico (2) Atlanta, GA, USA (3) Boydton, VA, USA (2) Chaska, MN, USA (2) Cheyenne, WY, USA (2) Chicago, IL, USA (4) Dallas, TX, USA (4) Des Moines,IA, USA (3) Detroit, MI, USA Englewood, CO, USA (2) Honolulu, HI, USA Houston, TX, USA (3) Jacksonville, FL, USA (2) Las Vegas, NV, USA (3) Los Angeles, CA, USA (2) Manassas, VA, USA (3) Memphis, TN, USA Miami, FL, USA (4) Minneapolis, MN, USA (2) Needham Heights, MA, USA (2) Nashville, TN, USA Newark, NJ, USA New York, NY, USA (2) Philadelphia, PA, USA Phoenix, AZ, USA Plano, TX, USA Portland, OR, USA Quincy, WA, USA (3) San Antonio, TX, USA (4) San Jose, CA, USA (4) Salt Lake City, UT, USA (3) Seattle, WA, USA (2) Secaucus, NJ, USA (2) Southfield, MI, USA Tempe, AZ, USA

https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Edge Nodes - List

Reference

Feedback

Service: CDN

API Version: 2024-02-01

Edgenodes are the global Point of Presence (POP) locations used to deliver CDN content to end users.

HTTP Copy Try It

```
GET https://management.azure.com/providers/Microsoft.Cdn/edgenodes?api-version=2024-02-01
```

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

EdgeNode

Edgenode is a global Point of Presence (POP) location used to deliver CDN content to end users.

Expand table

Name	Type	Description
id	string	Resource ID.
name	string	Resource name.
properties.ipAddressGroups	IpAddressGroup[]	List of ip address groups.
systemData	SystemData	Read only system data
type	string	Resource type.

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Connectivity within Microsoft's network along with new Regional Caching capabilities enables more consistent, more predictable cache fill performance by providing multi-tier caching along direct, private access to content in Azure from each CDN edge point of presence (POP). Azure CDN from Microsoft entered public preview providing access to 54 global Edge POPs in 33 countries and 16 Regional Cache POPs.  https://azure.microsoft.com/es-es/blog/new-locations-for-azure-cdn-now-available/

SentToOriginShield (deprecated) * **See notes on deprecation in the following section.**	If true, it means that request was answered from origin shield cache instead of the edge pop. Origin shield is a parent cache used to improve cache hit ratio.
isReceivedFromClient	If true, it means that the request came from the client. If false, the request is a miss in the edge (child POP) and is responded from origin shield (parent POP).

https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log

Sent to origin shield deprecation

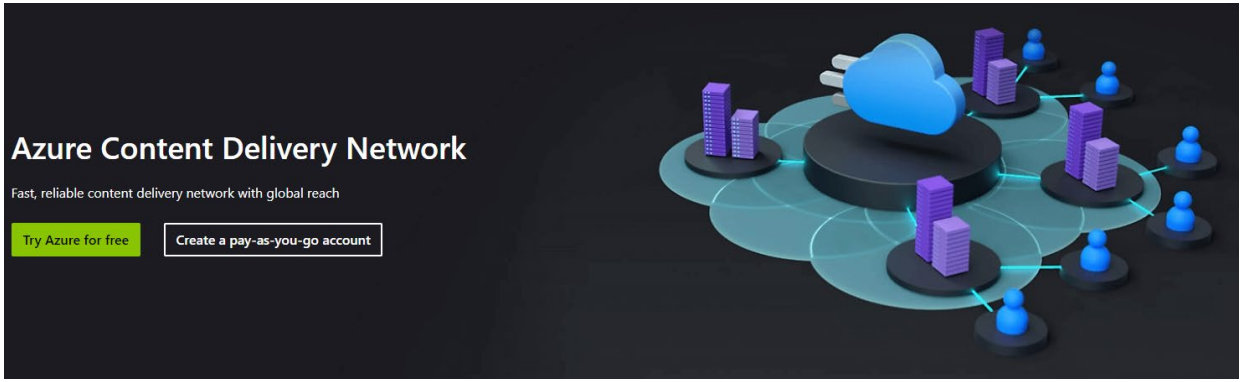
The raw log property **isSentToOriginShield** is deprecated and replaced by a new field **isReceivedFromClient**. Use the new field if you're already using the deprecated field.

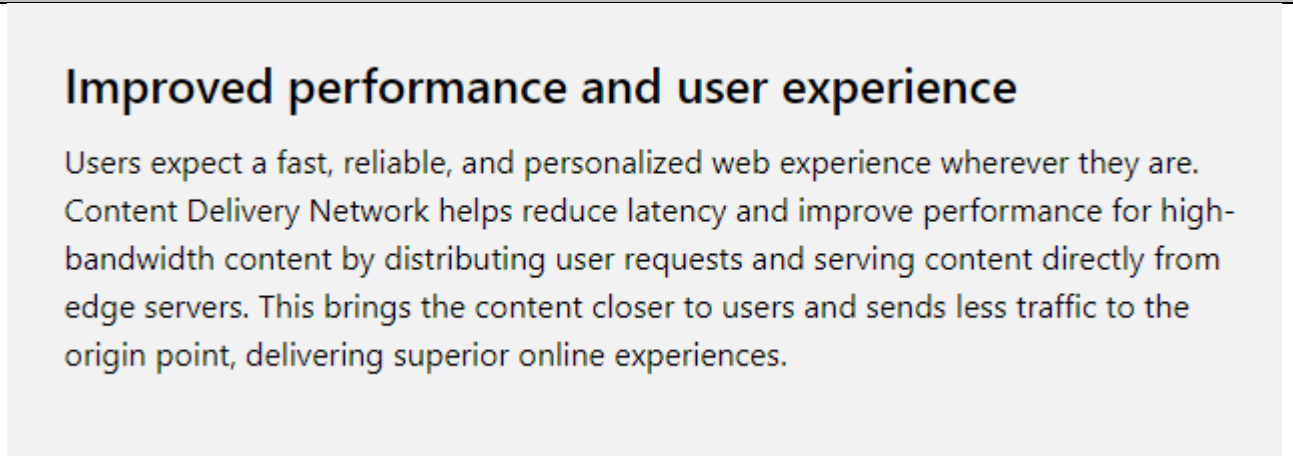
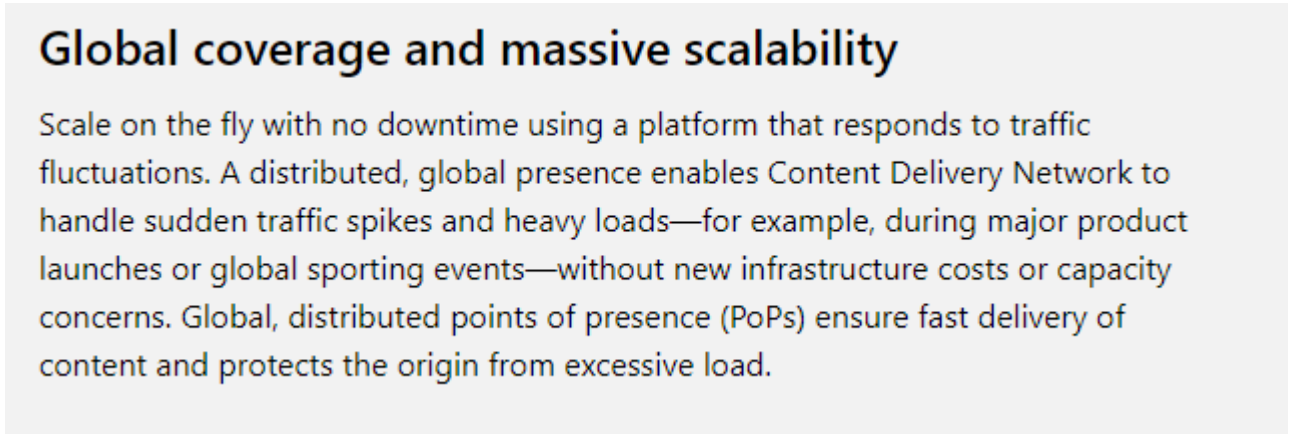
Raw logs include logs generated from both CDN edge (child POP) and origin shield. Origin shield refers to parent nodes that are strategically located across the globe. These nodes communicate with origin servers and reduce the traffic load on origin.

For every request that goes to origin shield, there are two log entries:

- One for edge nodes
- One for origin shield.

https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log

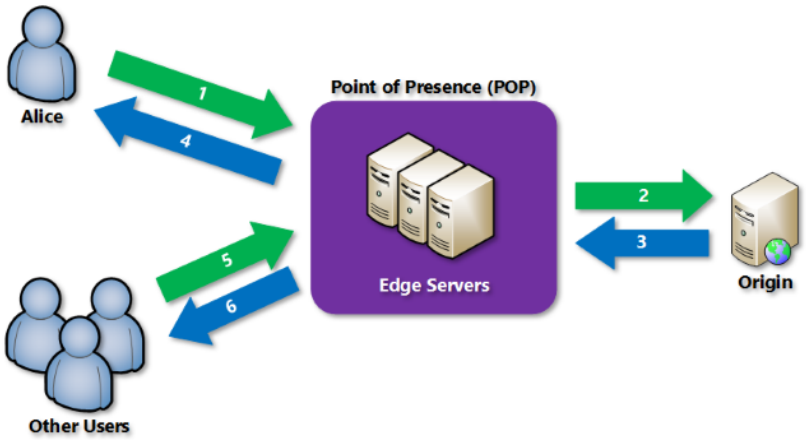
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
18. A method as recited in claim 15 wherein the first tier of servers comprises a plurality of edge servers and the second tier of servers comprises a plurality of parent servers, and wherein the plurality of tiers of servers further comprises an origin tier, and wherein the first server is a parent server in the second tier and the second server is a server in the origin tier.	Microsoft performs the method of claim 15 wherein the first tier of servers comprises a plurality of edge servers and the second tier of servers comprises a plurality of parent servers, and wherein the plurality of tiers of servers further comprises an origin tier, and wherein the first server is a parent server in the second tier and the second server is a server in the origin tier. For instance, Microsoft Azure CDN and/or Azure Front Door comprise at least a first tier of servers (e.g., edge nodes or edge POPs), a second tier of servers (e.g., parent or regional POPs), and an origin tier.  https://azure.microsoft.com/en-us/products/cdn **Increase speed and scalability for an optimal user experience** Azure Content Delivery Network offers a global solution for rapidly delivering content. Save bandwidth and improve responsiveness when encoding or distributing gaming software, firmware updates, and IoT endpoints. Reduce load times for websites, mobile apps, and streaming media to increase user satisfaction globally. https://azure.microsoft.com/en-us/products/cdn Stream media and download large files quickly with optimized delivery. Provide a scalable solution to handle worldwide traffic spikes and instantaneous high loads. Integrate seamlessly with your Azure services to activate within minutes. Protect content with custom domain HTTPS, DDoS, and WAF protection. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	 https://azure.microsoft.com/en-us/products/cdn  https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Protection at the edge Optimize security through features like custom domain HTTPS and DDoS mitigation. Enable Content Delivery Network to put edge servers on the front line serving and protect the origin server from malicious attacks. Set advanced rules to block requests or validate users and use geo-filtering to ensure the content is delivered only to specific regions. https://azure.microsoft.com/en-us/products/cdn **Automate processes and workflows with DevOps** Azure APIs and developer tools give you the performance, reliability, and security needed to build and automate global applications fast. A unified API allows you to deploy Content Delivery Network endpoints from each provider and still manage them the same way. Azure APIs also provide the flexibility to optimize features for availability and performance. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

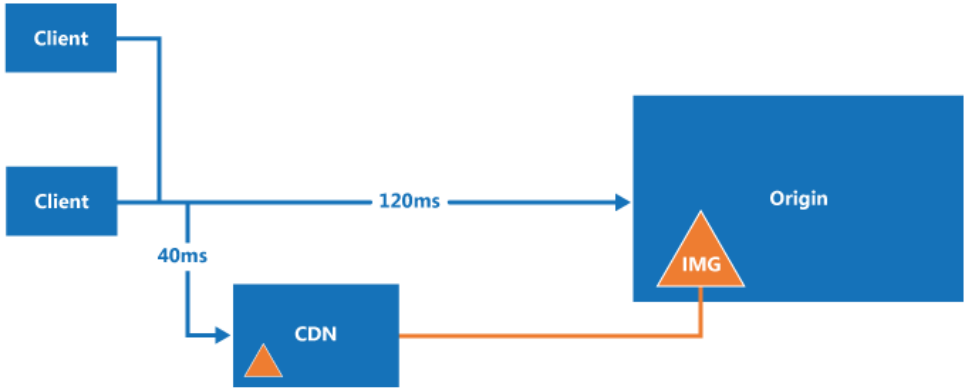
U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	This article lists current metros containing point of presence (POP) locations, sorted by region, for Azure Content Delivery Network products. Each metro might contain more than one POP. For example, Azure Content Delivery Network from Microsoft has 192 POPs across 109 metro cities. ⓘ **Important** Each Azure Content Delivery Network product has a distinct way of building its content delivery network infrastructures, hence Microsoft recommends against using POP locations to decide which Azure Content Delivery Network product to use. Instead, you should consider its features and end-user performance. Test the performance with each Azure Content Delivery Network product to choose the right product for your users. https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Region	Cities
North America	Etobicoke, Canada (2) Montreal, Canada Vancouver, Canada (2) Querétaro, Mexico (2) Atlanta, GA, USA (3) Boydton, VA, USA (2) Chaska, MN, USA (2) Cheyenne, WY, USA (2) Chicago, IL, USA (4) Dallas, TX, USA (4) Des Moines,IA, USA (3) Detroit, MI, USA Englewood, CO, USA (2) Honolulu, HI, USA Houston, TX, USA (3) Jacksonville, FL, USA (2) Las Vegas, NV, USA (3) Los Angeles, CA, USA (2) Manassas, VA, USA (3) Memphis, TN, USA Miami, FL, USA (4) Minneapolis, MN, USA (2) Needham Heights, MA, USA (2) Nashville, TN, USA Newark, NJ, USA New York, NY, USA (2) Philadelphia, PA, USA Phoenix, AZ, USA Plano, TX, USA Portland, OR, USA Quincy, WA, USA (3) San Antonio, TX, USA (4) San Jose, CA, USA (4) Salt Lake City, UT, USA (3) Seattle, WA, USA (2) Secaucus, NJ, USA (2) Southfield, MI, USA Tempe, AZ, USA

https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities

Edge Nodes - List

Reference

Feedback

Service: CDN

API Version: 2024-02-01

Edgenodes are the global Point of Presence (POP) locations used to deliver CDN content to end users.

HTTP Copy Try It

```
GET https://management.azure.com/providers/Microsoft.Cdn/edgenodes?api-version=2024-02-01
```

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

EdgeNode

Edgenode is a global Point of Presence (POP) location used to deliver CDN content to end users.

Expand table

Name	Type	Description
id	string	Resource ID.
name	string	Resource name.
properties.ipAddressGroups	IpAddressGroup[]	List of ip address groups.
systemData	SystemData	Read only system data
type	string	Resource type.

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

U.S. Patent No. 9,762,692	Accused Microsoft Functionalities
	Connectivity within Microsoft's network along with new Regional Caching capabilities enables more consistent, more predictable cache fill performance by providing multi-tier caching along direct, private access to content in Azure from each CDN edge point of presence (POP). Azure CDN from Microsoft entered public preview providing access to 54 global Edge POPs in 33 countries and 16 Regional Cache POPs.  https://azure.microsoft.com/es-es/blog/new-locations-for-azure-cdn-now-available/

SentToOriginShield (deprecated) * **See notes on deprecation in the following section.**	If true, it means that request was answered from origin shield cache instead of the edge pop. Origin shield is a parent cache used to improve cache hit ratio.
isReceivedFromClient	If true, it means that the request came from the client. If false, the request is a miss in the edge (child POP) and is responded from origin shield (parent POP).

https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log

Sent to origin shield deprecation

The raw log property **isSentToOriginShield** is deprecated and replaced by a new field **isReceivedFromClient**. Use the new field if you're already using the deprecated field.

Raw logs include logs generated from both CDN edge (child POP) and origin shield. Origin shield refers to parent nodes that are strategically located across the globe. These nodes communicate with origin servers and reduce the traffic load on origin.

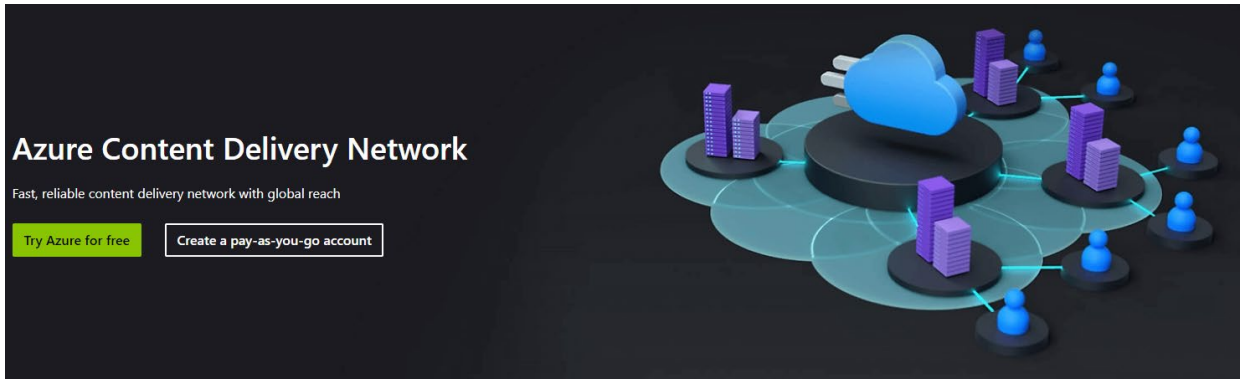
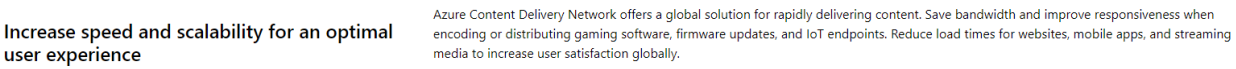
For every request that goes to origin shield, there are two log entries:

- One for edge nodes
- One for origin shield.

https://learn.microsoft.com/en-us/azure/cdn/monitoring-and-access-log

See also Claim 17.

EXHIBIT F

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
1[Pre]. A computer-implemented method, operable with a content delivery network (CDN), the method comprising:	Microsoft performs a computer-implemented method, operable with a content delivery network (CDN) as set forth below. For instance, Microsoft Azure CDN and/or Azure Front Door comprise nodes (e.g., EdgeNodes or Endpoints or Points of Presence) with locally cached content as well as policy-based caching and content versioning in the manner of claim 1.  https://azure.microsoft.com/en-us/products/cdn  https://azure.microsoft.com/en-us/products/cdn  https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Improved performance and user experience

Users expect a fast, reliable, and personalized web experience wherever they are. Content Delivery Network helps reduce latency and improve performance for high-bandwidth content by distributing user requests and serving content directly from edge servers. This brings the content closer to users and sends less traffic to the origin point, delivering superior online experiences.

https://azure.microsoft.com/en-us/products/cdn

Global coverage and massive scalability

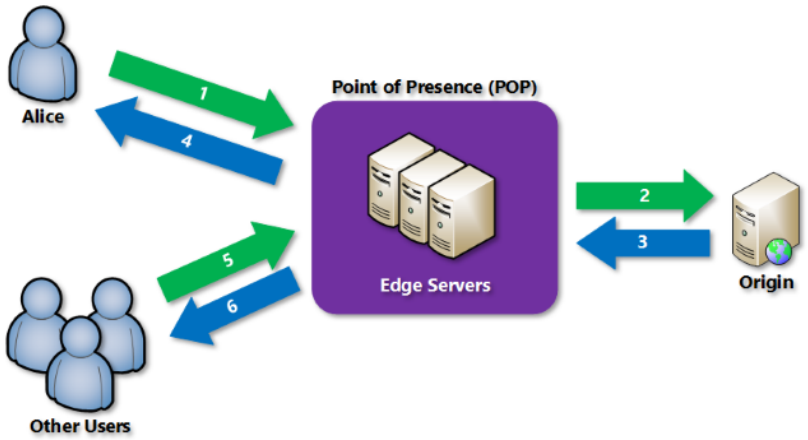
Scale on the fly with no downtime using a platform that responds to traffic fluctuations. A distributed, global presence enables Content Delivery Network to handle sudden traffic spikes and heavy loads—for example, during major product launches or global sporting events—without new infrastructure costs or capacity concerns. Global, distributed points of presence (PoPs) ensure fast delivery of content and protects the origin from excessive load.

https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Protection at the edge Optimize security through features like custom domain HTTPS and DDoS mitigation. Enable Content Delivery Network to put edge servers on the front line serving and protect the origin server from malicious attacks. Set advanced rules to block requests or validate users and use geo-filtering to ensure the content is delivered only to specific regions. https://azure.microsoft.com/en-us/products/cdn **Automate processes and workflows with DevOps** Azure APIs and developer tools give you the performance, reliability, and security needed to build and automate global applications fast. A unified API allows you to deploy Content Delivery Network endpoints from each provider and still manage them the same way. Azure APIs also provide the flexibility to optimize features for availability and performance. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.
2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.
3. The origin server returns the file to an edge server in the POP.
4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.
5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.
6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

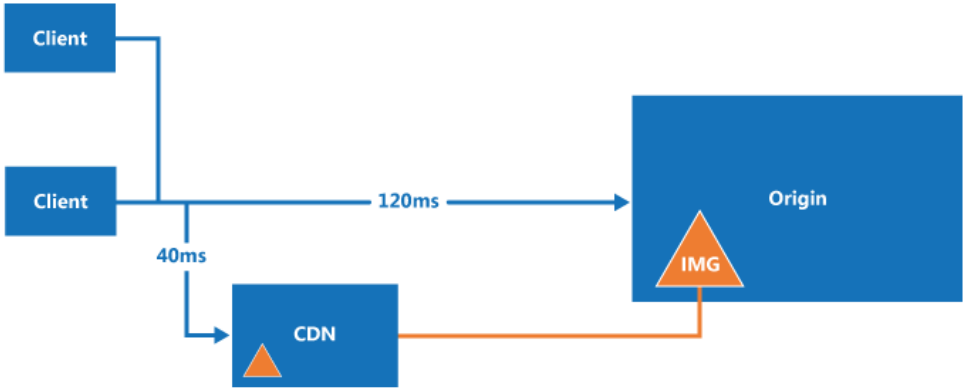
U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	This article lists current metros containing point of presence (POP) locations, sorted by region, for Azure Content Delivery Network products. Each metro might contain more than one POP. For example, Azure Content Delivery Network from Microsoft has 192 POPs across 109 metro cities. **Important** Each Azure Content Delivery Network product has a distinct way of building its content delivery network infrastructures, hence Microsoft recommends against using POP locations to decide which Azure Content Delivery Network product to use. Instead, you should consider its features and end-user performance. Test the performance with each Azure Content Delivery Network product to choose the right product for your users. https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Region	Cities
North America	Etobicoke, Canada (2) Montreal, Canada Vancouver, Canada (2) Querétaro, Mexico (2) Atlanta, GA, USA (3) Boydton, VA, USA (2) Chaska, MN, USA (2) Cheyenne, WY, USA (2) Chicago, IL, USA (4) Dallas, TX, USA (4) Des Moines,IA, USA (3) Detroit, MI, USA Englewood, CO, USA (2) Honolulu, HI, USA Houston, TX, USA (3) Jacksonville, FL, USA (2) Las Vegas, NV, USA (3) Los Angeles, CA, USA (2) Manassas, VA, USA (3) Memphis, TN, USA Miami, FL, USA (4) Minneapolis, MN, USA (2) Needham Heights, MA, USA (2) Nashville, TN, USA Newark, NJ, USA New York, NY, USA (2) Philadelphia, PA, USA Phoenix, AZ, USA Plano, TX, USA Portland, OR, USA Quincy, WA, USA (3) San Antonio, TX, USA (4) San Jose, CA, USA (4) Salt Lake City, UT, USA (3) Seattle, WA, USA (2) Secaucus, NJ, USA (2) Southfield, MI, USA Tempe, AZ, USA

https://learn.microsoft.com/en-us/azure/cdn/cdn-pop-locations

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Edge Nodes - List

Reference

Feedback

Service: CDN

API Version: 2024-02-01

Edgenodes are the global Point of Presence (POP) locations used to deliver CDN content to end users.

HTTP Copy Try It

```
GET https://management.azure.com/providers/Microsoft.Cdn/edgenodes?api-version=2024-02-01
```

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

EdgeNode

Edgenode is a global Point of Presence (POP) location used to deliver CDN content to end users.

Expand table

Name	Type	Description
id	string	Resource ID.
name	string	Resource name.
properties.ipAddressGroups	IpAddressGroup[]	List of ip address groups.
systemData	SystemData	Read only system data
type	string	Resource type.

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Microsoft Azure CDN additionally includes products such as Azure Front Door as well as Standard and Premium CDN services from Edgio (formerly Verizon). Azure Content Delivery Network includes three products: • Azure CDN Standard from Microsoft • Azure CDN Standard from Edgio (formerly Verizon) • Azure CDN Premium from Edgio (formerly Verizon). https://learn.microsoft.com/en-us/azure/cdn/cdn-features

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Performance features and optimizations	Standard Microsoft	Standard Edgio	Premium Edgio
Dynamic site acceleration	Offered via Azure Front Door	✓	✓
General web delivery optimization	✓	✓	✓
Video streaming optimization	via General Web Delivery	via General Web Delivery	via General Web Delivery
Large file optimization	via General Web Delivery	via General Web Delivery	via General Web Delivery
Origin Port	All TCP ports	All TCP ports	All TCP ports
Global server load balancing (GSLB)	✓	✓	✓
Fast purge	✓	✓	✓
Asset pre-loading		✓	✓
Cache/header settings (using caching rules)	✓ using Standard rules engine	✓	
Customizable, rules based content delivery engine	✓ using Standard rules engine		✓ using rules engine
Cache/header settings	✓ using Standard rules engine		✓ using Premium rules engine
URL redirect/rewrite	✓ using Standard rules engine		✓ using Premium rules engine
Mobile device rules	✓ using Standard rules engine		✓ using Premium rules engine
Query string caching	✓	✓	✓
Internet Protocol version 4 (IPv4)/Internet Protocol version 6 (IPv6) dual-stack	✓	✓	✓
HTTP/2 support	✓	✓	✓

https://learn.microsoft.com/en-us/azure/cdn/cdn-features

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Azure Content Delivery Network offers two ways to control how your files get cached: **Caching rules:** Azure Content Delivery Network provides two types of caching rules: global and custom. • Global caching rules - You can set one global caching rule for each endpoint in your profile, which affects all requests to the endpoint. The global caching rule overrides any HTTP cache-directive headers, if set. • Custom caching rules - You can set one or more custom caching rules for each endpoint in your profile. Custom caching rules match specific paths and file extensions, get processed in order, and override the global caching rule, if set. **Query string caching:** You can adjust how the Azure content delivery network treats caching for requests with query strings. For information, see Control Azure Content Delivery Network caching behavior with query strings. If the file isn't cacheable, the query string caching setting has no effect, based on caching rules and content delivery network default behaviors. For information about default caching behavior and caching directive headers, see How caching works. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Microsoft also performs the methods of the asserted claims of the '173 Patent to serve Microsoft resources and services using the same or similar infrastructure. For example, Microsoft uses internal CDN technologies such as its Azure CDN to support at least Xbox gaming, Office 365, Windows updates, and web resources like Linkedin and Bing. • Built on Microsoft's massive-scale private global network, Azure Front Door is a proven platform used to power some of the largest and latency sensitive global services at Microsoft, such as Microsoft Office 365, Bing, LinkedIn, and Xbox. https://azure.microsoft.com/en-us/blog/azure-front-door-enhances-secure-cloud-cdn-with-intelligent-threat-protection/ Azure CDN provides a world class platform to let you reduce load times, save bandwidth, and speed responsiveness across your businesses diverse workflows. Azure CDN from Microsoft enables Azure customers to use and deliver content from the same global CDN network leveraged by Microsoft properties such as Office 365, Skype, Bing, OneDrive, Windows, and Xbox. Connectivity within Microsoft's network along with new Regional Caching capabilities enables more consistent, more predictable cache fill performance by providing multi-tier caching along direct, private access to content in Azure from each CDN edge point of presence (POP). Azure CDN from Microsoft entered public preview providing access to 54 global Edge POPs in 33 countries and 16 Regional Cache POPs. https://azure.microsoft.com/en-us/blog/announcing-microsoft-s-own-cdn-network/

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	What powers Windows 365? Not surprising, Microsoft's own cloud computing platform, Azure. Specifically, **Azure Virtual Desktop**, which can be used to create Windows virtual machines on Redmond's servers, then stream those desktops to PC and non-PC devices. https://www.computerworld.com/article/1614540/what-is-windows-365-microsoft-s-cloud-based-os-explained.html **Use the Office 365 Content Delivery Network (CDN) with SharePoint Online** 02/28/2025 You can use the built-in Office 365 Content Delivery Network (CDN) to host static assets to provide better performance for your SharePoint Online pages. The Office 365 CDN improves performance by caching static assets closer to the browsers requesting them, which helps to speed up downloads and reduce latency. Also, the Office 365 CDN uses the HTTP/2 protocol for improved compression and HTTP pipelining. The Office 365 CDN service is included as part of your SharePoint Online subscription.

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	The Office 365 CDN is composed of multiple CDNs that allow you to host static assets in multiple locations, or *origins*, and serve them from global high-speed networks. Depending on the kind of content you want to host in the Office 365 CDN such as you can add **public** origins. If you're already familiar with the way that CDNs work, you only need to complete a few steps to enable the Office 365 CDN for your tenant. This article describes how. Read on for information about how to get started hosting your static assets. https://learn.microsoft.com/en-us/microsoft-365/enterprise/use-microsoft-365-cdn-with-spo?view=o365-worldwide On the Microsoft side, the company uses primarily Azure Front Door a cloud CDN platform and some third-party CDNs for some static delivery of content i.e Xbox downloads, Windows Updates. Microsoft does not https://www.streamingmediablog.com/2020/10/gaming-console-launch.html

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Teams is built on a microservices architecture, with a few hundred microservices working cohesively to deliver our product's many features including messaging, meetings, files, calendar, and apps. Using microservices helps each of our component teams to work and release their changes independently.

Azure is the cloud platform that underpins all of Microsoft's cloud services, including Microsoft Teams. Our workloads run in Azure virtual machines (VMs), with our older services being deployed through Azure Cloud Services and our newer ones on Azure Service Fabric. Our primary storage stack is Azure Cosmos DB, with some services using Azure Blob Storage. We count on Azure Cache for Redis for increased throughput and resiliency. We leverage Traffic Manager and Azure Front Door to route traffic where we want it to be. We use Queue Storage and Event Hubs to communicate, and we depend on Azure Active Directory to manage our tenants and users.

- **Resiliency-optimized caches**: We leverage caches between our components extensively, for both performance and resiliency. Caches help reduce average latency and provide a source of data in case a downstream service is unavailable. Keeping data in caches for a long time introduces data freshness issues yet keeping data in caches for a long time is the best defense against downstream failures. We focus on Time to Refresh (TTR) to our cache data as well as Time to Live (TTL). By setting a long TTL and a shorter TTR value, we can fine-tune how fresh to keep our data versus how long we want data to stick around whenever a downstream dependency fails.

https://azure.microsoft.com/en-us/blog/advancing-microsoft-teams-on-azure-operating-at-pandemic-scale/

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Background

A CDN is a system of distributed servers that delivers web content to users from geographical locations closest to them. At LinkedIn, CDNs deliver static content, such as CSS and JavaScript on the site, as well as member-generated content, such as profile pictures, media shares, and video uploads, to our members around the world. The content is delivered from edges that are closest to the user in question.

The Edge SRE team runs LinkedIn's edge infrastructure, including four external CDNs, an in-house CDN, three Domain Name System (DNS) platforms, and all LinkedIn points of presence (PoPs). We build and manage tools to automate all aspects of our Edge stack.

https://www.linkedin.com/blog/engineering/data-management/enabling-dual-stack-on-linkedin-cdns\

LinkedIn recently described how it uses the edge to support the popular professional networking web site. The engineering team wrote a blog post to talk about edge deployments and its migration away from an in-house solution to Azure Edge.

LinkedIn is one of many large web properties that is owned by a hyperscaler cloud provider — Microsoft, in this case — but doesn't actually use the in-house cloud. YouTube is another example of this — it has yet to be migrated to GCP.

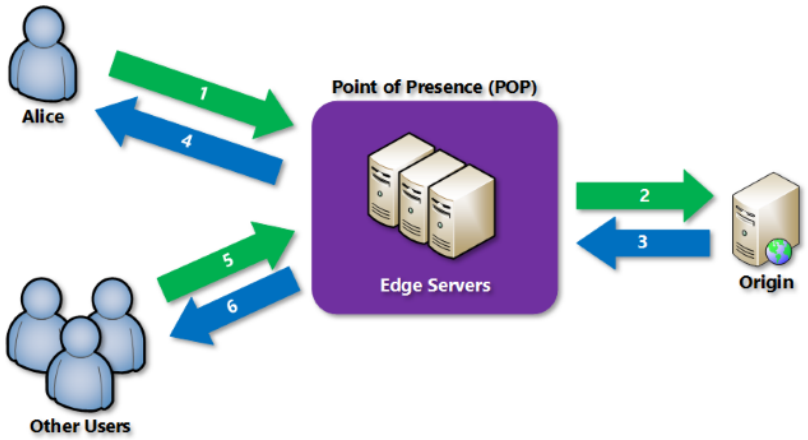
LinkedIn actually had its own quite extensive in-house CDN, with nineteen PoPs (points of presence) in ten countries. This is a common practice for major content owners as it defrays transport costs and improves performance. The nodes can cache content and perform tasks such as SSL offloading. But as LinkedIn noted, Microsoft's CDN – Azure Front Door – gives access to 300 PoPs around the globe, which is a big step up in terms of reach.

https://www.edgeir.com/how-linkedin-is-using-azure-front-door-for-edge-build-out-20220306

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
[1.a] (A) in response to a request for content, determining, by a node in the CDN, if the content is cached locally at the node;	Microsoft determines, by a node in the CDN, if content is cached locally at the node in response to a request for content. For instance, Microsoft Azure CDN and/or Azure Front Door nodes (e.g., EdgeNodes or Endpoints or Points of Presence) determine if content is cached locally at the CDN node in response to a request for content. **Improved performance and user experience** Users expect a fast, reliable, and personalized web experience wherever they are. Content Delivery Network helps reduce latency and improve performance for high-bandwidth content by distributing user requests and serving content directly from edge servers. This brings the content closer to users and sends less traffic to the origin point, delivering superior online experiences. https://azure.microsoft.com/en-us/products/cdn **Global coverage and massive scalability** Scale on the fly with no downtime using a platform that responds to traffic fluctuations. A distributed, global presence enables Content Delivery Network to handle sudden traffic spikes and heavy loads—for example, during major product launches or global sporting events—without new infrastructure costs or capacity concerns. Global, distributed points of presence (PoPs) ensure fast delivery of content and protects the origin from excessive load. https://azure.microsoft.com/en-us/products/cdn

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

How it works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.
2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.
3. The origin server returns the file to an edge server in the POP.
4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.
5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP.
6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

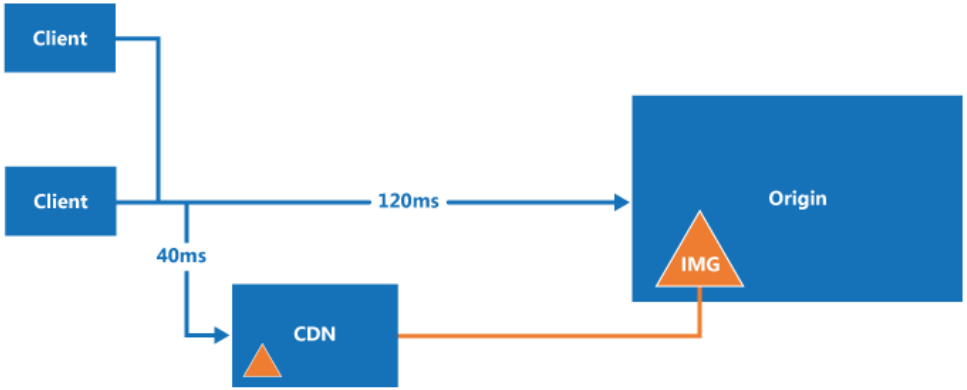
U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering For a complete list of features that each Azure Content Delivery Network product supports, see Compare Azure Content Delivery Network product features. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

CDN guidance

Azure Storage | Azure Blob Storage

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. CDNs store cached content on edge servers that are close to end users to minimize latency.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The major advantages of using a CDN are lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted. CDNs can also help to reduce load on a web application, because the application does not have to service requests for the content that is hosted in the CDN.



In Azure, the Azure Content Delivery Network is a global CDN solution for delivering high-bandwidth content that is hosted in Azure or any other location. Using Azure Content Delivery Network, you can cache publicly available objects loaded from Azure blob storage, a web application, virtual machine, any publicly accessible web server.

https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Edge Nodes - List

Reference

Feedback

Service: CDN

API Version: 2024-02-01

Edgenodes are the global Point of Presence (POP) locations used to deliver CDN content to end users.

HTTP Copy Try It

```
GET https://management.azure.com/providers/Microsoft.Cdn/edgenodes?api-version=2024-02-01
```

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

EdgeNode

Edgenode is a global Point of Presence (POP) location used to deliver CDN content to end users.

Expand table

Name	Type	Description
id	string	Resource ID.
name	string	Resource name.
properties.ipAddressGroups	IpAddressGroup[]	List of ip address groups.
systemData	SystemData	Read only system data
type	string	Resource type.

https://learn.microsoft.com/en-us/rest/api/cdn/edge-nodes/list?view=rest-cdn-2024-02-01&tabs=HTTP

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Introduction to caching Caching is the process of storing data locally so that future requests for that data can be accessed more quickly. In the most common type of caching, web browser caching, a web browser stores copies of static data locally on a local hard drive. By using caching, the web browser can avoid making multiple round-trips to the server and instead access the same data locally, thus saving time and resources. Caching is well-suited for locally managing small, static data such as static images, CSS files, and JavaScript files. Similarly, caching is used by a content delivery network on edge servers close to the user to avoid requests traveling back to the origin and reducing end-user latency. Unlike a web browser cache, which is used only for a single user, the content delivery network has a shared cache. In a content delivery network shared cache, a file request by a user can be used by another user, which greatly decreases the number of requests to the origin server. Dynamic resources that change frequently or are unique to an individual user can't be cached. Those types of resources, however, can take advantage of dynamic site acceleration (DSA) optimization on the Azure content delivery network for performance improvements. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Caching can occur at multiple levels between the origin server and the end user: • Web server: Uses a shared cache (for multiple users). • Content delivery network: Uses a shared cache (for multiple users). • Internet service provider (ISP): Uses a shared cache (for multiple users). • Web browser: Uses a private cache (for one user). Each cache typically manages its own resource freshness and performs validation when a file is stale. This behavior is defined in the HTTP caching specification, RFC 7234. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Resource freshness

Since a cached resource can potentially be out-of-date, or stale (as compared to the corresponding resource on the origin server), it's important for any caching mechanism to control when content gets a refresh. To save time and bandwidth consumption, a cached resource isn't compared to the version on the origin server every time it's accessed. Instead, as long as a cached resource is considered to be fresh, it's assumed to be the most current version and is sent directly to the client. A cached resource is considered to be fresh when its age is less than the age or period defined by a cache setting. For example, when a browser reloads a web page, it verifies that each cached resource on your hard drive is fresh and loads it. If the resource isn't fresh (stale), an up-to-date copy is loaded from the server.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

Validation

If a resource is considered stale, the origin server gets asked to validate it to determine whether the data in the cache still matches what's on the origin server. If the file has been modified on the origin server, the cache updates its version of the resource. Otherwise, if the resource is fresh, the data is delivered directly from the cache without validating it first.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Content delivery network caching Caching is integral to the way a content delivery network operates to speed up delivery and reduce origin load for static assets such as images, fonts, and videos. In content delivery network caching, static resources are selectively stored on strategically placed servers that are more local to a user and offers the following advantages: • Because most web traffic is static (for example, images, fonts, and videos), content delivery network caching reduces network latency by moving content closer to the user, thus reducing the distance that data travels. • By offloading work to a content delivery network, caching can reduce network traffic and the load on the origin server. Doing so reduces cost and resource requirements for the application, even when there are large numbers of users. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Similar to how caching is implemented in a web browser, you can control how caching is performed in a content delivery network by sending cache-directive headers. Cache-directive headers are HTTP headers, which are typically added by the origin server. Although most of these headers were originally designed to address caching in client browsers, they're now also used by all intermediate caches, such as content delivery networks. Two headers can be used to define cache freshness: `Cache-Control` and `Expires`. `Cache-Control` is more current and takes precedence over `Expires`, if both exist. There are also two types of headers used for validation (called validators): `ETag` and `Last-Modified`. `ETag` is more current and takes precedence over `Last-Modified`, if both are defined. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Azure Content Delivery Network supports the following HTTP cache-directive headers, which define cache duration and cache sharing. **Cache-Control:** • Introduced in HTTP 1.1 to give web publishers more control over their content and to address the limitations of the `Expires` header. • Overrides the `Expires` header, if both it and `Cache-Control` are defined. • When used in an HTTP request from the client to the content delivery network POP, `Cache-Control` gets ignored by all Azure Content Delivery Network profiles, by default. • When used in an HTTP response from the origin server to the content delivery network POP: ○ **Azure CDN Standard/Premium from Edgio** and **Azure CDN Standard from Microsoft** support all `Cache-Control` directives. ○ **Azure CDN Standard/Premium from Edgio** and **Azure CDN Standard from Microsoft** honors caching behaviors for Cache-Control directives in RFC 7234 - Hypertext Transfer Protocol (HTTP/1.1): Caching (ietf.org). **Expires:** • Legacy header introduced in HTTP 1.0; supported for backward compatibility. • Uses a date-based expiration time with second precision. • Similar to `Cache-Control: max-age`. • Used when `Cache-Control` doesn't exist. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

Validators

When the cache is stale, HTTP cache validators are used to compare the cached version of a file with the version on the origin server. **Azure CDN Standard/Premium from Edgio** supports both `ETag` and `Last-Modified` validators by default, while **Azure CDN Standard from Microsoft** supports only `Last-Modified`.

ETag:

- **Azure CDN Standard/Premium from Edgio** supports `ETag` by default, while **Azure CDN Standard from Microsoft** doesn't.
- `ETag` defines a string that is unique for every file and version of a file. For example, `ETag: "17f0ddd99ed5bbe4edffdd6496d7131f"`.
- Introduced in HTTP 1.1 and is more current than `Last-Modified`. Useful when the last modified date is difficult to determine.
- Supports both strong validation and weak validation; however, Azure Content Delivery Network supports only strong validation. For strong validation, the two resource representations must be byte-for-byte identical.
- A cache validates a file that uses `ETag` by sending an `If-None-Match` header with one or more `ETag` validators in the request. For example, `If-None-Match: "17f0ddd99ed5bbe4edffdd6496d7131f"`. If the server's version matches an `ETag` validator on the list, it sends status code 304 (Not Modified) in its response. If the version is different, the server responds with status code 200 (OK) and the updated resource.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

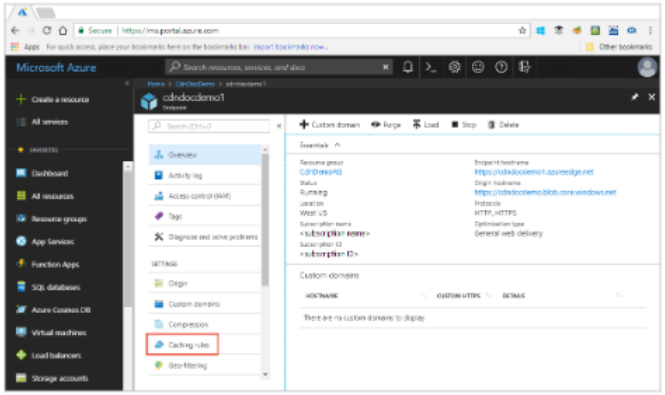
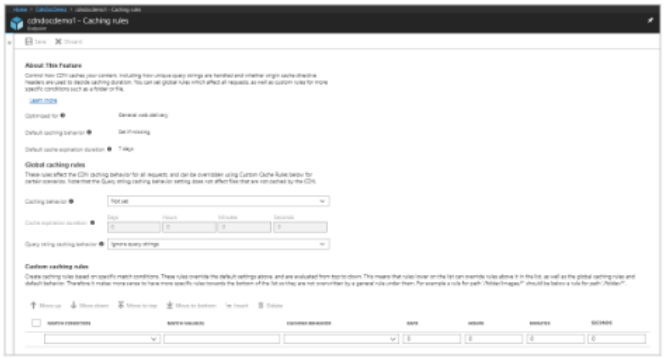
Last-Modified:

- For **Azure CDN Standard/Premium from Edgio** only, `Last-Modified` is used if `ETag` isn't part of the HTTP response.
- Specifies the date and time that the origin server has determined the resource was last modified. For example, `Last-Modified: Thu, 19 Oct 2017 09:28:00 GMT`.
- For content larger than 8 MB, origin backend servers should maintain consistent `Last-Modified` timestamps per asset. Returning inconsistent `Last-Modified` times from backend servers will cause validator mismatch errors and result in HTTP 5XX failures. Azure Storage may not support consistent `Last-Modified` timestamps across replicas, which can cause similar validator mismatch errors.
- A cache validates a file using `Last-Modified` by sending an `If-Modified-Since` header with a date and time in the request. The origin server compares that date with the `Last-Modified` header of the latest resource. If the resource hasn't been modified since the specified time, the server returns status code 304 (Not Modified) in its response. If the resource has been modified, the server returns status code 200 (OK) and the updated resource.

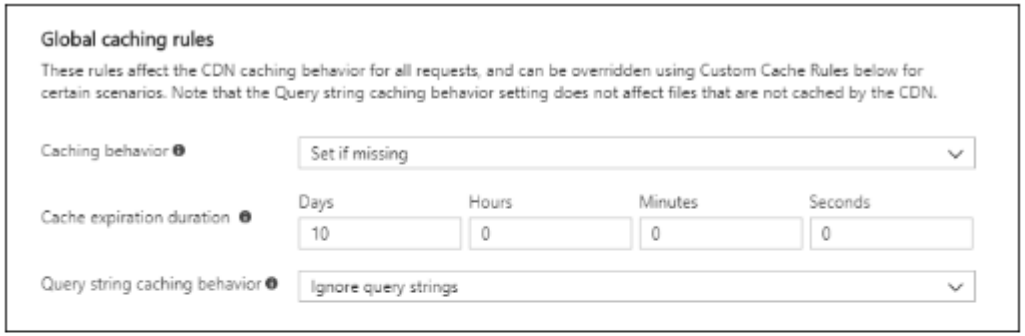
https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
[1.b] (B) when it is determined by the node that said content is cached locally, then, after determining that the content is cached locally: **[1.b.1]** (B)(1) determining, by the node, a current cache policy associated with the content, wherein determining the current cache policy comprises determining a time period for expiration of content in the cache; and then	When it is determined by the node that the content is cached locally, Microsoft then determines, by the node, a current cache policy associated with the content, wherein determining the current cache policy comprises determining a time period for expiration of content in the cache. For instance, Microsoft Azure CDN offers content caching rules which determine time periods for expiration of content in the cache. A Microsoft Azure CDN node (e.g., EdgeNode or Endpoint or Point of Presence) determines the current cache policy associated with requested content (e.g. a cache eviction policy, TTL, TTR, etc.) when it is determined the node has a local copy of the requested content.

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Azure Content Delivery Network offers two ways to control how your files get cached: **Caching rules:** Azure Content Delivery Network provides two types of caching rules: global and custom. • Global caching rules - You can set one global caching rule for each endpoint in your profile, which affects all requests to the endpoint. The global caching rule overrides any HTTP cache-directive headers, if set. • Custom caching rules - You can set one or more custom caching rules for each endpoint in your profile. Custom caching rules match specific paths and file extensions, get processed in order, and override the global caching rule, if set. **Query string caching:** You can adjust how the Azure content delivery network treats caching for requests with query strings. For information, see Control Azure Content Delivery Network caching behavior with query strings. If the file isn't cacheable, the query string caching setting has no effect, based on caching rules and content delivery network default behaviors. For information about default caching behavior and caching directive headers, see How caching works. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Accessing Azure Content Delivery Network caching rules 1. Open the Azure portal, select a content delivery network profile, then select an endpoint. 2. In the left pane under Settings, select **Caching rules**.  The **Caching rules** page appears.  https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Caching behavior settings For global and custom caching rules, you can specify the following **Caching behavior** settings: • **Bypass cache:** Don't cache and ignore origin-provided cache-directive headers. • **Override:** Ignore origin-provided cache duration; use the provided cache duration instead. This setting doesn't override cache-control: no-cache. ⓘ Note For **Azure CDN from Microsoft** profiles, cache expiration override is only applicable to status codes 200 and 206. • **Set if missing:** Honor origin-provided cache-directive headers, if they exist; otherwise, use the provided cache duration. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	  https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Cache expiration duration For global and custom caching rules, you can specify the cache expiration duration in days, hours, minutes, and seconds: • For the **Override** and **Set if missing Caching behavior** settings, valid cache durations range between 0 seconds and 366 days. For a value of 0 seconds, the content delivery network caches the content, but must revalidate each request with the origin server. • For the **Bypass cache** setting, the cache duration gets automatically set to 0 seconds, which isn't a modifiable value. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Custom caching rules match conditions For custom cache rules, two match conditions are available: • **Path:** This condition matches the path of the URL, excluding the domain name, and supports the wildcard symbol (*). For example, */myfile.html*, */my/folder/***, *and /my/images/.jpg*. The maximum length is 260 characters. • **Extension:** This condition matches the file extension of the requested file. You can provide a list of comma-separated file extensions to match. For example, *.jpg*, *.mp3*, or *.png*. The maximum number of extensions is 50 and the maximum number of characters per extension is 16. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Global and custom rule processing order

Global and custom caching rules get processed in the following order:

- Global caching rules take precedence over the default content delivery network caching behavior (HTTP cache-directive header settings).
- Custom caching rules take precedence over global caching rules, where they apply. Custom caching rules get processed in order from top to bottom. That is, if a request matches both conditions, rules at the bottom of the list take precedence over rules at the top of the list. Therefore, you should place more specific rules lower in the list.

Example:

- Global caching rule:
  - Caching behavior: **Override**
  - Cache expiration duration: One day
- Custom caching rule #1:
  - Match condition: **Path**
  - Match value: */home/**
  - Caching behavior: **Override**
  - Cache expiration duration: Two days
- Custom caching rule #2:
  - Match condition: **Extension**
  - Match value: *.html*
  - Caching behavior: **Set if missing**
  - Cache expiration duration: Three days

https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	When you set these rules, a request for <*endpoint hostname*>.azureedge.net/home/index.html triggers custom caching rule #2, which get set to: **Set if missing** and 3 days. Therefore, if the *index.html* file has `Cache-Control` or `Expires` HTTP headers, they get honored; otherwise, if you don't set these headers, the file gets cached for three days. ⓘ Note Files that are cached before a rule change maintain their origin cache duration setting. To reset their cache durations, you must purge the file. Azure Content Delivery Network configuration changes can take some time to propagate through the network: • For **Azure CDN Standard from Edgio** profiles, propagation usually completes in 10 minutes. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules
[1.b.2] (B)(2) determining, by the node and based on said current cache policy associated with the content, whether it is acceptable to	Microsoft determines, by the node and based on said current cache policy associated with the content, whether it is acceptable to serve the content that is cached locally. For instance, based on the associated caching rule, Microsoft Azure CDN and/or Azure Front Door nodes determine whether they can serve the requested content.

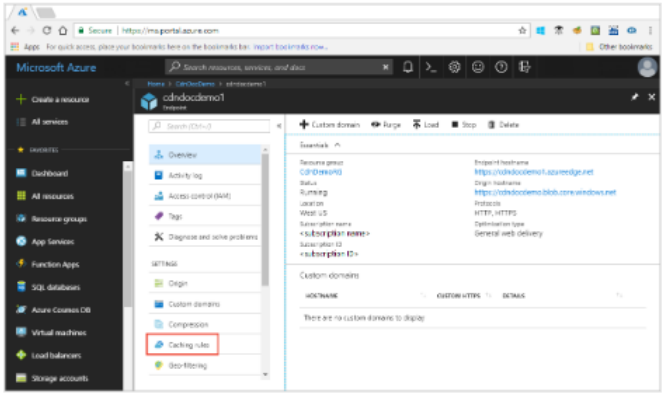
U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
serve the content that is cached locally;	Azure Content Delivery Network offers two ways to control how your files get cached: **Caching rules:** Azure Content Delivery Network provides two types of caching rules: global and custom. • Global caching rules - You can set one global caching rule for each endpoint in your profile, which affects all requests to the endpoint. The global caching rule overrides any HTTP cache-directive headers, if set. • Custom caching rules - You can set one or more custom caching rules for each endpoint in your profile. Custom caching rules match specific paths and file extensions, get processed in order, and override the global caching rule, if set. **Query string caching:** You can adjust how the Azure content delivery network treats caching for requests with query strings. For information, see Control Azure Content Delivery Network caching behavior with query strings. If the file isn't cacheable, the query string caching setting has no effect, based on caching rules and content delivery network default behaviors. For information about default caching behavior and caching directive headers, see How caching works. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

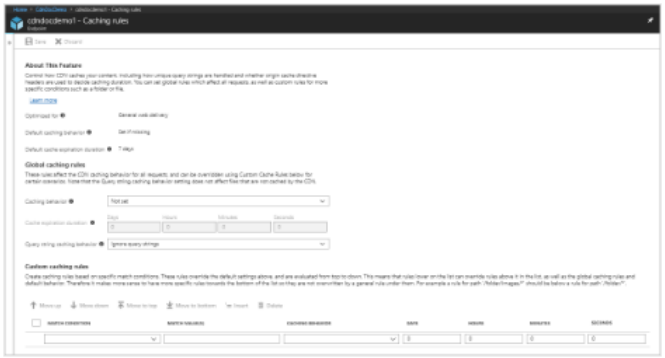
Accessing Azure Content Delivery Network caching rules

1. Open the Azure portal, select a content delivery network profile, then select an endpoint.

2. In the left pane under Settings, select **Caching rules**.



The **Caching rules** page appears.



https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Caching behavior settings

For global and custom caching rules, you can specify the following **Caching behavior** settings:

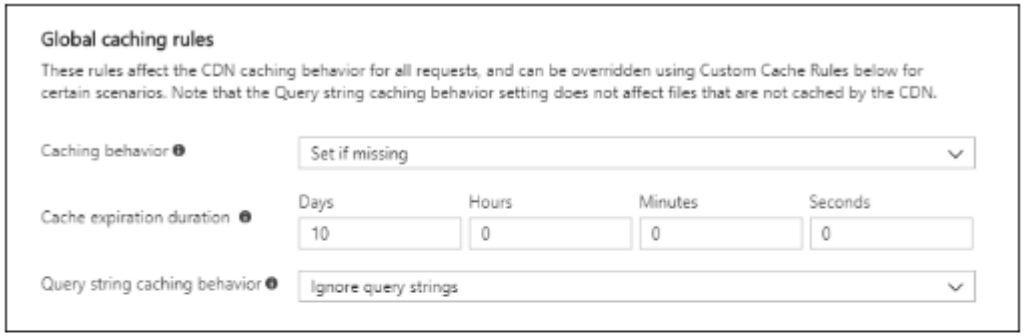
- **Bypass cache:** Don't cache and ignore origin-provided cache-directive headers.
- **Override:** Ignore origin-provided cache duration; use the provided cache duration instead. This setting doesn't override cache-control: no-cache.

> ⓘ Note
>
> For **Azure CDN from Microsoft** profiles, cache expiration override is only applicable to status codes 200 and 206.

- **Set if missing:** Honor origin-provided cache-directive headers, if they exist; otherwise, use the provided cache duration.

https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities





https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Cache expiration duration For global and custom caching rules, you can specify the cache expiration duration in days, hours, minutes, and seconds: • For the **Override** and **Set if missing Caching behavior** settings, valid cache durations range between 0 seconds and 366 days. For a value of 0 seconds, the content delivery network caches the content, but must revalidate each request with the origin server. • For the **Bypass cache** setting, the cache duration gets automatically set to 0 seconds, which isn't a modifiable value. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Custom caching rules match conditions For custom cache rules, two match conditions are available: • **Path:** This condition matches the path of the URL, excluding the domain name, and supports the wildcard symbol (*). For example, */myfile.html*, */my/folder/***, *and /my/images/.jpg*. The maximum length is 260 characters. • **Extension:** This condition matches the file extension of the requested file. You can provide a list of comma-separated file extensions to match. For example, *.jpg*, *.mp3*, or *.png*. The maximum number of extensions is 50 and the maximum number of characters per extension is 16. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Global and custom rule processing order

Global and custom caching rules get processed in the following order:

- Global caching rules take precedence over the default content delivery network caching behavior (HTTP cache-directive header settings).
- Custom caching rules take precedence over global caching rules, where they apply. Custom caching rules get processed in order from top to bottom. That is, if a request matches both conditions, rules at the bottom of the list take precedence over rules at the top of the list. Therefore, you should place more specific rules lower in the list.

Example:

- Global caching rule:
  - Caching behavior: **Override**
  - Cache expiration duration: One day
- Custom caching rule #1:
  - Match condition: **Path**
  - Match value: */home/**
  - Caching behavior: **Override**
  - Cache expiration duration: Two days
- Custom caching rule #2:
  - Match condition: **Extension**
  - Match value: *.html*
  - Caching behavior: **Set if missing**
  - Cache expiration duration: Three days

https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	When you set these rules, a request for <*endpoint hostname*>.azureedge.net/home/index.html triggers custom caching rule #2, which get set to: **Set if missing** and 3 days. Therefore, if the *index.html* file has `Cache-Control` or `Expires` HTTP headers, they get honored; otherwise, if you don't set these headers, the file gets cached for three days. ⓘ Note Files that are cached before a rule change maintain their origin cache duration setting. To reset their cache durations, you must purge the file. Azure Content Delivery Network configuration changes can take some time to propagate through the network: • For **Azure CDN Standard from Edgio** profiles, propagation usually completes in 10 minutes. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules
[1.b.3] (B)(3) based on said determining in (B)(2), (i) when it is not acceptable to serve the content that is	Microsoft, when it is not acceptable to serve the content that is cached locally, obtains, by the node, a new version of the content and then serving, by the node, the new version of the content. For instance, based on the cache policy and file validation, Microsoft Azure CDN and/or Azure Front Door nodes may obtain a new version of content to be served, such as when the cached content has expired or has become stale and needs to be refreshed.

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
cached locally, obtaining, by the node, a new version of the content and then serving, by the node, the new version of the content, and	# Introduction to caching Caching is the process of storing data locally so that future requests for that data can be accessed more quickly. In the most common type of caching, web browser caching, a web browser stores copies of static data locally on a local hard drive. By using caching, the web browser can avoid making multiple round-trips to the server and instead access the same data locally, thus saving time and resources. Caching is well-suited for locally managing small, static data such as static images, CSS files, and JavaScript files. Similarly, caching is used by a content delivery network on edge servers close to the user to avoid requests traveling back to the origin and reducing end-user latency. Unlike a web browser cache, which is used only for a single user, the content delivery network has a shared cache. In a content delivery network shared cache, a file request by a user can be used by another user, which greatly decreases the number of requests to the origin server. Dynamic resources that change frequently or are unique to an individual user can't be cached. Those types of resources, however, can take advantage of dynamic site acceleration (DSA) optimization on the Azure content delivery network for performance improvements. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Caching can occur at multiple levels between the origin server and the end user: • Web server: Uses a shared cache (for multiple users). • Content delivery network: Uses a shared cache (for multiple users). • Internet service provider (ISP): Uses a shared cache (for multiple users). • Web browser: Uses a private cache (for one user). Each cache typically manages its own resource freshness and performs validation when a file is stale. This behavior is defined in the HTTP caching specification, RFC 7234. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Resource freshness

Since a cached resource can potentially be out-of-date, or stale (as compared to the corresponding resource on the origin server), it's important for any caching mechanism to control when content gets a refresh. To save time and bandwidth consumption, a cached resource isn't compared to the version on the origin server every time it's accessed. Instead, as long as a cached resource is considered to be fresh, it's assumed to be the most current version and is sent directly to the client. A cached resource is considered to be fresh when its age is less than the age or period defined by a cache setting. For example, when a browser reloads a web page, it verifies that each cached resource on your hard drive is fresh and loads it. If the resource isn't fresh (stale), an up-to-date copy is loaded from the server.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

Validation

If a resource is considered stale, the origin server gets asked to validate it to determine whether the data in the cache still matches what's on the origin server. If the file has been modified on the origin server, the cache updates its version of the resource. Otherwise, if the resource is fresh, the data is delivered directly from the cache without validating it first.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Content delivery network caching Caching is integral to the way a content delivery network operates to speed up delivery and reduce origin load for static assets such as images, fonts, and videos. In content delivery network caching, static resources are selectively stored on strategically placed servers that are more local to a user and offers the following advantages: • Because most web traffic is static (for example, images, fonts, and videos), content delivery network caching reduces network latency by moving content closer to the user, thus reducing the distance that data travels. • By offloading work to a content delivery network, caching can reduce network traffic and the load on the origin server. Doing so reduces cost and resource requirements for the application, even when there are large numbers of users. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Similar to how caching is implemented in a web browser, you can control how caching is performed in a content delivery network by sending cache-directive headers. Cache-directive headers are HTTP headers, which are typically added by the origin server. Although most of these headers were originally designed to address caching in client browsers, they're now also used by all intermediate caches, such as content delivery networks. Two headers can be used to define cache freshness: `Cache-Control` and `Expires`. `Cache-Control` is more current and takes precedence over `Expires`, if both exist. There are also two types of headers used for validation (called validators): `ETag` and `Last-Modified`. `ETag` is more current and takes precedence over `Last-Modified`, if both are defined. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Azure Content Delivery Network supports the following HTTP cache-directive headers, which define cache duration and cache sharing. **Cache-Control:** • Introduced in HTTP 1.1 to give web publishers more control over their content and to address the limitations of the `Expires` header. • Overrides the `Expires` header, if both it and `Cache-Control` are defined. • When used in an HTTP request from the client to the content delivery network POP, `Cache-Control` gets ignored by all Azure Content Delivery Network profiles, by default. • When used in an HTTP response from the origin server to the content delivery network POP: ○ **Azure CDN Standard/Premium from Edgio** and **Azure CDN Standard from Microsoft** support all `Cache-Control` directives. ○ **Azure CDN Standard/Premium from Edgio** and **Azure CDN Standard from Microsoft** honors caching behaviors for Cache-Control directives in RFC 7234 - Hypertext Transfer Protocol (HTTP/1.1): Caching (ietf.org). **Expires:** • Legacy header introduced in HTTP 1.0; supported for backward compatibility. • Uses a date-based expiration time with second precision. • Similar to `Cache-Control: max-age`. • Used when `Cache-Control` doesn't exist. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

Validators

When the cache is stale, HTTP cache validators are used to compare the cached version of a file with the version on the origin server. **Azure CDN Standard/Premium from Edgio** supports both `ETag` and `Last-Modified` validators by default, while **Azure CDN Standard from Microsoft** supports only `Last-Modified`.

ETag:

- **Azure CDN Standard/Premium from Edgio** supports `ETag` by default, while **Azure CDN Standard from Microsoft** doesn't.
- `ETag` defines a string that is unique for every file and version of a file. For example, `ETag: "17f0ddd99ed5bbe4edffdd6496d7131f"`.
- Introduced in HTTP 1.1 and is more current than `Last-Modified`. Useful when the last modified date is difficult to determine.
- Supports both strong validation and weak validation; however, Azure Content Delivery Network supports only strong validation. For strong validation, the two resource representations must be byte-for-byte identical.
- A cache validates a file that uses `ETag` by sending an `If-None-Match` header with one or more `ETag` validators in the request. For example, `If-None-Match: "17f0ddd99ed5bbe4edffdd6496d7131f"`. If the server's version matches an `ETag` validator on the list, it sends status code 304 (Not Modified) in its response. If the version is different, the server responds with status code 200 (OK) and the updated resource.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Last-Modified:

- For **Azure CDN Standard/Premium from Edgio** only, `Last-Modified` is used if `ETag` isn't part of the HTTP response.
- Specifies the date and time that the origin server has determined the resource was last modified. For example, `Last-Modified: Thu, 19 Oct 2017 09:28:00 GMT`.
- For content larger than 8 MB, origin backend servers should maintain consistent `Last-Modified` timestamps per asset. Returning inconsistent `Last-Modified` times from backend servers will cause validator mismatch errors and result in HTTP 5XX failures. Azure Storage may not support consistent `Last-Modified` timestamps across replicas, which can cause similar validator mismatch errors.
- A cache validates a file using `Last-Modified` by sending an `If-Modified-Since` header with a date and time in the request. The origin server compares that date with the `Last-Modified` header of the latest resource. If the resource hasn't been modified since the specified time, the server returns status code 304 (Not Modified) in its response. If the resource has been modified, the server returns status code 200 (OK) and the updated resource.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
[1.b.4] (ii) when it is acceptable to serve the content that is cached locally, serving, by the node, the content that is cached locally.	Microsoft, when it is acceptable to serve the content that is cached locally, serves, by the node, the content that is cached locally. For instance, Microsoft Azure CDN and/or Azure Front Door nodes may serve content cached locally based on the cache policy and file validation.

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Introduction to caching Caching is the process of storing data locally so that future requests for that data can be accessed more quickly. In the most common type of caching, web browser caching, a web browser stores copies of static data locally on a local hard drive. By using caching, the web browser can avoid making multiple round-trips to the server and instead access the same data locally, thus saving time and resources. Caching is well-suited for locally managing small, static data such as static images, CSS files, and JavaScript files. Similarly, caching is used by a content delivery network on edge servers close to the user to avoid requests traveling back to the origin and reducing end-user latency. Unlike a web browser cache, which is used only for a single user, the content delivery network has a shared cache. In a content delivery network shared cache, a file request by a user can be used by another user, which greatly decreases the number of requests to the origin server. Dynamic resources that change frequently or are unique to an individual user can't be cached. Those types of resources, however, can take advantage of dynamic site acceleration (DSA) optimization on the Azure content delivery network for performance improvements. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Caching can occur at multiple levels between the origin server and the end user: • Web server: Uses a shared cache (for multiple users). • Content delivery network: Uses a shared cache (for multiple users). • Internet service provider (ISP): Uses a shared cache (for multiple users). • Web browser: Uses a private cache (for one user). Each cache typically manages its own resource freshness and performs validation when a file is stale. This behavior is defined in the HTTP caching specification, RFC 7234. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Resource freshness

Since a cached resource can potentially be out-of-date, or stale (as compared to the corresponding resource on the origin server), it's important for any caching mechanism to control when content gets a refresh. To save time and bandwidth consumption, a cached resource isn't compared to the version on the origin server every time it's accessed. Instead, as long as a cached resource is considered to be fresh, it's assumed to be the most current version and is sent directly to the client. A cached resource is considered to be fresh when its age is less than the age or period defined by a cache setting. For example, when a browser reloads a web page, it verifies that each cached resource on your hard drive is fresh and loads it. If the resource isn't fresh (stale), an up-to-date copy is loaded from the server.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

Validation

If a resource is considered stale, the origin server gets asked to validate it to determine whether the data in the cache still matches what's on the origin server. If the file has been modified on the origin server, the cache updates its version of the resource. Otherwise, if the resource is fresh, the data is delivered directly from the cache without validating it first.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Content delivery network caching Caching is integral to the way a content delivery network operates to speed up delivery and reduce origin load for static assets such as images, fonts, and videos. In content delivery network caching, static resources are selectively stored on strategically placed servers that are more local to a user and offers the following advantages: • Because most web traffic is static (for example, images, fonts, and videos), content delivery network caching reduces network latency by moving content closer to the user, thus reducing the distance that data travels. • By offloading work to a content delivery network, caching can reduce network traffic and the load on the origin server. Doing so reduces cost and resource requirements for the application, even when there are large numbers of users. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Similar to how caching is implemented in a web browser, you can control how caching is performed in a content delivery network by sending cache-directive headers. Cache-directive headers are HTTP headers, which are typically added by the origin server. Although most of these headers were originally designed to address caching in client browsers, they're now also used by all intermediate caches, such as content delivery networks. Two headers can be used to define cache freshness: `Cache-Control` and `Expires`. `Cache-Control` is more current and takes precedence over `Expires`, if both exist. There are also two types of headers used for validation (called validators): `ETag` and `Last-Modified`. `ETag` is more current and takes precedence over `Last-Modified`, if both are defined. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Azure Content Delivery Network supports the following HTTP cache-directive headers, which define cache duration and cache sharing. **Cache-Control:** • Introduced in HTTP 1.1 to give web publishers more control over their content and to address the limitations of the `Expires` header. • Overrides the `Expires` header, if both it and `Cache-Control` are defined. • When used in an HTTP request from the client to the content delivery network POP, `Cache-Control` gets ignored by all Azure Content Delivery Network profiles, by default. • When used in an HTTP response from the origin server to the content delivery network POP: ○ **Azure CDN Standard/Premium from Edgio** and **Azure CDN Standard from Microsoft** support all `Cache-Control` directives. ○ **Azure CDN Standard/Premium from Edgio** and **Azure CDN Standard from Microsoft** honors caching behaviors for Cache-Control directives in RFC 7234 - Hypertext Transfer Protocol (HTTP/1.1): Caching (ietf.org). **Expires:** • Legacy header introduced in HTTP 1.0; supported for backward compatibility. • Uses a date-based expiration time with second precision. • Similar to `Cache-Control: max-age`. • Used when `Cache-Control` doesn't exist. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

Validators

When the cache is stale, HTTP cache validators are used to compare the cached version of a file with the version on the origin server. **Azure CDN Standard/Premium from Edgio** supports both `ETag` and `Last-Modified` validators by default, while **Azure CDN Standard from Microsoft** supports only `Last-Modified`.

ETag:

- **Azure CDN Standard/Premium from Edgio** supports `ETag` by default, while **Azure CDN Standard from Microsoft** doesn't.
- `ETag` defines a string that is unique for every file and version of a file. For example, `ETag: "17f0ddd99ed5bbe4edffdd6496d7131f"`.
- Introduced in HTTP 1.1 and is more current than `Last-Modified`. Useful when the last modified date is difficult to determine.
- Supports both strong validation and weak validation; however, Azure Content Delivery Network supports only strong validation. For strong validation, the two resource representations must be byte-for-byte identical.
- A cache validates a file that uses `ETag` by sending an `If-None-Match` header with one or more `ETag` validators in the request. For example, `If-None-Match: "17f0ddd99ed5bbe4edffdd6496d7131f"`. If the server's version matches an `ETag` validator on the list, it sends status code 304 (Not Modified) in its response. If the version is different, the server responds with status code 200 (OK) and the updated resource.

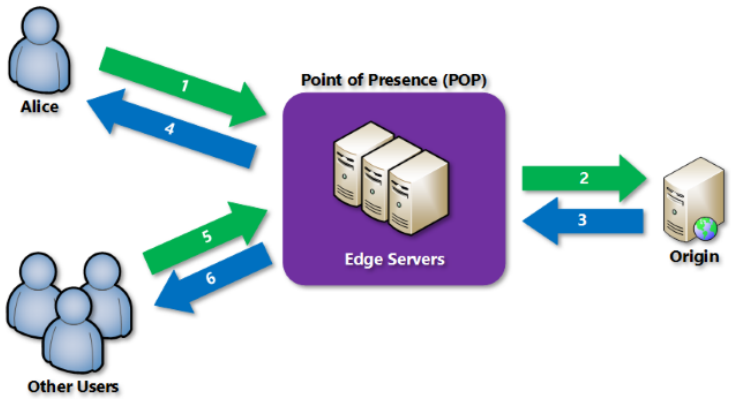
https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Last-Modified:

- For **Azure CDN Standard/Premium from Edgio** only, `Last-Modified` is used if `ETag` isn't part of the HTTP response.
- Specifies the date and time that the origin server has determined the resource was last modified. For example, `Last-Modified: Thu, 19 Oct 2017 09:28:00 GMT`.
- For content larger than 8 MB, origin backend servers should maintain consistent `Last-Modified` timestamps per asset. Returning inconsistent `Last-Modified` times from backend servers will cause validator mismatch errors and result in HTTP 5XX failures. Azure Storage may not support consistent `Last-Modified` timestamps across replicas, which can cause similar validator mismatch errors.
- A cache validates a file using `Last-Modified` by sending an `If-Modified-Since` header with a date and time in the request. The origin server compares that date with the `Last-Modified` header of the latest resource. If the resource hasn't been modified since the specified time, the server returns status code 304 (Not Modified) in its response. If the resource has been modified, the server returns status code 200 (OK) and the updated resource.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
2. The method as recited in claim 1 further comprising: (C) when it is determined that said content is not cached locally, then (C)(1) obtaining the content and then (C)(2) serving the content obtained in (C)(1).	When it is determined that said content is not cached locally, Microsoft then obtains the content and then serves the content obtained in (C)(1). For instance, if content is not cached locally, the node (e.g., an EdgeNode or Endpoint or Point of Presence) obtains the content (e.g. from a parent or origin server) and then serves it. How it works  1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <*endpoint name*>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user. 2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server. 3. The origin server returns the file to an edge server in the POP. 4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time to live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days. 5. More users can then request the same file by using the same URL that Alice used, and gets directed to the same POP. 6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience. https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Introduction to caching Caching is the process of storing data locally so that future requests for that data can be accessed more quickly. In the most common type of caching, web browser caching, a web browser stores copies of static data locally on a local hard drive. By using caching, the web browser can avoid making multiple round-trips to the server and instead access the same data locally, thus saving time and resources. Caching is well-suited for locally managing small, static data such as static images, CSS files, and JavaScript files. Similarly, caching is used by a content delivery network on edge servers close to the user to avoid requests traveling back to the origin and reducing end-user latency. Unlike a web browser cache, which is used only for a single user, the content delivery network has a shared cache. In a content delivery network shared cache, a file request by a user can be used by another user, which greatly decreases the number of requests to the origin server. Dynamic resources that change frequently or are unique to an individual user can't be cached. Those types of resources, however, can take advantage of dynamic site acceleration (DSA) optimization on the Azure content delivery network for performance improvements. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Caching can occur at multiple levels between the origin server and the end user: • Web server: Uses a shared cache (for multiple users). • Content delivery network: Uses a shared cache (for multiple users). • Internet service provider (ISP): Uses a shared cache (for multiple users). • Web browser: Uses a private cache (for one user). Each cache typically manages its own resource freshness and performs validation when a file is stale. This behavior is defined in the HTTP caching specification, RFC 7234. https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Resource freshness

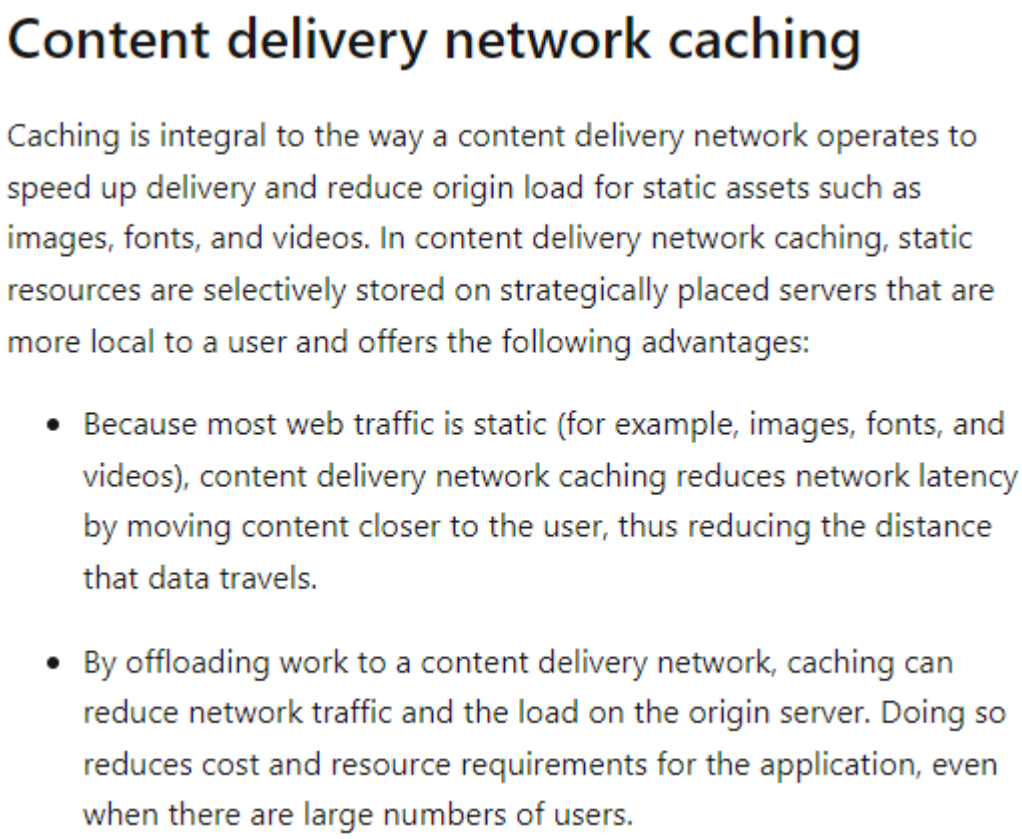
Since a cached resource can potentially be out-of-date, or stale (as compared to the corresponding resource on the origin server), it's important for any caching mechanism to control when content gets a refresh. To save time and bandwidth consumption, a cached resource isn't compared to the version on the origin server every time it's accessed. Instead, as long as a cached resource is considered to be fresh, it's assumed to be the most current version and is sent directly to the client. A cached resource is considered to be fresh when its age is less than the age or period defined by a cache setting. For example, when a browser reloads a web page, it verifies that each cached resource on your hard drive is fresh and loads it. If the resource isn't fresh (stale), an up-to-date copy is loaded from the server.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

Validation

If a resource is considered stale, the origin server gets asked to validate it to determine whether the data in the cache still matches what's on the origin server. If the file has been modified on the origin server, the cache updates its version of the resource. Otherwise, if the resource is fresh, the data is delivered directly from the cache without validating it first.

https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	 https://learn.microsoft.com/en-us/azure/cdn/cdn-how-caching-works
3. The method as recited in claim 1 wherein the request comprises an HTTP request and wherein the content comprises one or more resources.	Microsoft performs the method of claim 1, wherein the request comprises an HTTP request and wherein the content comprises one or more resources. For example, Microsoft hosts the web content of numerous third-party websites through its Azure services, as well as hosting Microsoft's own internet content, such as that of Office 365, Github, LinkedIn, Bing, and Xbox.

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Azure Content Delivery Network features Azure Content Delivery Network offers the following key features: • Dynamic site acceleration • Content delivery network caching rules • HTTPS custom domain support • Azure Diagnostics logs • File compression • Geo-filtering https://learn.microsoft.com/en-us/azure/cdn/cdn-overview

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Features and optimizations	Front Door Standard	Front Door Premium	Front Door (classic)	CDN Standard from Microsoft (classic)
Delivery and acceleration				
Static file delivery	✓	✓	✓	✓
Dynamic site delivery	✓	✓	✓	
WebSockets	Preview	Preview		
Domains and certs				
Custom domains	✓ (DNS TXT record based domain validation)	✓ (DNS TXT record based domain validation)	✓ (CNAME based validation)	✓ (CNAME based validation)
Prevalidated domain integration with Azure PaaS service	✓	✓		
HTTPS support	✓	✓	✓	✓
Custom domain HTTPS	✓	✓	✓	✓

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-cdn-comparison

HTTP/2 support	✓	✓	✓	✓

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-cdn-comparison

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Security The CDN can deliver content over HTTPS (SSL), by using the certificate provided by the CDN, as well as over standard HTTP. To avoid browser warnings about mixed content, you might need to use HTTPS to request static content that is displayed in pages loaded through HTTPS. https://learn.microsoft.com/en-us/azure/architecture/best-practices/cdn
4. The method as recited in claim 1 wherein the current cache policy is determined from a location distinct from the node.	Microsoft performs the method of claim 1, wherein the current cache policy is determined from a location distinct from the node. For instance, Microsoft Azure CDN and/or Azure Front Door offer content caching rules which determine time periods for expiration of content in the cache. A Microsoft Azure CDN node (e.g., EdgeNode or Endpoint or Point of Presence) determines the current cache policy from a distinct location (e.g., via propagation from Azure Portal or Azure Resource Manager).

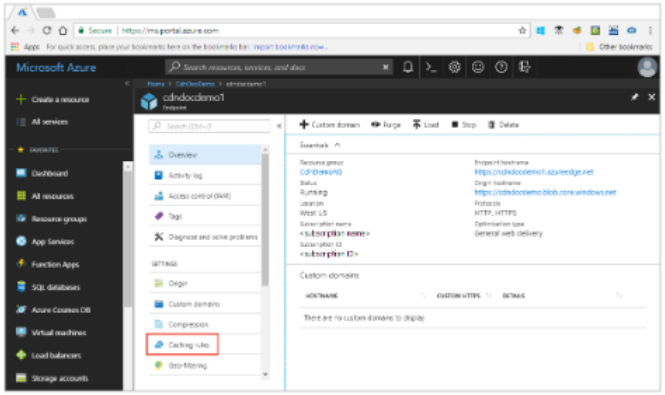
U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	Azure Content Delivery Network offers two ways to control how your files get cached: **Caching rules:** Azure Content Delivery Network provides two types of caching rules: global and custom. • Global caching rules - You can set one global caching rule for each endpoint in your profile, which affects all requests to the endpoint. The global caching rule overrides any HTTP cache-directive headers, if set. • Custom caching rules - You can set one or more custom caching rules for each endpoint in your profile. Custom caching rules match specific paths and file extensions, get processed in order, and override the global caching rule, if set. **Query string caching:** You can adjust how the Azure content delivery network treats caching for requests with query strings. For information, see Control Azure Content Delivery Network caching behavior with query strings. If the file isn't cacheable, the query string caching setting has no effect, based on caching rules and content delivery network default behaviors. For information about default caching behavior and caching directive headers, see How caching works. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

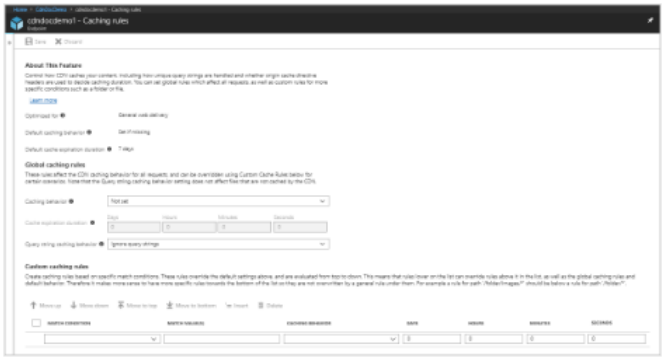
Accessing Azure Content Delivery Network caching rules

1. Open the Azure portal, select a content delivery network profile, then select an endpoint.

2. In the left pane under Settings, select **Caching rules**.

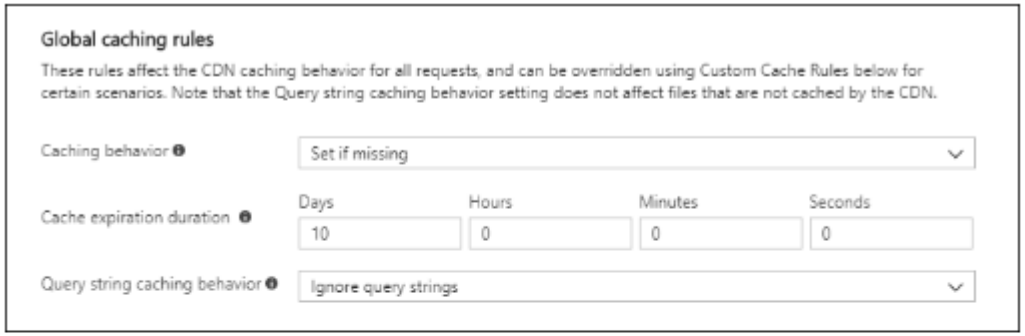


The **Caching rules** page appears.



https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Caching behavior settings For global and custom caching rules, you can specify the following **Caching behavior** settings: - **Bypass cache:** Don't cache and ignore origin-provided cache-directive headers. - **Override:** Ignore origin-provided cache duration; use the provided cache duration instead. This setting doesn't override cache-control: no-cache. ⓘ Note For **Azure CDN from Microsoft** profiles, cache expiration override is only applicable to status codes 200 and 206. - **Set if missing:** Honor origin-provided cache-directive headers, if they exist; otherwise, use the provided cache duration. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	  https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Cache expiration duration For global and custom caching rules, you can specify the cache expiration duration in days, hours, minutes, and seconds: • For the **Override** and **Set if missing Caching behavior** settings, valid cache durations range between 0 seconds and 366 days. For a value of 0 seconds, the content delivery network caches the content, but must revalidate each request with the origin server. • For the **Bypass cache** setting, the cache duration gets automatically set to 0 seconds, which isn't a modifiable value. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Custom caching rules match conditions For custom cache rules, two match conditions are available: • **Path:** This condition matches the path of the URL, excluding the domain name, and supports the wildcard symbol (*). For example, */myfile.html*, */my/folder/***, *and /my/images/.jpg*. The maximum length is 260 characters. • **Extension:** This condition matches the file extension of the requested file. You can provide a list of comma-separated file extensions to match. For example, *.jpg*, *.mp3*, or *.png*. The maximum number of extensions is 50 and the maximum number of characters per extension is 16. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Global and custom rule processing order

Global and custom caching rules get processed in the following order:

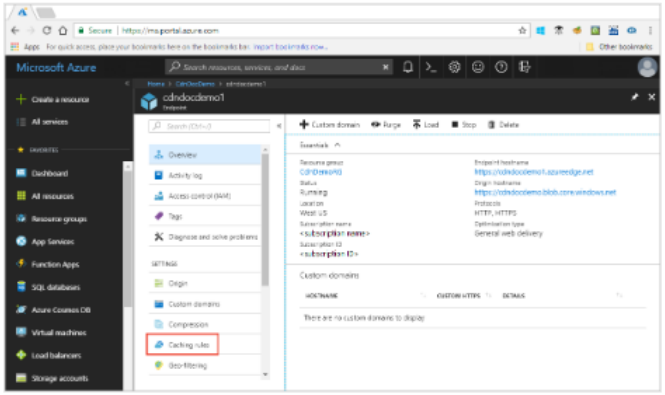
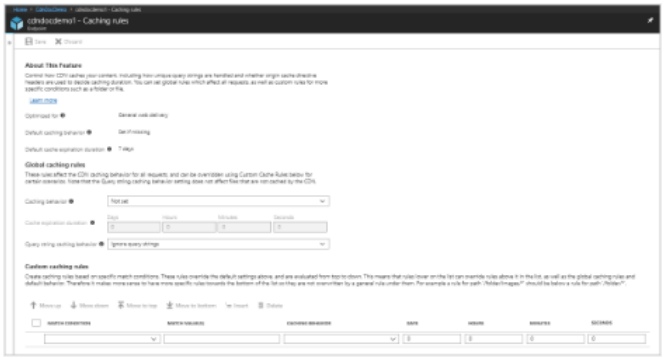
- Global caching rules take precedence over the default content delivery network caching behavior (HTTP cache-directive header settings).
- Custom caching rules take precedence over global caching rules, where they apply. Custom caching rules get processed in order from top to bottom. That is, if a request matches both conditions, rules at the bottom of the list take precedence over rules at the top of the list. Therefore, you should place more specific rules lower in the list.

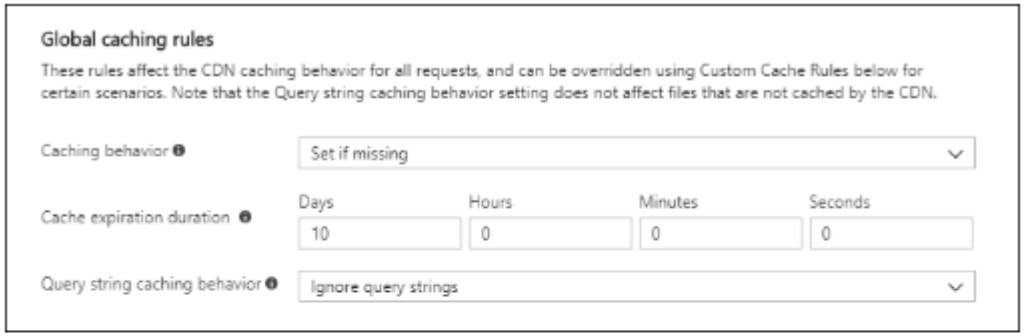
Example:

- Global caching rule:
  - Caching behavior: **Override**
  - Cache expiration duration: One day
- Custom caching rule #1:
  - Match condition: **Path**
  - Match value: */home/**
  - Caching behavior: **Override**
  - Cache expiration duration: Two days
- Custom caching rule #2:
  - Match condition: **Extension**
  - Match value: *.html*
  - Caching behavior: **Set if missing**
  - Cache expiration duration: Three days

https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	When you set these rules, a request for <*endpoint hostname*>.azureedge.net/home/index.html triggers custom caching rule #2, which get set to: **Set if missing** and 3 days. Therefore, if the *index.html* file has `Cache-Control` or `Expires` HTTP headers, they get honored; otherwise, if you don't set these headers, the file gets cached for three days. ⓘ Note Files that are cached before a rule change maintain their origin cache duration setting. To reset their cache durations, you must purge the file. Azure Content Delivery Network configuration changes can take some time to propagate through the network: • For **Azure CDN Standard from Edgio** profiles, propagation usually completes in 10 minutes. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules
5. The method as recited in claim 4 wherein the location is a database.	Microsoft performs the method of claim 4, wherein the location is a database. For instance, caching policies such as rules are stored in a database associated with Azure Portal or Azure Resource Manager for retrieval by the CDN node.

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	# Accessing Azure Content Delivery Network caching rules 1. Open the Azure portal, select a content delivery network profile, then select an endpoint. 2. In the left pane under Settings, select **Caching rules**.  The **Caching rules** page appears.  https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	  https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities

Global and custom rule processing order

Global and custom caching rules get processed in the following order:

- Global caching rules take precedence over the default content delivery network caching behavior (HTTP cache-directive header settings).
- Custom caching rules take precedence over global caching rules, where they apply. Custom caching rules get processed in order from top to bottom. That is, if a request matches both conditions, rules at the bottom of the list take precedence over rules at the top of the list. Therefore, you should place more specific rules lower in the list.

Example:

- Global caching rule:
  - Caching behavior: **Override**
  - Cache expiration duration: One day
- Custom caching rule #1:
  - Match condition: **Path**
  - Match value: */home/**
  - Caching behavior: **Override**
  - Cache expiration duration: Two days
- Custom caching rule #2:
  - Match condition: **Extension**
  - Match value: *.html*
  - Caching behavior: **Set if missing**
  - Cache expiration duration: Three days

https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
	When you set these rules, a request for <*endpoint hostname*>.azureedge.net/home/index.html triggers custom caching rule #2, which get set to: **Set if missing** and 3 days. Therefore, if the *index.html* file has `Cache-Control` or `Expires` HTTP headers, they get honored; otherwise, if you don't set these headers, the file gets cached for three days. ⓘ Note Files that are cached before a rule change maintain their origin cache duration setting. To reset their cache durations, you must purge the file. Azure Content Delivery Network configuration changes can take some time to propagate through the network: • For **Azure CDN Standard from Edgio** profiles, propagation usually completes in 10 minutes. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules See also Claim 4.
6. The method as recited in claim 1 wherein the cache	Microsoft performs the method of claim 1, wherein the cache policy relates to caching content within the CDN. For instance, caching policies such as rules relate to caching content within the CDN.

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
policy relates to caching content within the CDN.	Azure Content Delivery Network offers two ways to control how your files get cached: **Caching rules:** Azure Content Delivery Network provides two types of caching rules: global and custom. • Global caching rules - You can set one global caching rule for each endpoint in your profile, which affects all requests to the endpoint. The global caching rule overrides any HTTP cache-directive headers, if set. • Custom caching rules - You can set one or more custom caching rules for each endpoint in your profile. Custom caching rules match specific paths and file extensions, get processed in order, and override the global caching rule, if set. **Query string caching:** You can adjust how the Azure content delivery network treats caching for requests with query strings. For information, see Control Azure Content Delivery Network caching behavior with query strings. If the file isn't cacheable, the query string caching setting has no effect, based on caching rules and content delivery network default behaviors. For information about default caching behavior and caching directive headers, see How caching works. https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules See also Claim 4.

U.S. Patent No. 10,701,173	Microsoft Accused Functionalities
7. The method as recited in claim 6 wherein the cache policy also relates to caching content outside the CDN.	Microsoft performs the method of claim 6, wherein the cache policy also relates to caching content outside the CDN. For instance, cache policies such as rules related to cache expiration also relate to caching content outside the CDN (e.g. HTTP cache-directive header settings applicable to content outside of the CDN). **Global and custom rule processing order** Global and custom caching rules get processed in the following order: • Global caching rules take precedence over the default content delivery network caching behavior (HTTP cache-directive header settings). • Custom caching rules take precedence over global caching rules, where they apply. Custom caching rules get processed in order from top to bottom. That is, if a request matches both conditions, rules at the bottom of the list take precedence over rules at the top of the list. Therefore, you should place more specific rules lower in the list. **Example:** • Global caching rule: ○ Caching behavior: **Override** ○ Cache expiration duration: One day • Custom caching rule #1: ○ Match condition: **Path** ○ Match value: */home/** ○ Caching behavior: **Override** ○ Cache expiration duration: Two days • Custom caching rule #2: ○ Match condition: **Extension** ○ Match value: *.html* ○ Caching behavior: **Set if missing** ○ Cache expiration duration: Three days https://learn.microsoft.com/en-us/azure/cdn/cdn-caching-rules See also Claim 4.