Amazon EC2 backup and recovery with snapshots and AMIs

PDF (/pdfs/prescriptive-guidance/latest/backup-recovery/backup-recovery.pdf#ec2-backup)

RSS (backup-recovery.rss) Focus mode

Consider whether you need to create a full backup of an EC2 instance with an Amazon Machine Image (AMI) or take a snapshot of an individual volume.

Using AMIs or Amazon EBS snapshots for backups

An AMI includes the following:

- One or more snapshots. Instance-store-backed AMIs include a template for the root volume of the instance (for example, an operating system, an application server, and applications).
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

> **Note**
>
> In most cases, AMIs for Windows, Red Hat, SUSE, and SQL Server require correct licensing information to be present on the AMI. For more

Wiz - EXHIBIT 1131
Wiz, Inc. v. Orca Security Ltd.
IPR2024-01191

information, see Understand AMI billing information (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ami-billing-info.html) . When creating an AMI from a snapshot, the `RegisterImage` operation derives the correct billing information from the snapshot's metadata, but this requires the appropriate metadata to be present. To verify if the correct billing information was applied, check the **Platform details** field on the new AMI. If the field is empty or doesn't match the expected operating system code (for example, Windows, Red Hat, SUSE, or SQL), the AMI creation was unsuccessful, and you should discard the AMI and follow the instructions in Create an AMI from an instance (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-ebs.html#how-to-create-ebs-ami) .

You can use AMIs to launch new instances with preconfigured software and data. You can create AMIs when you want to establish a baseline, which is a reusable configuration for launching more instances. When you create an AMI of an existing EC2 instance, a snapshot is taken for all the volumes that are attached to the instance. The snapshot includes the device mappings.

You can't use snapshots to launch a new instance, but you can use them to replace volumes on an existing instance. If you experience data corruption or a volume failure, you can create a volume from a snapshot that you have taken and replace the old volume. You can also use snapshots to provision new volumes and attach them during a new instance launch.

If you are using platform and application AMIs maintained and published by AWS or from the AWS Marketplace, consider maintaining separate volumes for your data. You can back up your data volumes as snapshots that are separate from the operating system and application volumes. Then use the data volume snapshots with newly updated AMIs published by AWS or from the AWS Marketplace. This approach requires careful testing and planning to back up and restore all custom data, including configuration information, on the newly published AMIs.

The restore process is affected by your choice between AMI backups or snapshot backups. If you create AMIs to serve as instance backups, you must launch an EC2 instance from the AMI as a part of your restore process. You might also need to shut down the existing instance to avoid potential collisions. An example of a potential collision is security identifiers (SIDs) for domain-joined Windows instances. The restore process for snapshots might require you to detach the existing volume and attach the newly restored volume. Or you might need to

make a configuration change to point your applications to the newly attached volume.

AWS Backup supports both instance-level backups as AMIs and volume-level backups as separate snapshots:

- For a full backup of all EBS volumes on the instance, create an AMI of the EC2 instance (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-ebs.html) . When you want to roll back, use the launch instance wizard to create an instance. In the instance launch wizard, choose **My AMIs**.
- To back up an individual volume, create a snapshot (https://docs.aws.amazon.com/ebs/latest/userguide/ebs-creating-snapshot.html) . To restore the snapshot, see Create a volume from a snapshot (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-volume.html#ebs-create-volume-from-snapshot) . You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI).

The cost of an instance AMI is the storage of all the volumes on the instance, but not the metadata. The cost for an EBS snapshot is the storage of the individual volume. For more information about volume storage costs, see the Amazon EBS pricing page (https://aws.amazon.com/ebs/pricing/) .

Server volumes

EBS volumes are the primary persistent storage option for Amazon EC2. You can use this block storage for structured data, such as databases, or unstructured data, such as files in a file system on a volume.

EBS volumes are placed in a specific Availability Zone. The volumes are replicated across multiple servers to prevent the loss of data from the failure of any single component. Failure refers to a complete or partial loss of the volume, depending on the size and performance of the volume.

EBS volumes are designed for an annual failure rate (AFR) of 0.1-0.2 percent. This makes EBS volumes 20 times more reliable than typical commodity disk drives, which fail with an AFR of around 4 percent. For example, if you have 1,000 EBS volumes running for 1 year, you should expect one or two volumes will have a failure.

Amazon EBS also supports a snapshot feature for taking point-in-time backups of your data. All EBS volume types offer durable snapshot capabilities and are

designed for 99.999 percent availability. For more information, see the Amazon Compute Service Level Agreement (https://aws.amazon.com/ec2/sla/) .

Amazon EBS provides the ability to create snapshots (backups) of any EBS volume. A snapshot is a base feature for creating backups of your EBS volumes. A snapshot takes a copy of the EBS volume and places it in Amazon S3, where it is stored redundantly in multiple Availability Zones. The initial snapshot is a full copy of the volume; ongoing snapshots store incremental block-level changes only. See the Amazon EBS documentation (https://docs.aws.amazon.com/ebs/latest/userguide/ebs-creating-snapshot.html) for details on how to create Amazon EBS snapshots.

You can perform a restore operation, delete a snapshot, or update the snapshot metadata, such as tags, associated with the snapshot from the Amazon EC2 console (https://docs.aws.amazon.com/prescriptive-guidance/latest/backup-recovery/restore.html) in the same Region that you took the snapshot.

Restoring a snapshot creates a new Amazon EBS volume with full volume data. If you need only a partial restore, you can attach the volume to the running instance under a different device name. Then mount it, and use operating system copy commands to copy the data from the backup volume to the production volume.

Amazon EBS snapshots can also be copied between AWS Regions by using the Amazon EBS snapshot copy capability, as described in the Amazon EBS documentation (https://docs.aws.amazon.com/ebs/latest/userguide/ebs-copy-snapshot.html) . You can use this feature to store your backup in another Region without having to manage the underlying replication technology.

Establishing separate server volumes

You may already use a standard set of separate volumes for the operating system, logs, applications, and data. By establishing separate server volumes, you can reduce the scope of impact when there are application or platform failures caused by disk space exhaustion. This risk is usually greater with physical hard drives, because you don't have the flexibility to expand volumes quickly. With physical drives, you must purchase the new drives, back up the data, and then restore the data on the new drives. With AWS, this risk is greatly reduced because you can use Amazon EBS to expand your provisioned volumes. For more information, see the AWS documentation (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/modify-volume-requirements.html) .

Maintain separate volumes for application data, user data, logs, and swap files so that you can use separate backup and restore policies for these resources. By separating volumes for your data, you can also use different volume types based on the performance and storage requirements for the data. You can then optimize and fine-tune your costs for different workloads.

Considerations for instance store volumes

An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance stores are ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content. They are also preferable for data that are replicated across a fleet of instances, such as a load balanced pool of web servers.

The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data in the instance store is lost under any of the following circumstances.

- The underlying drive fails.
- The instance stops.
- The instance terminates.

Therefore, do not rely on an instance store for valuable, long-term data. Instead, use more durable data storage, such as Amazon S3, Amazon EBS, or Amazon EFS.

A common strategy with instance store volumes is to persist necessary data to Amazon S3 regularly as needed, based on the recovery point objective (RPO) and recovery time objective (RTO). You can then download the data from Amazon S3 to your instance store when a new instance is launched. You can also upload the data to Amazon S3 before an instance is stopped. For persistence, create an EBS volume, attach it to your instance, and copy the data from the instance store volume to the EBS volume on a periodic basis. For more information, see the AWS Knowledge Center (https://aws.amazon.com/premiumsupport/knowledge-center/back-up-instance-store-ebs/) .

Tagging and enforcing standards for EBS snapshots and AMIs

Tagging all your AWS resources is an important practice for cost allocation, auditing, troubleshooting, and notification. Tagging is important for EBS volumes so that the pertinent information required to manage and restore volumes is present. Tags are not automatically copied from EC2 instances to AMIs or from source volumes to snapshots. Make sure that your backup process includes the relevant tags from these sources. This helps you to set the snapshot metadata, such as access policies, attachment information, and cost allocation, to use these backups in the future. For more information on tagging your AWS resources, refer to the tagging best practices technical paper (https://d1.awsstatic.com/whitepapers/aws-tagging-best-practices.pdf) .

In addition to the tags you use for all AWS resources, use the following backup-specific tags:

- Source instance ID
- Source volume ID (for snapshots)
- Recovery point description

You can enforce tagging policies by using AWS Config rules and IAM permissions. IAM supports enforced tag usage, so you can write IAM policies that mandate the use of specific tags when acting on Amazon EBS snapshots. If a `CreateSnapshot` operation is attempted without the tags defined in the IAM permissions policy granting rights, the snapshot creation fails with access denied. For more information, see the blog post on tagging Amazon EBS snapshots on creation and implementing stronger security policies (https://aws.amazon.com/blogs/compute/tag-amazon-ebs-snapshots-on-creation-and-implement-stronger-security-policies/) .

You can use AWS Config rules to evaluate the configuration settings of your AWS resources automatically. To help you get started, AWS Config provides customizable, predefined rules called managed rules. You can also create your own custom rules. While AWS Config continuously tracks configuration changes among your resources, it checks whether these changes violate any of the conditions in your rules. If a resource violates a rule, AWS Config flags the resource and the rule

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.