SecurityAudit

PDF (/pdfs/aws-managed-policy/latest/reference/aws-managed-policy.pdf#SecurityAudit)

Focus mode

On this page

Using this policy(#SecurityAudit-how-to-use)
Policy details(#SecurityAudit-details)
Policy version(#SecurityAudit-version)
JSON policy document(#SecurityAudit-json)
Learn more(#SecurityAudit-learn-more)

Adjust text size

14 20

Description: The security audit template grants access to read security configuration metadata. It is useful for software that audits the configuration of an AWS account.

`SecurityAudit` is an AWS managed policy (https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#aws-managed-policies) .

Using this policy

You can attach `SecurityAudit` to your users, groups, and roles.

Policy details

Wiz - EXHIBIT 1138
Wiz, Inc. v. Orca Security Ltd.
IPR2024-00865

- **Type**: AWS managed policy
- **Creation time**: February 06, 2015, 18:41 UTC
- **Edited time:** May 05, 2025, 17:22 UTC
- **ARN**: `arn:aws:iam::aws:policy/SecurityAudit`

Policy version

Policy version: v51 (default)

The policy's default version is the version that defines the permissions for the policy. When a user or role with the policy makes a request to access an AWS resource, AWS checks the default version of the policy to determine whether to allow the request.

JSON policy document

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "BaseSecurityAuditStatement",
      "Effect" : "Allow",
      "Action" : [
        "a4b:ListSkills",
        "access-analyzer:GetAnalyzedResource",
        "access-analyzer:GetAnalyzer",
        "access-analyzer:GetArchiveRule",
        "access-analyzer:GetFinding",
        "access-analyzer:ListAnalyzedResources",
        "access-analyzer:ListAnalyzers",
        "access-analyzer:ListArchiveRules",
        "access-analyzer:ListFindings",
        "access-analyzer:ListTagsForResource",
        "account:GetAlternateContact",
        "account:GetPrimaryEmail",
        "account:GetRegionOptStatus",
        "acm-pca:DescribeCertificateAuthority",
        "acm-pca:DescribeCertificateAuthorityAuditReport",
```

```
            "acm-pca:GetPolicy",
            "acm-pca:ListCertificateAuthorities",
            "acm-pca:ListPermissions",
            "acm-pca:ListTags",
            "acm:Describe*",
            "acm:List*",
            "airflow:GetEnvironment",
            "airflow:ListEnvironments",
            "appflow:ListFlows",
            "appflow:ListTagsForResource",
            "application-autoscaling:Describe*",
            "appmesh:Describe*",
            "appmesh:List*",
            "apprunner:DescribeAutoScalingConfiguration",
            "apprunner:DescribeCustomDomains",
            "apprunner:DescribeObservabilityConfiguration",
            "apprunner:DescribeService",
            "apprunner:DescribeVpcConnector",
            "apprunner:DescribeVpcIngressConnection",
            "apprunner:ListAutoScalingConfigurations",
            "apprunner:ListConnections",
            "apprunner:ListObservabilityConfigurations",
            "apprunner:ListOperations",
            "apprunner:ListServices",
            "apprunner:ListTagsForResource",
            "apprunner:ListVpcConnectors",
            "apprunner:ListVpcIngressConnections",
            "appsync:GetApiCache",
            "appsync:List*",
            "athena:GetWorkGroup",
            "athena:List*",
            "auditmanager:GetAccountStatus",

        "auditmanager:ListAssessmentControlInsightsByControlDomain",
            "auditmanager:ListAssessmentFrameworks",
            "auditmanager:ListAssessmentFrameworkShareRequests",
            "auditmanager:ListAssessmentReports",
            "auditmanager:ListAssessments",
            "auditmanager:ListControlDomainInsights",
            "auditmanager:ListControlDomainInsightsByAssessment",
```

```
            "auditmanager:ListControlInsightsByControlDomain",
            "auditmanager:ListControls",
            "auditmanager:ListNotifications",
            "auditmanager:ListTagsForResource",
            "autoscaling-plans:DescribeScalingPlans",
            "autoscaling:Describe*",
            "backup:DescribeGlobalSettings",
            "backup:DescribeRegionSettings",
            "backup:GetBackupVaultAccessPolicy",
            "backup:GetBackupVaultNotifications",
            "backup:ListBackupVaults",
            "backup:ListTags",
            "batch:DescribeComputeEnvironments",
            "batch:DescribeJobDefinitions",
            "bedrock:GetCustomModel",
            "bedrock:GetModelInvocationLoggingConfiguration",
            "bedrock:ListCustomModels",
            "bedrock:ListTagsForResource",
            "braket:SearchJobs",
            "braket:SearchQuantumTasks",
            "chime:List*",
            "cleanrooms:BatchGetCollaborationAnalysisTemplate",
            "cleanrooms:BatchGetSchema",
            "cleanrooms:BatchGetSchemaAnalysisRule",
            "cleanrooms:GetAnalysisTemplate",
            "cleanrooms:GetCollaboration",
            "cleanrooms:GetCollaborationAnalysisTemplate",

        "cleanrooms:GetCollaborationConfiguredAudienceModelAssociation",
            "cleanrooms:GetCollaborationIdNamespaceAssociation",
            "cleanrooms:GetCollaborationPrivacyBudgetTemplate",
            "cleanrooms:GetConfiguredAudienceModelAssociation",
            "cleanrooms:GetConfiguredTable",
            "cleanrooms:GetConfiguredTableAnalysisRule",
            "cleanrooms:GetConfiguredTableAssociation",
            "cleanrooms:GetConfiguredTableAssociationAnalysisRule",
            "cleanrooms:GetIdMappingTable",
            "cleanrooms:GetIdNamespaceAssociation",
            "cleanrooms:GetMembership",
            "cleanrooms:GetPrivacyBudgetTemplate",
```

```
            "cleanrooms:GetProtectedQuery",
            "cleanrooms:GetSchema",
            "cleanrooms:GetSchemaAnalysisRule",
            "cleanrooms:ListAnalysisTemplates",
            "cleanrooms:ListCollaborationAnalysisTemplates",
            "cleanrooms:ListCollaborationConfiguredAudienceModelAssociations"
            ,
            "cleanrooms:ListCollaborationIdNamespaceAssociations",
            "cleanrooms:ListCollaborationPrivacyBudgetTemplates",
            "cleanrooms:ListCollaborationPrivacyBudgets",
            "cleanrooms:ListCollaborations",
            "cleanrooms:ListConfiguredAudienceModelAssociations",
            "cleanrooms:ListConfiguredTableAssociations",
            "cleanrooms:ListConfiguredTables",
            "cleanrooms:ListIdMappingTables",
            "cleanrooms:ListIdNamespaceAssociations",
            "cleanrooms:ListMembers",
            "cleanrooms:ListMemberships",
            "cleanrooms:ListPrivacyBudgetTemplates",
            "cleanrooms:ListPrivacyBudgets",
            "cleanrooms:ListProtectedQueries",
            "cleanrooms:ListSchemas",
            "cleanrooms:ListTagsForResource",
            "cleanrooms:PreviewPrivacyImpact",
            "cloud9:Describe*",
            "cloud9:ListEnvironments",
            "clouddirectory:ListDirectories",
            "cloudformation:DescribeStack*",
            "cloudformation:GetStackPolicy",
            "cloudformation:GetTemplate",
            "cloudformation:ListStack*",
            "cloudfront:Get*",
            "cloudfront:List*",
            "cloudsearch:DescribeDomainEndpointOptions",
            "cloudsearch:DescribeDomains",
            "cloudsearch:DescribeServiceAccessPolicies",
            "cloudtrail:DescribeTrails",
            "cloudtrail:GetEventSelectors",
            "cloudtrail:GetInsightSelectors",
```

```
            "cloudtrail:GetTrail",
            "cloudtrail:GetTrailStatus",
            "cloudtrail:ListTags",
            "cloudtrail:ListTrails",
            "cloudtrail:LookupEvents",
            "cloudwatch:Describe*",
            "cloudwatch:GetDashboard",
            "cloudwatch:ListDashboards",
            "cloudwatch:ListTagsForResource",
            "codeartifact:GetDomainPermissionsPolicy",
            "codeartifact:GetRepositoryPermissionsPolicy",
            "codeartifact:ListRepositories",
            "codebuild:BatchGetProjects",
            "codebuild:GetResourcePolicy",
            "codebuild:ListProjects",
            "codebuild:ListSourceCredentials",
            "codecommit:BatchGetRepositories",
            "codecommit:GetBranch",
            "codecommit:GetObjectIdentifier",
            "codecommit:GetRepository",
            "codecommit:GetRepositoryTriggers",
            "codecommit:List*",
            "codedeploy:Batch*",
            "codedeploy:Get*",
            "codedeploy:List*",
            "codepipeline:GetJobDetails",
            "codepipeline:GetPipeline",
            "codepipeline:GetPipelineExecution",
            "codepipeline:GetPipelineState",
            "codepipeline:ListPipelines",
            "codestar:Describe*",
            "codestar:List*",
            "cognito-identity:Describe*",
            "cognito-identity:GetIdentityPoolRoles",
            "cognito-identity:ListIdentityPools",
            "cognito-identity:ListTagsForResource",
            "cognito-idp:Describe*",
            "cognito-idp:ListDevices",
            "cognito-idp:ListGroups",
            "cognito-idp:ListIdentityProviders",
```

```
            "cognito-idp:ListResourceServers",
            "cognito-idp:ListTagsForResource",
            "cognito-idp:ListUserImportJobs",
            "cognito-idp:ListUserPoolClients",
            "cognito-idp:ListUserPools",
            "cognito-idp:ListUsers",
            "cognito-idp:ListUsersInGroup",
            "cognito-sync:Describe*",
            "cognito-sync:List*",
            "comprehend:Describe*",
            "comprehend:List*",
            "comprehendmedical:ListICD10CMInferenceJobs",
            "comprehendmedical:ListPHIDetectionJobs",
            "comprehendmedical:ListRxNormInferenceJobs",
            "comprehendmedical:ListSNOMEDCTInferenceJobs",
            "config:BatchGetAggregateResourceConfig",
            "config:BatchGetResourceConfig",
            "config:Deliver*",
            "config:Describe*",
            "config:Get*",
            "config:List*",
            "config:SelectAggregateResourceConfig",
            "config:SelectResourceConfig",
            "connect:ListApprovedOrigins",
            "connect:ListInstanceAttributes",
            "connect:ListInstances",
            "connect:ListInstanceStorageConfigs",
            "connect:ListIntegrationAssociations",
            "connect:ListLambdaFunctions",
            "connect:ListLexBots",
            "connect:ListSecurityKeys",
            "databrew:DescribeDataset",
            "databrew:DescribeProject",
            "databrew:ListJobs",
            "databrew:ListProjects",
            "dataexchange:ListDataSets",
            "datapipeline:DescribeObjects",
            "datapipeline:DescribePipelines",
            "datapipeline:EvaluateExpression",
            "datapipeline:GetPipelineDefinition",
```

```
            "datapipeline:ListPipelines",
            "datapipeline:QueryObjects",
            "datapipeline:ValidatePipelineDefinition",
            "datasync:Describe*",
            "datasync:List*",
            "dax:Describe*",
            "dax:ListTags",
            "deepracer:ListModels",
            "detective:GetGraphIngestState",
            "detective:ListGraphs",
            "detective:ListMembers",
            "devicefarm:ListProjects",
            "directconnect:Describe*",
            "discovery:DescribeAgents",
            "discovery:DescribeConfigurations",
            "discovery:DescribeContinuousExports",
            "discovery:DescribeExportConfigurations",
            "discovery:DescribeExportTasks",
            "discovery:DescribeImportTasks",
            "dms:Describe*",
            "dms:ListTagsForResource",
            "docdb-elastic:ListClusters",
            "ds:DescribeDirectories",
            "dynamodb:DescribeContinuousBackups",
            "dynamodb:DescribeExport",
            "dynamodb:DescribeGlobalTable",
            "dynamodb:DescribeKinesisStreamingDestination",
            "dynamodb:DescribeTable",
            "dynamodb:DescribeTimeToLive",
            "dynamodb:ListBackups",
            "dynamodb:ListExports",
            "dynamodb:ListGlobalTables",
            "dynamodb:ListStreams",
            "dynamodb:ListTables",
            "dynamodb:ListTagsOfResource",
            "ec2:Describe*",
            "ec2:GetEbsEncryptionByDefault",
            "ec2:GetImageBlockPublicAccessState",
            "ec2:GetManagedPrefixListAssociations",
            "ec2:GetManagedPrefixListEntries",
```

```
            "ec2:GetNetworkInsightsAccessScopeAnalysisFindings",
            "ec2:GetNetworkInsightsAccessScopeContent",
            "ec2:GetTransitGatewayAttachmentPropagations",
            "ec2:GetTransitGatewayMulticastDomainAssociations",
            "ec2:GetTransitGatewayPrefixListReferences",
            "ec2:GetTransitGatewayRouteTableAssociations",
            "ec2:GetTransitGatewayRouteTablePropagations",
            "ec2:SearchTransitGatewayRoutes",
            "ecr-public:DescribeImages",
            "ecr-public:DescribeImageTags",
            "ecr-public:DescribeRegistries",
            "ecr-public:DescribeRepositories",
            "ecr-public:GetRegistryCatalogData",
            "ecr-public:GetRepositoryCatalogData",
            "ecr-public:GetRepositoryPolicy",
            "ecr-public:ListTagsForResource",
            "ecr:BatchGetRepositoryScanningConfiguration",
            "ecr:DescribeImages",
            "ecr:DescribeImageScanFindings",
            "ecr:DescribeRegistry",
            "ecr:DescribeRepositories",
            "ecr:GetLifecyclePolicy",
            "ecr:GetRegistryPolicy",
            "ecr:GetRegistryScanningConfiguration",
            "ecr:GetRepositoryPolicy",
            "ecr:ListImages",
            "ecr:ListTagsForResource",
            "ecs:Describe*",
            "ecs:List*",
            "eks:DescribeCluster",
            "eks:DescribeFargateProfile",
            "eks:DescribeNodeGroup",
            "eks:ListClusters",
            "eks:ListFargateProfiles",
            "eks:ListNodeGroups",
            "eks:ListTagsForResource",
            "eks:ListUpdates",
            "elastic-inference:DescribeAccelerators",
            "elasticache:Describe*",
            "elasticache:ListTagsForResource",
```

```
            "elasticbeanstalk:Describe*",
            "elasticbeanstalk:ListTagsForResource",
            "elasticfilesystem:DescribeAccessPoints",
            "elasticfilesystem:DescribeAccountPreferences",
            "elasticfilesystem:DescribeBackupPolicy",
            "elasticfilesystem:DescribeFileSystemPolicy",
            "elasticfilesystem:DescribeFileSystems",
            "elasticfilesystem:DescribeLifecycleConfiguration",
            "elasticfilesystem:DescribeMountTargets",
            "elasticfilesystem:DescribeMountTargetSecurityGroups",
            "elasticfilesystem:DescribeReplicationConfigurations",
            "elasticfilesystem:DescribeTags",
            "elasticloadbalancing:Describe*",
            "elasticmapreduce:Describe*",
            "elasticmapreduce:GetAutoTerminationPolicy",
            "elasticmapreduce:GetBlockPublicAccessConfiguration",
            "elasticmapreduce:GetManagedScalingPolicy",
            "elasticmapreduce:ListClusters",
            "elasticmapreduce:ListInstances",
            "elasticmapreduce:ListSecurityConfigurations",
            "elastictranscoder:ListPipelines",
            "emr-serverless:GetApplication",
            "emr-serverless:ListApplications",
            "emr-serverless:ListJobRuns",
            "entityresolution:GetIdNamespace",
            "es:Describe*",
            "es:GetCompatibleVersions",
            "es:ListDomainNames",
            "es:ListElasticsearchInstanceTypeDetails",
            "es:ListElasticsearchVersions",
            "es:ListTags",
            "events:Describe*",
            "events:List*",
            "events:TestEventPattern",
            "finspace:ListEnvironments",
            "finspace:ListKxEnvironments",
            "firehose:Describe*",
            "firehose:List*",
            "fms:ListComplianceStatus",
            "fms:ListPolicies",
```

```
"forecast:ListDatasets",
"frauddetector:GetDetectors",
"fsx:Describe*",
"fsx:List*",
"gamelift:ListBuilds",
"gamelift:ListFleets",
"geo:ListMaps",
"glacier:DescribeVault",
"glacier:GetDataRetrievalPolicy",
"glacier:GetVaultAccessPolicy",
"glacier:GetVaultLock",
"glacier:ListVaults",
"globalaccelerator:Describe*",
"globalaccelerator:List*",
"glue:GetCrawlers",
"glue:GetDatabases",
"glue:GetDataCatalogEncryptionSettings",
"glue:GetDevEndpoints",
"glue:GetJobs",
"glue:GetResourcePolicy",
"glue:GetSecurityConfiguration",
"glue:GetSecurityConfigurations",
"glue:GetTags",
"grafana:ListWorkspaces",
"greengrass:List*",
"guardduty:DescribePublishingDestination",
"guardduty:Get*",
"guardduty:List*",
"health:DescribeAffectedAccountsForOrganization",
"health:DescribeAffectedEntities",
"health:DescribeAffectedEntitiesForOrganization",
"health:DescribeEntityAggregates",
"health:DescribeEventAggregates",
"health:DescribeEventDetails",
"health:DescribeEventDetailsForOrganization",
"health:DescribeEvents",
"health:DescribeEventsForOrganization",
"health:DescribeEventTypes",
"health:DescribeHealthServiceStatusForOrganization",
"healthlake:ListFHIRDatastores",
```

```
            "honeycode:ListTables",
            "iam:GenerateCredentialReport",
            "iam:GenerateServiceLastAccessedDetails",
            "iam:Get*",
            "iam:List*",
            "iam:SimulateCustomPolicy",
            "iam:SimulatePrincipalPolicy",
            "identitystore:DescribeGroupMembership",
            "identitystore:GetGroupId",
            "identitystore:GetGroupMembershipId",
            "identitystore:GetUserId",
            "identitystore:IsMemberInGroups",
            "identitystore:ListGroupMemberships",
            "identitystore:ListGroupMembershipsForMember",
            "identitystore:ListGroups",
            "identitystore:ListUsers",
            "inspector:Describe*",
            "inspector:Get*",
            "inspector:List*",
            "inspector:Preview*",
            "inspector2:BatchGetAccountStatus",
            "inspector2:BatchGetFreeTrialInfo",
            "inspector2:DescribeOrganizationConfiguration",
            "inspector2:GetConfiguration",
            "inspector2:GetDelegatedAdminAccount",
            "inspector2:GetFindingsReportStatus",
            "inspector2:GetMember",
            "inspector2:ListAccountPermissions",
            "inspector2:ListCoverage",
            "inspector2:ListCoverageStatistics",
            "inspector2:ListDelegatedAdminAccounts",
            "inspector2:ListFilters",
            "inspector2:ListFindingAggregations",
            "inspector2:ListFindings",
            "inspector2:ListTagsForResource",
            "inspector2:ListUsageTotals",
            "iot:Describe*",
            "iot:GetPolicy",
            "iot:GetPolicyVersion",
            "iot:List*",
```

```
            "iotanalytics:ListChannels",
            "iotevents:ListInputs",
            "iotfleetwise:ListModelManifests",
            "iotsitewise:DescribeGatewayCapabilityConfiguration",
            "iotsitewise:ListAssetModels",
            "iotsitewise:ListGateways",
            "iottwinmaker:ListWorkspaces",
            "kafka-cluster:Describe*",
            "kafka:Describe*",
            "kafka:GetBootstrapBrokers",
            "kafka:GetCompatibleKafkaVersions",
            "kafka:List*",
            "kafkaconnect:Describe*",
            "kafkaconnect:List*",
            "kendra:DescribeIndex",
            "kendra:ListDataSources",
            "kendra:ListIndices",
            "kendra:ListTagsForResource",
            "kinesis:DescribeLimits",
            "kinesis:DescribeStream",
            "kinesis:DescribeStreamConsumer",
            "kinesis:DescribeStreamSummary",
            "kinesis:ListShards",
            "kinesis:ListStreamConsumers",
            "kinesis:ListStreams",
            "kinesis:ListTagsForStream",
            "kinesisanalytics:ListApplications",
            "kinesisanalytics:ListTagsForResource",
            "kinesisvideo:DescribeEdgeConfiguration",
            "kinesisvideo:DescribeMappedResourceConfiguration",
            "kinesisvideo:DescribeMediaStorageConfiguration",
            "kinesisvideo:DescribeNotificationConfiguration",
            "kinesisvideo:DescribeSignalingChannel",
            "kinesisvideo:DescribeStream",
            "kinesisvideo:ListSignalingChannels",
            "kinesisvideo:ListStreams",
            "kinesisvideo:ListTagsForResource",
            "kinesisvideo:ListTagsForStream",
            "kms:Describe*",
            "kms:Get*",
```

```
            "kms:List*",
            "lambda:GetAccountSettings",
            "lambda:GetFunctionConfiguration",
            "lambda:GetFunctionEventInvokeConfig",
            "lambda:GetLayerVersionPolicy",
            "lambda:GetPolicy",
            "lambda:List*",
            "lex:DescribeBot",
            "lex:DescribeResourcePolicy",
            "lex:ListBots",
            "license-manager:List*",
            "lightsail:GetBuckets",
            "lightsail:GetContainerServices",
            "lightsail:GetDisks",
            "lightsail:GetDiskSnapshots",
            "lightsail:GetInstances",
            "lightsail:GetLoadBalancers",
            "logs:Describe*",
            "logs:GetLogDelivery",
            "logs:ListLogDeliveries",
            "logs:ListTagsForResource",
            "logs:ListTagsLogGroup",
            "lookoutequipment:ListDatasets",
            "lookoutmetrics:ListAnomalyDetectors",
            "lookoutvision:ListProjects",
            "m2:GetApplication",
            "m2:GetEnvironment",
            "m2:ListApplications",
            "m2:ListEnvironments",
            "m2:ListTagsForResource",
            "machinelearning:DescribeMLModels",
            "macie2:ListFindings",
            "managedblockchain:ListNetworks",
            "mechanicalturk:ListHITs",
            "mediaconnect:Describe*",
            "mediaconnect:List*",
            "medialive:ListChannels",
            "mediapackage-vod:DescribePackagingGroup",
            "mediapackage-vod:ListPackagingGroups",
            "mediapackage:DescribeOriginEndpoint",
```

```
            "mediapackage:ListOriginEndpoints",
            "mediastore:GetContainerPolicy",
            "mediastore:GetCorsPolicy",
            "mediastore:ListContainers",
            "memorydb:DescribeClusters",
            "mq:DescribeBroker",
            "mq:DescribeBrokerEngineTypes",
            "mq:DescribeBrokerInstanceOptions",
            "mq:DescribeConfiguration",
            "mq:DescribeConfigurationRevision",
            "mq:DescribeUser",
            "mq:ListBrokers",
            "mq:ListConfigurationRevisions",
            "mq:ListConfigurations",
            "mq:ListTags",
            "mq:ListUsers",
            "network-firewall:DescribeFirewall",
            "network-firewall:DescribeFirewallPolicy",
            "network-firewall:DescribeLoggingConfiguration",
            "network-firewall:DescribeResourcePolicy",
            "network-firewall:DescribeRuleGroup",
            "network-firewall:ListFirewallPolicies",
            "network-firewall:ListFirewalls",
            "network-firewall:ListRuleGroups",
            "networkmanager:DescribeGlobalNetworks",
            "nimble:ListStudios",
            "opsworks-cm:DescribeServers",
            "opsworks:DescribeStacks",
            "organizations:Describe*",
            "organizations:List*",
            "pcs:GetCluster",
            "pcs:GetComputeNodeGroup",
            "pcs:GetQueue",
            "pcs:ListClusters",
            "pcs:ListComputeNodeGroups",
            "pcs:ListQueues",
            "pcs:ListTagsForResource",
            "personalize:DescribeDatasetGroup",
            "personalize:ListDatasetGroups",
            "private-networks:ListNetworks",
```

```
            "profile:GetDomain",
            "profile:ListDomains",
            "profile:ListIntegrations",
            "qbusiness:ListApplications",
            "qbusiness:ListDataSources",
            "qbusiness:ListDataSourceSyncJobs",
            "qbusiness:ListDocuments",
            "qbusiness:ListGroups",
            "qbusiness:ListIndices",
            "qbusiness:ListPlugins",
            "qbusiness:ListRetrievers",
            "qbusiness:ListSubscriptions",
            "qbusiness:ListTagsForResource",
            "qbusiness:ListWebExperiences",
            "qldb:DescribeJournalS3Export",
            "qldb:DescribeLedger",
            "qldb:ListJournalS3Exports",
            "qldb:ListJournalS3ExportsForLedger",
            "qldb:ListLedgers",
            "quicksight:Describe*",
            "quicksight:List*",
            "ram:GetResourceShares",
            "ram:List*",
            "rds:Describe*",
            "rds:DownloadDBLogFilePortion",
            "rds:ListTagsForResource",
            "redshift-serverless:GetNamespace",
            "redshift-serverless:ListTagsForResource",
            "redshift-serverless:ListWorkgroups",
            "redshift:Describe*",
            "rekognition:Describe*",
            "rekognition:List*",
            "resource-groups:ListGroupResources",
            "robomaker:Describe*",
            "robomaker:List*",
            "route53:Get*",
            "route53:List*",
            "route53domains:GetDomainDetail",
            "route53domains:GetOperationDetail",
            "route53domains:ListDomains",
```

```
"route53domains:ListOperations",
"route53domains:ListTagsForDomain",
"route53resolver:Get*",
"route53resolver:List*",
"s3-outposts:ListEndpoints",
"s3-outposts:ListOutpostsWithS3",
"s3-outposts:ListSharedEndpoints",
"s3:DescribeJob",
"s3:GetAccelerateConfiguration",
"s3:GetAccessGrantsInstanceResourcePolicy",
"s3:GetAccessPoint",
"s3:GetAccessPointConfigurationForObjectLambda",
"s3:GetAccessPointPolicy",
"s3:GetAccessPointPolicyForObjectLambda",
"s3:GetAccessPointPolicyStatus",
"s3:GetAccessPointPolicyStatusForObjectLambda",
"s3:GetAccessPointForObjectLambda",
"s3:GetAccountPublicAccessBlock",
"s3:GetAnalyticsConfiguration",
"s3:GetBucket*",
"s3:GetEncryptionConfiguration",
"s3:GetInventoryConfiguration",
"s3:GetLifecycleConfiguration",
"s3:GetMetricsConfiguration",
"s3:GetMultiRegionAccessPoint",
"s3:GetMultiRegionAccessPointPolicy",
"s3:GetMultiRegionAccessPointPolicyStatus",
"s3:GetObjectAcl",
"s3:GetObjectVersionAcl",
"s3:GetReplicationConfiguration",
"s3:GetStorageLensGroup",
"s3:GetStorageLensConfiguration",
"s3:ListAccessPoints",
"s3:ListAllMyBuckets",
"s3:ListMultiRegionAccessPoints",
"s3:ListAccessPointsForObjectLambda",
"s3:ListStorageLensGroups",
"s3:ListJobs",
"s3:ListCallerAccessGrants",
"s3:ListAccessGrantsInstances",
```

```
            "s3:ListAccessGrants",
            "s3:ListStorageLensConfigurations",
            "s3express:ListAllMyDirectoryBuckets",
            "s3express:GetEncryptionConfiguration",
            "s3express:GetBucketPolicy",
            "s3-object-lambda:GetObjectAcl",
            "s3-object-lambda:GetObjectVersionAcl",
            "s3tables:ListTableBuckets",
            "s3tables:ListNamespaces",
            "s3tables:ListTables",
            "s3tables:GetNamespace",
            "s3tables:GetTableBucketPolicy",
            "s3tables:GetTableBucketMaintenanceConfiguration",
            "s3tables:GetTableMaintenanceConfiguration",
            "s3tables:GetTablePolicy",
            "sagemaker:Describe*",
            "sagemaker:List*",
            "schemas:DescribeCodeBinding",
            "schemas:DescribeDiscoverer",
            "schemas:DescribeRegistry",
            "schemas:DescribeSchema",
            "schemas:GetResourcePolicy",
            "schemas:ListDiscoverers",
            "schemas:ListRegistries",
            "schemas:ListSchemas",
            "schemas:ListSchemaVersions",
            "schemas:ListTagsForResource",
            "sdb:DomainMetadata",
            "sdb:ListDomains",
            "secretsmanager:DescribeSecret",
            "secretsmanager:GetResourcePolicy",
            "secretsmanager:ListSecrets",
            "secretsmanager:ListSecretVersionIds",
            "securityhub:BatchGetAutomationRules",
            "securityhub:BatchGetConfigurationPolicyAssociations",
            "securityhub:BatchGetControlEvaluations",
            "securityhub:BatchGetSecurityControls",
            "securityhub:BatchGetStandardsControlAssociations",
            "securityhub:Describe*",
            "securityhub:Get*",
```

```
        "securityhub:List*",
        "serverlessrepo:GetApplicationPolicy",
        "serverlessrepo:List*",
        "servicequotas:GetAssociationForServiceQuotaTemplate",
        "servicequotas:GetAWSDefaultServiceQuota",
        "servicequotas:GetRequestedServiceQuotaChange",
        "servicequotas:GetServiceQuota",

"servicequotas:GetServiceQuotaIncreaseRequestFromTemplate",
        "servicequotas:ListAWSDefaultServiceQuotas",
        "servicequotas:ListRequestedServiceQuotaChangeHistory",

"servicequotas:ListRequestedServiceQuotaChangeHistoryByQuota",

"servicequotas:ListServiceQuotaIncreaseRequestsInTemplate",
        "servicequotas:ListServiceQuotas",
        "servicequotas:ListServices",
        "servicequotas:ListTagsForResource",
        "ses:Describe*",
        "ses:GetAccount",
        "ses:GetAccountSendingEnabled",
        "ses:GetConfigurationSet",
        "ses:GetConfigurationSetEventDestinations",
        "ses:GetDedicatedIps",
        "ses:GetEmailIdentity",
        "ses:GetIdentityDkimAttributes",
        "ses:GetIdentityPolicies",
        "ses:GetIdentityVerificationAttributes",
        "ses:ListConfigurationSets",
        "ses:ListDedicatedIpPools",
        "ses:ListIdentities",
        "ses:ListIdentityPolicies",
        "ses:ListReceiptFilters",
        "ses:ListReceiptRuleSets",
        "ses:ListVerifiedEmailAddresses",
        "shield:Describe*",
        "shield:GetSubscriptionState",
        "shield:List*",
        "snowball:ListClusters",
        "snowball:ListJobs",
```

```
            "sns:GetPlatformApplicationAttributes",
            "sns:GetTopicAttributes",
            "sns:ListSubscriptions",
            "sns:ListSubscriptionsByTopic",
            "sns:ListTagsForResource",
            "sns:ListTopics",
            "sqs:GetQueueAttributes",
            "sqs:ListDeadLetterSourceQueues",
            "sqs:ListQueues",
            "sqs:ListQueueTags",
            "ssm:Describe*",
            "ssm:GetAutomationExecution",
            "ssm:GetServiceSetting",
            "ssm:ListAssociations",
            "ssm:ListAssociationVersions",
            "ssm:ListCommands",
            "ssm:ListComplianceItems",
            "ssm:ListComplianceSummaries",
            "ssm:ListDocumentMetadataHistory",
            "ssm:ListDocuments",
            "ssm:ListDocumentVersions",
            "ssm:ListInventoryEntries",
            "ssm:ListOpsMetadata",
            "ssm:ListResourceComplianceSummaries",
            "ssm:ListResourceDataSync",
            "ssm:ListTagsForResource",
            "sso:DescribeAccountAssignmentCreationStatus",
            "sso:DescribePermissionSet",
            "sso:DescribePermissionsPolicies",
            "sso:DescribeAccountAssignmentDeletionStatus",
            "sso:DescribeApplication",
            "sso:DescribeApplicationAssignment",
            "sso:DescribeApplicationProvider",
            "sso:DescribeInstance",

        "sso:DescribeInstanceAccessControlAttributeConfiguration",
            "sso:DescribePermissionSetProvisioningStatus",
            "sso:DescribeTrustedTokenIssuer",
            "sso:GetApplicationAccessScope",
            "sso:GetApplicationAssignmentConfiguration",
```

```
            "sso:GetApplicationAuthenticationMethod",
            "sso:GetApplicationGrant",
            "sso:GetInlinePolicyForPermissionSet",
            "sso:GetPermissionsBoundaryForPermissionSet",
            "sso:ListAccountAssignmentCreationStatus",
            "sso:ListAccountAssignmentDeletionStatus",
            "sso:ListAccountAssignments",
            "sso:ListAccountAssignmentsForPrincipal",
            "sso:ListAccountsForProvisionedPermissionSet",
            "sso:ListApplicationAccessScopes",
            "sso:ListApplicationAssignments",
            "sso:ListApplicationAssignmentsForPrincipal",
            "sso:ListApplicationAuthenticationMethods",
            "sso:ListApplicationGrants",
            "sso:ListApplicationInstanceCertificates",
            "sso:ListApplicationInstances",
            "sso:ListApplicationProviders",
            "sso:ListApplications",
            "sso:ListApplicationTemplates",
            "sso:ListCustomerManagedPolicyReferencesInPermissionSet",
            "sso:ListDirectoryAssociations",
            "sso:ListInstances",
            "sso:ListManagedPoliciesInPermissionSet",
            "sso:ListPermissionSetProvisioningStatus",
            "sso:ListPermissionSets",
            "sso:ListPermissionSetsProvisionedToAccount",
            "sso:ListProfileAssociations",
            "sso:ListProfiles",
            "sso:ListTagsForResource",
            "sso:ListTrustedTokenIssuers",
            "states:DescribeStateMachine",
            "states:ListStateMachines",
            "storagegateway:DescribeBandwidthRateLimit",
            "storagegateway:DescribeCache",
            "storagegateway:DescribeCachediSCSIVolumes",
            "storagegateway:DescribeGatewayInformation",
            "storagegateway:DescribeMaintenanceStartTime",
            "storagegateway:DescribeNFSFileShares",
            "storagegateway:DescribeSnapshotSchedule",
            "storagegateway:DescribeStorediSCSIVolumes",
```

```
            "storagegateway:DescribeTapeArchives",
            "storagegateway:DescribeTapeRecoveryPoints",
            "storagegateway:DescribeTapes",
            "storagegateway:DescribeUploadBuffer",
            "storagegateway:DescribeVTLDevices",
            "storagegateway:DescribeWorkingStorage",
            "storagegateway:List*",
            "sts:GetAccessKeyInfo",
            "support:DescribeTrustedAdvisorCheckRefreshStatuses",
            "support:DescribeTrustedAdvisorCheckResult",
            "support:DescribeTrustedAdvisorChecks",
            "support:DescribeTrustedAdvisorCheckSummaries",
            "synthetics:DescribeCanaries",
            "synthetics:DescribeCanariesLastRun",
            "synthetics:DescribeRuntimeVersions",
            "synthetics:GetCanary",
            "synthetics:GetCanaryRuns",
            "synthetics:GetGroup",
            "synthetics:ListAssociatedGroups",
            "synthetics:ListGroupResources",
            "synthetics:ListGroups",
            "synthetics:ListTagsForResource",
            "tag:GetResources",
            "tag:GetTagKeys",
            "transcribe:GetCallAnalyticsCategory",
            "transcribe:GetMedicalVocabulary",
            "transcribe:GetVocabulary",
            "transcribe:GetVocabularyFilter",
            "transcribe:ListCallAnalyticsCategories",
            "transcribe:ListCallAnalyticsJobs",
            "transcribe:ListLanguageModels",
            "transcribe:ListMedicalTranscriptionJobs",
            "transcribe:ListMedicalVocabularies",
            "transcribe:ListTagsForResource",
            "transcribe:ListTranscriptionJobs",
            "transcribe:ListVocabularies",
            "transcribe:ListVocabularyFilters",
            "transfer:Describe*",
            "transfer:List*",
            "translate:List*",
```

```
        "trustedadvisor:Describe*",
        "voiceid:DescribeDomain",
        "waf-regional:GetWebACL",
        "waf-regional:ListResourcesForWebACL",
        "waf-regional:ListTagsForResource",
        "waf-regional:ListWebACLs",
        "waf:GetWebACL",
        "waf:ListTagsForResource",
        "waf:ListWebACLs",
        "wafv2:GetLoggingConfiguration",
        "wafv2:GetWebACL",
        "wafv2:GetWebACLForResource",
        "wafv2:ListAvailableManagedRuleGroups",
        "wafv2:ListIPSets",
        "wafv2:ListLoggingConfigurations",
        "wafv2:ListRegexPatternSets",
        "wafv2:ListResourcesForWebACL",
        "wafv2:ListRuleGroups",
        "wafv2:ListTagsForResource",
        "wafv2:ListWebACLs",
        "wisdom:GetAssistant",
        "workdocs:DescribeResourcePermissions",
        "workspaces:Describe*",
        "xray:GetEncryptionConfig",
        "xray:GetGroup",
        "xray:GetGroups",
        "xray:GetSamplingRules",
        "xray:GetSamplingTargets",
        "xray:GetTraceSummaries",
        "xray:ListTagsForResource"
      ],
      "Resource" : "*"
    },
    {
      "Sid" : "APIGatewayAccess",
      "Effect" : "Allow",
      "Action" : [
        "apigateway:GET"
      ],
      "Resource" : [
```

```
        "arn:aws:apigateway:*::/apis",
        "arn:aws:apigateway:*::/apis/*/authorizers/*",
        "arn:aws:apigateway:*::/apis/*/authorizers",
        "arn:aws:apigateway:*::/apis/*/cors",
        "arn:aws:apigateway:*::/apis/*/deployments/*",
        "arn:aws:apigateway:*::/apis/*/deployments",
        "arn:aws:apigateway:*::/apis/*/exports/*",
        "arn:aws:apigateway:*::/apis/*/integrations/*",
        "arn:aws:apigateway:*::/apis/*/integrations",
        "arn:aws:apigateway:*::/apis/*/models/*",
        "arn:aws:apigateway:*::/apis/*/models",
        "arn:aws:apigateway:*::/apis/*/routes/*",
        "arn:aws:apigateway:*::/apis/*/routes",
        "arn:aws:apigateway:*::/apis/*/stages",
        "arn:aws:apigateway:*::/apis/*/stages/*",
        "arn:aws:apigateway:*::/clientcertificates",
        "arn:aws:apigateway:*::/clientcertificates/*",
        "arn:aws:apigateway:*::/domainnames",
        "arn:aws:apigateway:*::/domainnames/*/apimappings",
        "arn:aws:apigateway:*::/restapis",
        "arn:aws:apigateway:*::/restapis/*/authorizers/*",
        "arn:aws:apigateway:*::/restapis/*/authorizers",
        "arn:aws:apigateway:*::/restapis/*/deployments/*",
        "arn:aws:apigateway:*::/restapis/*/deployments",

"arn:aws:apigateway:*::/restapis/*/documentation/parts/*",
        "arn:aws:apigateway:*::/restapis/*/documentation/parts",

"arn:aws:apigateway:*::/restapis/*/documentation/versions/*",

"arn:aws:apigateway:*::/restapis/*/documentation/versions",
        "arn:aws:apigateway:*::/restapis/*/gatewayresponses/*",
        "arn:aws:apigateway:*::/restapis/*/gatewayresponses",
        "arn:aws:apigateway:*::/restapis/*/models/*",
        "arn:aws:apigateway:*::/restapis/*/models",
        "arn:aws:apigateway:*::/restapis/*/requestvalidators",
        "arn:aws:apigateway:*::/restapis/*/requestvalidators/*",
        "arn:aws:apigateway:*::/restapis/*/resources/*",
        "arn:aws:apigateway:*::/restapis/*/resources",
        "arn:aws:apigateway:*::/restapis/*/stages",
```

```
        "arn:aws:apigateway:*::/restapis/*/stages/*",
        "arn:aws:apigateway:*::/tags/*",
        "arn:aws:apigateway:*::/vpclinks"
      ]
    }
  ]
}
```