



User Guide

AWS Continuum



AWS Continuum: User Guide

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is AWS Continuum?	1
Features of AWS Continuum	1
Code vulnerabilities	2
Getting started	2
Connecting a code source	2
Getting started through GitHub	3
Getting started through Amazon S3	4
Creating an environment	9
Following the first scan	10
Adding context	11
Adding infrastructure context	11
Authorizing collector data	14
Adding continuous integration context	20
Connect an EKS cluster	21
Understanding how AWS Continuum accesses your cluster	21
Gathering prerequisites	22
Creating the reader IAM role in the AWS Continuum console	22
Enabling access entries in Amazon EKS	23
Mapping the role into your cluster	23
Granting read-only RBAC	24
Verifying access to your cluster	26
Resolving role and RBAC errors for EKS cluster onboarding	26
Understanding what AWS Continuum can and cannot see	27
Triggering a scan	27
Viewing issues	28
Viewing patches	29
Viewing and downloading a patch	29
Checking patch delivery	30
Security	31
Data protection	31
Opting out of data usage	33
Identity and access management	34
Audience	34
Authenticating with identities	35

Managing access using policies	36
How AWS Continuum works with IAM	38
Identity-based policy examples	42
AWS managed policies	51
Using service-linked roles	55
Troubleshooting	64
Compliance validation	69
Resilience	69
Quotas	70
Document history	72

What is AWS Continuum?

AWS Continuum helps enterprises defend against security exposures at machine speed. AWS Continuum continuously prioritizes security exposures, correlating vulnerabilities, threats, and exploit activity. It automatically remediates high-priority exposures based on business risk.

Frontier AI models have given attackers new capabilities, such as generating complex exploits from vulnerability disclosures in minutes rather than hours. Security teams already cannot keep up with the volume of findings across their environments, and that volume is now accelerating. Prioritizing which findings matter requires understanding which systems are business-critical, how identities connect to sensitive data, and what an attacker can realistically reach.

With AWS Continuum, you can discover, prioritize, validate, and remediate vulnerabilities at machine speed. AWS Continuum for code vulnerabilities focuses on finding, validating, and remediating first-party code vulnerabilities. It integrates with GitHub and Amazon S3 for code and context.

Topics

- [Features of AWS Continuum](#)

Features of AWS Continuum

AWS Continuum provides the following features.

Code vulnerabilities

AWS Continuum finds, validates, and remediates first-party code vulnerabilities across your repositories and uploaded code. For more information, see [Code vulnerabilities](#).

Penetration testing

AWS Continuum continuously validates exploitability against your environment. For more information, see [Perform a penetration test](#) in the *AWS Security Agent User Guide*.

Threat modeling

AWS Continuum models threats to identify security risks in your architecture. For more information, see [Perform a threat model](#) in the *AWS Security Agent User Guide*.

Code vulnerabilities

Continuum for code vulnerabilities is a feature of AWS Continuum that discovers, validates, prioritizes, and recommends remediation for code vulnerabilities.

Continuum for code vulnerabilities integrates with GitHub and Amazon S3 for code and context. You connect a code source, add context about your infrastructure and business, and then review the issues and patches that AWS Continuum produces.

Topics

- [Getting started with code vulnerabilities](#)
- [Adding AWS context](#)
- [Connecting an Amazon EKS cluster to AWS Continuum](#)
- [Triggering a scan](#)
- [Viewing discovered issues](#)
- [Viewing generated patches](#)

Getting started with code vulnerabilities

AWS Continuum scans your code sources for security vulnerabilities.

To start using AWS Continuum for code vulnerabilities, use a GitHub or Amazon S3 integration. If you don't have an integration, create one. Then create an environment that uses the integration.

By default, you can add up to 10 code repositories to an environment across all of its integrations. Integrations and environments are subject to additional quotas. For more information about quotas and requesting an increase, see [Quotas for AWS Continuum](#).

Connecting a code source

If you already have an integration, continue to [Creating an environment](#).

Otherwise, choose the procedure for your code source:

- For GitHub, follow [Getting started through GitHub](#).

- For Amazon S3, follow [Getting started through Amazon S3](#).

Getting started through GitHub

To scan code hosted in GitHub, connect a GitHub integration. Then create an environment that uses the integration.

Use a GitHub account that can install the GitHub app, or request approval from an organization owner.

To configure a GitHub integration, do the following:

1. Open the AWS Continuum console at <https://console.aws.amazon.com/continuum/>.
2. Under **Code vulnerabilities**, choose **Integrations**.
3. Choose **Add integration**.
4. For **Source provider**, select **GitHub**.
5. For **Name**, enter an integration name.
6. For **GitHub account**, select **Personal account** or **Organization**.
7. If you selected **Organization**, enter the GitHub organization name for **Organization name**.

To install the GitHub app and authorize the integration, continue in the same form:

1. If the GitHub app is not installed, choose **Install GitHub app**. Install the app in the account that you selected.
2. If GitHub requires organization approval, wait for an owner to approve the installation.
3. If you opened the GitHub installation tab, close it and return to the AWS Continuum console.
4. Choose **Authorize with GitHub and create**.
5. Complete authorization in the GitHub window.
6. Return to the AWS Continuum console.

If the console still shows **Waiting for GitHub** after you authorize, choose **Cancel authorization**. Start authorization again.

After you create the integration, the console opens **Create environment**. Continue to [Creating an environment](#).

Getting started through Amazon S3

To scan code stored in Amazon S3, prepare a bucket with your code. Create an AWS Identity and Access Management (IAM) role that grants AWS Continuum access to the bucket. Then connect an Amazon S3 integration and create an environment that uses it.

In the following example, replace *amzn-s3-demo-bucket* with your bucket name. Replace *my-app.zip* with your package file name.

The example source URI is `s3://amzn-s3-demo-bucket/code/my-app.zip`. Its object key is `code/my-app.zip`. The corresponding **Source code base path** value is `code/`.

To prepare an Amazon S3 source, do the following:

1. Create an Amazon S3 bucket in the same account and AWS Region where you enable AWS Continuum.
2. Choose a prefix in the bucket for your source code, such as `code/`. You configure this prefix when creating the environment.
3. Upload your code under the chosen prefix as a .zip package. For instructions on uploading objects, see [Uploading objects](#) in the *Amazon S3 User Guide*.
4. In the Amazon S3 console, on **Upload: status**, confirm that the upload succeeded. If the upload fails, resolve the error before continuing.

Granting access to the bucket

As an administrator, create an IAM role in the same account as the bucket. The role path must be `/continuum/integrations/`. The role needs a permissions policy for the bucket and a trust policy so that AWS Continuum can assume the role.

In addition to the read permissions for the console, the IAM principal that creates the integration must have the following permissions:

- `continuum:CreateIntegration` to create the integration
- `iam:ListRoles` to select the integration role
- `s3:ListAllMyBuckets` to browse Amazon S3 buckets
- `iam:PassRole` to pass the integration role

Restrict `iam:PassRole` to the integration role. Use the `iam:PassedToService` condition key with the value `continuum.amazonaws.com`.

For information about permissions required to create an integration in the console, see [Using the AWS Continuum console](#).

Save the following JSON permissions policy as `integration-permissions.json`. Use it to let AWS Continuum list the bucket and read source objects and versions under your code and context prefixes. It also grants AWS Continuum permission to upload and read patches under `continuum-patches/`.

Grant `s3:ListBucket` without a condition. AWS Continuum uses this permission to validate access to the bucket when you create the integration.

Use each configured prefix exactly, including any trailing slash. If you omit these values, the defaults are `code/` and `context/`, respectively. Your source files can be stored under other prefixes, but the policy must grant access to those prefixes.

Replace *bucket-name* with the bucket used by your integration. Replace *code-prefix* with the **Source code base path** value. Replace *context-prefix* with the **Knowledge documents base path** value.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListSourceBucket",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::bucket-name"
    },
    {
      "Sid": "ReadSourcePrefixes",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name/code-prefix*",
```

```

        "arn:aws:s3:::bucket-name/context-prefix*"
    ],
},
{
    "Sid": "DeliverContinuumPatches",
    "Effect": "Allow",
    "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Resource": "arn:aws:s3:::bucket-name/continuum-patches/*"
}
]
}

```

Use the following trust policy so that the AWS Continuum service principal, `continuum.amazonaws.com`, can assume the role with an external ID. Replace *external-id* with the external ID that you enter when creating the integration. Save the policy as `trust-policy.json`.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ContinuumAccess",
            "Effect": "Allow",
            "Principal": {
                "Service": "continuum.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "external-id"
                }
            }
        }
    ]
}

```

To create the role, configure the AWS CLI with a customer administrator principal that can create the role and attach an inline policy. Save both policy files in the current directory.

If a command fails, resolve the error before continuing. If `create-role` returns `EntityAlreadyExists`, choose an unused role name and use it consistently in these commands.

To create the integration role (AWS CLI)

1. Create the role with the required path.

```
aws iam create-role \
  --role-name ContinuumIntegrationRole \
  --path /continuum/integrations/ \
  --assume-role-policy-document file://trust-policy.json
```

The command returns details of the new role.

2. Attach the bucket permissions as an inline policy.

```
aws iam put-role-policy \
  --role-name ContinuumIntegrationRole \
  --policy-name ContinuumIntegrationAccess \
  --policy-document file://integration-permissions.json
```

This command produces no output on success.

3. Verify that the role's Amazon Resource Name (ARN) includes `role/continuum/integrations/`.

```
aws iam get-role \
  --role-name ContinuumIntegrationRole \
  --query 'Role.Arn' \
  --output text
```

The output has the following format, where *account-id* is your AWS account ID and *role-name* is the role name that you used in the commands:

```
arn:aws:iam::account-id:role/continuum/integrations/role-name
```

If a command returns an access denied error, ask your administrator to check your permissions. These commands require `iam:CreateRole`, `iam:PutRolePolicy`, and `iam:GetRole`, respectively.

For information about diagnosing policies that deny access, see [Troubleshoot access denied error messages](#) in the *IAM User Guide*.

Changes to IAM roles and policies can take time to become available. Before you use the role to create the integration, allow time for the changes to propagate. For more information, see [Changes that I make are not always immediately visible](#) in the *IAM User Guide*.

Keep the role and its policies while an integration uses them. When you no longer need an integration, first update or delete every environment that references it, and then delete the integration. After deleting every integration that uses the role, delete the role and its policies if no other workload uses them. For information about deleting roles, see [Delete roles or instance profiles](#) in the *IAM User Guide*.

For more information about external IDs, see [Access to AWS accounts owned by third parties](#) in the *IAM User Guide*.

Connecting an S3 integration

After you prepare the bucket and IAM role, create an Amazon S3 integration. The integration connects the bucket and role so that an environment can access your code.

1. Open the AWS Continuum console at <https://console.aws.amazon.com/continuum/>.
2. Under **Code vulnerabilities**, choose **Integrations**.
3. Choose **Add integration**.
4. For **Source provider**, select Amazon S3.
5. For **Name**, enter an integration name.
6. For **S3 bucket**, use one of the following options:
 - To select a bucket, choose **Browse S3**. Select your bucket.
 - To enter an ARN, choose **Enter bucket ARN manually**. For **Bucket ARN**, enter the bucket ARN.

To grant access and create the integration, continue in the same form:

1. For **IAM role ARN**, select the role that you created.
2. For **External ID**, enter the value required by the role's `sts:ExternalId` condition. The integration value must match the trust policy value.

3. Choose **Create integration**.

After you create the integration, the console opens **Create environment**. Continue to [Creating an environment](#).

If access validation fails, check the required role path. Verify the integration role trust policy and matching external ID. Check that the role's permissions allow access to the bucket and prefixes that you use.

Creating an environment

An environment defines the code sources, scan frequency, and patch delivery settings that AWS Continuum uses to scan your code.

After you create a GitHub or Amazon S3 integration, the console opens **Create environment**.

If you already have an integration, use these steps to open **Create environment**:

1. Open the AWS Continuum console at <https://console.aws.amazon.com/continuum/>.
2. In the navigation pane, choose **Environments**.
3. Choose **Create environment**.

Keep **Automatically deliver remediation patches** selected to deliver patches to your code source. AWS Continuum uploads patches to your Amazon S3 bucket or opens pull requests in your GitHub repositories.

To configure the environment, continue in the same form:

1. For **Environment name**, enter a name.
2. In **Code sources**, for **Integration**, select the integration that you want to use.
3. Configure the code source:
 - a. If you selected a GitHub integration, for **Repositories**, enter the repositories that you want to scan. Enter each repository as owner/repository.
 - b. If you selected an Amazon S3 integration, for **Source code base path**, enter the prefix where you uploaded your code. The default is code/.
 - c. If you selected an Amazon S3 integration and uploaded custom context, enter its prefix in **Knowledge documents base path**. The default is context/.

To set the scan frequency and create the environment, continue in the same form:

1. Expand **Scan schedule**.
2. In **Scan schedule**, for **Scan frequency**, select one of the following options:
 - **Every day**
 - **Every 2 days**
 - **Every 7 days**
 - **Manual only**
3. (Optional) To change patch delivery or allowed dependency domains, expand **Advanced**.
4. (Optional) Under **Advanced**, select or clear **Automatically deliver remediation patches**.
5. (Optional) Under **Advanced**, for **Allowed dependency domains**, enter additional domains needed to build your code.
6. Choose **Create environment**.

Following the first scan

After you create the environment, the console opens **Environments**. Creating the environment starts the first build and scan. The selected scan frequency applies to subsequent scans.

To follow the first scan, do the following:

1. On the **Environments** page, find the **Created** *environment-name* notification. Choose **View environment details**.
2. Choose the **Scans** tab.
3. On the **Scans** tab, choose the refresh icon (**Refresh scans**) to update the results.
4. Review the **Status** column for the first scan. Refresh the results to check for updates.

When a scan completes successfully, its **Status** changes to **Succeeded**. AWS Continuum may continue processing findings into issues and generating patches after the scan completes.

To follow this processing, choose the **Findings** or **Issues** tab and refresh the results. Open an issue to review its processing status and any available patches.

To stop automatic scans, set **Scan frequency** to **Manual only**. For instructions on changing the scan frequency, see [the section called "Triggering a scan"](#).

Adding AWS context

AWS Continuum uses context about your code, infrastructure, and application business to calibrate the severity of the vulnerabilities that it detects. AWS Continuum uses this context to answer questions such as the following:

- Is this vulnerability reachable over the network?
- Is this vulnerability deployed?
- Where is this code running?
- What could an attacker reach if this node was compromised?
- Is a credential escalation possible?

Note

AWS Continuum determines network reachability by reasoning over your AWS configuration constructs, such as security groups, routing, and load balancer and API configuration. It does not perform a formal network reachability proof, so treat reachability as a prioritization signal rather than a guarantee that a path does or does not exist.

Adding infrastructure context

To collect AWS context, you create an AWS Continuum Collector in the account and AWS Region where your code runs. The collector gathers context such as AWS Config resources, AWS Lambda APIs, and Amazon EKS context. You then authorize an AWS Continuum environment to use that inventory.

To integrate AWS infrastructure context, do the following.

1. Open the AWS Continuum console at <https://console.aws.amazon.com/continuum/> and sign in to the AWS account that contains the infrastructure that you want AWS Continuum to collect context from.
2. In the AWS Region where the code runs, create an AWS Continuum Collector.
 - a. In the navigation pane, choose **Collectors**.
 - b. Select the AWS collector.

- c. Choose **Enable**.
 - d. In the confirmation dialog box, choose **Enable**.
 - e. Wait for the collector status to change to **Active**.
3. Sign in to the account that owns the environment. Create or select an IAM role that AWS Continuum can assume to authorize access to collector data. For more information, see [Authorizing an environment to use collector data](#).
4. Sign in to the account that owns the collector and add a resource policy on the collector that grants the environment account access. Skip this step for same-account access. If `AssociateCollector` returns `AccessDeniedException`, check the caller's association and role-passing permissions. Verify that the collector data-access role can access the collector. For instructions on resolving authorization errors, see [Troubleshooting collector authorization](#).
5. Sign in to the account that owns the environment. Associate the collector with the environment and provide the role's Amazon Resource Name (ARN). You must provide the role for both same-account and cross-account collectors. For instructions on authorizing collector data, see [Authorizing an environment to use collector data](#).

The collector gathers the following information from an AWS account:

- AWS Config resources. The collector includes the following resource types:
 - Amazon API Gateway: REST API methods, REST APIs, and stages; and Amazon API Gateway Version 2 (HTTP and WebSocket) APIs, integrations, and stages
 - Amazon Elastic Compute Cloud (Amazon EC2): instances, internet gateways, NAT gateways, network ACLs, network interfaces, route tables, security groups, subnets, transit gateways, transit gateway attachments, VPCs, and VPC endpoints
 - AWS Identity and Access Management (IAM): groups, instance profiles, OpenID Connect (OIDC) identity providers, policies, roles, and users
 - Amazon Elastic Container Service (Amazon ECS): clusters, services, and task definitions
 - Amazon Elastic Kubernetes Service (Amazon EKS): clusters, Fargate profiles, and node groups
 - Amazon Elastic Container Registry (Amazon ECR): repositories and public repositories
 - Elastic Load Balancing: Classic Load Balancers; and Elastic Load Balancing Version 2 listeners, listener rules, Application Load Balancers and Network Load Balancers, and target groups
 - Amazon Relational Database Service (Amazon RDS): DB clusters and DB instances
 - Amazon Redshift: clusters

- Amazon DynamoDB: tables
- Amazon Keyspaces (for Apache Cassandra): keyspaces
- Amazon QLDB: ledgers
- AWS Lambda: functions
- Amazon Simple Storage Service (Amazon S3): buckets
- Amazon Elastic File System (Amazon EFS): file systems
- CloudFormation: stacks
- Amazon CloudFront: distributions
- AWS CodeBuild: projects
- AWS CodePipeline: pipelines
- Amazon Kinesis Data Streams: streams
- Amazon Data Firehose: delivery streams
- Amazon Managed Streaming for Apache Kafka (Amazon MSK): clusters
- Amazon Simple Notification Service (Amazon SNS): topics
- Amazon Simple Queue Service (Amazon SQS): queues
- Amazon OpenSearch Service: domains
- Amazon OpenSearch Service (legacy Elasticsearch): Elasticsearch domains
- Amazon Route 53: hosted zones
- AWS Key Management Service (AWS KMS): KMS keys
- AWS Secrets Manager: secrets
- Amazon GuardDuty: detectors
- AWS WAF: web ACLs (WAFV2)

The collector also gathers the following:

- AWS Lambda function data collected through API polling
- Amazon EKS Kubernetes objects: pods, deployments, nodes, and replicas
- Amazon ECR container images

After you complete these steps, AWS Continuum can use the authorized collector inventory when it generates context for the environment. Repeat these steps for each account from which you want to collect AWS context. For information about collector quotas, see [Quotas](#).

Authorizing an environment to use collector data

An environment can use inventory from a collector in the same AWS account or in another account. For either configuration, the environment owner associates the collector and supplies a customer-managed role that AWS Continuum assumes before it requests access to the inventory. For cross-account access, the collector owner must also add a resource-policy grant.

The following controls work together:

- The association identifies the collector that the environment wants to use.
- The collector data-access role represents the environment when AWS Continuum requests authorization.
- For cross-account access, the collector resource policy represents the collector owner's consent.

The following table lists the responsibilities for each account.

Account	Responsibilities
Collector account	Create the collector. For cross-account access, grant the environment account access in the collector resource policy and manage the policy with <code>GetResourcePolicy</code> and <code>PutResourcePolicy</code> .
Environment account	Create the environment and collector data-access role. Associate the collector with the environment and pass the role to AWS Continuum.

If the collector and environment are in the same account, that account performs both sets of responsibilities. You don't need to add an `Allow` statement to the collector resource policy for same-account access.

The examples use the AWS Continuum public API namespace, `continuum`.

In the following examples, replace *collector-account* with the ID of the account that owns the collector, and *environment-account* with the ID of the account that owns the environment. Replace *environment-id*, *collector-id*, and *region* with your resource values.

1. Open the [IAM console](#) in the account that owns the environment.

Create an IAM role for collector data access. The role trust policy must allow the AWS Continuum service principal, `continuum.amazonaws.com`, to assume it. The following trust policy limits role assumption to the specified environment:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowContinuumRuntime",
      "Effect": "Allow",
      "Principal": {
        "Service": "continuum.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "environment-account"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:continuum:region:environment-account:environments/environment-id"
        }
      }
    }
  ]
}
```

Attach an identity policy that permits the role to request authorization for every collector that can contribute data to the environment:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

        "Sid": "AuthorizeCollectorData",
        "Effect": "Allow",
        "Action": "continuum:GetCollectorDataAccessAuthorization",
        "Resource": "arn:aws:continuum:region:collector-account:collectors/collector-
id"
    }
]
}

```

2. For cross-account access, sign in to the account that owns the collector. Skip this step for same-account access. Add a resource policy that grants the environment account permission to call `continuum:GetCollectorDataAccessAuthorization`.

Use an account principal so the identity policies in the environment account can authorize the collector data-access role that AWS Continuum assumes for the environment.

The IAM principal that manages the resource policy must belong to the collector account and have the following permissions for the collector:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ManageCollectorResourcePolicy",
      "Effect": "Allow",
      "Action": [
        "continuum:GetResourcePolicy",
        "continuum:PutResourcePolicy"
      ],
      "Resource": "arn:aws:continuum:region:collector-account:collectors/collector-
id"
    }
  ]
}

```

Use the following collector resource policy:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEnvironmentAccount",

```

```

    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::environment-account:root"
    },
    "Action": "continuum:GetCollectorDataAccessAuthorization",
    "Resource": "arn:aws:continuum:region:collector-account:collectors/collector-
id"
  }
]
}

```

Call `GetResourcePolicy` with the full collector ARN in the `arn` input:

```

{
  "arn": "arn:aws:continuum:region:collector-account:collectors/collector-id"
}

```

Copy the `revisionId` from the response. Read the policy. Add the `AllowEnvironmentAccount` statement and retain all existing statements. Serialize the complete policy as a JSON string. Call `PutResourcePolicy` with the following input:

```

{
  "arn": "arn:aws:continuum:region:collector-account:collectors/collector-id",
  "policy": "JSON-serialized-policy-document",
  "expectedRevisionId": "resource-policy-revision"
}

```

If the policy changed after you read it, AWS Continuum returns `ConflictException`. Read the latest policy and `revisionId` with `GetResourcePolicy`. Reapply your change to that policy, retaining its current statements. Retry `PutResourcePolicy` with the updated policy and the latest revision in `expectedRevisionId`.

Important

`PutResourcePolicy` replaces the entire policy. Include every statement that you want to keep. If you omit policy, or supply a policy with an empty `Statement` array, AWS Continuum removes all cross-account grants. Same-account authorization remains subject to applicable identity policies. An applicable explicit Deny in the

collector resource policy still overrides an Allow. There is no separate delete operation for a collector resource policy.

3. To associate the collector with the environment, call `AssociateCollector` in the account that owns the environment. The caller must have `continuum:AssociateCollector` permission for the environment and `iam:PassRole` permission for the exact collector data-access role. AWS Continuum assumes that role and checks its access to the collector before creating the association.

The following example grants the caller these permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AssociateCollector",
      "Effect": "Allow",
      "Action": "continuum:AssociateCollector",
      "Resource": "arn:aws:continuum:region:environment-
account:environments/environment-id"
    },
    {
      "Sid": "PassCollectorDataAccessRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::environment-account:role/
ContinuumCollectorDataAccess"
    }
  ]
}
```

Then call `AssociateCollector` with the following input:

```
{
  "environmentId": "environment-id",
  "collectorArn": "arn:aws:continuum:region:collector-account:collectors/collector-
id",
  "collectorDataAccessRoleArn": "arn:aws:iam::environment-account:role/
ContinuumCollectorDataAccess"
}
```

The role must belong to the environment account. This request stores the role on the environment, replacing the previously configured collector data-access role. If the environment has other associated collectors, make sure that the replacement role also permits access to those collectors.

Verify that the response contains the expected `environmentId` and `environmentArn`. Check that the `associatedCollector` object contains the requested `collectorArn` and its `collectorId`.

4. To confirm the stored association, call `ListAssociatedCollectors` in the environment account with the following input. The caller needs `continuum:ListAssociatedCollectors` permission for the environment.

```
{
  "environmentId": "environment-id"
}
```

Verify that `associatedCollectors` contains the expected collector ARN. If the response includes `nextToken`, pass it in the next request to read the remaining results. Continue through empty pages while the response includes a token.

The collector data-access role doesn't need permission to access the collector's inventory storage. This role is separate from a role that AWS Continuum uses to read a customer-managed Amazon S3 source.

AWS Continuum assumes the collector data-access role when it generates context for the environment. AWS Continuum uses the resulting role session to request authorization for each associated collector. AWS Continuum returns an empty response when authorization succeeds. AWS Continuum uses inventory only when authorization succeeds.

For same-account access, configure the association, role trust policy, and role identity permission. For cross-account access, also add the collector resource-policy grant. You can revoke access independently from either account by removing the association, the role permission, or the cross-account resource-policy grant.

Troubleshooting collector authorization

If `AssociateCollector` returns `AccessDeniedException`, verify the following:

- The caller has `continuum:AssociateCollector` permission for the environment and `iam:PassRole` permission for the collector data-access role.
- The collector data-access role permits `continuum:GetCollectorDataAccessAuthorization` for the collector. Its trust policy allows AWS Continuum to assume it for this environment.
- For cross-account access, the collector resource policy grants the environment account `continuum:GetCollectorDataAccessAuthorization` permission for the exact collector ARN.
- The collector and environment are in the same AWS Region, and the collector ARN contains the correct partition, Region, account ID, and `collectors/collector-id` resource path.
- No identity policy, collector resource policy, permissions boundary, or AWS Organizations service control policy (SCP) contains an applicable explicit Deny.

If the association succeeds but the environment doesn't use the collector inventory, verify the following:

- The role trust policy allows `continuum.amazonaws.com` and uses the environment account ID for `aws:SourceAccount` and the exact environment ARN for `aws:SourceArn`.
- The role identity policy allows `continuum:GetCollectorDataAccessAuthorization` for every associated collector that can contribute data to the environment.
- The most recent `AssociateCollector` request supplied the intended role. Supplying a different role for another collector replaces the role stored on the environment.
- For cross-account access, the current collector resource policy still grants the environment account access. Call `GetResourcePolicy` to inspect the stored policy and revision.

You can also review CloudTrail events for the failed operation to confirm the caller and resource ARN that AWS Continuum evaluated.

Adding continuous integration context

When you connect a GitHub integration, AWS Continuum also collects continuous integration (CI) context from the repository. AWS Continuum uses this context to determine how your code is built and deployed, which helps it establish whether a vulnerability reaches a running workload.

AWS Continuum collects the following CI context from a connected GitHub repository:

- Workflows
- Actions
- Environment variables that are defined for the workflows and actions

Connecting an Amazon EKS cluster to AWS Continuum

For read-only Kubernetes inventory collection, connect your Amazon EKS cluster to AWS Continuum. The onboarding takes about 10 minutes for each cluster. This task requires Amazon EKS administrator permissions and `kubectl` cluster-admin access. If you don't have these permissions, ask your administrator to perform these steps or grant you the required access.

This topic covers the following sections.

- [Understanding how AWS Continuum accesses your cluster](#)
- [Gathering prerequisites](#)
- [Creating the reader IAM role in the AWS Continuum console](#)
- [Enabling access entries in Amazon EKS](#)
- [Mapping the role into your cluster](#)
- [Granting read-only RBAC](#)
- [Verifying access to your cluster](#)
- [Resolving role and RBAC errors for EKS cluster onboarding](#)
- [Understanding what AWS Continuum can and cannot see](#)

Understanding how AWS Continuum accesses your cluster

AWS Continuum reads your cluster's Kubernetes state through the Amazon EKS cluster communication service (CCS). There is no public endpoint, no VPC peering, and no agent to install. You control access with three elements:

- A trust-only AWS Identity and Access Management (IAM) role that the AWS Continuum collector can assume. The role grants no AWS permissions of its own.
- An Amazon EKS access entry that maps that role to a Kubernetes group.
- A read-only role-based access control (RBAC) binding that lets the group read resources. The binding never grants write access and never grants access to Secrets.

You can revoke access at any time by deleting the role or the access entry.

With AWS Enhanced Zero-Click Role Creation, you create only the reader IAM role. You set up the Amazon EKS access entry and the in-cluster RBAC yourself, because those use Amazon EKS and Kubernetes APIs instead of IAM.

Gathering prerequisites

Before you begin, make sure that you have the following:

- You have access to the AWS Continuum console Kubernetes collection, which lets you create the reader IAM role, enable the AWS collector, and turn on Kubernetes collection.
- You have Amazon EKS administrator permissions in the account.
- You have `kubectl` cluster-admin access to the cluster, which you use to grant read-only RBAC.
- You have your cluster name and AWS Region.
- If the cluster is private-only (no public endpoint), you have a host with a network path to the cluster endpoint from which to run `kubectl`.

Creating the reader IAM role in the AWS Continuum console

Complete this procedure one time for each account to create the reader IAM role. For more information about IAM roles, see [IAM roles](#) in the *IAM User Guide*.

To create the reader IAM role

1. Open the AWS Continuum console at <https://console.aws.amazon.com/continuum/> and sign in to the AWS account that contains the EKS cluster to collect.
2. Choose **Set up Kubernetes collection**.
3. Turn on **Kubernetes collection**.
4. Choose **Create default role**.
5. Choose **Create role**, and then save. Role creation completes in a few minutes.

If role creation fails, verify that you have the IAM permissions to create roles, and then try again.

The console creates an IAM role named `ContinuumKubernetesReader-collector-id`, which has no permissions policy attached. The procedures later in this topic decide what the role can actually read.

Enabling access entries in Amazon EKS

Complete this procedure one time for each cluster. Your cluster must use the API or API_AND_CONFIG_MAP authentication mode, and the modern default is API_AND_CONFIG_MAP.

The authentication mode changes in one direction only. Run the following command only when the cluster is currently in CONFIG_MAP mode. If the cluster is already in API or API_AND_CONFIG_MAP mode, skip this procedure. For more information about access entries, see [Manage access entries](#) in the *Amazon EKS User Guide*.

The following AWS Command Line Interface (AWS CLI) command updates the cluster configuration. Replace *region* with your AWS Region and *cluster* with your cluster name:

```
PROMPT> aws eks update-cluster-config \
  --region region \
  --name cluster \
  --access-config authenticationMode=API_AND_CONFIG_MAP
```

If the command returns an access denied error, verify that you have Amazon EKS administrator permissions, and then try again.

Mapping the role into your cluster

Complete this procedure one time for each cluster. Before you begin, create the reader IAM role in [Creating the reader IAM role in the AWS Continuum console](#), and make sure that you have Amazon EKS administrator permissions. The command maps the reader IAM role to the Kubernetes group `continuum:readers`. You reference this group when you grant read-only RBAC.

Map the reader IAM role to the Kubernetes group by running the following AWS CLI command. Replace *region*, *cluster*, *account-id*, and *collector-id* with your values:

```
PROMPT> aws eks create-access-entry \
  --region region \
  --cluster-name cluster \
  --principal-arn arn:aws:iam::account-id:role/ContinuumKubernetesReader-collector-id \
  --kubernetes-groups continuum:readers \
  --type STANDARD
```

If the command fails because the role is missing or can't be assumed, see [Resolving role and RBAC errors for EKS cluster onboarding](#).

Granting read-only RBAC

Complete this procedure one time for each cluster. This procedure requires `kubectl` cluster-admin access. If you don't have cluster-admin access, ask your administrator to perform these steps.

With this manifest, you grant read-only access (get, list, and watch) on the resources that AWS Continuum collects, while you exclude Secrets.

Apply the following Kubernetes RBAC YAML manifest with `kubectl`:

```
PROMPT> kubectl apply -f - <<'EOF'
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: continuum-reader
  labels:
    app.kubernetes.io/managed-by: continuum-k8s-collection
rules:
  # Core API group (list-only; secrets deliberately excluded).
  - apiGroups: [""]
    resources:
      - pods
      - services
      - nodes
      - namespaces
      - serviceaccounts
    verbs: ["get", "list", "watch"]

  # Workloads.
  - apiGroups: ["apps"]
    resources: ["deployments", "statefulsets", "daemonsets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["batch"]
    resources: ["jobs"]
    verbs: ["get", "list", "watch"]

  # Core networking and policy.
  - apiGroups: ["networking.k8s.io"]
    resources: ["ingresses", "networkpolicies"]
    verbs: ["get", "list", "watch"]

  # RBAC.
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings", "clusterroles", "clusterrolebindings"]
```

```

  verbs: ["get", "list", "watch"]

# Admission webhooks.
- apiGroups: ["admissionregistration.k8s.io"]
  resources: ["mutatingwebhookconfigurations", "validatingwebhookconfigurations"]
  verbs: ["get", "list", "watch"]

# Kubernetes Gateway API (CRD).
- apiGroups: ["gateway.networking.k8s.io"]
  resources: ["*"]
  verbs: ["get", "list", "watch"]

# CNI network policies (CRD).
- apiGroups: ["cilium.io"]
  resources: ["*"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["crd.projectcalico.org"]
  resources: ["*"]
  verbs: ["get", "list", "watch"]

# Istio service mesh (CRD).
- apiGroups: ["install.istio.io", "networking.istio.io", "security.istio.io"]
  resources: ["*"]
  verbs: ["get", "list", "watch"]

# Kyverno policy engine (CRD).
- apiGroups: ["kyverno.io"]
  resources: ["*"]
  verbs: ["get", "list", "watch"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: continuum-reader-binding
  labels:
    app.kubernetes.io/managed-by: continuum-k8s-collection
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: continuum-reader
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: continuum:readers # MUST match the access entry group.

```

EOF

If some resources return a forbidden error, confirm that the access entry group and the binding both use `continuum:readers`.

For a private-only cluster, run `kubectl` from a host that has a network path to the cluster endpoint, such as an in-VPC host.

Note

You grant read-only RBAC one time. The AWS Continuum collector never uses `kubectl`. The collector reaches your cluster only over CCS for read-only collection.

Verifying access to your cluster

From a cluster-admin context, confirm that the reader group can list nodes but cannot list Secrets. The following `kubectl` commands return yes for nodes and no for Secrets:

```
PROMPT> kubectl auth can-i list nodes --as-group=continuum:readers --as=continuum-check
yes
```

```
PROMPT> kubectl auth can-i list secrets --as-group=continuum:readers --as=continuum-check
no
```

Resolving role and RBAC errors for EKS cluster onboarding

Use the following table to resolve common issues.

Symptom	Likely cause	Fix
AWS Continuum can't assume the role.	The role is missing or its trust policy changed.	In the AWS Continuum console, repeat the procedure in Creating the reader IAM role in the AWS Continuum console to recreate the role and its trust policy correctly.

Symptom	Likely cause	Fix
Some resources are missing or return a forbidden error.	You didn't apply the RBAC, or the group name doesn't match.	Repeat the procedure in Granting read-only RBAC . Make sure that the access entry group and the binding both use <code>continuum:readers</code> .
kubectl can't reach a private cluster.	The cluster has a private-only endpoint.	Run kubectl from a host that has a network path to the cluster endpoint, such as an in-VPC host.

Understanding what AWS Continuum can and cannot see

The following list describes what the AWS Continuum collector can and cannot see:

- The collector reads the following resources: namespaces, nodes, pods, services, workloads (deployments, daemonsets, statefulsets, and jobs), ingresses, network policies, RBAC objects, admission webhooks, and common networking and service mesh CRDs.
- The collector does not read Secrets and performs no write operations.

Triggering a scan

When you create an environment, AWS Continuum starts the first scan, even if you select **Manual only**. Use the scan frequency to control recurring scans. By default, AWS Continuum scans your code every day. If you select **Manual only**, AWS Continuum doesn't run automatic recurring scans. You can also queue a scan manually from the console.

To change the scan frequency, do the following.

1. Open the AWS Continuum console at <https://console.aws.amazon.com/continuum/>.
2. In the navigation pane, choose **Environments**, and then choose the environment name.
3. Choose the **Scans** tab.
4. Choose **Configure scan frequency**.
5. For **Scan frequency**, select one of the following values:
 - **Every day**

- **Every 2 days**
 - **Every 7 days**
 - **Manual only**
6. Choose **Save**.

To queue a manual scan, do the following. You can't cancel a scan after you queue it.

1. Open the AWS Continuum console at <https://console.aws.amazon.com/continuum/>.
2. In the navigation pane, choose **Environments**, and then choose the environment name.
3. Choose the **Scans** tab.
4. Choose **Queue scan**.
5. In the **Queue scan?** dialog, choose **Queue scan**.
6. Choose the **Refresh scans** icon. Check the scan's status in the **Status** column.

If a failure reason is available for a failed scan, choose **Failed** in the **Status** column. The **Failure reason** popover shows the explanation.

After a scan reaches **Succeeded** status, AWS Continuum might continue processing issues and generating patches. To view available results, see [the section called "Viewing issues"](#) and [the section called "Viewing patches"](#).

Viewing discovered issues

You can review issues for an environment on the **Issues** tab in the AWS Continuum console. Use search and the severity filter to find an issue, then choose its title to review the available details.

To find and review an issue

1. Open the AWS Continuum console at <https://console.aws.amazon.com/continuum/>.
2. In the navigation pane, choose **Environments**.
3. Choose the name of your environment.
4. Choose the **Issues** tab to view the list of issues.
5. (Optional) To search for an issue, enter text in **Find issues**.
6. (Optional) For **Severity**, select **Critical**, **High**, **Medium**, or **Low**. Select **All severities** to remove the severity filter.

7. Choose the issue title to open its details.

Search, sorting, and counts include only issues that the console has loaded. When **Load more** appears, choose it to include additional issues. Use the page controls to move through the loaded results.

If the table shows **No matching issues**, choose **Clear filters** to remove your search and severity filter. To retrieve updated results, choose the refresh icon labeled **Refresh issues**.

If no issues appear after you clear filters, a scan or its results might still be processing. Check the **Scans** tab for progress, then refresh the issues list.

The available details depend on the issue's processing results. You might see the following:

- **Issue details**
- **Severity** and **Processing status**
- **Synopsis** and **Risk assessment**
- Validation evidence in **Proof**
- **Remediation summary** and **Patch details**
- **Contributing findings**

When **Proof** appears, review the evidence shown there, such as **Source analysis**. The proof type depends on the issue. A unit test or executable proof might not be available.

For more information about available patches, see [the section called "Viewing patches"](#).

Viewing generated patches

Review generated patches to understand the proposed code changes before you apply them. You can download available patches in the AWS Continuum console and check their delivery to your code sources.

Viewing and downloading a patch

Before you begin, your environment must have an issue with a generated patch. You can review the changes and download a patch file for a code source.

To view and download a patch

1. Open the AWS Continuum console at <https://console.aws.amazon.com/continuum/>.
2. In the navigation pane, choose **Environments**.
3. Choose the name of the environment that contains the issue.
4. Choose the **Issues** tab.
5. Choose the title of the issue that you want to review.
6. Choose **View patch** for your source to open its generated patch in **Patch details**.
7. To download the patch file, choose the source's **Download patch** icon, labeled **Download patch for *source name***.

Your browser saves the downloaded patch as a .patch file.

Checking patch delivery

When you select **Automatically deliver remediation patches** for an environment, AWS Continuum attempts to deliver generated patches to your code sources. For GitHub, AWS Continuum creates pull requests. For Amazon S3, AWS Continuum uploads patch files under `continuum-patches/` in your bucket.

On the issue details page, you can check the delivery destination and status. For a GitHub source, choose **View pull request** when the console displays that link.

In **Patch details**, choose the source's information icon, labeled **Patch information for *source name***. The popover shows **Destination** when available. For Amazon S3, this value identifies the bucket and patch object.

To check delivery status and failure details, choose the issue's **Patch delivery** status. You can download an available patch from the console even if delivery to your code source fails.

For instructions on configuring the Amazon S3 integration role's patch permissions, see [Granting access to the bucket](#). For information about console permissions, see [Using the AWS Continuum console](#).

Security in AWS Continuum

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS Compliance Programs](#). To learn about the compliance programs that apply to AWS Continuum, see [AWS Services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Continuum. The following topics show you how to configure AWS Continuum to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS Continuum resources.

Topics

- [Data protection in AWS Continuum](#)
- [Identity and access management for AWS Continuum](#)
- [Compliance validation for AWS Continuum](#)
- [Resilience in AWS Continuum](#)

Data protection in AWS Continuum

The AWS [shared responsibility model](#) applies to data protection in AWS Continuum. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for

the AWS services that you use. For more information about data privacy, see [Data Privacy FAQ](#). For information about data protection in Europe, see the [General Data Protection Regulation \(GDPR\) Center](#).

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see [Working with CloudTrail trails](#) in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-3](#).

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with AWS Continuum or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

 Note

The data protection information in this topic applies to AWS Continuum. It does not cover Continuum for Penetration testing (formerly known as Security Agent).

Note

AWS Continuum automatically selects the optimal AWS Region within your geographic area to process your inference requests. This approach maximizes available compute resources and model availability. Your data remains stored only in the Region where the request originated. However, AWS Continuum might process input prompts and output results outside that Region. AWS Continuum encrypts all data in transit across the Amazon secure network.

AWS Continuum securely routes inference requests to available compute resources within the geographic area where the request originated, as follows:

- Inference requests that originate in the United States are processed within the United States.

Opting out of using your data for service improvement

You can choose to opt out of having your data (defined as "AWS Continuum Content" in the AWS Continuum service terms) used to develop and improve AWS Continuum and other AWS security services by using the AWS Organizations opt-out policy. You can choose to opt out even if AWS Continuum does not currently collect any such Content. For more information about how to opt out, see [AI services opt-out policies](#) in the *AWS Organizations User Guide*.

Note

To use the opt-out policy, your AWS accounts must be centrally managed by AWS Organizations. If you haven't already created an organization for your AWS accounts, see [Creating and managing an organization](#) in the *AWS Organizations User Guide*.

Opting out has the following effects:

- AWS Continuum deletes the Content that it collected and stored for service-improvement purposes before you opted out, if any.
- After you opt out, AWS Continuum no longer collects or stores this Content for service-improvement purposes.

AWS Continuum data usage

Currently, AWS Continuum does not collect or store any AWS Continuum Content for service-improvement purposes. However, in the future, AWS Continuum might collect and use AWS Continuum Content that it receives from you or from upstream services, for the purpose of providing AWS Continuum capabilities, to improve AWS Continuum and other AWS security service capabilities.

Your trust, privacy, and the security of your Content are our highest priority. If AWS Continuum begins collecting AWS Continuum Content for service-improvement purposes in the future, this page is updated to reflect those changes and to provide details about what data is collected and how it is used. You can opt out of this AWS Continuum Content collection at any time by using the AWS Organizations opt-out policy described previously. For more information about AWS data privacy practices, see the [Data Privacy FAQ](#).

Identity and access management for AWS Continuum

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS Continuum resources. IAM is an AWS service that you can use with no additional charge.

Topics

- [Audience](#)
- [Authenticating with identities](#)
- [Managing access using policies](#)
- [How AWS Continuum works with IAM](#)
- [Identity-based policy examples for AWS Continuum](#)
- [AWS managed policies for AWS Continuum](#)
- [Using service-linked roles for AWS Continuum](#)
- [Troubleshooting AWS Continuum identity and access](#)

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see [Troubleshooting AWS Continuum identity and access](#))
- **Service administrator** - determine user access and submit permission requests (see [How AWS Continuum works with IAM](#))
- **IAM administrator** - write policies to manage access (see [Identity-based policy examples for AWS Continuum](#))

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see [How to sign in to your AWS account](#) in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see [AWS Signature Version 4 for API requests](#) in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see [Tasks that require root user credentials](#) in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see [What is IAM Identity Center?](#) in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An [IAM user](#) is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see [Require human users to use federation with an identity provider to access AWS using temporary credentials](#) in the *IAM User Guide*.

An [IAM group](#) specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see [Use cases for IAM users](#) in the *IAM User Guide*.

IAM roles

An [IAM role](#) is an identity with specific permissions that provides temporary credentials. You can assume a role by [switching from a user to an IAM role \(console\)](#) or by calling an AWS CLI or AWS API operation. For more information, see [Methods to assume a role](#) in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see [Cross account resource access in IAM](#) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see [Overview of JSON policies](#) in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see [Define custom IAM permissions with customer managed policies](#) in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see [Choose between managed policies and inline policies](#) in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must [specify a principal](#) in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see [Permissions boundaries for IAM entities](#) in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see [Service control policies](#) in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see [Resource control policies \(RCPs\)](#) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see [Session policies](#) in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see [Policy evaluation logic](#) in the *IAM User Guide*.

How AWS Continuum works with IAM

Before you use IAM to manage access to AWS Continuum, learn what IAM features are available to use with AWS Continuum.

IAM feature	AWS Continuum support
Identity-based policies	Yes
Resource-based policies	Yes
Policy actions	Yes
Policy resources	Yes
Policy condition keys	Yes
ACLs	No
ABAC (tags in policies)	Partial
Temporary credentials	Yes
Principal permissions	Yes
Service roles	Yes
Service-linked roles	Yes

To get a high-level view of how AWS Continuum and other AWS services work with most IAM features, see [AWS services that work with IAM](#) in the *IAM User Guide*.

Identity-based policies for AWS Continuum

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can

perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see [Define custom IAM permissions with customer managed policies](#) in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see [IAM JSON policy elements reference](#) in the *IAM User Guide*.

Identity-based policy examples for AWS Continuum

To view examples of AWS Continuum identity-based policies, see [Identity-based policy examples for AWS Continuum](#).

Resource-based policies within AWS Continuum

Supports resource-based policies: Yes

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must [specify a principal](#) in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see [Cross account resource access in IAM](#) in the *IAM User Guide*.

AWS Continuum supports resource-based policies for collectors. A collector resource policy controls access to the inventory that the collector gathers. You must use a resource policy to grant access to a principal outside the collector's AWS account. The resource policy does not associate the collector with an environment. Create that association separately and provide a collector data-access role.

You manage a collector resource policy with `GetResourcePolicy` and `PutResourcePolicy`. For instructions on setting up a resource policy and an example policy, see [Authorizing an environment to use collector data](#).

Policy resources for AWS Continuum

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Resource JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its [Amazon Resource Name \(ARN\)](#). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"

```

For information about scoping policies to a collector, an environment, and a data-access role, see [Authorizing an environment to use collector data](#). For information about specifying resources in a policy, see [IAM JSON policy elements: Resource](#) in the *IAM User Guide*.

To view examples of AWS Continuum identity-based policies, see [Identity-based policy examples for AWS Continuum](#).

Policy condition keys for AWS Continuum

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Condition element specifies when statements execute based on defined criteria. You can create conditional expressions that use [condition operators](#), such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see [AWS global condition context keys](#) in the *IAM User Guide*.

For information about conditions in collector authorization policies, see [Authorizing an environment to use collector data](#). For information about conditions in a policy, see [IAM JSON policy elements: Condition](#) in the *IAM User Guide*.

To view examples of AWS Continuum identity-based policies, see [Identity-based policy examples for AWS Continuum](#).

Using temporary credentials with AWS Continuum

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see [Temporary security credentials in IAM](#) and [AWS services that work with IAM](#) in the *IAM User Guide*.

Cross-service principal permissions for AWS Continuum

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see [Forward access sessions](#).

Service roles for AWS Continuum

Supports service roles: Yes

A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Create a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.

Warning

Changing the permissions for a service role might break AWS Continuum functionality. Edit service roles only when AWS Continuum provides guidance to do so.

Service-linked roles for AWS Continuum

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see [AWS services that work with IAM](#).

Identity-based policy examples for AWS Continuum

By default, users and roles don't have permission to create or modify AWS Continuum resources. An IAM administrator can grant these permissions through policies.

The IAM action prefix for AWS Continuum is `continuum`. For instructions on creating IAM policies, see [Creating IAM policies \(console\)](#) in the *IAM User Guide*. For information about collector authorization policies, see [Authorizing an environment to use collector data](#).

Topics

- [Policy best practices](#)
- [Using the AWS Continuum console](#)
- [Allow users to view their own permissions](#)
- [Allow users to view findings and issues](#)
- [Allow users to submit custom findings](#)

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS Continuum resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see [AWS managed policies](#) or [AWS managed policies for job functions](#) in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see [Policies and permissions in IAM](#) in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to

service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see [IAM JSON policy elements: Condition](#) in the *IAM User Guide*.

- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see [Validate policies with IAM Access Analyzer](#) in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see [Secure API access with MFA](#) in the *IAM User Guide*.

For more information about best practices in IAM, see [Security best practices in IAM](#) in the *IAM User Guide*.

Using the AWS Continuum console

To access the AWS Continuum console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS Continuum resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To use the AWS Continuum console, grant the operator the `continuum:` actions for their tasks. A broad operator grant of `continuum:*` includes all AWS Continuum actions. It does not grant IAM, AWS Config, or Amazon S3 permissions.

The following tables list permissions for console tasks. Grant the read actions for the pages and selectors that the operator uses in addition to each action that changes a resource. These action tables are not a complete IAM policy.

Use the following permissions for environment, scan, and results tasks.

Task	Principal	Permissions and role requirements
Open the overview and navigation	Console operator	<code>continuum:ListEnvironments</code> , <code>continuum:ListIntegrations</code> , and <code>continuum:ListCollectors</code> . The environment inventory can also require <code>continuum:GetEnvironment</code> .
View an environment	Console operator	<code>continuum:ListEnvironments</code> and <code>continuum:GetEnvironment</code> . Scope these actions to the intended environment collection and resources.
View issues, findings, and issue artifacts	Console operator	<code>continuum:ListCodeIssues</code> , <code>continuum:GetCodeIssue</code> , <code>continuum:ListCodeFindings</code> , <code>continuum:GetCodeFinding</code> , <code>continuum:ListCodeIssueArtifacts</code> , and <code>continuum:GetCodeIssueArtifact</code> . Issue and finding lists use the environment resource. Artifact actions use the code issue resource. Issue and finding detail actions use their corresponding resources.
Create an environment that uses an integration	Console operator	<code>continuum:CreateEnvironment</code> , <code>continuum:ListIntegrations</code> , and <code>continuum:GetEnvironment</code> . Allow <code>continuum:CreateEnvironment</code> on <code>arn:aws:continuum: <i>region</i>:<i>account-id</i>:environments/*</code> for your account and Region, and on every referenced integration ARN.
Edit an environment or its scan schedule	Console operator	<code>continuum:GetEnvironment</code> and <code>continuum:UpdateEnvironment</code> . Choosing sources also requires <code>continuum:ListIntegrations</code> . A source-bearing update requires <code>continuum:UpdateEnvironment</code> on the environment and every referenced integration ARN.
Queue a scan and view scan history	Console operator	<code>continuum:CreateCodeScan</code> and <code>continuum:ListCodeScans</code> . Scan creation requires access to the environment and code-scan resources.

Task	Principal	Permissions and role requirements
Delete an environment	Console operator	<code>continuum:DeleteEnvironment</code> , <code>continuum:ListEnvironments</code> , and <code>continuum:GetEnvironment</code> .
Submit a custom finding	Console operator	<code>continuum:CreateCodeFinding</code> , <code>continuum:ListCodeFindings</code> , and <code>continuum:GetCodeFinding</code> .

Use the following permissions to create and manage integrations and their roles.

Task	Principal	Permissions and role requirements
View integrations	Console operator	<code>continuum:ListIntegrations</code> and <code>continuum:GetIntegration</code> . These read permissions do not grant permission to reference an integration from an environment.
Create a GitHub integration	Console operator	<code>continuum:StartIntegrationAuthorization</code> , <code>continuum:CreateIntegration</code> , <code>continuum:ListIntegrations</code> , and <code>continuum:GetIntegration</code> . GitHub installation and OAuth authorization are separate prerequisites. Creating a GitHub integration does not require <code>iam:PassRole</code> .
Create an Amazon S3 integration	Console operator	<code>continuum:CreateIntegration</code> , <code>continuum:ListIntegrations</code> , and <code>continuum:GetIntegration</code> . The role picker requires <code>iam:ListRoles</code> with <code>Resource: "*"</code> . Bucket browsing requires <code>s3:ListAllMyBuckets</code> with <code>Resource: "*"</code> . Creation also requires <code>iam:PassRole</code> on the exact same-account integration role under <code>/continuum/integrations/</code> . Restrict <code>iam:PassRole</code> with <code>iam:PassedToService</code> set to <code>continuum.amazonaws.com</code> .

Task	Principal	Permissions and role requirements
Create the Amazon S3 integration role	Customer administrator	<code>iam:CreateRole</code> and <code>iam:PutRolePolicy</code> . The verification command also requires <code>iam:GetRole</code> . Create the role under <code>/continuum/integrations/</code> . For the trust and access policies, see Granting access to the bucket .
Read source code and deliver patches	Amazon S3 integration role	AWS Continuum assumes this role by using its trust policy and matching external ID. Grant the integration role these permissions: • Bucket – unconditional <code>s3:ListBucket</code> on the bucket. • Source objects – <code>s3:GetObject</code> and <code>s3:GetObjectVersion</code> . • Patch destination – <code>s3:PutObject</code> , <code>s3:GetObject</code> , and <code>s3:GetObjectVersion</code> . These permissions belong to the integration role, not the console operator.
Delete an integration	Console operator	<code>continuum>DeleteIntegration</code> , <code>continuum:ListIntegrations</code> , and <code>continuum:GetIntegration</code> . You cannot delete an integration that an environment references.

Use the following permissions for collector tasks.

Task	Principal	Permissions and role requirements
View collectors and their connections	Console operator	<code>continuum:ListCollectors</code> , <code>continuum:GetCollector</code> , <code>continuum:ListAssociatedEnvironments</code> , and <code>continuum:ListAssociatedCollectors</code> .

Task	Principal	Permissions and role requirements
Enable the AWS collector	Console operator	<code>continuum:CreateCollector</code> , <code>continuum:ListCollectors</code> , and <code>continuum:GetCollector</code> . The setup also requires <code>iam:CreateServiceLinkedRole</code> , <code>config:DescribeConfigurationRecorders</code> , and <code>config:PutServiceLinkedConfigurationRecorder</code> . The collector service-linked role is separate from integration and collector data-access roles. For more information, see Using service-linked roles for AWS Continuum .
Change a collector name or Kubernetes role	Console operator	<code>continuum:UpdateCollector</code> , <code>continuum:ListCollectors</code> , and <code>continuum:GetCollector</code> . Selecting an existing role also requires <code>iam:ListRoles</code> with <code>Resource: "*"</code> . Setting a Kubernetes role requires <code>iam:PassRole</code> on that exact role. Renaming the collector or removing the role does not require <code>iam:PassRole</code> .
Associate a collector with an environment	Console operator and collector data-access role	The operator needs <code>continuum:AssociateCollector</code> , <code>continuum:ListEnvironments</code> , <code>continuum:GetEnvironment</code> , <code>continuum:ListCollectors</code> , <code>continuum:GetCollector</code> , <code>continuum:ListAssociatedCollectors</code> , and <code>continuum:ListAssociatedEnvironments</code> . The operator also needs <code>iam:PassRole</code> for the exact data-access role. The data-access role needs <code>continuum:GetCollectorDataAccessAuthorization</code> on the collector. Cross-account access also requires the collector resource policy to grant that action.

Task	Principal	Permissions and role requirements
Disassociate a collector from an environment	Console operator	<code>continuum:DisassociateCollector</code> , <code>continuum:GetEnvironment</code> , <code>continuum:GetCollector</code> , <code>continuum:ListAssociatedCollectors</code> , and <code>continuum:ListAssociatedEnvironments</code> . Disassociating a collector does not require <code>iam:PassRole</code> .
View or edit a collector access policy	Console operator	<code>continuum:ListCollectors</code> , <code>continuum:GetCollector</code> , <code>continuum:GetResourcePolicy</code> , and <code>continuum:PutResourcePolicy</code> . Scope <code>continuum:GetResourcePolicy</code> and <code>continuum:PutResourcePolicy</code> to the collector ARN. A policy update replaces the policy document, so retain intended grants and use the current revision.
Delete a collector	Console operator	<code>continuum>DeleteCollector</code> , <code>continuum:ListCollectors</code> , and <code>continuum:GetCollector</code> . Deleting a collector also requires <code>config:DeleteServiceLinkedConfigurationRecorder</code> .

For more information, see [Adding permissions to a user](#) in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
```

```

        "Action": [
            "iam:GetUserPolicy",
            "iam:ListGroupsForUser",
            "iam:ListAttachedUserPolicies",
            "iam:ListUserPolicies",
            "iam:GetUser"
        ],
        "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
        "Sid": "NavigateInConsole",
        "Effect": "Allow",
        "Action": [
            "iam:GetGroupPolicy",
            "iam:GetPolicyVersion",
            "iam:GetPolicy",
            "iam:ListAttachedGroupPolicies",
            "iam:ListGroupPolicies",
            "iam:ListPolicyVersions",
            "iam:ListPolicies",
            "iam:ListUsers"
        ],
        "Resource": "*"
    }
]
}

```

Allow users to view findings and issues

Use this policy to view findings and issues in one environment through the AWS Continuum API.

The `ListFindingsAndIssuesInEnvironment` statement grants list permissions on the environment's Amazon Resource Name (ARN). The `GetFindingsAndIssuesInEnvironment` statement grants get permissions on finding and issue ARNs within that environment.

For more information about permissions to use the console, see [Using the AWS Continuum console](#).

Replace *region* with your AWS Region. Replace *account-id* with your AWS account ID. Replace *environment-id* with the ID of the environment that you want to access.

Use the following policy:

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "ListFindingsAndIssuesInEnvironment",
    "Effect": "Allow",
    "Action": [
      "continuum:ListCodeFindings",
      "continuum:ListCodeIssues"
    ],
    "Resource": "arn:aws:continuum:region:account-id:environments/environment-id"
  },
  {
    "Sid": "GetFindingsAndIssuesInEnvironment",
    "Effect": "Allow",
    "Action": [
      "continuum:GetCodeFinding",
      "continuum:GetCodeIssue"
    ],
    "Resource": [
      "arn:aws:continuum:region:account-id:environments/environment-id/code-
findings/*",
      "arn:aws:continuum:region:account-id:environments/environment-id/code-
issues/*"
    ]
  }
]
}

```

Allow users to submit custom findings

Use this policy to submit custom findings to one environment through the AWS Continuum API.

`continuum:CreateCodeFinding` requires permission on both the environment ARN and the `code-findings/*` resources within that environment. Granting access to only one of these resources doesn't authorize the request.

For more information about permissions to use the console, see [Using the AWS Continuum console](#).

Replace *region* with your AWS Region. Replace *account-id* with your AWS account ID. Replace *environment-id* with the ID of the environment that you want to access.

Use the following policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SubmitCustomFindings",
      "Effect": "Allow",
      "Action": "continuum:CreateCodeFinding",
      "Resource": [
        "arn:aws:continuum:region:account-id:environments/environment-id",
        "arn:aws:continuum:region:account-id:environments/environment-id/code-findings/*"
      ]
    }
  ]
}
```

AWS managed policies for AWS Continuum

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to [create IAM customer managed policies](#) that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see [AWS managed policies](#) in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see [AWS managed policies for job functions](#) in the *IAM User Guide*.

AWS managed policy: AWSContinuumCollectorServiceRolePolicy

You can't attach `AWSContinuumCollectorServiceRolePolicy` to your IAM entities. This policy is attached to the `AWSServiceRoleForContinuumCollector` service-linked role that allows AWS Continuum to collect resource information and security findings and validate network reachability on your behalf. For more information, see [the section called "Using service-linked roles"](#).

Several statements use `aws:ResourceAccount` conditions to require the resource account to match the calling principal's account. These conditions apply to the following permissions:

- Lambda function and layer version reads
- Access Analyzer findings
- GuardDuty findings
- ECR image metadata
- WAFv2 resource associations
- API Gateway metadata
- Amazon EKS metadata reads
- Network Insights create, start, tag, and delete actions
- Tiros reachability queries
- S3 bucket reads

GuardDuty uses `Resource: "*" with the account condition.`

IAM policy reads are limited to AWS managed policies under `arn:aws:iam::aws:policy/*`. The following statements use `Resource: "*" without an aws:ResourceAccount condition:`

- Lambda event source mapping reads
- Elastic Load Balancing metadata reads
- Network Insights describe actions
- Security service discovery and Amazon Inspector findings
- S3 account public access reads

Permissions details

This policy includes the following permissions. AWS Continuum uses them to map your resources and their relationships and determine whether exposures are network-reachable in your environment.

- Lambda allows AWS Continuum to read function information, function policies, aliases, versions, and function URL configurations (`lambda:GetFunction`, `lambda:GetPolicy`, `lambda:ListAliases`, `lambda:ListVersionsByFunction`, and `lambda:ListFunctionUrlConfigs`), read layer versions and their policies (`lambda:GetLayerVersion` and `lambda:GetLayerVersionPolicy`), and list event source mappings (`lambda:ListEventSourceMappings`).
- IAM allows AWS Continuum to read AWS managed policy documents (`iam:GetPolicy` and `iam:GetPolicyVersion`).
- IAM Access Analyzer allows AWS Continuum to list analyzers and findings (`access-analyzer:ListAnalyzers` and `access-analyzer:ListFindings`).
- GuardDuty allows AWS Continuum to list and retrieve findings (`guardduty:ListFindings` and `guardduty:GetFindings`).
- Amazon ECR allows AWS Continuum to read private image metadata (`ecr:DescribeImages`).
- WAFv2 allows AWS Continuum to list resource associations for regional web ACLs and global web ACLs in `us-east-1` (`wafv2:ListResourcesForWebACL`).
- API Gateway allows AWS Continuum to read REST API resources, stage exports, authorizers, custom domain names, base path mappings, API exports, routes, and API mappings on the specified resource paths (`apigateway:GET`).
- Elastic Load Balancing allows AWS Continuum to read load balancer, listener, and target group attributes and target health (`elasticloadbalancing:DescribeLoadBalancerAttributes`, `elasticloadbalancing:DescribeListenerAttributes`, `elasticloadbalancing:DescribeTargetGroupAttributes`, and `elasticloadbalancing:DescribeTargetHealth`).
- Amazon EKS allows AWS Continuum to list and describe Pod Identity associations and access entries and list associated access policies (`eks:ListPodIdentityAssociations`, `eks:DescribePodIdentityAssociation`, `eks:ListAccessEntries`, `eks:DescribeAccessEntry`, and `eks:ListAssociatedAccessPolicies`).

These actions are limited to clusters, Pod Identity associations, and access entries in your account.

- Amazon EC2 allows AWS Continuum to read Network Insights paths and analyses (`ec2:DescribeNetworkInsightsPaths` and `ec2:DescribeNetworkInsightsAnalyses`).
- Amazon EC2 allows AWS Continuum to create Network Insights paths and start analyses (`ec2:CreateNetworkInsightsPath` and `ec2:StartNetworkInsightsAnalysis`).

Creation requests must include `AWSContinuum=true` and may specify only the `AWSContinuum` tag key. To start an analysis, the path must have the `AWSContinuum=true` resource tag.

- Amazon EC2 allows AWS Continuum to tag Network Insights paths and analyses during creation (`ec2:CreateTags`).

Tagging is permitted only during `CreateNetworkInsightsPath` or `StartNetworkInsightsAnalysis`. Tagging requests must include `AWSContinuum=true` and may specify only the `AWSContinuum` tag key.

- Amazon EC2 allows AWS Continuum to delete Network Insights paths and analyses (`ec2:DeleteNetworkInsightsPath` and `ec2:DeleteNetworkInsightsAnalysis`).

The paths and analyses must have the `AWSContinuum=true` resource tag.

- Tiros allows AWS Continuum to create reachability queries and retrieve their answers (`tiros:CreateQuery` and `tiros:GetQueryAnswer`).
- Amazon Inspector allows AWS Continuum to list findings (`inspector2:ListFindings`).
- Amazon S3 allows AWS Continuum to read account-level public access block settings (`s3:GetAccountPublicAccessBlock`) and bucket ACLs, locations, policy status, and public access block settings (`s3:GetBucketAcl`, `s3:GetBucketLocation`, `s3:GetBucketPolicyStatus`, and `s3:GetBucketPublicAccessBlock`).

For the complete policy JSON, see [Permissions policy](#).

To view the policy in the AWS Managed Policy Reference, see [AWSContinuumCollectorServiceRolePolicy](#).

AWS Continuum managed policy documentation updates

The following table lists documentation updates for AWS Continuum managed policies. For alerts about guide updates, subscribe to the RSS feed on the [Document history](#) page.

Change	Description	Date
AWSContinuumCollectorServiceRolePolicy documentation	Updated the permissions documentation for <code>AWSServiceRoleForContinuumCollector</code> . The guide now includes the collector policy statements, resource scopes, and tag requirements.	October 13, 2026

Using service-linked roles for AWS Continuum

AWS Continuum uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to AWS Continuum. Service-linked roles are predefined by AWS Continuum and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up AWS Continuum easier because you don't have to manually add the necessary permissions. AWS Continuum defines the permissions of its service-linked roles, and unless defined otherwise, only AWS Continuum can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy can't be attached to any other IAM entity.

You can delete a service-linked role only after first deleting its related resources. This protects your AWS Continuum resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see [AWS services that work with IAM](#) and look for the services that have **Yes** in the **Service-linked role** column. Choose a **Yes** link to view the service-linked role documentation for that service.

Topics

- [Service-linked role permissions for AWS Continuum](#)
- [Creating a service-linked role for AWS Continuum](#)
- [Editing a service-linked role for AWS Continuum](#)
- [Deleting a service-linked role for AWS Continuum](#)

Service-linked role permissions for AWS Continuum

AWS Continuum uses the service-linked role named `AWSServiceRoleForContinuumCollector`. This role allows AWS Continuum to collect resource information and security findings and validate network reachability. The role trusts the `collector.continuum.amazonaws.com` service to assume the role.

Important

Network Insights creation requests must include the `AWSContinuum=true` tag and may specify only the `AWSContinuum` tag key. Tagging is permitted only during path or analysis creation. The role can start analyses only on paths with this tag and delete only paths and analyses with this tag. These create, start, tag, and delete permissions require the resource account to match the calling principal's account.

The permissions policy for the role, named `AWSContinuumCollectorServiceRolePolicy`, allows AWS Continuum to perform the following tasks. Network Insights paths and analyses and Tiros reachability queries help AWS Continuum validate network reachability.

- **Lambda** – Retrieve function information, policies, aliases, versions, and function URL configurations. Retrieve layer versions, their policies, and event source mappings.
- **IAM** – Retrieve AWS managed policy documents and their versions.
- **IAM Access Analyzer** – List analyzers and findings.
- **GuardDuty** – List and retrieve findings in your account.
- **Amazon Inspector** – List findings.
- **Amazon ECR** – Retrieve metadata about private container images.
- **WAFv2** – List resources associated with regional web ACLs and global web ACLs in `us-east-1`.
- **API Gateway** – Retrieve API resource, export, authorizer, custom domain, route, and mapping metadata for the resource paths in the policy.
- **Elastic Load Balancing** – Retrieve load balancer, listener, and target group attributes and target health.
- **Amazon EKS** – List and describe Pod Identity associations and access entries and list associated access policies. These actions are limited to clusters, Pod Identity associations, and access entries in your account.
- **Amazon EC2** – Retrieve information about Network Insights paths and analyses.

- **Amazon EC2** – Create Network Insights paths and analyses and tag them with `AWSContinuum=true` during creation.
- **Amazon EC2** – Start analyses on paths tagged `AWSContinuum=true`. Delete paths and analyses with this tag.
- **Tiros** – Create reachability queries and retrieve their answers.
- **Amazon S3** – Retrieve account-level public access block settings and bucket ACLs, locations, policy status, and public access block settings.

Note

If you monitor API calls, you might see calls to Tiros APIs. Tiros is a service that is only accessible by AWS services and that surfaces network reachability findings to Reachability Analyzer. Calls to the Tiros endpoint are required for Reachability Analyzer to function. To access Reachability Analyzer, users must also have the same API permissions.

Permissions policy

The role is configured with the following permissions policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LambdaFunctionSupplementalMetadataRead",
      "Effect": "Allow",
      "Action": [
        "lambda:GetPolicy",
        "lambda:ListAliases",
        "lambda:ListVersionsByFunction",
        "lambda:ListFunctionUrlConfigs",
        "lambda:GetFunction"
      ],
      "Resource": "arn:aws:lambda:*:*:function:*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    }
  ],
}
```

```

{
  "Sid": "LambdaLayerVersionSupplementalMetadataRead",
  "Effect": "Allow",
  "Action": [
    "lambda:GetLayerVersion",
    "lambda:GetLayerVersionPolicy"
  ],
  "Resource": "arn:aws:lambda:*:*:layer:*:*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
  }
},
{
  "Sid": "LambdaEventSourceMappingRead",
  "Effect": "Allow",
  "Action": "lambda:ListEventSourceMappings",
  "Resource": "*"
},
{
  "Sid": "IamAwsManagedPolicyDocumentRead",
  "Effect": "Allow",
  "Action": [
    "iam:GetPolicy",
    "iam:GetPolicyVersion"
  ],
  "Resource": "arn:aws:iam::aws:policy/*"
},
{
  "Sid": "AccessAnalyzerFindingsRead",
  "Effect": "Allow",
  "Action": "access-analyzer:ListFindings",
  "Resource": "arn:aws:access-analyzer:*:*:analyzer/*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
  }
},
{
  "Sid": "GuardDutyFindingsRead",
  "Effect": "Allow",
  "Action": [

```

```

        "guardduty:ListFindings",
        "guardduty:GetFindings"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
    }
},
{
    "Sid": "EcrPrivateImageMetadataRead",
    "Effect": "Allow",
    "Action": "ecr:DescribeImages",
    "Resource": "arn:aws:ecr:*:*:repository/*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
    }
},
{
    "Sid": "Wafv2ResourceAssociationRead",
    "Effect": "Allow",
    "Action": "wafv2:ListResourcesForWebACL",
    "Resource": [
        "arn:aws:wafv2:*:*:regional/webacl/*/*",
        "arn:aws:wafv2:us-east-1:*:global/webacl/*/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
    }
},
{
    "Sid": "ApiGatewaySupplementalMetadataRead",
    "Effect": "Allow",
    "Action": "apigateway:GET",
    "Resource": [
        "arn:aws:apigateway:*::/restapis/*/resources",
        "arn:aws:apigateway:*::/restapis/*/stages/*/exports/*",
        "arn:aws:apigateway:*::/restapis/*/authorizers",
        "arn:aws:apigateway:*::/domainnames",

```

```

    "arn:aws:apigateway:*::/domainnames/*/basepathmappings",
    "arn:aws:apigateway:*::/apis/*/exports/*",
    "arn:aws:apigateway:*::/apis/*/routes",
    "arn:aws:apigateway:*::/apis/*/authorizers",
    "arn:aws:apigateway:*::/domainnames/*/apimappings"
  ],
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
  }
},
{
  "Sid": "Elbv2SupplementalMetadataRead",
  "Effect": "Allow",
  "Action": [
    "elasticloadbalancing:DescribeLoadBalancerAttributes",
    "elasticloadbalancing:DescribeListenerAttributes",
    "elasticloadbalancing:DescribeTargetGroupAttributes",
    "elasticloadbalancing:DescribeTargetHealth"
  ],
  "Resource": "*"
},
{
  "Sid": "EksSupplementalMetadataRead",
  "Effect": "Allow",
  "Action": [
    "eks:ListPodIdentityAssociations",
    "eks:DescribePodIdentityAssociation",
    "eks:ListAccessEntries",
    "eks:DescribeAccessEntry",
    "eks:ListAssociatedAccessPolicies"
  ],
  "Resource": [
    "arn:aws:eks:*::cluster/*",
    "arn:aws:eks:*::podidentityassociation/*",
    "arn:aws:eks:*::access-entry/*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
  }
}
},

```

```

{
  "Sid": "NetworkInsightsRead",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeNetworkInsightsPaths",
    "ec2:DescribeNetworkInsightsAnalyses"
  ],
  "Resource": "*"
},
{
  "Sid": "CreateTaggedNetworkInsightsPath",
  "Effect": "Allow",
  "Action": "ec2:CreateNetworkInsightsPath",
  "Resource": "arn:aws:ec2:*:*:network-insights-path/*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}",
      "aws:RequestTag/AWSContinuum": "true"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": "AWSContinuum"
    }
  }
},
{
  "Sid": "CreateTaggedNetworkInsightsAnalysis",
  "Effect": "Allow",
  "Action": "ec2:StartNetworkInsightsAnalysis",
  "Resource": "arn:aws:ec2:*:*:network-insights-analysis/*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}",
      "aws:RequestTag/AWSContinuum": "true"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": "AWSContinuum"
    }
  }
},
{
  "Sid": "UseTaggedNetworkInsightsPath",
  "Effect": "Allow",
  "Action": "ec2:StartNetworkInsightsAnalysis",
  "Resource": "arn:aws:ec2:*:*:network-insights-path/*",

```

```

    "Condition": {
      "StringEquals": {
        "aws:ResourceAccount": "${aws:PrincipalAccount}",
        "aws:ResourceTag/AWSContinuum": "true"
      }
    },
  },
  {
    "Sid": "TagNetworkInsightsResourcesOnCreate",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": [
      "arn:aws:ec2:*:*:network-insights-path/*",
      "arn:aws:ec2:*:*:network-insights-analysis/*"
    ],
    "Condition": {
      "StringEquals": {
        "aws:ResourceAccount": "${aws:PrincipalAccount}",
        "aws:RequestTag/AWSContinuum": "true",
        "ec2:CreateAction": [
          "CreateNetworkInsightsPath",
          "StartNetworkInsightsAnalysis"
        ]
      }
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": "AWSContinuum"
    }
  }
},
{
  "Sid": "DeleteTaggedNetworkInsightsResources",
  "Effect": "Allow",
  "Action": [
    "ec2:DeleteNetworkInsightsAnalysis",
    "ec2:DeleteNetworkInsightsPath"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:network-insights-path/*",
    "arn:aws:ec2:*:*:network-insights-analysis/*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}",
      "aws:ResourceTag/AWSContinuum": "true"
    }
  }
}

```

```

    }
  }
},
{
  "Sid": "TirosQueryRead",
  "Effect": "Allow",
  "Action": [
    "tiros:CreateQuery",
    "tiros:GetQueryAnswer"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
  }
},
{
  "Sid": "SecurityServiceDiscoveryAndFindings",
  "Effect": "Allow",
  "Action": [
    "access-analyzer:ListAnalyzers",
    "inspector2:ListFindings"
  ],
  "Resource": "*"
},
{
  "Sid": "S3AccountPublicAccessRead",
  "Effect": "Allow",
  "Action": "s3:GetAccountPublicAccessBlock",
  "Resource": "*"
},
{
  "Sid": "S3BucketExposureRead",
  "Effect": "Allow",
  "Action": [
    "s3:GetBucketAcl",
    "s3:GetBucketLocation",
    "s3:GetBucketPolicyStatus",
    "s3:GetBucketPublicAccessBlock"
  ],
  "Resource": "arn:aws:s3:::*",
  "Condition": {
    "StringEquals": {

```

```
        "aws:ResourceAccount": "${aws:PrincipalAccount}"
      }
    }
  }
]
```

You must configure permissions to allow your users, groups, or roles to create, edit, or delete a service-linked role. For more information, see [Service-linked role permissions](#) in the *IAM User Guide*.

Creating a service-linked role for AWS Continuum

You can use the IAM console, the AWS CLI, or the AWS API to create `AWSServiceRoleForContinuumCollector`. In the AWS CLI or the AWS API, specify `collector.continuum.amazonaws.com` as the service name.

For instructions, see [Creating a service-linked role](#) in the *IAM User Guide*.

Editing a service-linked role for AWS Continuum

AWS Continuum does not allow you to edit the `AWSServiceRoleForContinuumCollector` service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see [Editing a service-linked role](#) in the *IAM User Guide*.

Deleting a service-linked role for AWS Continuum

You can use the IAM console, the AWS CLI, or the AWS API to delete `AWSServiceRoleForContinuumCollector`. For instructions on preparing the role for deletion, see [Deleting a service-linked role](#) in the *IAM User Guide*.

Note

If AWS Continuum is using the role, deletion might fail. Wait a few minutes and try again.

Troubleshooting AWS Continuum identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS Continuum and IAM.

To change permissions, use an identity authorized to update the relevant IAM policies, or ask your AWS administrator to make the changes.

Topics

- [I want to allow people outside of my AWS account to access my AWS Continuum resources](#)
- [I am not authorized to use an integration when I create or update an environment](#)
- [I am not authorized to create the Continuum collector service-linked role](#)
- [I am not authorized to configure a service-linked recorder for the collector](#)
- [I am not authorized to pass the collector data-access role](#)
- [I can't read or update the resource policy on my collector](#)
- [I am not authorized to associate a collector with my environment](#)
- [My Amazon S3 integration can't access my bucket](#)
- [I am not authorized to pass the role for an integration](#)

I want to allow people outside of my AWS account to access my AWS Continuum resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether AWS Continuum supports these features, see [How AWS Continuum works with IAM](#).
- To learn how to provide access to your resources across AWS accounts that you own, see [Providing access to an IAM user in another AWS account that you own](#) in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see [Providing access to AWS accounts owned by third parties](#) in the *IAM User Guide*.
- To learn how to provide access through identity federation, see [Providing access to externally authenticated users \(identity federation\)](#) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see [Cross account resource access in IAM](#) in the *IAM User Guide*.

I am not authorized to use an integration when I create or update an environment

You might see an `AccessDeniedException` error when you create or update an AWS Continuum environment that references an integration. This can happen when the caller can create or update the environment but can't use its referenced integration. AWS Continuum checks that the caller can use the referenced integration before it accepts the environment configuration.

To resolve this error, grant the calling principal the permission for the operation that you are performing:

- When creating an environment, `continuum:CreateEnvironment` on the `account- and Region-scoped environments/*` resource and on every referenced integration ARN.
- When updating source configuration, `continuum:UpdateEnvironment` on the environment ARN and on every referenced integration ARN.

For information about read actions and resource guidance for these console tasks, see [Using the AWS Continuum console](#).

I am not authorized to create the Continuum collector service-linked role

You might see an `AccessDeniedException` error when you create a collector, or when you create the service-linked role that the collector uses, if you don't have permission to create that role. AWS Continuum uses the service-linked role named `AWSServiceRoleForContinuumCollector`, which trusts the `collector.continuum.amazonaws.com` service principal. Creating this role requires the `iam:CreateServiceLinkedRole` action.

To resolve this error, grant the calling principal the `iam:CreateServiceLinkedRole` action for the `collector.continuum.amazonaws.com` service-linked role. For more information about this role, see [Using service-linked roles for AWS Continuum](#).

I am not authorized to configure a service-linked recorder for the collector

You might see an `AccessDeniedException` error when the collector configures the service-linked recorder that it uses to record resource configuration, if the calling principal lacks the required permission.

To resolve this recorder authorization error, grant the calling principal both of the following permissions:

- `config:DescribeConfigurationRecorders`
- `config:PutServiceLinkedConfigurationRecorder`

I am not authorized to pass the collector data-access role

You might see an `AccessDeniedException` error when you call `AssociateCollector` to associate a collector with an environment and provide a collector data-access role. AWS Continuum assumes this role when it requests an authorization decision for the collector, so you must have permission to pass the role to AWS Continuum.

To resolve this error, grant the calling principal the `iam:PassRole` action for the exact collector data-access role. For an example policy and the full association procedure, see [Authorizing an environment to use collector data](#).

I can't read or update the resource policy on my collector

You might see an `AccessDeniedException` error when you call `GetResourcePolicy` or `PutResourcePolicy` to manage the resource policy on a collector. This happens when the calling principal doesn't have permission for these actions on the collector, or when the principal doesn't belong to the collector's AWS account.

To resolve this error, grant the calling principal the `continuum:GetResourcePolicy` and `continuum:PutResourcePolicy` actions for the collector ARN, and make sure that the principal belongs to the collector account.

If `PutResourcePolicy` instead returns a `ConflictException` error, the stored policy changed after you read it. Repeat the read-and-update procedure to preserve the latest policy statements when you retry. For instructions on updating a collector resource policy, see [Authorizing an environment to use collector data](#).

I am not authorized to associate a collector with my environment

You might see an `AccessDeniedException` error when you call `AssociateCollector` to associate a collector with an environment. This happens when the caller lacks required permissions or the collector data-access role lacks permission to access the collector. AWS Continuum assumes the data-access role and checks its access to the collector before creating the association.

To resolve this error, ask an IAM administrator to grant the caller both of the following permissions:

- `continuum:AssociateCollector` for the environment.

- `iam:PassRole` for the collector data-access role. For more information, see [I am not authorized to pass the collector data-access role](#).

Grant the collector data-access role `continuum:GetCollectorDataAccessAuthorization` for the collector ARN. The caller's permissions do not replace this role permission. Confirm that the role trust policy allows AWS Continuum (`continuum.amazonaws.com`) to assume the role.

For cross-account access, also confirm that the collector resource policy grants the environment account `continuum:GetCollectorDataAccessAuthorization` for the exact collector ARN. For the full list of checks, see [Troubleshooting collector authorization](#).

My Amazon S3 integration can't access my bucket

You might see an access-validation failure or an `AccessDeniedException` error when you create an Amazon S3 integration, or when AWS Continuum tries to read your code from the bucket. AWS Continuum assumes the IAM role that you supply for the integration and then reads objects from the bucket, so this error usually means that the role, its trust policy, or its permissions are misconfigured.

To resolve this error, verify the following:

- The role ARN uses the `/continuum/integrations/` path.
- The integration role trust policy and external ID match the integration.
- The role's permissions policy grants `s3:ListBucket` on the bucket without conditions. It also grants `s3:GetObject` and `s3:GetObjectVersion` on the code and context prefixes that your environment uses.

If role validation succeeds, verify that the console operator has `continuum:CreateIntegration` and `iam:PassRole` for the exact integration role. For the complete console permission guidance, see [Using the AWS Continuum console](#).

For the full set of bucket permissions, the trust policy, and the integration procedure, see [Getting started through Amazon S3](#).

I am not authorized to pass the role for an integration

You might see an `AccessDeniedException` error that mentions `iam:PassRole` when you create an integration that supplies an IAM role for AWS Continuum to use, such as an Amazon S3

integration. This happens when the calling principal doesn't have permission to pass the specified role to AWS Continuum.

To resolve this error, grant the calling principal the `iam:PassRole` action for the exact role that the integration uses. For an Amazon S3 integration, the role must belong to the same account and use the `/continuum/integrations/` path. Restrict the permission by using the `iam:PassedToService` condition key with the value `continuum.amazonaws.com`.

Compliance validation for AWS Continuum

Our new AWS sign-up experience is not designed for regulated workloads. If you're using our new AWS sign-up experience, but you want to use AWS for regulated workloads, you can [sign up for AWS \(advanced\)](#) or [activate advanced features](#) for your AWS environment.

To learn whether an AWS service is within the scope of specific compliance programs, see [AWS services in Scope by Compliance Program](#) and choose the compliance program that you are interested in. For general information, see [AWS Compliance Programs](#).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading Reports in AWS Artifact](#).

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see [AWS Security Documentation](#).

Resilience in AWS Continuum

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see [AWS Global Infrastructure](#).

In addition to the AWS global infrastructure, AWS Continuum offers several features to help support your data resiliency and backup needs.

Quotas for AWS Continuum

Your AWS account has default quotas to maintain the performance of the service and to ensure appropriate usage of AWS Continuum. The default quotas for your account might change depending on regional factors, payment history, fraudulent usage, or approval of a quota increase request.

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific.

To request a quota increase for AWS Continuum, contact your AWS account team.

Your AWS account has the following quotas related to AWS Continuum.

Resource	Default	Description
Integrations	10	The maximum number of integrations, such as GitHub integrations, that you can connect in this account in the current Region.
Environments	10	The maximum number of environments that you can create in this account in the current Region.
Repositories per environment	10	The maximum number of code repositories that you can add to a single environment across all integrations.
Collectors	1	The maximum number of AWS Continuum collectors that you can create in this account in the current Region.
Collectors per environment	100	The maximum number of AWS Continuum collectors

Resource	Default	Description
		that you can associate with a single environment.

** Note**

If an existing environment contains more than 10 repositories, you can view the environment and update settings that do not change its repository selection. To change its repository selection or start a new scan, reduce the selection to your account's current repository quota or fewer.

Document history for the AWS Continuum User Guide

The following table describes the documentation releases for AWS Continuum.

Change	Description	Date
Collector service-linked role documentation	Updated the service-linked role documentation for <code>AWSServiceRoleForContinuumCollector</code> . The guide now describes the <code>AWSContinuumCollectorServiceRolePolicy</code> permissions policy.	October 13, 2026
Initial release	Initial release of the AWS Continuum User Guide	December 31, 2020