# AWS Resilience Hub

## User Guide

# AWS Resilience Hub: User Guide

# Table of Contents

# What is AWS Resilience Hub?

AWS Resilience Hub gives you a central place to define, validate, and track the resiliency of your AWS application. AWS Resilience Hub helps you to protect your applications from disruptions, and reduce recovery costs to optimize business continuity to help meet compliance and regulatory requirements. Use AWS Resilience Hub to do the following:

- Analyze your infrastructure and get recommendations to improve the resiliency of your applications. In addition to architectural guidance for improving your application resiliency, the recommendations provide code for meeting your resiliency policy, implementing tests, alarms, and standard operating procedures (SOPs) that you can deploy and run with your application in your integration and delivery (CI/CD) pipeline.
- Validate recovery time objective (RTO) and recovery point objective (RPO) targets under different conditions.
- Optimize business continuity while reducing recovery costs.
- Identify and resolve issues before they occur in production.

After you deploy an application into production, you can add AWS Resilience Hub to your CI/CD pipeline to validate every build before it is released into production.

AWS Trusted Advisor now inspects and provides resilience score and indications of meeting or breaching an application's resilience policy (RTO/RPO). With the resiliency checks from AWS Trusted Advisor, you can see which applications have resiliency risks and address them in AWS Resilience Hub. For more information about working with AWS Trusted Advisor, see AWS Trusted Advisor (p. 73).

**Describe**

Describe your applications using AWS CloudFormation with cross-Region and cross-account stacks. Alternatively, use AWS Terraform state files. Applications can also be described using resource groups, or you can choose from applications that are already defined in AWS Service Catalog AppRegistry.

**Define**

Define the resilience policies for your applications. These policies include RTO and RPO targets for applications, infrastructure, Availability Zone, and Region disruptions.

**Assess**

The AWS Resilience Hub assessment uses best practices from the AWS Well-Architected Framework to analyze the components of an application and uncover potential resilience weaknesses. These weaknesses can be caused by incomplete infrastructure setup, misconfiguration, or situations where additional configuration improvements are needed.

**Validate**

After the application and standard operating procedures (SOPs) are updated to incorporate recommendations from the resilience assessment, you can use AWS Resilience Hub to test and verify your application to see if it meets its resilience targets before releasing it into production. AWS Resilience Hub works with AWS Fault Injection Simulator (AWS FIS), a chaos engineering service, to provide fault-injection simulations of real-world failures such as network errors or too many open connections to a database, to validate the application recovers within the resilience targets you defined. AWS Resilience Hub also provides API operations for you to integrate its resilience assessment and testing into your CI/CD pipelines for ongoing resilience validation. Including resilience validation in CI/CD pipelines helps make sure that changes to the application’s underlying infrastructure don't compromise resilience.

**View and track**

AWS Resilience Hub provides a comprehensive view of your overall application portfolio resilience status through its dashboard. To help you track the resilience of applications, AWS Resilience Hub aggregates and organizes resilience events (such as unavailable database or failed resilience validation), alerts, and insights from services like Amazon CloudWatch, Amazon Route 53 Application Recovery Controller, and AWS FIS. AWS Resilience Hub also generates a resilience score, a scale that indicates the level of implementation for recommended resilience tests, alarms, and recovery SOPs. This score is used to measure resilience improvements over time.

# AWS Resilience Hub concepts

These concepts can help you better understand the AWS Resilience Hub's approach to helping improve application resiliency and prevent application outages.

## Resiliency

The ability to maintain availability and to recover from software and operational disruption in a designated time frame.

## Recovery point objective (RPO)

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

## Recovery time objective (RTO)

The maximum acceptable delay between the interruption of service and restoration of service. This determines what is considered an acceptable time window when service is unavailable.

## Application

An AWS Resilience Hub application is a collection of AWS resources used to discover and prevent application disruptions and outages. It also helps systems automatically recover from disruptions.

## Application component

A group of related AWS resources that work and fail as a single unit. For example, if you have a primary and replica database, then both databases belong to the same application component.

AWS Resilience Hub determines which AWS resources can belong to which type of application component. For example, a `DBInstance` cannot belong to `AWS::ResilienceHub::ComputeAppComponent` compute.

## Application compliance status

AWS Resilience Hub reports the following compliance status types for your applications.

**Policy met**

The application is estimated to meet its RTO and RPO objectives defined in the policy. All its components meet the defined policy objectives. For example, you selected an RTO and RPO of 24 hours for disruptions across AWS Regions. AWS Resilience Hub can see that your backups are copied to your fallback Region. You are still expected to maintain a recover from a backup standard operating procedure (SOP), and to test and time it. This is in the operational recommendations and part of your overall resiliency score.

**Policy breached**

The application could not be estimated to meet the RTO and RPO objectives defined in the policy. One or more of its application components do not satisfy the policy objectives. For example, you selected an RTO and RPO of 24 hours for disruptions across AWS Regions, but your database configuration does not include any cross-Region recovery method, such as a global replication and backup copies.

**Not assessed**

The application requires an assessment. It's not currently assessed or tracked.

**Changes detected**

There is a new published version of the application that has not yet been assessed.

# Resiliency assessment

AWS Resilience Hub uses a list of gaps and potential remedies to measure the effectiveness of a selected policy to recover and continue from a disaster. It evaluates each application component or application compliance status with the policy. This report includes cost optimization recommendations and references to potential issues.

# Resiliency score

AWS Resilience Hub generates a score that indicates how closely your application follows our recommendations for meeting the application's resiliency policy, alarms, standard operating procedures (SOPs), and tests.

# Disruption type

AWS Resilience Hub helps you assess resiliency against the following types of outages:

**Application RTO and RPO**

The infrastructure is healthy, but the application or software stack doesn't operate as needed. This may occur after deployment of new code, configuration changes, data corruption, or malfunction of downstream dependencies.

**Cloud Infrastructure RTO and RPO**

The cloud infrastructure is not functioning as expected because of an outage. An outage may occur because of a local error in one or more components. In most cases, this type of outage is resolved by rebooting, recycling, or reloading the faulty components.

**Cloud Infrastructure AZ outage**

One or more Availability Zones are unavailable. This type of outage can be resolved by switching to a different Availability Zone.

**Cloud Infrastructure Region outage**

One or more Regions are unavailable. This type of outage can be resolved by switching to a different Region.

# Fault injection experiments

AWS Resilience Hub recommends tests to verify application resiliency against different types of outages. These outages include application, infrastructure, Availability Zones (AZ), or Region outages of application components.

These experiments let you do the following:

- Inject a failure.
- Verify that alarms can detect an outage.
- Verify that recovery procedures, or standard operating procedures (SOPs), work correctly to recover the application from the outage.

Tests for SOPs measure RTO and RPO. You can test different application configurations and measure whether the output RTO and RPO meet the objectives defined in your policy.

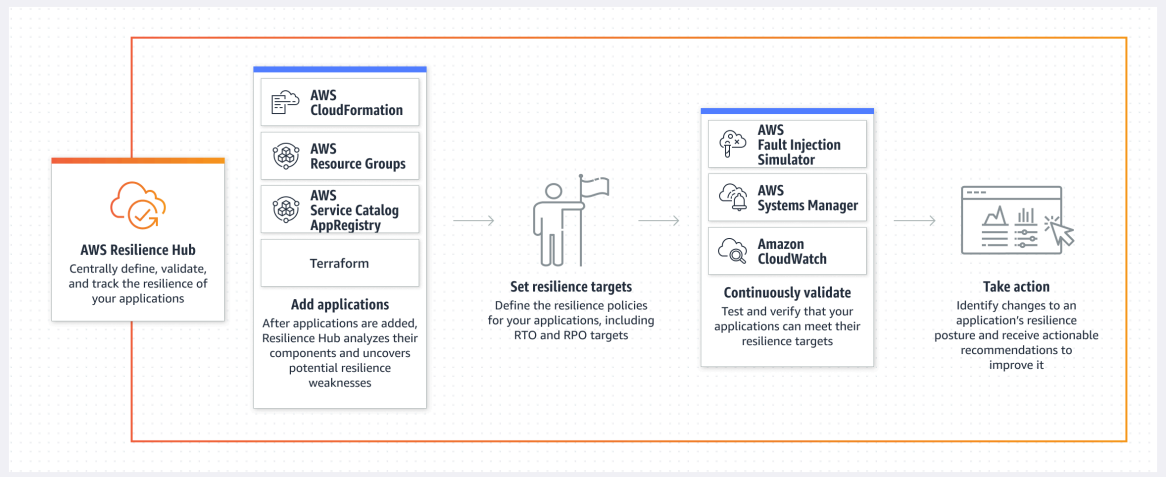
## SOP

The SOP (standard operating procedure) manages recovery procedures that are based on the outage type and application components in the application.

# How AWS Resilience Hub works

AWS Resilience Hub helps you proactively prepare and protect your AWS applications from disruptions. The AWS Resilience Hub offers resiliency assessment and validation that integrate into your software development lifecycle to uncover resiliency weaknesses. AWS Resilience Hub helps you to estimate whether or not the recovery time objective (RTO) and recovery point objective (RPO) for your applications can be met, and helps resolve issues before they are released into production.

After you deploy an AWS application into production, you can use AWS Resilience Hub to continue tracking the resiliency posture of your application. If an outage occurs, AWS Resilience Hub sends a notification to the operator to launch the associated recovery process.




The following steps provide a high-level outline of how AWS Resilience Hub works.

**1. Describe the existing AWS application that you want to protect from disruptions as a AWS Resilience Hub application and then set resiliency objectives for the application.**

When you describe the application, you import resources from AWS CloudFormation stacks, Terraform state files, resource groups, or an AppRegistry to form the structural basis of an application in AWS Resilience Hub. You can also use an existing application to build off an existing structure. Then, you attach a resiliency policy to the application.

An AWS Resilience Hub resiliency policy contains the information and objectives that are used to assess whether your application can recover from a disruption type, such as software or hardware disruption. When you create a resiliency policy, you define RTO and RPO for the disruption types. These objectives are used to estimate whether the application meets the resiliency policy.

**2. Assess the application to learn whether it meets your objectives.**

After you describe your application and attach a resiliency policy to it, run a resiliency assessment. The assessment evaluates your application configuration against the resiliency policy that is attached to the application and generates a report. The report shows how your application measures against the objectives in your resiliency policy.

**3. Receive recommendations to improve resiliency.**

To improve resiliency, update your application and resiliency policy according to the recommendations from the assessment report. Recommendations include configurations of components, alarms, tests, and recovery SOPs. Then, you can run another assessment and compare the results with the previous report to see how much resiliency improves. Reiterate this process until your RTO and RPO estimates meet your RTO and RPO goals.

**4. Validate objectives and disaster recovery procedures.**

Run tests to measure the resiliency of your AWS resources and the amount of time it takes to recover from application, infrastructure, Availability Zone, and AWS Region outages. To measure resiliency, these tests simulate outages of your AWS resources. Examples of outages include network unavailable errors, failovers, stopped processes, Amazon RDS boot recovery, and problems with your Availability Zone.

When the test concludes, you can determine whether an application can recover from the outage types defined in the RTO in the resiliency policy.

**5. View and track your application resiliency over time.**

After you deploy an AWS application into production, you can use AWS Resilience Hub to continue tracking the resiliency posture of the application. If an outage occurs, the operator can view the outage in AWS Resilience Hub and launch the associated recovery process.

**6. Start recovery if there is a disruption.**

If an application disruption occurs, AWS Resilience Hub helps identify the type of disruption and alerts the operator. Then, the operator can launch the associated SOP for recovery.

# AWS Resilience Hub supported resources

Resources that affect RTO and RPO are fully supported by AWS Resilience Hub top-level resources such as `AWS::RDS::DBInstance` and `AWS::RDS::DBCluster`.

AWS Resilience Hub supports the following resources:

- Compute
  - Amazon EC2
  - AWS Lambda
- Container
  - Amazon ECS
- Database
  - Amazon RDS
  - Amazon DynamoDB

  - Amazon DocumentDB
- Networking and Content Delivery
  - Amazon Route 53
  - Elastic Load Balancing
  - Network Address Translation (NAT)
- Storage
  - Amazon EBS
  - Amazon EFS
  - Amazon S3
- Others
  - Amazon API Gateway
  - Amazon Route 53 Application Recovery Controller (Amazon Route 53 ARC)
  - Amazon SNS
  - Amazon SQS
  - AWS Auto Scaling
  - AWS Backup
  - AWS Elastic Disaster Recovery

**Note**

- AWS Backup in AWS Resilience Hub assesses only Amazon Elastic Block Store, Amazon EFS, Amazon S3, Amazon Aurora Global Database, Amazon DynamoDB, and Amazon RDS services.
- Amazon Route 53 ARC in AWS Resilience Hub assesses only Amazon DynamoDB global, Elastic Load Balancing, Amazon RDS, and AWS Auto Scaling groups.
- Amazon Elastic Block Store in AWS Resilience Hub assesses only Amazon Elastic Block Store volumes.
- For AWS Resilience Hub to assess the cross-region resources, group the resources under a single application component. For more information about the resources supported by each of the AWS Resilience Hub application components and grouping resources, see Grouping resources in an AppComponent (p. 18).

AWS Resilience Hub ignores the following types of resources:

- **Resources that do not affect RTO or RPO** – Resources such as `AWS::RDS::DBParameterGroup`, which never affects RTO or RPO and is always ignored by AWS Resilience Hub.
- **Non-top level resources** – AWS Resilience Hub only imports top-level resources, because they can derive other properties by querying the properties of top-level resources. For example, `AWS::ApiGateway::RestApi` and `AWS::ApiGatewayV2::Api` are supported resources for Amazon API Gateway. However, `AWS::ApiGatewayV2::Stage` is not a top-level resource. Therefore, it is not imported by AWS Resilience Hub. This doesn't mean that AWS Resilience Hub ignores these resources.

**Note**
**Unsupported resources**

- Multiple resources cannot be identified using Resource Groups (Amazon Route 53 RecordSets and API-GW HTTP) and Amazon Aurora Global resources. If you would like to analyze these resources as part of your assessment, you have to manually add the resource to the application. However, when you add Amazon Aurora Global resources for assessment, it must be grouped with the Amazon RDS instances app component. For more information about editing resources, see the section called “Edit application resources” (p. 16).

- These resources can affect RTO and RPO, but they aren't fully supported by AWS Resilience Hub at this time. AWS Resilience Hub makes an effort to warn users about unsupported resources if the application is backed by a CloudFormation stack, Terraform state file, resource group, or AppRegistry application.

# Getting started

This section describes how to start using AWS Resilience Hub. This includes creating AWS Identity and Access Management (IAM) permissions for an account.

## Prerequisites

Before you can use the Resilience Hub, you must set up:

- One or more AWS accounts.
- AWS Identity and Access Management (IAM) permissions.

### Create IAM roles for an account

Resilience Hub integrates with AWS Identity and Access Management (IAM) so you can grant Resilience Hub users the permissions to access, test, and monitor applications to prevent disruptions and implement disaster recovery.

Permissions for Resilience Hub depend on the type of resources that comprise your applications, and the specific Resilience Hub features that they use. For example, if an application consists of RDS DB, then you need to have `rds:DescribeDBInstances` permission.

For instructions on working with IAM roles and policies, see How AWS Resilience Hub works with IAM (p. 49).

## Add an application to AWS Resilience Hub

AWS Resilience Hub offers resiliency assessment and validation that integrates into your software development lifecycle. Resilience Hub helps you proactively prepare and protect your AWS applications from disruptions by:

- Uncovering resiliency weaknesses.
- Estimating whether your recovery time objective (RTO) and recovery point objective (RPO) can be met.
- Resolving issues before they are released into production.

This section guides you through adding an application. You gather resources from an existing application, AWS CloudFormation stacks, resource groups, or AppRegistry and create an appropriate resiliency policy. After describing an application, you can publish it in Resilience Hub, and generate an assessment report on the resiliency of your application. You can then use recommendations from the assessment to improve resiliency. You can run another assessment, compare results, and then iterate then iterate until RTO & RPO estimations achieve your RTO & RPO goals.

**Topics**

# Get started by adding an application

Get started with AWS Resilience Hub by describing the details of your existing AWS application and running a report to assess resiliency.

> **Note**
> For more information about cost and billing associated with AWS Resilience Hub, see AWS Resilience Hub pricing.

**To get started**

- On the AWS Resilience Hub home page under **Get started**, choose **Add application**.

## Next

Next (p. 9)

# Step 1: Discover the structure and describe your Resilience Hub application

This section discusses the following methods that you use to form the basis of your application structure:

- CloudFormation stacks
- Resource Groups
- AppRegistry
- Terraform state files
- An existing AWS Resilience Hub application

## CloudFormation

You can add up to 20 CloudFormation stacks.

Choose the CloudFormation stacks that contain the resources you want to use in the application you're describing. The stacks can be from the AWS account that you are using to describe the application, or they can be from different accounts or different Regions.

**To discover the resources that form the basis of your application structure**

1. Select **Start with CloudFormation stacks** to discover your stack-based resources.
2. Choose stacks from **Select stacks** that are associated with your AWS account and Region.

   To use stacks that are in a different AWS account or different Region, enter the Amazon Resource Name (ARN) of the stack in the **Stack ARN** box, and then choose **Stack ARN**. For more information about ARNs, see Amazon Resource Names (ARNs) in the *AWS General Reference*.
3. Enter a name for every stack that you add.

4. Choose **Next**.

## Terraform

You can use up to five Terraform state files.

Choose the Terraform state file that contains your S3 bucket resources that you want to use in the application you're describing. You can navigate to the location of your AWS Terraform state file or provide a link to a Terraform state file you have access to that's located in a different Region.

**To discover the resources that form the basis of your application structure**

1. Select **Start with Terraform state files** to discover your S3 bucket resources.
2. Choose **Browse S3** to navigate to the location of your AWS Terraform state file.

   To use AWS Terraform state files located in a different Region, provide the link to the location of the state file in the **S3 URL** field.

   **Note**
   The limit for Terraform state files is 4 megabytes (MB).
3. Select your S3 buckets from the **Terraform state file s3** screen and choose **Choose**.
4. Choose **Next**.

## Resource groups

Choose the resource groups that contain the resources that you want to use in the application that you're describing. You can use up to five resource groups.

**To discover the resources that form the basis of your application structure**

1. Select **Start with Resource Groups** to discover your group-based resources.
2. Choose resources from **Select resource groups**. You can also add **Resource Group ARNs**.
3. Choose **Next**.

## AppRegistry

Choose the AppRegistry applications that contain the resources you want to use in the application that you're describing. You can add only one AppRegistry application at a time.

**To discover the resources that form the basis of your application structure**

1. Select **Start with AppRegistry** to select from a list of applications created in AppRegistry.
2. Choose applications from **Select application** that were created in AppRegistry.
3. Choose **Next**.

## Existing application

To get started, use an existing application.

**To discover the resources that form the basis of your application structure**

1. Select **Start with an existing application** to build off an existing structure.
2. Choose **Next**.

## Next step

Step 2: Describe the details of your application in Resilience Hub (p. 11)

# Step 2: Describe the details of your application in Resilience Hub

This section shows you how to describe the details of your existing AWS application in AWS Resilience Hub.

**To describe the details of your application**

1. Enter a name for the application.
2. (Optional) Enter a description for the application.
3. Make sure that your name and description are what you want.

   Choose **Next**.

## Next step

Next step (p. 11)

# Step 3: Schedule assessment

Let Resilience Hub run a daily assessment of your application or turn off this setting and manually run the assessment on your own schedule. When enabled, the daily assessment schedule begins only after the application is manually assessed successfully for the first time and if the `AwsResilienceHubPeriodicAssessmentRole` IAM role is created. For more information, see AWS Resilience Hub policy examples (p. 45).

> **Note**
> Daily assessments can have an impact on your quota for runs. For more information about quotas, see AWS Resilience Hub endpoints and quotas in the *AWS General Reference*.

**To stop your application from running daily assessments**

- (Optional) Use the toggle to turn off the recommended daily assessment schedule for your application.

## Next step

Next step (p. 11)

# Step 4: Add tags

Assign a tag or label to an AWS resource to search and filter your resources, or track your AWS costs.

**To add tags to your application**

- (Optional) Choose **Add new tag** if you want to associate one or more tags with the application. For more information about tags, see Tagging resources in the *AWS General Reference*.

## Next step

Next step (p. 12)

# Step 5: Review resources for your Resilience Hub application

You should identify the resources in your application to make sure that it contains the ones that you want. You can add resources that are missing, or remove resources that you don't need. The assessment reports, validation, and recommendations are based on the listed resources.

Resources are grouped into logical application components. You can edit the application components to better reflect the structure of your application. Editing the resources modifies only the Resilience Hub reference of your application. No changes are made to your actual resources, the AWS CloudFormation stacks, or the AWS Terraform state files that contain the resources.

**To identify application resources**

1. The resources from the CloudFormation stacks, manually added application, AppRegistry, or resource groups that you chose for your application description are listed under **Resources**. You can identify them by the following:

   - **Logical ID** – A logical ID is a name used to identify resources in your CloudFormation stack, Terraform state file, manually added application, AppRegistry, or resource groups.

     **Note**
     Terraform lets you use the same name for different resource types. Therefore, you see "-*resource type*" at the end of the logical ID for resources that share the same name.
   - **Resource type** – The resource type identifies the component resource for your application. For example, `AWS::EC2::Instance` declares an Amazon EC2 instance. For more information about application component resources, see AWS Resilience Hub supported resources (p. 5).
   - **Source stack** – The stack (lists all the stacks including CloudFormation stacks) that contains the resource. This column depends on the type of application structure that you have selected.
   - **Physical ID** – The actual assigned identifier for that resource, such as an EC2 instance ID or an S3 bucket name.
   - **Included** – This indicates whether or not AWS Resilience Hub includes these resources in the application.
   - **Assessable** – This indicates whether or not the AWS Resilience Hub will assess your resource for resiliency.
   - **Name** – A name for the resource that you can add and edit.
   - **Component name** – The AWS Resilience Hub component that was assigned to this resource when its application structure was discovered.
   - **CloudFormation stack** – The CloudFormation stack that contains the resource. This column depends on the type of application structure that you selected.
2. To find a resource that is not listed under **Resources**, enter the logical ID for the resource in the search box.
3. To remove a resource from your application, select the resource, and then choose **Exclude resource**.

   When prompted, choose **Exclude** to remove the resource from your application.

   To see the list of excluded resources, choose the **Exclude resources** tab.

   **Note**
   You cannot import an excluded resource until the exclude for the resource is removed.

4. Choose **Next**.

## Next step

Next step (p. 13)

# Step 6: Select a policy for your application

A Resilience Hub resiliency policy contains information and objectives that are used to assess whether your application can recover from a disruption, such as an application or infrastructure disruption. When you create a resiliency policy, you define the recovery time objective (RTO) and recovery point objective (RPO) for the disruption types. These objectives determine whether the application meets the resiliency policy.

When you create a new policy or select an existing Resilience Hub resiliency policy, consider your resiliency objectives and potential disruptions. For example, consider the business impact of the application, and the RTO and RPO targets that you want your application to meet. Also, identify the most concerning disruptions that your application might encounter since you can set RTO and RPO goals for each disruption type.

Use a suggested policy or create a new policy to get started.

### To create a resiliency policy

1. Select **Create a new policy**.
2. Enter a name for the policy.
3. (Optional) Enter a description for the policy.
4. Choose one of the following from **Tier**:

   - **Foundational IT core services**
   - **Mission critical**
   - **Critical**
   - **Important**
   - **Non critical**
5. Under **Customer Application RTO and RPO**, enter a numeric value in the box and then choose the unit of time that the value represents, for both **RTO** and **RPO**.

   Repeat these entries under **Infrastructure RTO and RPO** for **Infrastructure** and **Availability Zone**.
6. (Optional) If you have a multi-Region application, you may want to define a Region RTO and RPO.

   Under **Region - Optional** enter a numeric value in the box and then choose the unit of time that the value represents, for both **RTO** and **RPO**.
7. (Optional) If you want to add tags, you can do that later as you continue creating your policy. For more information about tags, see Tagging resources in the *AWS General Reference*.
8. To create the policy, choose **Create**.
9. Verify that your newly created policy is selected by default from the list of policies under **Resiliency policies**, and choose **Next**.

### To use a suggested resiliency policy

1. Enter a name for the resiliency policy.
2. (Optional) Enter a description for the policy.

3. Under **Suggested resiliency policies**, view and choose one of the following predetermined resiliency policy tiers:

   - **Non-critical application**
   - **Important Application**
   - **Critical Application**
   - **Global Critical Application**
   - **Mission Critical Application**
   - **Global Mission Critical Application**
   - **Foundational Core Service**
4. To create the resiliency policy, choose **Create policy**.
5. Verify that your newly created policy is selected by default from the list of policies under **Resiliency policies**, and choose **Next**.

## Next step

Next step (p. 14)

# Step 7: Review and publish your Resilience Hub application

After you set up your application, identify your resources, and select your resiliency policy, you can review and publish your AWS Resilience Hub application.

**To review and publish your application**

1. Review all the information that you entered in the previous steps.

   If you see information that you need to change, or if you want to turn the scheduled assessment feature on or off, choose **Edit**. After you make an edit, choose **Next** for each step until you get back to Step 7: Review and publish.
2. After you finish your review, choose **Publish**.

## Next Step

Next Step (p. 14)

# Step 8: Run an assessment of your Resilience Hub application

The application that you published is listed on the **Summary** page.

After you publish your AWS Resilience Hub application, you are redirected to the application summary page where you can run a resiliency assessment. The assessment evaluates your application configuration against the resiliency policy that is attached to your application. An assessment report is generated that shows how your application measures against the objectives in your resiliency policy.

**To run a resiliency assessment**

1. On the **Applications summary** page, choose **Assess resiliency**.

2. Under **Report name**, enter a unique name for the report or use the generated name.
3. Choose **Run**.
4. After you are notified that the assessment report has been generated, choose the **Assessment** tab and your assessment to view the report.
5. Choose the **Review** tab to your application's assessment report.

## Next step

Next step (p. 15)

# Step 9: Review the application and operational recommendations for your Resilience Hub application

Review the resiliency and operational recommendations for the application that you published from the **Review** page. This page displays the application assessment overview, RTO and RPO summary, and disruption type details, as follows:

- **Overview** - The overview section contains information such as the application name, attached policy name, assessment ARN, and the assessment creation date.
- **RTO Summary** - The RTO summary displays the targeted RTO time against the estimated time assessed.
- **RPO Summary** - The RPO summary displays the targeted RPO time against the estimated time assessed.
- **Details** - The details section lists the disruption type, application component, and the estimated RTO and RPO times tested against the attached policy configurations.

To improve resiliency, you can update your application and resiliency policy according to the recommendations from the report. Then, run another assessment, compare results, and reiterate the process until your estimated RTO and RPO meets your goals for RTO and RPO.

### To view application recommendations

1. On the **Review** page, choose **Application recommendations**.
2. On the **Application recommendations** tab, choose a component from the **Components** section to view application recommendations.

The **Application recommendations** displays recommendation information for optimizing for RTO and RPO (AZ), cost, and minimal changes.

- **Optimize for Availability Zone (AZ) RTO/RPO** - This section provides the lowest possible estimated RTO and RPO during an AZ disruption.
- **Optimize for region RTO/RPO** - This section provides the lowest possible estimated RTO and RPO during a regional disruption.
- **Optimize for cost** - This section provides the lowest cost that you can consume and still meet your policy.
- **Optimize for minimal changes** - This section provides how to achieve your policy limit and keep implementation changes minimal.

# Using Resilience Hub

AWS Resilience Hub helps you improve the resiliency of your applications on AWS and reduce the recovery time in the event of application outages.

To use Resilience Hub, you:

- Describe your AWS applications in Resilience Hub.
- Manage your AWS resources in Resilience Hub.
- Create effective resiliency policies.
- Manage assessments that indicate the resiliency of your applications.
- Manage alarms, standard operating procedures (SOPs), and tests for your applications.

## Describing and managing AWS Resilience Hub Applications

An AWS Resilience Hub application is a collection of AWS resources structured to prevent and recover AWS application disruptions.

To describe an AWS Resilience Hub application, you provide an application name, resources from one or more–up to five–AWS CloudFormation stacks, and an appropriate resiliency policy. You can also use any existing AWS Resilience Hub application as a template to describe your application.

After you describe a AWS Resilience Hub application, you publish it so that you can run a resiliency assessment on it. You can then use recommendations from the assessment to improve resiliency by running another assessment, comparing results, and then reiterating the process until your estimated RTO and RPO meet your RTO and RPO goals.

The following topics show the different approaches for describing a AWS Resilience Hub application and how to manage them.

**Topics**



### Editing AWS Resilience Hub application resources

To receive accurate and helpful resiliency assessments, make sure that your application description is current and matches your actual AWS application and resources. Assessment reports, validation, and recommendations are based on the listed resources. If you add or remove resources from an AWS application, then you should reflect those changes exactly in AWS Resilience Hub.

You can identify and edit the resources in your application. Editing the resources modifies only the AWS Resilience Hub reference of your application. No changes are made to your actual resources or AWS CloudFormation stacks.

You can add resources that are missing, or remove resources that you don't need. Resources are grouped into logical application components. You can edit the application components to better reflect the structure of your application.

Add to or update your AWS Resilience Hub application resources by editing a draft version of your application. Make changes to the draft version and then publish a new version (a release version), which is the version that's assessed when you run resiliency assessments.

### To edit application resources

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the name of the application that you want to edit.
3. Choose the **Versions** tab.
4. Under **Versions**, select **draft**, if it's not already selected.
5. The resources from the application you chose to use as a template for your application description are listed under the **Resources** tab. You can identify the resources by the following:

   - **Logical ID** – A logical ID is a name used to identify resources in your CloudFormation stack, Terraform state file, manually added application, AppRegistry, or resource groups.

     **Note**
     AWS Terraform lets you to use the same name for different resource types. Therefore, you see "*- resource type*" at the end of the logical ID for resources that share the same name.
   - **Resource type** – The resource type identifies the component resource for your application. For example, `AWS::EC2::Instance` declares an Amazon EC2 instance. For more information about application component resources, see AppComponent grouping, see Grouping resources in an AppComponent (p. 18).
   - **Source stack** – The CloudFormation stack that contains the resource. This column depends on the type of application structure that you selected.
   - **Physical ID** – The actual assigned identifier for that resource, such as an Amazon EC2 instance ID or an S3 bucket name.
   - **Included** - This indicates whether or not AWS Resilience Hub includes these resources in the application.
   - **Assessable** – This indicates whether or not the AWS Resilience Hub will assess your resource for resiliency.
   - **Component name** – The AWS Resilience Hub component that was assigned to this resource when its application structure was discovered.
6. To find a resource that is not listed, enter the logical ID for the resource in the search box.
7. To remove a resource from your application, select the resource, and then choose **Exclude** from **Edit**.

   To see the list of excluded resources, choose the **Excluded resources** tab.
8. To add a resource to your application, from **Actions**, choose **Add resource**.
9. To resolve resources on your application, from **Actions**, choose **Resolve resources**.
10. To update stacks on your application, from **Actions**, choose **Update Stacks**.
11. If a CloudFormation stack associated with your application changes, you can reimport the stack. All new resources to the stack are imported, except for resources that are currently excluded from AWS Resilience Hub.

    To reimport CloudFormation stacks, choose **Update stacks**.

    a. In **Select stacks**, select stacks that are associated with your AWS account and Region.

       To use stacks that are in a different AWS account or Region, enter the Amazon Resource Name (ARN) of the stack in the **Stack ARN** box and then choose **Add stack ARN**. For more information about ARNs, see Amazon Resource Names (ARNs) in the *AWS General Reference*.

b. Choose **Update**.

12. If a Terraform state file associated with your application changed, you can reimport the S3 bucket. All new resources to the state file are imported, except for resources that are currently excluded from AWS Resilience Hub.

    To reimport Terraform state files, choose **Update stacks**.

    a. Choose **Browse S3** to navigate to the location of your AWS Terraform state file.

       To use AWS Terraform state files located in a different Region, provide the link to the location of the state file in the **S3 URL** field.

       **Note**
       The limit for Terraform state files is 4 megabytes (MB).

    b. Select your S3 buckets from the **Terraform state file s3** screen.

    c. Choose **Update**.

13. To view the logical components that the resources are grouped into, choose the **Components** tab.

    Under the **Components** tab, you can add new components, rename a component, or delete a component by using the **Actions** menu.

After you make changes to your resource list, you receive an alert that indicates changes have been made to the draft version of your application. To run an accurate resiliency assessment, you must publish a new version of your application. For information about how to publish a new version, see Publishing a new AWS Resilience Hub application version (p. 22).

# Grouping resources in an AppComponent

An AppComponent is a group of related AWS resources that work and fail as a single unit. For example, if you have a primary and replica database, then both databases belong to the same application component. AWS Resilience Hub has rules governing which AWS resources can belong to which type of application component. For example, a `DBInstance` can belong to `AWS::ResilienceHub::DatabaseAppComponent` but not to `AWS::ResilienceHub::ComputeAppComponent`.

When a CloudFormation stack, Terraform state file, resource group, or AppRegistry application is imported into AWS Resilience Hub, it makes its best effort to group related resources into the same application component, but might not always be 100 percent accurate. You know the architecture of your application the best, so you can regroup resources that have already been grouped by Resilience Hub into a different AppComponent. For example, if you have three EC2 instances in an AWS CloudFormation stack, AWS Resilience Hub creates a single application component per EC2 instance, but all three EC2 instances might be running the same application software. In this case, the correct choice is to regroup the three EC2 instances under a single `ComputeAppComponent`. When regrouping resources, you should only regroup the resource to a single AppComponent. You can also expand your resource list and combine ungrouped resources into an AppComponent.

The AWS Resilience Hub application components supports the following resources:

- `AWS::ResilienceHub::ComputeAppComponent`
  - `AWS::ApiGateway::RestApi`
  - `AWS::ApiGatewayV2::Api`
  - `AWS::AutoScaling::AutoScalingGroup`
  - `AWS::EC2::Instance`
  - `AWS::ECS::Service`
  - `AWS::Lambda::Function`

- `AWS::ResilienceHub::DatabaseAppComponent`
  - `AWS::DocDB::DBCluster`
  - `AWS::DynamoDB::Table`
  - `AWS::RDS::DBCluster`
  - `AWS::RDS::DBInstance`
- `AWS::ResilienceHub::NetworkingAppComponent`
  - `AWS::EC2::NatGateway`
  - `AWS::ElasticLoadBalancing::LoadBalancer`
  - `AWS::ElasticLoadBalancingV2::LoadBalancer`
  - `AWS::Route53::RecordSet`
- `AWS:ResilienceHub::NotificationAppComponent`
  - `AWS::SNS::Topic`
- `AWS::ResilienceHub::QueueAppComponent`
  - `AWS::SQS::Queue`
- `AWS::ResilienceHub::StorageAppComponent`
  - `AWS::Backup::BackupPlan`
  - `AWS::EC2::Volume`
  - `AWS::EFS::FileSystem`
  - `AWS::S3::Bucket`

The following are examples of correct groupings:

- Group primary databases and replicas under a single application component.
- Group an Amazon S3 bucket and its replication under a single application component.
- Group Amazon EC2 instances that run the same application under a single application component.
- Group an Amazon SQS queue and its dead-letter queue under a single application component.
- Group Amazon ECS services in one region and failover Amazon ECS services in another region under a single application component.

  **Note**
  AWS Resilience Hub requires the correct grouping so that it can compute RTO and RPO estimates to generate recommendations.

### To assign resources to an AppComponent

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the application that contains the resource that you want to regroup.
3. Choose the **Versions** tab.
4. Under **Versions**, choose **draft**, if it's not already selected.
5. The resources from the application you choose are listed under the **Resources** tab. Choose the **Resources** tab.
6. Choose the resource you want to regroup.
7. Choose **Actions**, and then choose **Change component**. A **Change component** window displays.
8. Delete the component under which the resource is currently grouped from the **Component** section by choosing the empty component and then choosing **Actions** and then **Delete component**.
9. Choose a different component to group the resource in from the **Choose component** menu.
10. Choose **Add**.

11. Delete any empty components from the **Components** tab.
12. Choose **Publish new version**.
13. Under **Versions**, choose **release**.

### To group resources

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the application that contains the resources that you want to group.
3. Choose the **Versions** tab.
4. Under **Versions**, choose **draft**, if it's not already selected.
5. The resources from the application you choose are listed under the **Resources** tab. Choose the **Resources** tab.
6. Choose the resources you want to group.
7. Choose **Actions**, and then choose **Combine resources**. A **Combine component** window displays.
8. Choose a component in which to group the resource from the **Choose component** menu.
9. Choose **Save**.
10. Choose **Publish new version**.
11. Under **Versions**, choose **release**.

# Viewing a Resilience Hub application summary

The application summary page in the AWS Resilience Hub console provides an overview of your application information and resiliency health.

### To view an application summary

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the name of the application.

The applications summary page has the following sections.

**Topics**



## Details

The application summary **Details** section shows a summary of the selections for the application.

- **Resiliency policy** - Shows the name of the resiliency policy attached to your application. For more information about resiliency policies, see Managing resiliency policies (p. 22).
- **Description** – The description of the application.

- **Status** – Indicates if the policy is active or inactive.
- **Creation time** – The date and time that the application was created.
- **Version** – Indicates whether the application is released or in draft.
- **Scheduled assessment** - Indicates whether the daily assessment is active or inactive.

### To update the scheduled assessment

1. To update the scheduled assessment on your application, from **Actions**, choose **Update scheduled assessment**.
2. In the **Scheduled assessment** dialog box, use the toggle to turn the recommended daily assessment schedule on or off.
3. Choose **Save changes**.

> **Note**
> To activate scheduled assessments on existing applications, you must manually run an assessment after enabling the scheduled assessments feature for the first time. For more information about running assessments, see Running resiliency assessments (p. 26).

## Application resiliency

The metrics shown on the **Application resiliency** section are from the most recent resiliency assessment of the application.

#### Resiliency score

The resiliency score helps you quantify your readiness to handle a potential disruption. This score reflects how closely your application has followed the Resilience Hub recommendations for meeting the application's resiliency policy, alarms, standard operating procedures (SOPs), and tests.

The maximum resiliency score that your application can achieve is 100%. The score represents all recommended tests that run in a predefined period of time. It indicates that the tests are initiating the correct alarm, and that the alarm initiates the correct SOP.

For example, suppose that Resilience Hub recommends one test with one alarm and one SOP. When the test runs, the alarm initiates the associated SOP, and then runs successfully. For more information about the resiliency score, see Understanding resiliency scores (p. 30).

#### Resiliency score over time

With the resiliency score over time, you can view a graph of your application's resiliency over the past 30 days. While the dropdown menu can list 10 of your applications, Resilience Hub only shows you a graph of up to four applications at a time. For more information about scheduled assessments, see Next step (p. 8).

> **Note**
> Resilience Hub does not run scheduled assessments at the same time. As a result, you may need to return to the resiliency score over time graph at a later time to view the daily assessment of your applications.

Resilience Hub also uses Amazon CloudWatch to generate these graphs. Choose **View metrics in CloudWatch** to create and view more granular information about your application's resiliency in your CloudWatch dashboard. For more information about CloudWatch, see Using dashboards in the *Amazon CloudWatch User Guide*.

### Alarms

The application summary **Alarms** section lists the alarms that you set up in Amazon CloudWatch to monitor the application. For more information about alarms, see Managing alarms (p. 38).

### Experiments

The application summary **Fault injection experiments** section shows a list of the fault injection experiments. For more information about fault injection experiments, see Fault injection experiments (p. 35).

## Publishing a new AWS Resilience Hub application version

After you make changes to your AWS Resilience Hub application resources as described in Editing AWS Resilience Hub application resources (p. 16), you must publish a new version of your application to run an accurate resiliency assessment. Also, you might need to publish a new version of your application if you added new recommended alarms, SOPs, and tests to your application.

**To publish a new version of an application**

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the name of the application.
3. Choose the **Versions** tab.
4. Choose the **Resources** tab.
5. Choose **Publish new version**. When you publish a new version of your application, this becomes the version that is assessed when you run resiliency assessments.
6. Choose **Publish**.

After you publish a new version of your application, we recommend you to run a new resiliency assessment report to confirm your application still meets your resiliency policy. For information about running an assessment, see Running and managing Resilience Hub resiliency assessments (p. 26).

## Deleting an AWS Resilience Hub application

After you've reached the maximum of ten application limits, you must delete one or more applications before you can add more.

**To delete an application**

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, select the application that you want to delete.
3. Choose **Actions**, and then choose **Delete**.
4. To confirm the deletion, enter **Delete**.

# Managing resiliency policies

This section describes how to create resiliency policies for your applications. Setting resiliency policies correctly enables you to understand your applications resiliency posture. A resiliency policy contains

information and objectives that you use to assess whether your application can recover from a disruption type, such as software, hardware, Availability Zone, or AWS Region. Resiliency policies are guidelines that measure your objectives. These policies do not change or affect an actual application. Multiple applications can have the same resiliency policy.

When you create a resiliency policy, you define the objectives: Recovery Time Objective (RTO) and Recovery Point Objective (RPO). The objectives determine whether the application meets the resiliency policy. Attach the policy to your application and run a resiliency assessment. You can create different policies for the different types of applications in your portfolio. For example, a real-time trading application would have a different resiliency policy than a monthly reporting application.

The assessment evaluates your application configuration against the attached resiliency policy. At the end of the process,AWS Resilience Hub provides an assessment of how your application measures against the objectives in your resiliency policy.

You can create resiliency policies in Applications, and also in Resiliency policies. You can access relevant details about your policies, and also modify and delete them.

AWS Resilience Hub uses your RTO and RPO objectives to measure resiliency for these potential types of disruptions:

- **Application** – Loss of a required software service or process.
- **Cloud infrastructure** – Loss of hardware, such as EC2 instances.
- **Cloud infrastructure Availability Zone (AZ)** – One or more Availability Zones are unavailable.
- **Cloud infrastructure Region** – One or more Regions are unavailable.

AWS Resilience Hub enables you to create customized resiliency policies or use our recommended, open standard resiliency policies. When you create customized policies, name and describe your policy and choose the appropriate level or tier that defines your policy. These tiers include: Foundational IT core services, Mission critical, Critical, Important, and Non-critical.

Choose the tier that is appropriate for your class of application. For example, you might classify a real-time trading system as mission critical, while you might classify a monthly reporting application as non-critical. When you use our standard policies, you can choose a resiliency policy with a preconfigured tier and values for RTO and RPO objectives by disruption type. If necessary, you can change the tier and RTO and RPO values.

You can create resiliency policies in Resiliency policies, or when you describe a new application.

# Creating resiliency policies

In AWS Resilience Hub, you can create a resiliency policy. A resiliency policy contains information and objectives that you use to assess whether your application can recover from a disruption type, such as software, hardware, Availability Zone, or AWS Region. Resiliency policies are guidelines that measure your objectives. These policies do not change or affect an actual application. Multiple applications can have the same resiliency policy.

When you create a resiliency policy, you define the objectives: Recovery Time Objective (RTO) and Recovery Point Objective (RPO). The objectives determine whether the application meets the resiliency policy. Attach the policy to your application and run a resiliency assessment.

The assessment evaluates your application configuration against the attached resiliency policy. At the end of the process, AWS Resilience Hub provides an assessment of how your application measures against the objectives in your resiliency policy.

You can create resiliency policies in Applications, and also in Resiliency policies. You can access relevant details about your policies, and also modify and delete them.

### To create resiliency policies in Applications

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, choose **Add Application**. Then choose either **Quick Start** or **Walk-through** (guided instructions). Based on your selection, proceed to step 3 or step 4.
3. If you choose **Quick Start**, enter a name and optional description, and then choose **Add**. You can add resources and resiliency policies later.
4. If you choose **Walk-through**:
   - In **Describe application details**, enter the name and an optional description. Choose **Next**.
   - Specify how your application discovers resources from either existing applications, or AWS CloudFormation stacks.
   - In **Resiliency policies**, choose **Create a new policy**.
     - If you know how you want to set up your resiliency policy, choose **Create a policy**.
       - Name and describe the policy and select the tier that defines the policy.
       - Enter RTO and RPO values for Hardware disruption, Software disruption, Availability Zone disruption, and Region disruption (Region is optional).
       - Choose **Create** to complete the process.
     - If you need recommendations to set up your resiliency policy, choose **Select a policy from suggestions**.
       - Name and describe the policy.
       - Choose one of the following resiliency policies. You can get any details about the policy later.

         Non-Critical Application, Important Application Tier, Critical Application Tier, Global Critical Application Tier, Mission Critical Application Tier, Global Mission Critical Application Tier, and Foundational Core Service Tier. You can get details about the policy later.
       - Choose **Create** to complete the process.

### To create resiliency policies in Resiliency policies

1. In the left navigation menu, choose **Resiliency policies**.
2. In **Resiliency policies**, choose **Create a new policy**.
   - Name and describe the policy and select the tier that defines the policy.
   - Enter RTO and RPO values for Hardware disruption, Software disruption, Availability Zone disruption, and Region disruption (optional).
   - Choose **Create** to complete the process.
3. You can add internal tags to search, filter, and manage your AWS resources in your application.
   - To add tags, choose **Add new tag**.
   - Enter information in the Key and Value fields.

### To create resiliency policies based on a suggested policy

1. In the left navigation menu, choose **Resiliency policies**.
2. In **Resiliency policies**, choose **Select a policy based on a suggested policy**.
   - Name and describe the policy.
   - Choose one of these resiliency policies:

Non-Critical Application, Important Application Tier, Critical Application Tier, Global Critical Application Tier, Mission Critical Application Tier, Global Mission Critical Application Tier, and Foundational Core Service Tier. You can get details about the policy later.

- Input the Customer application RTO and RPO targets.
- Input the Cloud Infrastructure RTO and RPO targets.
- Choose **Create** to complete the process.

3. You can add internal tags to search, filter, and manage your AWS resources in your application.

   - To add tags, choose **Add new tag**.
   - Enter information in the Key and Value fields.

# Accessing resiliency policy details

When you open a resiliency policy, you see important details about the policy. You can also edit or delete the resiliency.

Resiliency policy details consist of two major views: **Summary** and **Tags**.

**Summary**

*Basic information*

Provides the following information about resiliency policy: Name, Description, Tier, Cost Tier, and Date Created.

*RTO and RPO*

Shows the RTO and RPO disruption type associated with this resiliency policy.

**Tags**

Use this view to manage, add, and delete tags internal to this application.

### To edit resiliency policies in Resiliency policy details

1. In the left navigation menu, choose **Policies**.
2. In **Resiliency policies**, open a resiliency policy.
3. Choose **Edit**. Enter appropriate changes to **Basic Info** and **RTO** and **RPO** fields. Then choose **Save changes**.

### To edit resiliency policies in Resiliency policy

1. In the left navigation menu, choose **Policies**.
2. In **Resiliency policies**, choose a resiliency policy.
3. Choose **Actions**, and then select **Edit**.
4. Enter appropriate changes to **Basic Info** and **RTO** and **RPO** fields. Then choose **Save changes**.

### To delete resiliency policies in resiliency policy details

1. In the left navigation menu, choose **Policies**.
2. In **Resiliency policies**, open a resiliency policy.
3. Choose **Delete**. Confirm your deletion, and then choose **Delete**.

**To delete resiliency policies in resiliency policy**

1. In the left navigation menu, choose **Policies**.
2. In **Resiliency policies**, choose a resiliency policy.
3. Choose **Actions**, and then select **Delete**.
4. Confirm your deletion, and then choose **Delete**.

# Running and managing Resilience Hub resiliency assessments

When your application changes, you should run a resiliency assessment. The assessment compares each application component configuration to the policy and makes alarm, SOP, and test recommendations. These configuration recommendations can improve the speed of recovery procedures.

Alarm recommendations help you set alarms that detect outages. SOP recommendations provide scripts that manage common recovery processes, such as recovery from a backup. Test recommendations offer suggestions to verify your configurations work properly. For example, you can test whether an application recovers during automatic recovery processes, such as automatic scaling or load balancing because of network issues. You can test whether application alarms are triggered when resources reach their limits. You can also test how well SOPs work under the conditions that you indicate.

## Running resiliency assessments

You can run a resiliency assessment report from your application's **Actions** menu, the **Assessments** view, or in the **Get started** banner on the **Application** page. You can identify and filter your applications from the Applications menu by the following:

- **Name** – The name you assigned to the application when adding it to Resilience Hub.
- **Compliance status** – Resilience Hub sets the application status as **Assessed**, **Not assessed**, **Policy breached**, or is **Changes detected**.
  - **Assessed** - Resilience Hub has assessed your application.
  - **Not assessed** - Resilience Hub has not assesses your application.
  - **Policy breached** - Resilience Hub has determined your application did not meet your resiliency policy's objectives for Recovery Time Objective (RTO) and Recovery Point Objective (RPO). Review and use the recommendations provided by Resilience Hub before reassessing your application for resiliency. For more information about recommendations, see Next step (p. 8).
  - **Changes detected** - Resilience Hub has detected changes made to the resiliency policy associated with your application. You must reassess your application for Resilience Hub to determine if your application meets your resiliency policy's objectives.
- **Scheduled assessments** – The resource type identifies the component resource for your application. For more information about scheduled assessments, see Application resiliency (p. 20).
  - **Active** - This indicates your application is automatically assessed daily by Resilience Hub.
  - **Disabled** - This indicates your application is not automatically assessed daily by Resilience Hub and you must manually assess your application.
- **Creation time** – The date and time that the application was created.

- **ARN** – The Amazon Resource Name (ARN) of the stack associated to your application. For more information about ARNs, see Amazon Resource Names (ARNs) in the *AWS General Reference*.

  **Note**
  Resilience Hub can fully assess the resiliency of cross-region Amazon ECS resources only if you are using Amazon ECR for the image repository.

### To run a resiliency assessment from the Actions menu

1. In the left navigation menu, choose **Applications**.
2. Choose your **Application**.
3. Choose the **Run resiliency assessment** from the **Actions** menu.
4. Enter a unique name or use the generated name.
5. Choose **Run**.

   To review the assessment report, choose **Assessments** in your application. For more information, see the section called “Reviewing assessments reports” (p. 27).

### To run a resiliency assessment from the Get started banner

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application. In the **Get started** banner in the middle of the application page, choose **Resiliency assessed**.
3. In **Run resiliency assessment**, enter a unique name or use the generated name, and then choose **Run**.

   To review the assessment report, choose **Assessments** in your application. For more information, see the section called “Reviewing assessments reports” (p. 27).

### To run a resiliency assessment from the Assessments view

You can run a new resiliency assessment when your application or resiliency policy changes.

1. In the left navigation menu, choose **Applications**.
2. Choose your application under **Application**.
3. Choose the **Assessment** tab.
4. In the **Assessments**, choose **Run resiliency assessment**. Enter a unique name or use the generated name.
5. Choose **Run**.

   To review the assessment report, choose **Assessments** in your application. For more information, see the section called “Reviewing assessments reports” (p. 27).

# Reviewing assessments reports

You find assessment reports in the **Assessments** view of your application.

### To find an assessment report

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application.

3. In **Assessments**, open an assessment report in the **Resiliency assessments** table.

When you open the report, you see the following:

- An overall overview of the assessment report
- Recommendations to improve resiliency
- Recommendations to set up alarms, SOPs, and tests
- How to create and manage tags to search and filter your AWS resources

## Review

This section provides an overview of the assessment report. AWS Resilience Hub lists each disruption type and the associated application component. It also lists your actual RTO and RPO policies and determines whether the application component can achieve the policy goals.

**Overview**

Shows the name of the application, the name of the resiliency policy, and the creation date of the report.

**RTO summary**

Shows a graphical representation of whether the application meets resiliency policies objectives. This is based on the amount of time that an application can be down without causing significant damage to the organization. The assessment is an estimate of the expected RTO.

**RPO summary**

Shows a graphical representation of whether the application meets resiliency policies objectives. This is based on the amount of time that data can be lost before a significant harm to the business occurs. The assessment is an estimate of the expected RPO.

**Details**

Provides detailed descriptions of each disruption type (application, infrastructure, Availability Zone, and Region), and provides the following information about it:

- **Component**

  The resources that comprise the application. For example, your application might have a database or compute component.

- **Actual RTO (Policy RTO)**

  Indicates whether your policy configuration aligns with your policy requirement. We provide two values, our RTO estimate and your current RTO. For example, suppose you see this value for Actual RTO policy: 40 min (2 hours). Here we estimate an RTO of 40 minutes, while your current RTO is two hours. We base our RTO calculation on the configuration, not the policy. As a result, a multi-Availability Zone database will have the same RTO for Availability Zone failure, no matter which policy you select.

- **Actual RPO (Policy RPO**

  Shows the actual RPO policy that AWS Resilience Hub estimates, based on the RPO policy that you set for each application component. For example, you might have set the policy RPO for Availability Zone failures to one hour. The actual result might be calculated to zero. This assumes that Aurora, where we commit every transaction, is successful in four out of six nodes, spanning multiple Availability Zones. It might be five minutes for point-in-time restore.

The only RTO and RPO that you can opt not to supply is Region. For some applications, it is useful to plan for recovery when there is a crucial dependency on an AWS service, which might become unavailable in the entire Region.

If you choose this option, such as setting RTO or RPO targets for the Region, you'll receive an estimated recovery time and operational recommendations for such failures.

## Reviewing resiliency recommendations

Resiliency recommendations evaluate application components and recommend how to optimize by RTO and RPO, costs, and minimal changes.

With AWS Resilience Hub, you can optimize resiliency using one of the following recommended options in **Why you should choose this option**:

**Note**

- AWS Resilience Hub provides up to three AWS Resilience Hub recommended options.
- If you set regional RTO and RPO, AWS Resilience Hub displays **Optimize for region RTO/RPO** in the recommended options. If regional RTO and RPO is not set, **Optimize for Availability Zone (AZ) RTO/RPO** is displayed. For more information about setting regional RTO/RPO targets while creating resiliency policies, see Creating resiliency policies (p. 23).
- RTO and RPO values for the applications and their configurations are determined by considering the amount of data and individual AppComponents. However, the RTO and RPO values are only estimates. You should use your own testing (such as AWS Fault Injection Simulator) to test your application for actual recovery times.

**Optimize for Availability Zone (AZ) RTO/RPO**

The lowest possible RTO and RPO during an AZ disruption.

**Optimize for region RTO/RPO**

The lowest possible RTO and RPO during a regional disruption.

**Optimize for cost**

The lowest cost that you can incur and still meet your policy.

**Optimize for minimal changes**

Achieve your policy targets and keep implementation changes minimal.

The following items are included in the optimization category breakdowns:

- **Description**

  Describes the configurations suggested by AWS Resilience Hub.
- **Changes**

  A list of text changes that describe the necessary tasks to switch to the suggested configuration.
- **Base cost**

  The estimated cost associated with the changes to meet your RTO and RPO estimates.

  **Note**
  **Base cost** does not include any discounts or offers from Enterprise Discount Program (EDP).

- **RTO and RPO**

  The estimated RTO and RPO after changes.

## Reviewing operational recommendations

Operational recommendations contain recommendations to set up alarms, SOPs, and AWS FIS experiments through AWS CloudFormation templates.

AWS Resilience Hub provides CloudFormation templates for you to download. AWS Resilience Hub manages applications infrastructure as code. As a result, we supply the recommendations in CloudFormation so that you can add the recommendations to the application code.

You provision the selected alarms, SOPs, and AWS FIS experiments. To provision alarms, SOPs, and AWS FIS experiments, select the appropriate CloudFormation template and enter a unique name. AWS Resilience Hub creates a template based on your selected recommendations. In **Templates**, you can access your created templates through an Amazon Simple Storage Service (Amazon S3) URL.

You can also create and manage tags. You can add tags to an application and see all the tags associated with it. You can also search, add, and remove tags for an application.

# Deleting resiliency assessments

You can delete resiliency assessments in the **Assessments** view of your application.

**To delete a resiliency assessment**

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application.
3. In **Assessments**, choose an assessment report in the **Resiliency assessments** table.
4. To confirm the deletion, choose **Delete**.

   The report no longer appears in the **Resiliency assessments** table.

# Understanding resiliency scores

This section describes how AWS Resilience Hub quantifies application readiness from different disruption scenarios.

AWS Resilience Hub indicates readiness through a resiliency score. This score reflects how closely the application follows our recommendations for meeting the application's resiliency policy, alarms, standard operating procedures (SOPs), and tests. Based on the type of resources the application uses, AWS Resilience Hub recommends alarms, SOPs, and a set of tests for each disruption type.

The top resiliency score is 100 percent. To achieve a top score, all recommended tests must complete in a predefined time period, initiate all the correct alarms, and initiate of all the SOPs attached to them. For example, AWS Resilience Hub recommends one test with one alarm and one SOP. The test runs and fires the alarm and initiates the associated SOP. If they perform successfully and the application meets the resiliency policy, it receives a resiliency score of 100 percent.

**Recommendation types**

AWS Resilience Hub associates recommendations with disruption types. Disruption types include application, infrastructure, Availability Zone, and AWS Region. Some recommendations might apply to

multiple disruption types. For example, a recommendation might be to simulate the Amazon Relational Database Service (Amazon RDS) infrastructure disruption type with a test that reboots the Amazon RDS instance. To improve resiliency scores, you should regularly implement and verify higher priority recommendations.

# Calculating resiliency scores

AWS Resilience Hub computes coverage scores for alarms, SOPs, tests, and meeting the resiliency policy for each combination of application component and disruption type. It then aggregates them based on the weight of the application component and disruption type.

This table presents the formulas AWS Resilience Hub uses to determine the resiliency scores for alarms, SOPs, tests, and meeting the resiliency policy.

**Resiliency score formulas**

| Name | Description | Formula |
|---|---|---|
| AWS Resilience Hub Test coverage (T) | A normalized score (0 -100 percent) based on number of tests that successfully completed out of the number of tests that AWS Resilience Hub recommended. | T = Number of tests run / Total number of tests recommended. |
| Alarms coverage (A) | A normalized score (0 -100 percent) based on number of CloudWatch alarms that were successfully implemented with data, out of the number of CloudWatch alarms that AWS Resilience Hub recommended. | A = Number of alarms implemented / Total number of alarms recommended. |
| SOP coverage (S) | A normalized score (0 -100 percent) based on number of SOPs (manual or automated) that were successfully tested using AWS Resilience Hub Tests out of the number of SOPs that AWS Resilience Hub recommended. | S = Number of testable SOPs initiated / Total number of testable SOPs recommended. |
| Resiliency policy (P) | A normalized score (0 -100 percent) based on the application meeting its resiliency policy. | P = Total weights of disruption types meeting the resiliency policy / Total weights of all disruption types. |

# Calculating application component level and disruption types

This section explains how we aggregate the recommendation type score for alarms (A), SOPs (S), tests (T), and meeting resiliency policy (P) to calculate the resiliency score for application components and applications.

• Resiliency score per application component per disruption type, `RSao = T * Weight(T) + A * Weight(A) + S * Weight(S) + P * Weight(P)`. Additionally, the Resiliency Score per application component per disruption type is `RSao = Weighted Average (T, M, S, P)`.

• Resiliency score per application component, `RSa = RSao(Application) * Weight(Application) + RSao(Infrastructure) * Weight(Infrastructure) + RSao(AZ) * Weight(AZ) + RSao(Region) * Weight(Region)`.

• Resiliency score for application, `RS = SUM(RSao * Weight of corresponding disruption type)/SUM(Weight of corresponding disruption type)`. Additionally, the resiliency score for application is `RS = SUM(RSao * Weight of corresponding application component per disruption type)/SUM(Weight of corresponding application component per disruption type)`.

# Weight tables

AWS Resilience Hub assigns a weight to each recommendation type for the total resiliency score.

These tables show the weight for alarms, SOPs, tests, meeting resiliency policy, and disruption types.

**Weights for alarms, SOPs, tests, policy target**

| Recommendation type | Weight |
|---|---|
| Alarms | 20 percent |
| SOPs | 20 percent |
| Tests | 20 percent |
| Meeting resiliency policy | 40 percent |

**Weight for disruption type**

| Recommendation type | Weight |
|---|---|
| Region | 10 percent |
| Availability Zone | 20 percent |
| Infrastructure | 30 percent |
| Application | 40 percent |

> **Note**
> If you choose not to define RTO or RPO targets for your policy, the weight for the Region is removed and the weights for the other disruption types are increased by 3.33 percent to equal 100 percent.

# Accessing the resiliency scores

You can see resiliency scores in the dashboard or from applications.

**Accessing the resiliency score from the dashboard**

1. In the left navigation menu, choose **Dashboard**.
2. In **Latest resiliency score**, choose one or more applications in the **Filter applications** dropdown.
3. See the resiliency score for the application.

**Accessing the resiliency score from Applications**

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application.

3. Choose **Summary**. In **Resiliency health**, you can see the resiliency score.

# Standard operating procedures

A standard operating procedure (SOP) is a prescriptive set of steps designed to efficiently recover your application in the event of an outage or alarm. Prepare, test, and measure your SOPs in advance to ensure timely recovery in the event of an operational outage.

Based on your application components, AWS Resilience Hub recommends the SOPs you should prepare. AWS Resilience Hub works with Systems Manager to automate the steps of your SOPs by providing a number of SSM documents you can use as the basis for those SOPs.

For example, AWS Resilience Hub may recommend an SOP for adding disk space based on an existing SSM Automation document. To run this SSM document, you require a specific IAM role with the correct permissions. AWS Resilience Hub creates metadata in your application indicating which SSM automation document to run in the case of disk shortage, and which IAM role is required to run that SSM document. This metadata is then saved in an SSM parameter.

In addition to configuring the SSM automation, it is also best practice to test it with a AWS FIS experiment. Therefore, AWS Resilience Hub also provides an AWS FIS experiment that calls the SSM automation document - this way, you can proactively test your application to make sure the SOP you've created does the intended job.

AWS Resilience Hub provides its recommendations in the form of an AWS CloudFormation template you can add to your application code base. This template provides:

- The IAM role with the permissions required to run the SOP
- An AWS FIS experiment you can use to test the SOP
- An SSM parameter that contains application metadata indicating which SSM document and which IAM role is to be run as the SOP, and on which resource. For example: `$(DocumentName) for SOP $(HandleCrisisA) on $(ResourceA)`.

Creating an SOP may require some trial and error. Running a resiliency assessment against your application and generating an AWS CloudFormation template from the AWS Resilience Hub recommendations is a good start. Use the AWS CloudFormation template to generate an AWS CloudFormation stack, then use the SSM parameters and their default values in your SOP. Run the SOP and see what refinements you need to make.

Because all applications have differing requirements, the default list of SSM documents that AWS Resilience Hub provides will not be sufficient for all of your needs. You can, however, copy the default SSM documents and use them as a basis to create your own custom documents tailored for your application. You can also create your own entirely new SSM documents. If you create your own SSM documents instead of modifying the defaults, you must associate them with SSM parameters, so the correct SSM document is called when the SOP runs.

When you've finalized your SOP by creating the necessary SSM documents and updating the parameter and document associations as necessary, add the SSM documents directly to your code base, and make any subsequent changes or customizations there. That way, every time you deploy your application, you'll also deploy the most up-to-date SOP.

**Topics**

# Building an SOP based on AWS Resilience Hub recommendations

To build an SOP based on AWS Resilience Hub recommendations, you need an AWS Resilience Hub application with a resiliency policy attached to it, and you need to have run a resiliency assessment against that application. The resiliency assessment generates the recommendations for your SOP.

To build an SOP based on AWS Resilience Hub recommendations:

1. Create an AWS CloudFormation template for the SOP:
    a. Open the AWS Resilience Hub console.
    b. In the navigation pane, choose **Applications**.
    c. From the list of applications, choose the application you want to create an SOP for.
    d. If necessary, expand the **Get started** area.
    e. Choose **Set up recommendations**.
    f. Under **Operational recommendations**, choose **Standard operating procedures**.
    g. Select all of the steps you want to include in your SOP.
    h. Choose **Create CloudFormation template**. This can take up to a few minutes to create the template.
2. Consume the AWS Resilience Hub recommendations into your code base:
    a. When the template is created, under **Operational recommendations**, choose **Templates**.
    b. In the list of templates, choose the name of the SOP template you just created.
    c. Under **Template details**, choose the link under **Templates S3 Path** to open the template object in Amazon S3.
    d. Under the list of Objects, choose the SOP folder link.
    e. Select the box in front of the JSON file and either **Open** or **Download** it. The JSON file contains the resources required for the SOP: IAM Role, AWS FIS experiment, and SSM Parameter.
    f. Add these resources to your code base. Replace direct references to application resources with references to the logical names used in your scripts: ("Ref": "LogicalName").

# Creating a custom SSM document

To fully automate the recovery of your application, you may need to create a custom SSM document for your SOP. Create SSM documents in Systems Manager. You can use an existing SSM document as a base and change its content, or you can create a new document entirely.

For detailed information on using Systems Manager to create an SSM document, see Walkthrough: Using Document Builder to create a custom runbook.

For information about SSM document syntax, see SSM document syntax.

For information about automating SSM document actions, see Systems Manager automation actions reference.

# Using a custom SSM document instead of the default

To replace the SSM document AWS Resilience Hub suggested for your SOP with a custom document you've created, work directly in your code base. In addition to adding your new custom SSM automation document, you'll also:

1. Add the IAM permissions required to run the automation.
2. Add a AWS FIS experiment to test your SSM document.
3. Add an SSM parameter that points to the automation document you want to use as the SOP.

Generally, it's most efficient to work with the suggested default values in AWS Resilience Hub and customize them as necessary. For example, add or remove permissions as necessary for the IAM role, change the AWS FIS experiment setup to point to the new SSM document, or change the SSM parameter to point to your new SSM document.

## Testing SOPs

As previously mentioned, best practice is to add AWS FIS experiments to your CI/CD pipelines to test your SOPs regularly; this ensures they're ready to go if an outage occurs.

Test both AWS Resilience Hub-provided and custom SOPs.

# Fault injection experiments

This section describes how to create and run fault injection experiments in AWS Resilience Hub. You run fault injection experiments to measure the resiliency of your AWS resources and the amount of time it takes to recover from application, infrastructure, availability zone, and AWS Region outages.

To measure resiliency, these fault injection experiments simulate outages to your AWS resources. Examples of outages include network unavailable errors, failovers, stopped processes on EC2/ASG, boot recovery in Amazon RDS, and problems with your Availability Zone. When the fault injection experiment concludes, you can determine whether an application can recover from the outage types defined in the RTO of the resiliency policy.

The experiments in Resilience Hub provide AWS Systems Manager (Systems Manager) automation documents that you use to define what experiments you want Systems Manager to perform. The Systems Manager automation documents:

- Implement different failure scenarios.
- Validate alarms when failure happens.
- Validate that the application can recover when the failure scenario completes.

You can use the Systems Manager automation documents in their default state, or customize them based on your requirements. You can access your experiments Systems Manager documents from either the Application fault injection experiments, or the Application assessment report.

For more information about Systems Manager documents, see Systems Manager document syntax and Systems Manager document automation action reference

In the assessment report, choose a recommendation from a list of Resilience Hub experiments recommendations. Then create a AWS CloudFormation template that you can open, and copy the template path.

Use the path in AWS CloudFormation to create a stack that contains Resilience Hub fault injection experiment templates. After creating the stack, open the application to view provisioned fault injection experiment templates.

A Systems Manager document contains the list of steps that comprise the experiment. Each step should run in the specified orders. You can see how each step ran in an Systems Manager document when you view it in your Systems Manager account.

**Topics**



# Running a fault injection experiment

In your application, you must first create an fault injection experiment template and upload the SSM parameter to AWS FIS before Resilience Hub can run the resilience assessment.

### To create an fault injection experiment template

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application, and choose **Experiments**.
3. In **Provisioned view**, choose **Create experiment template**.
4. In **Create experiment template**, enter a name and an optional description. You can optionally add tags for this template.
5. Choose **Save**. Now you can add components to your suite.

### To create an fault injection experiment in Applications

You must provide the Systems Manager document and parameters so that the Systems Manager automation can configure the fault injection experiment.

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application, and choose **Experiment**.
3. In **Provisioned**, open a provisioned test suite.
4. In **Experiments**, choose either **Create** or **Create experiment template**.
5. In **Experiment template details**, enter a name and an optional description. You can optionally enter the account ID and AWS Region that you want to target for the template.
6. In **AWS Systems Manager document**, choose **Enter Systems Manager document name**.
7. In **Document name**, enter the Systems Manager document that contains the template that you want.
8. Choose **Find document**. If the Systems Manager document is found, the Systems Manager document found message displays.
9. You can expand the menu to select the fields that you want to customize your test. There are fields that require you to make an entry.
10. Choose **Save**. The fault injection experiment appears in your test suite.

### Running an fault injection experiment

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application, and choose **Experiments**.
3. Choose an fault injection experiment template, and choose **Execute experiment**.
4. In **Experiments**, review the information, and choose **Execute**.
5. In **Experiments**, you can see your fault injection experiment and its status.

### Viewing experiments

1. In **Experiments**, choose **Executions**.

2. In **Experiment templates**, open an fault injection experiment.

# Creating fault injection experiments from the assessment report

Resilience Hub recommends that you test your application after you run an assessment report. You can access and run these experiments from your application's Assessment report.

Resilience Hub provides a list of fault injection experiments, which are Systems Manager documents with testing parameters. When you select an fault injection experiment from the list, Resilience Hub creates an AWS CloudFormation template with the parameters you define in the Systems Manager document. After the creation of the CloudFormation stack, you can see your provisioned fault injection experiments for your application.

The CloudFormation template consists of an IAM role for each Systems Manager document, with the minimum permissions required to run.

**To create and run an fault injection experiment from the assessment report**

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application, and choose **Assessments**.
3. In **Resiliency assessments**, open an assessment report, and choose **Operational recommendations**.
4. Choose a test, and then choose **Create CloudFormation template**.
5. In **Create CloudFormation template**, enter a name for the fault injection experiment or use a randomly generated name. Resilience Hub displays a message to confirm the template is complete.
6. In **Operational readiness**, choose **Templates**.
7. Copy the URL in the Templates S3 path and create the stack with the template. The fault injection experiment appears in your Systems Manager account.

# Fault injection experiment failures/status check

In the **Actions** list, you can follow the status of an executed action being preformed by the experiment.

**Analyze AWS FIS experiment execution**

After running an AWS FIS experiment, you can view the execution details in the AWS Systems Manager.

1. Go to **CloudTrail** > **Event History**.
2. Filter events by **User name** using the experiment ID.
3. View the StartAutomationExecution entry. **Request ID** is the SSM automation ID.
4. Go to **AWS Systems Manager** > **Automation**.
5. Filter by **Execution ID** using SSM automation ID and view the automation's details.

   You can analyze the execution with any Systems Manager automation. For more information, see the AWS Systems Manager Automation user guide. The execution input parameters appear in the **Input parameters** section of the **Execution Detail** and include optional parameters not appearing in the AWS FIS experiment.

You can find information on step status and other step details by drilling down to specific steps within the Execution steps.

**Common failures**

The following are common failures encountered when executing an assessment report:

- Alarm template was not deployed before the Test/SOP experiment was executed. This causes an error message during the automation step.
  - **Failure message:** The following parameters were not found: [/ResilienceHub/Alarm/3dee49a1-9877-452a-bb0c-a958479a8ef2/nat-gw-alarm-bytes-out-to-source-2020-09-21_nat-02ad9bc4fbd4e6135]. Make sure all the SSM parameters in automation document are created in SSM Parameter Store
  - **Remediation:** Make sure to render the relevant alarm and deploy the resulting template before rerunning the fault injection experiment.
- Missing permissions in the execution role. This error message occurs if the provided execution role is missing a permission and appears within the step details.
  - **Failure message:** An error occurred (Unauthorized Operation) when calling the DescribeInstanceStatus operation: You are not authorized to perform this operation. Please Refer to Automation Service Troubleshooting Guide for more diagnosis details.
  - **Remediation**: Verify you provided the correct execution role. If this was done, add the required permission and rerun the assessment.
- Execution succeeded but did not have the expected result. This is the result of incorrect parameters or an internal automation issue.
  - **Failure message:** The execution succeeded, so no error message is shown.
  - **Remediation:** Check the input parameters and look at the executed steps as explained in the Analyze AWS FIS experiment execution before examining the individual steps for expected inputs and outputs.

# Managing alarms

When you run a resiliency assessment, AWS Resilience Hub recommends setting up Amazon CloudWatch alarms to monitor your application resiliency. We recommend these alarms based on the resources and components of your current application configuration. If the resources and components in your application change, you should run a resiliency assessment to ensure you have the correct alarms for your updated application.

## Creating alarms

You can create alarms based on operational recommendations. This view contains recommendations to set up alarms. AWS Resilience Hub provides AWS CloudFormation templates for you to download.

**To create alarms in operational recommendations:**

1. Review the alarm recommendations, choose the appropriate AWS CloudFormation template, and enter a unique name. AWS Resilience Hub creates a template based on your selected recommendations.
2. Access the templates you created through an Amazon S3 URL. To do so:
   - In **Templates**, open an alarm recommendation.

- In **Templates S3 Path**, open the link to see the list of all objects in your Amazon S3 bucket.

## Viewing alarms

AWS Resilience Hub creates a AWS CloudFormation template that contains details to create the selected alarms in Amazon CloudWatch. After the template is generated, you can access it through an Amazon S3 URL, and can download and place it into your code pipeline or create a stack through the AWS CloudFormation console.

From the resiliency assessment, select the alarms that you want to set up for your application, and choose **Create CloudFormation template**.

### To view alarms

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application.
3. In **Alarms**, you can see a list of alarms generated from alarm recommendations in **Operational recommendations**. It categorizes the alarms by name, status, metric, and resource.

*Viewing recommended alarms*

From the resiliency assessment, select alarms that you would like to set up for your application, and choose **Create CloudFormation template**.

AWS Resilience Hub creates an AWS CloudFormation template that contains details to create the selected alarms in Amazon CloudWatch. Once the template is generated, you can access it through an Amazon S3 URL, and can download and place it into your code pipeline or create a stack through the AWS CloudFormation console.

# Integrating operational recommendations into your application with AWS CloudFormation

After you choose **Create CloudFormation template** in the **Operational recommendations** page, AWS Resilience Hub creates an AWS CloudFormation template that describes the specific alarm, standard operating procedure (SOP), or AWS FIS experiment for your application. The AWS CloudFormation template is stored in an Amazon S3 bucket, and you can check the S3 path to the template in the **Template details** tab on the **Operational recommendations** page.

For example, the listing below shows a JSON-formatted AWS CloudFormation template that describes an alarm recommendation rendered by AWS Resilience Hub. It's a Read Throttling Alarm for a DynamoDB table called `Employees`.

The `Resources` section of the template describes the `AWS::CloudWatch::Alarm` alarm that's activated when the number of read throttle events for the DynamoDB table exceeds 1. And the two `AWS::SSM::Parameter` resources define metadata that allow AWS Resilience Hub to identify installed resources without having to scan the actual application.

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "SNSTopicARN" : {
```

```
      "Type" : "String",
      "Description" : "The ARN of the SNS topic to which alarm status changes are to be
 sent. This must be in the same region being deployed.",
      "AllowedPattern" : "^arn:(aws|aws-cn|aws-iso|aws-iso-[a-z]{1}|aws-us-gov):sns:([a-z]
{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-[0-9]):[0-9]{12}:[A-Za-z0-9/][A-Za-z0-9:_/+=,@.-]
{1,256}$"
    }
  },
  "Resources" : {
    "ReadthrottleeventsthresholdexceededEmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm" :
 {
      "Type" : "AWS::CloudWatch::Alarm",
      "Properties" : {
        "AlarmDescription" : "Alarm by Resilience Hub that reports when amount of read
 throttle events is greater than 1",
        "AlarmName" : "ResilienceHub-ReadThrottleEventsAlarm-2020-04-01_Employees-ON-
DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9",
        "AlarmActions" : [ {
          "Ref" : "SNSTopicARN"
        } ],
        "MetricName" : "ReadThrottleEvents",
        "Namespace" : "AWS/DynamoDB",
        "Statistic" : "Sum",
        "Dimensions" : [ {
          "Name" : "TableName",
          "Value" : "Employees-ON-DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9"
        } ],
        "Period" : 60,
        "EvaluationPeriods" : 1,
        "DatapointsToAlarm" : 1,
        "Threshold" : 1,
        "ComparisonOperator" : "GreaterThanOrEqualToThreshold",
        "TreatMissingData" : "notBreaching",
        "Unit" : "Count"
      },
      "Metadata" : {
        "AWS::ResilienceHub::Monitoring" : {
          "recommendationId" : "dynamodb:alarm:health-read_throttle_events:2020-04-01"
        }
      }
    },

 "dynamodbalarmhealthreadthrottleevents20200401EmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9AlarmSSMParam
 {
      "Type" : "AWS::SSM::Parameter",
      "Properties" : {
        "Name" : "/ResilienceHub/Alarm/3f904525-4bfa-430f-96ef-58ec9b19aa73/dynamodb-alarm-
health-read-throttle-events-2020-04-01_Employees-ON-DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9",
        "Type" : "String",
        "Value" : {
          "Fn::Sub" :
 "${ReadthrottleeventsthresholdexceededEmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm}"
        },
        "Description" : "SSM Parameter for identifying installed resources."
      }
    },

 "dynamodbalarmhealthreadthrottleevents20200401EmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9AlarmInfoSSMP
 {
      "Type" : "AWS::SSM::Parameter",
      "Properties" : {
        "Name" : "/ResilienceHub/Info/Alarm/3f904525-4bfa-430f-96ef-58ec9b19aa73/dynamodb-
alarm-health-read-throttle-events-2020-04-01_Employees-ON-DEMAND-0-DynamoDBTable-
PXBZQYH3DCJ9",
        "Type" : "String",
        "Value" : {
```

```
          "Fn::Sub" : "{\"alarmName\":
\"${ReadthrottleeventsthresholdexceededEmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm}\",
\"referenceId\":\"dynamodb:alarm:health_read_throttle_events:2020-04-01\",
\"resourceId\":\"Employees-ON-DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9\",\"relatedSOPs\":
[\"dynamodb:sop:update_provisioned_capacity:2020-04-01\"]}"
        },
        "Description" : "SSM Parameter for identifying installed resources."
      }
    }
  }
}
```

# Modifying the AWS CloudFormation template

The easiest way to integrate an alarm, SOP, or AWS FIS resource into your main application is to simply add it as another resource in the template that describes your application template. The JSON-formatted file provided below provides a basic outline of how a DynamoDB table is described in an AWS CloudFormation template. A real application is likely to include several more resources, such as additional tables.

```
{
   "AWSTemplateFormatVersion": "2010-09-09T00:00:00.000Z",
   "Description": "Application Stack with Employees Table",
   "Outputs": {
      "DynamoDBTable": {
         "Description": "The DynamoDB Table Name",
         "Value": {"Ref": "Employees"}
      }
   },
   "Resources": {
      "Employees": {
         "Type": "AWS::DynamoDB::Table",
         "Properties": {
            "BillingMode": "PAY_PER_REQUEST",
            "AttributeDefinitions": [
               {
                  "AttributeName": "USER_ID",
                  "AttributeType": "S"
               },
               {
                  "AttributeName": "RANGE_ATTRIBUTE",
                  "AttributeType": "S"
               }
            ],
            "KeySchema": [
               {
                  "AttributeName": "USER_ID",
                  "KeyType": "HASH"
               },
               {
                  "AttributeName": "RANGE_ATTRIBUTE",
                  "KeyType": "RANGE"
               }
            ],
            "PointInTimeRecoverySpecification": {
               "PointInTimeRecoveryEnabled": true
            },
            "Tags": [
               {
                  "Key": "Key",
                  "Value": "Value"
               }
            ],
```

```
            "LocalSecondaryIndexes": [
               {
                  "IndexName": "resiliencehub-index-local-1",
                  "KeySchema": [
                     {
                        "AttributeName": "USER_ID",
                        "KeyType": "HASH"
                     },
                     {
                        "AttributeName": "RANGE_ATTRIBUTE",
                        "KeyType": "RANGE"
                     }
                  ],
                  "Projection": {
                     "ProjectionType": "ALL"
                  }
               }
            ],
            "GlobalSecondaryIndexes": [
               {
                  "IndexName": "resiliencehub-index-1",
                  "KeySchema": [
                     {
                        "AttributeName": "USER_ID",
                        "KeyType": "HASH"
                     }
                  ],
                  "Projection": {
                     "ProjectionType": "ALL"
                  }
               }
            ]
         }
      }
   }
}
```

To allow the alarm resource to be deployed with your application, you now need to replace the hardcoded resources with a dynamic reference in the application stacks.

So in the `AWS::CloudWatch::Alarm` resource definition, change the following:

```
"Value" : "Employees-ON-DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9"
```

to the below:

```
"Value" : {"Ref": "Employees"}
```

And under in the `AWS::SSM::Parameter` resource definition, change the following:

```
"Fn::Sub" : "{\"alarmName\":
\"${ReadthrottleeventsthresholdexceededDynamoDBEmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm}\",
\"referenceId\":\"dynamodb:alarm:health_read_throttle_events:2020-04-01\",
\"resourceId\":\"Employees-ON-DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9\",\"relatedSOPs\":
[\"dynamodb:sop:update_provisioned_capacity:2020-04-01\"]}"
```

to the below:

```
"Fn::Sub" : "{\"alarmName\":
\"${ReadthrottleeventsthresholdexceededEmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm}\",
\"referenceId\":\"dynamodb:alarm:health_read_throttle_events:2020-04-01\",\"resourceId\":
```

```
\"${Employees}\",\"relatedSOPs\":
[\"dynamodb:sop:update_provisioned_capacity:2020-04-01\"]}"
```

When modifying AWS CloudFormation templates for SOPs and AWS FIS experiments, you will take the same approach, replacing hardcoded reference IDs with dynamic references that continue to work even after hardware changes.

By using a reference to the DynamoDB table, you allow AWS CloudFormation to do the following:

- Create the database table first.
- Always use the actual ID of the generated resource in the alarm, and update the alarm dynamically if AWS CloudFormation needs to replace the resource.

> **Note**
> You can choose more advanced methods for managing your application resources with AWS CloudFormation such as nesting stacks or referring to resource outputs in a separate AWS CloudFormation stack. (But if you want to keep the recommendation stack separate from the main stack, you need to configure a way to pass information between the two stacks.)
> In addition, third-party tools, such as Terraform by HashiCorp, can also be used to provision Infrastructure as Code (IaC).

# Security in AWS Resilience Hub

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 Resilience Hub, 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 Resilience Hub. The following topics show you how to configure AWS Resilience Hub 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 Resilience Hub resources.

**Contents**



# Data protection in AWS Resilience Hub

The AWS shared responsibility model applies to data protection in AWS Resilience Hub. 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. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center (successor to AWS Single Sign-On) 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 recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- 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 140-2 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) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with Resilience Hub or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form 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.

## Encryption at rest

AWS Resilience Hub encrypts your data at rest. Data in Resilience Hub is encrypted at rest using transparent server-side encryption. This helps reduce the operational burden and complexity involved in protecting sensitive data. With encryption at rest, you can build security-sensitive applications that meet encryption compliance and regulatory requirements.

## Encryption in transit

Resilience Hub encrypts data in transit between the service and other integrated AWS services. All data that passes between Resilience Hub and integrated services is encrypted using Transport Layer Security (TLS). Resilience Hub provides preconfigured actions for specific types of targets across AWS services, and supports actions for target resources.

# Identity and access management for AWS Resilience Hub

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 Resilience Hub resources. IAM is an AWS service that you can use with no additional charge.

**Contents**



## Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Resilience Hub.

**Service user** – If you use the Resilience Hub service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Resilience Hub features to do your work, you might need additional permissions. Understanding how access is managed can help you

request the right permissions from your administrator. If you cannot access a feature in Resilience Hub, see Troubleshooting AWS Resilience Hub identity and access (p. 69).

**Service administrator** – If you're in charge of Resilience Hub resources at your company, you probably have full access to Resilience Hub. It's your job to determine which Resilience Hub features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Resilience Hub, see How AWS Resilience Hub works with IAM (p. 49).

**IAM administrator** – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Resilience Hub. To view example Resilience Hub identity-based policies that you can use in IAM, see AWS Resilience Hub policy examples (p. 52).

# Authenticating with identities

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

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (successor to AWS Single Sign-On) (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signature Version 4 signing process in the *AWS General Reference*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

## AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *AWS Account Management Reference Guide*.

## IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys.

For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

## IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see LIST URL in the *Service Authorization Reference*.
  - **Service role** – 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 Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – 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 IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests.

This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) 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 is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

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**.

Every IAM entity (user or role) starts with no permissions. By default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

## Identity-based policies

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 Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing 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 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.

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.

## Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

## Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. 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 Resilience Hub works with IAM

Before you use IAM to manage access to Resilience Hub, make sure to understand what IAM features are available to use with Resilience Hub. To get a high-level view of how Resilience Hub and other AWS services work with IAM, see AWS Services That Work with IAM in the *IAM User Guide*.

**Contents**

- Troubleshooting AWS Resilience Hub identity and access (p. 69)

# AWS Resilience Hub identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources, and also the conditions under which actions are allowed or denied. Resilience Hub supports specific actions, resources, and condition keys. For more information about all of the elements that you use in a JSON policy, see IAM JSON Policy Elements Reference in the *IAM User Guide*.

## Actions

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 `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in AWS Resilience Hub use the following prefix before the action: `resiliencehub:`. For example, to grant someone permission to create an app, you include the `resiliencehub:CreateApp` action in their policy. Policy statements must include either an `Action` or `NotAction` element. AWS Resilience Hub defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [
      "resiliencehub:action1",
      "resiliencehub:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `List`, include the following action:

```
"Action": "resiliencehub:List*"
```

To see a list of AWS Resilience Hub actions, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_Operations.html in the *AWS Resilience Hub API Reference*.

## Resources

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. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

For example, an application has the following ARN:

```
arn:${Partition}:resiliencehub:${Region}:${Account}:app/${appId}
```

For more information about the format of ARNs, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

You can't perform some Resilience Hub actions, such as those for creating resources, on a specific resource. In those cases, you must use the wildcard (*).

```
"Resource": "*"
```

Some Resilience Hub API actions involve multiple resources. To specify multiple resources in a single statement, separate the ARNs with commas, as in the following example.

```
"Resource": [
      "resource1",
      "resource2"
```

### Condition keys

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 (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. 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.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of AWS Resilience Hub condition keys, see CONDITIONS URL in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_Operations.html.

## Resource-based policies

Resilience Hub does not support resource-based policies.

## Authorization based on Resilience Hub tags

You can attach tags to Resilience Hub resources or pass tags in a request to Resilience Hub. To use tags to control access, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

For an example of a policy, see Example: Use tags to control the use of resources (p. 55).

# Resilience Hub IAM roles

An IAM role is an entity within your AWS account that has specific permissions.

## Using temporary credentials with Resilience Hub

You can use temporary credentials to sign in with federation, to assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations, such as AssumeRole or GetFederationToken.

AWS Resilience Hub supports using temporary credentials.

# AWS Resilience Hub policy examples

By default, IAM users and roles don't have permission to create or modify AWS Resilience Hub resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

For information about how to create an IAM identity-based policy using these example JSON policy documents, see Creating Policies on the JSON Tab in the *IAM User Guide*.

**Examples**



## Policy best practices

Identity-based policies are effective for determining whether someone can create, access, or delete Resilience Hub 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:

- **Grant minimum required permissions** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant least privilege in the *IAM User Guide*.
- **Enable MFA for sensitive operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.

## Example: Use the Resilience Hub console

To access the AWS Resilience Hub console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Resilience Hub 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 (IAM users or roles) with that policy.

The following policy grants users permission to list and view all resources in the Resilience Hub console, but not to create, update, or delete them.

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                 "resiliencehub:List*",
                 "resiliencehub:Describe*"
             ],
             "Resource": "*"
         }
     ]
 }
```

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 you're trying to perform.

## Example: List available Resilience Hub applications

The following policy grants users permission to list the available Resilience Hub applications.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "resiliencehub:ListApps"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

## Example: Start an application assessment

The following policy grants users permission to start an assessment for a specific AWS Resilience Hub application.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "resiliencehub:StartAppAssessment"
```

```
      ],
      "Resource": [
        "arn:aws:resiliencehub:*:*:app/appId"
      ]
    }
  ]
}
```

## Example: Delete an application assessment

The following policy grants users permission to delete an assessment for a specific AWS Resilience Hub application.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "resiliencehub:DeleteAppAssessment"
      ],
      "Resource": [
        "arn:aws:resiliencehub:*:*:app/appId"
      ]
    }
  ]
}
```

## Example: Create a recommendation template for a specific application

The following policy grants users permission to create a recommendation template for a specific AWS Resilience Hub application.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "resiliencehub:CreateRecommendationTemplate"
      ],
      "Resource": [
        "arn:aws:resiliencehub:*:*:app/appId"
      ]
    }
  ]
}
```

## Example: Delete a recommendation template for a specific application

The following policy grants users permission to delete a recommendation template for a specific AWS Resilience Hub application.

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

```
      "Action": [
          "resiliencehub:DeleteRecommendationTemplate"
      ],
      "Resource": [
        "arn:aws:resiliencehub:*:*:app/appId"
      ]
    }
  ]
}
```

## Example: Update an app with a specific resiliency policy

The following policy grants users permission to update a AWS Resilience Hub application with a specific resiliency policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "resiliencehub:UpdateApp"
      ],
      "Resource": [
        "arn:aws:resiliencehub:*:*:app/appId"
      ],
      "Condition": {
        "StringLike" : { "resiliencehub:policyArn" : "arn:aws:resiliencehub:us-
west-2:111122223333:resiliency-policy/*" }
      }
    }
  ]
}
```

## Example: Use tags to control the use of resources

The following policy allows users to use tags to control the use of resources.

```
{
     "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "resiliencehub:StartAppAssessment"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Test"
                }
            }
        }
    ]
}
```

## Example: Delete an application with a specific tag

The following policy grants users permission to delete a AWS Resilience Hub application with a specific tag.

```
{
     "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "resiliencehub:DeleteApp"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Test"
                }
            }
        }
    ]
}
```

# AWS Resilience Hub permissions reference

The following IAM policies and policy snippets define the permissions necessary to use AWS Resilience Hub.

**Contents**



## Permissions required to use AWS Resilience Hub to manage an application in a single AWS account

The following IAM policy is required for a single AWS account that will have the permissions to perform all the actions for AWS Resilience Hub.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "resiliencehub:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:GetTopicAttributes",
                "sns:ListSubscriptionsByTopic",
                "sns:GetSubscriptionAttributes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
            "cloudformation:DescribeStacks",
            "cloudformation:ListStackResources",
            "cloudformation:ValidateTemplate"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "servicecatalog:GetApplication",
            "servicecatalog:ListAssociatedResources"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "resource-groups:ListGroupResources",
            "resource-groups:GetGroup",
            "tag:GetResources"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "cloudwatch:DescribeAlarms",
            "cloudwatch:GetMetricData",
            "cloudwatch:GetMetricStatistics",
            "cloudwatch:PutMetricData"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "fis:GetExperimentTemplate",
            "fis:ListExperimentTemplates",
            "fis:ListExperiments"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ssm:GetParametersByPath"
        ],
        "Resource": "arn:aws:ssm:*:customer_account_id:parameter/ResilienceHub/*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetBucketPolicyStatus",
            "s3:PutBucketVersioning",
            "s3:GetBucketTagging",
            "s3:GetBucketVersioning",
            "s3:GetReplicationConfiguration",
            "s3:ListBucket",
            "s3:ListAllMyBuckets",
            "s3:GetBucketLocation"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
```

```
            "s3:CreateBucket",
            "s3:PutObject",
            "s3:GetObject"
        ],
        "Resource": "arn:aws:s3:::aws-resilience-hub-artifacts-*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "autoscaling:DescribeAutoScalingGroups"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeAvailabilityZones",
            "ec2:DescribeVpcEndpoints",
            "ec2:DescribeFastSnapshotRestores",
            "ec2:DescribeInstances",
            "ec2:DescribeSnapshots",
            "ec2:DescribeVolumes",
            "ec2:DescribeNatGateways",
            "ec2:DescribeSubnets",
            "ec2:DescribeRegions",
            "ec2:DescribeTags"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "rds:DescribeDBClusters",
            "rds:DescribeDBInstanceAutomatedBackups",
            "rds:DescribeDBInstances",
            "rds:DescribeGlobalClusters",
            "rds:DescribeDBClusterSnapshots"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:DescribeTargetGroups",
            "elasticloadbalancing:DescribeLoadBalancers",
            "elasticloadbalancing:DescribeTargetHealth"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "lambda:GetFunction",
            "lambda:GetFunctionConcurrency",
            "lambda:ListAliases",
            "lambda:ListVersionsByFunction"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ecr:DescribeRegistry"
        ],
        "Resource": "*"
    },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "backup:DescribeBackupVault",
                "backup:GetBackupPlan",
                "backup:GetBackupSelection",
                "backup:ListBackupPlans",
                "backup:ListBackupSelections"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:ListTagsOfResource",
                "dynamodb:DescribeTable",
                "dynamodb:DescribeGlobalTable",
                "dynamodb:ListGlobalTables",
                "dynamodb:DescribeContinuousBackups",
                "dynamodb:DescribeLimits"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:DescribeMountTargets",
                "elasticfilesystem:DescribeFileSystems"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:GetQueueAttributes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "apigateway:GET"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecs:DescribeClusters",
                "ecs:ListServices",
                "ecs:DescribeServices",
                "ecs:DescribeCapacityProviders",
                "ecs:DescribeContainerInstances",
                "ecs:ListContainerInstances",
                "ecs:DescribeTaskDefinition"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "route53-recovery-control-config:ListControlPanels",
                "route53-recovery-control-config:ListRoutingControls",
                "route53-recovery-readiness:ListReadinessChecks",
                "route53-recovery-readiness:GetResourceSet",
```

```
                "route53-recovery-readiness:GetReadinessCheckStatus",
                "route53-recovery-control-config:ListClusters",
                "route53:ListHealthChecks",
                "route53:ListHostedZones",
                "route53:ListResourceRecordSets",
                "route53:GetHealthCheck"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "drs:DescribeSourceServers",
                "drs:DescribeJobs",
                "drs:GetReplicationConfiguration"
            ],
            "Resource": "*"
        }

    ]
}
```

**Note**
If you want to use your own Amazon S3 bucket, you can pass the `bucketName` parameter to the `CreateRecommendationTemplate` API action. If that's the case, you won't need the `s3:CreateBucket` permission, but you will need the `s3:PutObject` and `s3:GetObject` permissions for the input bucket.

## Permissions required to use AWS Resilience Hub to manage scheduled assessments in a single AWS account

The following IAM policy is required for the `AwsResilienceHubPeriodicAssessmentRole` role to have the permissions to perform scheduled assessment actions in AWS Resilience Hub .

**Note**
The role name is `AwsResilienceHubPeriodicAssessmentRole`.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
             "Effect": "Allow",
             "Action": [
                "iam:GetRole",
                "iam:PassRole",
                "sts:AssumeRole"
            ],
            "Resource": "arn:aws:iam::primary_account_id:role/
AwsResilienceHubAdminAccountRole"
    },
        {
            "Effect": "Allow",
            "Action": [
                "resiliencehub:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:GetTopicAttributes",
                "sns:ListSubscriptionsByTopic",
                "sns:GetSubscriptionAttributes"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:DescribeStacks",
                "cloudformation:ListStackResources",
                "cloudformation:ValidateTemplate"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:GetApplication",
                "servicecatalog:ListAssociatedResources"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "resource-groups:ListGroupResources",
                "resource-groups:GetGroup",
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "fis:GetExperimentTemplate",
                "fis:ListExperimentTemplates",
                "fis:ListExperiments"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParametersByPath"
            ],
            "Resource": "arn:aws:ssm:*:customer_account_id:parameter/ResilienceHub/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketPolicyStatus",
                "s3:PutBucketVersioning",
                "s3:GetBucketTagging",
                "s3:GetBucketVersioning",
                "s3:GetReplicationConfiguration",
                "s3:ListBucket",
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::aws-resilience-hub-artifacts-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeFastSnapshotRestores",
                "ec2:DescribeInstances",
                "ec2:DescribeSnapshots",
                "ec2:DescribeVolumes",
                "ec2:DescribeNatGateways",
                "ec2:DescribeSubnets",
                "ec2:DescribeRegions",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "rds:DescribeDBClusters",
                "rds:DescribeDBInstanceAutomatedBackups",
                "rds:DescribeDBInstances",
                "rds:DescribeGlobalClusters",
                "rds:DescribeDBClusterSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeTargetHealth"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:GetFunction",
                "lambda:GetFunctionConcurrency",
                "lambda:ListAliases",
                "lambda:ListVersionsByFunction"
            ],
            "Resource": "*"
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "ecr:DescribeRegistry"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "backup:DescribeBackupVault",
                "backup:GetBackupPlan",
                "backup:GetBackupSelection",
                "backup:ListBackupPlans",
                "backup:ListBackupSelections"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:ListTagsOfResource",
                "dynamodb:DescribeTable",
                "dynamodb:DescribeGlobalTable",
                "dynamodb:ListGlobalTables",
                "dynamodb:DescribeContinuousBackups",
                "dynamodb:DescribeLimits"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:DescribeMountTargets",
                "elasticfilesystem:DescribeFileSystems"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:GetQueueAttributes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "apigateway:GET"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecs:DescribeClusters",
                "ecs:ListServices",
                "ecs:DescribeServices",
                "ecs:DescribeCapacityProviders",
                "ecs:DescribeContainerInstances",
                "ecs:ListContainerInstances",
                "ecs:DescribeTaskDefinition"
            ],
            "Resource": "*"
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "route53-recovery-control-config:ListControlPanels",
                "route53-recovery-control-config:ListRoutingControls",
                "route53-recovery-readiness:ListReadinessChecks",
                "route53-recovery-readiness:GetResourceSet",
                "route53-recovery-readiness:GetReadinessCheckStatus",
                "route53-recovery-control-config:ListClusters",
                "route53:ListHealthChecks",
                "route53:ListHostedZones",
                "route53:ListResourceRecordSets",
                "route53:GetHealthCheck"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "drs:DescribeSourceServers",
                "drs:DescribeJobs",
                "drs:GetReplicationConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

The associated trust policy for the scheduled assessments role, (`AwsResilienceHubPeriodicAssessmentRole`), gives permissions for the AWS Resilience Hub service to assume the scheduled assessments role.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "resiliencehub.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

## Permissions required to use AWS Resilience Hub to manage application in multiple accounts

The following IAM permissions policies are necessary if you're using AWS Resilience Hub with multiple accounts. Each account might need different permissions depending on your use case.

### Calling account permissions

The following IAM policy is required for the AWS account that will have only the permissions necessary to call into AWS Resilience Hub.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "iam:GetRole",
```

```
            "iam:PassRole",
            "sts:AssumeRole"
        ],
        "Resource": "arn:aws:iam::primary_account_id:role/AwsResilienceHubAdminAccountRole"
    },
    {
        "Effect": "Allow",
        "Action": [
            "resiliencehub:*"
        ],
        "Resource": "*"
    }
  ]
}
```

### Admin account permissions

The following IAM policy is required for the AWS account that will have admin permissions for AWS Resilience Hub.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["sts:AssumeRole"],
      "Resource": ["arn:aws:iam::secondary_account_id:role/
AwsResilienceHubExecutorAccountRole"],
      "Effect": "Allow"
    }
  ]
}
```

The associated trust policy for the admin role is as follows, where *caller_IAM_role* is the role used in the primary account to call the APIs for AWS Resilience Hub.

```
{
  "Version": "2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::primary_account_id:role/caller_IAM_role"
        },
            "Action": "sts:AssumeRole"
        },
        {
             "Effect": "Allow",
             "Principal": {
                "AWS": "arn:aws:iam::primary_account_id:role/
AwsResilienceHubPeriodicAssessmentRole"
        },
            "Action": "sts:AssumeRole"
    }
  ]
}
```

### Executor account role permissions

The following IAM policy is required for the AWS account that will have the executor account role permissions for AWS Resilience Hub.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "cloudformation:DescribeStacks",
            "cloudformation:ListStackResources",
            "cloudformation:ValidateTemplate"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "sns:GetTopicAttributes",
            "sns:ListSubscriptionsByTopic",
            "sns:GetSubscriptionAttributes"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "resource-groups:ListGroupResources",
            "resource-groups:GetGroup",
            "tag:GetResources"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "cloudwatch:DescribeAlarms",
            "cloudwatch:GetMetricData",
            "cloudwatch:GetMetricStatistics",
            "cloudwatch:PutMetricData"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "fis:GetExperimentTemplate",
            "fis:ListExperimentTemplates",
            "fis:ListExperiments"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ssm:DescribeAutomationExecutions"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "autoscaling:DescribeAutoScalingGroups"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeAvailabilityZones",
```

```
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeFastSnapshotRestores",
                "ec2:DescribeInstances",
                "ec2:DescribeSnapshots",
                "ec2:DescribeVolumes",
                "ec2:DescribeNatGateways",
                "ec2:DescribeSubnets",
                "ec2:DescribeRegions",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "rds:DescribeDBClusters",
                "rds:DescribeDBInstanceAutomatedBackups",
                "rds:DescribeDBInstances",
                "rds:DescribeGlobalClusters",
                "rds:DescribeDBClusterSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeTargetHealth"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:GetFunction",
                "lambda:GetFunctionConcurrency",
                "lambda:ListAliases",
                "lambda:ListVersionsByFunction"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:DescribeRegistry"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "backup:DescribeBackupVault",
                "backup:GetBackupPlan",
                "backup:GetBackupSelection",
                "backup:ListBackupPlans",
                "backup:ListBackupSelections"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:ListTagsOfResource",
                "dynamodb:DescribeTable",
                "dynamodb:DescribeGlobalTable",
```

```
                "dynamodb:ListGlobalTables",
                "dynamodb:DescribeContinuousBackups",
                "dynamodb:DescribeLimits"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:DescribeMountTargets",
                "elasticfilesystem:DescribeFileSystems"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketPolicyStatus",
                "s3:PutBucketVersioning",
                "s3:GetBucketTagging",
                "s3:GetBucketVersioning",
                "s3:GetReplicationConfiguration",
                "s3:ListBucket",
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:GetQueueAttributes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "apigateway:GET"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParametersByPath"
            ],
            "Resource": "arn:aws:ssm:*:secondary_account_id:parameter/ResilienceHub/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecs:DescribeClusters",
                "ecs:ListServices",
                "ecs:DescribeServices",
                "ecs:DescribeCapacityProviders",
                "ecs:DescribeContainerInstances",
                "ecs:ListContainerInstances",
                "ecs:DescribeTaskDefinition"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "route53-recovery-control-config:ListControlPanels",
                "route53-recovery-control-config:ListRoutingControls",
                "route53-recovery-readiness:ListReadinessChecks",
                "route53-recovery-readiness:GetResourceSet",
                "route53-recovery-readiness:GetReadinessCheckStatus",
                "route53-recovery-control-config:ListClusters",
                "route53:ListHealthChecks",
                "route53:ListHostedZones",
                "route53:ListResourceRecordSets",
                "route53:GetHealthCheck"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "drs:DescribeSourceServers",
                "drs:DescribeJobs",
                "drs:GetReplicationConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

The associated trust policy for the executor account role. This gives permission for the primary account role (`AwsResilienceHubAdminAccountRole`) to assume the secondary accounts.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::primary_account_id:role/AwsResilienceHubAdminAccountRole"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

# Troubleshooting AWS Resilience Hub identity and access

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

**Issues**

- I am not authorized to perform iam:PassRole (p. 69)
- I want to view my access keys (p. 70)
- I'm an administrator and want to allow others to access Resilience Hub (p. 70)
- I want to allow people outside of my AWS account to access my Resilience Hub resources (p. 70)

## I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Resilience Hub.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Resilience Hub. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

## I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

> **Important**
> Do not provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see Managing access keys in the *IAM User Guide*.

## I'm an administrator and want to allow others to access Resilience Hub

To allow others to access Resilience Hub, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in Resilience Hub.

To get started right away, see Creating your first IAM delegated user and group in the *IAM User Guide*.

## I want to allow people outside of my AWS account to access my Resilience Hub 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 Resilience Hub supports these features, see How AWS Resilience Hub works with IAM (p. 49).
- 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 How IAM roles differ from resource-based policies in the *IAM User Guide*.

# Infrastructure security in AWS Resilience Hub

As a managed service, AWS Resilience Hub; (Resilience Hub) is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access Resilience Hub through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

# Working with other services

This section describes AWS services that interact with AWS Resilience Hub.

## Creating AWS Resilience Hub resources with AWS CloudFormation

AWS Resilience Hub is integrated with AWS CloudFormation, a service that helps you to model and set up your AWS resources so that you can spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want (such as AWS::ResilienceHub:::ResiliencyPolicy and AWS::ResilienceHub:::App), and AWS CloudFormation provisions and configures those resources for you.

When you use AWS CloudFormation, you can reuse your template to set up your Resilience Hub resources consistently and repeatedly. Describe your resources one time, and then provision the same resources repeatedly in multiple AWS accounts and Regions.

### Resilience Hub and AWS CloudFormation templates

To provision and configure resources for Resilience Hub and related services, you must understand AWS CloudFormation templates. Templates are formatted text files in JSON or YAML. These templates describe the resources that you want to provision in your AWS CloudFormation stacks. If you're unfamiliar with JSON or YAML, you can use AWS CloudFormation Designer to help you get started with AWS CloudFormation templates. For more information, see What is AWS CloudFormation Designer? in the *AWS CloudFormation User Guide*.

Resilience Hub supports creating AWS::ResilienceHub:::ResiliencyPolicy and AWS::ResilienceHub:::App in AWS CloudFormation. For more information, including examples of JSON and YAML templates for AWS::ResilienceHub:::ResiliencyPolicy and AWS::ResilienceHub:::App, see the AWS Resilience Hub resource type reference in the *AWS CloudFormation User Guide*.

You can use AWS CloudFormation stacks to define Resilience Hub applications. A stack lets you manage related resources as a single unit. A stack can contain all the resources that you need to run a web application, such as a web server or networking rules. .

### Learn more about AWS CloudFormation

For more information about AWS CloudFormation, see the following resources:

- AWS CloudFormation
- AWS CloudFormation User Guide
- AWS CloudFormation API Reference
- AWS CloudFormation Command Line Interface User Guide

## AWS CloudTrail

AWS Resilience Hub (Resilience Hub) is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, a role, or an AWS service in Resilience Hub. CloudTrail captures all API calls for

Resilience Hub as events. The calls that are captured include calls from the Resilience Hub console and code calls to the Resilience Hub API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Resilience Hub. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in Event history. Using the information collected by CloudTrail, you can determine the request that was made to Resilience Hub, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information about CloudTrail, see the AWS CloudTrail User Guide.

# AWS Systems Manager

AWS Resilience Hub works with Systems Manager to automate the steps of your SOPs by providing a number of SSM documents you can use as the basis for those SOPs.

AWS Resilience Hub provides you AWS CloudFormation templates that contains the IAM roles required to run different Systems Manager documents, one role per document with permissions required for the specific document. After creating a stack with the AWS CloudFormation template, it will setup the IAM roles and save metadata in Systems Manager parameter for the Systems Manager automation document to run for different recovery procedures.

For more information on using SOPs, see Standard operating procedures (p. 33).

# AWS Trusted Advisor

AWS Resilience Hub is integrated with AWS Trusted Advisor, a service that is a centralized home of AWS best practice recommendations that helps you to identify, prioritize, and optimize your deployment on AWS.

AWS Trusted Advisor (under Fault tolarence category in the Dashboard menu) now provides multiple high-level recommendations through checks for each application in the AWS Resilience Hub. These checks indicate if the applications are breaching their resilience policy and also displays their resilience score. For more insights about the recommendations for each application in the AWS Trusted Advisor, we recommend you to view the detailed recomendations provided in the AWS Resilience Hub.

AWS Trusted Advisor provides the following checks for each application in AWS Resilience Hub:

- **AWS Resilience Hub resilience scores** - Checks if you have run an assessment for your applications in AWS Resilience Hub and alerts you if your resilience scores are below a specific value. This check provides resilience scores using the following alert criterias:
  - **Green** - Indicates if your application has a resilience score of 70 and above.
  - **Yellow** - Indicates if your application has a resilience score between 40 and 69.
  - **Red** - Indicates if your application has a resilience score less than 40.
- **AWS Resilience Hub policy breached** - Checks if the AWS Resilience Hub applications meet the RTO and RPO you have set for an application and alerts you if the application does not meet the RTO and RPO objectives. This check provides resilience scores using the following alert criterias:
  - **Green** - The application has a policy and the RTO and RPO estimates meet the RTO and RPO objectives.
  - **Yellow** - The application has a policy and has not been assessed.
  - **Red** - The application has a policy and the RTO and RPO estimates do not meet the RTO and RPO objectives.

For more information on using AWS Trusted Advisor, see the AWS Support User Guide.

# Document history for the AWS Resilience Hub User Guide

The following table describes the documentation for this release of AWS Resilience Hub.

- **API version: latest**
- **Latest documentation update:** November 16, 2022

| Change | Description | Date |
| --- | --- | --- |
| Integration with AWS Trusted Advisor (p. 75) | AWS Trusted Advisor users will be able to view applications associated with their account that have been assessed by AWS Resilience Hub. AWS Trusted Advisor shows the latest resilience score and provides a status that indicates if the resilience policy (RTO and RPO) has been met or not. Each time an assessment is run, AWS Resilience Hub updates AWS Trusted Advisor with the latest results. AWS Trusted Advisor is a service that continuously analyzes your AWS accounts and provides recommendations to help you to follow AWS best practices and AWS Well-Architected guidelines.<br><br>For more information, see the section called "AWS Trusted Advisor" (p. 73). | November 18, 2022 |
| Support for Amazon Simple Notification Service (Amazon SNS) (p. 75) | AWS Resilience Hub now assesses applications using Amazon SNS by analyzing Amazon SNS configuration, including subscribers, and provides recommendations to meet the organization's recovery objectives (RTO and RPO) for the applications. Amazon SNS is a managed service that delivers message from publishers (producers) to subscribers (consumers).<br><br>For more information, see the following topics: | November 16, 2022 |

| | | |
|---|---|---|
| | • the section called "Supported AWS Resilience Hub resources" (p. 5)<br>• the section called "Identity and access management" (p. 45)<br>• the section called "Grouping resources in an AppComponent" (p. 18) | |
| Additional Support for Amazon Route 53 Application Recovery Controller (Amazon Route 53 ARC) (p. 75) | AWS Resilience Hub now assesses Amazon Route 53 ARC for Elastic Load Balancing and Amazon Relational Database Service (Amazon RDS), which includes advising when Amazon Route 53 ARC would be beneficial. Extending AWS Resilience Hub, Amazon Route 53 ARC assessment support beyond AWS Auto Scaling Group (AWS ASG) and Amazon DynamoDB. Amazon Route 53 ARC provides high availability for your application, allowing you to quickly failover your entire application to a failover region.<br><br>For more information, see the following topics:<br>• the section called "Supported AWS Resilience Hub resources" (p. 5)<br>• the section called "Identity and access management" (p. 45) | November 16, 2022 |

| | | |
|---|---|---|
| Additional Support for AWS Backup (p. 75) | AWS Resilience Hub now assesses Amazon Route 53 ARC for Elastic Load Balancing and Amazon Relational Database Service (Amazon RDS), which includes advising when Amazon Route 53 ARC would be beneficial. Extending AWS Resilience Hub, Amazon Route 53 ARC assessment support beyond AWS Auto Scaling Group (AWS ASG) and Amazon DynamoDB. Amazon Route 53 ARC provides high availability for your application, allowing you to quickly failover your entire application to a failover region.<br><br>For more information, see the following topics:<br><br>• the section called "Supported AWS Resilience Hub resources" (p. 5)<br>• the section called "Identity and access management" (p. 45) | November 16, 2022 |
| Updated content: Permissions (p. 45) | Updated the documentation with permissions required to use AWS Resilience Hub. | July 7, 2022 |
| Updated content: Added new application component resources (p. 1) | Added Route53 and AWS Backup to the list of supported application component resources in the AppComponent grouping section. | July 1, 2022 |
| New content: Application compliance status concept (p. 2) | Added the Changes detected status type. | June 2, 2022 |
| Introducing AWS Resilience Hub (p. 75) | AWS Resilience Hub is now available. This guide describes how to use AWS Resilience Hub to analyze your infrastructure, get recommendations to improve the resiliency of your AWS apps, review resiliency scores, and more. | November 10, 2021 |

# AWS glossary

For the latest AWS terminology, see the AWS glossary in the *AWS General Reference*.