# AWS Well-Architected Tool

## User Guide

# AWS Well-Architected Tool: User Guide

# Table of Contents

# What is AWS Well-Architected Tool?

AWS Well-Architected Tool (AWS WA Tool) is a service in the cloud that provides a consistent process for measuring your architecture using AWS best practices. AWS WA Tool helps you throughout the product lifecycle by:

- Assisting with documenting the decisions that you make
- Providing recommendations for improving your workload based on best practices
- Guiding you in making your workloads more reliable, secure, efficient, and cost-effective

You can use AWS WA Tool to document and measure your workload using the best practices from the AWS Well-Architected Framework. These best practices were developed by AWS Solutions Architects based on their years of experience building solutions across a wide variety of businesses. The framework provides a consistent approach for measuring architectures and provides guidance for implementing designs that scale with your needs over time.

In addition to AWS best practices, you can use custom lenses to measure your workload using your own best practices. You can tailor the questions in a custom lens to be specific to a particular technology or to help you meet the governance needs within your organization. Custom lenses extend the guidance provided by the AWS lenses.

This service is intended for those involved in technical product development, such as chief technology officers (CTOs), architects, developers, and operations team members. AWS customers use AWS WA Tool to document their architectures, provide product launch governance, and to understand and manage the risks in their technology portfolio.

**Topics**



## The AWS Well-Architected Framework

The AWS Well-Architected Framework documents a set of foundational questions that enable you to understand how a specific architecture aligns with cloud best practices. The framework provides a consistent approach for evaluating systems against the qualities that are expected from modern cloud-based systems. Based on the state of your architecture, the framework suggests improvements that you can make to better achieve those qualities.

By using the framework, you learn architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. It provides a way for you to consistently measure your architectures against best practices and identify areas for improvement. The framework is based on six pillars: operational excellence, security, reliability, performance efficiency, cost optimization, and sustainability.

When designing a workload, you make trade-offs between these pillars based on your business needs. These business decisions help drive your engineering priorities. In development environments, you might optimize to reduce cost at the expense of reliability. In mission-critical solutions, you might optimize reliability and be willing to accept increased costs. In ecommerce solutions, you might prioritize

performance, since customer satisfaction can drive increased revenue. Security and operational excellence are generally not traded off against the other pillars.

For much more information on the framework, visit the AWS Well-Architected website.

# Definitions

In AWS WA Tool and the AWS Well-Architected Framework:

- A **workload** identifies a set of components that deliver business value. The workload is usually the level of detail that business and technology leaders communicate about. Examples of workloads include marketing websites, ecommerce websites, the backend for a mobile app, and analytic platforms. Workloads vary in their level of architectural complexity. They can be simple, such as a static website, or complex, such as microservices architectures with multiple data stores and many components.
- **Milestones** mark key changes in your architecture as it evolves throughout the product lifecycle — design, testing, go live, and production.
- **Lenses** provide a way for you to consistently measure your architectures against best practices and identify areas for improvement.

  In addition to the lenses provided by AWS, you also can create and use your own lenses, or use lenses that have been shared with you.
- **High risk issues (HRIs)** are architectural and operational choices that AWS has found might result in significant negative impact to a business. These HRIs might affect organizational operations, assets, and individuals.
- **Medium risk issues (MRIs)** are architectural and operational choices that AWS has found might negatively impact business, but to a lesser extent than HRIs.

  For additional information, see High Risk Issues (HRIs) and Medium Risk Issues (MRIs) (p. 16).

# AWS lenses

In addition to the AWS Well-Architected Framework Lens, which is applied to all workloads, AWS provides the following additional lenses:

## AWS Serverless Application Lens

The AWS Serverless Application Lens provides a set of additional questions that enable you to understand how a specific serverless application workload aligns with cloud best practices. The framework provides a consistent approach for evaluating key elements in a serverless architecture against the qualities that are expected from modern cloud-based systems. Based on the state of your architecture, the framework helps you understand potential risks and identifies next steps for improvement.

For more information, see the Serverless Applications Lens whitepaper.

## AWS SaaS Lens

The AWS SaaS Lens provides a set of additional questions for you to consider for your software as a service (SaaS) applications.

For more information, see the SaaS Lens whitepaper.

# AWS Foundational Technical Review (FTR) Lens

The AWS Foundational Technical Review (FTR) Lens provides a set of specific questions for independent software vendors (ISVs) to perform a workload self-assessment before requesting a Foundational Technical Review in the AWS Partner Network (APN).

# Getting started with AWS Well-Architected Tool

This section describes how to get started with AWS WA Tool.

**Topics**



## Provisioning an IAM user

In this step, you grant an IAM user permission to use AWS WA Tool.

**To provision an IAM user**

1. Create an IAM user or use an existing one associated with your AWS account. For more information, see Creating an IAM User in the *IAM User Guide*.
2. Grant the IAM user access to AWS Well-Architected Tool.

**Full access**

Full access allows the user to perform all actions in AWS WA Tool. This access is required to define workloads, delete workloads, view workloads, and update workloads.

Apply the **WellArchitectedConsoleFullAccess** managed policy to the user.

If you prefer to apply a custom inline policy, here is an example:

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

**Read-only access**

Read-only access allows the user to view workloads.

Apply the **WellArchitectedConsoleReadOnlyAccess** managed policy to the user.

If you prefer to apply a custom inline policy, here is an example:

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

The managed policies can be attached to an IAM user, group, or role.

To learn how to attach a policy to an IAM user, see Working with Policies. For more information on setting AWS WA Tool permissions, see Security (p. 43).

# Defining a workload

The next step is to define a workload.

### To define a workload

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. If this is your first time using AWS WA Tool, you see a page that introduces you to the features of the service. In the **Define a workload** section, choose **Define workload**.

   Alternately, in the left navigation pane, choose **Workloads** and choose **Define workload**.

   For details on how AWS uses your workload data, choose **Why does AWS need this data, and how will it be used?**
3. In the **Name** box, enter a name for your workload.

   **Note**
   The name must be between 3 and 100 characters. At least three characters must not be spaces. Workload names must be unique. Spaces and capitalization are ignored when checking for uniqueness.
4. In the **Description** box, enter a description of the workload. The description must be between 3 and 250 characters.
5. In the **Review owner** box, enter the name, email address, or identifier for the primary group or individual that owns the workload review process.
6. In the **Environment** box, choose the environment for your workload:
   - **Production** – Workload runs in a production environment.
   - **Pre-production** – Workload runs in a pre-production environment.
7. In the **Regions** section, choose the Regions for your workload:
   - **AWS Regions** – Choose the AWS Regions where your workload runs, one at a time.
   - **Non-AWS regions** – Enter the names of the regions outside of AWS where your workload runs. You can specify up to five unique regions, separated by commas.

   Use both options if appropriate for your workload.

8. (Optional) In the **Account IDs** box, enter the IDs of the AWS accounts associated with your workload. You can specify up to 100 unique account IDs, separated by commas.
9. (Optional) In the **Architectural design** box, enter the URL for your architectural design.
10. (Optional) In the **Industry type** box, choose the type of industry associated with your workload.
11. (Optional) In the **Industry** box, choose the industry that best matches your workload.
12. (Optional) In the **Tags** section, add any tags you want to associate with the workload.

    For more information on tags, see Tagging your AWS WA Tool resources (p. 60).
13. Choose **Next**.

    If a required box is blank or if a specified value is not valid, you must correct the issue before you can continue.
14. Choose the lenses that apply to this workload. Up to 20 lenses can be added to a workload. The following lenses are provided by AWS.
    - **AWS Well-Architected Framework** – This lens provides a set of foundation questions for you to consider for all of your cloud architectures. This lens is applied to all workloads.
    - **FTR Lens** – Select this lens for a set of additional questions to consider before requesting a Foundational Technical Review (FTR) in the AWS Partner Network (APN).
    - **Serverless Lens** – Select this lens for a set of additional questions to consider for your serverless application workloads.
    - **SaaS Lens** – Select this lens for a set of additional questions to consider for your software as a service (SaaS) workloads.

    If you have created custom lenses, or custom lenses have been shared with you, they also appear in the list.

    **Disclaimer**
    By accessing and/or applying custom lenses created by another AWS user or account, you acknowledge that custom lenses created by other users and shared with you are Third Party Content as defined in the AWS Customer Agreement.
15. Choose **Define workload**.

    If a required box is blank or if a specified value is not valid, you must correct the issue before your workload is defined.

# Documenting a workload

After a workload is defined, you document its state.

### To document the state of a workload

1. After you initially define a workload, you see a page that shows the current details of your workload. Choose **Start reviewing** to begin.

   Otherwise, in the left navigation pane, choose **Workloads** and select the name of the workload to open the workload details page. Choose **Continue reviewing**.
2. You are now presented with the first question. For each question:

   a. Read the question and determine if the question applies to your workload.

      For additional guidance, choose **Info** and view the information in the right panel.
      - If the question does not apply to your workload, choose **Question does not apply to this workload**.
      - Otherwise, select the best practices that you are currently following from the list.

If you are currently not following any of the best practices, choose **None of these**.

For additional guidance on any item, choose **Info** and view the information in the right panel.

b. (Optional) If one or more best practices do not apply to your workload, choose **Mark best practice(s) that don't apply to this workload** and select them. For each selected best practice, you can optionally select a reason and provide additional details.

c. (Optional) Use the **Notes** box to record information related to the question.

For example, you might describe why the question does not apply or provide additional details about the best practices selected.

d. Choose **Next** to continue to the next question.

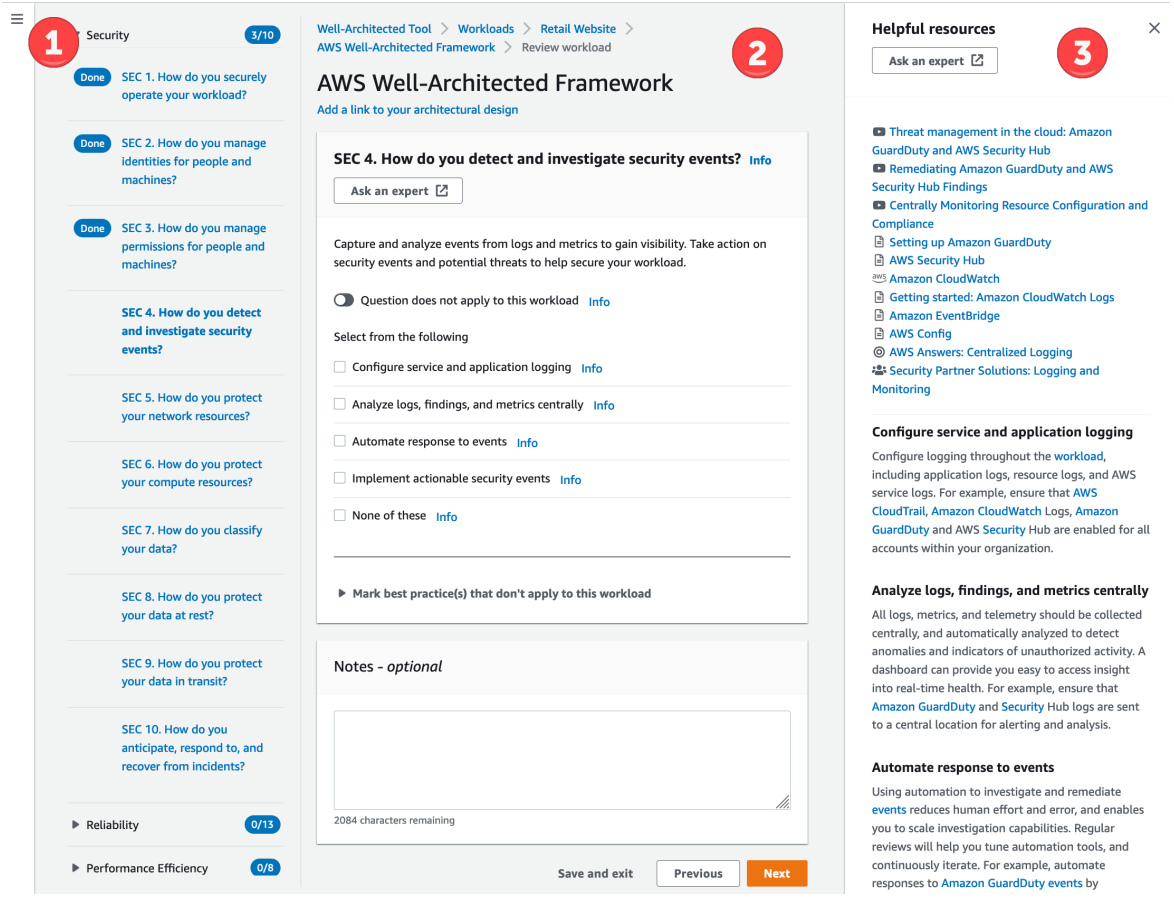
Repeat these steps for each question in each pillar.

3. Choose **Save and exit** at any time to save your changes and pause documenting your workload.

To return to the questions, go to the workload details page and choose **Continue reviewing**.

# Question page

The question page has three panels.




1. The left panel shows the questions for each pillar. Questions that you have answered are marked **Done**. The number of questions answered in each pillar is shown next to the pillar name.

You can navigate to questions in other pillars by choosing the pillar name and then choosing the question you want to answer.

2. The middle panel displays the current question. Select the best practices that you are following. Choose **Info** to get additional information about the question or a best practice. Choose **Ask an expert** to access the AWS re:Post community dedicated to AWS Well-Architected. AWS re:Post is a topic-based question-and-answer community replacement for AWS Forums. With re:Post, you can find answers, answer questions, join a group, follow popular topics, and vote on your favorite questions and answers.

   To optionally mark one or more best practices as not applicable, choose **Mark best practice(s) that don't apply to this workload** and select them.

   Use the buttons at the bottom of this panel to go to the next question, return to the previous question, or save your changes and exit.
3. The right panel displays additional information and helpful resources. Choose **Ask an expert** to access the AWS re:Post community dedicated to AWS Well-Architected. In this community, you can ask questions related to designing, building, deploying, and operating workloads on AWS.

# Saving a milestone

You can save a milestone at any time. A milestone records the current state of the workload.

**To save a milestone**

1. From the workload details page, choose **Save milestone**.
2. In the **Milestone name** box, enter a name for your milestone.

   **Note**
   The name must be between 3 and 100 characters. At least three characters must not be spaces. Milestone names associated with a workload must be unique. Spaces and capitalization are ignored when checking for uniqueness.
3. Choose **Save**.

After a milestone is saved, you cannot change the workload data that was captured in that milestone.

For more information, see Milestones (p. 37).

# Tutorial

This tutorial describes using AWS Well-Architected Tool to document and measure a workload. This example illustrates, step by step, how to define and document a workload for a retail ecommerce website.

**Topics**



## Step 1: Define a workload

You begin by defining a workload.

**To define a workload**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.

   **Note**
   The IAM user who documents the workload state must have full access permissions (p. 4) to AWS WA Tool.
2. In the **Define a workload** section, choose **Define workload**.
3. In the **Name** box, enter **`Retail Website - North America`** as the workload name.
4. In the **Description** box, enter a description for the workload.
5. In the **Review owner** box, enter the name of the person responsible for the workload review process.
6. In the **Environment** box, indicate that the workload is in a production environment.
7. Our workload runs on both AWS and at our local data center:

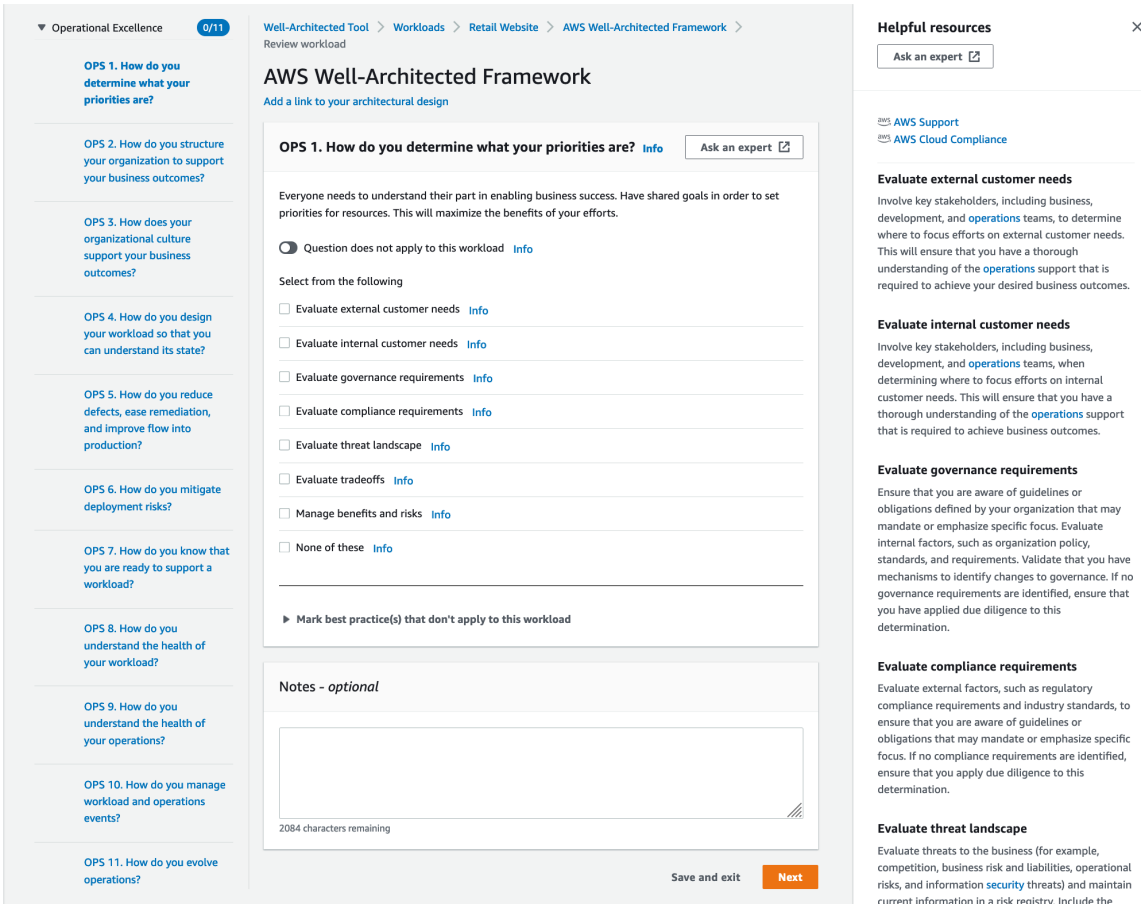
   a. Select **AWS Regions**, and choose the two Regions in North America where the workload runs.
   b. Also select **Non-AWS regions**, and enter a name for the local data center.
8. The **Account IDs** box is optional. Do not associate any AWS accounts with this workload.
9. The **Architectural diagram** box is optional. Do not associate an architectural diagram with this workload.
10. The **Industry type** and **Industry** boxes are optional and are not specified for this workload.
11. For this example, do not apply any tags to the workload. Choose **Next**.
12. For this example, apply the AWS Well-Architected Framework lens, which is automatically selected. Choose **Define workload** to save these values and define the workload.
13. After the workload is defined, choose **Start reviewing** to begin documenting the state of the workload.

## Step 2: Document the workload state

To document the state of the workload, you are presented with questions for the selected lens that span the pillars of the AWS Well-Architected Framework: operational excellence, security, reliability, performance efficiency, cost optimization, and sustainability.

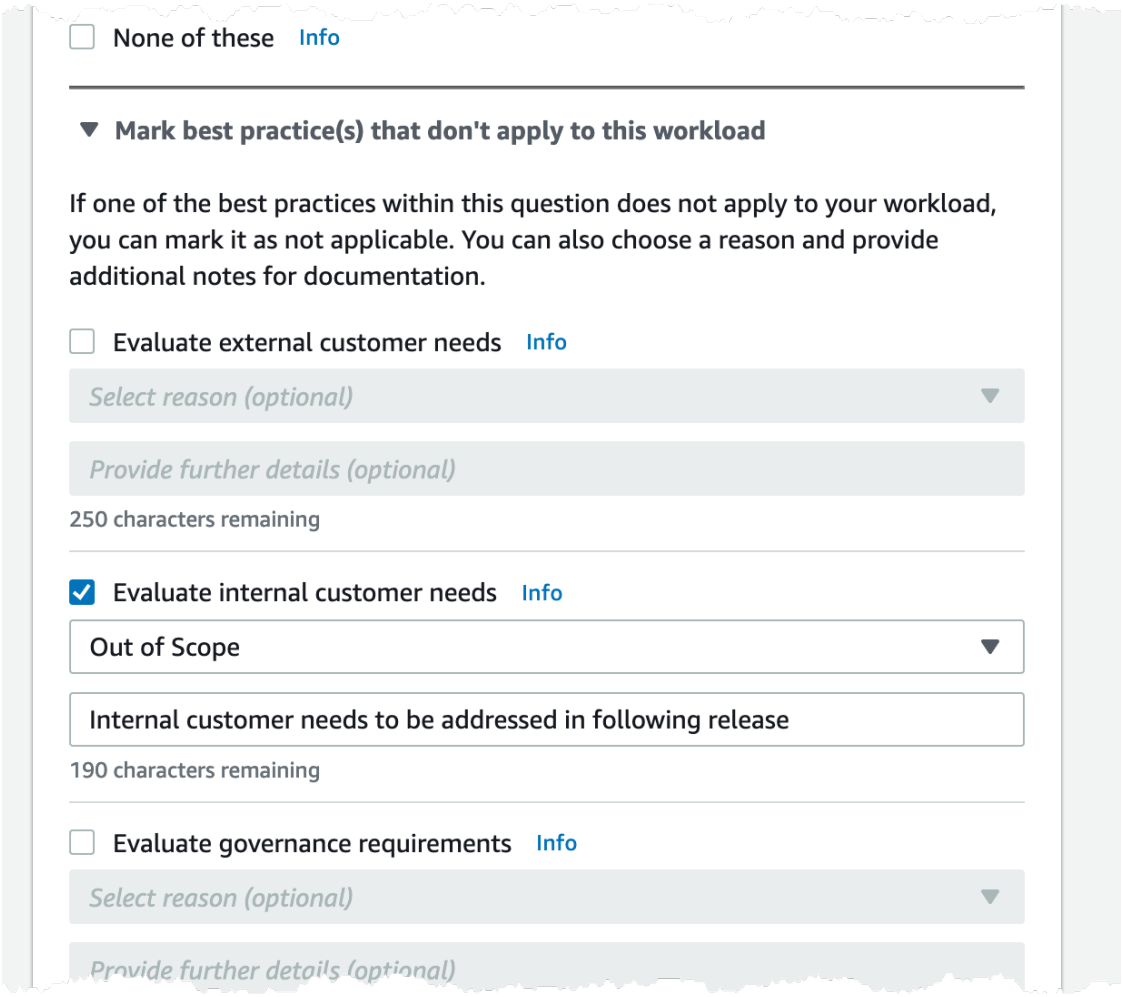
For each question, choose the best practices that you are following from the list provided. If you need details about a best practice, choose **Info** and view the additional information and resources in the right panel.

Choose **Ask an expert** to access the AWS re:Post community dedicated to AWS Well-Architected. In this community, you can ask questions related to designing, building, deploying, and operating workloads on AWS.




1. Choose **Next** to proceed to the next question. You can use the left panel to navigate to a different question in the same pillar or to a question in a different pillar.
2. If you choose **Question does not apply to this workload** or **None of these**, AWS recommends that you include the reason in the **Notes** box. These notes are included as part of the workload report and can be helpful in the future as changes are made to the workload.

   **Note**
   Optionally, you can mark one or more individual best practices as not applicable. Choose **Mark best practice(s) that don't apply to this workload** and select the best practice that does not apply. You can optionally select a reason and provide additional details. Repeat for each best practice that does not apply.




**Note**
You can pause this process at any time by choosing **Save and exit**.To resume later, open the AWS WA Tool console and choose **Workloads** in the left navigation pane.

3. Select the name of the workload to open the workload details page.
4. Choose **Continue reviewing** and then navigate to where you left off.
5. After you complete all of the questions, an overview page for the workload appears. You can review these details now or navigate to them later by choosing **Workloads** in the left navigation pane and selecting the workload name.

After documenting the state of your workload for the first time, you should save a milestone and generate a workload report.

A milestone captures the current state of the workload and enables you to measure progress as you make changes based on your improvement plan.

From the workload details page:

1. In the **Workload overview** section, choose the **Save milestone** button.
2. Enter **`Version 1.0 - initial review`** as the **Milestone name**.
3. Choose **Save**.

4. To generate a workload report, select the desired lens and choose **Generate report** and a PDF file is created. This file contains the state of the workload, the number of risks identified, and a list of suggested improvements.

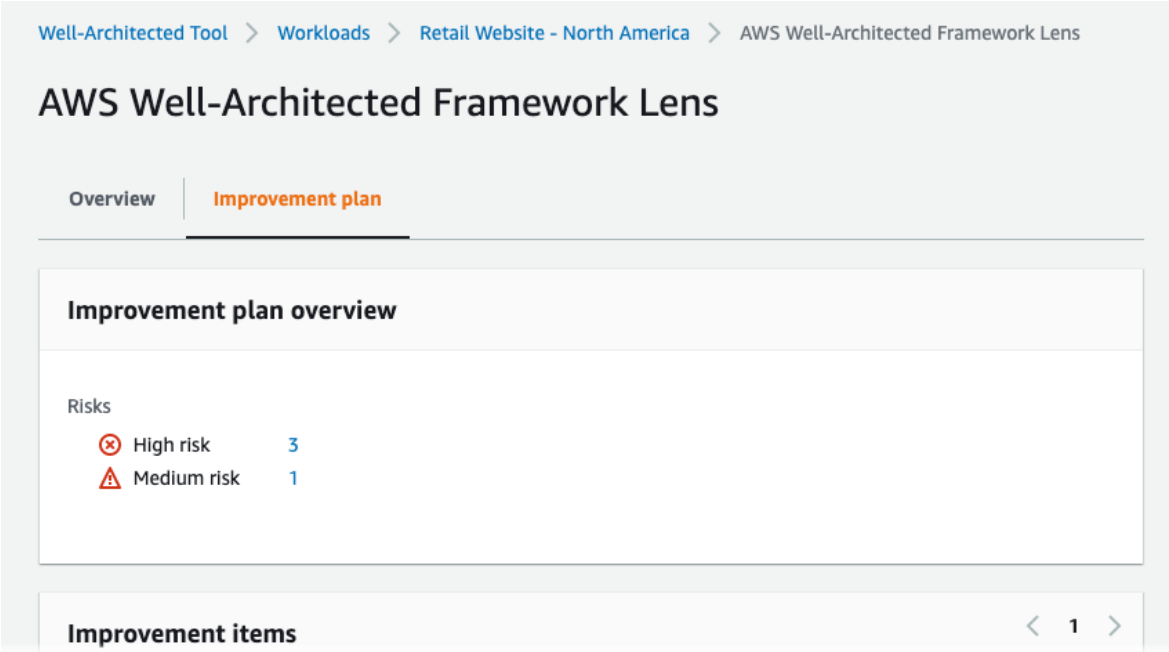
# Step 3: Review the improvement plan

Based on the best practices selected, AWS WA Tool identifies areas of high and medium risk as measured against the AWS Well-Architected Framework Lens.

To review the improvement plan:

1. Choose **AWS Well-Architected Framework** from the **Lenses** section of the **Overview** page.
2. Then choose **Improvement plan**.

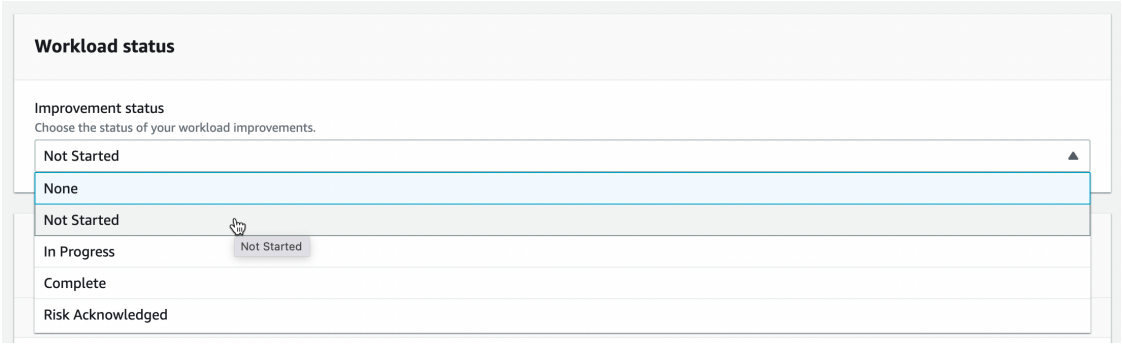
For this particular example workload, three high risk issues and one medium risk issue were identified by the AWS Well-Architected Framework Lens.




Update the **Improvement status** for the workload to indicate that improvements to the workload have not been started.

To change the **Improvement status**:

1. From the Improvement plan, click on the name of the workload (**`Retail Website - North America`**) in the breadcrumbs at the top of the page.
2. Click on the **Properties** tab.
3. Navigate to the **Workload status** section and select **Not Started** from the dropdown list.




4. Navigate back to the Improvement plan from the **Properties** tab by clicking on the **Overview** tab and then clicking on the **AWS Well-Architected Framework** link in the **Lenses** section. Then click on the **Improvement plan** tab at the top of the page.

The **Improvement items** section shows the recommended improvement items identified in the workload. The questions are ordered based on the pillar priority that is set, with any high risk issues listed first followed by any medium risk issues.

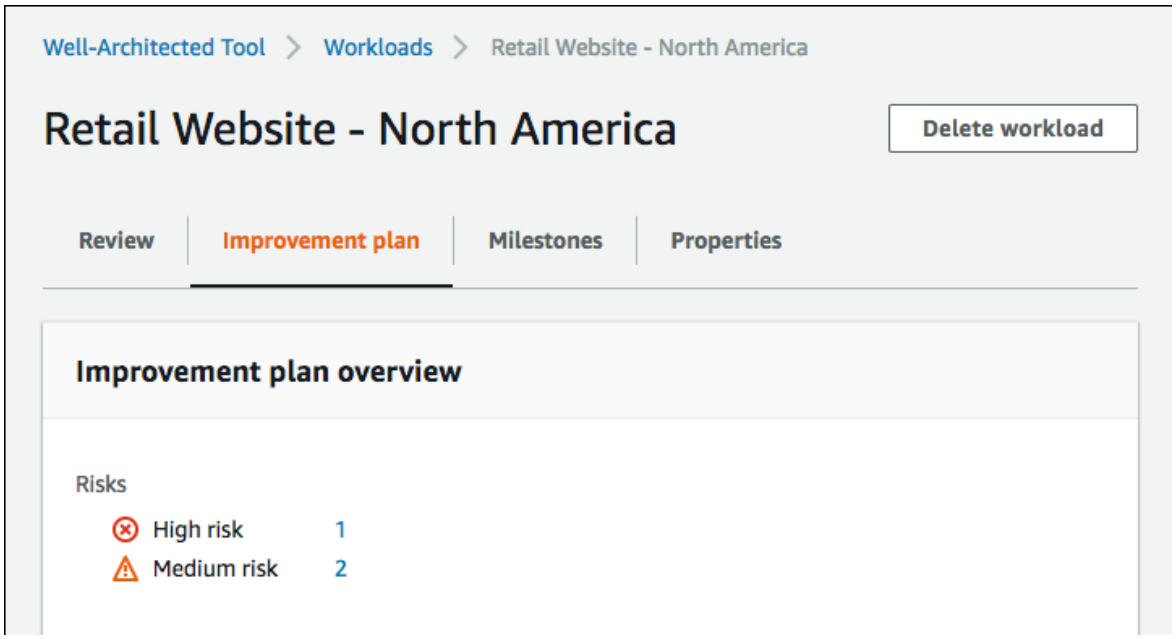
Expand **Recommended improvement items** to show the best practices for a question. Each recommended improvement action links to detailed expert guidance to help you eliminate, or at least mitigate, the risks identified.

# Step 4: Make improvements and measure progress

As part of this improvement plan, one of the high risk issues was addressed by adding Amazon CloudWatch and AWS Auto Scaling support to the workload.

From the **Improvement items** section:

1. Choose the pertinent question and update the selected best practices to reflect the changes. **Notes** are added to record the improvements.
2. Then choose **Save and exit** to update the state of the workload.
3. After making changes, you can return to the **Improvement plan** and see the effect those changes had on the workload. In this example, those actions have improved the risk profile — reducing the number of high risk issues from three to only one.




You can save a milestone at this point, and then go to **Milestones** to see how the workload has improved.

# Workloads

A workload is a collection of resources and code that delivers business value, such as a customer-facing application or a backend process.

A workload might consist of a subset of resources in a single AWS account or be a collection of multiple resources spanning multiple AWS accounts. A small business might have only a few workloads while a large enterprise might have thousands.

The **Workloads** page, available from the left navigation, provides information about your workloads and any workloads that have been shared with you.

The following information is displayed for each workload:

**Name**

The name of the workload.

**Owner**

The AWS account ID that owns the workload.

**Questions answered**

The number of questions answered.

**High risks**

The number of high risk issues (HRIs) identified.

**Medium risks**

The number of medium risk issues (MRIs) identified.

**Improvement status**

The improvement status that you have set for the workload:

- None
- Not Started
- In Progress
- Complete
- Risk Acknowledged

**Last updated**

Date and time that the workload was last updated.

After you choose a workload from the list:

- To review the details of the workload, choose **View details**.
- To change the properties of the workload, choose **Edit**.
- To manage sharing of the workload with other AWS accounts, IAM users, AWS Organizations, or organization units (OUs), choose **View details** and then **Shares**.
- To delete the workload and all of its milestones, choose **Delete**. Only the owner of the workload can delete it.

**Warning**
Deleting a workload cannot be undone. All data associated with the workload is deleted.

To define a new workload, choose **Define workload**. For details, see Defining a workload (p. 5).

# High Risk Issues (HRIs) and Medium Risk Issues (MRIs)

**High risk issues (HRIs)** identified in the AWS Well-Architected Tool are architectural and operational choices that AWS has found might result in significant negative impact to a business. These HRIs might affect organizational operations, assets, and individuals. **Medium risk issues (MRIs)** also might negatively impact business, but to a lesser extent. These issues are based on your responses in the AWS Well-Architected Tool. The corresponding best practices are widely applied by AWS and AWS customers. These best practices are the guidance defined by the AWS Well-Architected Framework and lenses.

**Note**
These are guidelines only and customers should evaluate and measure what impact not implementing the best practice would have on their business. If there are specific technical or business reasons that prevent applying a best practice to the workload, then the risk might be lower than indicated. AWS suggests that customers document these reasons, and how they affect the best practice, in the workload notes. For all identified HRIs and MRIs, AWS suggests customers implement the best practice as defined in the AWS Well-Architected Tool. If the best practice is implemented, indicate that the issue has been resolved by marking the best practice as met in the AWS Well-Architected Tool. If customers choose not to implement the best practice, AWS suggests that they document the applicable business level approval and reasons for not implementing it.

# Viewing a workload

You can view the details of workloads that you own and workloads that have been shared with you.

**To view a workload**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Workloads**.
3. Select the workload to view in one of the following ways:
   - Choose the name of the workload.
   - Select the workload and choose **View details**.

The workload details page is displayed.

**Note**
A required field, **Review owner**, was added to allow you to easily identify the primary person or group that is responsible for the review process.
The first time you view a workload that was defined before this field was added, you are notified of this change. Choose **Edit** to set the **Review owner** field and no further action is required. Choose **Acknowledge** to defer setting the **Review owner** field. For the next 60 days, a banner is displayed to remind you that the field is blank. To remove the banner, edit your workload and specify a **Review owner**.

If you do not set the field by the specified date, your access to the workload is restricted. You can continue to view the workload and delete it, but you cannot edit it, except to set the **Review owner** field. Shared access to the workload is not affected while your access is limited.

# Editing a workload

You can edit the details of a workload that you own.

**To edit a workload**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Workloads**.
3. Select the workload that you want to edit and choose **Edit**.
4. Make your changes to the workload.

   For a description of each of the fields, see Defining a workload (p. 5).
5. Choose **Save** to save your changes to the workload.

   If a required field is blank or if a specified value is not valid, you must correct the issue before your updates to the workload are saved.

# Sharing a workload

You can share a workload that you own with other AWS accounts, IAM users, an organization, and organization units (OUs) in the same AWS Region.

**Note**
You can only share workloads within the same AWS Region.

**To share a workload with other AWS accounts and IAM users**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Workloads**.
3. Select a workload that you own in one of the following ways:
   - Choose the name of the workload.
   - Select the workload and choose **View details**.
4. Choose **Shares**. Then choose **Create** and **Create shares to IAM users or accounts** to create a workload invitation.
5. Enter the 12-digit AWS account ID or the ARN of the IAM user that you want to share the workload with.
6. Choose the permission that you want to grant.

   **Read-Only**

   Provides read-only access to the workload.

   **Contributor**

   Provides update access to answers and their notes, and read-only access to the rest of the workload.
7. Choose **Create** to send a workload invitation to the specified AWS account or IAM user.

If the workload invitation is not accepted within seven days, the invitation is automatically expired.

If an IAM user and the user's AWS account both have workload invitations, the workload invitation with the highest level permission is applied to the IAM user.

**Important**
Before sharing a workload with an organization or organization units (OUs), you must enable AWS Organizations access (p. 58).

### To share a workload with your organization or OUs

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Workloads**.
3. Select a workload that you own in one of the following ways:
   - Choose the name of the workload.
   - Select the workload and choose **View details**.
4. Choose **Shares**. Then choose **Create** and **Create shares to Organizations**.
5. On the **Create workload share** page, choose whether to grant permissions to the entire organization, or to one or more OUs.
6. Choose the permission that you want to grant.

   **Read-Only**

   Provides read-only access to the workload.

   **Contributor**

   Provides update access to answers and their notes, and read-only access to the rest of the workload.
7. Choose **Create** to share the workload.

To see who has shared access to a workload, choose **Shares** from the Workload details (p. 21) page.

To prevent an entity from sharing workloads, attach a policy that denies `wellarchitected:CreateWorkloadShare` actions.

You can also share custom lenses that you own with other AWS accounts, IAM users, your organization, and OUs in the same AWS Region. For details, refer to Sharing a custom lens (p. 29).

# Sharing considerations

A workload can be shared with up to 20 different AWS accounts and IAM users. A workload can only be shared with accounts and users that are in the same AWS Region as the workload.

To share a workload in a Region introduced after March 20, 2019, both you and the shared AWS account must enable the Region in the AWS Management Console. For more information, refer to AWS Global Infrastructure.

You can share a workload with an AWS account, individual IAM users in an account, or both. When you share a workload with an AWS account, all IAM users in that account are given access to the workload. If only specific users in an account require access, follow the best practice of granting least privilege and share the workload individually with those IAM users.

If both an AWS account and an IAM user in the account have workload invitations, the workload invitation with the highest level permissions determines the user's permission to the workload. If you delete the workload invitation for the IAM user, the user's access is determined by the workload

invitation for the AWS account. Delete both workload invitations to remove the user's access to the workload.

Before sharing a workload with an organization or one or more organization units (OUs), you must enable AWS Organizations access.

If you share a workload with both an organization and one or more OUs, the workload invitation with the highest level permissions determines the account's permission to the workload.

**To enable AWS Organizations sharing**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Settings**.
3. Choose **Enable AWS Organizations support**.
4. Choose **Save settings**.

# Deleting shared access

You can delete a workload invitation. Deleting a workload invitation removes shared access to the workload.

**To delete shared access to a workload**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Workloads**.
3. Select the workload in one of the following ways:
   - Choose the name of the workload.
   - Select the workload and choose **View details**.
4. Choose **Shares**.
5. Select the workload invitation to delete and choose **Delete**.
6. Choose **Delete** to confirm.

If an IAM user and the user's AWS account have workload invitations, you must delete both workload invitations to remove the user's permission to the workload.

# Modifying shared access

You can modify a pending or accepted workload invitation.

**To modify shared access to a workload**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Workloads**.
3. Select a workload that you own in one of the following ways:
   - Choose the name of the workload.
   - Select the workload and choose **View details**.
4. Choose **Shares**.
5. Select the workload invitation to modify and choose **Edit**.

6. Choose the new permission that you want to grant to the AWS account or IAM user.

   **Read-Only**

   Provides read-only access to the workload.

   **Contributor**

   Provides update access to answers and their notes, and read-only access to the rest of the workload.
7. Choose **Save**.

If the modified workload invitation is not accepted within seven days, it's automatically expired.

## Accepting and rejecting workload invitations

A workload invitation is a request to share a workload that is owned by another AWS account. If you accept the workload invitation, the workload is added to your **Workloads** and **Dashboard** pages. If you reject the workload invitation, it's removed from the workload invitation list.

You have seven days to accept a workload invitation. If you do not accept the invitation within seven days, it's automatically expired.

**Note**
Workloads can only be shared within the same AWS Region.

**To accept or reject a workload invitation**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Workload invitations**.
3. Select the workload invitation to accept or reject.
   - To accept the workload invitation, choose **Accept**.

     The workload is added to the **Workloads** and **Dashboard** pages.
   - To reject the workload invitation, choose **Reject**.

     The workload invitation is removed from the list.

To reject shared access after a workload invitation has been accepted, choose **Reject share** from the Workload details (p. 21) page for the workload.

# Deleting a workload

You can delete a workload when it's no longer needed. Deleting a workload removes all data associated with the workload including any milestones and workload share invitations. Only the owner of a workload can delete it.

**Warning**
Deleting a workload cannot be undone. All data associated with the workload is permanently removed.

**To delete a workload**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.

2. In the left navigation pane, choose **Workloads**.
3. Select the workload you want to delete and choose **Delete**.
4. In the **Delete** window, choose **Delete** to confirm the deletion of the workload and its milestones.

To prevent an entity from deleting workloads, attach a policy that denies `wellarchitected:DeleteWorkload` actions.

# Generating a workload report

You can generate a workload report for a lens. The report contains your responses to the workload questions, your notes, and the current number of high and medium risks identified. If a question has one or more risks identified, the improvement plan for that question lists actions to take to mitigate those risks.

A report enables you to share details about your workload with others who do not have access to AWS Well-Architected Tool.

**To generate a workload report**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Workloads**.
3. Select the desired workload and choose **View details**.
4. Select the lens you want to generate a report for and choose **Generate report**.

   The report is generated and you can download or view it.

# Workload details

The workload details page provides information about your workload including its milestones, improvement plan, and any workload shares. Use the tabs at the top of the page to navigate to the different detail sections.

To delete the workload, choose **Delete workload**. Only the owner of a workload can delete it.

To remove your access to a shared workload, choose **Reject share**.

**Topics**



## Overview tab

When you initially view a workload, the **Overview** tab is the first information displayed. This tab provides the overall state of your workload followed by the state of each lens.

If you have not completed all of the questions, a banner appears to remind you to start or continue documenting your workload.

The **Workload overview** section shows the current overall state of the workload and any **Workload notes** that you have entered. Choose **Edit** to update the state or notes.

To capture the current state of the workload, choose **Save milestone**. Milestones are immutable and cannot be changed after they are saved.

To continue documenting the state of the workload, choose **Start reviewing** and select the desired lens.

# Milestones tab

To display the milestones for your workload, choose the **Milestones** tab.

After you select a milestone, choose **Generate report** to create the workload report associated with the milestone. The report contains the responses to the workload questions, your notes, and the number of high and medium risks in the workload at the time that the milestone was saved.

You can view details about the state of your workload at the time of a specific milestone by either:

- Choosing the name of the milestone.
- Selecting the milestone and choosing **View milestone**.

# Properties tab

To display the properties of your workload, choose the **Properties** tab. Initially, these properties are the values that were specified when the workload was defined. Choose **Edit** to make changes. Only the owner of the workload can make changes.

For descriptions of the properties, see Defining a workload (p. 5).

# Shares tab

To display or modify your workload invitations, choose the **Shares** tab. This tab is only displayed for the owner of a workload.

The following information is displayed for each AWS account and IAM user that has shared access to the workload:

**Principal**

The AWS account ID or IAM user ARN with shared access to the workload.

**Status**

The status of the workload invitation.

- Pending

  The invitation is waiting to be accepted or rejected. If a workload invitation is not accepted within seven days, it's automatically expired.
- Accepted

  The invitation was accepted.
- Rejected

  The invitation was rejected.
- Expired

  The invitation was not accepted or rejected within seven days.

**Permission**

The permission granted to the AWS account or IAM user.

- Read-Only

  The principal has read-only access to the workload.
- Contributor

  The principal can update answers and their notes, and has read-only access to the rest of the workload.

**Permission details**

Detailed description of the permission.

To share the workload with another AWS account or IAM user in the same AWS Region, choose **Create**. A workload can be shared with up to 20 different AWS accounts and IAM users.

To delete a workload invitation, select the invitation and choose **Delete**.

To modify a workload invitation, select the invitation and choose **Edit**.

# Lenses

Lenses provide a way for you to consistently measure your architectures against best practices and identify areas for improvement. The **AWS Well-Architected Framework Lens** is automatically applied when a workload is defined.

A workload can have one or more lenses applied. Each lens has its own set of questions, best practices, notes, and improvement plan.

The following lenses are provided by AWS:

- The **Serverless Lens** focuses on designing, deploying, and architecting your serverless application workloads in the AWS Cloud. This lens covers scenarios such as RESTful microservices, mobile app backends, stream processing, and web applications. Using this lens helps you apply best practices when building serverless application workloads on AWS.

  For more information, see the Serverless Applications Lens whitepaper.
- The **SaaS Lens** focuses on designing, deploying, and architecting your software as a service (SaaS) workloads in the AWS Cloud. Using this lens helps you apply best practices when building SaaS workloads on AWS.

  For more information, see the SaaS Lens whitepaper.
- The **FTR Lens** is designed for independent software vendors (ISVs) preparing for a Foundational Technical Review (FTR) in the AWS Partner Network (APN). It provides a set of specific questions for ISVs to perform a workload self-assessment before requesting a review.

Up to 20 lenses can be added to a workload.

You also can create and use your own custom lenses (p. 27), or use custom lenses that have been shared with you.

If a lens is removed from a workload, the data associated with the lens is retained. The data is restored if you add the lens back to the workload.

For more information about AWS provided lenses, visit AWS Well-Architected Lenses.

# Adding a lens to a workload

**To add a lens to a workload**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Workloads**.
3. Select the desired workload and choose **View details**.
4. Select the lens to add and choose **Save**.

Up to 20 lenses can be added to a workload.

**Disclaimer**
By accessing and/or applying custom lenses created by another AWS user or account, you acknowledge that custom lenses created by other users and shared with you are Third Party Content as defined in the AWS Customer Agreement.

# Removing a lens from a workload

**To remove a lens from a workload**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Workloads**.
3. Select the desired workload and choose **View details**.
4. Unselect the lens that you want to remove and choose **Save**.

   The AWS Well-Architected Framework Lens cannot be removed from a workload.

The data associated with the lens is retained. If the lens is added back to the workload, the data is restored.

# Lens details

To view details about a lens, select the lens.

## Overview tab

The **Overview** tab provides general information about the lens, such as the number of questions answered. From this tab, you can continue reviewing a workload, generate a report, or edit the lens notes.

## Improvement plan tab

The **Improvement Plan** tab provides a list of recommended actions to improve your workload. You can filter recommendations based on risk and pillar.

## Shares tab

For a custom lens, the **Shares** tab provides a list of IAM principals that the lens has been shared with.

# Custom lenses

You can create custom lenses with your own pillars, questions, best practices, and improvement plan. You apply custom lenses to a workload in the same way that you apply AWS provided lenses. You can also share custom lenses that you create with other AWS accounts, and custom lenses owned by others can be shared with you.

You can tailor the questions in a custom lens to be specific to a particular technology, help you meet the governance needs within your organization, or extend the guidance provided by the Well-Architected

Framework and the AWS lenses. Like the existing lenses, you can track progress over time by creating milestones, and provide periodic status by generating reports.

**Topics**



# Viewing custom lenses

You can view the details of custom lenses that you own and custom lenses that have been shared with you.

### To view a lens

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Custom lenses**.
3. Choose which custom lenses you want to view:
   - **Owned by me** – Shows custom lenses that you have created.
   - **Shared with me** – Shows custom lenses that have been shared with you.
4. Select the custom lens to view in one of the following ways:
   - Choose the name of the lens.
   - Select the lens and choose **View details**.

The Lens details (p. 25) page is displayed.

The **Custom lenses** page has the following fields:

**Name**

The name of the lens.

**Owner**

The AWS account ID that owns the custom lens.

**Status**

A status of **PUBLISHED** means that the custom lens has been published and can be applied to workloads or shared with other AWS accounts.

A status of **DRAFT** means that the custom lens has been created but has not yet been published. A custom lens must be published before it can be applied to workloads or shared.

**Version**

The version name of the custom lens.

**Last updated**

Date and time that the custom lenses was last updated.

# Creating a custom lens

### To create a custom lens

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Custom lenses**.
3. Choose **Create custom lens**.
4. Choose **Download file** to download the JSON template file.
5. Open the JSON template file with your favorite text editor and add the data for your custom lens. This data includes your pillars, questions, best practices, and improvement plan links.

   Refer to Lens format specification (p. 31) for details. A custom lens cannot exceed 500 KB in size.
6. Choose **Choose file** to select your JSON file.
7. (Optional) In the **Tags** section, add any tags you want to associate with the custom lens.
8. Choose **Submit & Preview** to preview the custom lens, or **Submit** to submit the custom lens without previewing.

   If you choose to **Submit & Preview** your custom lens, you can select **Next** to navigate through the lens preview, or select **Exit Preview** to go back to **Custom lenses**.

If validation fails, edit your JSON file and try creating the custom lens again.

After AWS WA Tool validates your JSON file, your custom lens is displayed in **Custom lenses**.

After a custom lens has been created, it's in **DRAFT** status. You must publish the lens (p. 28) before it can be applied to workloads or shared with other AWS accounts.

You can create up to 15 custom lenses in an AWS account.

**Disclaimer**
Do not include or gather personal identifiable information (PII) of end users or other identifiable individuals in or via your custom lenses. If your custom lens or those shared with you and used in your account do include or collect PII you are responsible for: ensuring that the included PII is processed in accordance with applicable law, providing adequate privacy notices, and obtaining necessary consents for processing such data.

# Previewing a custom lens

### To preview a custom lens

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Custom lenses**.
3. Only lenses in a **DRAFT** status can be previewed. Select the desired **DRAFT** custom lens and choose **Preview experience**.
4. Choose **Next** to navigate through the lens preview.
5. (Optional) You can review your **Improvement plan** by selecting best practices within each question in the preview, and choosing **Update based on answers** to test your risk logic. If there are changes needed, you can update the Risk Rules (p. 34) in your JSON template before publishing.

6. Choose **Exit Preview** to go back to the custom lens.

   **Note**
   You can also preview a custom lens by selecting **Submit & Preview** when Creating a custom lens (p. 27).

# Publishing a custom lens for the first time

**To publish a custom lens**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Custom lenses**.
3. Select the desired custom lens and choose **Publish lens**.
4. In the **Version name** box, enter a unique identifier for the version change. This value can be up to 32 characters and must only contain alphanumeric characters and periods (".").
5. Choose **Publish custom lens**.

   After a custom lens has been published, it's in **PUBLISHED** status.

The custom lens can now be applied to workloads or shared with other AWS accounts or IAM users.

# Publishing an update to a custom lens

**To publish an update to an existing custom lens**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Custom lenses**.
3. Select the desired custom lens and choose **Edit**.
4. If you do not have an updated JSON file ready, choose **Download file** to download a copy of the current custom lens. Edit the downloaded JSON file with your favorite text editor and make your desired changes.
5. Choose **Choose file** to select your updated JSON file and choose **Submit & Preview** to preview the custom lens, or **Submit** to submit the custom lens without previewing.

   A custom lens cannot exceed 500 KB in size.

   After AWS WA Tool validates your JSON file, your custom lens is displayed in **Custom lenses** in **DRAFT** status.
6. Select the custom lens again and choose **Publish lens**.
7. Choose **Review changes before publishing** to verify that the changes made to your custom lens are correct. This includes validating:

   - The name of the custom lens
   - The pillar names
   - The new, updated, and deleted questions

   Choose **Next**.
8. Specify the type of version change.

**Major version**

Indicates that substantial changes have been made to the lens. Use for changes that impact the meaning of the custom lens.

Any workloads with the lens applied will be notified that a new version of the custom lens is available.

Major version changes are *not* automatically applied to workloads using the lens.

**Minor version**

Indicates that minor changes have been made to the lens. Use for small changes, such as text changes or updates to the URL links.

Minor version changes are automatically applied to workloads using the custom lens.

Choose **Next**.

9. In the **Version name** box, enter a unique identifier for the version change. This value can be up to 32 characters and must only contain alphanumeric characters and periods (".").
10. Choose **Publish custom lens**.

    After a custom lens has been published, it's in **PUBLISHED** status.

The updated custom lens can now be applied to workloads or shared with other AWS accounts or IAM users.

If the update is a *major version change*, any workloads with the previous version of the lens applied will be notified that a new version is available and given the option to upgrade.

Minor version updates are automatically applied without any notification.

You can create up to 100 versions of a custom lens.

# Sharing a custom lens

You can share a custom lens with other AWS accounts, IAM users, AWS Organizations, and organization units (OUs).

### To share a custom lens with other AWS accounts and IAM users

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Custom lenses**.
3. Select the custom lens to be shared and choose **View details**.
4. On the Lens details (p. 25) page, choose **Shares**. Then choose **Create** and **Create shares to IAM users or accounts** to create a lens share invitation.
5. Enter the 12-digit AWS account ID or the ARN of the IAM user that you want to share the custom lens with.
6. Choose **Create** to send a lens share invitation to the specified AWS account or IAM user.

You can share a custom lenses with up to 300 AWS accounts or IAM users.

If the lens share invitation is not accepted within seven days, the invitation is automatically expired.

**Important**

**Important**
Before sharing a custom lens with an organization or organization units (OUs), you must enable AWS Organizations access (p. 58).

**To share a custom lens with your organization or OUs**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Custom lenses**.
3. Select the custom lens to be shared.
4. On the Lens details (p. 25) page, choose **Shares**. Then choose **Create** and **Create shares to Organizations**.
5. On the **Create custom lens share** page, choose whether to grant permissions to the entire organization, or to one or more OUs.
6. Choose **Create** to share the custom lens.

To see who has shared access to a custom lens, choose **Shares** from the Lens details (p. 25) page.

**Disclaimer**
By sharing your custom lenses with other AWS accounts, you acknowledge that AWS will make your custom lenses available to those other accounts. Those other accounts may continue to access and use your shared custom lenses even if you delete the custom lenses from your own AWS account or terminate your AWS account.

# Adding tags to a custom lens

**To add tags to a custom lens**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Custom lenses**.
3. Select the custom lens you want to update.
4. In the **Tags** section, choose **Manage Tags**.
5. Select **Add new tag** and enter the **Key** and **Value** for each tag you want to add.
6. Select **Save**.

To remove a tag, choose **Remove** next to the tag you want to remove.

# Deleting a custom lens

**To delete a custom lens**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
2. In the left navigation pane, choose **Custom lenses**.
3. Select the custom lens to be deleted and choose **Delete**.
4. Choose **Delete**.

   Existing workloads with the lens applied are notified that the custom lens has been deleted, but can continue to use it. The custom lens can no longer be applied to new workloads.

**Disclaimer**
By sharing your custom lenses with other AWS accounts, you acknowledge that AWS will make your custom lenses available to those other accounts. Those other accounts may continue to access and use your shared custom lenses even if you delete the custom lenses from your own AWS account or terminate your AWS account.

# Lens format specification

Lenses are defined using a specific JSON format. When you start to create a custom lens, you have the option to download a template JSON file. You can use this file as the basis for your custom lenses as it defines the basic structure for the pillars, questions, best practices, and improvement plan.

## Lens section

This section defines the attributes for the custom lens itself. This is its name and description.

- `schemaVersion`: The version of the custom lens schema to use. Set by the template, do not change.
- `name`: Name of the lens. The name can be up to 128 characters.
- `description`: Text description of the lens. This text is displayed when selecting lenses to add during workload creation, or when selecting a lens to apply to an existing workload later. The description can be up to 2048 characters.

```
   "schemaVersion": "2021-11-01",
   "name": "Company Policy ABC",
   "description": "This lens provides a set of specific questions to assess compliance
with company policy ABC-2021 as revised on 2021/09/01.",
```

## Pillars section

This section defines the pillars associated with the custom lens. You can map your questions to the pillars of the AWS Well-Architected Framework, define your own pillars, or both.

You can define up to 10 pillars in a custom lens.

- `id`: ID for the pillar. The ID can be between 3 and 128 characters and contain only alphanumeric and underscore ("_") characters. The IDs used in a pillar must be unique.

  When mapping your questions to the pillars of the Framework, use the following IDs:

  - `operationalExcellence`
  - `security`
  - `reliability`
  - `performance`
  - `costOptimization`
  - `sustainability`
- `name`: Name of the pillar. The name can be up to 128 characters.

```
   "pillars": [
      {
          "id": "company_Privacy",
          "name": "Privacy Excellence",
          .
          .
          .
```

```
        },
        {
            "id": "company_Security",
            "name": "Security",
            .
            .
            .
        }
    ]
```

## Questions section

This section defines the questions associated with a pillar.

You can define up to 20 questions in a pillar in a custom lens.

- `id`: ID for the question. The ID can be from 3 to 128 characters and contain only alphanumeric and underscore ("_") characters. The IDs used in a question must be unique.
- `title`: Title of the question. The title can be up to 128 characters.
- `description`: Describes the question in more detail. The description can be up to 2048 characters.
- `helpfulResource displayText`: Optional. Text that provides helpful information about the question. The text can be up to 2048 characters. Must be specified if `helpfulResource url` is specified.
- `helpfulResource url`: Optional. A URL resource that explains the question in more detail. The URL must start with `http://` or `https://`.

```
"questions": [
    {
        "id": "privacy01",
        "title": "How do you ensure HR conversations are private?",
        "description": "Career and benefits discussions should occur on secure channels
 only and be audited regularly for compliance.",
        "helpfulResource": {
            "displayText": "This is helpful text for the first question",
            "url": "https://example.com/poptquest01_help.html"
        },
        .
        .
        .
    },
    {
        "id": "privacy02",
        "title": "Is your team following the company privacy policy?",
        "description": "Our company requires customers to opt-in to data use and does not
 disclose customer data to third parties either individually or in aggregate.",
        "helpfulResource": {
            "displayText": "This is helpful text for the second question",
            "url": "https://example.com/poptquest02_help.html"
        },
        .
        .
        .
    }
]
```

## Choices section

This section defines the choices that are associated with a question.

You can define up to 15 choices for a question in a custom lens.

- `id`: ID for the choice. The ID can be between 3 and 128 characters and contain only alphanumeric and underscore ("_") characters. The IDs used in a choice must be unique. Adding a choice with a suffix of `_no` will act as a `None of these` choice for the question.
- `title`: Title of the choice. The title can be up to 128 characters.
- `helpfulResource displayText`: Optional. Text that provides helpful information about a choice. The text can be up to 2048 characters. Must be included if `helpfulResource url` is specified.
- `helpfulResource url`: Optional. A URL resource that explains the choice in more detail. The URL must start with `http://` or `https://`.
- `improvementPlan displayText`: Text that describes how a choice can be improved upon. The text can be up to 2048 characters. An `improvementPlan` is required for each choice, except for a `None of these` choice.
- `improvementPlan url`: Optional. A URL resource that can help with improvement. The URL must start with `http://` or `https://`.
- `additionalResources type`: Optional. The type of additional resources. Value can be either `HELPFUL_RESOURCE` or `IMPROVEMENT_PLAN`.
- `additionalResources content`: Optional. Specifies the `displayText` and `url` values for the additional resource. Up to five additional helpful resources and up to five additional improvement plan items can be specified for a choice.
  - `displayText`: Optional. Text that describes the helpful resource or improvement plan. The text can be up to 2048 characters. Must be included if `url` is specified.
  - `url`: Optional. A URL resource for the helpful resource or improvement plan. The URL must start with `http://` or `https://`.

```
   "choices": [
        {
            "id": "choice_1",
            "title": "Option 1",
            "helpfulResource": {
                "displayText": "This is helpful text for the first choice",
                "url": "https://example.com/popt01_help.html"
            },
            "improvementPlan": {
                "displayText": "This is text that will be shown for improvement of this
choice.",
                "url": "https://example.com/popt01_iplan.html"
            }
        },
        {
            "id": "choice_2",
            "title": "Option 2",
            "helpfulResource": {
                "displayText": "This is helpful text for the second choice",
                "url": "https://example.com/hr_manual_CORP_1.pdf"
            },
            "improvementPlan": {
                "displayText": "This is text that will be shown for improvement of this
choice.",
                "url": "https://example.com/popt02_iplan_01.html"
            }
            "additionalResources":[
               {
                 "type": "HELPFUL_RESOURCE",
                 "content": [
                   {
                     "displayText": "This is the second set of helpful text for this
choice.",
```

```
                    "url": "https://example.com/hr_manual_country.html"
                  },
                  {
                    "displayText": "This is the third set of helpful text for this
choice.",
                    "url": "https://example.com/hr_manual_city.html"
                  }
                ]
              },
              {
                "type": "IMPROVEMENT_PLAN",
                "content": [
                  {
                    "displayText": "This is additional text that will be shown for
improvement of this choice.",
                    "url": "https://example.com/popt02_iplan_02.html"
                  },
                  {
                    "displayText": "This is the third piece of improvement plan text.",
                    "url": "https://example.com/popt02_iplan_03.html"
                  }
                  {
                    "displayText": "This is the fourth piece of improvement plan text.",
                    "url": "https://example.com/popt02_iplan_04.html"
                  }
                ]
              }
            ]
        },
        {
             "id": "option_no",
             "title": "None of these",
             "helpfulResource": {
               "displayText": "Choose this if your workload does not follow these best
practices.",
               "url": "https://example.com/popt02_iplan_none.html"
             }

           }
```

## Risk Rules section

This section defines how the choices selected determine the risk level.

You can define a maximum of three risk rules per question, one for each level of risk.

- `condition`: A Boolean expression of the choices that maps to a risk level for the question, or `default`.

  There must be a `default` risk rule for each question.
- `risk`: Indicates the risk associated with the condition. Valid values are `HIGH_RISK`, `MEDIUM_RISK`, and `NO_RISK`.

The order of your risk rules is significant. The first `condition` that evaluates to `true` sets the risk for the question. A common pattern for implementing risk rules is to start with your least risky (and typically most granular) rules and work your way down to your most risky (and least specific) rules.

For example:

```
"riskRules": [
```

```
        {
          "condition": "choice_1 && choice_2 && choice_3",
          "risk": "NO_RISK"
        },
        {
          "condition": "((choice_1 || choice_2) && choice_3) || (!choice_1 && choice_3)",
          "risk": "MEDIUM_RISK"
        },
        {
          "condition": "default",
          "risk": "HIGH_RISK"
        }
]
```

If the question has three choices (`choice_1`, `choice_2`, and `choice_3`), these risk rules result in the following behavior:

- If all three choices are selected, there is no risk.
- If either `choice_1` or `choice_2` is selected **and** `choice_3` is selected, there is medium risk.
- If `choice_1` is **not** selected but `choice_3` is selected, there is also medium risk.
- If none of these prior conditions were true, there is high risk.

# Lens upgrades

The AWS Well-Architected Framework Lens and other lenses provided by AWS are updated as new services are introduced, existing best practices for cloud-based systems are refined, and new best practices are added. When a new version of a lens is made available, AWS WA Tool is upgraded to reflect the latest best practices. Any new workloads that are defined use the new version of the lens.

A lens upgrade also occurs when a custom lens that you have applied to a workload has a new major version published.

A lens upgrade can consist of any combination of:

- Adding new questions or best practices
- Removing old questions or practices that are no longer recommended
- Updating existing questions or best practices
- Adding or removing pillars

Your answers to existing questions are retained.

> **Note**
> You cannot undo a lens upgrade. After a workload has been upgraded to the latest lens version, you cannot go back to the previous version of the lens.

## Notifications

When a new version of a lens is available, a banner appears at the top of the **Workloads** page to notify you.

Choose **View available upgrades** for a list of workloads that can be upgraded.

If you view a specific workload, you will also see a banner indicating that a new lens version is available.

# Selecting a lens upgrade

The **Lens upgrades** page displays information for each workload that is not using the most current lens version.

The following information is displayed for each workload:

**Workload**

The name of the workload.

**Lens**

The name of the lens.

**Notification type**

The type of upgrade notification.

- **Not current** – The workload is using a version of the lens that is no longer current. Upgrade to the current lens version for better guidance.
- **Deprecated** – The workload is using a version of the lens that no longer reflects best practices. Upgrade to the current lens version.
- **Deleted** – The workload is using a lens that has been deleted by its owner.

**Version in use**

The lens version currently used for the workload.

**Current available version**

The lens version available for upgrade, or **None** if the lens has been deleted.

To upgrade the lens associated with a workload, select the workload and choose **Upgrade lens version**.

# Upgrading a lens

After you select a workload to upgrade, information about what changed in each pillar is displayed.

Before upgrading the lens, a milestone is created to save the state of your existing workload for future reference. Enter a unique name for the milestone.

> **Note**
> You cannot undo a lens upgrade. After a workload has been upgraded to the latest lens version, you cannot go back to the previous version of the lens.

To upgrade the lens for the selected workload, choose **I understand and accept these changes**.

Repeat these steps for each workload that you want to upgrade.

# Milestones

A milestone records the state of a workload at a particular point in time.

Save a milestone after you initially complete all the questions associated with a workload. As you change your workload based on items in your improvement plan, you can save additional milestones to measure progress.

A best practice is to save a milestone every time you make improvements to a workload.

## Saving a milestone

A milestone records the current state of a workload. The owner of a workload can save a milestone at any time.

**To save a milestone**

1. From the workload details page, choose **Save milestone**.
2. In the **Milestone name** box, enter a name for your milestone.

   **Note**
   The name must be between 3 and 100 characters. At least three characters must not be spaces. Milestone names associated with a workload must be unique. Spaces and capitalization are ignored when checking for uniqueness.
3. Choose **Save** to save the milestone.

After a milestone is saved, you cannot change the workload data that was recorded. When you delete a workload, its associated milestones are also deleted.

## Viewing milestones

You can view milestones for a workload in the following ways:

- On the workload details page, choose **Milestones** and choose the milestone you want to view.
- On the **Dashboard** page, choose the workload and in the **Milestones** section, choose the milestone you want to view.

## Generating a milestone report

You can generate a milestone report. The report contains the responses to the workload questions, your notes, and any high and medium risks that were present when the milestone was saved.

A report enables you to share details about the milestone with others who do not have access to the AWS Well-Architected Tool.

**To generate a milestone report**

1. Select the milestone in one of the following ways.

- From the workload details page, choose **Milestones** and choose the milestone.
- From the **Dashboard** page, choose the workload with the milestone that you want to report on. In the **Milestones** section, choose the milestone.

2. Choose **Generate report** to generate a report.

   The PDF file is generated and you can download or view it.

# Share invitations

A share invitation is a request to share a workload or custom lens owned by another AWS account. A workload or lens can be shared with all users in an AWS account, individual IAM users, or both. If you accept a workload invitation, the workload is added to your **Workloads** and **Dashboard** pages. If you accept a custom lens invitation, the lens is added to your **Custom lenses** page. If you reject the invitation, it's removed from the list.

> **Note**
> Workloads and custom lenses can only be shared within the same AWS Region.

The owner of the workload or custom lens controls who has shared access.

The **Share invitations** page, available from the left navigation, provides information about your pending workload and custom lens invitations.

The following information is displayed for each workload invitation:

**Name**

The name of the workload or custom lens to be shared.

**Resource type**

The type of invitation, either **Workload** or **Custom lens**.

**Owner**

The AWS account ID that owns the workload.

**Permission**

The permission that you are being granted to the workload.

- Read-Only

  Provides read-only access to the workload or custom lens.
- Contributor

  Provides update access to answers and their notes, and read-only access to the rest of the workload. This permission is only available for workloads.

**Permission details**

Detailed description of the permission.

## Accepting a share invitation

**To accept a share invitation**

1. Select the share invitation to accept.
2. Choose **Accept**.

For workload invitations, the workload is added to the **Workloads** and **Dashboard** pages. For custom lens invitations, the custom lens is added to the **Custom lenses** page.

You have seven days to accept an invitation. If you do not accept the invitation within seven days, it's automatically expired.

If an IAM user and the user's AWS account both have accepted workload invitations, the workload invitation for the IAM user determines the user's permission.

# Rejecting a share invitation

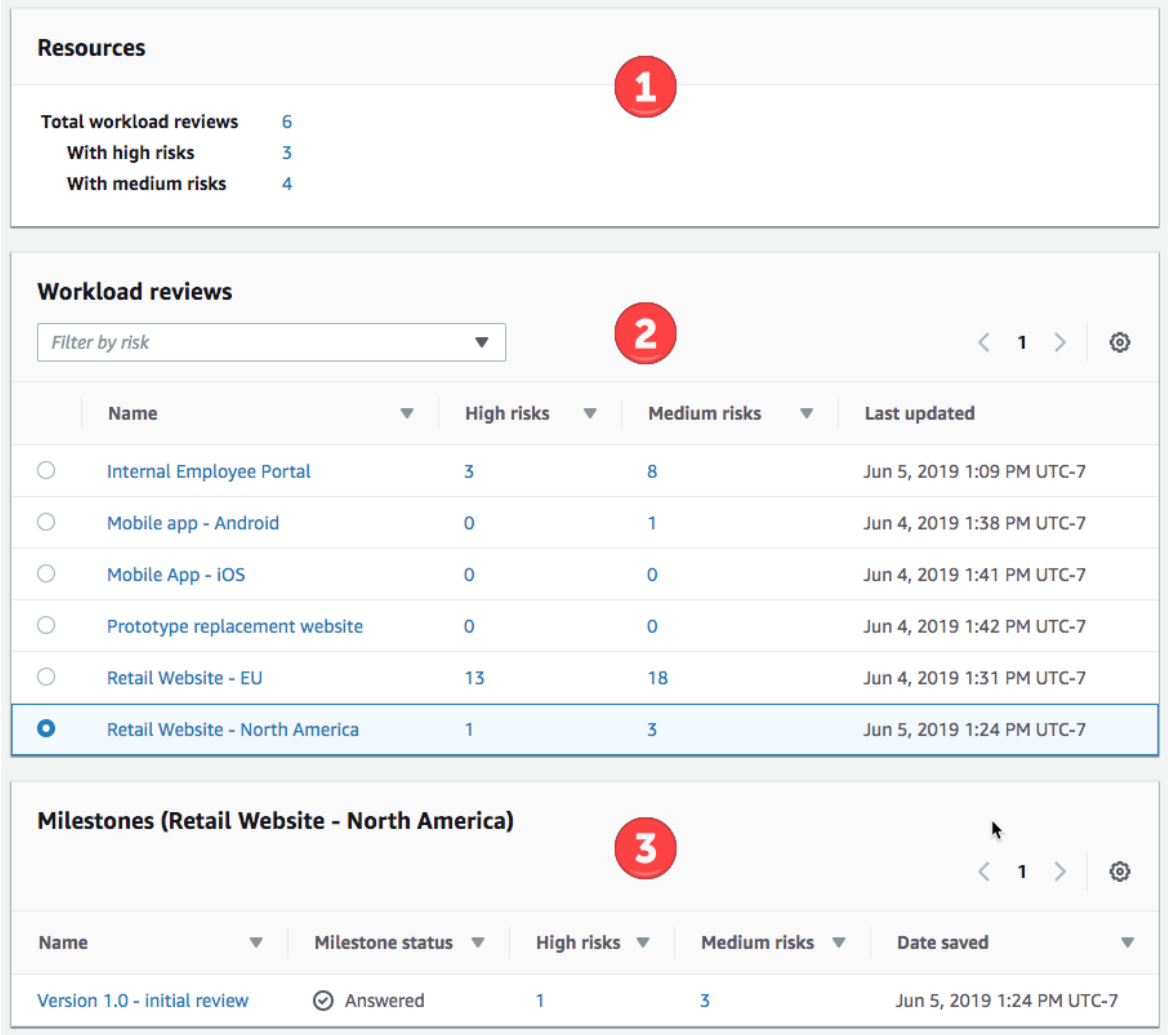
**To reject a share invitation**

1. Select the workload or custom lens invitation to reject.
2. Choose **Reject**.

The invitation is removed from the list.

# Dashboard

The **Dashboard**, available from the left navigation, gives you access to your workloads and their associated milestones. The **Dashboard** consists of three sections.




1. **Resources** — Shows the number of workloads and how many have high and medium risks.
2. **Workload reviews** — Shows a list of your workloads, which you can filter by level of risk.
3. **Milestones** — Shows any milestones associated with the workload that is selected in **Workload reviews**.

## Resources

The **Resources** section shows the number of workloads and the number of workloads with high or medium risks. Choose a count to filter which workloads are displayed in the **Workload reviews** section.

# Workload reviews

The **Workload reviews** section displays information for each workload. You can filter the list based on risk.

The following information is displayed for each workload:

**Name**

The name of the workload.

**Questions answered**

The number of questions answered.

**High risks**

The number of high risk issues (HRIs) identified.

**Medium risks**

The number of medium risk issues (MRIs) identified.

**Last updated**

Date and time that the workload was last updated.

To display a workload's milestones, select it.

Choose the workload name to view the workload details page.

# Milestones

The **Milestones** section displays the milestones associated with the workload selected in the **Workload reviews** section.

The following information is displayed for each milestone:

**Name**

The name of the milestone.

**Questions answered**

The number of questions answered.

**High risks**

The number of high risk issues (HRIs) identified.

**Medium risks**

The number of medium risk issues (HRIs) identified.

**Date saved**

Date and time that the milestone was saved.

Choose a milestone to view the milestones detail page.

# Security in AWS Well-Architected Tool

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is 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 Well-Architected Tool, 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 WA Tool. The following topics show you how to configure AWS WA Tool 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 WA Tool resources.

**Topics**



## Data protection in AWS Well-Architected Tool

The AWS shared responsibility model applies to data protection in AWS Well-Architected Tool. 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 user accounts with 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 personal 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 AWS WA Tool 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

All data stored by AWS WA Tool is encrypted at rest.

## Encryption in transit

All data sent to and from AWS WA Tool is encrypted in transit.

## How AWS uses your data

The AWS Well-Architected team collects aggregated data from the AWS Well-Architected Tool to provide and improve the AWS WA Tool service for customers. Individual customer data may be shared with AWS account teams to support our customers' efforts to improve their workloads and architecture. The AWS Well-Architected team can only access workload properties and selected choices for each question. AWS does not share any data from the AWS WA Tool outside of AWS.

Workload properties that the AWS Well-Architected team has access to include:

- Workload name
- Review owner
- Environment
- Regions
- Account IDs
- Industry type

The AWS Well-Architected team does *not* have access to:

- Workload description
- Architecture design
- Any notes that you entered

# Identity and access management for AWS Well-Architected Tool

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and

*authorized* (have permissions) to use AWS WA Tool resources. IAM is an AWS service that you can use with no additional charge.

**Topics**



# Audience

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

**Service user** – If you use the AWS WA Tool service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more AWS WA Tool 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 AWS WA Tool, see Troubleshooting AWS Well-Architected Tool identity and access (p. 55).

**Service administrator** – If you're in charge of AWS WA Tool resources at your company, you probably have full access to AWS WA Tool. It's your job to determine which AWS WA Tool features and resources your employees 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 AWS WA Tool, see How AWS Well-Architected Tool works with IAM (p. 48).

**IAM administrator** – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to AWS WA Tool. To view example AWS WA Tool identity-based policies that you can use in IAM, see AWS Well-Architected Tool identity-based policy examples (p. 51).

# Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see The IAM Console and Sign-in Page in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication, or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the AWS Management Console, use your password with your root user email or your IAM user name. You can access AWS programmatically using your root user or IAM user access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don’t use AWS tools, you must sign the request yourself. Do this using *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the *AWS General Reference*.

Regardless of the authentication method that you use, you might also 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 Using Multi-Factor Authentication (MFA) in AWS in the *IAM User Guide*.

## AWS account root user

When you first create an AWS account, you begin with a single 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 do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

## IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see Managing access keys for IAM users in the *IAM User Guide*. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

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:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the *IAM User Guide*.
- **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*.

To learn whether to use IAM roles, 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 IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an entity (root user, IAM user, or IAM role) 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*.

An IAM administrator can use policies to specify who has access to AWS resources, and what actions they can perform on those resources. Every IAM entity (user or role) starts with no permissions. In other words, 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 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 Well-Architected Tool works with IAM

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

**Topics**



## AWS WA Tool identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. AWS WA Tool supports specific actions, resources, and condition keys. To learn 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 WA Tool use the following prefix before the action: `wellarchitected:`. For example, to allow an entity to define a workload, an administrator must attach a policy that allows `wellarchitected:CreateWorkload` actions. Similarly, to prevent an entity from deleting workloads, an administrator can attach a policy that denies `wellarchitected:DeleteWorkload` actions. Policy statements must include either an `Action` or `NotAction` element. AWS WA Tool defines its own set of actions that describe tasks that you can perform with this service.

To see a list of AWS WA Tool actions, see Actions Defined by AWS Well-Architected Tool in the *Service Authorization 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": "*"
```

The AWS WA Tool workload resource has the following ARN:

```
arn:${Partition}:wellarchitected:${Region}:${Account}:workload/${ResourceId}
```

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

The ARN can be found on the **Workload properties** page for a workload. For example, to specify a specific workload:

```
"Resource": "arn:aws:wellarchitected:us-
east-1:123456789012:workload/11112222333344445555666677778888"
```

To specify all workloads that belong to a specific account, use the wildcard (*):

```
"Resource": "arn:aws:wellarchitected:us-east-1:123456789012:workload/*"
```

Some AWS WA Tool actions, such as those for creating and listing workloads, cannot be performed on a specific resource. In those cases, you must use the wildcard (*).

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

To see a list of AWS WA Tool resource types and their ARNs, see Resources Defined by AWS Well-Architected Tool in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by AWS Well-Architected Tool.

### Condition Keys

AWS WA Tool does not provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see AWS Global Condition Context Keys in the *Service Authorization Reference*.

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*.

### Examples

To view examples of AWS WA Tool identity-based policies, see AWS Well-Architected Tool identity-based policy examples (p. 51).

## AWS WA Tool resource-based policies

AWS WA Tool does not support resource-based policies.

## Authorization based on AWS WA Tool tags

You can attach tags to AWS WA Tool resources or pass tags in a request to AWS WA Tool. To control access based on tags, you provide tag information in the condition element of a policy using the `wellarchitected:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, or `aws:TagKeys` condition keys. For more information about tagging AWS WA Tool resources, see Tagging your AWS WA Tool resources (p. 60).

## AWS WA Tool IAM roles

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

### Using temporary credentials with AWS WA Tool

AWS WA Tool does not support using temporary credentials.

### Service-linked roles

AWS WA Tool does not support service-linked roles.

### Service roles

AWS WA Tool does not support service roles.

# AWS Well-Architected Tool identity-based policy examples

By default, IAM users and roles don't have permission to create or modify AWS WA Tool resources. They also can't perform tasks using the AWS Management Console 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.

To learn 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*.

**Topics**



## Policy best practices

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

- **Get started using AWS managed policies** – To start using AWS WA Tool quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get started using permissions with AWS managed policies in the *IAM User Guide*.
- **Grant least privilege** – 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*.

## Using the AWS WA Tool console

To access the AWS Well-Architected Tool console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS WA Tool 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.

To ensure that those entities can still use the AWS WA Tool console, also attach the following AWS managed policy to the entities:

```
WellArchitectedConsoleReadOnlyAccess
```

To allow the ability to create, change, and delete workloads, attach the following AWS managed policy to the entities:

```
WellArchitectedConsoleFullAccess
```

For more information, see Adding Permissions to a User in the *IAM User Guide*.

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

## Allow users to view their own permissions

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

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

## Granting full access to workloads

In this example, you want to grant an IAM user in your AWS account full access to your workloads. Full access allows the user to perform all actions in AWS WA Tool. This access is required to define workloads, delete workloads, view workloads, and update workloads.

```
{
```

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

## Granting read-only access to workloads

In this example, you want to grant an IAM user in your AWS account read-only access to your workloads. Read-only access only allows the user to view workloads in AWS WA Tool.

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

## Accessing one workload

In this example, you want to grant an IAM user in your AWS account read-only access to one of your workloads, `99999999999955555555555566666666`, in the `us-west-2` Region. Your account ID is `777788889999`.

```
{
   "Version": "2012-10-17",
   "Statement" : [
     {
     "Effect" : "Allow",
     "Action" : [
          "wellarchitected:Get*",
          "wellarchitected:List*"
     ],
     "Resource": "arn:aws:wellarchitected:us-
west-2:777788889999:workload/99999999999955555555555666666666"
     }
   ]
}
```

# AWS managed policies for AWS Well-Architected Tool

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

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

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

## AWS managed policy: AWSWellArchitectedOrganizationsServiceRolePolicy

You can attach the `AWSWellArchitectedOrganizationsServiceRolePolicy` policy to your IAM identities.

This policy grants administrative permissions in AWS Organizations that are required to support AWS Well-Architected Tool integration with Organizations. These permissions allow the organization management account to enable resource sharing with AWS WA Tool.

**Permissions details**

This policy includes the following permissions.

- `organizations:ListAWSServiceAccessForOrganization` – Allows principals to check if the AWS service access is enabled for AWS WA Tool.
- `organizations:DescribeAccount` – Allows principals to retrieve information about an account in the organization.
- `organizations:DescribeOrganization` – Allows principals to retrieve information about the organization configuration.
- `organizations:ListAccounts` – Allows principals to retrieve the list of accounts that belong to an organization.
- `organizations:ListAccountsForParent` – Allows principals to retrieve the list of accounts that belong to an organization from a given root node in the organization.
- `organizations:ListChildren` – Allows principals to retrieve the list of accounts and organization units that belong to an organization from a given root node in the organization.
- `organizations:ListParents` – Allows principals to retrieve the list of immediate parents specified by the OU or account within an organization.
- `organizations:ListRoots` – Allows principals to retrieve the list of all root nodes within an organization.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization",
                "organizations:ListAccounts",
                "organizations:ListAccountsForParent",
```

```
                "organizations:ListChildren",
                "organizations:ListParents",
                "organizations:ListRoots"
            ],
            "Resource": "*"
        }
    ]
}
```

## AWS WA Tool updates to AWS managed policies

View details about updates to AWS managed policies for AWS WA Tool since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AWS WA Tool Document history (p. 69) page.

| Change | Description | Date |
| --- | --- | --- |
| AWS WA Tool added permissions | Added a new action to grant `ListAWSServiceAccessForOrganization` to allow AWS WA Tool to check if the AWS service access is enabled for AWS WA Tool. | July 22, 2022 |
| AWS WA Tool started tracking changes | AWS WA Tool started tracking changes for its AWS managed policies. | July 22, 2022 |

# Troubleshooting AWS Well-Architected Tool identity and access

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

**Topics**



## I'm not authorized to perform an action in AWS WA Tool

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The following example error occurs when the *mateojackson* user tries to use the console to perform the DeleteWorkload action, but does not have permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to
 perform: wellarchitected:DeleteWorkload on resource: 11112222333344445555666677778888
```

For this example, ask your administrator to update your policies to allow you to access the `11112222333344445555666677778888` resource using the `wellarchitected:DeleteWorkload` action.

### I'm an administrator and want to allow others to access AWS WA Tool

To allow others to access AWS WA Tool, 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 AWS WA Tool.

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

# Compliance Validation for AWS Well-Architected Tool

AWS Well-Architected Tool is not in scope of any AWS compliance programs.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

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

Your compliance responsibility when using AWS WA Tool is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Config – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

# Resilience in AWS Well-Architected Tool

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

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

# Infrastructure security in AWS Well-Architected Tool

As a managed service, AWS Well-Architected Tool is protected by the AWS global network security procedures that are described in Best Practices for Security, Identity, & Compliance.

# Sharing your AWS WA Tool resources

To share a workload or custom lens resource that you own, do the following:

- Enable resource sharing within AWS Organizations (p. 58) (optional)
- Share a workload (p. 17)
- Share a custom lens (p. 29)

**Notes**

- Sharing a resource makes it available for use by principals outside of the AWS account that created the resource. Sharing doesn't change any permissions that apply to the resource in the account that created it.
- AWS WA Tool is a Regional service. The principals that you share with can access resource shares in only the AWS Regions in which they were created.
- To share resources in a Region introduced after March 20, 2019, both you and the shared AWS account must enable the Region in the AWS Management Console. For more information, refer to AWS Global Infrastructure.

# Enable resource sharing within AWS Organizations

When your account is managed by AWS Organizations, you can take advantage of that to share resources more easily. With or without Organizations, a user can share with individual accounts. However, if your account is in an organization, then you can share with individual accounts, or with all accounts in the organization or in an OU without having to enumerate each account.

To share resources within an organization, you must first use the AWS WA Tool console or AWS Command Line Interface (AWS CLI) to enable sharing with AWS Organizations. When you share resources in your organization, AWS WA Tool doesn't send invitations to principals. Principals in your organization gain access to shared resources without exchanging invitations.

When you enable resource sharing within your organization, AWS WA Tool creates a service-linked role called `AWSServiceRoleForWellArchitected`. This role can be assumed by only the AWS WA Tool service, and grants AWS WA Tool permission to retrieve information about the organization it is a member of, by using the AWS managed policy `AWSWellArchitectedOrganizationsServiceRolePolicy`.

If you no longer need to share resources with your entire organization or OUs, you can disable resource sharing.

**Requirements**

- You can perform these steps only while signed in as a principal in the organization's management account.
- The organization must have all features enabled. For more information, see Enabling all features in your organization in the *AWS Organizations User Guide*.

**Important**
You must enable sharing with AWS Organizations by using the AWS WA Tool console. This ensures that the `AWSServiceRoleForWellArchitected` service-linked role is created. If you

enable trusted access with AWS Organizations by using the AWS Organizations console or the enable-aws-service-access AWS CLI command, the `AWSServiceRoleForWellArchitected` service-linked role isn't created, and you can't share resources within your organization.

### To enable resource sharing within your organization

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.

   You must sign in as a principal in the organization's management account.
2. In the left navigation pane, choose **Settings**.
3. Choose **Enable AWS Organizations support**.
4. Choose **Save settings**.

### To disable resource sharing within your organization

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.

   You must sign in as a principal in the organization's management account.
2. In the left navigation pane, choose **Settings**.
3. Unselect **Enable AWS Organizations support**.
4. Choose **Save settings**.

# Tagging your AWS WA Tool resources

To help you manage your AWS WA Tool resources, you can assign your own metadata to each resource in the form of *tags*. This topic describes tags and shows you how to create them.

**Contents**



## Tag basics

A tag is a label that you assign to an AWS resource. Each tag consists of a *key* and an optional *value*, both of which you define.

Tags enable you to categorize your AWS resources by, for example, purpose, owner, or environment. When you have many resources of the same type, you can quickly identify a specific resource based on the tags you've assigned to it. For example, you can define a set of tags for your AWS WA Tool services to help you track each service's owner and stack level. We recommend that you devise a consistent set of tag keys for each resource type.

Tags are not automatically assigned to your resources. After you add a tag, you can edit tag keys and values or remove tags from a resource at any time. If you delete a resource, any tags for the resource are also deleted.

Tags don't have any semantic meaning to AWS WA Tool and are interpreted strictly as a string of characters. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value.

You can work with tags using the AWS Management Console, the AWS CLI, and the AWS WA Tool API.

If you're using AWS Identity and Access Management (IAM), you can control which users in your AWS account have permission to create, edit, or delete tags.

## Tagging your resources

You can tag new or existing AWS WA Tool workloads.

If you're using the AWS WA Tool console, you can apply tags to new workloads when they are created or to existing workloads at any time from its **Properties** tab.

If you're using the AWS WA Tool API, the AWS CLI, or an AWS SDK, you can apply tags to new workloads using the `tags` parameter on the relevant API action or to existing workloads using the `TagResource` API action. For more information, see TagResource.

Some resource-creating actions enable you to specify tags for a resource when the resource is created. If tags cannot be applied during resource creation, the resource creation process fails. This ensures that resources you intended to tag on creation are either created with specified tags or not created at all. If

you tag resources at the time of creation, you don't need to run custom tagging scripts after resource creation.

The following table describes the AWS WA Tool resources that can be tagged, and the resources that can be tagged on creation.

**Tagging support for AWS WA Tool resources**

| Resource | Supports tags | Supports tag propagation | Supports tagging on creation (AWS WA Tool API, AWS CLI, AWS SDK) |
|---|---|---|---|
| AWS WA Tool workloads | Yes | No | Yes |
| AWS WA Tool custom lenses | Yes | Yes | Yes |

# Tag restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource – 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length – 128 Unicode characters in UTF-8
- Maximum value length – 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple AWS services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are letters, numbers, spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case sensitive.
- Don't use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values, as it is reserved for AWS use. You can't edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags-per-resource limit.

# Working with tags using the console

Using the AWS WA Tool console, you can manage the tags associated with new or existing workloads.

## Adding tags on an individual resource on creation

You can add tags to AWS WA Tool workloads when you create them.

## Adding and deleting tags on an individual resource

AWS WA Tool allows you to add or delete tags associated with your workloads directly from the **Properties** page.

**To add or delete a tag on a workload**

1. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.

2. From the navigation bar, choose the Region to use.
3. In the navigation pane, choose **Workloads**.
4. Select the workload to modify and choose **Properties**.
5. In the **Tags** section, choose **Manage tags**.
6. Add or delete your tags as necessary.
   - To add a tag, choose **Add new tag** and fill in the **Key** and **Value** fields.
   - To delete a tag, choose **Remove**.
7. Repeat this process for each tag you want to add, modify, or delete. Choose **Save** to save your changes.

# Working with tags using the API

Use the following AWS WA Tool API operations to add, update, list, and delete the tags for your resources.

### Tagging support for AWS WA Tool resources

| Task | API action |
|---|---|
| Add or overwrite one or more tags. | TagResource |
| Delete one or more tags. | UntagResource |
| List tags for a resource. | ListTagsForResource |

Some resource-creating actions enable you to specify tags when you create the resource. The following actions support tagging on creation.

| Task | API action |
|---|---|
| Create a workload | CreateWorkload |
| Import a new lens | ImportLens |

# Logging AWS WA Tool API calls with AWS CloudTrail

AWS Well-Architected Tool is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS WA Tool. CloudTrail captures all API calls for AWS WA Tool as events. The calls captured include calls from the AWS WA Tool console and code calls to the AWS WA Tool API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS WA Tool. 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 AWS WA Tool, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

## AWS WA Tool information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS WA Tool, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for AWS WA Tool, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All AWS WA Tool actions are logged by CloudTrail and are documented in Actions Defined by AWS Well-Architected Tool. For example, calls to the `CreateWorkload`, `DeleteWorkload`, and `CreateWorkloadShare` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

# Understanding AWS WA Tool log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `CreateWorkload` action.


```
   {
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:dev-dsk-xiulan-2a-1111111c.us-
west-2.amazon.com",
        "arn": "arn:aws:sts::444455556666:assumed-role/well-architected-api-svc-integ-test-
read-write/dev-dsk-xiulan-2a-1111111c.us-west-2.amazon.com",
        "accountId": "444455556666",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::444455556666:role/well-architected-api-svc-integ-test-
read-write",
                "accountId": "444455556666",
                "userName": "well-architected-api-svc-integ-test-read-write"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-10-14T03:41:39Z"
            }
        }
    },
    "eventTime": "2020-10-14T04:43:13Z",
    "eventSource": "wellarchitected.amazonaws.com",
    "eventName": "CreateWorkload",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.178",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.848
 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.262-b10
 java/1.8.0_262 vendor/Oracle_Corporation",
    "requestParameters": {
            "ClientRequestToken": "08af866a-0238-4070-89c2-b689ca8339f7",
            "Description": "***",
            "AwsRegions": [
                "us-west-2"
            ],
            "ReviewOwner": "***",
            "Environment": "PRODUCTION",
            "Name": "***",
            "Lenses": [
                "wellarchitected",
                "serverless"
            ]
    },
    "responseElements": {
         "Arn": "arn:aws:wellarchitected:us-
west-2:444455556666:workload/8cdcdf7add10b181fdd3f686dacffdac",
         "Id": "8cdcdf7add10b181fdd3f686dacffdac"
```

```
    },
    "requestID": "22bad4e3-aa51-4ff1-b480-712ee07cedbd",
    "eventID": "50849dfd-36ed-418e-a901-49f6ac7087e8",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "444455556666"
}
```

# EventBridge

AWS Well-Architected Tool sends events to Amazon EventBridge when actions are taken on Well-Architected resources. You can use EventBridge and these events to write rules that take actions, such as notifying you, when a resource change occurs. For more information, see What is Amazon EventBridge?

**Note**
Events are delivered on a best-effort basis.

The following actions result in EventBridge events:

- Workload-related
  - Creating or deleting a workload
  - Creating a milestone
  - Updating the properties of a workload
  - Sharing or unsharing a workload
  - Updating the status of a share invitation
  - Adding or removing tags
  - Updating an answer
  - Updating review notes
  - Adding or removing a lens from a workload
- Lens-related
  - Importing or exporting a custom lens
  - Publishing a custom lens
  - Deleting a custom lens
  - Sharing or unsharing a custom lens
  - Updating the status of a share invitation
  - Adding or removing a lens from a workload

# Sample events from AWS WA Tool

This section includes example events from AWS Well-Architected Tool.

**Updating an answer in a workload**


```
{
  "version":"0",
  "id":"00de336a-83cc-b80b-f0e6-f44c88a96050",
  "detail-type":"AWS API Call via CloudTrail",
  "source":"aws.wellarchitected",
  "account":"123456789012",
  "time":"2022-02-17T08:01:25Z",
  "region":"us-west-2",
  "resources":[],
  "detail":{
     "eventVersion":"1.08",
     "userIdentity":{
        "type":"AssumedRole",
        "principalId":"AROA4JUSXMN5ZR6S7LZNP:sample-user",
```

```
         "arn":"arn:aws:sts::123456789012:assumed-role/Admin/example-user",
         "accountId":"123456789012",
         "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
         "sessionContext":{
            "sessionIssuer":{
               "type":"Role",
               "principalId":"AROA4JUSXMN5ZR6S7LZNP",
               "arn":"arn:aws:iam::123456789012:role/Admin",
               "accountId":"123456789012",
               "userName":"Admin"
            },
            "webIdFederationData":{},
            "attributes":{
               "creationDate":"2022-02-17T07:21:54Z",
               "mfaAuthenticated":"false"
            }
         }
      },
      "eventTime":"2022-02-17T08:01:25Z",
      "eventSource":"wellarchitected.amazonaws.com",
      "eventName":"UpdateAnswer",
      "awsRegion":"us-west-2",
      "sourceIPAddress":"10.246.162.39",
      "userAgent":"aws-internal/3 aws-sdk-java/1.12.127
 Linux/5.4.156-94.273.amzn2int.x86_64 OpenJDK_64-Bit_Server_VM/25.312-b07 java/1.8.0_312
 vendor/Oracle_Corporation cfg/retry-mode/standard",
      "requestParameters":{
         "Status":"Acknowledged",
         "SelectedChoices":"***",
         "ChoiceUpdates":"***",
         "QuestionId":"priorities",
         "WorkloadId":"ee73fda518f9bd4aa804c6252e4e37b0",
         "IsApplicable":true,
         "LensAlias":"wellarchitected",
         "Reason":"NONE",
         "Notes":"***"
      },
      "responseElements":{
         "Answer":"***",
         "LensAlias":"wellarchitected",
         "WorkloadId":"ee73fda518f9bd4aa804c6252e4e37b0"
      },
      "requestID":"7bae1153-26a8-4dc0-9307-68b17b107619",
      "eventID":"8339c258-4ddd-48aa-ab21-3f82ce9d79cd",
      "readOnly":false,
      "eventType":"AwsApiCall",
      "managementEvent":true,
      "recipientAccountId":"123456789012",
      "eventCategory":"Management"
   }
}
```


**Publishing a custom lens**


```
{
   "version":"0",
   "id":"4054a34b-60a9-53c1-3146-c1a384dba41b",
   "detail-type":"AWS API Call via CloudTrail",
   "source":"aws.wellarchitected",
   "account":"123456789012",
   "time":"2022-02-17T08:58:34Z",
   "region":"us-west-2",
   "resources":[],
```

```
   "detail":{
      "eventVersion":"1.08",
      "userIdentity":{
         "type":"AssumedRole",
         "principalId":"AROA4JUSXMN5ZR6S7LZNP:example-user",
         "arn":"arn:aws:sts::123456789012:assumed-role/Admin/example-user",
         "accountId":"123456789012",
         "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
         "sessionContext":{
            "sessionIssuer":{
               "type":"Role",
               "principalId":"AROA4JUSXMN5ZR6S7LZNP",
               "arn":"arn:aws:iam::123456789012:role/Admin",
               "accountId":"123456789012",
               "userName":"Admin"
            },
            "webIdFederationData":{},
            "attributes":{
               "creationDate":"2022-02-17T07:21:54Z",
               "mfaAuthenticated":"false"
            }
         }
      },
      "eventTime":"2022-02-17T08:58:34Z",
      "eventSource":"wellarchitected.amazonaws.com",
      "eventName":"CreateLensVersion",
      "awsRegion":"us-west-2",
      "sourceIPAddress":"10.246.162.39",
      "userAgent":"aws-internal/3 aws-sdk-java/1.12.127
 Linux/5.4.156-94.273.amzn2int.x86_64 OpenJDK_64-Bit_Server_VM/25.312-b07 java/1.8.0_312
 vendor/Oracle_Corporation cfg/retry-mode/standard",
      "requestParameters":{
         "IsMajorVersion":true,
         "LensVersion":"***",
         "ClientRequestToken":"03f46163-e95c-4455-8479-266373aa09c7",
         "LensAlias":"***"
      },
      "responseElements":{
         "LensArn":"arn:aws:wellarchitected:us-
west-2:123456789012:lens/6261deecb9def44f9aecc938ca25d94e",
         "LensVersion":"***"
      },
      "requestID":"167b7051-980d-42ee-9967-0b4b3163e948",
      "eventID":"c7ef2b47-419d-45b7-8982-fbade9b558c7",
      "readOnly":false,
      "eventType":"AwsApiCall",
      "managementEvent":true,
      "recipientAccountId":"123456789012",
      "eventCategory":"Management"
   }
}
```

# Document history

The following table describes the documentation for this release of the AWS Well-Architected Tool.

- **API version: latest**
- **Latest documentation update:** August 2, 2022

| update-history-change | update-history-description | update-history-date |
|---|---|---|
| Content update (p. 69) | The `choices` section of the custom lens JSON specification was updated. | August 2, 2022 |
| Updated functionality (p. 53) | This release adds tracking changes for its AWS managed policies and added a new action to grant the `ListAWSServiceAccessForOrganization` permission to the `AWSWellArchitectedOrganizationsServiceRolePolicy`. | July 22, 2022 |
| Organization sharing added (p. 58) | This release adds the ability to share workloads and custom lenses with an organization and organization units (OUs). | June 30, 2022 |
| Updated functionality (p. 69) | This release adds the ability to specify additional resources for choices in a custom lens, to preview a custom lens before publishing it, and add tags to custom lenses. | June 21, 2022 |
| Updated functionality (p. 69) | This release adds the ability to access the AWS Well-Architected community on AWS re:Post. | May 31, 2022 |
| Updated functionality (p. 69) | This release adds the sustainability pillar and minor updates to Tutorial. | March 31, 2022 |
| EventBridge support added (p. 66) | AWS WA Tool now sends an event to Amazon EventBridge when a change is made to a Well-Architected resource. | March 3, 2022 |
| Custom lenses added (p. 25) | The ability to add custom lenses has been added. | November 29, 2021 |
| Updated functionality (p. 69) | Individual best practices can now be marked as not applicable. | July 14, 2021 |
| Resource tagging available (p. 60) | This release adds the ability to add tags to workloads. | March 3, 2021 |

| | | |
|---|---|---|
| API now available (p. 63) | This release adds the AWS WA Tool API. AWS CloudTrail logging information added. | December 16, 2020 |
| Updated functionality (p. 69) | This release adds the FTR and SaaS lenses to the tool. | December 3, 2020 |
| Data protection updated (p. 43) | Data protection information updated. | November 5, 2020 |
| Content update (p. 69) | Clarified that after you upgrade a workload to use a new lens that you cannot go back to the previous version. | July 8, 2020 |
| Content update (p. 69) | Clarified sharing in AWS Regions introduced after March 20, 2019. | June 24, 2020 |
| Updated functionality (p. 69) | Access to a workload share is removed immediately when a workload share invitation is rejected. Shared access is granted when the share is accepted. | June 17, 2020 |
| Content update (p. 16) | Definitions for high risk issues (HRIs) and medium risk issues (MRIs) added. | June 12, 2020 |
| Content update (p. 44) | Section on how AWS uses your data was added. | May 21, 2020 |
| Updated functionality (p. 69) | This release adds a review owner to the workload. | April 1, 2020 |
| Updated functionality (p. 69) | This release adds an architectural diagram link to the workload. | March 10, 2020 |
| Content update (p. 69) | Clarified that workload shares are AWS Region-specific. | January 10, 2020 |
| Updated functionality (p. 69) | This release adds workload sharing. | January 9, 2020 |
| Content update (p. 69) | Security section updated with latest guidance. | December 6, 2019 |
| Updated functionality (p. 69) | This release makes the industry fields optional when defining a workload. | August 19, 2019 |
| Updated functionality (p. 69) | This release adds improvement plan items to the workload report. | July 29, 2019 |
| Updated functionality (p. 69) | The release adds the DeleteWorkload action to the policy. | July 18, 2019 |

| | | |
|---|---|---|
| Content update (p. 69) | The content in this guide has been updated with minor fixes. | June 19, 2019 |
| Content update (p. 69) | The content in this guide has been updated with minor fixes. | May 30, 2019 |
| Updated functionality (p. 69) | This release supports upgrading the version of the framework used for a workload review. | May 1, 2019 |
| Updated functionality (p. 69) | This release adds the ability to specify non-AWS Regions when defining a workload. | February 14, 2019 |
| AWS Well-Architected Tool general availability (p. 69) | This release introduces the AWS Well-Architected Tool. | November 29, 2018 |

# AWS glossary

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