

---

# AWS Well-Architected Tool

## User Guide



## **AWS Well-Architected Tool: User Guide**

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

## Table of Contents

What is AWS Well-Architected Tool? .....	1
The AWS Well-Architected Framework .....	1
The AWS Serverless Application Lens .....	1
Definitions .....	2
Getting Started .....	3
Provisioning an IAM User .....	3
Defining a Workload .....	4
Documenting a Workload .....	5
Question Page .....	6
Saving a Milestone .....	6
Tutorial .....	8
Step 1: Define a Workload .....	8
Step 2: Document the Workload State .....	11
Step 3: Review the Improvement Plan .....	14
Step 4: Make Improvements and Measure Progress .....	16
Workloads .....	20
Viewing a Workload .....	21
Editing a Workload .....	21
Sharing a Workload .....	21
Sharing Considerations .....	22
Deleting Shared Access .....	22
Modifying Shared Access .....	23
Accepting and Rejecting Workload Invitations .....	23
Deleting a Workload .....	24
Generating a Workload Report .....	24
Workload Details .....	25
Overview Tab .....	25
Milestones Tab .....	25
Properties Tab .....	26
Shares Tab .....	26
Milestones .....	28
Saving a Milestone .....	28
Viewing Milestones .....	28
Generating a Milestone Report .....	28
Lenses .....	30
Adding a Lens .....	30
Removing a Lens .....	30
Lens Details .....	30
Overview Tab .....	31
Improvement Plan Tab .....	31
Lens Upgrades .....	31
Notifications .....	31
Selecting the Lens Upgrade .....	31
Upgrading the Lens .....	32
Workload Invitations .....	33
Accepting a Workload Invitation .....	33
Rejecting a Workload Invitation .....	34
Dashboard .....	35
Resources .....	35
Workload Reviews .....	36
Milestones .....	36
Security .....	37
Data Protection .....	37
Encryption at Rest .....	38

Encryption in Transit .....	38
How AWS Uses Your Data .....	38
Identity and Access Management .....	38
Audience .....	39
Authenticating With Identities .....	39
Managing Access Using Policies .....	40
How AWS Well-Architected Tool Works with IAM .....	42
Identity-Based Policy Examples .....	44
Troubleshooting .....	47
Compliance Validation .....	47
Resilience .....	48
Infrastructure Security .....	48
Document History .....	49
AWS glossary .....	50

# 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

Today, 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.

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.

## 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 five pillars: operational excellence, security, reliability, performance efficiency, and cost optimization.

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, see the [AWS Well-Architected website](#).

## The AWS Serverless Application Lens

The AWS Serverless Application Lens documents 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](#).

## 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.

# Getting Started with AWS Well-Architected Tool

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

## Topics

- [Provisioning an IAM User \(p. 3\)](#)
- [Defining a Workload \(p. 4\)](#)
- [Documenting a Workload \(p. 5\)](#)
- [Saving a Milestone \(p. 6\)](#)

## 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 \(p. 37\)](#).

## 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. 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.

13. Choose the lenses that apply to this workload.
  - **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.
  - **Serverless Lens** – Select this lens for a set of additional questions to consider for your serverless application workloads.
14. 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) 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.

- c. 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.

The screenshot displays the AWS Well-Architected Tool's Question Page, which is divided into three main panels. Panel 1 (left) shows a list of questions categorized by pillars: Security (0/11), Reliability (0/9), Performance Efficiency (0/8), and Cost Optimization (0/9). Questions are marked as 'Done' or 'N/A'. Panel 2 (middle) shows the detailed view of 'OPS 9. How do you evolve operations?'. It includes a description, a radio button to 'Question does not apply to this workload', a list of checkboxes for various practices (e.g., 'Have a process for continuous improvement'), and a text area for optional notes. Panel 3 (right) shows 'Helpful resources' including a video and text links for 'Have a process for continuous improvement', 'Implement feedback loops', 'Define drivers for improvement', 'Validate insights', and 'Perform operations metrics reviews'.

1. In the left panel, the questions for each pillar are shown. Questions that you have answered are marked **Done** and questions that do not apply to this workload are marked with **N/A**. 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. In the middle panel, the current question is displayed. Select the best practices that you are following. Choose **Info** to get additional information about the question or a best practice.

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. In the right panel, additional information and helpful resources are displayed.

## 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 \(p. 28\)](#).

# 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 \(p. 8\)](#)
- [Step 2: Document the Workload State \(p. 11\)](#)
- [Step 3: Review the Improvement Plan \(p. 14\)](#)
- [Step 4: Make Improvements and Measure Progress \(p. 16\)](#)

## 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 \(p. 3\)](#) 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, we enter a description for the workload.
5. In the **Review owner** box, we enter the name of the person responsible for the workload review process.
6. In the **Environment** box, we indicate that the workload is in a production environment.

Well-Architected Tool > Workloads > Define workload

Step 1  
**Specify properties**

Step 2  
Apply lenses

## Specify properties

### Workload properties

**Why does AWS need this data, and how will it be used?**

**Name**  
A unique identifier for the workload

Retail Website - North America

The name must be from 3 to 100 characters. At least 3 characters must be non-whitespace. 67 characters remaining

**Description**  
A brief description of the workload to document its scope and intended purpose

Customer-facing website for retail order processing

The description must be from 3 to 250 characters. 199 characters remaining

**Review owner**  
The name, email address, or identifier for the primary individual or group that owns the review process

Ana Carolina Silva

The review owner must be from 3 to 255 characters. At least 3 characters must be non-whitespace. 237 characters remaining

**Environment**  
The environment in which your workload runs

☒ Production  
☐ Pre-production

Regions

7. Our workload runs on both AWS and at our local data center:
  - a. We select **AWS Regions**, and choose the two Regions in North America where the workload runs.
  - b. We also select **Non-AWS regions**, and enter a name for our local data center.

☒ Production  
☐ Pre-production

**Regions**  
☒ **AWS Regions**  
The AWS Regions in which your workload runs

☒ **Non-AWS regions**  
Type the regions outside of AWS in which your workload runs

Specify up to 5 unique regions separated by commas. Each region name can be at most 25 characters

**Account IDs - optional**  
Type the IDs of the AWS accounts your workload spans across

8. The **Account IDs** box is optional, and we chose not to associate any AWS accounts with this workload.
9. The **Architectural diagram** box is optional, and we chose not to associate an architectural diagram with this workload.
10. The **Industry type** and **Industry** boxes are optional and are not specified for this workload.
11. Choose **Next**.

**Account IDs - optional**  
Type the IDs of the AWS accounts your workload spans across

Specify up to 100 unique account IDs separated by commas

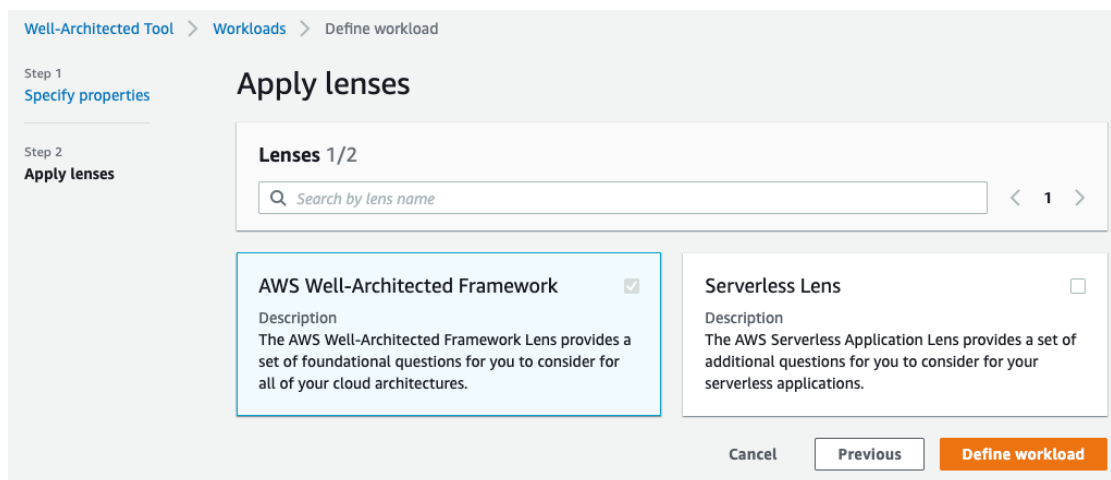
**Architectural design - optional**  
A link to your architectural design

The URL can be up to 2048 characters and must begin with one of the follow protocols: [http, https, ftp]. 2048 characters remaining

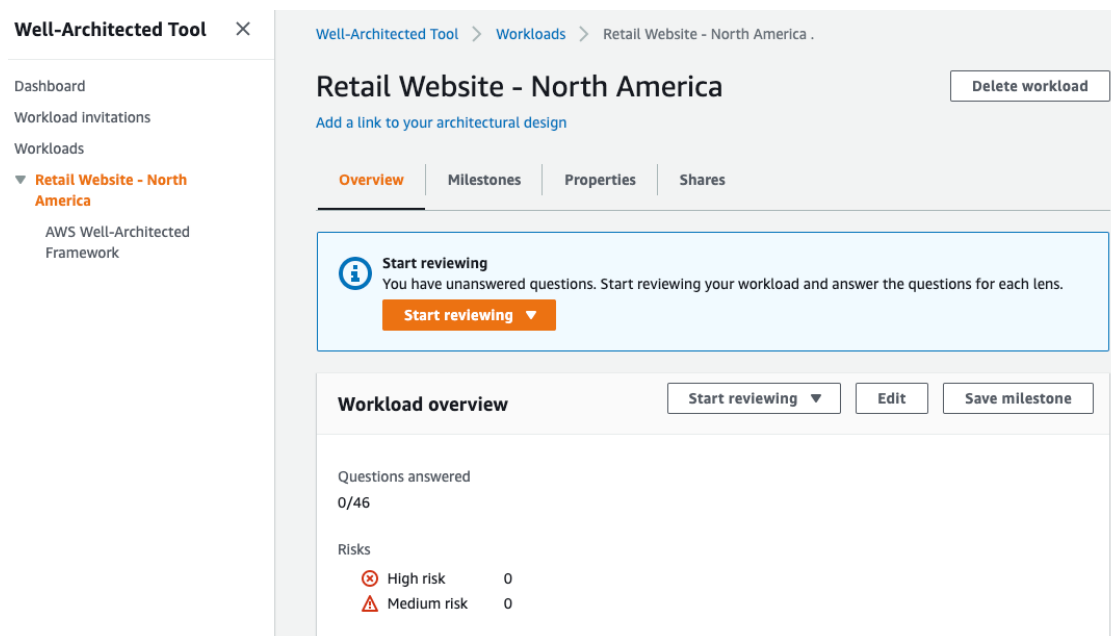
**Industry type - optional**  
The industry that your workload is associated with

**Industry - optional**  
The category within your industry that your workload is associated with

12. For this example, we apply the AWS Well-Architected Framework lens. 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, and cost optimization.

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.

The screenshot shows the AWS Well-Architected Tool interface. On the left, a navigation pane lists nine questions under the 'Operational Excellence' pillar. The main content area displays 'OPS 1. How do you determine what your priorities are?' with a radio button selected for 'Question does not apply to this workload'. Below this, there is a list of options: 'Evaluate external customer needs', 'Evaluate internal customer needs', 'Evaluate compliance requirements', 'Evaluate threat landscape', 'Evaluate tradeoffs', 'Manage benefits and risks', and 'None of these'. A 'Notes - optional' text box is present, showing '2084 characters remaining'. At the bottom right, there are 'Save and exit' and 'Next' buttons. On the right sidebar, there is a video player showing a person speaking, and links to 'AWS Support' and 'AWS Cloud Compliance'.

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.

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.

☒ **None of these** [Info](#)

**Notes - optional**

We don't do any of these as part of version 3. We've added this as a high priority item for version 4.

1982 characters remaining

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. Select the name of the workload to open the workload details page. Choose **Continue reviewing** and then navigate to where you left off.

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.



Well-Architected Tool > Workloads > Retail Website - North America

## Retail Website - North America

Architectural design [↗](#)

**Overview** | Milestones | Properties | Shares

**Workload overview** Continue reviewing ▼ Edit Save milestone

Questions answered  
46/46

Risks

⊗	High risk	3
⚠	Medium risk	1

Last updated  
Mar 9, 2020 2:55 PM UTC-6

Workload notes

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, choose **Save milestone**, enter **Version 1.0 - initial review** as the **Milestone name**, and choose **Save**.

**Save milestone** ×

Save a milestone to capture the architectural health of your workload at this stage of your review. You can save multiple milestones for each workload and compare milestones with each other.

Milestone name

Milestone names within a workload must be unique. The name must be from 3 to 100 characters. At least 3 characters must be non-whitespace. 72 characters remaining.

Cancel Save

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


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


The screenshot shows the 'AWS Well-Architected Framework Lens' page with the 'Improvement plan' tab selected. The 'Improvement plan overview' section displays the following risks:

Risks		
⊗	High risk	3
⚠	Medium risk	1

Below this, the 'Improvement items' section is visible with a pagination control showing '1'.

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

 High risk 3

 Medium risk 1

Improvement status

Choose the status of your workload improvements.

Not Started ▼

Improvement plan configuration

Edit

Pillar priority

1. Security

2. Operational Excellence

3. Cost Optimization

4. Performance Efficiency

5. Reliability

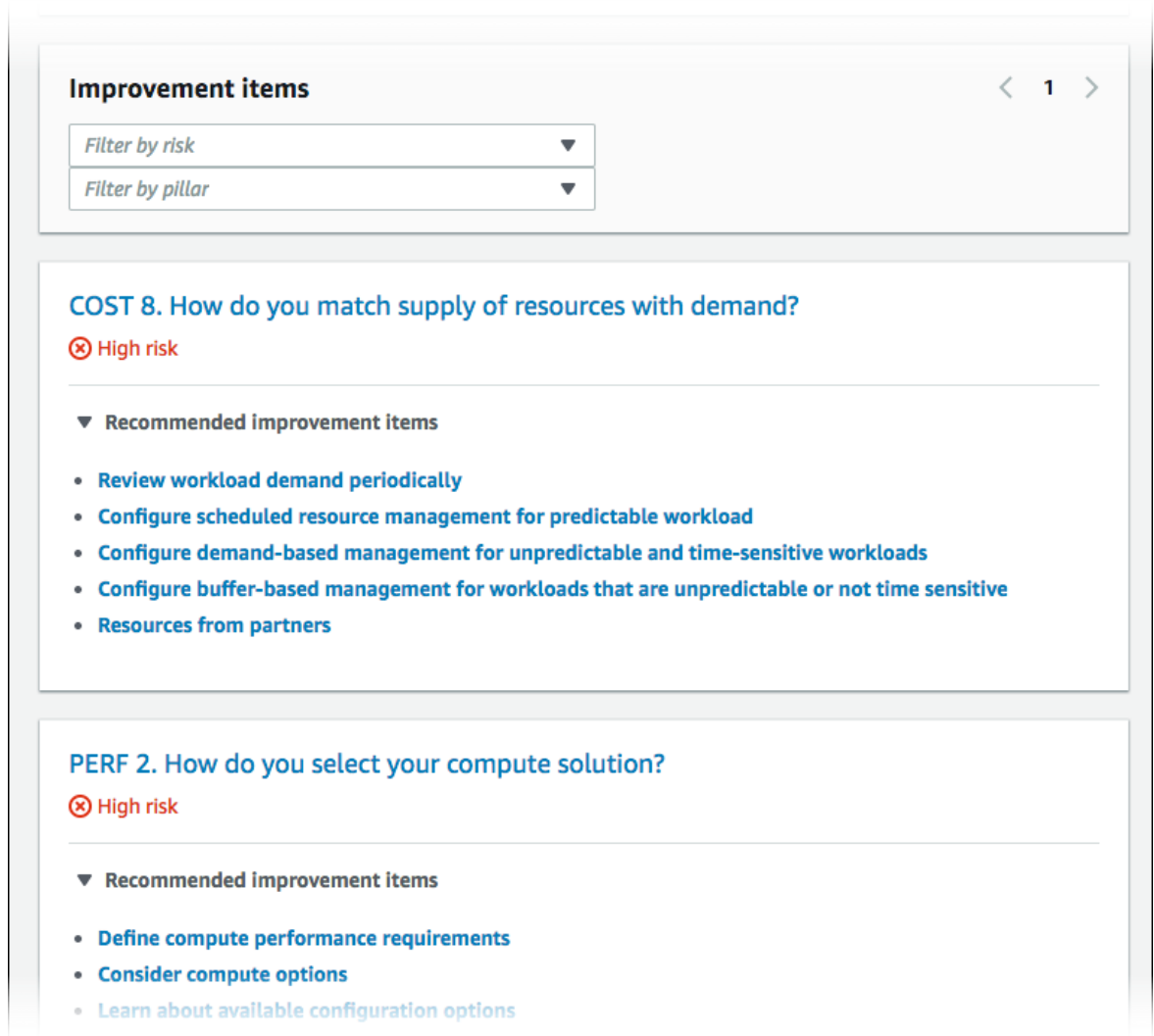
Improvement items

< 1 >

Filter by risk ▼

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 items listed first followed by any medium risk items.

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

After deciding what improvement actions to take, update the **Improvement status** to indicate that improvements are in progress.

**Risks**

⊗ High risk 3

⚠ Medium risk 1

**Improvement status**  
Choose the status of your workload improvements.

In Progress ▼

None

Not Started

In Progress

Complete

Risk Acknowledged

**Edit**

**Pillar priority**  
1. Security

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

From the **Improvement items** section, choose the pertinent question and update the selected best practices to reflect the changes. **Notes** are added to record the improvements, and then choose **Save and exit** to update the state of the workload.

▶ Operational Excellence

▶ Security

▶ Reliability

▼ Performance Efficiency

✔ PERF 1. How do you select the best performing architecture?

✔ PERF 2. How do you select your compute solution?

✔ PERF 3. How do you select your storage solution?

✔ PERF 4. How do you select your database solution?

✔ PERF 5. How do you configure your networking solution?

⊖ PERF 6. How do you evolve your workload to take advantage of new releases?

✔ PERF 7. How do you monitor your resources to ensure they are performing as expected?

PERF 2. How do you select your compute solution? [Info](#) [Save and exit](#)

The optimal compute solution for a system varies based on application design, usage patterns, and configuration settings. Architectures may use different compute solutions for various components and enable different features to improve performance. Selecting the wrong compute solution for an architecture can lead to lower performance efficiency.

☐ Question does not apply to this workload [Info](#)

Select from the following

☒ Evaluate the available compute options [Info](#)

☒ Understand the available compute configuration options [Info](#)

☒ Collect compute-related metrics [Info](#)

☒ Determine the required configuration by right-sizing [Info](#)

☒ Use the available elasticity of resources [Info](#)

☒ Re-evaluate compute needs based on metrics [Info](#)

☐ None of these [Info](#)

Notes - optional

Added Amazon CloudWatch to get better metrics and AWS Auto Scaling to enable elasticity.

1996 characters remaining

Question status

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 items from three to only one.

Well-Architected Tool > Workloads > Retail Website - North America

## Retail Website - North America

Delete workload

Review | **Improvement plan** | Milestones | Properties

### Improvement plan overview

Risks

- High risk 1
- Medium risk 2

Improvement status

Choose the status of your workload improvements.

In Progress ▼

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

Well-Architected Tool > Workloads > Retail Website - North America

## Retail Website - North America

Delete workload

Review | Improvement plan | **Milestones** | Properties

### Milestones

Generate report View milestone

< 1 > ⚙

	Name	Milestone status	High risks	Medium risks	Date saved
<input type="radio"/>	Version 1.1	☑ Answered	1	2	Jun 5, 2019 1:47 PM UTC-7
<input type="radio"/>	Version 1.0 - initial review	☑ Answered	1	3	Jun 5, 2019 1:24 PM UTC-7

# 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 items identified.

**Medium risks**

The number of medium risk items 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 and IAM users, 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**.



## 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 \(p. 4\)](#).

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 and IAM users in the same AWS Region.

#### Note

You can only share workloads within the same AWS Region.

### To share 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** and choose **Create** 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. Shared access to the workload is not removed until the workload invitation is deleted.

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

To see who has shared access to a workload, choose **Shares** from the [Workload Details \(p. 25\)](#) page.

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

## 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.

Shared access to a workload is not removed until the workload invitation is deleted.

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 for the IAM user 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.

## 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. Shared access to the workload is not removed until the workload invitation is deleted.

## 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. Shared access to the workload is not removed until the workload owner deletes the workload invitation.

**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. Shared access to the workload is not removed until the workload owner deletes the workload invitation.

To reject shared access after a workload invitation has been accepted, choose **Reject share** from the [Workload Details \(p. 25\)](#) 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 \(p. 25\)](#)
- [Milestones Tab \(p. 25\)](#)
- [Properties Tab \(p. 26\)](#)
- [Shares Tab \(p. 26\)](#)

## 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 \(p. 4\)](#).

## 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.

### Note

The principal retains shared access to the workload until the workload invitation is deleted.

### 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**. You must delete a workload invitation to remove shared access to the workload.

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

# 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.

# 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 **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.

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.

## Adding a Lens

### 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 that you want to add and choose **Edit**.
5. Select the lens to add and choose **Save**.

## Removing a Lens

### 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. Select the lens that you want to remove and choose **Edit**.
5. Unselect the lens 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.

## Lens Upgrades

The AWS Well-Architected Framework Lens and other lenses 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.

A lens upgrade might 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

## 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 also see a banner indicating that a new lens version is available.

## Selecting the 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.

### 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 AWS best practices. Upgrade to the current lens version.

### Version used

The lens version currently used for the workload.

### Current available version

The lens version available for upgrade.

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

## Upgrading the 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.

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.

# Workload Invitations

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

## Note

Workloads can only be shared within the same AWS Region.

The owner of the workload controls who has shared access. Access to the workload is not removed until the owner deletes the workload invitation.

The **Workload invitations** page, available from the left navigation, provides information about your pending workload invitations.

The following information is displayed for each workload invitation:

### Workload

The name of the workload to be shared.

### 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.

- Contributor

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

### Permission details

Detailed description of the permission.

## Accepting a Workload Invitation

### To accept a workload invitation

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

The workload is added to the **Workloads** and **Dashboard** pages.

You have seven days to accept a workload invitation. If you do not accept the invitation within seven days, it's automatically expired. Shared access to the workload is not removed until the workload owner deletes the workload invitation.

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 Workload Invitation

### To reject a workload invitation

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

The workload invitation is removed from the list.

Shared access to the workload is not removed until the workload owner deletes the workload invitation.

# Dashboard

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

### Resources

Total workload reviews

6

With high risks

3

With medium risks

4

### Workload reviews

Filter by risk

1

	Name	High risks	Medium risks	Last updated
<input type="radio"/>	Internal Employee Portal	3	8	Jun 5, 2019 1:09 PM UTC-7
<input type="radio"/>	Mobile app - Android	0	1	Jun 4, 2019 1:38 PM UTC-7
<input type="radio"/>	Mobile App - iOS	0	0	Jun 4, 2019 1:41 PM UTC-7
<input type="radio"/>	Prototype replacement website	0	0	Jun 4, 2019 1:42 PM UTC-7
<input type="radio"/>	Retail Website - EU	13	18	Jun 4, 2019 1:31 PM UTC-7
<input checked="" type="radio"/>	Retail Website - North America	1	3	Jun 5, 2019 1:24 PM UTC-7

### Milestones (Retail Website - North America)

1

Name	Milestone status	High risks	Medium risks	Date saved
Version 1.0 - initial review	<input checked="" type="radio"/> Answered	1	3	Jun 5, 2019 1:24 PM UTC-7

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 items identified.

**Medium risks**

The number of medium risk items 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 items identified.

**Medium risks**

The number of medium risk items 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 \(p. 37\)](#)
- [Identity and Access Management for AWS Well-Architected Tool \(p. 38\)](#)
- [Compliance Validation for AWS Well-Architected Tool \(p. 47\)](#)
- [Resilience in AWS Well-Architected Tool \(p. 48\)](#)
- [Infrastructure Security in AWS Well-Architected Tool \(p. 48\)](#)

## Data Protection in AWS Well-Architected Tool

AWS Well-Architected Tool conforms to the AWS [shared responsibility model](#), which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all the AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN Partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

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), so that 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.
- 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.

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into 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 AWS WA Tool or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

For more information about data protection, see the [AWS Shared Responsibility Model and GDPR](#) blog post on the *AWS Security Blog*.

## 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 \(p. 39\)](#)
- [Authenticating With Identities \(p. 39\)](#)
- [Managing Access Using Policies \(p. 40\)](#)
- [How AWS Well-Architected Tool Works with IAM \(p. 42\)](#)
- [AWS Well-Architected Tool Identity-Based Policy Examples \(p. 44\)](#)
- [Troubleshooting AWS Well-Architected Tool Identity and Access \(p. 47\)](#)

## Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work 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 \(p. 47\)](#).

**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 \(p. 42\)](#).

**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 \(p. 44\)](#).

## 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.

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](#). 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.

## Identity-Based Policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, role, or group. These policies control what actions that identity 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 such as an Amazon S3 bucket. Service administrators can use these policies to define what actions a specified principal (account member, user, or role) can perform on that resource and under what conditions. Resource-based policies are inline policies. There are no managed resource-based policies.

## Access Control Lists (ACLs)

Access control lists (ACLs) are a type of policy that controls 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](#) (p. 42)
- [AWS WA Tool Resource-Based Policies](#) (p. 43)
- [Authorization Based on AWS WA Tool Tags](#) (p. 44)
- [AWS WA Tool IAM Roles](#) (p. 44)

## 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

The `Action` element of an IAM identity-based policy describes the specific action or actions that will be allowed or denied by the policy. Policy actions usually have the same name as the associated AWS API operation. The action is used 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 *IAM User Guide*. Use these actions in IAM policies to DENY action, and not for granting permissions.

### Resources

The `Resource` element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. You specify a resource using an ARN or using the wildcard (\*) to indicate that the statement applies to all resources.

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 *IAM User Guide*. 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 *IAM User Guide*.

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

## Examples

To view examples of AWS WA Tool identity-based policies, see [AWS Well-Architected Tool Identity-Based Policy Examples \(p. 44\)](#).

## AWS WA Tool Resource-Based Policies

AWS WA Tool does not support resource-based policies.



## Authorization Based on AWS WA Tool Tags

AWS WA Tool does not support tagging resources or controlling access based on tags.

## 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. 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](#) (p. 44)
- [Using the AWS WA Tool Console](#) (p. 45)
- [Allow Users to View Their Own Permissions](#) (p. 45)
- [Granting Full Access to Workloads](#) (p. 46)
- [Granting Read-only Access to Workloads](#) (p. 46)
- [Accessing One Workload](#) (p. 46)

## 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*.

## 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/99999999999955555555555566666666"
}
```

## 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 \(p. 47\)](#)
- [I'm an Administrator and Want to Allow Others to Access AWS WA Tool \(p. 47\)](#)

### 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 the [Amazon Web Services: Overview of Security Processes](#) whitepaper.

# Document History

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

- **Latest documentation update:** May 21, 2020

update-history-change	update-history-description	update-history-date
<a href="#">Content update (p. 38)</a>	Section on how AWS uses your data was added.	May 21, 2020
<a href="#">Updated functionality (p. 49)</a>	This release adds a review owner to the workload.	April 1, 2020
<a href="#">Updated functionality (p. 49)</a>	This release adds an architectural diagram link to the workload.	March 10, 2020
<a href="#">Content update (p. 49)</a>	Clarified that workload shares are AWS Region-specific.	January 10, 2020
<a href="#">Updated functionality (p. 49)</a>	This release adds workload sharing.	January 9, 2020
<a href="#">Content update (p. 49)</a>	Security section updated with latest guidance.	December 6, 2019
<a href="#">Updated functionality (p. 49)</a>	This release makes the industry fields optional when defining a workload.	August 19, 2019
<a href="#">Updated functionality (p. 49)</a>	This release adds improvement plan items to the workload report.	July 29, 2019
<a href="#">Updated functionality (p. 49)</a>	The release adds the DeleteWorkload action to the policy.	July 18, 2019
<a href="#">Content update (p. 49)</a>	The content in this guide has been updated with minor fixes.	June 19, 2019
<a href="#">Content update (p. 49)</a>	The content in this guide has been updated with minor fixes.	May 30, 2019
<a href="#">Updated functionality (p. 49)</a>	This release supports upgrading the version of the framework used for a workload review.	May 1, 2019
<a href="#">Updated functionality (p. 49)</a>	This release adds the ability to specify non-AWS Regions when defining a workload.	February 14, 2019
<a href="#">AWS Well-Architected Tool general availability (p. 49)</a>	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*.