aws

User Guide

# Amazon One

# Amazon One: User Guide

# Table of Contents

# What is Amazon One Enterprise?

Amazon One Enterprise is a new palm-based authentication service that provides employees with secure access to buildings and enterprise assets, without the use of badges, PINs, or passcodes.

**Topics**



# Amazon One device

The Amazon One device is designed for Amazon One Enterprise, a secure, palm-based identity service for enterprise access control. Note the following device specifications:

- User inputs — Palm Biometrics, QR Code matching
- Host interface — Wi-Fi (2.4 GHz and 5 GHz), Ethernet, 2x USB Type-A, 1 USB Type-B
- User feedback — 5.5” Touchscreen, Lightring, speaker, headphone
- Physical Access Control Protocol — OSDP and Wiegand
- Power supply — POE, 110/220 VAC input AC to DC adaptor provided, 30W @ 15V
- Security — Tamper switches
- Dimension (HxWxD mm) — 86 x 85 x 256


# Amazon One Enterprise console

Amazon One Enterprise includes a console, which can be used in the following ways:

- An IT or facility manager uses Amazon One Enterprise to create and manage a site. The site resembles a physical location for the tasks that the team performs while monitoring and managing Amazon One Enterprise devices and user profiles. The IT or facility manager tasks include:
  - Creating a site to contain all of the Amazon One device instances in a physical location
  - Adding an admin user to manage the site, and an installer user to access activation QR codes
- An admin uses Amazon One Enterprise to create device instances and to manage Amazon One devices. Admin tasks include:
  - Creating a device instance under a site

  - Creating a configuration template to be applied to a device instance
  - Monitoring device health and updating device configurations
  - Canceling user enrollments

- An installer uses Amazon One Enterprise to access activation QR codes to activate devices. Installer tasks include:
  - Accessing an activation QR code on the console
  - Selecting a QR code that corresponds to the device instance to be activated
  - Scanning the selected QR code with the Amazon One device installed

## Purchasing Amazon One devices

Contact us to learn more about Amazon One Enterprise, and a Business Development team member will get in touch to share more details about our offering, including pricing, and answer any questions that you may have.

## Amazon One Enterprise pricing

Contact us to learn more about Amazon One Enterprise pricing.

# How Amazon One works

Amazon One is a cloud-based biometric service that uses an Amazon One device to authenticate a user with their palm biometrics. You can order Amazon One devices by contacting us.

After installing the Amazon One device, you can activate and register your devices with your AWS account on the Amazon One Console and the authentication application. You can view enrolled user biometric profiles. If needed, you can cancel their enrollment and delete their biometric data.

The Amazon One Console serves as a centralized hub for managing operational activities, such as tracking devices and viewing monthly bills. Users can enroll by scanning their palms at supervised enrollment stations on-site. Once enrolled, users can seamlessly enter or exit secure locations by hovering their palm over an Amazon One enabled device.

**Topics**



## Amazon One workflow

The following details the basic workflow of Amazon One:

1. Purchase and install the Amazon One devices by contacting us.
2. After installing the device, activate Amazon One.
3. Sign in to your Amazon One account.
4. Configure user enrollment and entry devices.
5. Enroll employee palms.
6. Use management and monitoring features to ensure device health, keep configurations up to date, and track user enrollments for comprehensive oversight.

## Amazon One key terms

These are the key terms for Amazon One:

- Site — The customer managed physical buildings where the customer installs Amazon One devices. A site must meet the facility, networking, and power requirements for your Amazon One devices.
- Device — An Amazon One palm scanning biometric device for authentication.
- Device Instance — A logical representation of a device with configurations. Use of device instances allows for swapping Amazon One devices while automatically inheriting the previously set configurations and names. A device instance has a user-defined name (shared naming convention with your access control software) and a set of communication configurations. Device instances have three primary states:
  - Needs configuration
  - Ready for activation
  - Active
- Configuration Template — An all-inclusive set of configurations applied on a device instance.

# Setting up the Amazon One console

This chapter explains the basic steps to get started with Amazon One console.

**Setting up a site, device instances, and configuration templates**—Follow these steps to create a framework for adding a physical location to house your Amazon One devices, and then to configure and manage them using the Amazon One Enterprise console. You'll use this process only occasionally, or even just once, depending on the number of sites, device instances, and your configuration templates.

**Topics**



# Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

**To sign up for an AWS account**

1. Open https://portal.aws.amazon.com/billing/signup
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an AWS account root user is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform tasks that require root user access

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**

# Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

**Topics**



## Securing your AWS account

Now that you've signed in to your Amazon One account, secure your account.

**To secure your AWS account root user**

1. Sign in to the AWS Management Console as the account owner by choosing Root user and entering your AWS account email address.
2. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the AWS Sign-In User Guide.
3. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the IAM User Guide.

## Creating a user with administrative access

Now that you've secured your Amazon One account, create a user with administrative access.

**To create a user with administrative access**

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the AWS IAM Identity Center User Guide.
2. In IAM Identity Center, grant administrative access to a user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the AWS IAM Identity Center User Guide.

## Signing in as an administrator

Now that you've created a user with administrative access, sign in as an administrator.

**To sign in as the user with administrative access**

- Sign in with your IAM Identity Center user, using the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

  For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the AWS Sign-In User Guide.

## Assigning access to additional users

Now that you've signed in as an administrator, you can assign access to additional users.

**To assign access to additional users**

- Assign users to a group, and then assign single sign-on access to the group.

  For instructions, see Add groups in the AWS IAM Identity Center User Guide.

# Add Amazon One users

In addition to admin users, you can also add users who lack admin permissions. For example, these users might be installers who access the Amazon One console only to retrieve device activation QR codes to activate Amazon One devices.

**To add an Amazon One user**

1. Follow the sign-in procedure appropriate to your user type as described in How to sign in to AWS in the *AWS Sign-In User Guide*.
2. In the navigation pane, select **Users**, and then select **Add users**.
3. On the **Specify user details** page, under **User details**, in **User name**, enter the name for the new user. This is their sign-in name for AWS.

> **Note**
>
> The number and size of IAM resources in an AWS account are limited. For more information, see IAM and AWS STS quotas. User names can be a combination of up to 64 letters, digits, and the following characters: plus (+), equal (=),comma (,), period (.), at sign (@), underscore (_), and hyphen (-). Names must be unique within an account. They are not distinguished by case. For example, you cannot create two users named *TESTUSER* and *testuser*. When a user name is used in a policy or as part of an ARN, the name is case sensitive. When a user name appears to customers in the console, such as during the sign-in process, the user name is case insensitive.

4. You are asked whether you are providing console access to a person. Select **Provide user access to the – AWS Management Console *optional***.
5. Select **I want to create an IAM user**.
6. For **Console password**, select one of the following:

   - **Autogenerated password** – The user is given a randomly generated password that meets the account password policy. You can view or download the password when you get to the **Retrieve password** page.
   - **Custom password** – The user is assigned the password that you enter in the field.

7. (Optional) By default, **Users must create a new password at next sign-in (recommended)** is selected to ensure that the user is required to change their password the first time they sign in.

> **Note**
>
> If an administrator has enabled the **Allow users to change their own password** account password policy setting, then this check box does nothing. Otherwise, it

automatically attaches an AWS managed policy named `IAMUserChangePassword` to the new users. The policy grants them permission to change their own passwords.

8. Select **Next**.
9. On the **Set permissions** page, choose **Attach policies directly.**
10. Select the policies that you want to attach to the user.

    - AmazonOneEnterpriseReadOnlyAccess
    - AmazonOneEnterpriseInstallerAccess

    **Note**

    AmazonOneEnterpriseInstallerAccess managed policy will provide user access to activation QR codes *only* in the Amazon One Enterprise console. This policy is ideal for enterprises that hire a third party to install Amazon One devices.

11. Select **Next**.
12. (Optional) On the **Review and create** page, under **Tags**, select **Add new tag** to add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM resources.
13. Review all of the choices that you made up to this point. When you are ready to proceed, select **Create user**.
14. On the **Retrieve password** page, get the password assigned to the user:

    - Select **Show** next to the password to view the user's password so that you can record it manually.
    - Select **Download .csv** to download the user's sign-in credentials as a .csv file that you can save to a safe location.

15. Select **Email sign-in instructions**. Your local mail client opens with a draft that you can customize and send to the user. The email template includes the following details for each user:

    - User name
    - URL to the account sign-in page. Use the following example, substituting the correct account ID number or account alias:

```
https://AWS-account-ID or alias.signin.aws.amazon.com/console
```

> **Important**
>
> The user's password is ***not*** included in the generated email. You must provide the password to the user in a way that complies with your organization's security guidelines.

# Create a site

Now that you've signed in to the AWS Management Console, you can use the Amazon One console to create your site.

> **Important**
>
> Amazon One is available only in the US East (N. Virginia) Region.

**To create a site**

1. Open the Amazon One console at https://console.aws.amazon.com/one-enterprise.
2. Choose **Go to Overview**.
3. In the navigation pane, choose **Sites**.
4. Choose **Create sites**.
5. Under **Site information**, for **Site name**, enter a name for the site.
6. Under **Physical address**, enter the address for the site where your Amazon One devices will be installed.
7. (Optional) To add a tag to the site, enter a key-value pair under **Tags**, and then choose **Add new tag**. To remove this tag before creating the site, choose **Remove**.
8. Choose **Create site** to create the site.

# Create device instances

Now that you've created a site in the AWS Management Console, you can use the Amazon One console to create device instances.

**To create a device instance**

1. Open the Amazon One console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **device instances**. Make sure you are on the **Unactivated instances** tab.
3. Under **Instance details**, choose a site from the **Site** drop-down, or create a new site by choosing the **Create site** button.
4. Manually input each individual **Device instance name**.
5. (Optional) To add a tag to the device instance, enter a key-value pair under **Tags**, and then choose **Add new tag**. To remove this tag before creating the device instance, choose **Remove**.
6. Choose **Create instances** to create the device instances.

> **Note**
>
> Note: device instances need to be configured before installation can occur.

# Create a configuration template

Now that you've created device instances, you can use the Amazon One console to create a configuration template.

**To create a configuration template**

1. Open the Amazon One console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **Configuration templates**.
3. Choose **Create template**.
4. Under **Template information**, for **Template name**, enter a name for the configuration template.
5. Under **Device configurations**, select an **Operation mode**.

To configure Enrollment operating mode

1. (Optional) Under **Wifi configuration**, provide your **Wifi credentials**.
2. (Optional) To add a tag to the site, enter a key-value pair under **Tags**, and then choose **Add new tag**. To remove this tag before creating the site, choose **Remove**.
3. Choose **Configure**.

To configure Entry operating mode

1. Under **Control panel settings**, provide the communication settings for Amazon One devices to communicate with your control panel.
2. Under **Badge format settings**, provide the configuration settings that specify the layout of your company badge format.
3. (Optional) Under **Wifi configuration**, provide your **Wifi credentials**.
4. (Optional) To add a tag to the site, enter a key-value pair under **Tags**, and then choose **Add new tag**. To remove this tag before creating the site, choose **Remove**.
5. Choose **Configure**.

> **Important**
>
> You must configure at least one Enrollment device and one Entry device to enable the full capabilities of Amazon One for secure access.

# Configure a device instance for activation

After a device instance is created, you configure the device instance with a previously created configuration template (see Create a configuration template), or you can add configurations manually.

**To configure a device instance for activation**

1. Open the Amazon One console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **Device instances**. Make sure you are on the **Unactivated instances** tab.

3. Select one or more instances to configure.
4. Choose **Configure**.
5. Under **Device Configurations**, select one of the two input methods:

   a. For the **Use template** option, choose a template from the drop-down. Review or make changes to this imported configuration information.

      For the **Create template** option, see Create a configuration template.

   b. For the **Manually input** option, select an **Operating mode**.

      To configure Enrollment operating mode

      a. (Optional) Under **Wifi configuration**, provide a **Wifi credential**.
      b. (Optional) To add a tag to the site, enter a key-value pair under **Tags**, and then choose **Add new tag**. To remove this tag before creating the site, choose **Remove**.
      c. Choose **Configure**.

      To configure Entry operating mode

      a. Under **Control panel settings**, provide the communication settings for Amazon One devices to communicate with your control panel.
      b. Under **Badge format settings**, provide the configuration settings that specify the layout of your company badge format.
      c. (Optional) Under **Wifi configuration**, provide a **Wifi credential**.
      d. (Optional) To add a tag to the site, enter a key-value pair under **Tags**, and then choose **Add new tag**. To remove this tag before creating the site, choose **Remove**.
      e. Choose **Configure**.

6. Under the **Unactivated instances** table, the Instance state should show

7. Validate that activation QR codes are available for activation. In the navigation pane, choose **Activation QR Code**.
8. From the **Select a site** drop-down list, select a **Site**.
9. Under **Site information**, validate the Site address.

10. Under **Activation QR codes**, each device instance has a corresponding QR code. Choose **Get QR code** to show the activation QR codes.

> **Important**
>
> You must configure at least one Enrollment device and one Entry device to enable the full capabilities of Amazon One for secure access.

# Installing and activating Amazon One

After successfully setting up your Amazon One console, the next steps involve installing Amazon One devices at your site and ensuring they are properly activated. This process includes physically placing the devices in designated areas, connecting them to your network, and completing the activation process to enable seamless user identification and transaction capabilities. Once activated, your Amazon One devices will be ready to deliver a secure, touchless experience for your customers or employees.

> **Note**
>
> This section focuses on installation, and uses a mobile browser to access AWS Management Console to obtain device activation QR codes.

**Topics**



# Understanding requirements

An Amazon One device can be installed in any corporate or business location that has doors that can be electrically controlled.

## Control panel requirement

Amazon One devices can connect to most standard access control panels as a reader. Amazon One devices support the following protocols:

- OSDP (v1 and v2)
- Wiegand

## Network requirement

Amazon One devices must always be connected to the internet for normal operation. Internet connectivity can be provided by either wired Ethernet or Wi-Fi. The minimum required bandwidth is 10 Mbps.

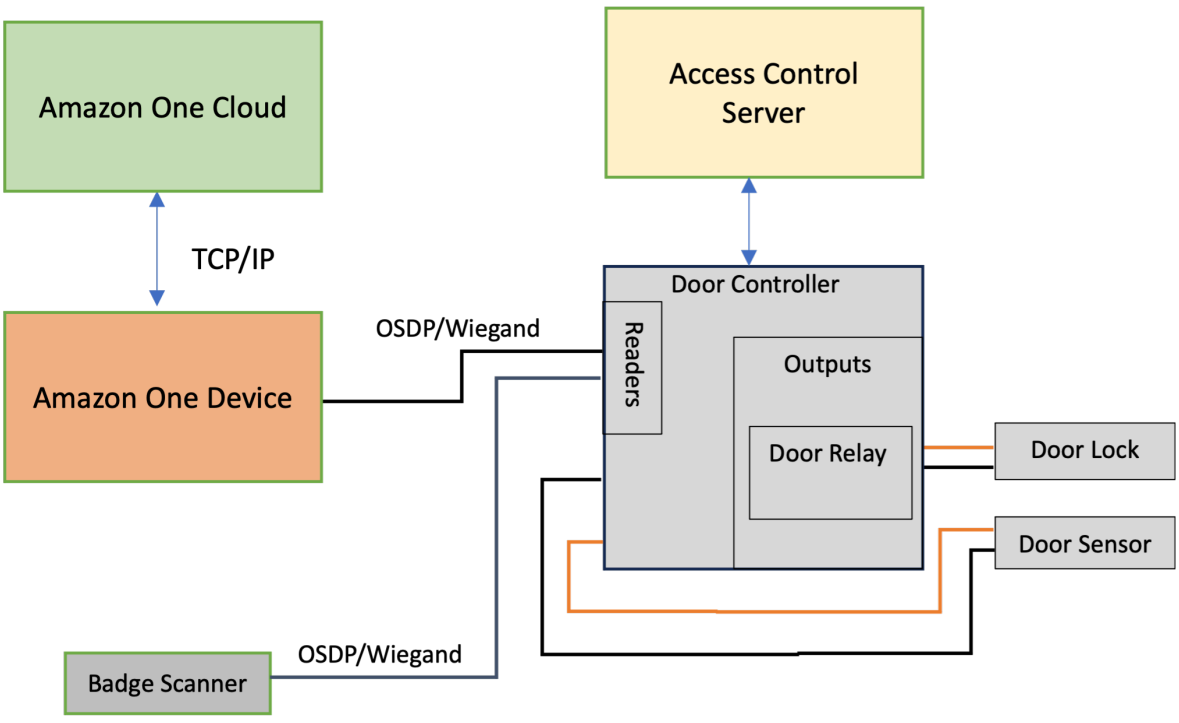
## Power requirement

Amazon One devices can be powered in one of two ways:

- By using the 120V power adaptor provided in the box.
- By using a PoE+ enabled device.

# Understanding installation concepts

To properly secure building access, Amazon One recommends that you install the device as part of a typical access control environment, as described in the following block diagram.




An access control environment typically consists of these components:

- Amazon One device: This is the palm recognition device that will perform biometric authentication to identify the individual who is attempting to gain access to a secure area of the building.
- Access control server: This component typically controls the access rights of users to the secure area. The badge IDs of individuals who have access to the area are stored on this server. This server caches the relevant IDs to the appropriate door controllers.
- Door controller:
  - An Amazon One device connects to the door controller server through an OSDP interface.
  - If a Wiegand interface is necessary, a COTS OSDP-to-Wiegand converter can be used.
  - Upon successful authentication, the Amazon One device sends the badge ID of the user to the door controller.
  - The door controller responds with a decision, which then allows the Amazon One device to display either an Access Granted or Access Denied message.
- Badge scanner: A badge scanner is typically used to scan RFID badges and send the badge number to the access control server. With Amazon One, a badge scanner connects to the Amazon One device, allowing users to scan their badges, which associates them with their palm profiles.

# Installing Amazon One Pedestal

The Amazon One Pedestal is a key component of the Amazon One identification and transaction system, designed to deliver a seamless, touchless experience for users. This device features secure biometric authentication. You can integrate it into various locations to provide frictionless access or payment solutions.

This section provides the location requirements and step-by-step instructions for installing Amazon One Pedestal. Proper preparation and installation are key to ensuring the system operates securely and efficiently, providing users with a smooth, reliable experience.

**Prerequisites and preparation for installing the Amazon One Pedestal**

Before starting the installation, ensure the following conditions are met for a safe, secure, and effective setup:

- Power requirements: If you use POE+ (Power over Ethernet) to power the device, verify that Cat6 cabling is already installed, and a POE+ injector or switch is available for use. Alternatively, if AC power (120V) is being used, ensure that an accessible AC outlet is located within 20 feet of the pedestal.
- Physical setup: The floor must be level, clean, and free of any debris to ensure stable and safe pedestal installation.

- Pedestal location: Install the pedestal in a location where it will not block doors, lanes, or access points, allowing for easy movement around the area.
- Cable management: Route and secure all excess cables inside the pedestal to avoid clutter and prevent any potential damage during normal use.

Once these prerequisites are confirmed, you can proceed with the installation process.

**To install Amazon One Pedestal**

1. Remove the Amazon One Pedestal from the packaging.
2. Remove the door by unscrewing both M4 tamper resistant screws.
3. Plug in the power cable.
4. Route the cable through the hole in the pedestal base plate.
5. Coil any excess power cable inside the pedestal.
6. Route the Ethernet cable (Cat5E or better) through the bottom plate of the pedestal and plug into the Ethernet port.
7. Install a ferrite loop on the Ethernet cable 2 inches above the base of the pedestal.
8. Feed RS485 serial cable from the access control panel (or the badge reader) to the pedestal, with 1 ft excess in length.
9. Install a ferrite loop on the RS485 cable 2 inches above the base of the pedestal.
10. Plug in power to the outlet and confirm that the Amazon One device turns on.
11. Reattach the door to the pedestal and rescrew the two M4 tamper resistance screws to secure.

After installing your Amazon One device, you are ready to activate the device.

# Installing the wall-mountable Amazon One device

The wall-mountable Amazon One device is a versatile, compact biometric identification system designed to provide a seamless, touchless experience for users in various environments. It uses advanced palm recognition technology for secure access or payment, making it ideal for high-traffic locations such as retail spaces, office entrances, and more.

This section outlines the necessary location requirements and detailed steps for installing the wall-mountable Amazon One device to ensure optimal performance and security.

### Prerequisites and preparation for installing the wall-mountable Amazon One device

Before you begin the installation, ensure that the following conditions are met to guarantee the device operates effectively and is properly set up within your space:

- Indoor use only: The wall-mountable Amazon One device is intended for indoor use only, so ensure it is being installed in an appropriate environment.
- Wall requirements: The wall must be level to ensure proper alignment and functionality of the device.
- Mounting height: The top of the wall mount should be positioned no higher than 44-46 inches from the ground after installation, ensuring ease of access for users.
- Cable management: Ensure that all excess cables are routed behind the wall mount and securely fastened to prevent damage or clutter.
- Power Over Ethernet (PoE++): If using Power Over Ethernet (PoE++), verify that an IEEE 802.3bt (Type 3) Class 6 PoE++ switch (end span) or injector (midspan) is available. The PoE++ source must be listed or certified and comply with IEC 62368-1 standards. Importantly, the PoE++ source must be located within the same building as the device. Only use an approved PoE++ source with the AOE device.
- 15V DC Power Input: If using 15V DC power input, ensure that only an NEC Class 2 or a power-limited approved power supply is used. The power supply must be listed or certified for safety and compatibility.

### Required tools

- 1/4″ dry wall or masonry drill bit if wall anchors are required
- Wire stripper
- 7/64″ drill bit for drilling pilot holes
- #2 Phillips screwdriver
- 0.5mm x 2mm flathead screwdriver
- T12 Secure Torx Driver
- Pencil
- Level

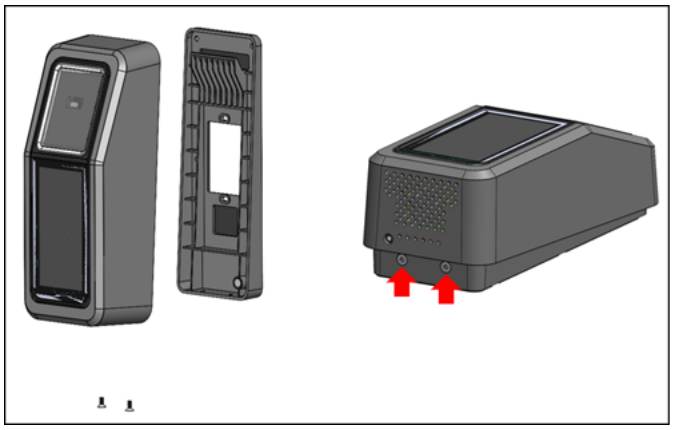
### Included with the wall-mountable Amazon One device

- 6x #8 Drywall anchors
- 6x #8-32 1in long screws
- 2x #6-32 1in Machine Screws
- 2x 6 Position terminal block connectors
- 2 Torx Security M4x10 flathead screws

Once these prerequisites are confirmed, you can proceed with the installation steps to securely mount and configure the wall-mountable Amazon One device.

**To install the wall-mounting plate for your Amazon One device**

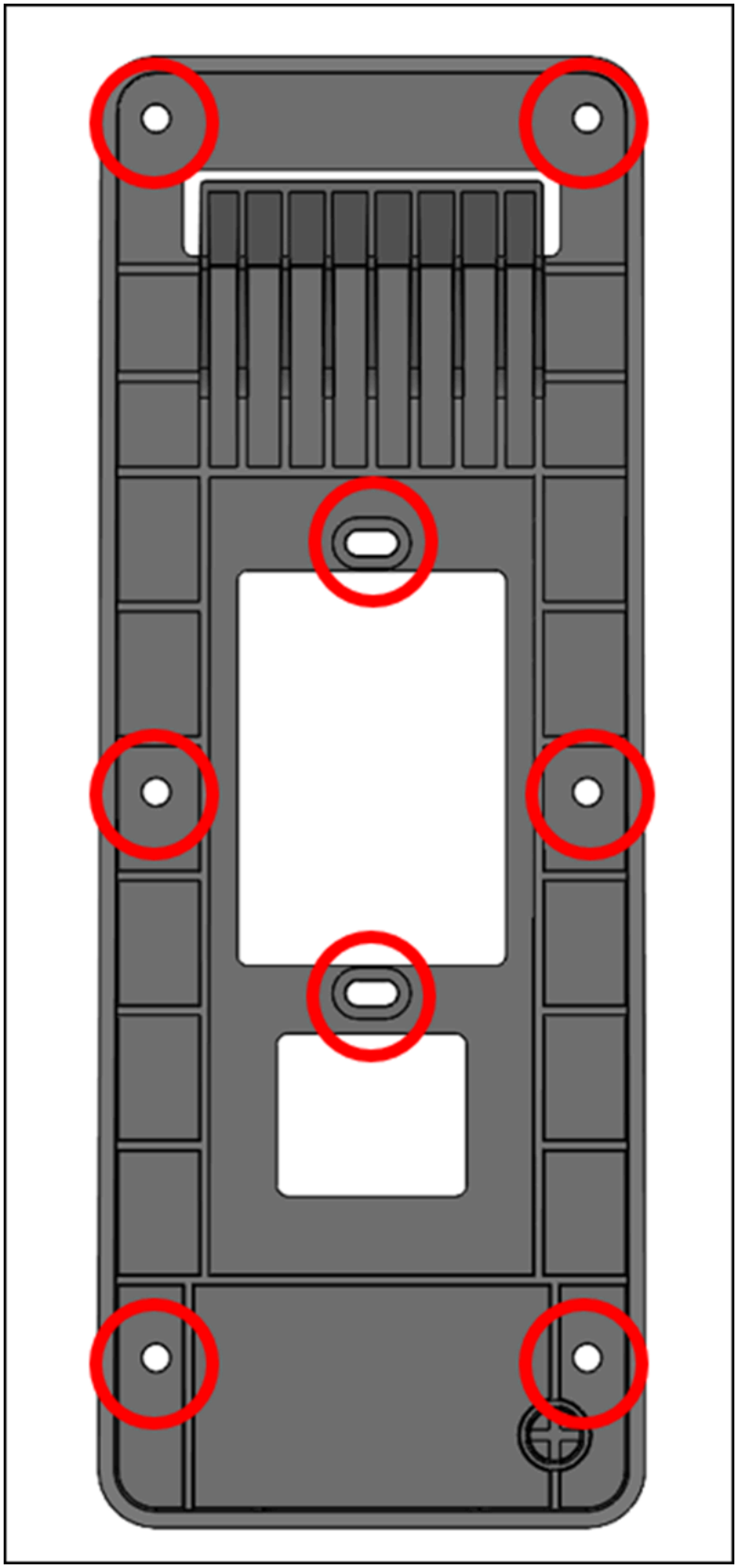
1. Remove your Amazon One device from the packaging.
2. Separate the mounting plate from your Amazon One device by removing the two bottom Torx security screws.



3. Position the mounting plate on the wall in the desired location. Use the bracket as a template to mark the outer six screw holes as shown in the following image.

   (Optional) If a single gang box is available in the installation position, perform the following:

   - Loosely mount the plate to the gang box by inserting the included #6-32 machine screws through the oblong holes.
   - Ensure the mounting plate is level.
   - Use the mounting plate as a template to mark the six screw positions with a pencil. You can use the oblong holes and #6-32 screw as extra support for the mounting plate. Don't use the #6-32 screw positions as the primary means of mounting the wall plate.



4. If mounting into stucco, drywall, brick, or concrete surfaces, drill 1/4” holes at each marked location, and then install wall anchors by pressing them into the hole until the anchor is flush with the wall.

   If mounting onto a wooden surface, the anchors are not required and only 7/64” pilot holes are required in the marked locations.
5. Loosely fasten the wall plate to the wall using the #8 wood screws in the anchor positions.
6. After all the fasteners are in place, ensure the mounting plate is level.
7. Tighten the screws to secure the mounting plate to the wall.

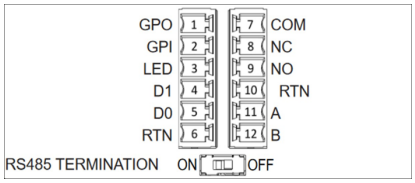
**To connect your wall-mountable Amazon One device**

You can configure Amazon One device with OSDP and Weigand access control protocols. To simplify installation, Amazon One device utilizes terminal block connectors (Mfg P/N: Phoenix Contact 1767694). You also have the option to configure Amazon One device to directly control external devices by using the internal relay or the General Purpose Input and Output connections.

1. To determine the appropriate wiring configuration for your application, refer to the following diagram and Connections Table.

   For detailed electrical characteristics of the signals, refer to the Wiring instructions.

   **Connections**




| Pin | Connection | Description | Use |
|---|---|---|---|
| 1 | GPO | General purpose output | Digital output signal - Optional |
| 2 | GPI | General purpose input | Digital input signal – Optional |

| Pin | Connection | Description | Use |
|---|---|---|---|
| 3 | LED | Wiegand LED | Wiegand LED – Optional |
| 4 | D1 | Wiegand D1 | Wiegand data 1 – White wire |
| 5 | D0 | Wiegand D0 | Wiegand data 0 – Green wire |
| 6 | RTN | Signal return | Wiegand Ground – Black wire |
| 7 | Com | Relay common | Contact relay common – White wire |
| 8 | NC | Relay normally closed | Contact relay normally closed – Orange wire |
| 9 | NO | Relay normally open | Contact relay normally open – Yellow wire |
| 10 | RTN | Signal return | OSDP return – Black wire |
| 11 | A | RS485_A/D1/ Clock | OSDP D1 – White wire |
| 12 | B | RS485_B/D0/ Data | OSDP D0 – Green wire |

2. When installing a wire, strip 3mm-5mm off the end of the wire.
3. Insert the stripped end of the wire into the desired terminal position.

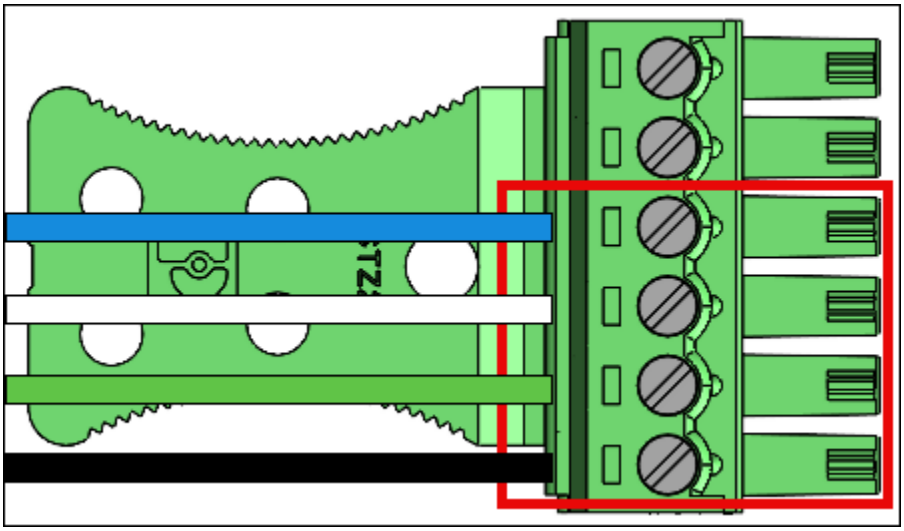
4. Using a flathead screwdriver, turn the terminal retention screw clockwise to clamp down on the wire until it is snug. Do not over tighten.
5. After fastening, gently tug on the wire to ensure that it is seated.
6. After you make the necessary connections, insert the plug into the corresponding receptacle of your Amazon One device terminal block.
7. Insert the Cat6 Ethernet cable to RJ45 jack.
8. Position Amazon One device so the hook on the wall plate slides into the opening on the rear of the device.
9. Ensure the cables are not caught between the device and the mounting plate, and let the device pivot and seat into position.
10. Secure your Amazon One device to the mounting plate with two Torx Security M4x10 flathead screws.
11. Hand tighten the screws. Don't over tighten.

**To wire your wall-mountable Amazon One device**

Install only the required wires for your application.

**Wiegand connections**

- Insert the blue wire in Pin 3 (LED).
- Insert the white wire in Pin 4 (D1).
- Insert the green wire in Pin 5 (D0).
- Insert the black wire in Pin 6 (RTN).



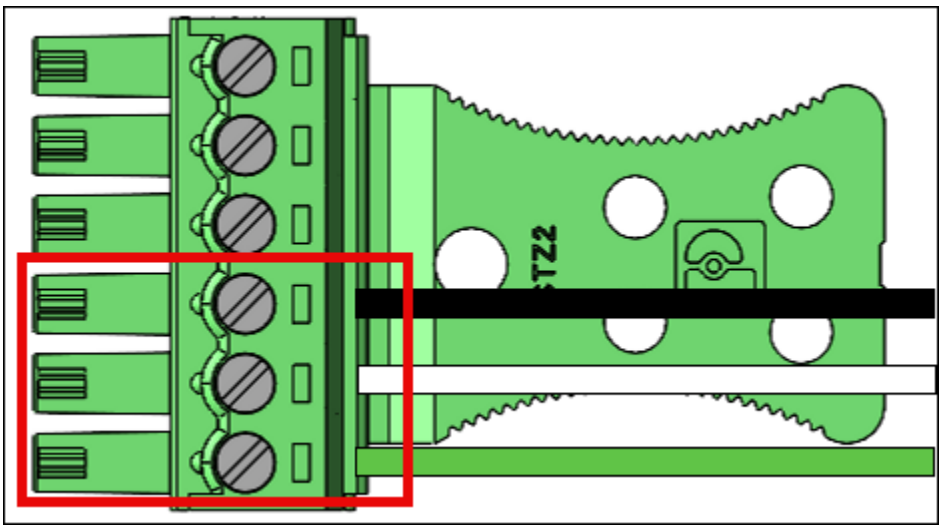
### Wiegand output wiring

| Pin | Connection | Description | Use |
|---|---|---|---|
| 3 | LED | Wiegand LED | Wiegand LED input – Optional (5V TTL) |
| 4 | D1 | Wiegand D1 | Wiegand D1 output (5V TTL) |
| 5 | D0 | Wiegand D0 | Wiegand D0 output (5V TTL) |
| 6 | RTN | Signal return | Wiegand GND reference |

Turn RS485 termination switch “ON” if the device is the last unit on the line. This switch activates 120 Ohms resistor termination on the line.

### RS485 connections

- Insert the black wire in Pin 10 (RTN).
- Insert the white wire in Pin 11 (A).
- Insert the green wire in Pin 12 (B).

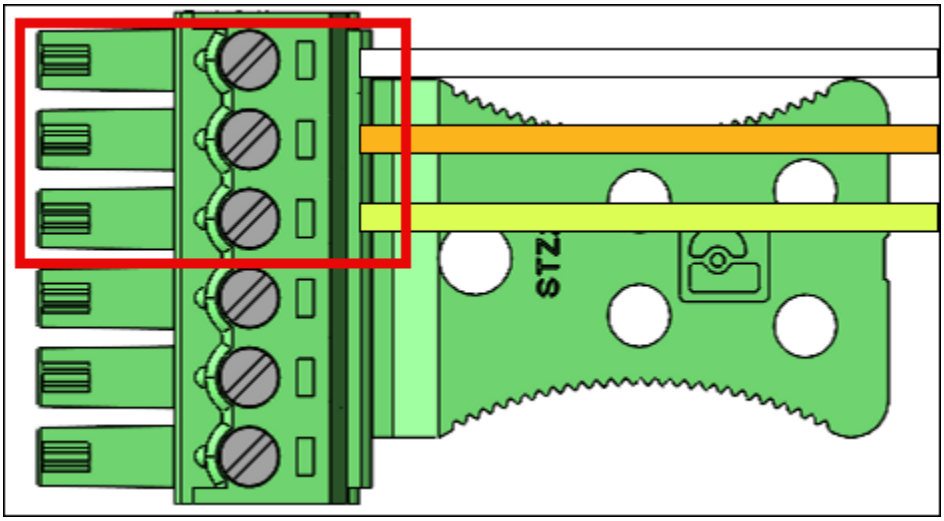


### RS485 wiring

| Pin | Connection | Description | Use |
|---|---|---|---|
| 10 | RTN | Signal return | Ground |
| 11 | A | RS485_A/D1/ Clock | RS485 non-inver ting signal |
| 12 | B | RS485_B/D0/ Data | RS485 inverting signal |

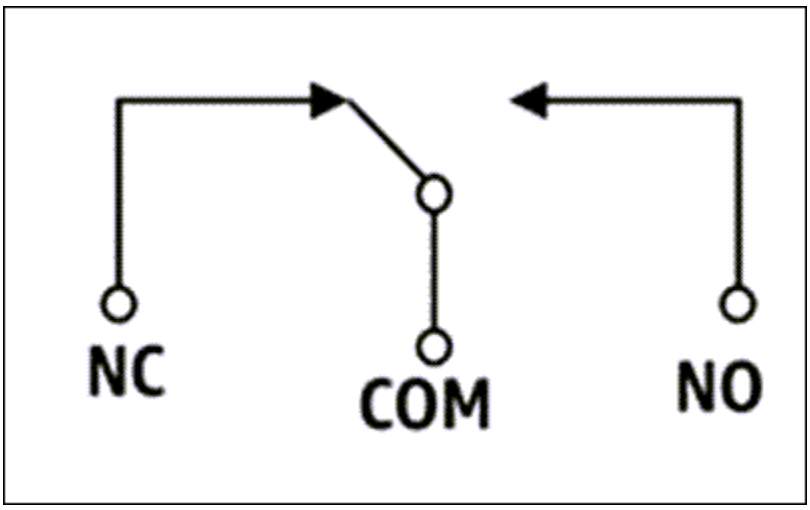
### Relay connections

- Insert the white wire in Pin 7 (COM).
- Insert the orange wire in Pin 8 (NC).
- Insert the yellow wire in Pin 9 (NO).



## Relay wiring

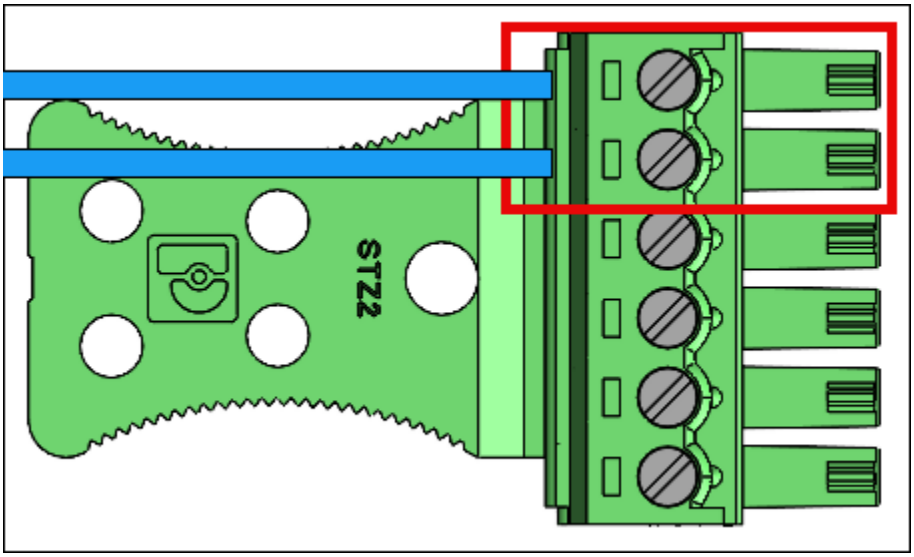
| Pin | Connection | Description | Use |
|---|---|---|---|
| 7 | COM | Relay common | Contact relay Common – White wire |
| 8 | NC | Relay normally closed | Contact relay normally closed – Orange wire |
| 9 | NO | Relay normally open | Contact relay normally open – Yellow wire |




The relay should be operated in accordance to the specified safety ratings 30VAC/60VDC, 60W Max.

### Digital input/output connections

- Insert the blue wire in Pin 1 (GPO).
- Insert the blue wire in Pin 2 (GPI).




### Digital input/output wiring

| Pin | Connection | Description | Use |
|---|---|---|---|
| 1 | GPO | General purpose output | Digital output signal (5V) |
| 2 | GPI | General purpose input | Digital input signal (3.6V – 5V) |

- The digital input/output connections should be operated as listed.

After installing your Amazon One device, you are ready to activate the device.

# Installing Amazon One device I/O Hub for secure access

The Amazon One device with I/O Hub is an integral part of the Amazon One Enterprise system, designed to enhance security and streamline access control for a variety of environments. The device leverages biometric palm recognition to provide secure, touchless authentication for users, making it ideal for use in high-security areas such as office buildings, restricted entry points, or facilities requiring seamless access management. The I/O Hub acts as a bridge between the device and your existing security infrastructure, enabling communication with door locks, alarms, and other access control systems.

This section provides the location requirements and step-by-step instructions for installing the Amazon One device with I/O Hub. Proper preparation and installation are key to ensuring the system operates securely and efficiently, providing users with a smooth, reliable experience.

**Prerequisites and preparation for installing the Amazon One Device with I/O Hub**

Before starting the installation, ensure the following conditions are met to ensure a safe, secure, and effective setup:

- Indoor use only: The Amazon One device with I/O Hub is designed for indoor use only. Ensure it is installed in an appropriate environment.
- Power Over Ethernet (PoE++): If using Power Over Ethernet (PoE++), verify that an IEEE 802.3bt (Type 3) Class 6 PoE++ switch (end span) or injector (midspan) is available. The PoE++ source

must be listed or certified and comply with IEC 62368-1 standards. Importantly, the PoE++ source must be located within the same building as the device. Only use an approved PoE++ source with the AOE device.

- 15V DC power input: If you are using 15V DC power input, ensure that only an NEC Class 2 or power-limited, approved power supply is used. The power supply must be listed or certified for safety. For further details, refer to the Optional DC section below.

**Required tools**

- Wire stripper
- #2 Phillips screwdriver
- 0.5mm x 2mm flathead screwdriver

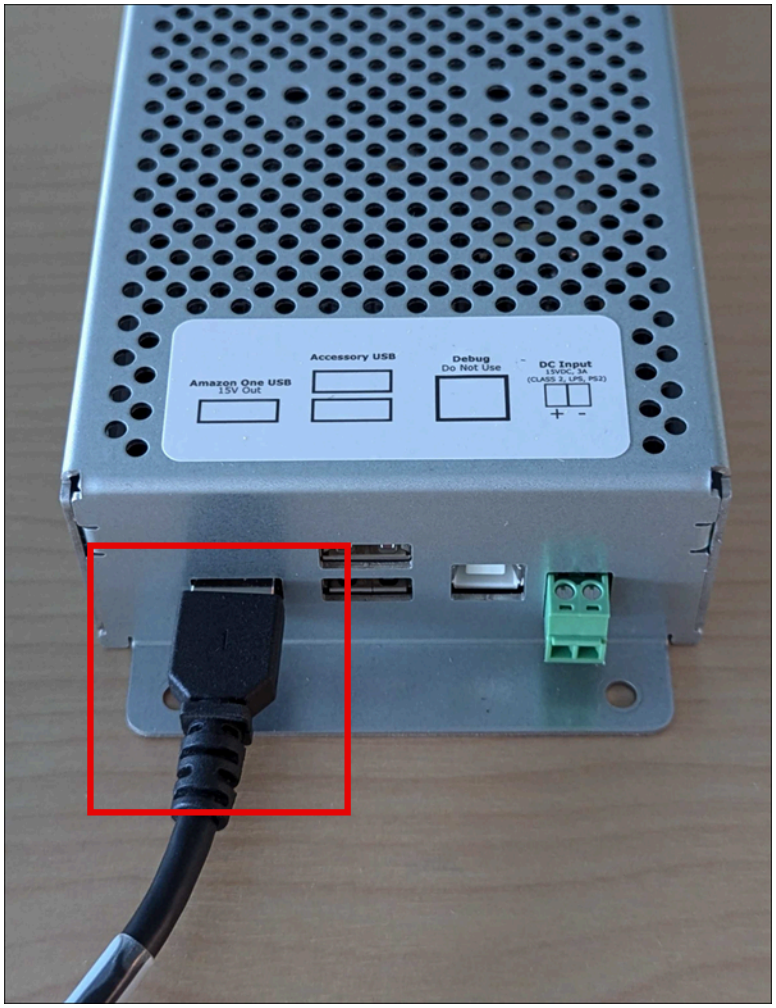
**Included with the Amazon One device with I/O Hub**

- 2x 6 position terminal block connectors
- DC plug connector
- 72" power/data cable

Once these prerequisites are confirmed, you can proceed with the installation process, ensuring a secure and efficient setup of your Amazon One device with I/O Hub. Proper preparation will help guarantee the device functions as intended and integrates smoothly into your secure access system.

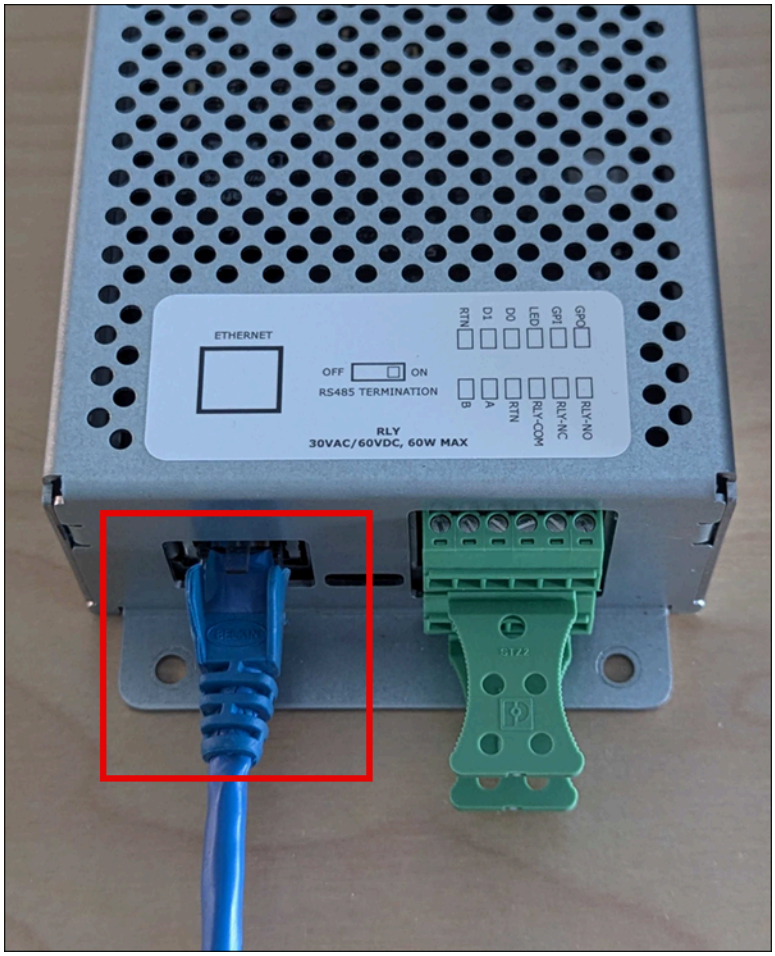
**To install the I/O hub for your Amazon One device**

1. Remove your Amazon One device with I/O Hub from the packaging.
2. Secure the I/O hub in the desired location.
3. Plug in the Amazon One USB cable into the I/O hub port.



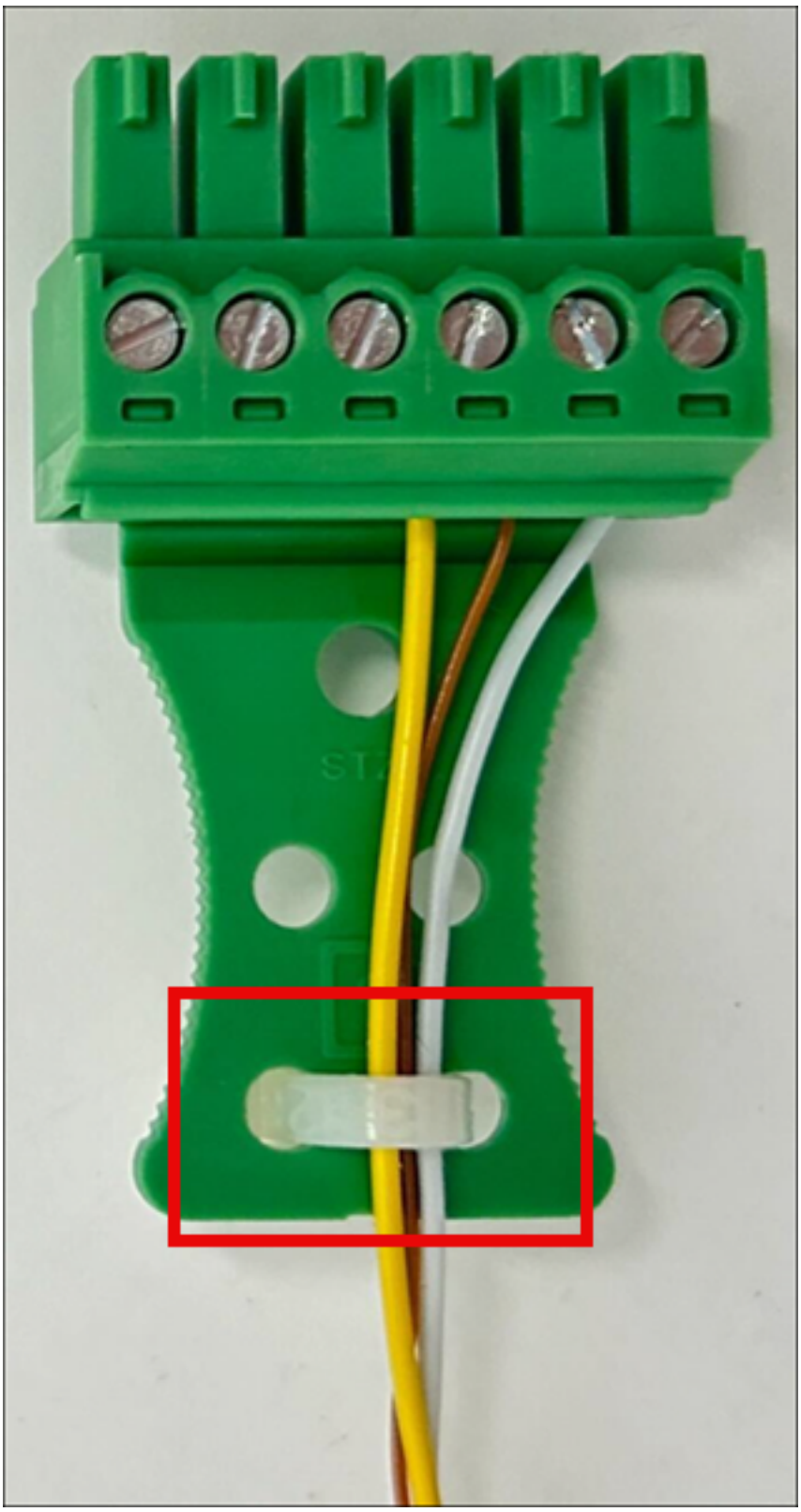

4. For POE++ power, plug in the Ethernet cable from the POE++ source into the I/O hub port.

   Optional: For DC power, refer to the install DC wiring section below.




**To wire the I/O hub for your Amazon One device**

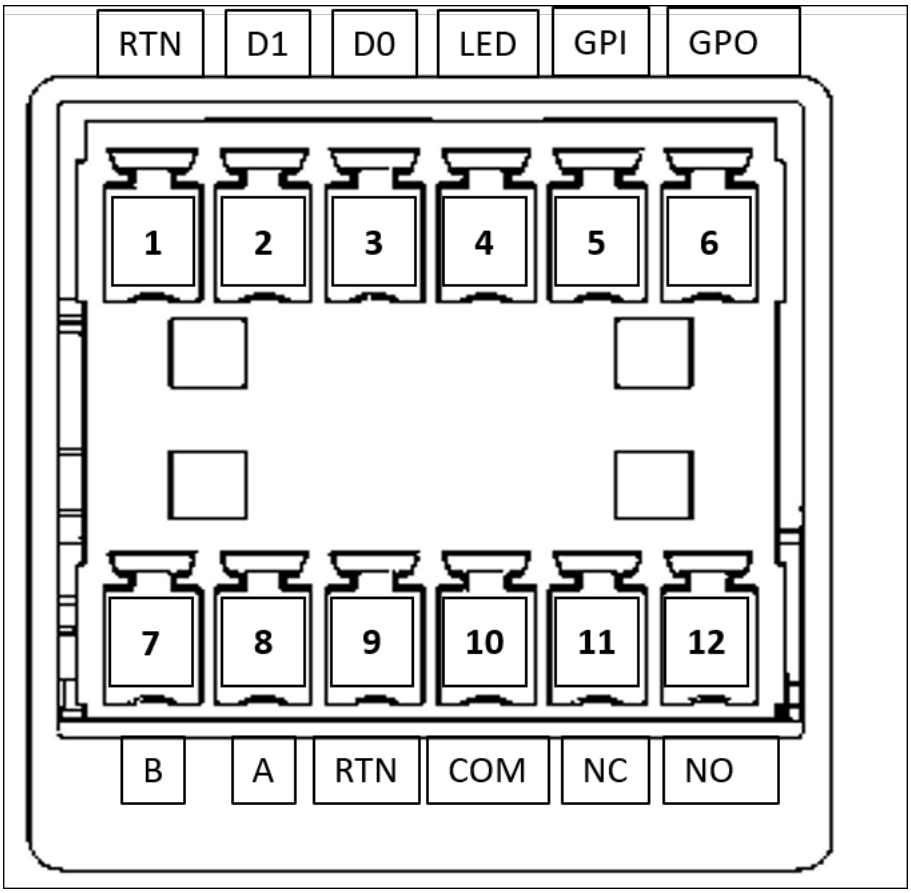
- Install a drip loop to avoid liquids accidently running down the cord and into the I/O hub.
- Attach a strain relief clamp to protect the wires from damage or stress, as shown in the following image.



1. Insert the terminal block plugs into the I/O hub.

2. Insert only the required wires for your application through the terminal block plugs. Refer to the following wiring table and diagrams.

### Connections

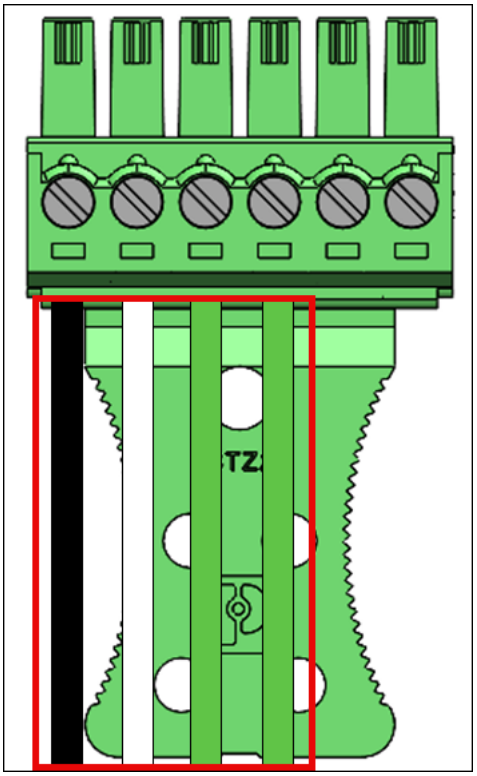



| Pin | Connection | Description | Use |
|---|---|---|---|
| 1 | RTN | Signal return | Wiegand ground – Black wire |
| 2 | D1 | Wiegand D1 | Wiegand Data 1 – White wire |
| 3 | D0 | Wiegand D0 | Wiegand data 0 – Green wire |
| 4 | LED | Wiegand LED | Wiegand LED – Optional |

| Pin | Connection | Description | Use |
|---|---|---|---|
| 5 | GPI | General purpose input | Digital input signal – Optional |
| 6 | GPO | General purpose output | Digital output signal - Optional |
| 7 | B | RS485_B/D0/ Data | OSDP D0 – Green wire |
| 8 | A | RS485_A/D1/ Clock | OSDP D1 – White wire |
| 9 | RTN | Signal return | OSDP return – Black wire |
| 10 | COM | Relay Common | Contact relay common – White wire |
| 11 | NC | Relay normally closed | Contact relay normally closed – Orange wire |
| 12 | NO | Relay Normally Open | Contact relay normally open – Yellow wire |

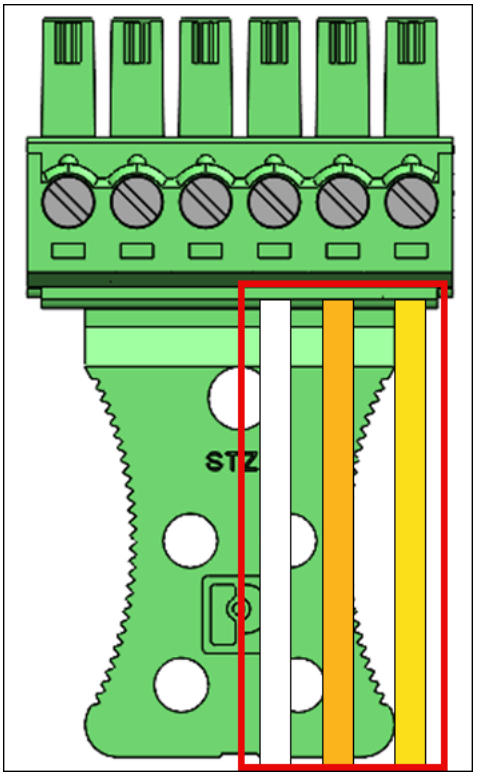
### Wiegand connections

- Insert the black wire in Pin 1 (RTN).
- Insert the white wire in Pin 2 (D1).
- Insert the green wire in Pin 3 (D0).
- Optional: Insert the green wire in Pin 4 (LED).

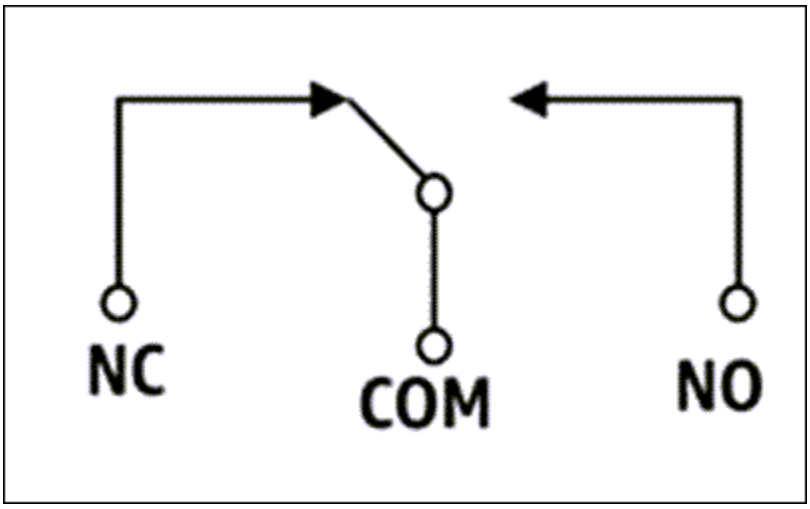


### Relay connections

- Insert the white wire in Pin 10 (COM).
- Insert the orange wire in Pin 11 (NC).
- Insert the yellow wire in Pin 12 (NO).




### Relay diagram

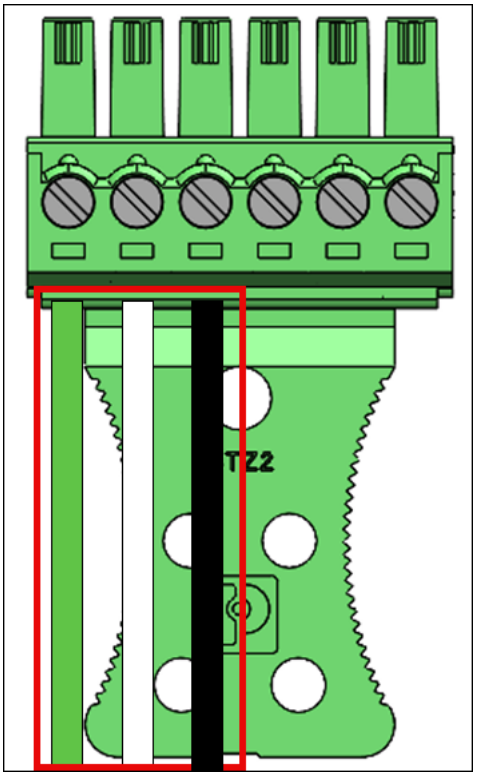



The relay should be operated in accordance to the specified safety ratings 30VAC/60VDC, 60W Max.

### RS485 connections

- Insert the green wire in Pin 7 (B).
- Insert the white wire in Pin 8 (A).

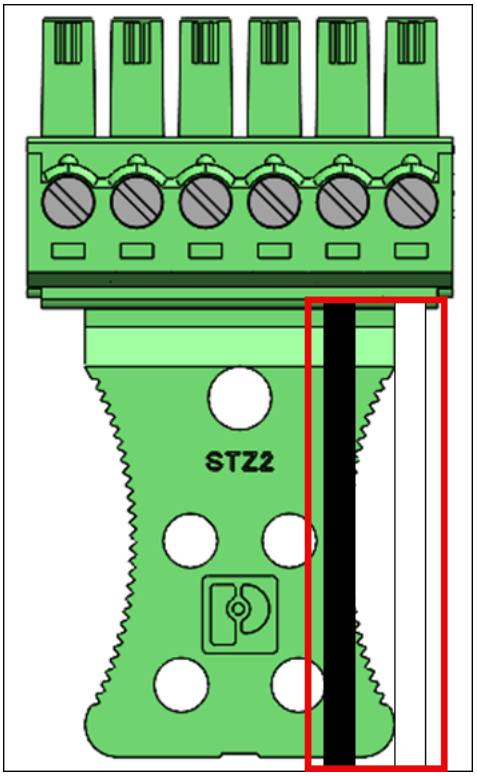
- Insert the black wire in Pin 9 (RTN).




Turn RS485 termination switch “ON” if the device is the last unit on the line. This switch activates 120 Ohms resistor termination on the line.

Digital input/output connections

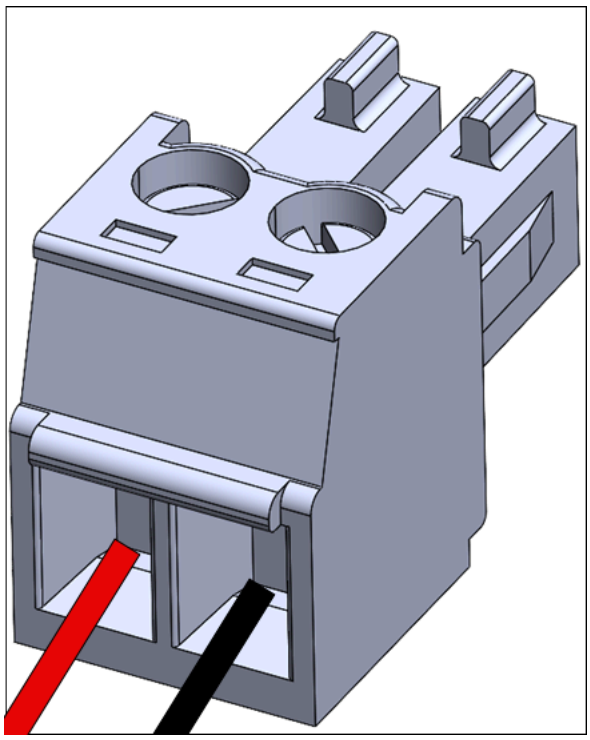
- Insert the black wire in Pin 5 (GPI).
- Insert the white wire in Pin 6 (GPO).



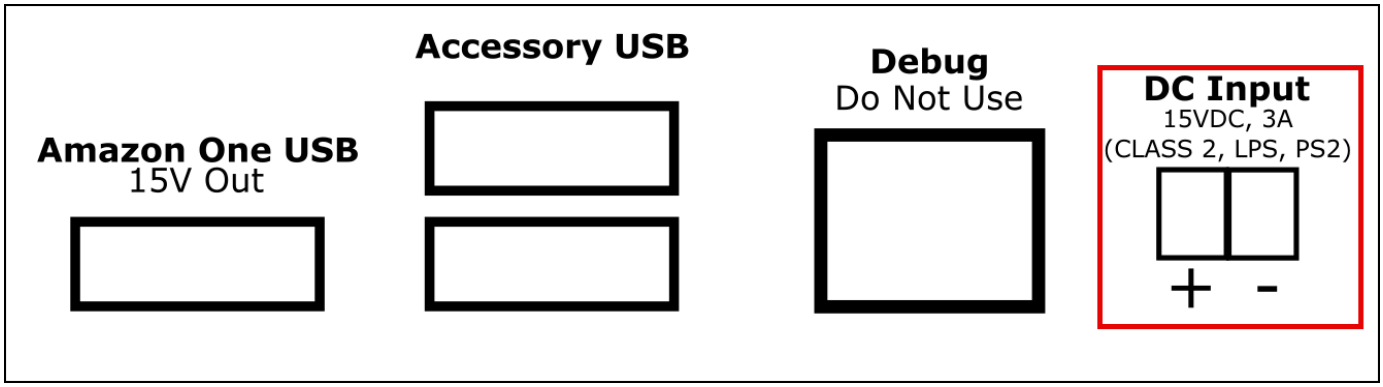

- The digital input/output connections should be operated as listed.

**Optional: To install DC wiring**

1. Strip off 3mm-5mm from the end of a red wire for positive (+) and a black wire for negative (-).
2. Insert the stripped end of the DC wire into the DC plug.



3. Screw the wire into position.
4. Insert the wired DC plug into the DC Input port.




After installing your Amazon One device, you are ready to activate the device.

# Activating Amazon One device

When your Amazon One device is installed and powered on, you are ready to activate it.

**To activate your Amazon One device**

1. On the Amazon One device, tap the screen to get started.

2. Choose **Ethernet** or **Wifi** to connect to the internet.

   As soon as the device is connected to the internet, it will start downloading the latest software package.
3. When the screen shows **Software download completed!**, select **OK**.
4. Select **QR code**.

   The Amazon One device screen will show **Scan QR code**.
5. To retrieve the activation QR code, open the Amazon One Enterprise console at https://console.aws.amazon.com/one-enterprise.

   > **Note**
   >
   > We highly recommend you grant limited permission to your installers so they only have access to activation QR codes in your Amazon One Enterprise console. See Add Amazon One users.
6. In the navigation pane, choose **Activation QR codes**.
7. From the **Select a site** drop-down list, select the Site where the Amazon One device is installed.
8. Under **Site information**, confirm the Site address.
9. Under **Activation QR codes**, look for the device instance name that you are activating, and select the corresponding **Get QR code** to retrieve the QR code.
10. Scan the QR code with the Amazon One device. Note that the QR code is refreshed periodically for security, you may only use a QR code once.
11. Enter the site zip code, and select **Confirm Settings** after verifying the correct site is shown.
12. When the Amazon One device screen shows **Activation complete!**, the device is ready for use.

# Enrolling and entering users

Now that your Amazon One device is activated, your employees can start enrolling their palms and authenticate their palms to get access.

**Topics**



## Creating an endpoint policy

Before users can authenticate their palms for entry, they will have to go through the enrollment process. Security personnel should always check the identity of the user before allowing the user to enroll.

**To enroll your palms on an Amazon One device**

1. On the Amazon One Enterprise enrollment device, press **Get started**.
2. Scan a employee badge with the badge scanner that's connected to your Amazon One Enterprise enrollment device.

   When the badge is successfully scanned, the Amazon One device screen shows **Badge scanned**.
3. Read through the **Terms of Use**, and then press **OK**.
4. Read through **Consent - Your Palm Biometric Information**, and press **I agree** if you consent.
5. Follow the on-screen instructions to complete the enrollment process.

## Authenticating for entry

After you have successfully enrolled your palms, you are ready to authenticate with your palm on your Amazon One Enterprise entry device.

**To authenticate your palm for entry on an Amazon One device**

- Hover your palm on top of the device and follow the on-screen instructions to scan your palm.

# Managing users

You can use the Enrolled user management page to keep track of enrolled users and to delete user biometrics. A user whose associated biometric is deleted will no longer have access to Amazon One devices for authentication.

**Topics**



# Viewing enrolled users

The following procedure details how to enroll users.

**To view enrolled users**

1. Open the Amazon One Enterprise console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **Enrolled user management**.
3. Under **Enrolled users**, you'll find all of the enrolled users and the following details:
   - **Badge ID** — Badge identifier information captured by an RFID badge reader at the time of enrollment.
   - **Enrollment source** — Details of the Amazon One device that was used for enrollment.
   - **Enrollment date** — Date and time of enrollment.

# Deleting enrolled users and their biometrics

The following procedure details how to delete enrolled users and their biometrics.

**To delete enrolled users and their biometrics**

1. Open the Amazon One Enterprise console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **Enrolled user management**.
3. Under **Enrolled users**, select the badge ID of the user whose palm biometric data you want to delete.

4. Choose **Delete Biometric**.
5. Choose **Delete** to confirm deletion of the user biometric data.

> **Important**
>
> This action results in the permanent deletion of a user's palm biometric from Amazon One Enterprise. The user will need to enroll again with an Amazon One Enterprise enrollment device to be able to use Amazon One Enterprise for authentication. Deleting a user's biometric will also permanently delete other profile attributes like badge ID from Amazon One Enterprise.

# Managing Amazon One devices

After your Amazon One device is installed and activated, it starts reporting device health on the Amazon One Enterprise console. You can use the Amazon One Enterprise console to perform device management tasks such as rebooting devices or updating configurations.

**Topics**



# Maintaining and cleaning Amazon One devices

Maintaining your Amazon One device provides the optimal device operating environment and device experience.

Before cleaning the Amazon One device, ensure the following:

- While you don't have to enable or disable Amazon One, ensure that the devices are connected to power, have network connectivity, and any peripheral and companion devices (if applicable) are connected.
- Escalate issues to your administrator if network connectivity is unavailable (an error screen will be visible on the Amazon One device if this occurs), an error screen will be visible on the Amazon One device or a device connection issue would be visible on the console.
- Physically secure devices so that unauthorized individuals cannot tamper with them.
- Visually inspect Amazon One devices daily, checking for any unauthorized connections to Amazon One device.
- Inspect all sides of the device looking for any signs of tampering, including visible screws of the device and the casing to ensure there are no gaps/openings exposing the internal components/ circuitry of either the Amazon One device.
- In case of any errors or failures, follow instructions on the Amazon One device screen or refer to troubleshooting guide to remediate issues.

## To clean the Amazon One device

Cleaning your Amazon One device regularly removes any smudges or marks such as fingerprints and handprints.

> **Note**
>
> Do not use any other cleaning products outside of those listed in this guide. The recommended cleaning schedule is once or twice per week, or whenever dirt, dust, or smudges are visible on device, but never more than once per day.

1. Wipe the Amazon One device with Isopropyl Alcohol (IPA) Wipes. Only clean the touch surface of the device. Don't touch the optical window, or use any other cleaning product unless instructed to do so by Amazon One.
2. Wipe away any streaks with a dry microfiber cloth.
3. Lightly Dust (don't wipe) any visible dirt or debris from the optical window. Limit the cleaning of the optical window to no more than once per day and/or when the window is visually dirty (e.g., finger/hand prints/smudges). This part of the device is not intended to be touched, but there could be inadvertent touching from new customers.
4. Use a KIC Smart card cleaner to clean the inside of a card reader, if applicable.
5. Clean the device once or twice per week, or whenever dirt, dust, or smudges are visible on device.

# Site Management

A site represents a physical location where a collection of device instances are installed and operating at. You can use sites to organize Amazon One devices that share the same physical address.

**Topics**

## Changing site name

The following procedure details how to change the site name for your device.

**To change the site name**

1. Open the Amazon One Enterprise console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **Site**.
3. Under **Sites**, select the site you intend to edit the name for.
4. Choose **Edit**.
5. Under **Site information** enter the desired site name and site description (optional).
6. Choose **Save changes** to update.

## Updating site address

The following procedure details how to update the site address for your device.

**To update the site address**

1. Open the Amazon One Enterprise console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **Site**.
3. Under **Sites**, select the site you intend to update the address for.
4. Under **Device instances**, ensure number of activated instances is 0.
5. (Optional) If number of activated instances is not 0, see
6. Choose **Edit**.
7. Under **Physical address** enter the correct physical address.
8. Choose **Save changes** to update.

# Device Instance Management

A device instance is a logical representation of a device with configurations. Use of device instances allows for swapping Amazon One devices while automatically inheriting the previously set configurations and names. A device instance has a user-defined name (shared naming convention with your access control software) and a set of communication configurations.

**Topics**



## Viewing device instance status

The following procedure details how to view the status of your device instance.

**To view device instance status**

1. Open the Amazon One Enterprise console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **Device instance**.
3. Under **Activated instances**, you'll see a list of activated Amazon One devices.
4. Choose a device instance name to view device instance details.

## Rebooting an Amazon One device

The following procedure details how to reboot your Amazon One device.

**To reboot an Amazon One device**

1. Open the Amazon One Enterprise console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **Device instance**.
3. Under **Activated instances**, choose the instance name of the device that you want to reboot.
4. Choose **Reboot** to restart the Amazon One device.

## Updating Amazon One device configurations

The following procedure details how to update Amazon One device configurations.

**To Update Amazon One device configurations**

1. Open the Amazon One Enterprise console at https://console.aws.amazon.com/one-enterprise.

2. In the navigation pane, choose **Device instance**.
3. Under **Activated instances**, choose the instance name of the device that you want to update.
4. Under **Device configurations**, choose **Edit**.

> **Note**
>
> To change the Amazon One device mode, you must first deactivate the device instance, and then configure it with the desired device mode (see Configure a device instance for activation). Then, you can go through the device activation process (see Activating Amazon One device).

5. After you have made the desired changes, choose **Update device configurations** to confirm the update.

## Updating Wi-fi credentials

The following procedure details how to update Wi-Fi credentials.

**To update Wifi credentials**

1. Open the Amazon One Enterprise console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **Device instance**.
3. Under **Activated instances**, choose the instance name of the device that you want to update.
4. Under **Network**, choose **Edit**.
5. Under **Wi-Fi configurations**, make the desired changes.
6. Choose **Update network** to confirm the update.

## Deactivating device instances

The following procedure details how to deactivate device instances.

**To deactivate device instances**

1. Open the Amazon One Enterprise console at https://console.aws.amazon.com/one-enterprise.
2. In the navigation pane, choose **Device instance**.
3. Under **Activated instances**, select the name of the device instance you want to deactivate.

4. Choose **Deactivate device**.
5. To confirm deactivation, type 'deactivate' in the message box and choose **Deactivate device**.

# Security

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

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

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon One Enterprise, 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 Amazon One Enterprise. The following topics show you how to configure Amazon One Enterprise to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon One Enterprise resources.

**Topics**



# Data protection in Amazon One Enterprise

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

for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

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

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

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

## To use the default encryption of data at rest

Amazon One Enterprise provides encryption by default to protect sensitive data at rest using AWS encryption keys.

**AWS owned keys** — Amazon One Enterprise uses these keys by default to automatically encrypt sensitive end user data. You can't view, manage, or use AWS owned keys, or audit their use.

However, you don't have to take any action or change any programs to protect the keys that encrypt your data. For more information, see AWS owned keys in the AWS Key Management Service Developer Guide.

## Encryption of data in transit

Amazon One Enterprise uses Transport Layer Security (TLS) to secure data and Signature Version 4 to authenticate all inbound API requests to AWS services. This encryption is enabled by default.

# Identity and access management for Amazon One Enterprise

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 Amazon One Enterprise 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 Amazon One Enterprise.

**Service user** – If you use the Amazon One Enterprise service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Amazon One Enterprise 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 Amazon One Enterprise, see Troubleshooting Amazon One identity and access.

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

**IAM administrator** – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Amazon One Enterprise. To view example Amazon One Enterprise identity-based policies that you can use in IAM, see Identity-based policy examples for Amazon One Enterprise.

## Authenticating with identities

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

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

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

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

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in

the *AWS IAM Identity Center User Guide* and AWS Multi-factor authentication in IAM in the *IAM User Guide*.

### AWS account root user

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

### Federated identity

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

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

### IAM users and groups

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

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier

to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see Use cases for IAM users 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. To temporarily assume an IAM role in the AWS Management Console, you can switch from a user to an IAM role (console). 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 Methods to assume a role in the *IAM User Guide*.

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

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Create a role for a third-party identity provider (federation) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.

- **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
- **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.
- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Use an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

## Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

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

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

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 Define custom IAM permissions with customer managed 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 Choose between managed policies and inline policies in the *IAM User Guide*.

## Resource-based policies

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

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

## Access control lists (ACLs)

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

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

## Other policy types

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

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – RCPs are JSON policies that you can use to set the maximum available permissions for resources in your accounts without updating the IAM policies attached to each resource that you own. The RCP limits permissions for resources in member accounts and can impact the effective permissions for identities, including the AWS account root user, regardless of whether they belong to your organization. For more information about Organizations and RCPs, including a list of AWS services that support RCPs, see Resource control policies (RCPs) 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 Amazon One Enterprise works with IAM

Before you use IAM to manage access to Amazon One Enterprise, learn what IAM features are available to use with Amazon One Enterprise.

**IAM features you can use with Amazon One Enterprise**

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

To get a high-level view of how Amazon One Enterprise and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

## Identity-based policies for Amazon One Enterprise

**Supports identity-based policies:** Yes

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

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

### Identity-based policy examples for Amazon One Enterprise

To view examples of Amazon One Enterprise identity-based policies, see Identity-based policy examples for Amazon One Enterprise.

## Resource-based policies within Amazon One Enterprise

**Supports resource-based policies:** No

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

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant

the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

## Policy actions for Amazon One Enterprise

**Supports policy actions:** Yes

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

The `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.

To see a list of Amazon One Enterprise actions, see Actions, resources, and condition keys for Amazon One Enterprise.

Policy actions in Amazon One Enterprise use the following prefix before the action:

```
one
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "one:action1",
      "one:action2"
         ]
```

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

```
"Action": "one:Describe*"
```

To view examples of Amazon One Enterprise identity-based policies, see Identity-based policy examples for Amazon One Enterprise.

## Policy resources for Amazon One Enterprise

**Supports policy resources:** Yes

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

The `Resource` JSON policy element specifies the object or objects to which the action applies. 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": "*"
```

To see a list of Amazon One Enterprise resource types and their ARNs, and to learn which actions you can use to specify the ARN of each resource, see Actions, resources, and condition keys for Amazon One Enterprise.

To view examples of Amazon One Enterprise identity-based policies, see Identity-based policy examples for Amazon One Enterprise.

## Policy condition keys for Amazon One Enterprise

**Supports service-specific policy condition keys:** Yes

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

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

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

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

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

To see a list of Amazon One Enterprise condition keys and to learn which actions and resources you can use a condition key with, see Actions, resources, and condition keys for Amazon One Enterprise.

To view examples of Amazon One Enterprise identity-based policies, see Identity-based policy examples for Amazon One Enterprise.

## ACLs in Amazon One Enterprise

**Supports ACLs:** No

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.

## ABAC with Amazon One Enterprise

**Supports ABAC (tags in policies):** Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

## Using temporary credentials with Amazon One Enterprise

**Supports temporary credentials:** Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switch from a user to an IAM role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

## Cross-service principal permissions for Amazon One Enterprise

**Supports forward access sessions (FAS):** Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

### Service roles for Amazon One Enterprise

**Supports service roles:** No

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

> **Warning**
>
> Changing the permissions for a service role might break Amazon One Enterprise functionality. Edit service roles only when Amazon One Enterprise provides guidance to do so.

### Service-linked roles for Amazon One Enterprise

**Supports service-linked roles:** No

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

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a `Yes` in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

## Identity-based policy examples for Amazon One Enterprise

By default, users and roles don't have permission to create or modify Amazon One Enterprise resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by Amazon One Enterprise, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for Amazon One Enterprise in the *Service Authorization Reference*.

**Topics**



## Policy best practices

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

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

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

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

## Using the Amazon One Enterprise console

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

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

To ensure that users and roles can still use the Amazon One Enterprise console, also attach the Amazon One Enterprise `ConsoleAccess` or `ReadOnly` AWS managed policy to the entities. 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": "*"
        }
    ]
}
```

## Read-only access to Amazon One Enterprise

The following example shows an AWS managed policy, `AmazonOneEnterpriseReadOnlyAccess` that grants read-only access to Amazon One Enterprise.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "one:Get*",
        "one:List*"
```

```
      ],
      "Resource": "*"
    }
  ]
}
```

In the policy statements, the `Effect` element specifies whether the actions are allowed or denied. The `Action` element lists the specific actions that the user is allowed to perform. The `Resource` element lists the AWS resources the user is allowed to perform those actions on. For policies that control access to Amazon One Enterprise actions, the `Resource` element is always set to `*`, a wildcard that means "all resources."

The values in the `Action` element correspond to the APIs that the services support. The actions are preceded by `config:` to indicate that they refer to Amazon One Enterprise actions. You can use the `*` wildcard character in the `Action` element, such as in the following examples:

- `"Action": ["one:*DeviceInstanceConfiguration"]`

  This allows all Amazon One Enterprise actions that end with "DeviceInstance" (`GetDeviceInstanceConfiguration`, `CreateDeviceInstanceConfiguration`).
- `"Action": ["one:*"]`

  This allows all Amazon One Enterprise actions, but not actions for other AWS services.
- `"Action": ["*"]`

  This allows all AWS actions. This permission is suitable for a user who acts as an AWS administrator for your account.

The read-only policy doesn't grant user permission for actions such as `CreateDeviceInstance`, `UpdateDeviceInstance`, and `DeleteDeviceInstance`. Users with this policy are not allowed to create a device instance, update a device instance, or delete a device instance. For the list of Amazon One Enterprise actions, see Actions, resources, and condition keys for Amazon One Enterprise.

## Full access to Amazon One Enterprise

The following example shows a policy that grants full access to Amazon One Enterprise. It grants users the permission to perform all Amazon One Enterprise actions.

**Important**

This policy grants broad permissions. Before granting full access, consider starting with a minimum set of permissions and granting additional permissions as necessary. Doing so is better practice than starting with permissions that are too lenient and then trying to tighten them later.

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

## Supported Resource-Level Permissions for Amazon One Enterprise Rule API Actions

Resource-level permissions refers to the ability to specify which resources users are allowed to perform actions on. Amazon One Enterprise supports resource-level permissions for certain Amazon One Enterprise rule API actions. This means that for certain Amazon One Enterprise rule actions, you can control the conditions under which when users are allowed to use those actions. These conditions can be actions that must be fulfilled, or specific resources that users are allowed to use.

The following table describes the Amazon One Enterprise rule API actions that currently support resource-level permissions. It also describes the supported resources and their ARNs for each action. When specifying an ARN, you can use the * wildcard in your paths; for example, when you cannot or do not want to specify exact resource IDs.

**Important**

If an Amazon One Enterprise rule API action is not listed in this table, then it does not support resource-level permissions. If an Amazon One Enterprise rule action does not support resource-level permissions, you can grant users permissions to use the action, but you have to specify a * for the resource element of your policy statement.

| API Action | Resources |
| --- | --- |
| CreateDeviceInstance | Device Instance<br><br>arn:aws:one:*region:accountID* :device-instance/*deviceInstanceId* |
| GetDeviceInstance | Device Instance<br><br>arn:aws:one:*region:accountID* :device-instance/*deviceInstanceId* |
| UpdateDeviceInstance | Device Instance<br><br>arn:aws:one:*region:accountID* :device-instance/*deviceInstanceId* |
| DeleteDeviceInstance | Device Instance<br><br>arn:aws:one:*region:accountID* :device-instance/*deviceInstanceId* |
| CreateDeviceActivationQrCode | Device Instance<br><br>arn:aws:one:*region:accountID* :device-instance/*deviceInstanceId* |
| DeleteAssociatedDevice | Device Instance<br><br>arn:aws:one:*region:accountID* :device-instance/*deviceInstanceId* |

| API Action | Resources |
| --- | --- |
| RebootDevice | Device Instance<br><br>arn:aws:one:*region:accountID* :device-instance/*deviceInstanceId* |
| CreateDeviceInstanceConfiguration | Device Instance Configuration<br><br>arn:aws:one:*region:accountID* :device-instance/*deviceInstanceId* /configuration/*version* |
| GetDeviceInstanceConfiguration | Device Instance Configuration<br><br>arn:aws:one:*region:accountID* :device-instance/*deviceInstanceId* /configuration/*version* |
| CreateSite | Site<br><br>arn:aws:one:*region:accountID* :site/*siteId* |
| DeleteSite | Site<br><br>arn:aws:one:*region:accountID* :site/*siteId* |
| GetSiteAddress | Site<br><br>arn:aws:one:*region:accountID* :site/*siteId* |
| UpdateSite | Site<br><br>arn:aws:one:*region:accountID* :site/*siteId* |
| UpdateSiteAddress | Site<br><br>arn:aws:one:*region:accountID* :site/*siteId* |
| CreateDeviceConfigurationTemplate | Device Configuration Template<br><br>arn:aws:one:*region:accountID* :device-configuration-template/*templateId* |

| API Action | Resources |
| --- | --- |
| DeleteDeviceConfigurationTemplate | Device Configuration Template<br><br>arn:aws:one:*region:accountID* :device-configuration-template/*templateId* |
| GetDeviceConfigurationTemplate | Device Configuration Template<br><br>arn:aws:one:*region:accountID* :device-configuration-template/*templateId* |
| UpdateDeviceConfigurationTemplate | Device Configuration Template<br><br>arn:aws:one:*region:accountID* :device-configuration-template/*templateId* |

For example, you want to allow read access and deny write access to specific rules to specific users.

In the first policy, you allow the AWS Config rule read actions such as `GetSite` on the specified rules.

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "VisualEditor0",
                "Effect": "Allow",
                "Action": [
                    "one:GetSite",
                    "one:GetSiteAddress"
                ],
                "Resource": [
                    "arn:aws:one:region:accountID:site/siteId"
                ]
            }
        ]
    }
```

In the second policy, you deny the Amazon One Enterprise rule write actions on the specific rule.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Deny",
            "Action": [
                "one:DeleteSite",
                "one:UpdateSiteAddress"
               ],
            "Resource": "arn:aws:one:region:accountID:site/siteId"
           }
      ]
   }
```

With resource-level permissions, you can allow read access and deny write access to perform specific actions on Amazon One Enterprise rule API actions.

### Additional Information

To learn more about creating IAM users, groups, policies, and permissions, see Creating Your First IAM User and Administrators Group and Access Management in the *IAM User Guide*.

## AWS managed policies for Amazon One Enterprise

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users,

groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

## AmazonOneEnterpriseFullAccess

This policy grants administrative permissions that allow access to all Amazon One Enterprise resources and operations.

one:* Lets you perform all Amazon One Enterprise actions.

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

## AmazonOneEnterpriseReadOnlyAccess

This policy grants read only permissions to all Amazon One Enterprise resources and operations.

one:Get* Gets the Amazon One Enterprise resources.

one:List* Lists the Amazon One Enterprise resources.

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

## AmazonOneEnterpriseInstallerAccess

This policy grants limited read and write permissions that allow you to create activation QR code for any configured device instance to activate device at any site.

one:CreateDeviceActivationQrCode Let you create QR code to activate device.

one:GetDeviceInstance Let you fetch the information about an Amazon One device instance.

one:GetSite Let you fetch the information about an Amazon One Enterprise site.

one:GetSiteAddress Let you fetch the physical address of an Amazon One Enterprise site.

one:ListDeviceInstances Let you list the Amazon One device instances.

one:ListSites Let you list the Amazon One Enterprise sites.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "InstallerAccessStatementID",
   "Effect": "Allow",
   "Action": [
    "one:CreateDeviceActivationQrCode",
    "one:GetDeviceInstance",
    "one:GetSite",
```

```
    "one:GetSiteAddress",
    "one:ListDeviceInstances",
    "one:ListSites"
   ],
   "Resource": "*"
  }
 ]
}
```

### Amazon One Enterprise updates to AWS managed policies

View details about updates to AWS managed policies for Amazon One Enterprise that have been made since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Amazon One Enterprise Document history page.

| Change | Description | Date |
| --- | --- | --- |
| Amazon One Enterprise added AmazonOneMetricPublishAccess | The role permissions policy named AmazonOneMetricPublishAccess allows Amazon One Enterprise to perform CloudWatch:PutMetricData on CloudWatch Namespace AWS/AmazonOne. | February 6, 2025 |
| Amazon One Enterprise started tracking changes | Amazon One Enterprise started tracking changes for its AWS managed policies. | December 1, 2023 |

## Actions, resources, and condition keys for Amazon One Enterprise

Amazon One Enterprise (service prefix: one) provides the following service-specific resources, actions, and condition context keys for use in IAM permission policies.

**Topics**

- Actions defined by Amazon One Enterprise
- Resource types defined by Amazon One Enterprise
- Condition keys for Amazon One Enterprise

## Actions defined by Amazon One Enterprise

You can specify the following actions in the `Action` element of an IAM policy statement. Use policies to grant permissions to perform an operation in AWS. When you use an action in a policy, you usually allow or deny access to the API operation or CLI command with the same name. However, in some cases, a single action controls access to more than one operation. Alternatively, some operations require several different actions.

The **Resource types** column of the Actions table indicates whether each action supports resource-level permissions. If there is no value for this column, you must specify all resources ("*") to which the policy applies in the `Resource` element of your policy statement. If the column includes a resource type, then you can specify an ARN of that type in a statement with that action. If the action has one or more required resources, the caller must have permission to use the action with those resources. Required resources are indicated in the table with an asterisk (*). If you limit resource access with the `Resource` element in an IAM policy, you must include an ARN or pattern for each required resource type. Some actions support multiple resource types. If the resource type is optional (not indicated as required), then you can choose to use one of the optional resource types.

The **Condition keys** column of the Actions table includes keys that you can specify in a policy statement's `Condition` element. For more information on the condition keys that are associated with resources for the service, see the **Condition keys** column of the Resource types table.

> **Note**
>
> Resource condition keys are listed in the Resource types table. You can find a link to the resource type that applies to an action in the **Resource types (*required)** column of the Actions table. The resource type in the Resource types table includes the **Condition keys** column, which are the resource condition keys that apply to an action in the Actions table.

For details about the columns in the following table, see Actions table.

| Actions | Description | Access level | Resource types (*required) | Condition keys | Dependent actions |
|---|---|---|---|---|---|
| CreateDeviceInstance | Grant permission to create device instance | Write | | aws:RequestTag/${TagKey}<br>aws:TagKeys | |
| GetDeviceInstance | Grant permission to get information about device instance | Read | device-instance* | | |
| ListDeviceInstances | Grant permission to list device instances | Read | | | |
| UpdateDeviceInstance | Grant permission to update device instance | Write | device-instance* | | |
| DeleteDeviceInstance | Grant permission to delete device instance | Write | device-instance* | | |
| CreateDeviceActivationQrCode | Grant permission to create QR code to activate a device at a device instance | Write | device-instance* | | |
| DeleteAssociatedDevice | Grant permission to delete association between device and device-instance | Write | device-instance* | | |
| RebootDevice | Grant permission to reboot device | Write | device-instance* | | |
| CreateDeviceInstan | Grant permission to create device instance configuration | Write | | | |

| Actions | Description | Access level | Resource types (*required) | Condition keys | Dependent actions |
|---|---|---|---|---|---|
| ceConfiguration | | | | | |
| GetDeviceInstanceConfiguration | Grant permission to get information about device instance configuration | Read | configuration* | | |
| CreateSite | Grant permission to create site | Write | | aws:RequestTag/${TagKey}<br>aws:TagKeys | |
| DeleteSite | Grant permission to delete device instance | Write | sites* | | |
| GetSite | Grant permission to get information about site | Read | sites* | | |
| ListSites | Grant permission to list sites | Read | | | |
| GetSiteAddress | Grant permission to get information about site address | Read | sites* | | |
| UpdateSite | Grant permission to update site | Write | sites* | | |
| UpdateSiteAddress | Grant permission to update site address | Write | sites* | | |

| Actions | Description | Access level | Resource types (*required) | Condition keys | Dependent actions |
|---|---|---|---|---|---|
| CreateDeviceConfigurationTemplate | Grant permission to create device instance | Write | | aws:RequestTag/${TagKey}<br>aws:TagKeys | |
| DeleteDeviceConfigurationTemplate | Grant permission to delete device configuration template | Write | device-configuration-template* | | |
| GetDeviceConfigurationTemplate | Grant permission to get information about device configuration template | Read | device-configuration-template* | | |
| ListDeviceConfigurationTemplates | Grant permission to list device configuration templates | Read | | | |
| UpdateDeviceConfigurationTemplate | Grant permission to update device configuration template | Write | device-configuration-template* | | |

| Actions | Description | Access level | Resource types (*required) | Condition keys | Dependent actions |
|---|---|---|---|---|---|
| TagResource | Grants permission to tag a resource | Tagging | device-instance, site, device-configuration-template | aws:RequestTag/${TagKey}<br>aws:TagKeys | |
| UntagResource | Grants permission to untag a resource | Tagging | device-instance, site, device-configuration-template | aws:TagKeys | |
| ListTagForResources | Grants permission to list tags for a resource | Read | | | |

## Resource types defined by Amazon One Enterprise

The following resource types are defined by this service and can be used in the `Resource` element of IAM permission policy statements. Each action in the Actions table identifies the resource types that can be specified with that action. A resource type can also define which condition keys you can include in a policy. These keys are displayed in the last column of the Resource types table. For details about the columns in the following table, see Resource types table.

| Resource types | ARN | Condition keys |
| --- | --- | --- |
| Device Instance | arn:aws:one: *region:accountID* :device-instance/ *deviceInstanceId* | aws:ResourceTag/${TagKey} |
| Device Instance Configuration | arn:aws:one: *region:accountID* :device-instance/ *deviceInstanceId* /configuration/ *version* | |
| Site | arn:aws:one: *region:accountID* :site/*siteId* | aws:ResourceTag/${TagKey} |
| Device Configuration Template | arn:aws:one: *region:accountID* :device-configuration-template/ *templateId* | aws:ResourceTag/${TagKey} |

## Condition keys for Amazon One Enterprise

Amazon One Enterprise defines the following condition keys that can be used in the `Condition` element of an IAM policy. You can use these keys to further refine the conditions under which the policy statement applies. For details about the columns in the following table, see Condition keys table.

To view the global condition keys that are available to all services, see Available global condition keys.

| Condition keys | Description | Type |
| --- | --- | --- |
| aws:RequestTag/${TagKey} | Filters access by tags from request | String |
| aws:ResourceTag/${TagKey} | Filters access by tags associated with the resource | String |

| Condition keys | Description | Type |
| --- | --- | --- |
| aws:TagKeys | Filters access by tag keys from request | ArrayOfString |

# Compliance validation for Amazon One Enterprise

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

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

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

- Security Compliance & Governance – These solution implementation guides discuss architectural considerations and provide steps for deploying security and compliance features.
- HIPAA Eligible Services Reference – Lists HIPAA eligible services. Not all AWS services are HIPAA eligible.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config 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. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.

- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

# Monitoring Amazon One Enterprise

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon One Enterprise and your other AWS solutions. AWS provides the following monitoring tools to watch Amazon One Enterprise, report when something is wrong, and take automatic actions when appropriate:

- *Amazon EventBridge* can be used to automate your AWS services and respond automatically to system events, such as application availability issues or resource changes. Events from AWS services are delivered to EventBridge in near real time. You can write simple rules to indicate which events are of interest to you and which automated actions to take when an event matches a rule. For more information, see the Amazon EventBridge User Guide.
- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

# Monitoring Amazon One Enterprise events in Amazon EventBridge

You can monitor Amazon One Enterprise events in EventBridge, which delivers a stream of real-time data from your own applications, software-as-a-service (SaaS) applications, and AWS services. EventBridge routes that data to targets such as AWS Lambda and Amazon Simple Notification Service. These events deliver a near real-time stream of system events that describe changes in AWS resources.

## Subscribe to Amazon One Enterprise events

Amazon One device and user profile status change events are published using EventBridge, and can be enabled in the EventBridge console by creating a new rule. Although events are not ordered, they have a timestamp which enables you to consume the data. Events are emitted on a best effort basis.

**To subscribe to Amazon One Enterprise events**

1. Log in to your AWS console at https://console.aws.amazon.com/events/.

2. Open the EventBridge console at https://console.aws.amazon.com/events/.
3. In the navigation pane, under **Buses**, choose **Rules**.
4. Choose **Create rule**.
5. On the **Default rule detail** page, assign a name to the rule.
6. Choose **Rule with an event pattern**, and then choose **Next**.
7. On the **Build event pattern** page, under **Event source**, verify that **AWS events or EventBridge partner events** is selected.
8. Under **Sample event type**, choose **AWS Events**.
9. For **Creation method**, choose **Custom pattern**.
10. In the **Event pattern** section, add a JSON with event source as `aws:one` and the required detail-type:

```
"
source": ["aws.one"],
"detail-type": ["New Successful Enrollment",
"New Successful Un-enrollment",
"Unsuccessful Enrollment",
"Unsuccessful Un-enrollment",
"Successful Recognition",
"Unsuccessful Recognition",
"New Alert(s) Detected",
"Some Alert(s) Cleared"]
}
```

    You can choose the required detail type from the above list and remove what is not required.
11. Choose **Next**.
12. On the **Select target(s)** page, select a target of your choice, which includes a Lambda function, SQS queue, or SNS topic. For information about configuring targets, see Amazon EventBridge targets.

    For example, to view when someone clocks-in, choose **“Successful Recognition”**. Then look at the event detail (given in Appendix) to see who clocked in.

    To complete your workflow, you can execute an external API or another target.
13. Optionally, you can configure tags.

14. On the **Review and create** page, choose **Create rule**. For more information about configuring rules, see EventBridge rules in the EventBridge User Guide.

## Device status change event types

Device status change events are generated in JSON. For each event type, a JSON blob is sent to the target of your choice, as configured in the rule. The following detail types are available:

**Some Alert(s) Cleared**

Device passed one or more health checks.

**New Alert(s) Detected**

Device failed one or more health checks.

**resources**

Contains the list of deviceInstance arn for which the Device Status Change event was published.

**data**

clearedAlerts

- Represents the health checks the deviceInstance was previously failing.
- Consists of a statusCode for the type of alert and a reportedAt timestamp.
- Possible statusCode values: NetworkDisconnected, USBDisconnected

currentAlerts

- Represents the current status of the deviceInstance.
- Consists of a statusCode for the type of alert and a reportedAt timestamp.
- Possible statusCode values: NetworkDisconnected, USBDisconnected

newAlerts

- Represents newly failed health checks of the deviceInstance.
- Consists of a statusCode for the type of alert and a reportedAt timestamp.
- Possible statusCode values: NetworkDisconnected, USBDisconnected

currentAlertsCount

- The count of health checks currently failing with the deviceInstance.

assetTagId

- The assetTagId of the device associated with the deviceInstance.

deviceInstanceName

- The name of the deviceInstance for which the Device Status Event was published.

siteName

- Name of the site where the deviceInstance is present.

siteArn

- Arn for the site where the deviceInstance is present.

## User profile event types

The User profile related event details types are:

**New Successful Enrollment**

When a user enrolled successfully.

**New Successful Un-enrollment**

When a user un-enrolled successfully.

**Unsuccessful Enrollment**

When a user failed to enroll.

**Unsuccessful Un-enrollment**

When a user failed to un-enroll.

**Successful Recognition**

When a user scans palm for authentication successfully.

**Unsuccessful Recognition**

When the recognition of a palm scan failed.

**resources**

Contains the list of user profile arn for which the user profile event was published.

**data**

accountId

- The relevant AWS account for the device that initiated the request.

requestSource

- This is the deviceInstanceId of the device that initiated the request.

createdTimestamp

- The time of event being created.

userStatus

- The current status of the user.
- Possible values: ACTIVE, DELETED

associatedId

- The associated id of the user, for example the badge id.

reason

- This value will present for unsuccessful events. It contains the reason why the event was unsuccessful.

## Sample events

The following examples show events for Amazon One Enterprise.

**Topics**



## Device health status changed to healthy

The device passed all the health checks.

```
{
        "version": "0",
        "id": "51e022b4-7ce6-34e0-264b-370948fc1123",
        "detail-type": "Some Alert(s) Cleared",
        "source": "aws.one",
        "account": "123456789012",
        "time": "2025-07-17T19:32:42Z",
        "region": "us-east-1",
        "resources":
        [
            "arn:aws:one:us-east-1:123456789012:deviceInstance/F5JRte5Jz21Tqx"
        ],
        "detail":
        {
            "version": "1.0.0",
            "data":
            {
                "clearedAlerts":
                [
                    {
                        "statusCode": "USBDisconnected",
                        "reportedAt": "Thu Jul 17 19:32:42 UTC 2025"
                    }
                ],
                "currentAlerts":
                [],
                "currentAlertsCount": 0,
                "assetTagId": "0000123456",
                "deviceInstanceName": "device_name",
                "siteName": "site_name",
                "siteArn": "arn:aws:one:us-east-1:123456789012:site/12345678901234"
            }
        }
      }
```


## Device health status changed to critical

The device failed one or more health checks.


```
{
        "version": "0",
        "id": "07af4893-ef9f-965a-d245-3f0c8bd3c123",
        "detail-type": "New Alert(s) Detected",
```

```
        "source": "aws.one",
        "account": "123456789012",
        "time": "2025-07-17T19:26:58Z",
        "region": "us-east-1",
        "resources":
        [
            "arn:aws:one:us-east-1:123456789012:deviceInstance/12345678901234"
        ],
        "detail":
        {
            "version": "1.0.0",
            "data":
            {
                "newAlerts":
                [
                    {
                        "statusCode": "USBDisconnected",
                        "reportedAt": "Thu Jul 17 19:26:58 UTC 2025"
                    }
                ],
                "currentAlerts":
                [
                    {
                        "statusCode": "USBDisconnected",
                        "reportedAt": "Thu Jul 17 19:26:58 UTC 2025"
                    }
                ],
                "currentAlertsCount": 1,
                "assetTagId": "0000123456",
                "deviceInstanceName": "device_name",
                "siteName": "site_name",
                "siteArn": "arn:aws:one:us-east-1:123456789012:site/12345678901234"
            }
        }
      }
```


## Device connectivity changed to online

The device is now connected to the internet.


```
{
        "version": "0",
        "id": "e6ecea28-dd60-5061-29f8-dfbc902f4123",
```

```
        "detail-type": "Some Alert(s) Cleared",
        "source": "aws.one",
        "account": "123456789012",
        "time": "2025-07-17T18:28:23Z",
        "region": "us-east-1",
        "resources":
        [
            "arn:aws:one:us-east-1:123456789012:deviceInstance/12345678901234"
        ],
        "detail":
        {
            "version": "1.0.0",
            "data":
            {
                "clearedAlerts":
                [
                    {
                        "statusCode": "NetworkDisconnected",
                        "reportedAt": "Thu Jul 17 18:28:23 UTC 2025"
                    }
                ],
                "currentAlerts":
                [],
                "currentAlertsCount": 0,
                "assetTagId": "0000123456",
                "deviceInstanceName": "device_name",
                "siteName": "site_name",
                "siteArn": "arn:aws:one:us-east-1:123456789012:site/12345678901234"
            }
        }
    }
```

## Device connectivity changed to offline

The device is no longer connected to the internet.

```
{
        "version": "0",
        "id": "e6ecea28-dd60-5061-29f8-dfbc902f4123",
        "detail-type": "New Alert(s) Detected",
        "source": "aws.one",
        "account": "123456789012",
        "time": "2025-07-17T18:28:23Z",
```

```
        "region": "us-east-1",
        "resources":
        [
            "arn:aws:one:us-east-1:123456789012:deviceInstance/12345678901234"
        ],
        "detail":
        {
            "version": "1.0.0",
            "data":
            {
                "newAlerts":
                [
                    {
                        "statusCode": "NetworkDisconnected",
                        "reportedAt": "Thu Jul 17 18:28:23 UTC 2025"
                    }
                ],
                "currentAlerts":
                [
                    {
                        "statusCode": "NetworkDisconnected",
                        "reportedAt": "Thu Jul 17 18:28:23 UTC 2025"
                    }
                ],
                "currentAlertsCount": 1,
                "assetTagId": "0000123456",
                "deviceInstanceName": "device_name",
                "siteName": "site_name",
                "siteArn": "arn:aws:one:us-east-1:123456789012:site/12345678901234"
            }
        }
    }
```

# Logging Amazon One Enterprise API calls using AWS CloudTrail

Amazon One Enterprise is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon One Enterprise. CloudTrail captures all API calls for Amazon One Enterprise as events. The calls captured include calls from the Amazon One Enterprise console and code calls to the Amazon One Enterprise API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon One Enterprise. 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 Amazon One Enterprise, 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.

## Amazon One Enterprise information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon One Enterprise, 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 Amazon One Enterprise, 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 Amazon One Enterprise actions are logged by CloudTrail and are documented in the Actions, resources, and condition keys for Amazon One Enterprise. For example, calls to the `ListSites`, `RebootDevice` and `DeleteDeviceInstance` 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 Amazon One Enterprise 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 `CreateSite` action.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDAKDBGOAT6C2EXAMPLE:J_DOE",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/J_DOE",
        "accountId": "123456789012",
        "accessKeyId": "AKIALAVPULGA71EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDAKDBGOAT6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-10-11T06:28:04Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-10-11T07:19:09Z",
    "eventSource": "one.amazonaws.com",
    "eventName": "CreateSite",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "XXX.XXX.XXX.XXX",
    "userAgent": "userAgent",
    "requestParameters": {
        "name": "***",
```

```
        "description": "***",
        "address": {
            "addressLine1": "***",
            "addressLine2": "***",
            "addressLine3": "***",
            "city": "EXAMPLE_CITY",
            "postalCode": "12345",
            "countryCode": "EXAMPLE_COUNTRY",
            "stateOrRegion": "EXAMPLE_STATE"
        },
        "clientToken": "abc12d34-567e-8910-1112-12fghi0jk13l"
    },
    "responseElements": {
        "stateOrRegion": "EXAMPLE_STATE",
        "createdAtInMillis": 1697008749263,
        "city": "EXAMPLE_CITY",
        "countryCode": "EXAMPLE_COUNTRY",
        "deviceInstanceCount": 0,
        "postalCode": "12345",
        "name": "***",
        "description": "***",
        "siteId": " abCdefG12hijkL",
        "siteArn": "arn:aws:one:us-east-1:123456789012:site/abCdefG12hijkL",
        "tags": "***"
    },
    "requestID": "1abcd23e-f4gh-567j-klm8-9np01q234r56",
    "eventID": "1234a56b-c78d-9e0f-g1h2-34jk56m7n890",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

# Troubleshooting Amazon One

If you have problems with the Amazon One Application or one of your Amazon One devices, use these suggestions to troubleshoot the problem. Then, if you're still having trouble, contact AWS Support.

**Topics**



# Troubleshooting Amazon One identity and access

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

**Topics**



## I am not authorized to perform an action in Amazon One

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional one:*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 one:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the one:*`GetWidget`* action.

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

### I want to allow people outside of my AWS account to access my Amazon One resources

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

To learn more, consult the following:

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

## Troubleshooting the Amazon One Console

If you have problems with the Amazon One Application or one of your Amazon One devices, use these suggestions to troubleshoot the problem. Then, if you're still having trouble, contact AWS Support.

**Topics**

## I am unable to create a site

- Contact your Amazon One Console administrator to provide you access.
- If the issue persists, contact AWS Support.

## I am unable to create a device instance

- Contact your Amazon One Console administrator to provide you access.
- If the issue persists, contact AWS Support.

## I am unable to create a configuration template

- Contact your Amazon One Console administrator to provide you access.
- If the issue persists, contact AWS Support.

## I am unable to create an activation QR code

- Contact your Amazon One Console administrator to provide you access.
- If the issue persists, contact AWS Support.

# Troubleshooting the Amazon One device

If you have problems with Amazon One Console or one of your Amazon One Devices, use these suggestions to troubleshoot the problem. Then, if you're still having trouble, contact AWS Support.

**Topics**

- Device instance does not exist
- Site not found
- ZIP Code does not match
- Gateway timed out
- I am unable to configure device
- Device restarted with error message and error code
- Amazon logo on the device screen with no further activity
- Temporarily unavailable
- Something went wrong on our end
- Temporarily out of service
- Amazon One device has physical damage
- Unable to read palm
- Palm not recognized
- Device locked due to extended inactivity
- Device locked due to tamper event

## Blank screen

This occurs when the device doesn't have power or gets stuck during reboot.

Perform the following to troubleshoot this issue:

- Wait a few moments (less than 30 seconds) in case the device is rebooting.
- If the light ring is pulsing while the device is blank, wait up to 30 seconds.
- Check if the power cord is plugged into both the power outlet as well as firmly in the back of the Amazon One device. Also, check that the cord is not damaged.
- Check the power source.
- Check that all the cables are connected properly to the Amazon One and USB hub.
- Reboot the device from the console.
- If rebooting the device doesn't fix the issue, unplug the Amazon One USB hub from the power supply, and then plug it back in.
- If the issue persists, contact AWS Support.

## I am unable to connect to Wi-Fi or network

This occurs when the device loses connectivity.

Perform the following to troubleshoot this issue:

- If connected to Wi-Fi, use another device to check if the Wi-Fi shows up in the available networks.
- Check if the Wi-Fi router is switched on and within range.
- The device will reconnect once the network recovers.
- If the issue persists, contact AWS support.

## Rebooting a device with active alerts

When a Reboot is requested from the console, the operation waits up to 15 minutes for the device to receive the command and attempt a reboot, even if it's offline or facing network issues.

Perform the following to troubleshoot this issue:

- Wait for the reboot to complete.
- If the issue persists, contact AWS support.

## System error

This occurs due to an internal error.

Perform the following to troubleshoot this issue:

- Choose **Restart** on the screen to restart the application.
- After 2 attempts, if the issue is not resolved, contact AWS Support.

## QR code is not recognized

This occurs because of an unauthorized QR code or expired QR code.

Perform the following to troubleshoot this issue:

- Choose **Try again** to navigate back to the QR code screen.

- Create a new QR code on the AWS console, and then scan the valid QR code.

## Unable to read QR code

This occurs when the application is unable to read the QR code.

Perform the following to troubleshoot this issue:

- Choose **Try again** to navigate back to the QR code screen.
- If the issue persists, cancel the activation workflow and restart.

## Multiple QR codes detected

This occurs when multiple QR codes are scanned.

Perform the following to troubleshoot this issue:

- Choose **Try again** to navigate back to the QR code screen.
- Scan only one valid QR code at a time.

## Device instance does not exist

This occurs when the device instance is deleted or does not exist in the AWS console.

Perform the following to troubleshoot this issue:

- Choose **Try again** to navigate back to the QR code screen.
- Check the AWS console for the correct device instance. If the device instance is missing, contact your administrator.
- Create a new QR code for that device instance, and then scan the new QR code.

## Site not found

This occurs when the site is deleted or does not exist in the AWS console.

Perform the following to troubleshoot this issue:

- Check the AWS console for the site information. If the site doesn't exist, contact your administrator.

## ZIP Code does not match

This occurs when entering a different ZIP Code than the one configured for the device.

Perform the following to troubleshoot this issue:

- Choose **Try again** to navigate back to the ZIP Code screen.
- Check if you have the correct site ZIP Code.
- If the issue persists, contact your administrator to check the site ZIP Code on the AWS console.

## Gateway timed out

This occurs when there is no response from gateway within a specified time.

Perform the following to troubleshoot this issue:

- Choose **Restart** to restart the application.
- After two attempts, if the issue is not resolved, contact AWS Support.

## I am unable to configure device

This occurs when the operation failed to save the configuration on the device disk.

Perform the following to troubleshoot this issue:

- Choose **Restart** to restart the application.
- After two attempts, if the issue is not resolved, contact AWS Support.

## Device restarted with error message and error code

Perform the following to troubleshoot this issue:

- Choose **Restart**, and let the device recover.
- If the device doesn't recover, unplug the USB hub from the power supply and reconnect.

- If the issue persists, contact AWS Support.

## Amazon logo on the device screen with no further activity

Perform the following to troubleshoot this issue:

- Wait a few moments (less than 30 seconds) in case the device is rebooting.
- Unplug the USB hub from the power supply and reconnect.
- If the issue persists, contact AWS Support.

## Temporarily unavailable

Perform the following to troubleshoot this issue:

- Ensure that the USB connections with the host device/system are secure.
- Disconnect and reconnect all the cables going into the USB hub.
- If the issue persists, contact AWS Support.

## Something went wrong on our end

This occurs when there is an internal error.

Perform the following to troubleshoot this issue:

1. Shut down the device.
2. Disconnect it from its power supply.
3. Wait 30 seconds.
4. Plug the device back into its power source.
5. Power on the device.
6. If the issue persists, contact AWS Support.

## Temporarily out of service

This occurs when the device has been moved out of service by Amazon One.

Perform the following to troubleshoot this issue:

- Contact AWS Support.

## Amazon One device has physical damage

Perform the following to troubleshoot this issue:

- Contact AWS Support for next steps, and provide as many details as possible, such as what happened, when it happened, and why it happened.

## Unable to read palm

Perform the following to troubleshoot this issue:

- Double-check that the Amazon One device is free from streaks and smudges.
- Ensure the customer's palm is free of occlusions such as bandages, sleeves, and significant dirt/oil.
- If the issue persists, and the device does not read any palm, contact AWS Support.

## Palm not recognized

Perform the following to troubleshoot this issue:

- Have the customer try using their other palm.
- Ensure the customer is already enrolled. If not, have them enroll online or on the device.
- If the issue persists, and the device does not read any palm contact, contact AWS Support.

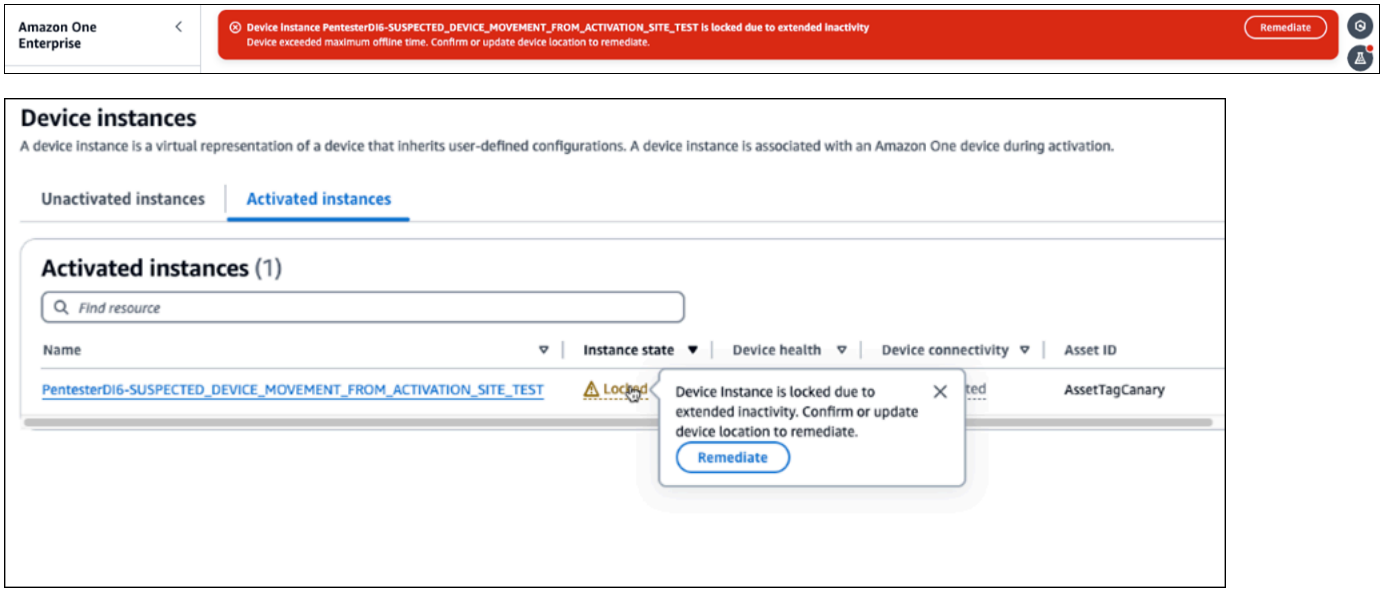
## Device locked due to extended inactivity

When the device suspects it has been moved from the activation site, it locks out users. This occurs when the device exceeds the maximum 120 hours of offline time.

Perform the following to unlock the device:

1. Log in to your AWS console, and choose the device instance.
2. From the error banner on the top of the page, select **Remediate**.

   Optionally: From **Activated instances**, select **Locked**, and choose **Remediate**.




3. If the device is still at the original activation site, choose **Yes, device is at this site**.
4. If the device is in a different site, choose **No, the device is at a different site**. Choosing **No** deactivates the device. Activate the device at the new site.

## Device locked due to tamper event

For security reasons, Amazon One device will be locked in case of any tamper event.

Perform the following to troubleshoot this issue:

- Contact AWS Support.

# Document history for the Amazon One Enterprise User Guide

The following table describes the documentation releases for Amazon One Enterprise.

| Change | Description | Date |
| --- | --- | --- |
| Update | Added Service-Linked Roles section | February 4, 2025 |
| Update | Added: Scenario-driven content | October 10, 2024 |
| Update | Added topic: Troubleshooting the Amazon One Enterprise console | October 10, 2024 |
| Update | Added topic: Troubleshooting the Amazon One Enterprise device | October 10, 2024 |
| Update | Added chapter: Setting up Amazon One Enterprise | October 10, 2024 |
| Update | Added topic: Maintaining and cleaning Amazon One Enterprise devices | October 10, 2024 |
| Update | Reorganized content | October 10, 2024 |
| Update | Added topic: Installing Amazon One Enterprise device I/O Hub for secure access | August 14, 2024 |
| Update | Added topic: Installing a wall-mountable Amazon One Enterprise device | June 5, 2024 |

| | | |
|---|---|---|
| Initial release | Initial release of the Amazon One Enterprise User Guide | November 27, 2023 |