Administrator Guide

Amazon WorkSpaces Thin Client

Amazon WorkSpaces Thin Client: Administrator Guide

Table of Contents

What is the Amazon WorkSpaces Thin Client administrator console?

With the Amazon WorkSpaces Thin Client administrator console, administrators can manage WorkSpaces Thin Client environments and devices through a WorkSpaces Thin Client portal. From this web console, administrators can create environments, manage devices, and set parameters for WorkSpaces Thin Client users within their network.

Virtual desktop environments that you use for WorkSpaces Thin Client must be created or modified within their own console.

> ⚠ **Important**
>
> For WorkSpaces Thin Client administrator console to work properly, your system must first meet specific requirements. These requirements are listed in Prerequisites and Configurations.

Topics

- Are you a first-time user?
- Architecture

Are you a first-time user?

If you are a first-time user of WorkSpaces Thin Client administrator console, we recommend that you begin by reading the following sections:

- *Starting the WorkSpaces Thin Client administrator console*
- *Using WorkSpaces Thin Client administrator console*

Architecture

Each WorkSpaces Thin Client is associated with a virtual desktop interface (VDI) provider. WorkSpaces Thin Client supports three VDI providers:

- Amazon WorkSpaces
- AppStream 2.0
- Amazon WorkSpaces Secure Browser

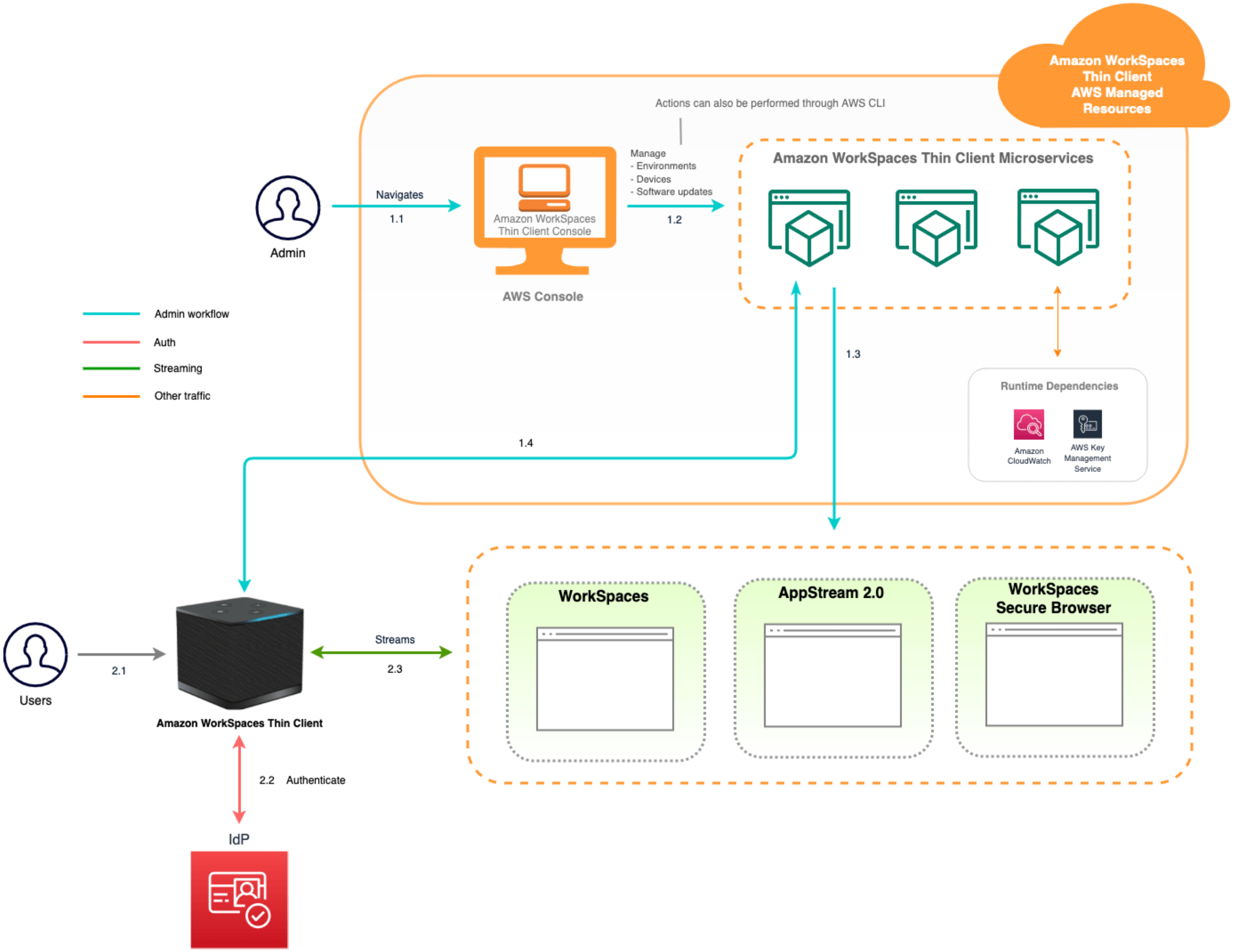
Depending on the VDI used, information for your WorkSpaces Thin Client is accessed and managed either via directories for WorkSpaces, stacks for AppStream 2.0, and web portal endpoints for WorkSpaces Secure Browser.

For more information on Amazon WorkSpaces, see Get started with WorkSpaces quick setup. Directories are managed through the AWS Directory Service, which offers the following options: Simple AD, AD Connector, or AWS Directory Service for Microsoft Active Directory, also known as AWS Managed Microsoft AD. For more information, see the AWS Directory Service Administration Guide.

For more information on AppStream 2.0, see Get Started with Amazon AppStream 2.0: Set Up With Sample Applications. AppStream 2.0 manages the AWS resources required to host and run your applications, scales automatically, and provides access to your users on demand. AppStream 2.0 provides users access to the applications they need on the device of their choice, with a responsive, fluid user experience that is indistinguishable from natively installed applications.

For information on WorkSpaces Secure Browser, see Getting started with Amazon WorkSpaces Secure Browser. Amazon WorkSpaces Secure Browser is an on-demand, fully managed, Linux-based service designed to facilitate secure browser access to internal websites and software-as-a-service (SaaS) applications. Access the service from existing web browsers, without the administrative burden of infrastructure management, specialized client software, or virtual private network (VPN) solutions.

The following diagram shows the architecture of WorkSpaces Thin Client.


Amazon WorkSpaces Thin Client AWS Managed Resources
Actions can also be performed through AWS CLI
Admin
Navigates
1.1
Amazon WorkSpaces Thin Client Console
AWS Console
Manage
- Environments
- Devices
- Software updates
1.2
Amazon WorkSpaces Thin Client Microservices
1.3
Runtime Dependencies
Amazon CloudWatch
AWS Key Management Service
1.4
Admin workflow
Auth
Streaming
Other traffic
Users
2.1
Amazon WorkSpaces Thin Client
Streams
2.3
2.2 Authenticate
IdP
WorkSpaces
AppStream 2.0
WorkSpaces Secure Browser

Setting up Amazon WorkSpaces Thin Client administrator console

Topics

Sign up for AWS

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.

Create an IAM user

To create an administrator user, choose one of the following options.

Choose one way to manage your administrator	To	By	You can also
In IAM Identity Center (Recommended)	Use short-term credentials to access AWS. This aligns with the security best practices. For information about best practices, see Security best practices in IAM in the *IAM User Guide*.	Following the instructions in Getting started in the *AWS IAM Identity Center User Guide*.	Configure programmatic access by Configuring the AWS CLI to use AWS IAM Identity Center in the *AWS Command Line Interface User Guide*.
In IAM (Not recommended)	Use long-term credentials to access AWS.	Following the instructions in Creating your first IAM admin user and user group in the *IAM User Guide*.	Configure programmatic access by Managing access keys for IAM users in the *IAM User Guide*.

Getting started with your VDI for Amazon WorkSpaces Thin Client

Amazon WorkSpaces Thin Client is a cost-effective thin client device built to work with AWS End User Computing services to provide you with secure, instant access to applications and virtual desktops.

Choose a virtual desktop infrastructure (VDI), and configure it to work with WorkSpaces Thin Client.

> ⚠ **Important**
>
> For WorkSpaces Thin Client administrator console to work properly, your system must first meet specific requirements. These requirements are listed in the configuration procedure for each virtual desktop provider.

WorkSpaces Thin Client requires specific software configurations, depending on your virtual desktop provider.

Topics

Configuring WorkSpaces for Amazon WorkSpaces Thin Client

For WorkSpaces Thin Client to be used with Amazon WorkSpaces, your service will need to be configured to access the WorkSpaces directories. Amazon WorkSpaces are listed based on their directory names on the WorkSpaces Thin Client **Create environment** page within AWS console.

> ⓘ **Note**
>
> Configurations must be made before using the console for the first time. It is not recommended that you modify any prerequisite features after you start using the console.

Before you begin

Make sure that you have an AWS account to create or administer a WorkSpace. Device users, however, don't need an AWS account to connect to and use their WorkSpaces.

Review and understand the following concepts before you proceed with your configuration:

- When you launch a WorkSpace, select a WorkSpace bundle. For more information, see Amazon WorkSpaces Bundles.
- When you launch a WorkSpace, select which protocol that you want to use with your bundle. For more information, see Protocols for Amazon WorkSpaces.
- When you launch a WorkSpace, specify the profile information for each user, including username and email address. Users complete their profiles by creating a password. Information about WorkSpaces and users is stored in a directory. For more information, see Manage directories for WorkSpaces.
- When you launch a WorkSpace, enable and configure the WorkSpaces web access. For more information, see Enable and configure Amazon WorkSpaces Web Access

Step 1: Verify that your system meets WorkSpaces required features

For WorkSpaces Thin Client administrator console to work properly with Amazon WorkSpaces, your system must meet the following specific requirements. This table lists all of these supported features and their requirements.

Feature	Requirement
Web access	Enabled
Supported operating system	• Windows 10 • Windows 10 (Bring Your Own License) • Windows 11 • Windows 11 (Bring Your Own License)
Supported bundles	• Microsoft Power with Windows 10 (Server 2016, 2019, and 2022 based) • Microsoft Power with Windows 10 (Server 2016, 2019, and 2022 based) w Office

Feature	Requirement
	• Microsoft PowerPro with Windows 10 (Server 2016, 2019, and 2022 based) • Microsoft PowerPro with Windows 10 (Server 2016, 2019, and 2022 based) w Office • Microsoft Performance with Windows 10 (Server 2016, 2019, and 2022 based) • Microsoft Performance with Windows 10 (Server 2016, 2019, and 2022 based) w Office
Supported protocol	WSP only

Step 2: Use advanced setup to launch your WorkSpace

To use advanced setup to launch your WorkSpace

1. Open the WorkSpaces console at https://console.aws.amazon.com/workspaces/.
2. Choose one of the following directory types, and then choose **Next**:
   - AWS Managed Microsoft AD
   - Simple AD
   - AD Connector
3. Enter the directory information.
4. Choose two subnets in a VPC from two different Availability Zones. For more information, see Configure a VPC with public subnets.
5. Review your directory information and choose **Create directory**.

Configuring AppStream 2.0 for Amazon WorkSpaces Thin Client

AppStream 2.0 instances will be listed based on Stack names and will require an IdP login URL to be configured on the create environment page. Because SAML authentication for AppStream 2.0

only supports initiated authentication, the administrator will have to enter the correct login URL manually.

> **Note**
>
> Configurations must be made before using the console for the first time. It is not recommended that you modify any prerequisite features after you start using the console.

Step 1: Verify that your system meets AppStream 2.0 required features

For WorkSpaces Thin Client administrator console to work with AppStream 2.0 properly, your system must meet the following specific requirements. This table lists all of these supported features and their requirements.

Feature	Requirement
Identity Provider	Go to Setting Up SAML in the AppStream 2.0 Administrator Guide to create an Identity Provider. When prompted to **Create env console**, enter your IDP Login URL.
Operating system	Windows
Platform Type	Windows Server (2012 R2, 2016 or 2019)
Streaming protocol	TCP Streaming There is an auto fallback mechanism to TCP if UDP is not available.
Local Copy and Paste	Disable Configured at AppStream 2.0 stack level
Local Folder Sharing	Disable Configured at AppStream 2.0 stack level

Feature	Requirement
Local Printing	Disable Configured at AppStream 2.0 stack level

The screen lock requirement through SAML authentication on AppStream 2.0 is also supported. The **User Pool** and **Programmatic** authentication mechanisms are not supported on WorkSpaces Thin Client.

Step 2: Set up your AppStream 2.0 stacks

To stream your applications, AppStream 2.0 requires an environment that includes a fleet that is associated with a stack, and at least one application image. Follow these steps to set up a fleet and stack and give users access to the stack. If you haven't already done so, we recommend that you try the procedures in Get Started with AppStream 2.0: Set Up With Sample Applications.

If you want to create an image to use, see Tutorial: Create a Custom AppStream 2.0 Image by Using the AppStream 2.0 Console.

If you plan to join a fleet to an Active Directory domain, configure your Active Directory domain before completing the following steps. For more information, see Using Active Directory with AppStream 2.0.

Tasks

- Create a Fleet
- Create a Stack
- Provide Access to Users
- Clean Up Resources

Configuring Amazon WorkSpaces Secure Browser for Amazon WorkSpaces Thin Client

Amazon WorkSpaces Secure Browser are based on their web portal endpoints on the WorkSpaces Thin Client **Create environment** page within AWS console.

Note

Configurations must be made before using the console for the first time. It is not recommended that you modify any prerequisite features after you start using the console.

Step 1: Verify that your system meets Amazon WorkSpaces Secure Browser required features

For WorkSpaces Thin Client Administrator Console to work properly with Amazon WorkSpaces Secure Browser, your system must meet the following specific requirements. This table lists all of these supported features and their requirements.

Feature	Requirement
Local Copy and Paste	Disable
Local Folder Sharing	Disable

Note

The WorkSpaces Secure Browser extension for single sign-on is not currently supported on WorkSpaces Thin Client.

Step 2: Set up WorkSpaces Secure Browser portals

WorkSpaces Thin Client works with the WorkSpaces Secure Browser VPC in a specific configuration:

1. Create a VPC using the AWS CodeBuild Cloudformation template.
2. Set up your Identity Provider.
3. Create an Amazon WorkSpaces Secure Browser portal.
4. Test your new Amazon WorkSpaces Secure Browser portal.

Starting the WorkSpaces Thin Client administrator console

WorkSpaces Thin Client is a cost-effective thin client device built to work with AWS End User Computing services to provide you with secure, instant access to applications and virtual desktops.

Topics

- Covered Regions
- Launching the WorkSpaces Thin Client administrator console

Covered Regions

WorkSpaces Thin Client is available in the following Regions.

Only the WorkSpaces Thin Client administrator console is available in these Regions. WorkSpaces Thin Client devices are only currently available in the US, Germany, France, Italy, and Spain.

Region Name	Region	Endpoint	Console link
US East (N. Virginia)	us-east-1	thinclient.us-east-1.amazonaws.com	https://us-east-1.console.aws.amazon.com/workspaces-thin-client/home
US West (Oregon)	us-west-2	thinclient.us-west-2.amazonaws.com	https://us-west-2.console.aws.amazon.com/workspaces-thin-client/home
Asia Pacific (Mumbai)	ap-south-1	thinclient.ap-south-1.amazonaws.com	https://ap-south-1.console.aws.amazon.com/workspaces-thin-client/home

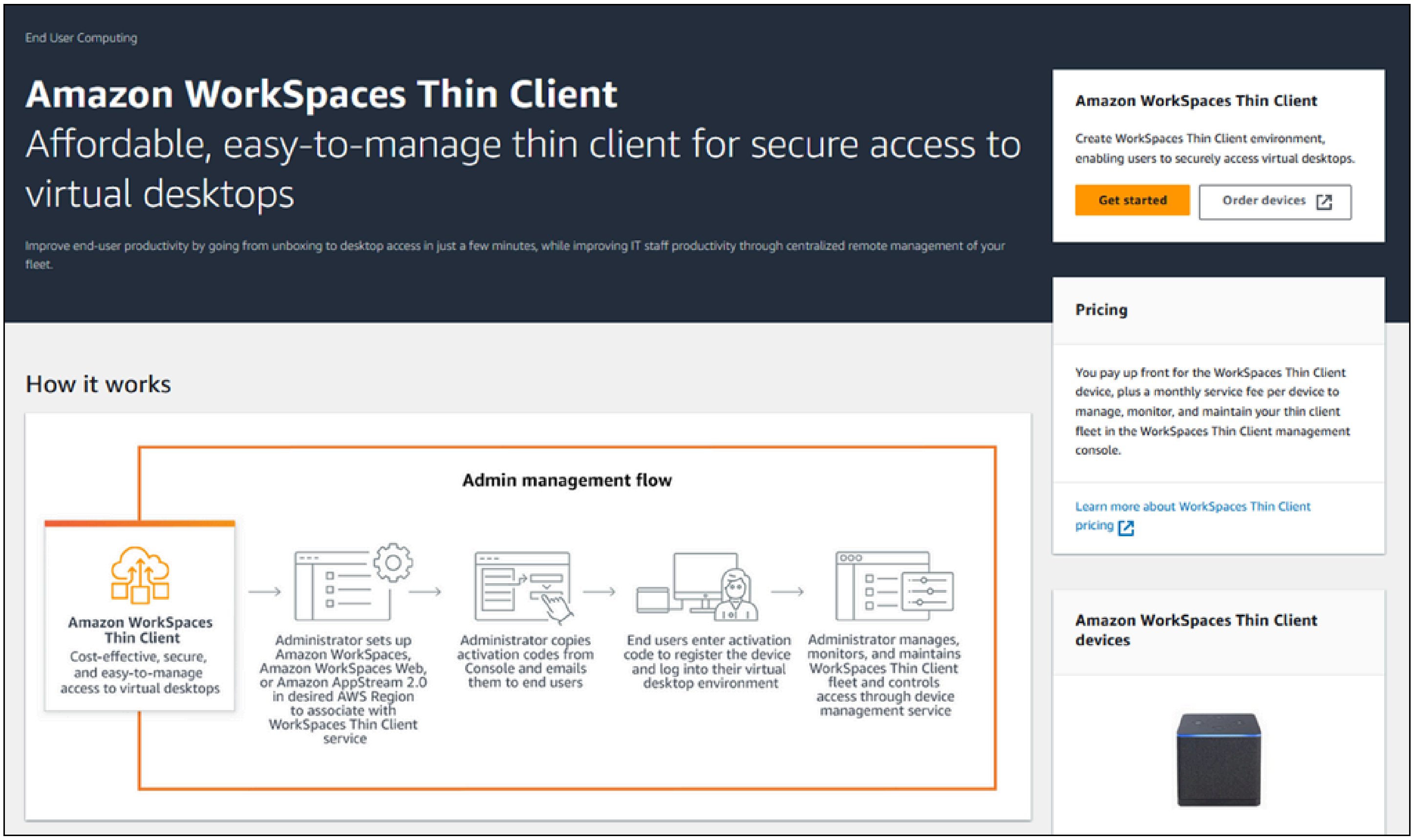
Region Name	Region	Endpoint	Console link
Europe (Ireland)	eu-west-1	thinclient.eu-west-1.amazonaws.com	https://eu-west-1.console.aws.amazon.com/workspaces-thin-client/home
Canada (Central)	ca-central-1	thinclient.ca-central-1.amazonaws.com	https://ca-central-1.console.aws.amazon.com/workspaces-thin-client/home
Europe (Frankfurt)	eu-central-1	thinclient.eu-central-1.amazonaws.com	https://eu-central-1.console.aws.amazon.com/workspaces-thin-client/home
Europe (London)	eu-west-2	thinclient.eu-west-2.amazonaws.com	https://eu-west-2.console.aws.amazon.com/workspaces-thin-client/home

Launching the WorkSpaces Thin Client administrator console

When you have an AWS account, you can launch the administrator console and go to the WorkSpaces Thin Client console. To launch the console, do the following:

1. Log on to your AWS account.
2. Access the WorkSpaces Thin Client console.
3. Select **Get Started** and you will be directed to Environments.

Using WorkSpaces Thin Client administrator console



End User Computing

Amazon WorkSpaces Thin Client

Affordable, easy-to-manage thin client for secure access to virtual desktops

Improve end-user productivity by going from unboxing to desktop access in just a few minutes, while improving IT staff productivity through centralized remote management of your fleet.

How it works

Admin management flow

Amazon WorkSpaces Thin Client

Cost-effective, secure, and easy-to-manage access to virtual desktops

Administrator sets up Amazon WorkSpaces, Amazon WorkSpaces Web, or Amazon AppStream 2.0 in desired AWS Region to associate with WorkSpaces Thin Client service

Administrator copies activation codes from Console and emails them to end users

End users enter activation code to register the device and log into their virtual desktop environment

Administrator manages, monitors, and maintains WorkSpaces Thin Client fleet and controls access through device management service

Amazon WorkSpaces Thin Client

Create WorkSpaces Thin Client environment, enabling users to securely access virtual desktops.

Pricing

You pay up front for the WorkSpaces Thin Client device, plus a monthly service fee per device to manage, monitor, and maintain your thin client fleet in the WorkSpaces Thin Client management console.

Learn more about WorkSpaces Thin Client pricing

Amazon WorkSpaces Thin Client devices

Welcome to the WorkSpaces Thin Client Administrator Console!

From here, you can manage your fleet of WorkSpaces Thin Client devices and environments for your team.

For information regarding the WorkSpaces Thin Client device, please refer to the WorkSpaces Thin Client User Guide.

Let's get started.

Topics

- Environments
- Devices
- Software updates

Environments

Each WorkSpaces Thin Client device uses an individual virtual desktop environment to access its online resources. Users access this environment by using one of the following virtual desktop providers:

- Amazon WorkSpaces
- AppStream 2.0
- Amazon WorkSpaces Secure Browser

Environment list

Environment list details

Name - The unique identifier associated with this environment.

Virtual desktop service - The virtual desktop provider that this environment uses.

Virtual desktop service ID - The unique identifier that the virtual desktop service provider assigns to this environment.

Activation code - The code that is used by end users to access the virtual desktop environment.

Device count - The number of WorkSpaces Thin Client devices that are accessing this environment.

Environment list actions

Search - Searches all environments that you manage.

Refresh - Refreshes the environment list.

View details - Displays Environment details.

Actions - Opens a dropdown list where you can Edit or Delete an environment.

Create environment - Starts the process of creating an environment

Create environment - Starts the process of creating an environment.

Topics

- Environment Details
- Creating an environment
- Editing an environment
- Deleting an environment

Environment Details

When you select an environment, the WorkSpaces Thin Client console displays the details for that environment for you to review. The console also displays the details about the virtual desktop provider that this environment uses.

Topics

- Summary
- Virtual desktop environment details

Summary

Name - The unique identifier associated with this environment.

Virtual desktop service - The virtual desktop provider that this environment uses.

Virtual desktop service ID - The unique identifier that the virtual desktop service provider assigns to this environment.

Activation code - This code is used by end users to access the virtual desktop environment.

Always keep software up-to-date - This setting enables automatic software updates.

Maintenance window start time - The time each week when automatic software updates begin.

Maintenance window end time - The time each week when automatic software updates finish.

Maintenance window days of the week - The days that automatic software updates occur.

Associated devices - The number of WorkSpaces Thin Client devices that are accessing this environment.

Time created - The date and time that this environment was created.

Virtual desktop environment details

Amazon WorkSpaces directory details

Directory ID - The Amazon WorkSpaces directory associated with this environment.

Directory name - The unique identifier associated with this Amazon WorkSpaces directory.

Organization name - The name of the organization that controls the Amazon WorkSpaces directory.

Directory type - The format of the Amazon WorkSpaces directory.

Registered - Whether this Amazon WorkSpaces directory is registered.

Status - Whether this Amazon WorkSpaces directory is active.

Amazon WorkSpaces Secure Browser portal details

Name - The unique identifier associated with this Amazon WorkSpaces Secure Browser portal.

Time created - The date and time when this AppStream 2.0 stack was created.

Web portal endpoint - The url used to access your virtual desktop environment.

AppStream 2.0 details

Stack name - The unique identifier associated with this AppStream 2.0 stack.

IdP login url - The identity provider url that is used to log in and out of your AppStream 2.0 stack.

Time created - The date and time when this AppStream 2.0 stack was created.

Creating an environment

To begin, each device requires an AWS End User Computing service. WorkSpaces Thin Client uses the following services:

- Amazon WorkSpaces through an assigned directory

- AppStream 2.0 through an assigned stack
- Amazon WorkSpaces Secure Browser through a web portal address

You must either assign a service to an existing environment or create a new one.

> **Note**
>
> WorkSpaces Thin Client only displays virtual desktops in the same Region.

Topics

- Step 1: Enter your environment details
- Step 2: Select your virtual desktop provider
- Step 3: Send the activation code to your device users

Step 1: Enter your environment details

1. Enter a name for your environment in the **Environment details** field.
2. To set up automatic software patches, check the box for **Always keep software up-to-date**.

   > **Note**
   >
   > If automatic software updates is not enabled, the devices registered to this environment won't receive software updates until you manually push the update or when the software reaches its expiration and the system forces an update.
   > Also, the devices Software Set version is determined by the system. This version may not be the most recent one.

3. Select when you want to schedule the maintenance window for your environment.
   - **Apply system wide maintenance window** - Automatically updates the environment software at a determined time each week.
   - **Apply custom maintenance window** - Set a day and time when you want the environment software to update each week.
4. Select a virtual desktop service.

- Amazon WorkSpaces
- Amazon WorkSpaces Secure Browser
- AppStream 2.0

Step 2: Select your virtual desktop provider

You must have a service to provide your users access to their virtual desktop and compatible resources.

> ⚠ **Important**
>
> For WorkSpaces Thin Client Administrator Console to work properly, your system must meet specific requirements. These requirements are listed in Prerequisites and Configurations.
> Make sure that your system meets these requirements before you set up your console.

Using Amazon WorkSpaces

Amazon WorkSpaces is a fully managed desktop virtualization service for Windows that enables you to access resources from any supported device.

1. To use Amazon WorkSpaces, do one of the following:
   - Select the directory that you want to use for your environment. You can either browse through the dropdown list or you can search the directories by using the search field.
   - Create a directory by selecting the **Create WorkSpaces directory** button. For more information on creating WorkSpaces directories, see Manage directories for WorkSpaces.
2. Select the **Create environment** button.

When you create your environment, you can still edit the details later. For more information, see Editing an environment.

Using AppStream 2.0

AppStream 2.0 is a fully managed, secure application streaming service that you can use to stream desktop applications from AWS to a web browser.

> ⚠ **Important**
>
> In order to create an AppStream 2.0 environment, you must have `cli_follow_urlparam` set to `false`. To achieve this, do the following:
>
> - For a default profile, run `aws configure set cli_follow_urlparam false`.
> - For a profile with name `ProfileName`, run `aws configure set cli_follow_urlparam false --profile ProfileName`.

1. To set up AppStream 2.0, do one of the following:

   - Select the stack that you want to use for your environment. You can either browse through the dropdown list or you can search the stacks by using the search field.

     > ⓘ **Note**
     >
     > If you do not see your existing stacks on the list, verify in the AppStream 2.0 Management Console that it meets the WorkSpaces Thin Client requirements.

   - Create a stack by selecting the **Create Stack** button. For more information on creating AppStream 2.0 stacks, see Create a Stack.

2. Enter your identity provider login and logout URL in the **IdP login URL** field. This provides users with a place to log in and out of WorkSpaces Thin Client.
3. Select the **Create environment** button.

After you create your environment, you can still edit the details later. For more information, see Editing an environment.

Using Amazon WorkSpaces Secure Browser

Amazon WorkSpaces Secure Browser is a low-cost, fully managed WorkSpaces console that is built to deliver secure web-based workloads and software as a service (SaaS) application access to users within existing web browsers.

1. To set up Amazon WorkSpaces Secure Browser, do one of the following:

- Select the web portal that you want to use for your environment. You can either browse through the dropdown list or you can search the web portals by using the search field.

  > **Note**
  >
  > If you do not see your existing web portals in the list, verify in the WorkSpaces Secure Browser Management Console that it meets the WorkSpaces Thin Client requirements.

- Create a web portal by selecting the **Create WorkSpaces Secure Browser** button. For more information on creating WorkSpaces Secure Browser web portals, see Setting up Amazon WorkSpaces Secure Browser.

2. Select the **Create environment** button.

After you create your environment, you can still edit the details later. For more information, see Editing an environment.

Step 3: Send the activation code to your device users

After you set your environment and virtual desktop service, you will receive a unique activation code for your setup on the AWS Management Console.

Provide this activation code to any WorkSpaces Thin Client device user, and they can use it to access their virtual desktop.

See the WorkSpaces Thin Client User Guide for additional information on how to help your device user set up their Amazon WorkSpaces Thin Client.

Editing an environment

The WorkSpaces Thin Client administration console manages virtual desktop environments for individual users. From this console, you can edit or delete virtual desktop environments.

1. Select the environment that you want to edit.

> **Note**
>
> You can either browse through the dropdown list or you can search the environments by using the search field.

2. Select the **Actions** button.
3. Select **Edit** from the dropdown list. You will be directed to the **Edit environment** window.
4. Edit any of the following:

   - Change the name of your environment in the **Environment name** field.
   - Change the check box for **Software updates details** for automatic software patch updates.
   - Change when you want to schedule the maintenance window for your environment.
5. Select the **Edit environment** button.

Deleting an environment

> **Note**
>
> You cannot delete an environment if it has any devices registered to it. First, you must deregister and delete all devices in an environment.

1. Select the environment that you want to delete. You can either browse through the dropdown list or you can search the environments by using the search field.
2. Select the **Actions** button.
3. Select **Delete** from the dropdown list. The **Delete environment** confirmation window appears.
4. Type "delete" in the confirmation field.
5. Select the **Delete** button.

Devices

Each WorkSpaces Thin Client end user has a dedicated device that connects them to their virtual desktop environments and online resources. These devices are managed through the WorkSpaces Thin Client administrator console on the AWS site.

From this console, you can order devices for your team.

Device list

Device list details

Device ID - The identification number assigned to an individual device.

Device name - (*optional*) The unique name that you give to a device.

Activity status - The current status of a device. There are two status states:

- **Active** - Connected to a network at least once in the past seven days.
- **Inactive** - Not connected to a network in the past seven days.

Enrollment status - Confirmation that a device has been set up, is associated with this AWS account, and is part of a specific environment. It can be in one of the following states:

- **Registered** - This is the default status.
- **Deregistering** - The device is in the **Reset and Deregister** process.

> **Note**
>
> You can delete a device if it is in a deregistering state.

- **Deregsitered** - The device has been successfully deregistered.

> **Note**
>
> You can only delete a device if it's in either a **Deregistering** or **Deregistered** status.

- **Archived** - The device is archived.

Environment ID - The identifier of the environment to which this device is attached.

Software compliance - The compliance status of the device software. There are two status states:

- Compliant

- Not compliant

Device list actions

Search - Searches all devices that you manage.

Refresh - Refreshes the device list.

View details - Displays Device details.

Actions - Opens a dropdown list where you can do the following:

- Edit device name
- Deregister
- Archive
- Delete
- Export device details

Order devices - Starts the process of ordering devices.

Topics

- Device details
- Editing a device name
- Resetting and deregistering a device
- Archiving a device
- Deleting a device
- Exporting device details

Device details

Summary

Device serial number - The identification number assigned to an individual device.

ARN - The unique identifier for the device in Amazon Resource Name (ARN) format.

Device name - The name that you give to a device. If you have not created a name, you can name it, or it will get a default name.

Device type - The type of end user device that is linked to the account.

Activity status - The current status of this device. The two status states are:

- Active
- Inactive

Environment ID - The identification number of the environment that the device uses.

Enrollment status - Confirmation that a device has been set up, is associated with this AWS account, and is part of a specific environment. It can be in one of the following four states:

- **Registered** - This is the default status.
- **Deregistering** - The device is in the **Reset and Deregister** process.
- **Deregistered** - The device has been successfully deregistered.

> **Note**
>
> You can only delete the device if it's in either a **Deregistered** or **Archived** status.

- **Archived** - This device has been marked by the administrator as not currently in service.

Enrolled since - The date the device was activated.

Last logged in - The date and time of the most recent login.

Last posture checked at - The date and time of the most recent device check-in.

Current software version - The software version that this device is currently using.

Scheduled for software update - The scheduled software version on the device.

Software compliance - Confirmation that the software set is valid. There are two status states:

- Compliant
- Not Compliant

User log

Last device access - The date and time when this device was last used.

Editing a device name

1. Select the device that you want to edit. You can either browse through the dropdown list or you can search for device by using the search field.
2. Select the **Actions** button.
3. Select **Edit device name** from the dropdown list. The **Edit device name** window appears.
4. Enter the new device name in the **Device name** confirmation field.
5. Select the **Save** button.

Resetting and deregistering a device

1. Select the device that you want to deregister. You can either browse through the dropdown list or you can search for the device by using the search field.
2. Select the **Actions** button.
3. Select **Deregister** from the dropdown list. The **Deregister** window appears.
4. Enter "deregister" in the confirmation field.
5. Select the **Deregister** button.

> **Note**
>
> Deregistering forcibly logs out the user and require a reboot of their WorkSpaces Thin Client device in the middle of a session.

Archiving a device

1. Select the device that you want to archive. You can either browse through the dropdown list or you can search for the device by using the search field.
2. Select the **Actions** button.
3. Select **Archive** from the dropdown list. The **Archive** window appears.
4. Enter "reset and archive" in the confirmation field.

5. Select the **Reset and archive** button.

> **Note**
>
> Archiving a device forcibly logs out the user and require a reboot of their WorkSpaces Thin Client device in the middle of a session.

Deleting a device

1. Select the device that you want to delete. You can either browse through the dropdown list or you can search for the device by using the search field.
2. Select the **Actions** button.
3. Select **Delete** from the dropdown list. The **Delete** window appears.
4. Enter "delete" in the confirmation field.
5. Select the **Delete** button.

> **Note**
>
> When the device has been successfully deleted, the user must return the WorkSpaces Thin Client device back to Amazon.

Exporting device details

1. Select the device from which you want to export the details. You can either browse through the dropdown list or you can search for the device by using the search field.
2. Select the **Actions** button.
3. Select **Export device details** from the dropdown list. The details for the selected device download in a spreadsheet format.

Software updates

WorkSpaces Thin Client sometimes requires software updates that introduce new functionality and apply security patches. These updates are represented by a versioned **Software set**.

A **Software set** can contain updates to the software applications or operating system for the WorkSpaces Thin Client device. From this console, you can choose to update the software immediately or you can schedule an automatic update during the maintenance window for the environments.

Refer to WorkSpaces Thin Client environment software sets for the list of released Software Sets.

Topics

- Updating environment software
- Updating device software
- WorkSpaces Thin Client software releases

Updating environment software

WorkSpaces Thin Client is an AWS End User Computing service that provides users access to virtual desktops. These virtual desktops are periodically updated with new software sets. To update environment software, do the following:

1. Select the software set from the list in **Available software updates**. For a list of software sets, refer to WorkSpaces Thin Client environment software sets.
2. Select the **Install** button.
3. Select **Environments** at the top of the page.
4. Select the environment to update from the list in the **Environments** section.
5. Select when to update the environment in the **Schedule the update** by choosing one of the following:

   - **Update software now** - Starts the update of the environment software on all registered devices.

     > **Note**
     >
     > Updating software now may interrupt any active user sessions.

   - **Update software during each environments maintenance window** - Updates the environment software during the scheduled maintenance window for the environment.
6. Check the box to authorize the update. This box must be checked for the software to update.

7. Select the **Install** button.

Updating device software

WorkSpaces Thin Client is an AWS End User Computing service that provides a thin client device that connects users to dedicated virtual desktops. These devices are periodically updated with new software. To update device software, do the following:

1. Select the software set from the list in **Available software updates**.
2. Select the **Install** button.
3. Select **Device** at the top of the page.
4. Select the device or devices to update from the list in the **Devices** section. For a list of software sets, refer to WorkSpaces Thin Client environment software sets.
5. Select when to update the environment from the **Schedule the update** options by choosing one of the following:

   - **Update software now** - Immediately updates the device software.

     > **Note**
     >
     > Updating the software now may interrupt any active user sessions.

   - **Update software during each devices maintenance window** - Updates the environment software during the scheduled maintenance window for the device.
6. Check the box to authorize the update. This box must be checked for the software to update.
7. Select the **Install** button.

WorkSpaces Thin Client software releases

WorkSpaces Thin Client is an AWS End User Computing service that provides users access to virtual desktops on a device. These devices are periodically updated with new software sets. The following table describes all the released software sets. Administrators can use the AWS management console to view available software sets.

Software set	Release date	Changes
2.5.0	06-13-2024	• Fixed the issue where device showed keyboard and mouse setup screen briefly on waking up from sleep before launching the session. • The **Home** button on the device toolbar renamed to **Sign In**. • Improvements to performance of audio/video calls in the session.
2.4.3	05-29-2024	• Zero-day fix for Chromium's CVE-2024-5274 critical security issue.
2.4.2	05-17-2024	• Zero-day fix for Chromium's CVE-2024-4947 critical security issue.
2.4.1	05-15-2024	• Zero-day fixes for Chromium's CVE-2024-4671 and CVE-2024-4761 critical security issues. • Fixed the issue that allowed right-clicking on AWS and Privacy links on WorkSpaces sign-in page to open the browser in a stand-alone mode.
2.4.0	05-09-2024	• Fixed an issue blocking "accounts.google.com" and preventing the use

Software set	Release date	Changes
		of Google Workspace as the IDP for AppStream 2.0 session. • Device settings toolbar auto-collapses with a click in any area on the screen.
2.3.0	04-05-2024	• Device settings show up in a collapsed toolbar allowing better utilization of the visible screen. • End users can now configure the duration to wait before the device sleeps on inactivity. • Fixed the issue where "about:blank" URL shows up on the second display. • Fixed the issue that resulted in a white screen when extended display is closed. • Volume levels set by end users now persists across device restarts.
2.2.1	02-16-2024	• Fixed an issue that occurs during the sign-in process that prevented users from logging into WorkSpaces configured with SAML 2.0 authentication.

Software set	Release date	Changes
2.2.0	02-08-2024	• Added support for ISO keyboards with English (United Kingdom), French, German, Italian, Spanish locales.
2.1.2	01-26-2024	• Zero-day fix for Chromium' s CVE-2024-0519 critical security issue. • Improvement to end user latency associated with Lock functionality. • Internal device-facing endpoints are switched over to 'thinclient*' domain.
2.1.1	12-21-2023	• Zero-day fix for Chromium' s CVE-2023-7024 critical security issue.
2.1.0	12-20-2023	• Adds a **Home** button to the device settings and enables support for Meta keys. This allows ends users to invoke the lock screen by pressing Meta+L.
2.0.1	12-06-2023	• Zero-day fix for Chromium' s CVE-2024-6345 critical security issue.
2.0.0	11-15-2023	• Initial release

Using tags on WorkSpaces Thin Client resources

You can organize and manage the resources for your WorkSpaces Thin Client by assigning your own metadata to each resource as tags. You specify a key and a value for each tag. A key can be a general category, such as "project," "owner," or "environment," with specific associated values. You can use tags as a simple yet powerful way to manage AWS resources and to organize data, including billing data.

When you add tags to an existing resource, those tags don't appear in your cost allocation report until the first day of the following month. For example, if you add tags to an existing WorkSpaces Thin Client device on July 15, the tags won't appear in your cost allocation report until August 1. For more information, see Using Cost Allocation Tags in the *AWS Billing User Guide*.

> **Note**
>
> To view your WorkSpaces Thin Client resource tags in the Cost Explorer, you must activate the tags that you have applied to your WorkSpaces Thin Client resources by following the instructions in Activating User-Defined Cost Allocation Tags in the *AWS Billing User Guide*. Tags appear 24 hours after activation, but it can take 4–5 days for values associated with those tags to appear in the Cost Explorer. Additionally, to appear and provide cost data in Cost Explorer, WorkSpaces Thin Client resources that have been tagged must incur charges during that time. Cost Explorer only shows cost data from the time when the tags were activated. No historical data is available at this time.

Resources that you can tag:

- You can add tags to the following resources when you create them—WorkSpaces Thin Client environments.
- You can add tags to existing resources of the following types—WorkSpaces Thin Client environments, devices, and software sets.
- You can configure the tags for a device in an environment to be automatically applied when you register a device.

Tag restrictions

- Maximum number of tags per resource—50

- Maximum key length—128 Unicode characters
- Maximum value length—256 Unicode characters
- Tag keys and values are case-sensitive. Allowed characters are letters, spaces, and numbers representable in UTF-8, plus the following special characters: + - = . _ : / @. Do not use leading or trailing spaces.
- Do not use the `aws:` prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix.

To manage tags for an existing environment by using the console

1. Open the WorkSpaces Thin Client console.
2. Select the **Environment** to open its details page
3. Choose **Edit**.
4. In **Tags** section, do one or more of the following:.
    - To add a tag, choose **Add new tag** and then edit the values of **Key** and **Value**.
    - To update a tag, edit the value of **Value**.
    - To delete a tag, choose the **Remove** next to the tag.
5. When you are finished updating the tags, choose **Save**.

To manage tags for an existing device by using the console

1. Open the WorkSpaces Thin Client console.
2. Select the device to open its details page.
3. Choose **Tags**.
4. Choose **Manage tags**.
5. Do one or more of the following:
    - To add a tag, choose **Add new tag** and then edit the values of **Key** and **Value**.
    - To update a tag, edit the value of **Value**.
    - To delete a tag, choose the **Remove** next to the tag.
6. When you are finished updating the tags, choose **Save**.

To manage tags for a new device by using the console

1. Open the WorkSpaces Thin Client console.
2. Select the **Environment** to open its details page.
3. Choose **Edit**.
4. In **Device creation tags** section, do one or more of the following:
   - To add a tag, choose **Add new tag** and then edit the values of **Key** and **Value**.
   - To update a tag, edit the value of **Value**.
   - To delete a tag, choose the **Remove** next to the tag.
5. When you are finished updating the tags, choose Save.

When a device is created, it is registered with the environment and the device creation tags are applied. This only happens during new device registration. Additionally, the `aws:thinclient:environment-id` system tag is applied with the environment Id used as value.

To manage tags for a software update by using the console

1. Open the WorkSpaces Thin Client console.
2. Select the **Software update** to open its details page.
3. In **Tags** section, choose **Manage tags**.
4. Do one or more of the following:
   - To add a tag, choose **Add new tag** and then edit the values of **Key** and **Value**.
   - To update a tag, edit the value of **Value**.
   - To delete a tag, choose the **Remove** next to the tag.
5. When you are finished updating the tags, choose **Save**.

Security in Amazon WorkSpaces Thin Client

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 WorkSpaces Thin Client, 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 WorkSpaces Thin Client. The following topics show you how to configure WorkSpaces Thin Client to meet your security and compliance objectives. You can also learn how to use other AWS services that help you to monitor and secure your WorkSpaces Thin Client resources.

Topics

Data protection in Amazon WorkSpaces Thin Client

The AWS shared responsibility model applies to data protection in Amazon WorkSpaces Thin Client. 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.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with WorkSpaces Thin Client 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.

Amazon WorkSpaces Thin Client collects and provide information about user use of WorkSpaces Thin Client devices and their interaction with the virtual desktop services. For example, available memory, network diagnostics, network information, device connectivity, SAML credentials, device identification information, and crash reports. This information is used to provide you the service and may be used to improve ythe user experience with the service. Further, solely to provide you with the service, the information may be transferred outside of the AWS Region where users are using the service. We process this information in accordance with the AWS Privacy Notice.

Topics

- Data encryption
- Data encryption at rest for Amazon WorkSpaces Thin Client
- Encryption in transit
- Key management
- Internet work traffic privacy

Data encryption

WorkSpaces Thin Client collects environment and device customization data, such as user settings, device identifiers, identity provider information, and streaming desktop identifiers. WorkSpaces Thin Client also collects session timestamps. Collected data is stored in Amazon DynamoDB and Amazon S3. WorkSpaces Thin Client uses AWS Key Management Service (KMS) for encryption.

To secure your content, follow these guidelines:

- Implement least privilege access and create specific roles to be used for WorkSpaces Thin Client actions.
- Protect data end-to-end by providing a customer-managed key, so WorkSpaces Thin Client can encrypt your data at rest with the keys you supply.
- Be careful with sharing environment activation codes and user credentials:
  - Admins are required to log into the WorkSpaces Thin Client console, and users are required to provide activation codes for WorkSpaces Thin Client setup use credentials to log into the streaming desktop.
  - Anyone with physical access can set up a WorkSpaces Thin Client, but they can't start a session unless they have a valid activation code and user credentials to log in.
- Users can explicitly end their sessions by choosing to lock their screen, reboot, or shut down the device by using the device toolbar. This discards the device session and clears session credentials.

WorkSpaces Thin Client secures content and metadata by default by encrypting all sensitive data with AWS KMS. If there is an error applying existing settings, a user can't access new sessions and devices cannot apply software updates.

Data encryption at rest for Amazon WorkSpaces Thin Client

Amazon WorkSpaces Thin Client provides encryption by default to protect sensitive customer data at rest by using AWS owned encryption keys.

- **AWS owned keys** — Amazon WorkSpaces Thin Client uses these keys by default to automatically encrypt personally identifiable data. You cannot 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 at rest by default helps reduce the operational overhead and complexity involved in protecting sensitive data. At the same time, it enables you to build secure applications that meet strict encryption compliance and regulatory requirements.

While you can't disable this layer of encryption or select an alternate encryption type, you can add a second layer of encryption over the existing AWS owned encryption keys by choosing a customer managed key when you create your Thin Client Environment:

- **Customer managed keys** — Amazon WorkSpaces Thin Client supports the use of a symmetric customer managed key that you create, own, and manage to add a second layer of encryption on the existing AWS owned encryption. Because you have full control of this layer of encryption, you can perform such tasks as the following:
  - Establishing and maintaining key policies
  - Establishing and maintaining IAM policies and grants
  - Enabling and disabling key policies
  - Rotating key cryptographic material
  - Adding tags
  - Creating key aliases
  - Scheduling keys for deletion

For more information, see customer managed key in the AWS Key Management Service Developer Guide.

The following table summarizes how Amazon WorkSpaces Thin Client encrypts personally identifiable data.

Data type	AWS owned key encryption	Customer managed key encryption (Optional)
Environment Name	Enabled	Enabled

Data type	AWS owned key encryption	Customer managed key encryption (Optional)
WorkSpaces Thin Client Environment name		
Device Name WorkSpaces Thin Client Device name	Enabled	Enabled
Device creation tags WorkSpaces Thin Client Environment device creation tags	Enabled	Enabled

> **Note**
>
> Amazon WorkSpaces Thin Client automatically enables encryption at rest by using AWS owned keys to protect personally identifiable data at no charge.
> However, AWS KMS charges apply for using a customer managed key. For more information about pricing, see the AWS Key Management Service pricing.

How Amazon WorkSpaces Thin Client uses grants in AWS KMS

Amazon WorkSpaces Thin Client requires a grant for you to use your customer managed key.

When you create a WorkSpaces Thin Client Environment encrypted with a customer managed key, Amazon WorkSpaces Thin Client creates a grant on your behalf by sending a CreateGrant request to AWS KMS. Grants in AWS KMS are used to give Amazon WorkSpaces Thin Client access to a KMS key in a customer account.

When a new Thin Client Device is registered to a WorkSpaces Thin Client encrypted Environment with a customer managed key, and the name of that device is changed, Amazon WorkSpaces Thin Client creates a grant on your behalf by sending a CreateGrant request to AWS KMS. Grants in AWS KMS are used to give Amazon WorkSpaces Thin Client access to a KMS key in a customer account.

Amazon WorkSpaces Thin Client requires the grant to use your customer managed key for the following internal operations:

- Send Decrypt requests to AWS KMS to decrypt the encrypted data

You can revoke access to the grant, or you can remove the service's access to the customer managed key at any time. If you do, Amazon WorkSpaces Thin Client won't be able to access any of the data encrypted by the customer managed key, which affects operations that are dependent on that data. For example, if you attempt to get environment details that Amazon WorkSpaces Thin Client can't access, then the operation returns an `AccessDeniedException` error. Additionally, the WorkSpaces Thin Client device will not be able to use a WorkSpaces Thin Client Environment.

Create a customer managed key

You can create a symmetric customer managed key by using the AWS Management Console or the AWS KMS API operations.

To create a symmetric customer managed key

Follow the steps for Creating symmetric customer managed key in the AWS Key Management Service Developer Guide.

Key policy

Key policies control access to your customer managed key. Every customer managed key must have exactly one key policy, which contains statements that determine who can use the key and how they can use it. When you create your customer managed key, you can specify a key policy. For more information, see Managing access to customer managed keys in the AWS Key Management Service Developer Guide.

To use your customer managed key with your Amazon WorkSpaces Thin Client resources, the following API operations must be permitted in the key policy:

- `kms:DescribeKey` — Provides the customer managed key details so Amazon WorkSpaces Thin Client can validate the key.
- `kms:GenerateDataKey` — Allows using the customer managed key to encrypt the data.
- `kms:Decrypt` — Allows using the customer managed key to decrypt the data.
- `kms:CreateGrant` — Adds a grant to a customer managed key. Grants control access to a specified KMS key, which allows access to the grant operations that Amazon WorkSpaces Thin

Client requires. For more information about Using Grants, see the AWS Key Management Service Developer Guide.

This allows Amazon WorkSpaces Thin Client to do the following:

- Call `Decrypt` to decrypt the encrypted data.

The following are policy statement examples you can add for Amazon WorkSpaces Thin Client:

```
{
    "Statement": [
        {
            "Sid": "Allow access to principals authorized to use Amazon WorkSpaces Thin
 Client",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Decrypt",
                "kms:CreateGrant"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "thinclient.region.amazonaws.com",
                    "kms:CallerAccount": "111122223333"
                }
            }
        },
        {
            "Sid": "Allow access for key administrators",
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
            "Action": ["kms:*"],
            "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
        },
        {
            "Sid": "Allow read-only access to key metadata to the account",
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
            "Action": [
                "kms:Describe*",
                "kms:Get*",
                "kms:List*",
                "kms:RevokeGrant"
            ],
            "Resource": "*"
        }
    ]
}
```

For more information about specifying permissions in a policy, see the AWS Key Management Service Developer Guide.

For more information about troubleshooting key access, see the AWS Key Management Service Developer Guide.

Specifying a customer managed key for WorkSpaces Thin Client

You can specify a customer managed key as a second layer encryption for the following resources:

- WorkSpaces Thin Client Environment

When you create an Environment, you can specify the data key by providing a `kmsKeyArn`, which Amazon WorkSpaces Thin Client uses to encrypt the identifiable personal data.

- `kmsKeyArn` — A key identifier for an AWS KMS customer managed key. Provide a key ARN.

When a new WorkSpaces Thin Client device is added to the WorkSpaces Thin Client Environment encrypted with a customer managed key, the WorkSpaces Thin Client Device inherits the customer managed key setting from the WorkSpaces Thin Client Environment.

An encryption context is an optional set of key-value pairs that contains additional contextual information about the data.

AWS KMS uses the encryption context as additional authenticated data to support authenticated encryption. When you include an encryption context in a request to encrypt data, AWS KMS binds the encryption context to the encrypted data. To decrypt data, include the same encryption context in the request.

Amazon WorkSpaces Thin Client encryption context

Amazon WorkSpaces Thin Client uses the same encryption context in all AWS KMS cryptographic operations, where the key is `aws:thinclient:arn` and the value is the Amazon Resource Name (ARN).

The following is the Environment encryption context:

```
"encryptionContext": {
    "aws:thinclient:arn": "arn:aws:thinclient:region:111122223333:environment/
environment_ID"
}
```

The following is the Device encryption context:

```
"encryptionContext": {
    "aws:thinclient:arn": "arn:aws:thinclient:region:111122223333:device/device_ID"
}
```

Using encryption context for monitoring

When you use a symmetric customer managed key to encrypt your WorkSpaces Thin Client Environment and Device data, you can also use the encryption context in audit records and logs to identify how the customer managed key is being used. The encryption context also appears in logs generated by AWS CloudTrail or Amazon CloudWatch Logs.

Using encryption context to control access to your customer managed key

You can use the encryption context in key policies and IAM policies as conditions to control access to your symmetric customer managed key. You can also use encryption context constraints in a grant.

Amazon WorkSpaces Thin Client uses an encryption context constraint in grants to control access to the customer managed key in your account or Region. The grant constraint requires that the operations that the grant allows use the specified encryption context.

The following are example key policy statements to grant access to a customer managed key for a specific encryption context. The condition in this policy statement requires that the `kms:Decrypt` call has an encryption context constraint that specifies the encryption context.

```
{
    "Sid": "Enable Decrypt to access Thin Client Environment",
    "Effect": "Allow",
    "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"},
    "Action": "kms:Decrypt",
    "Resource": "*",
    "Condition": {
        "StringEquals": {"kms:EncryptionContext:aws:thinclient:arn":
 "arn:aws:thinclient:region:111122223333:environment/environment_ID"}
    }
}
```

Monitoring your encryption keys for Amazon WorkSpaces Thin Client

When you use an AWS KMS customer managed key with your Amazon WorkSpaces Thin Client resources, you can use AWS CloudTrail or Amazon CloudWatch Logs to track requests that Amazon WorkSpaces Thin Client sends to AWS KMS.

The following examples are AWS CloudTrail events for `DescribeKey`, `CreateGrant`, `GenerateDataKey`, `Decrypt`, `Decrypt` (using `Grant`) to monitor KMS operations called by Amazon WorkSpaces Thin Client to access data encrypted by your customer managed key:

In the following examples, you can see `encryptionContext` for the WorkSpaces Thin Client Environment. Similar CloudTrail events are recorded for the WorkSpaces Thin Client Device.

DescribeKey

Amazon WorkSpaces Thin Client uses the `DescribeKey` operation to verify the AWS KMS customer managed key.

The following example event records the `DescribeKey` operation:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-11-21T13:43:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "thinclient.amazonaws.com"
    },
    "eventTime": "2023-11-21T13:44:22Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "thinclient.amazonaws.com",
    "userAgent": "thinclient.amazonaws.com",
    "requestParameters": {"keyId": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"},
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

CreateGrant

Amazon WorkSpaces Thin Client uses the `CreateGrant` operation to create a KMS Grant, which allows you to Decrypt data when the Device is accessing it.

The following example event records the `CreateGrant` operation:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-11-21T13:43:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "thinclient.amazonaws.com"
    },
    "eventTime": "2023-11-21T13:44:23Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "thinclient.amazonaws.com",
    "userAgent": "thinclient.amazonaws.com",
    "requestParameters": {
        "granteePrincipal": "thinclient.eu-west-1.amazonaws.com",
        "operations": ["Decrypt"],
        "retiringPrincipal": "thinclient.eu-west-1.amazonaws.com",
        "constraints": {
            "encryptionContextSubset": {"aws:thinclient:arn":
 "arn:aws:thinclient:eu-west-1:111122223333:environment/abcSAMPLE"}
        },
        "keyId": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

GenerateDataKey

Amazon WorkSpaces Thin Client uses the `GenerateDataKey` operation to encrypt data.

The following example event records the `GenerateDataKey` operation:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-03-12T12:21:03Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "thinclient.amazonaws.com"
    },
    "eventTime": "2024-03-12T13:03:56Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "thinclient.amazonaws.com",
    "userAgent": "thinclient.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "encryptionContext": {
            "aws-crypto-public-key": "ABC123def4567890abc12345678/90dE/F123abcDEF
+4567890abc123D+ef1==",
            "aws:thinclient:arn": "arn:aws:thinclient:eu-
west-1:111122223333:environment/abcSAMPLE"
        },
        "numberOfBytes": 32
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Decrypt

Amazon WorkSpaces Thin Client uses the `Decrypt` operation to decrypt data.

The following example event records the `Decrypt` operation:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-11-21T13:43:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "thinclient.amazonaws.com"
    },
    "eventTime": "2023-11-21T13:44:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "thinclient.amazonaws.com",
    "userAgent": "thinclient.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "encryptionContext": {
            "aws-crypto-public-key": "ABC123def4567890abc12345678/90dE/F123abcDEF
+4567890abc123D+ef1==",
            "aws:thinclient:arn": "arn:aws:thinclient:eu-
west-1:111122223333:environment/abcSAMPLE"
```

```
         },
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Decrypt (using Grant)

When WorkSpaces Thin Client Device accesses Environment or Device information, the `Decrypt` operation is used, which is allowed through a KMS key `Grant`.

The following example event records the `Decrypt` operation, authorized through a `Grant`:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "thinclient.amazonaws.com"
    },
    "eventTime": "2023-11-21T13:44:23Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "thinclient.amazonaws.com",
    "userAgent": "thinclient.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws-crypto-public-key": "ABC123def4567890abc12345678/90dE/F123abcDEF
+4567890abc123D+ef1==",
            "aws:thinclient:arn": "arn:aws:thinclient:eu-
west-1:111122223333:environment/abcSAMPLE"
        },
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventCategory": "Management"
}
```

Learn More

The following resources provide more information about data encryption at rest:

- For more information about AWS Key Management Service basic concepts, see the AWS Key Management Service Developer Guide.
- For more information about Security best practices for AWS Key Management Service, see theAWS Key Management Service Developer Guide.

Encryption in transit

WorkSpaces Thin Client encrypts data in transit over HTTPS and TLS 1.2. You can send a request to WorkSpaces Thin Client by using the console or direct API calls. The request data that is transferred is encrypted by sending it through an HTTPS or TLS connection. Request data can be transferred from the AWS Console, AWS Command Line Interface, or AWS SDK to WorkSpaces Thin Client. This also includes any software updates on the device.

Encryption in transit is configured by default, and secure connections (HTTPS, TLS) are configured by default.

Key management

You can supply your own Customer Managed AWS KMS Key to encrypt your customer information. If you don't supply a key, WorkSpaces Thin Client uses an AWS Owned Key. You can set your key by using the AWS SDK.

Internet work traffic privacy

Administrators are able to view WorkSpaces Thin Client session events, including start times and pending software update information. These logs are encrypted and securely delivered to customers in the WorkSpaces Thin Client console. User information and further details about individual streaming desktop sessions is recorded by the desktop services. For more information, see Monitor your WorkSpaces, Monitoring and Reporting for AppStream 2.0, or User access logging for WorkSpaces Web.

Identity and access management for Amazon WorkSpaces Thin Client

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 WorkSpaces Thin Client resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Troubleshooting Amazon WorkSpaces Thin Client identity and access

Audience

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

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

Service administrator – If you're in charge of WorkSpaces Thin Client resources at your company, you probably have full access to WorkSpaces Thin Client. It's your job to determine which WorkSpaces Thin Client 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 WorkSpaces Thin Client, see How Amazon WorkSpaces Thin Client 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 WorkSpaces Thin Client. To view example WorkSpaces Thin Client identity-based policies that you can use in IAM, see Identity-based policy examples for Amazon WorkSpaces Thin Client.

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 Signing AWS 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 Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

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

IAM roles

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

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

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center 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 Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role –** A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your 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 Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

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

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

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 Creating IAM policies in the *IAM User Guide*.

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

Resource-based policies

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

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

Access control lists (ACLs)

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

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

Other policy types

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

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If

you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.

- **Session policies –** Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

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

How Amazon WorkSpaces Thin Client works with IAM

Before you use IAM to manage access to WorkSpaces Thin Client, learn what IAM features are available to use with WorkSpaces Thin Client.

IAM features you can use with Amazon WorkSpaces Thin Client

IAM feature	WorkSpaces Thin Client 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

IAM feature	WorkSpaces Thin Client support
Principal permissions	Yes
Service roles	No
Service-linked roles	No

To get a high-level view of how WorkSpaces Thin Client 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 WorkSpaces Thin Client

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 Creating IAM 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 WorkSpaces Thin Client

To view examples of WorkSpaces Thin Client identity-based policies, see Identity-based policy examples for Amazon WorkSpaces Thin Client.

Resource-based policies within WorkSpaces Thin Client

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 WorkSpaces Thin Client

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 WorkSpaces Thin Client actions, see Actions Defined by Amazon WorkSpaces Thin Client in the *Service Authorization Reference*.

Policy actions in WorkSpaces Thin Client use the following prefix before the action:

```
workspaces-thin-client
```

To specify multiple actions in a single statement, separate them with commas, as shown in the following example:

```
"Action": [
      "workspaces-thin-client:action1",
      "workspaces-thin-client:action2"
         ]
```

To view examples of WorkSpaces Thin Client identity-based policies, see Identity-based policy examples for Amazon WorkSpaces Thin Client.

Policy resources for WorkSpaces Thin Client

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 WorkSpaces Thin Client resource types and their ARNs, see Resources Defined by Amazon WorkSpaces Thin Client in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by Amazon WorkSpaces Thin Client.

To view examples of WorkSpaces Thin Client identity-based policies, see Identity-based policy examples for Amazon WorkSpaces Thin Client.

Policy condition keys for WorkSpaces Thin Client

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 WorkSpaces Thin Client condition keys, see Condition Keys for Amazon WorkSpaces Thin Client in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions Defined by Amazon WorkSpaces Thin Client.

To view examples of WorkSpaces Thin Client identity-based policies, see Identity-based policy examples for Amazon WorkSpaces Thin Client.

ACLs in WorkSpaces Thin Client

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 WorkSpaces Thin Client

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 What is ABAC? 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 WorkSpaces Thin Client

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 Switching to a 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 WorkSpaces Thin Client

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 WorkSpaces Thin Client

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 Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

> ⚠ **Warning**
>
> Changing the permissions for a service role might disrupt WorkSpaces Thin Client functionality. Edit service roles only when WorkSpaces Thin Client provides guidance to do so.

Service-linked roles for WorkSpaces Thin Client

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 WorkSpaces Thin Client

By default, users and roles don't have permission to create or modify WorkSpaces Thin Client 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 Creating IAM policies in the *IAM User Guide*.

For details about actions and resource types defined by WorkSpaces Thin Client, including the format of the ARNs for each of the resource types, see Actions, Resources, and Condition Keys for Amazon WorkSpaces Thin Client in the *Service Authorization Reference*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete WorkSpaces Thin Client 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 IAM Access Analyzer policy validation 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 Configuring MFA-protected API access 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 WorkSpaces Thin Client console

To access the Amazon WorkSpaces Thin Client console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the WorkSpaces Thin Client 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.

Grant read-only access to WorkSpaces Thin Client

This example shows how you can create a policy that allows IAM users to view a WorkSpaces Thin Client configuration, but not make changes. This policy includes permissions to complete this action on the console or program by using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "thinclient:GetEnvironment",
                "thinclient:ListEnvironments",
                "thinclient:GetDevice",
                "thinclient:ListDevices",
                "thinclient:ListDeviceSessions",
                "thinclient:GetSoftwareSet",
                "thinclient:ListSoftwareSets",
                "thinclient:ListTagsForResource"
            ],
            "Resource": "arn:aws:thinclient:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": ["workspaces:DescribeWorkspaceDirectories"],
            "Resource": "arn:aws:workspaces:*:*:directory/*"
        },
        {
            "Effect": "Allow",
            "Action": ["workspaces-web:GetPortal"],
            "Resource": ["arn:aws:workspaces-web:*:*:portal/*"]
        },
        {
            "Effect": "Allow",
            "Action": ["workspaces-web:GetUserSettings"],
            "Resource": ["arn:aws:workspaces-web:*:*:userSettings/*"]
        },
        {
            "Effect": "Allow",
            "Action": ["appstream:DescribeStacks"],
            "Resource": ["arn:aws:appstream:*:*:stack/*"]
        }
    ]
}
```

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": "*"
        }
    ]
}
```

Grant full access to WorkSpaces Thin Client

This example shows how you can create a policy that grants full access to WorkSpaces Thin Client IAM users. This policy includes permissions to complete all WorkSpaces Thin Client actions on the console or program by using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["thinclient:*"],
            "Resource": "arn:aws:thinclient:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": ["workspaces:DescribeWorkspaceDirectories"],
            "Resource": "arn:aws:workspaces:*:*:directory/*"
        },
        {
            "Effect": "Allow",
            "Action": ["workspaces-web:GetPortal"],
            "Resource": ["arn:aws:workspaces-web:*:*:portal/*"]
        },
        {
            "Effect": "Allow",
            "Action": ["workspaces-web:GetUserSettings"],
            "Resource": ["arn:aws:workspaces-web:*:*:userSettings/*"]
        },
        {
            "Effect": "Allow",
            "Action": ["appstream:DescribeStacks"],
            "Resource": ["arn:aws:appstream:*:*:stack/*"]
        }
    ]
}
```

Troubleshooting Amazon WorkSpaces Thin Client identity and access

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

Topics

- I am not authorized to perform an action in WorkSpaces Thin Client
- I want to view my access keys
- I'm an administrator and want to allow others to access WorkSpaces Thin Client
- I want to allow people outside of my AWS account to access my WorkSpaces Thin Client resources

I am not authorized to perform an action in WorkSpaces Thin Client

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

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *my-thin-client-device* resource but does not have the fictional `workspaces-thin-client:`*ListDevices* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 workspaces-thin-client:ListDevices on resource: my-thin-client-device
```

In this case, Mateo asks his administrator to update his policies to allow him to access the *my-thin-client-device* resource by using the `workspaces-thin-client:`*ListDevices* action.

I want to view my access keys

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

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

⚠ **Important**

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

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

I'm an administrator and want to allow others to access WorkSpaces Thin Client

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

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

For more information, see Grant full access to WorkSpaces Thin Client.

I want to allow people outside of my AWS account to access my WorkSpaces Thin Client 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 WorkSpaces Thin Client supports these features, see How Amazon WorkSpaces Thin Client 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*.

Resilience in Amazon WorkSpaces Thin Client

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

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

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

Vulnerability analysis and management in Amazon WorkSpaces Thin Client

Configuration and IT controls are a shared responsibility between AWS and you. For more information, see the AWS shared responsibility model.

Amazon WorkSpaces Thin Client cross-integrates with Amazon WorkSpaces, Amazon AppStream 2.0, and WorkSpaces Web. See the following links for more information about update management for each of these services:

- Update Management in Amazon AppStream 2.0
- Update management in Amazon WorkSpaces
- Configuration and vulnerability analysis in Amazon WorkSpaces Web

Monitoring Amazon WorkSpaces Thin Client

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

- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to the Amazon S3 bucket that you specify. You can identify users and accounts that 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.

Logging Amazon WorkSpaces Thin Client API calls by using AWS CloudTrail

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

WorkSpaces Thin Client information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in WorkSpaces Thin Client, 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 WorkSpaces Thin Client, 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 WorkSpaces Thin Client actions are logged by CloudTrail and are documented in the Amazon WorkSpaces Thin Client API Reference. For example, calls to the `CreateEnvironment`, `ListDevices`, and `GetSoftwareSet` 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 WorkSpaces Thin Client 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 `GetDevice` action.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "<principal-id>",
        "arn": "<arn>",
        "accountId": "<account-id>",
        "accessKeyId": "<access-key-id>",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "<principal-id>",
                "arn": "arn:aws:iam::<arn>",
                "accountId": "<accpimt-id>",
                "userName": "<user-name>"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-11-18T23:07:01Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-11-18T23:11:57Z",
    "eventSource": "thinclient.amazonaws.com",
    "eventName": "GetDevice",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "<source-ip-address>",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0)
 Gecko/20100101 Firefox/115.0",
    "requestParameters": {
        "id": "<ip>"
    },
    "responseElements": null,
    "requestID": "<request-id>",
    "eventID": "<event-id>",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "<recipient-account-id>",
    "eventCategory": "Management"
}
```

Creating Amazon WorkSpaces Thin Client resources with AWS CloudFormation

Amazon WorkSpaces Thin Client is integrated with AWS CloudFormation, a service that helps you model and set up your AWS resources. This way, you can spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want (such as Environments), and AWS CloudFormation provisions and configures those resources for you.

When you use AWS CloudFormation, you can reuse your template to set up your WorkSpaces Thin Client resources consistently and repeatedly. Describe your resources once, and then provision the same resources repeatedly in multiple AWS accounts and Regions.

WorkSpaces Thin Client and AWS CloudFormation templates

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

WorkSpaces Thin Client supports creating Environments in AWS CloudFormation. For more information, including examples of JSON and YAML templates for Environments, see the Amazon WorkSpaces Thin Client resource type reference in the *AWS CloudFormation User Guide*.

Learn more about AWS CloudFormation

To learn more about AWS CloudFormation, see the following resources:

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

Access Amazon WorkSpaces Thin Client by using an interface endpoint (AWS PrivateLink)

You can use AWS PrivateLink to create a private connection between your VPC and Amazon WorkSpaces Thin Client. You can access WorkSpaces Thin Client as a VPC, without the use of an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't require public IP addresses to access WorkSpaces Thin Client.

You establish this private connection by creating an *interface endpoint* that's powered by AWS PrivateLink. We create an endpoint network interface in each subnet that you enable for the interface endpoint. These are requester-managed network interfaces that serve as the entry point for traffic destined for WorkSpaces Thin Client.

For more information, see Access AWS services through AWS PrivateLink in the *AWS PrivateLink Guide*.

Considerations for WorkSpaces Thin Client

Before you set up an interface endpoint for WorkSpaces Thin Client, review Considerations in the *AWS PrivateLink Guide*.

WorkSpaces Thin Client supports making calls to all of its API actions through the interface endpoint.

Create an interface endpoint for WorkSpaces Thin Client

You can create an interface endpoint for WorkSpaces Thin Client by using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Create an interface endpoint in the *AWS PrivateLink Guide*.

Create an interface endpoint for WorkSpaces Thin Client by using the following service name:

```
com.amazonaws.region.thinclient.api
```

If you enable private DNS for the interface endpoint, you can make API requests to WorkSpaces Thin Client by using its default Regional DNS name. For example, `api.thinclient.us-east-1.amazonaws.com`.

Create an endpoint policy for your interface endpoint

An endpoint policy is an IAM resource that you can attach to an interface endpoint. The default endpoint policy gives you full access to WorkSpaces Thin Client through the interface endpoint. To control the access granted to WorkSpaces Thin Client from your VPC, attach a custom endpoint policy to the interface endpoint.

An endpoint policy specifies the following information:

- The principals that can perform actions (AWS accounts, IAM users, and IAM roles).
- The actions that can be performed.
- The resources on which the actions can be performed.

For more information, see Control access to services using endpoint policies in the *AWS PrivateLink Guide*.

Example: VPC endpoint policy for WorkSpaces Thin Client actions

The following is an example of a custom endpoint policy. When you attach this policy to your interface endpoint, it grants access to the listed WorkSpaces Thin Client actions for all principals on all resources.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
            "thinclient:ListEnvironments",
            "thinclient:ListDevices",
            "thinclient:ListSoftwareSets"
         ],
         "Resource":"*"
      }
   ]
}
```

Document history for the WorkSpaces Thin Client Administrator Guide

The following table describes the documentation history for releases of the WorkSpaces Thin Client Administrator Guide.

Change	Description	Date
• Configuring WorkSpaces for Amazon WorkSpaces Thin Client • Configuring AppStream 2.0 for Amazon WorkSpaces Thin Client	• Updated the operating system list. • Updated the Identity Provider procedure.	February 12, 2024
Initial release	Initial release	November 26, 2023