Amazon WorkSpaces Web

Administration Guide

Amazon WorkSpaces Web: Administration Guide

Table of Contents

What is Amazon WorkSpaces Web?

WorkSpaces Web is a low cost, fully managed WorkSpace built specifically to facilitate secure, web-based workloads. WorkSpaces Web makes it easy for customers to safely provide their employees with access to internal websites and SaaS web applications without the administrative burden of appliances or specialized client software. WorkSpaces Web provides simple policy tools tailored for user interactions, while offloading common tasks like capacity management, scaling, and maintaining browser images.

WorkSpaces Web delivers a simple administration and commitment-free solution to support workers that only need access to internal and SaaS web applications. WorkSpaces Web pixel streams web content from AWS, so sensitive company web data never resides on remote devices reducing the risk of data exfiltration. Streaming also provides a barrier between internal servers and local devices, preventing the transmission of device-borne malware to internal servers. WorkSpaces Web applies browser policies on your or the customer's behalf to isolate users to the web browser interface. As a result, they are unable to install applications from the internet or access the terminal or operating system menus during a session.

WorkSpaces Web is automatically managed, with capacity, scaling and browser images updated to the latest version of Chrome by AWS. Each WorkSpaces Web session starts with a fresh and fully updated Chrome browser, with your enterprise browser policy applied. At the end of the session, the instance is terminated, so company data never resides on remote devices. WorkSpaces Web provides customers with simple tools to customize the browser experience, like setting a startURL or bookmarks, and allows them to apply polices for use of the clipboard, printer, and file transfer. WorkSpaces Web also supports full Chrome Enterprise Policy on Linux, so customers can make use of over 300 user and device browser policies.

Getting started with WorkSpaces Web takes two steps. First, administrators create an WorkSpaces Web Portal from the WorkSpaces Web console. Second, administrators distribute the endpoint URL so users can access their streaming browser, either by adding to an existing SAML2.0 application gateway or by emailing the URL to users. Then, users access the endpoint from their existing browser, sign in with their SAML credentials, and start their session from the startup URL set by the administrator. Administrators are in full control of what content users may browse to. They can set URL allowlist and denylist policies with Chrome Policy, or filter browser traffic through their VPC. WorkSpaces Web works with SAML2.0 identity providers (like Okta and Ping) and honors existing enforcement policies. WorkSpaces Web only charges customers monthly for users that sign in to the streaming web browser, and requires no up-front costs, licenses, or ongoing agreements.

Topics

Terms to know when using WorkSpaces Web

To help you get started with WorkSpaces Web, you should get familiar with the following concepts.

Identity provider (IdP)

An identity provider verifies your users' credentials. It then issues authentication assertions to provide access to a service provider. You can configure your existing IdP to work with WorkSpaces Web.

The process for configuring your identity provider (IdP) varies, depending on your IdP.

You must upload the service provider metadata file to your IdP. Otherwise, your users won't be able to log in. You must also grant access for your users to use WorkSpaces Web in your IdP.

Identity provider (IdP) metadata document

WorkSpaces Web requires specific metadata from your identity provider (IdP) to establish trust. You can add this metadata to WorkSpaces Web by uploading a metadata exchange file downloaded from your IdP.

Service provider (SP)

A service provider accepts authentication assertions and provides a service to the user. WorkSpaces Web acts as a service provider to users who have been authenticated by their IdP.

Service provider (SP) metadata document

You will need to add the service provider metadata details to your identity provider's (IdP's) configuration interface. The details of this configuration process varies between providers.

SAML 2.0

A standard for exchanging authentication and authorization data between an IdP and a service provider.

Virtual Private Cloud (VPC)

You can use an existing or new VPC, corresponding subnets, and security groups to link your content with WorkSpaces Web.

Subnets must with a stable connection to the internet, and the VPC and subnets must also have a stable connection to any internal and Software as a Service (SaaS) websites for users to access these resources.

The VPCs, subnets, and security groups listed are taken from the same region as your WorkSpaces Web console.

Trust store

If a user accessing a web site through WorkSpaces Web receives a privacy error, such as NET::ERR_CERT_INVALID, that site might be using a certificate signed by a private certificate authority (PCA). You may need to add or change the PCAs in your trust store. In addition, if a user's device requires you to install a specific certificate in order to load a web site, you will need to add that certificate to your trust store to allow your user to access that site in WorkSpaces Web.

Publicly accessible web sites usually don't require any changes to a trust store.

Web portal

A web portal provides your users with access to internal and SaaS websites from their browsers. You can create one web portal in any supported region per account. To request a limit increase for more than one portal, contact support.

Web portal endpoint

The web portal endpoint is the access point your users will launch your web portal from after signing in with the identity provider configured for the portal.

The endpoint is publicly available on the internet and can be embedded into your network.

Related services

WorkSpaces Web is a capability from Amazon WorkSpaces in the AWS End User Computing portfolio. Compared with WorkSpaces and AppStream 2.0, WorkSpaces Web is built specifically to facilitate secure, web-based workloads. WorkSpaces Web is automatically managed, with capacity, scaling, and images provisioned and updated on demand by AWS. For example, you can choose to offer a persistent Workspace Desktop to your software developers who need access to desktop resources, and Amazon WorkSpaces Web to the contact center users that only need access to a handful of internal and SaaS websites (including those hosted outside your network) on desktop computers.

Accessing Amazon WorkSpaces Web

Administrators access Amazon WorkSpaces Web through the AWS WorkSpaces Web Console, SDK, CLI, or API. Your users access it through the Amazon WorkSpaces Web endpoint.

Amazon WorkSpaces Web is currently available in in US East (N. Virginia), US West (Oregon), and Europe (Ireland).

Setting up Amazon WorkSpaces Web

Before you can configure Amazon WorkSpaces Web to reach your internal websites and SaaS applications, you must complete the following prerequisites.

Topics

Get an AWS account and your root user credentials

To access AWS, you must sign up for an AWS account.

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.

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

Creating an IAM user

If your account already includes an IAM user with full AWS administrative permissions, you can skip this section.

When you first create an Amazon Web Services (AWS) account, you begin with a single sign-in identity. That identity has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user*. When you sign in, enter the email address and password that you used to create the account.

> **Important**
> We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform

only a few account and service management tasks. To view the tasks that require you to sign in as the root user, see Tasks that require root user credentials.

To create an administrator user for yourself and add the user to an administrators group (console)

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

   Note
   We strongly recommend that you adhere to the best practice of using the **`Administrator`** IAM user that follows and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.
2. In the navigation pane, choose **Users** and then choose **Add users**.
3. For **User name**, enter **`Administrator`**.
4. Select the check box next to **AWS Management Console access**. Then select **Custom password**, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter **`Administrators`**.
10. Choose **Filter policies**, and then select **AWS managed - job function** to filter the table contents.
11. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.

    Note
    You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.
12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the *IAM User Guide*.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access management and Example policies.

Signing in as an IAM user

Sign in to the IAM console by choosing **IAM user** and entering your AWS account ID or account alias. On the next page, enter your IAM user name and your password.

> **Note**
> For your convenience, the AWS sign-in page uses a browser cookie to remember your IAM user name and account information. If you previously signed in as a different user, choose the sign-in

link beneath the button to return to the main sign-in page. From there, you can enter your AWS account ID or account alias to be redirected to the IAM user sign-in page for your account.

Creating IAM user access keys

Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. If you don't have access keys, you can create them from the AWS Management Console. As a best practice, do not use the AWS account root user access keys for any task where it's not required. Instead, create a new administrator IAM user with access keys for yourself.

The only time that you can view or download the secret access key is when you create the keys. You cannot recover them later. However, you can create new access keys at any time. You must also have permissions to perform the required IAM actions. For more information, see Permissions required to access IAM resources in the *IAM User Guide*.

To create access keys for an IAM user

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. Choose the name of the user whose access keys you want to create, and then choose the **Security credentials** tab.
4. In the **Access keys** section, choose **Create access key**.
5. To view the new access key pair, choose **Show**. You will not have access to the secret access key again after this dialog box closes. Your credentials will look something like this:
   - Access key ID: AKIAIOSFODNN7EXAMPLE
   - Secret access key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
6. To download the key pair, choose **Download .csv file**. Store the keys in a secure location. You will not have access to the secret access key again after this dialog box closes.

   Keep the keys confidential in order to protect your AWS account and never email them. Do not share them outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your secret key.
7. After you download the `.csv` file, choose **Close**. When you create an access key, the key pair is active by default, and you can use the pair right away.

Related topics

- What is IAM? in the *IAM User Guide*
- AWS security credentials in *AWS General Reference*

Set up your network

Follow these steps to set up your network.

Topics

- Create and configure a new VPC (p. 7)
- Add a NAT gateway to an existing VPC (p. 11)
- Set up VPC endpoints in your subnets (p. 11)

Review general requirements

During WorkSpaces Web portal creation, you will select a VPC in your account, and choose at least two private subnets in two different Availability Zones that meet following requirements:

- All subnets must be private. This means that for each end user browser session, it will be assigned a private IP address that is not directly accessible from the internet. However, your end users can still have internet access if you complete the steps in Set up internet access in your subnets (p. 7). For private subnet definition, see VPC with public and private subnets (NAT).
- All private subnets must have a stable connection to S3, KMS, and CloudWatch Logs service. You can choose to connect over the internet and/or through a VPC endpoint. For information about how to set up internet access in your private subnets, see Set up internet access in your subnets (p. 7). For information about how to set up required VPC endpoints in your private subnets, see Set up VPC endpoints in your subnets (p. 11).

  For example, you can choose to configure both internet access from your private subnet, so that your end users can browse internet content, while your WorkSpaces Web web portal has private connectivity to S3, KMS, and CloudWatch. WorkSpaces Web will not function as expected without these service connections.
- All private subnets must have a stable connection to any internal content, either located in AWS or on premises that users will access with WorkSpaces Web .

Set up internet access in your subnets

To configure a VPC with private subnets and a NAT gateway

1. Create your WorkSpaces Web portal with private subnets and configure at least one NAT gateway in a public subnet in your VPC.
2. You can choose to create and configure a new VPC to use with a NAT gateway, or add a NAT gateway to an existing VPC. As a result, your WorkSpaces Web web portal will be able to browse the public internet.
3. For availability consideration, WorkSpaces Web requires at least two private subnets created in two different AZs, and your subnets should have sufficient IP space to support expected WorkSpaces Web traffic. You should allocate one IP for the maximum number of concurrent sessions.

Create and configure a new VPC

This section describes how to use the VPC wizard to create a VPC with a public subnet and one private subnet. As part of this process, the wizard creates an internet gateway and a NAT gateway. It also creates a custom route table associated with the public subnet, and updates the main route table associated with the private subnet. The NAT gateway is automatically created in the public subnet of your VPC.

After you use the wizard to create the initial VPC configuration, you'll add a second private subnet. For more information about this configuration, see VPC with public and private subnets (NAT).

> **Note**
> If you already have a VPC, skip this step and proceed to the next step, Add a NAT gateway to an existing VPC (p. 11).

Step 1: Allocate an Elastic IP address

Before you create your VPC, you must allocate an Elastic IP address in your WorkSpaces Web Region. You must first allocate an Elastic IP address for use in your VPC, and then associate it with your NAT gateway. With an Elastic IP address, you can mask the failure of streaming instance by rapidly remapping the address to another streaming instance in your VPC. For more information, see Elastic IP addresses.

> **Note**
> Charges might apply to Elastic IP addresses that you use. For more information, see the Elastic IP addresses pricing page.

If you don't already have an Elastic IP address, complete the following steps. If you want to use an existing Elastic IP address, verify that it's not currently associated with another instance or network interface.

To allocate an Elastic IP address

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Network & Security**, choose **Elastic IPs**.
3. Choose **Allocate New Address**, and then choose **Allocate**.
4. Note the Elastic IP address.
5. In the upper-right corner of the **Elastic IPs** pane, click the **X** icon to close the pane.

Step 2: Create a new VPC

Complete the following steps to create a new VPC with one public subnet and one private subnet.

To create a new VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **VPC Dashboard**.
3. Choose **Launch VPC Wizard**.
4. In **Step 1: Select a VPC Configuration**, choose **VPC with Public and Private Subnets**, and then choose **Select**.
5. In **Step 2: VPC with Public and Private Subnets**, configure the VPC as follows:
   - For **IPv4 CIDR block**, specify an IPv4 CIDR block for the VPC.
   - For **IPv6 CIDR block**, keep the default value, **No IPv6 CIDR Block**.
   - For **VPC name**, type a unique name for the VPC.
   - Configure the public subnet as follows:
     - For **Public subnet's IPv4 CIDR**, specify the CIDR block for the subnet.
     - For **Availability Zone**, keep the default value, **No Preference**.
     - For **Public subnet name**, type a name for the subnet; for example, **`WorkSpaces Web Public Subnet`**.
   - Configure the first private subnet as follows:
     - For **Private subnet's IPv4 CIDR**, specify the CIDR block for the subnet. Make a note of the value that you specify.
     - For **Availability Zone**, select a specific zone and make a note of the zone that you select.
     - For **Private subnet name**, type a name for the subnet; for example, **`WorkSpaces Web Private Subnet1`**.
   - For the remaining fields, where applicable, keep the default values.
   - For **Elastic IP Allocation ID**, click in the text box and select the value that corresponds to the Elastic IP address that you created. This address is assigned to the NAT gateway. If you

don't have an Elastic IP address, create one by using the Amazon VPC console at https://console.aws.amazon.com/vpc/.

- For **Service endpoints**, if an Amazon S3 endpoint is required for your environment, specify one. An S3 endpoint is required to provide users with access to home folders or to enable application settings persistence for your users in a private network. For more information, see Enable and Administer Home Folders for Your AppStream 2.0 Users and Enable Application Settings Persistence for Your AppStream 2.0 Users.

  To specify an Amazon S3 endpoint, do the following:

  1. Choose **Add Endpoint**.
  2. For **Service**, select the entry in the list that ends with "s3" (the com.amazonaws.region.s3 entry that corresponds to the Region in which the VPC is being created).
  3. For **Subnet**, choose **Private subnet**.
  4. For **Policy**, keep the default value, **Full Access**.
- For **Enable DNS hostnames**, keep the default value, **Yes**.
- For **Hardware tenancy**, keep the default value, **Default**.
- Choose **Create VPC**.
- It takes several minutes to set up your VPC. After the VPC is created, choose **OK**.

Step 3: Add a second private subnet

In the previous step, you created a VPC with one public subnet and one private subnet. Complete the following steps to add a second private subnet. We recommend that you add a second private subnet in a different Availability Zone than your first private subnet.

To add a second private subnet

1. In the navigation pane, choose **Subnets**.
2. Select the first private subnet that you created in the previous step. On the **Description** tab, below the list of subnets, make a note of the Availability Zone for this subnet.
3. On the upper left of the subnets pane, choose **Create Subnet**.
4. For **Name tag**, type a name for the private subnet; for example, **`WorkSpaces Web Private Subnet2`**.
5. For **VPC**, select the VPC that you created in the previous step.
6. For **Availability Zone**, select an Availability Zone other than the one you are using for your first private subnet. Selecting a different Availability Zone increases fault tolerance and helps prevent insufficient capacity errors.
7. For **IPv4 CIDR block**, specify a unique CIDR block range for the new subnet. For example, if your first private subnet has an IPv4 CIDR block range of **`10.0.1.0/24`**, you could specify a CIDR block range of **`10.0.2.0/24`** for the new private subnet.
8. Choose **Create**.
9. After your subnet is created, choose **Close**.

Step 4: Verify and name your subnet route tables

After you've created and configured your VPC, complete the following steps to specify a name for your route tables, and to verify the following:

- The route table associated with the subnet in which your NAT gateway resides includes a route that points internet traffic to an internet gateway. This ensures that your NAT gateway can access the internet.

- The route tables associated with your private subnets are configured to point internet traffic to the NAT gateway. This enables the streaming instances in your private subnets to communicate with the internet.

To verify and name your subnet route tables

1. In the navigation pane, choose **Subnets**, and select the public subnet that you created; for example, **WorkSpaces Web 2.0 Public Subnet**.
2. On the **Route Table** tab, choose the ID of the route table; for example, **rtb-12345678**.
3. Select the route table. Under **Name**, choose the edit icon (the pencil), and type a name (for example, `workspacesweb-public-routetable`), and then select the check mark to save the name.
4. With the public route table still selected, on the **Routes** tab, verify that there is one route for local traffic, and another route that sends all other traffic to the internet gateway for the VPC. The following table describes these two routes:

Destination	Target	Description
Public subnet IPv4 CIDR block (for example, 10.0.0/20)	Local	All traffic from the resources destined for IPv4 addresses within the public subnet IPv4 CIDR block is routed locally within the VPC.
Traffic destined to all other IPv4 addresses (for example, 0.0.0.0/0)	Outbound (igw-ID)	Traffic destined for all other IPv4 addresses is routed to the internet gateway (identified by igw-ID) that was created by the VPC wizard.

5. In the navigation pane, choose **Subnets**, and select the first private subnet that you created (for example, `WorkSpaces Web Private Subnet1`).
6. On the **Route Table** tab, choose the ID of the route table.
7. Select the route table. Under **Name**, choose the edit icon (the pencil), and enter a name (for example, `workspacesweb-private-routetable`), and then choose the check mark to save the name.
8. On the **Routes** tab, verify that the route table includes the following routes:

Destination	Target	Description
Public subnet IPv4 CIDR block (for example, 10.0.0/20)	Local	All traffic from the resources destined for IPv4 addresses within the public subnet IPv4 CIDR block is routed locally within the VPC.
Traffic destined to all other IPv4 addresses (for example, 0.0.0.0/0)	Outbound (nat-ID)	Traffic destined for all other IPv4 addresses is routed to the NAT gateway (identified by nat-ID).
Traffic destined for S3 buckets (applicable if you specified an S3 endpoint) [pl-ID (com.amazonaws.region.s3)]	Storage (vpce-ID)	Traffic destined for S3 buckets is routed to the S3 endpoint (identified by vpce-ID).

9. In the navigation pane, choose **Subnets**, and select the second private subnet that you created (for example, `WorkSpaces Web Private Subnet2`).
10. On the **Route Table** tab, verify that the route table is the private route table (for example, `workspacesweb-private-routetable`). If the route table is different, choose **Edit** and select this route table.

Add a NAT gateway to an existing VPC

If you have already configured a VPC, complete the following steps to add a NAT gateway to your VPC. If you need to create a new VPC, see Create and configure a new VPC (p. 7).

To add a NAT gateway to an existing VPC

1. To create your NAT gateway, complete the steps in Create a NAT gateway.
2. You must specify two private subnets from different Availability Zones for high availability and fault tolerance. For information about how to create a second private subnet, see Step 3: Add a Second Private Subnet.
3. Update the route table associated with one or more of your private subnets to point internet-bound traffic to the NAT gateway. This enables the streaming instances in your private subnets to communicate with the internet. To do this, complete the steps in Work with route tables.

Set up VPC endpoints in your subnets

WorkSpaces Web needs a connection with Amazon S3, KMS, and CloudWatch to provide basic functionality. If you choose to not enable a connection to these servcies over the internet from your VPC, you will need to ensure your VPC has a private connection.

Step 1: Configure the Amazon S3 VPC Endpoint

WorkSpaces Web requires an Amazon S3 connection to download customized browser settings before launching your session.

To create the Amazon S3 Gateway VPC Endpoint in your VPC

1. In the AWS VPC console, choose **Endpoints**.
2. Choose **Create Endpoint**, then select the **com.amazonaws.<Region>.s3** service with the gateway type.
3. For **VPC**, choose the VPC where you want to launch WorkSpaces Web.
4. For **Configure route tables**, choose all the route tables that are associated with the subnets that you selected during WorkSpaces Web portal creation.
5. Choose **Create Endpoint**.

Step 2: Configure the AWS Key Management Service VPC endpoint

With WorkSpaces Web, all customer settings are encrypted by default. You can choose whether to use AWS Key Management Service to decrypt those encrypted customer settings and apply them before launching a session.

To create the AWS Key Management Service Interface VPC endpoint in your VPC

1. In the AWS VPC console, choose **Endpoints**.
2. Choose **Create Endpoint**, then select the **com.amazonaws.<Region>.kms** service with the Interface type.
3. For **VPC**, choose the VPC where you want to launch WorkSpaces Web.
4. For **Subnets**, choose all the subnets that you selected during WorkSpaces Web portal creation.
5. Choose **Create Endpoint**.

   Note
   Using Interface VPCE will result in additional charges. For more information, see AWS PrivateLink pricing.

Step 3: Configure the Amazon CloudWatch Logs VPC endpoint

WorkSpaces Web requires the Amazon CloudWatch Logs service connection to collect log data to help improve the user experience.

To create the Amazon CloudWatch Logs Interface VPC endpoint in your VPC

1. In the AWS VPC console, choose **Endpoints**.
2. Choose **Create Endpoint**, then select the **com.amazonaws.<Region>.logs** service with the Interface type.
3. For **VPC**, choose the VPC where you want to launch WorkSpaces Web.
4. For **Subnets**, choose all the subnets that you selected during WorkSpaces Web portal creation.
5. Choose **Create Endpoint**.

Review browser policies

To provide a more secure browsing experience for your users, WorkSpaces Web applies several browser policies in addition to the ones that you specify. The following is the list of policies we apply, in JSON format:

```
{
  "chromePolicies": {
    "DefaultDownloadDirectory": {
      "value": "/home/as2-streaming-user/MyFiles/TemporaryFiles"
    },
    "DownloadDirectory": {
      "value": "/home/as2-streaming-user/MyFiles/TemporaryFiles"
    },
    "DownloadRestrictions": {
      "value": 1
    },
    "URLAllowlist": {
      "value": [
        "file:///home/as2-streaming-user/MyFiles/TemporaryFiles",
        "file:///home/as2-streaming-user/MyFiles/TemporaryFiles/*",
        "file:///opt/appstream/tmp/TemporaryFiles",
        "file:///opt/appstream/tmp/TemporaryFiles/*"
      ]
    },
    "URLBlocklist": {
      "value": [
        "file://*",
        "http://169.254.169.254 (http://169.254.169.254/)",
        "http://169.254.169.254/*",
        "http://[fd00:ec2::254]",
        "http://[fd00:ec2::254]/*"
      ]
    }
  }
}
```

The following policies will override any existing values that you have set:

- DefaultDownloadDirectory
- DownloadDirectory
- DownloadRestrictions

The following policies will be merged with any existing values that you have set:

- URLAllowlist
- URLBlocklist

Set up your SAML 2.0 identity provider

Follow these steps to learn how to integrate Amazon WorkSpaces Web with the following SAML 2.0 identity providers.

Topics

Set up AWS SSO as your IdP

The following steps describe how to set up AWS Single Sign-On to use with WorkSpaces Web. This setup does not include any advanced features, such as Directory Services support.

If this is your first time visiting AWS SSO, you will be prompted to enable the service, which involves setting up AWS Organizations. For more information, see What is AWS Organizations?.

To set up AWS SSO as your IdP

1. From the AWS SSO console, choose **Users**, **Add user**, and enter the user's details.

   Note
   The email address entered will be used to send password reset requests.
2. (Optional) Choose **Next: Groups**, create a new group to assign this user to and choose **Add user**.
3. Choose **Applications**, **Add a new application**, and **Add a custom SAML 2.0 application**.
4. In another tab, from the WorkSpaces Web console, follow steps 1-3 of Step 1: Create a web portal (p. 17) to download the service provider (SP) metadata file. Keep this tab open.

5. Return to the AWS SSO console, and under **Application Metadata**, upload the downloaded SP metadata file.
6. Under the **AWS SSO Metadata** section, choose **Download** for the AWS SSO SAML metadata file, and then choose **Save changes** to finish creating the AWS SSO Application.
7. In the other tab, from the WorkSpaces Web console, follow step 5 and the remaining steps of Step 1: Create a web portal (p. 17) to upload the IdP metadata file and finish creating your web portal.
8. To configure SSO Application for users, follow these steps from the AWS SSO console:

    1. Choose **Attribute mappings** and enter the following fields:
        - For **User attribute in the application**, enter **Subject**.
        - For **Maps to this string value or user attribute in AWS SSO**, enter **${user:email}**.
        - For **Format**, enter **emailAddress**.
    2. Choose **Assigned users** to grant access to either an individual user or an entire group.
9. Follow the steps in Step 2: Test the endpoint (p. 20) to validate setup.

Set up Azure AD as your IdP

The following steps describe how to set up Azure AD to use with WorkSpaces Web. This setup doesn't include any advanced features, such as connecting to an on-premises Active Directory controller. You must have an Azure account to complete these steps.

To set up Azure AD as your IdP

1. Open the Microsoft Azure console.
2. To configure users and groups, follow these steps:

    1. On the overview page of your directory, choose **Users**, **New user**, **Create user**, and fill out the required user fields and password.
    2. If you want to manage user access to WorkSpaces Web with groups, choose a group.
    3. Choose **Create**.
3. To create a custom enterprise application, follow these steps:

    1. On the overview page of your directory, choose **Enterprise applications**, **New application**, and then **Create your own application**.
    2. Enter the name of the test application, and choose **Integrate any other application you don’t find in the gallery (Non-gallery)**.
4. To assign users and groups, follow these steps:

    1. On the overview page for your enterprise application, choose **Assign users and groups**, **Add user/group**, and **Users and groups**.
    2. Choose your users, and then choose **Select** and **Assign**.
5. To set up single sign-on, follow these steps:

    1. On the overview page of your enterprise application, under item 2 of the **Getting Started** section (**Set up single sign-on**), choose **Get started**.
    2. For the single sign-on method, choose **SAML**.
    3. In another tab, from the WorkSpaces Web console, follow steps 1-3 of Step 1: Create a web portal (p. 17) to download the service provider metadata file. Keep this tab open.
    4. Choose **Upload metadata file**, choose the file that you downloaded in the previous step, and choose **Add**.
    5. Under **Basic SAML Configuration**, verify that the **Entity ID** and **Assertion Consumer Service URL** fields are filled, and choose **Save**.

   6. Under **SAML Signing Certificate**, download the **Federation Metadata XML**. It might take a couple minutes for the file to be generated and downloaded.
   7. In the other tab, from the WorkSpaces Web console, follow step 5 and the remaining steps of Step 1: Create a web portal (p. 17) to upload the IdP metadata file and finish creating your web portal.
6. Follow the steps in Step 2: Test the endpoint (p. 20) to validate setup.

Set up Okta as your IdP

The following steps describe how to set up Okta to use with WorkSpaces Web. This setup does not use any advanced features, such as Dynamic IdP Metadata URL, or submitting an application template to be added to the Okta Integration Network. You must have Okta set up in order to proceed.

To set up Okta as your IdP

1. To create application integration between Okta and Workspaces Web, follow these steps:
   1. From the Okta console, choose **Applications**, **Applications**, and **Create App Integration**.
   2. Choose **SAML 2.0**, **Next**, enter an **App Name**, and then choose **Next**.
   3. In another tab, from the WorkSpaces Web console, follow steps 1-3 of Step 1: Create a web portal (p. 17) to **Show individual metadata values** of the service provider metadata file. Keep this tab open.
   4. Enter the following values for Okta's **SAML Settings**:
      - For **Single sign on URL**, enter the **ACS URL** from the previous step.
      - For **Audience URI (SP Entity ID)**, enter the **SP Entity ID** from the previous step.
      - Change **Name ID Format** to **EmailAddress**.
      - Leave **Application username** as specified.
   5. Choose **Next**, specify if you are a customer or a partner when prompted, and choose **Finish**.
2. Retrieve and upload the the IdP Metadata XML file from Okta.
   1. In the Okta console for your new application, on the **Sign On** tab, right-click **Identity Provider metadata**.
   2. Choose **Save Link as...** and enter a name for the IdP metadata file that ends in `.xml`.
   3. In the other tab, from the WorkSpaces Web console, follow step 5 and the remaining steps of Step 1: Create a web portal (p. 17) to upload the IdP metadata file from Okta and finish creating your web portal.
3. (Optional) Set up a test user.
   1. From the Okta dashboard, expand the sidebar and choose **Directory**, **People**, and **Add Person**.
   2. Fill out the fields in the form, and choose **Save**.
4. Assign a test user to your application.
   1. In the Okta console for your new application, choose **Assignments**, **Assign**, and **Assign to People**.
   2. Assign your Test User, yourself, or both with the credentials setup during Okta registration and choose **Save and Go back** and **Done**.
5. Follow the steps in Step 2: Test the endpoint (p. 20) to validate setup.

Set up PingIdentity as your IdP

The following steps describe how to set up PingIdentity to use with WorkSpaces Web. You must have PingIdentity set up in order to proceed.

To set up PingIdentity as your IdP

1. Setup an environment. (Skip this step if you want to use the Administrator environment for SAML integration.)

   1. From the Okta console, choose **Add Environment**, **Customer Solution**, **Next**, and then **Next**.
   2. For your license's deployment options, enter the **Environment Name**, and select the **Generate sample populations and users in this environment** checkbox.
   3. Choose **Finish**.
2. In another tab, from the WorkSpaces Web console, follow steps 1-3 of Step 1: Create a web portal (p. 17) to download the service provider metadata file. Keep this tab open.
3. Setup SAML for your PingIdentity customer environment.

   1. From the PingIdentity homepage, select the environment you want to set up SAML on.
   2. From the left-hand navigation menu, choose **Connections**.
   3. In the top-right corner, choose **Add application**.
   4. Under **Select an application type**, select the first option for web application and choose **Configure** for the **SAML** connection type.
   5. Enter an **Application Name** and choose **Next**.
   6. On **Configure SAML Connection**, choose **Choose File** and select the SP SAML metadata that you downloaded in step 2.
   7. View the values and, for **ASSERTION VALIDITY DURATION (IN SECONDS)**, enter how long a user can stay logged in, in seconds.
   8. Choose **Save and Continue**, and then choose **Save and Close**.
4. Retrieve Ping Application's IdP metadata and upload it to WorkSpaces Web.

   1. Navigate to your environment and choose **Connections** and **Applications**.
   2. Expand your application info and choose **Configuration**.
   3. Under **Connection Details**, choose the **Download** button to download the IdP metadata.
   4. In the other tab, from the WorkSpaces Web console, follow step 5 and the remaining steps of Step 1: Create a web portal (p. 17) to upload the IdP metadata file from Ping and finish creating your web portal.
5. Add a test user.

   1. From the PingIdentity console, choose **Identities**, **Add User**, fill in the fields, and choose **Save**.
   2. The user will be assigned to a **Population** that you automatically generated or a pre-existing population. If you don't have a population, choose **Population** and create one.
   3. Choose **Reset Password**, **Generate Password**, and choose the eye button to view the password. Copy the password for validation later and choose **Save**.
6. Validate.

   1. If you want to restrict users of the WorkSpaces Web application to a certain group, select groups to grant access to under the Ping Application's **Access** tab.
   2. To activate the Ping Application for **User Access**, navigate to your environment and choose **Connections** and **Applications**.
   3. Between the **Application name** and **Avg daily sign-ons**, toggle the application to **Enabled** for user access.
   4. Follow the steps in Step 2: Test the endpoint (p. 20) to validate setup.

Getting started with Amazon WorkSpaces Web

Follow these steps to create a WorkSpaces Web web portal and provide users with access to internal and SaaS websites from their existing browsers. You can create one web portal in any supported region per account.

Note
To request a limit increase for more than one portal, please contact support with your AWS Account ID, number of portals to request, and region.

This process typically takes five minutes with the web portal creation wizard, plus up to 15 minutes for the portal to become **Active**.

There are no costs associated with setting up a web portal. WorkSpaces Web offers pay-as-you-go pricing, including a low, monthly price for users who actively use the service. There are no up-front costs, licenses, or long-term commitments.

Topics

Prerequisites

Before you begin, make sure that you've completed all of the necessary prerequisites. For more information, see Setting up Amazon WorkSpaces Web (p. 4).

Step 1: Create a web portal

Follow these steps to create a web portal.

If you already completed these steps in Set up your SAML 2.0 identity provider (p. 13), you can skip this section and go to Step 2: Test the endpoint (p. 20).

1. Open the WorkSpaces Web console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. Choose **WorkSpaces Web**, **Web portals**, and then choose **Create web portal**.
3. On the **Step 1: Specify networking connection** page, complete the following steps to connect your VPC to your web portal, and configure your VPC and subnets.

   Note
   You can choose to skip this step for now and complete it after you create a web portal, in step 13 below.

   1. For **Networking details**, choose a VPC.

   2. Choose at lease two private subnets that meet all requirements. For more information, see Set up your network (p. 6).
   3. Choose a security group.

4. On the **Step 2: Configure web portal settings** page, complete the following steps to customize your users' browsing experience when they start a session, and then choose **Next**:

   > **Note**
   > WorkSpaces Web applies additional browser policies to isolate users to the browser interface sessions, on behalf of the customer. For more information, see Review browser policies (p. 12).

   1. Under **Web portal details**, for **Display name**, enter an identifiable name for your web portal.
   2. Under **Policy settings**, enter the following details:
      - For **Policy options**, choose **Visual editor** or **JSON file upload** to choose how to provide the policy configuration details for your web portal. WorkSpaces Web includes support for Chrome enterprise policies, and you can add and manage policies using either a visual editor, or a manual upload for policy files. You can switch between either option at any time.

        When you upload a policy file, you will see the available policies in the file. However, not all policies can be edited in the visual editor. You might need to manually edit the JSON data to make changes to a policy.
      - For **Startup URL - optional**, you can enter a domain to use as the homepage when users launch their browser. Your VPC must have a stable connection to this URL.
      - For **Browser bookmarks - optional**, you can enter the **Display name**, **Domain**, and **Folder** for any bookmarks you want your users to see in their browser, and choose **Add bookmark**.

        > **Note**
        > **Domain** is a required field for browser bookmarks.

5. On the **Step 3: Select user settings** page, complete the following steps to choose which features your users can access from the top navigation bar during their session, and then choose **Next**:

   1. For **Clipboard**, choose **Disabled** or **Enabled**.
   2. Under **File transfer**, choose **Disabled** or **Enabled**.
   3. For **Print to local device**, choose **Allowed** or **Not allowed**.
   4. For **User session details**, specify the following:
      - For **Disconnect timeout in minutes**, choose the amount of time that a streaming session remains active after users disconnect. If users try to reconnect to the streaming session after a disconnection or network interruption within this time interval, they are connected to their previous session. Otherwise, they are connected to a new session with a new streaming instance.

        If a user ends the session, the disconnect timeout does not apply. Instead, the user is prompted to save any open documents, and then is immediately disconnected from the streaming instance. The instance the user was using is then terminated.
      - For **Idle disconnect timeout in minutes**, choose the amount of time that users can be idle (inactive) before they are disconnected from their streaming session and the **Disconnect timeout in minutes** time interval begins. Users are notified before they are disconnected due to inactivity. If they try to reconnect to the streaming session before the time interval specified in **Disconnect timeout in minutes** has elapsed, they are connected to their previous session. Otherwise, they are connected to a new session with a new streaming instance. Setting this value to 0 disables it. When this value is disabled, users are not disconnected due to inactivity.

        > **Note**
        > Users are considered idle when they stop providing keyboard or mouse input during their streaming session. File uploads and downloads, audio in, audio out, and pixels

changing do not qualify as user activity. If users continue to be idle after the time interval in **Idle disconnect timeout in minutes** elapses, they are disconnected.

6. On the **Step 4: Configure identity provider** page of the creation wizard, choose **Download metadata file** to download the service provider (SP) metadata document that you will upload to your identity provider (IdP) in the next step. You must upload the service provider metadata file to your IdP. Otherwise, your users won't be able to log in.

   Note
   WorkSpaces Web supports service provider initiated (SP-initiated) sign-in flows with your SAML 2.0-compliant IdP. WorkSpaces Web does not yet support identity provider initiated (IdP-initiated) sign-in flows.

7. Open another tab in your browser, and complete the following steps for your IdP:

   1. Upload the SP metadata document that you downloaded in the previous step to your IdP. You must either upload the file to your IdP, or copy and paste the metadata values (for providers like Okta). The details of this configuration process vary between providers. Check your provider's documentation for detailed help on adding the details provided by WorkSpaces Web to your configuration.
   2. Grant access to your users in your IdP to use WorkSpaces Web.
   3. Download a metadata exchange file from your IdP. You will upload this metadata to WorkSpaces Web in the next step.

8. Return to the WorkSpaces Web console, and on the **Configure identity provider** page of the creation wizard, under **IdP metadata document**, choose **Choose file** to upload the XML-formatted metadata file from IdP that you downloaded in the previous step. WorkSpaces Web requires this metadata from your IdP to establish trust. When you are done, choose **Next**.

   Note
   WorkSpaces Web requires the `subject` or `NameID` to be mapped and set in the SAML assertion within your IdP's settings. Your IdP can create these mappings automatically. If these mappings are not configured correctly, a user who attemps to sign in to the web portal might be unable to start a session.

9. On the **Step 5: Review and launch** page, review the settings you've selected for your web portal. You can choose **Edit** to make any changes, or you can change these settings later on from the **Web portals** tab of the console.

10. When you're done, choose **Launch web portal**.

11. To view the status of your web portal, choose **Web portals**, choose your portal, and choose **View details**.

    A web portal can have one of the following statuses:

    - **Incomplete** - The web portal's configuration is missing required identity provider settings.
    - **Pending** - The web portal is applying changes to its settings.
    - **Active** - The web portal is ready and available for use.

12. Wait up to 15 minutes for your portal to become **Active**.

13. If you skipped step 3 above, follow these steps to configure your subnets:

    1. Choose **Web portals**, choose your portal, and then choose **Edit**.
    2. In **Networking details**, choose a VPC with VPC endpoints.
    3. Choose at lease two private subnets with all three VPC endpoints that you created previously. Make sure they are in different AZs.
    4. Choose **Save**, and wait up to 15 minutes for the changes to take effect.

Step 2: Test the endpoint

After you create a web portal, you can sign into the WorkSpaces Web endpoint to browse your connected websites as an end user would.

If you already completed these steps in Set up your SAML 2.0 identity provider (p. 13), you can skip this section and go to Step 3: Distribute the endpoint (p. 20).

1. Open the WorkSpaces Web console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. Choose **WorkSpaces Web**, **Web portals**, choose your web portal, and then choose **View details**
3. Under **Web portal endpoint**, go to the specified URL for your portal. The web portal endpoint is the access point your users will launch your web portal from after signing in with the identity provider configured for the portal. It's publicly available on the internet and can be embedded into your network.
4. On the WorkSpaces Web sign-in page, choose **Sign in**, **SAML**, and enter your SAML credentials.
5. When you see the **Your session is being prepared** page, your WorkSpaces Web session is launching. Do not close or exit this page.
6. The web browser launches, displaying your startup URL and any other additional behavior configured through your browser policy settings.
7. You can now browse to connected websites by choosing links or enter URLs into the address bar.

Step 3: Distribute the endpoint

When you are ready for your users to begin using WorkSpaces Web to access their streaming browser, you choose from the following options to distribute the endpoint to them:

- Email the endpoint URL to your users.
- Use a URL that you own, by choosing one of the following options:
  - Use your IdP to register an arbitrary link (in this case, the web portal endpoint) as something that will show up as an application for users who log into their IdP directly.
  - Add the endpoint to a website that you own, and use a browser redirect to direct users to the web portal.

Next steps

After you create your first web portal, you can view details, edit details, or delete the web portal at any time. For more information, see Managing your web portal (p. 21).

Managing your web portal

After you set up your web portal, you can view or edit its details, as well as delete the portal if it is no longer needed.

Topics

View web portal details

To view web portal details

1. Open the WorkSpaces Web console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. Choose **WorkSpaces Web**, **Web portals**, choose your web portal, and then choose **View details**.

Edit a web portal

To edit a web portal

1. Open the WorkSpaces Web console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. Choose **WorkSpaces Web**, **Web portals**, choose your web portal, and then choose **Edit**.

   Note
   If you make changes to a user's settings while the user is actively using a session, your changes will take effect the next time the user starts a new session.

Delete a web portal

To delete a web portal

1. Open the WorkSpaces Web console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. Choose **WorkSpaces Web**, **Web portals**, choose your web portal, and then choose **Delete**.

Request a service quota increase

When you create your AWS account, we automatically set default service quotas (also referred to as limits) for resource usage with AWS Services. WorkSpaces Web sets quotas on two types of resources - web portals (per region) and maximum concurrent sessions (per web portal). WorkSpaces Web currently has the following service quotas limits:

Default quotas per AWS region per account	Value
Web portals	1
Maximum concurrent sessions	25

A *web portal* is the foundational resource for the WorkSpaces Web service. It is an association between your SAML 2.0 identity provider, and your networking connection to the internet and your content. You can create a web portal in any region where WorkSpaces Web is available. See the region table for current availability.

The *maximum concurrent sessions* is the highest amount of users that will be connected at the same time to a given web portal. If the service quota limit for maximum concurrent sessions is not set appropriately, users may find that their session is not available when they sign into WorkSpaces Web. You should also ensure that your VPC and subnets have sufficient IP space to support the maximum concurrent sessions, or users might be unable to connect to a session.

For example, a customer has two web portals in US East (Northern Virginia) and 125 users. The first web portal (portal A) will be used by 25 users, and does not require a service quota increase. The second web portal (portal B) will be used by 100 users. These users are spread across two shifts, and their working hours do not overlap. Therefore, the customer would need to request a service quota increase for Portal B to a maximum concurrent session of 50 users.

You can request an increase for either one of these service quota limits. For more information, see Requesting a quota increase.

To request a service quota increase

1. Open the AWS Support dashboard.
2. Choose **Service Limit Increase**.

   Important
   WorkSpaces Web service quotas affect one Region at a time. You must request service quota increases in each AWS Region where you need more resources. For more information, see AWS service endpoints.
3. Under **Use case description**, enter the following information:

   - If you are requesting an increase for the number of web portals, specify this resource type, and include your AWS Account ID, the region where you would like the increase, and the new limit value.
   - If you are requesting an increase for maximum concurrent sessions, specify this resource type, and include your AWS Account ID, the region where you would like the increase, the web portal ARN, and the new limit value.
4. (Optional) To request multiple service quota increases at the same time, complete one quota increase request in the **Requests section**, and then choose **Add another request**.

Control the interval for re-authenticating a SAML IdP token

When a user visits a WorkSpaces Web portal, they can sign in to launch a streaming session. Every sessions begins on the start page, unless they sign in less than 5 minutes ago. The portal checks for identity provider (IdP) tokens to determine whether to prompt the user for credentials when it launches a session. A user without a valid IdP token must enter a user name, password, and (optionally multifactor authentication (MFA) to launch a streaming session. If a user already generated a SAML IdP token by signing into their IdP or an app protected by the same IdP, they won't be asked for a user name or password.

If a user has a valid SAML IdP token, they can access WorkSpaces Web. You can control the interval required for re-authenticating a SAML IdP token.

To control the interval for re-authenticating a SAML IdP token

1. Set the IdP timeout duration with your SAML IdP provider. We recommend configuring your IdP timeout duration with the shortest amount of time necessary for a user to complete their tasks.

   - For more information about Okta, see Enforce a limited session lifetime for all policies.
   - For more information about Azure AD, see Configuring authentication session controls.
   - For more information about Ping, see Sessions.
   - For more information about AWS Single Sign-On, see Set session duration.
2. Set your WorkSpaces Web portal's inactivity and idle timeout values. These values controls the amount of time between a user’s last interaction and when a WorkSpaces Web session ends due to inactivity. When a session ends, a user will lose their session state (including open tabs, unsaved web content, and history), and return to a fresh state at the start of the next session. For more information, see step 5 in the section called “Step 1: Create a web portal” (p. 17).

   Note
   If a user's session times out but the user still has a valid SAML IdP token, they don't have to enter their user name and password to start a new WorkSpaces Web session. To control how tokens are re-authenticated, follow the guides in the previous step.

Security in Amazon WorkSpaces Web

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

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

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon WorkSpaces Web, 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 any applicable laws and regulations that apply to your data.

This documentation helps you understand how to apply the shared responsibility model when using Amazon WorkSpaces Web. It shows you how to configure Amazon WorkSpaces Web to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon WorkSpaces Web resources.

Contents

Data protection in Amazon WorkSpaces Web

The AWS shared responsibility model applies to data protection in Amazon WorkSpaces Web. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

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

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.

- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

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

Data encryption

Amazon WorkSpaces Web collects portal customization data, such as browser settings, user settings, network settings, identity provider information, trust store data, and trust store certificate data. WorkSpaces Web also collects browser policy data, user preferences (for browser settings), and session logs. Collected data is stored in Amazon DynamoDB and Amazon S3. WorkSpaces Web uses AWS Key Management Service for encryption.

To secure your content, follow these guidelines:

- Implement least privilege access and create specific roles to be used for WorkSpaces Web actions. Use IAM templates to create a Full Access role or Read Only role. For more information, see AWS managed policies for WorkSpaces Web (p. 37).
- Protect data end to end by providing a customer managed key, so WorkSpaces Web can encrypt your data at rest with the keys you supply.
- Be careful with sharing portal domains and user credentials:
  - Admins are required to log into the Amazon WorkSpaces console, and users are required to log into the WorkSpaces Web portal.
  - Anyone on the internet can access the web portal, but they can't start a session unless they have valid user credentials to the portal.
- Users can explicitly end their sessions by choosing **End Session**. This discards the instance hosting the browser session, and results in browser isolation.

WorkSpaces Web secures content and metadata by default by encrypting all sensitive data with AWS KMS. It collects browser policy and user preferences to enforce policy and settings during WorkSpaces Web sessions. If there is an error applying existing settings, a user can't access new sessions and can't access the company's internal sites and SaaS applications.

Encryption at rest

Encryption at rest is configured by default. Customer-specific data used in WorkSpaces Web is encrypted using AWS KMS. WorkSpaces Web provides encryption at rest for resources you create. The service accepts a AWS KMS Customer Managed Key on resource creation, and if one is not provided, an AWS Owned Key will be used to encrypt the resources at rest. The service encrypts the Browser Policy document you can provide to customize your browser sessions, as well as your identity provider configuration, and display names for your portals. This information will remain encrypted using either the Customer Managed Key, or the AWS Owned Key, while it is stored in our backend.

You can decide which key will be used when you create a WorkSpaces Web resource. If data that is part of that resource is encrypted, WorkSpaces Web accepts the `customerManagedKeyArn` field as part of the `create` API. The key provided must be a Symmetric AWS KMS key, and the administrator who creates the resource using this key must have `kms:Decrypt`, `kms:GenerateDataKey`, and `kms:CreateGrant` permissions. After a resource is created with the key, the key can't be removed or changed. If you used a Customer Managed Key, the administrator who accesses the resource must have `kms:Decrypt` and `kms:GenerateDataKey` permissions. If you see an error about access being denied while using the console, make sure that the user using the console has these permissions with the key that was used.

You can troubleshoot and audit key usage by checking the status of the AWS KMS grants. For more information, see Managing grants. During portal creation, WorkSpaces Web create a grant to allow the service to access the key asynchronously. You can check the status of our key usage by checking the grant, as well as the Encryption Context provided when the grant is used. The encryption context always contains an entry with the key `aws:workspaces-web:portal:id` and a value equal to your portal ID. For other resources, the encryption context will always contain an entry in the format `aws:workspaces-web:`*`RESOURCE_TYPE`*`:id` and the corresponding resource ID.

Encryption in transit

WorkSpaces Web encrypts data in transit over HTTPS and TLS 1.2. You can send a request to WorkSpaces by using the console or direct API calls. The request data that is transferred is encrypted by sending everything through a HTTPS or TLS connection. Request data can be transferred from the AWS Console, AWS Command Line Interface, or AWS SDK to WorkSpaces Web.

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 one, WorkSpaces Web will use an AWS Owned Key. You can set your key using the AWS SDK.

Inter-network traffic privacy

To secure connections between WorkSpaces Web and on-premise applications, you use WorkSpaces Web to launch browser sessions inside of your own VPC. The connection to on-premise applications is configured in your own VPC, and is not controlled by WorkSpaces Web.

To secure connections between accounts, WorkSpaces Web uses a service-linked role to securely connect to customer accounts and run operations on behalf of the customer. For more information, see Using service-linked roles for WorkSpaces Web (p. 42).

Identity and Access Management for Amazon WorkSpaces Web

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

Topics

Audience

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

Service user – If you use the WorkSpaces Web service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more WorkSpaces Web 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 Web, see Troubleshooting Amazon WorkSpaces Web identity and access (p. 40).

Service administrator – If you're in charge of WorkSpaces Web resources at your company, you probably have full access to WorkSpaces Web. It's your job to determine which WorkSpaces Web features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with WorkSpaces Web, see How Amazon WorkSpaces Web works with IAM (p. 31).

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 Web. To view example WorkSpaces Web identity-based policies that you can use in IAM, see Identity-based policy examples for Amazon WorkSpaces Web (p. 35).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see Signing in to the AWS Management Console as an IAM user or root user in the *IAM User Guide*.

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

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

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

AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM users and groups

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

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

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

IAM roles

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

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

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the *IAM User Guide*.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.

  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for Amazon WorkSpaces Web in the *Service Authorization Reference*.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

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

Managing access using policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. You can sign in as the root user or an IAM user, or you can assume an IAM role. When you then make a request, AWS evaluates the related identity-based or resource-based policies. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

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

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

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

Identity-based policies

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

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

Resource-based policies

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

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

Access control lists (ACLs)

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

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

Other policy types

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

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

Multiple policy types

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

How Amazon WorkSpaces Web works with IAM

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

IAM features you can use with Amazon WorkSpaces Web

IAM feature	WorkSpaces Web support
Identity-based policies (p. 31)	Yes
Resource-based policies (p. 32)	No
Policy actions (p. 32)	Yes
Policy resources (p. 33)	Yes
Policy condition keys (p. 33)	Yes
ACLs (p. 34)	No
ABAC (tags in policies) (p. 34)	Partial
Temporary credentials (p. 34)	Yes
Principal permissions (p. 34)	Yes
Service roles (p. 35)	No
Service-linked roles (p. 35)	Yes

To get a high-level view of how WorkSpaces Web 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 Web

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 Web

To view examples of WorkSpaces Web identity-based policies, see Identity-based policy examples for Amazon WorkSpaces Web (p. 35).

Resource-based policies within WorkSpaces Web

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 How IAM roles differ from resource-based policies in the *IAM User Guide*.

Policy actions for WorkSpaces Web

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 Web actions, see Actions defined by Amazon WorkSpaces Web in the *Service Authorization Reference*.

Policy actions in WorkSpaces Web use the following prefix before the action:

```
workspaces-web
```

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

```
"Action": [
      "workspaces-web:action1",
      "workspaces-web:action2"
         ]
```

To view examples of WorkSpaces Web identity-based policies, see Identity-based policy examples for Amazon WorkSpaces Web (p. 35).

Policy resources for WorkSpaces Web

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

To view examples of WorkSpaces Web identity-based policies, see Identity-based policy examples for Amazon WorkSpaces Web (p. 35).

Policy condition keys for WorkSpaces Web

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

To view examples of WorkSpaces Web identity-based policies, see Identity-based policy examples for Amazon WorkSpaces Web (p. 35).

Access control lists (ACLs) in WorkSpaces Web

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.

Attribute-based access control (ABAC) with WorkSpaces Web

Supports ABAC (tags in policies)	Partial

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.

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 Web

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 Web

Supports principal permissions	Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for Amazon WorkSpaces Web in the *Service Authorization Reference*.

Service roles for WorkSpaces Web

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 break WorkSpaces Web's functionality. Edit service roles only when WorkSpaces Web provides guidance to do so.

Service-linked roles for WorkSpaces Web

Supports service-linked roles	Yes

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

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 Web

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

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

Topics

Policy best practices

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

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

Using the WorkSpaces Web console

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

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

To ensure that users and roles can still use the WorkSpaces Web console, also attach the WorkSpaces Web `ConsoleAccess` or `ReadOnly` AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

Allow users to view their own permissions

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

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

AWS managed policies for WorkSpaces Web

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

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

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

AWS managed policy: AmazonWorkSpacesWebServiceRolePolicy

You can't attach the `AmazonWorkSpacesWebServiceRolePolicy` policy to your IAM entities. This policy is attached to a service-linked role that allows WorkSpaces Web to perform actions on your behalf. For more information, see the section called “Using Service-Linked Roles” (p. 42).

This policy grants administrative permissions that allow access to AWS services and resources used or managed by Amazon WorkSpaces Web.

Permissions details

This policy includes the following permissions:

- `WorkSpaces Web` – Allows access to AWS services and resources used or managed by Amazon WorkSpaces Web.
- `ec2` – Allows principals to describe VPCs, subnets, and availability zones; create, describe, and delete network interfaces; associate or disassociate an address; and describe route tables, security groups, and VPC endpoints.
- `CloudWatch` – Allows principals to put metric data.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeAvailabilityZones",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface",
        "ec2:AssociateAddress",
        "ec2:DisassociateAddress",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcEndpoints"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*",
      "Condition": {
   "StringEquals": {
       "cloudwatch:namespace": [
    "AWS/WorkSpacesWeb",
    "AWS/Usage"
       ]
        }
      }
    }
  ]
}
```

AWS managed policy: AmazonWorkSpacesWebReadOnly

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

This policy grants read-only permissions that allow access to WorkSpaces Web and its dependencies through the AWS Management Console, SDK, and CLI.

Permissions details

This policy includes the following permissions.

- `WorkSpaces Web` – Provides read-only access to Amazon WorkSpaces Web and its dependencies through the AWS Management Console, SDK, and CLI.
- `ec2` – Allows principals to describe VPCs, subnets, and security groups. This is used in the AWS Management Console in WorkSpaces Web to show you your VPCs, subnets, and security groups that are available foruse with the service.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "workspaces-web:GetBrowserSettings",
        "workspaces-web:GetIdentityProvider",
        "workspaces-web:GetNetworkSettings",
        "workspaces-web:GetPortal",
        "workspaces-web:GetPortalServiceProviderMetadata",
        "workspaces-web:GetTrustStore",
        "workspaces-web:GetTrustStoreCertificate",
        "workspaces-web:GetUserSettings",
        "workspaces-web:ListBrowserSettings",
        "workspaces-web:ListIdentityProviders",
        "workspaces-web:ListNetworkSettings",
        "workspaces-web:ListPortals",
        "workspaces-web:ListTagsForResource",
        "workspaces-web:ListTrustStoreCertificates",
        "workspaces-web:ListTrustStores",
        "workspaces-web:ListUserSettings"
      ],
      "Resource": "arn:aws:workspaces-web:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

WorkSpaces Web updates to AWS managed policies

View details about updates to AWS managed policies for WorkSpaces Web since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the *Document history* (p. 51) page.

Change	Description	Date
AmazonWorkSpacesWebServiceRolePolicy (p. 37) – Updated policy	WorkSpaces Web updated the policy to add the AWS/Usage namespace to the PutMetricData API permissions.	April 6, 2022

Change	Description	Date
AmazonWorkSpacesWebReadOnly (p. 38) – New policy	WorkSpaces Web added a new policy to provide read-only access to Amazon WorkSpaces Web and its dependencies through the AWS Management Console, SDK, and CLI.	November 30, 2021
AmazonWorkSpacesWebServiceRolePolicy (p. 37) – New policy	WorkSpaces Web added a new policy to allow access to AWS services and resources used or managed by Amazon WorkSpaces Web.	November 30, 2021
WorkSpaces Web started tracking changes	WorkSpaces Web started tracking changes for its AWS managed policies.	November 30, 2021

Troubleshooting Amazon WorkSpaces Web identity and access

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

Topics

I am not authorized to perform an action in WorkSpaces Web

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-example-widget`* resource but does not have the fictional `workspaces-web:`*`GetWidget`* permissions.

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

In this case, Mateo asks his administrator to update his policies to allow him to access the *`my-example-widget`* resource using the `workspaces-web:`*`GetWidget`* action.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to WorkSpaces Web.

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

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

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

In this case, Mary asks her administrator to update her policies to allow her to perform the `iam:PassRole` 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 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 Web

To allow others to access WorkSpaces Web, 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 Web.

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

I want to allow people outside of my AWS account to access my WorkSpaces Web 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 Web supports these features, see How Amazon WorkSpaces Web works with IAM (p. 31).

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

Using service-linked roles for WorkSpaces Web

WorkSpaces Web uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to WorkSpaces Web. Service-linked roles are predefined by WorkSpaces Web and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up WorkSpaces Web easier because you don’t have to manually add the necessary permissions. WorkSpaces Web defines the permissions of its service-linked roles, and unless defined otherwise, only WorkSpaces Web can assume its roles. The defined permissions include the trust and permissions policies.The permissions policy can't be attached to any other IAM entity.

You can delete a service-linked role only after first deleting its related resources. This protects your WorkSpaces Web resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for WorkSpaces Web

WorkSpaces Web uses the service-linked role named `AWSServiceRoleForAmazonWorkSpacesWeb` – WorkSpaces Web uses this service-linked role to access Amazon EC2 resources of customer accounts for streaming instances and CloudWatch metrics.

The `AWSServiceRoleForAmazonWorkSpacesWeb` service-linked role trusts the following services to assume the role:

- `workspaces-web.amazonaws.com`

The role permissions policy named `AmazonWorkSpacesWebServiceRolePolicy` allows WorkSpaces Web to complete the following actions on the specified resources:

- Action: `ec2:DescribeVpcs` on `all AWS resources`
- Action: `ec2:DescribeSubnets` on `all AWS resources`
- Action: `ec2:DescribeAvailabilityZones` on `all AWS resources`
- Action: `ec2:CreateNetworkInterface` on `all AWS resources`
- Action: `ec2:DescribeNetworkInterfaces` on `all AWS resources`
- Action: `ec2:DeleteNetworkInterface` on `all AWS resources`
- Action: `ec2:DescribeSubnets` on `all AWS resources`
- Action: `ec2:AssociateAddress` on `all AWS resources`
- Action: `ec2:DisassociateAddress` on `all AWS resources`
- Action: `ec2:DescribeRouteTables` on `all AWS resources`

- Action: `ec2:DescribeSecurityGroups` on `all AWS resources`
- Action: `ec2:DescribeVpcEndpoints` on `all AWS resources`
- Action: `cloudwatch:PutMetricData` on `all AWS resources`

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a service-linked role for WorkSpaces Web

You don't need to manually create a service-linked role. When you create your first portal in the AWS Management Console, the AWS CLI, or the AWS API, WorkSpaces Web creates the service-linked role for you.

> **Important**
> This service-linked role can appear in your account if you completed an action in another service that uses the features supported by this role.

If you delete this service-linked role and later need to create it again, you can use the same process to recreate the role in your account. When you create your first portal, WorkSpaces Web creates the service-linked role for you again.

You can also use the IAM console to create a service-linked role with the **WorkSpaces Web** use case. In the AWS CLI or the AWS API, create a service-linked role with the `workspaces-web.amazonaws.com` service name. For more information, see Creating a Service-Linked Role in the *IAM User Guide*. If you delete this service-linked role, you can use this same process to create the role again.

Editing a service-linked role for WorkSpaces Web

WorkSpaces Web doesn't allow you to edit the `AWSServiceRoleForAmazonWorkSpacesWeb` service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for WorkSpaces Web

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

> **Note**
> If the WorkSpaces Web service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete WorkSpaces Web resources used by the AWSServiceRoleForAmazonWorkSpacesWeb

- Choose from one of the following options:
  - If you use the console, delete all of your portals on the console.
  - If you use the CLI or API, disassociate all of your resources (including browser settings, network settings, user settings, and trust stores) from your portals, delete these resources, and then delete the portals.

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForAmazonWorkSpacesWeb service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported regions for WorkSpaces Web service-linked roles

WorkSpaces Web supports using service-linked roles in all of the regions where the service is available. For more information, see AWS Regions and Endpoints.

Incident response in Amazon WorkSpaces Web

You can detect incidents by monitoring the `SessionFailure` Amazon CloudWatch metric. To receive alerts for incidents, use a CloudWatch alarm for the `SessionFailure` metric. For more information, see Monitoring Amazon WorkSpaces Web with Amazon CloudWatch (p. 47).

Compliance validation for Amazon WorkSpaces Web

Third-party auditors assess the security and compliance of AWS services as part of multiple AWS compliance programs, such as SOC, PCI, FedRAMP, and HIPAA.

To learn whether Amazon WorkSpaces Web or other AWS services are within the scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

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

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

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

  > **Note**
  > Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in Amazon WorkSpaces Web

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 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.

The following are currently not supported by WorkSpaces Web:

- Backing up content across AZs or regions
- Encrypted backups
- Encrypting in-transit content between AZs or regions
- Default or automatic backups

To configure for high internet availability, you can tune your VPC configuration. For high API availability, you can request the right amount of TPS.

Infrastructure security in Amazon WorkSpaces Web

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

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

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

WorkSpaces Web isolates service traffic by applying Standard AWS SigV4 Authentication and Authorization to all services. The customer resource endpoint (or web portal endpoint) is protected by your identity provider. You can further isolate traffic by using Multi-factor Authorization and other security mechanism in your identity provider (IdP).

All internet access can be controlled by configuring network settings, such as the VPC, subnet, or security group. Multi-tenancy and VPC endpoints (PrivateLink) are not currently supported.

Configuration and vulnerability analysis in Amazon WorkSpaces Web

WorkSpaces Web updates and patches applications and platforms as needed on your behalf, including Chrome and Linux. You are not required to patch or rebuild. However, it is your responsibility to configure WorkSpaces Web according to specifications and guidelines, and to monitor WorkSpaces Web usage by your users. All service-related configs and vulnerability analysis are the responsibility of WorkSpaces Web.

You can request a limit increase for WorkSpaces Web resources, such as the number of web portals and number of users. WorkSpaces Web ensures the availability of the service and SLA.

Security best practices for Amazon WorkSpaces Web

Amazon WorkSpaces Web provides a number of security features you can use as you develop and implement your own security policies. The following best practices are general guidelines and don’t represent a complete security solution. Because these best practices might not be appropriate or sufficient for your environment, treat them as helpful considerations rather than prescriptions.

Best practices for Amazon WorkSpaces Web include the following:

- To detect potential security events associated with your use of WorkSpaces Web, use AWS CloudTrail or Amazon CloudWatch to detect and track access history and process logs. For more information, see Monitoring Amazon WorkSpaces Web with Amazon CloudWatch (p. 47) and Logging Amazon WorkSpaces Web API calls using AWS CloudTrail (p. 48).
- To implement detective controls and identify anomalies, use CloudTrail logs and CloudWatch metrics. For more information, see Monitoring Amazon WorkSpaces Web with Amazon CloudWatch (p. 47) and Logging Amazon WorkSpaces Web API calls using AWS CloudTrail (p. 48).

To prevent potential security events associated with your use of WorkSpaces Web, follow these best practices:

- Implement least privilege access and create specific roles to be used for WorkSpaces Web actions. Use IAM templates to create a Full Access or Read Only role. For more information, see AWS managed policies for WorkSpaces Web (p. 37).
- Be careful with sharing portal domains and user credentials. Anyone on the internet can access the web portal, but they can't start a session unless they have a valid user credential to the portal. Be cautious about how, when, and to whom you share web portal credentials.

Monitoring Amazon WorkSpaces Web

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

- *Amazon CloudWatch* monitors your AWS resources and and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a specified threshold. For example, you can have CloudWatch track CPU usage or other metrics for your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the Amazon CloudWatch User Guide.
- *Amazon CloudWatch Logs* lets you monitor, store, and access your log files from Amazon EC2 instances, CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the Amazon CloudWatch Logs User Guide.
- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Monitoring Amazon WorkSpaces Web with Amazon CloudWatch

You can monitor Amazon WorkSpaces Web using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These statistics are kept for 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing. You can also set alarms that watch for certain thresholds, and send notifications or take actions when those thresholds are met. For more information, see the Amazon CloudWatch User Guide.

The `AWS/WorkSpacesWeb` namespace includes the following metrics.

CloudWatch metrics for Amazon WorkSpaces Web

Metric	Description	Dimensions	Statistics	Units
`SessionAttempt`	The number of Amazon WorkSpaces Web session attempts.	`PortalId`	Average, Sum, Maximum, Minimum	Count
`SessionSuccess`	The number of successful Amazon WorkSpaces Web session starts.	`PortalId`	Average, Sum, Maximum, Minimum	Count

Metric	Description	Dimensions	Statistics	Units
`SessionFailure`	The number of failed Amazon WorkSpaces Web session starts.	`PortalId`	Average, Sum, Maximum, Minimum	Count

Logging Amazon WorkSpaces Web API calls using AWS CloudTrail

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

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

Amazon WorkSpaces Web information in CloudTrail

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

For an ongoing record of events in your AWS account, including events for Amazon WorkSpaces Web, you can 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 AWSRegions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple regions and Receiving CloudTrail log files from multiple accounts

All Amazon WorkSpaces Web actions are logged by CloudTrail and are documented in the *Amazon WorkSpaces API Reference*. For example, calls to the `CreatePortal`, `DeleteUserSettings` and `ListBrowserSettings` actions generate entries in the CloudTrail log files.

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

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

- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity element.

Understanding Amazon WorkSpaces Web 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 other details. 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 `ListBrowserSettings` action.

```
{
    "Records": [{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "myUserName"
        },
        "eventTime": "2021-11-17T23:44:51Z",
        "eventSource": "workspaces-web.amazonaws.com",
        "eventName": "ListBrowserSettings",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "[]",
        "requestParameters": null,
        "responseElements": null,
        "requestID": "159d5c4f-c8c8-41f1-9aee-b5b1b632e8b2",
        "eventID": "d8237248-0090-4c1e-b8f0-a6e8b18d63cb",
        "readOnly": true,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    },
    {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "myUserName"
        },
        "eventTime": "2021-11-17T23:55:51Z",
        "eventSource": "workspaces-web.amazonaws.com",
        "eventName": "CreateUserSettings",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "5127.0.0.1",
        "userAgent": "[]",
        "requestParameters": {
            "clientToken": "some-token",
            "copyAllowed": "Enabled",
            "downloadAllowed": "Enabled",
            "pasteAllowed": "Enabled",
            "printAllowed": "Enabled",
            "uploadAllowed": "Enabled"
        },
        "responseElements": "arn:aws:workspaces-web:us-
west-2:111122223333:userSettings/04a35a2d-f7f9-4b22-af08-8ec72da9c2e2",
        "requestID": "6a4aa162-7c1b-4cf9-a7ac-e0c8c4622117",
        "eventID": "56f1fbee-6a1d-4fc6-bf35-a3a71f016fcb",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }]
}
```

Document history for the Amazon WorkSpaces Web User Guide

The following table describes the documentation releases for Amazon WorkSpaces Web.

update-history-change	update-history-description	update-history-date
Timeout values	Specify the **Disconnect timeout in minutes** and **Idle disconnect timeout in minutes**	May 16, 2022
Updated managed policy	Updated the AmazonWorkSpacesWebServiceRolePolicy managed policy to add the AWS/Usage namespace to the PutMetricData API permissions	April 6, 2022
Service-linked role	New AWSServiceRoleForAmazonWorkSpacesWeb service-linked role	November 30, 2021
Managed policy	New AmazonWorkSpacesWebReadOnly managed policy	November 30, 2021
Managed policy	New AmazonWorkSpacesWebServiceRolePolicy managed policy	November 30, 2021
Initial release	Initial release of the WorkSpaces Web Administration Guide	November 30, 2021