aws

Administration Guide

Amazon WorkSpaces Secure Browser

Amazon WorkSpaces Secure Browser: Administration Guide

Table of Contents

What is Amazon WorkSpaces Secure Browser?

> **Note**
>
> Amazon WorkSpaces Secure Browser was previously known as Amazon WorkSpaces Web.

Amazon WorkSpaces Secure Browser is a fully managed, cloud-native, hosted browser service used to securely access private websites and software-as-a-service (SaaS) web applications, interact with online resources, and browse the internet from a disposable container. WorkSpaces Secure Browser works with a user's existing web browsers, without burdening IT with managing appliances, infrastructure, specialized client software, or virtual private network (VPN) connections. Web content is streamed to the user's web browser, while the actual browser and web content is isolated in AWS. By using the same underlying technologies that power AWS End User Computing services like Amazon WorkSpaces and Amazon AppStream 2.0, WorkSpaces Secure Browser can be more cost effective than traditional virtual desktops, and reduce complexity compared to providing company-owned devices with management software. WorkSpaces Secure Browser reduces the risk of data exfiltration by streaming web content. No HTML, document object model (DOM), or sensitive company data is transmitted to the local machine. By isolating the device, corporate network, and internet from each other, the browser attack surface is virtually eliminated.

You can enforce enterprise browser policy (including URL allow/blocking) on all sessions, and includes session-level controls for clipboard, file transfer, and printer. You can also restrict access to trusted networks or devices by using IP Access Controls. WorkSpaces Secure Browser is easy to set up and operate. Each session launches with a fresh and fully patched version of the Chrome Browser, with company policies and settings applied.

Release history for Amazon WorkSpaces Secure Browser

On May 20, 2024, Amazon WorkSpaces Web was renamed to Amazon WorkSpaces Secure Browser. For existing customers, there was no change to how they manage users or resources with the service. The following list describes the applicable updates that also took place as a result of this rename.

The *workspaces-web* API namespace remains unchanged for backward compatibility. As a result, the following resources are still the same:

- CLI commands.
- Amazon CloudWatch metrics. For more information, see the section called “Monitoring with CloudWatch”.
- Service endpoints. For more information, see Amazon WorkSpaces Secure Browser endpoints and quotas.
- AWS CloudFormation resources. For more information, see Amazon WorkSpaces Secure Browser resource type reference.
- Service-linked role containing *workspaces-web*. For more information, see the section called “Using service-linked roles”.
- Console URLs containing *workspaces-web*.
- Documentation URLs containing *workspaces-web*. For more information, see Amazon WorkSpaces Secure Browser Documentation.
- Existing ReadOnly managed role. For more information, see the section called “AWS managed policies”.
- KMS grant name.
- UAL(User-Activity Logging) Kinesis stream prefix.

In addition, existing portal URLs remain the same. URLs for portals created before May 20, 2024 used the format <UUID>.workspaces-web.com. WorkSpaces Secure Browser portals continue to use this format and the workspaces-web.com domain.

Terms to know when using Amazon WorkSpaces Secure Browser

To help you get started with WorkSpaces Secure Browser, 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 Secure Browser.

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 Secure Browser in your IdP.

Identity provider (IdP) metadata document

WorkSpaces Secure Browser requires specific metadata from your identity provider (IdP) to establish trust. You can add this metadata to WorkSpaces Secure Browser 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 Secure Browser 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 Secure Browser.

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 Secure Browser console.

Trust store

If a user accessing a web site through WorkSpaces Secure Browser 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 Secure Browser.

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.

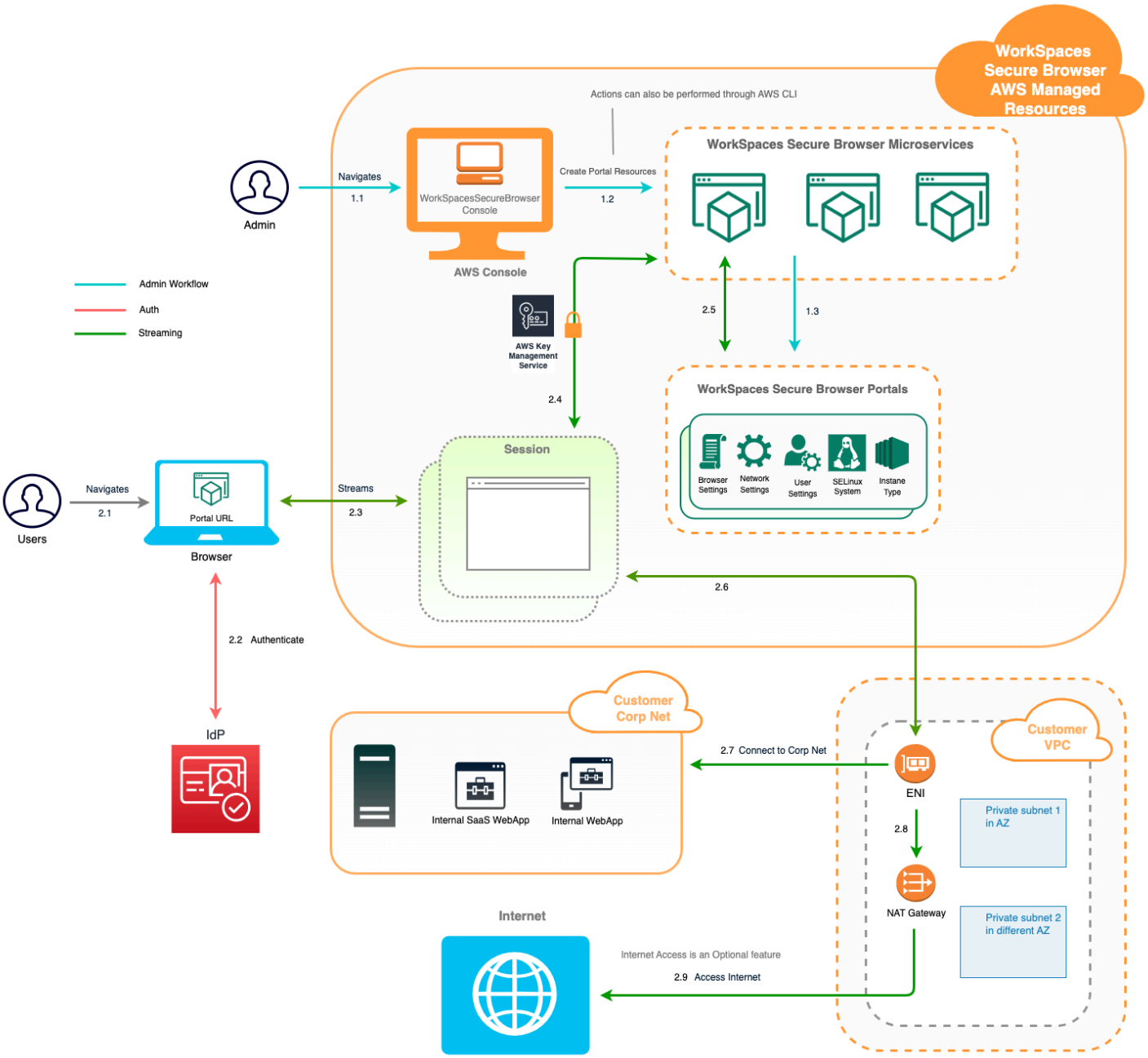
AWS services related to Amazon WorkSpaces Secure Browser

There are several AWS services that are related to WorkSpaces Secure Browser.

WorkSpaces Secure Browser is a capability from Amazon WorkSpaces in the AWS End User Computing portfolio. Compared with WorkSpaces and AppStream 2.0, WorkSpaces Secure Browser is built specifically to facilitate secure, web-based workloads. WorkSpaces Secure Browser 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 WorkSpaces Secure Browser 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.

Architecture of Amazon WorkSpaces Secure Browser

The following diagram shows the architecture of WorkSpaces Secure Browser.



Accessing Amazon WorkSpaces Secure Browser

You can access WorkSpaces Secure Browser in several ways.

Administrators access WorkSpaces Secure Browser through the WorkSpaces Secure Browser Console, SDK, CLI, or API. Your users access it through the WorkSpaces Secure Browser endpoint.

Setting up Amazon WorkSpaces Secure Browser

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

Topics

Signing up and creating a user

Sign up for an AWS account

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

To sign up for an AWS account

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

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

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

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

Create a user with administrative access

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

Secure your AWS account root user

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

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

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

Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

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

Sign in as the user with administrative access

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

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

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.

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

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

Granting programmatic access

Users need programmatic access if they want to interact with AWS outside of the AWS Management Console. The way to grant programmatic access depends on the type of user that's accessing AWS.

To grant users programmatic access, choose one of the following options.

Which user needs programmatic access?	To	By
Workforce identity (Users managed in IAM Identity Center)	Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions for the interface that you want to use. • For the AWS CLI, see Configuring the AWS CLI to use AWS IAM Identity Center in the *AWS Command Line Interface User Guide*. • For AWS SDKs, tools, and AWS APIs, see IAM Identity Center authentication in the *AWS SDKs and Tools Reference Guide*.
IAM	Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions in Using temporary credentials with AWS resources in the *IAM User Guide*.

Which user needs programmatic access?	To	By
IAM	(Not recommended) Use long-term credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions for the interface that you want to use. • For the AWS CLI, see Authenticating using IAM user credentials in the *AWS Command Line Interface User Guide*. • For AWS SDKs and tools, see Authenticate using long-term credentials in the *AWS SDKs and Tools Reference Guide*. • For AWS APIs, see Managing access keys for IAM users in the *IAM User Guide*.

Networking for Amazon WorkSpaces Secure Browser

The following topics explain how to set up WorkSpaces Secure Browser streaming instances so that users can connect to them. It also explains how to enable your WorkSpaces Secure Browser streaming instances to access VPC resources, as well as the internet.

Topics

Setting up a VPC for Amazon WorkSpaces Secure Browser

To set up and configure a VPC for WorkSpaces Secure Browser complete the following steps.

Topics

VPC requirements for Amazon WorkSpaces Secure Browser

During WorkSpaces Secure Browser portal creation, you'll select a VPC in your account. You'll also choose at least two subnets in two different Availability Zones. These VPCs and subnets must meet following requirements:

- The VPC must have default tenancy. VPCs with dedicated tenancy are not supported.
- For availability consideration, we require at least two subnets created in two different Availability Zones. Your subnets must have sufficient IP addresses to support the expected WorkSpaces Secure Browser traffic. Configure each of your subnets with a subnet mask that allows for enough client IP addresses to account for the maximum number of concurrent sessions. For more information, see Creating a new VPC for Amazon WorkSpaces Secure Browser.
- All subnets must have a stable connection to any internal content, either located in the AWS Cloud or on premises, that users will access with WorkSpaces Secure Browser.

We recommend you choose three subnets in different Availability Zones for availability and scaling consideration. For more information, see Creating a new VPC for Amazon WorkSpaces Secure Browser.

WorkSpaces Secure Browser doesn't assign any public IP address to streaming instances to enable internet access. This would make your streaming instances accessible from the internet. Therefore, any streaming instance connected to your public subnet won’t have internet access. If you want your WorkSpaces Secure Browser portal to have access to both public internet content and private VPC content, complete the steps in Enabling unrestricted internet browsing for Amazon WorkSpaces Secure Browser (recommended).

Creating a new VPC for Amazon WorkSpaces Secure Browser

This section describes how to use the VPC wizard to create a VPC with one 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. It then updates the main route table associated with the private subnet. The NAT gateway is automatically created in your VPC's public subnet.

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

Topics

Allocating an Elastic IP address

Before you create your VPC, you must allocate an Elastic IP address in your WorkSpaces Secure Browser Region. Once allocated, you can associate the Elastic IP address with your NAT gateway. With an Elastic IP address, you can mask a failure of your 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, you must first verify that it isn't 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 shown on the console.
5. In the upper-right corner of the **Elastic IPs** pane, click the × icon to close the pane.

Creating 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**, enter 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**, enter a name for the subnet. For example, **`WorkSpaces Secure Browser 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**, enter a name for the subnet. For example, **`WorkSpaces Secure Browser Private Subnet1`**.
   - For the remaining fields, keep the default values where applicable.

- For **Elastic IP Allocation ID**, enter the value that corresponds to the Elastic IP address that you created. This address is then 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.

  To specify an Amazon S3 endpoint, do the following:

  1. Choose **Add Endpoint**.
  2. For **Service**, select the **com.amazonaws.*Region*.s3** entry, where *Region* is the AWS Region you're creating your VPC in.
  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**.

Adding 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 to your VPC. 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**, enter a name for the private subnet. For example, **`WorkSpaces Secure Browser 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're 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 second private subnet.
8. Choose **Create**.
9. After your subnet is created, choose **Close**.

Verifying and naming your subnet route tables

After you've created and configured your VPC, complete the following steps to specify a name for your route tables. You'll need to verify that the following details are correct for your route table:

- The route table associated with the subnet that your NAT gateway resides in must include 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 must be 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 then select the public subnet that you created. For example, **WorkSpaces Secure Browser 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 (pencil) icon, and enter a name for the table. For example, enter the name **`workspacesweb-public-routetable`**. Then select the check mark to save the name.
4. With the public route table still selected, on the **Routes** tab, verify that there are two routes: one for local traffic, and one that sends all other traffic through the VPC's internet gateway. 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. This traffic 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**. Then, select the first private subnet that you created (for example, **`WorkSpaces Secure Browser Private Subnet1`**).
6. On the **Route Table** tab, choose the route table's ID.
7. Select the route table. Under **Name**, choose the edit (pencil) icon, and enter a name for the table. For example, enter the name **`workspacesweb-private-routetable`**. 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

Destination	Target	Description
		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**. Then select the second private subnet that you created (for example, **`WorkSpaces Secure Browser Private Subnet2`**).
10. On the **Route Table** tab, verify that the selected route table is the private route table (for example, **`workspacesweb-private-routetable`**). If the route table is different, choose **Edit** and select your private route table instead.

Enabling internet browsing for Amazon WorkSpaces Secure Browser

You can choose to enable unrestricted internet browsing (the recommended option) or restricted internet browsing.

Topics

- Enabling unrestricted internet browsing for Amazon WorkSpaces Secure Browser (recommended)
- Enabling restricted internet browsing for Amazon WorkSpaces Secure Browser
- Internet connectivity ports for Amazon WorkSpaces Secure Browser

Enabling unrestricted internet browsing for Amazon WorkSpaces Secure Browser (recommended)

Follow these steps to configure a VPC with a NAT gateway for unrestricted internet browsing. This grants WorkSpaces Secure Browser access to sites on the public internet, and private sites hosted in or with a connection to your VPC.

To configure a VPC with a NAT gateway for unrestricted internet browsing

If you want your WorkSpaces Secure Browser portal to have access to both public internet content and private VPC content, follow these steps:

> **Note**
>
> If you 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 Creating a new VPC for Amazon WorkSpaces Secure Browser.

1. To create your NAT gateway, complete the steps in Create a NAT gateway. Make sure that this NAT gateway has public connectivity, and is in a public subnet in your VPC.
2. You must specify at least two private subnets from different Availability Zones. Assigning your subnets to different Availability Zones helps to ensure better availability and fault tolerance. For information about how to create a second private subnet, see the section called "Second private subnet".

   > **Note**
   >
   > To make sure every streaming instance has internet access, do not attach a public subnet to your WorkSpaces Secure Browser portal.

3. Update the route table associated with 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. For information on how to associate a route table with a private subnet, complete the steps in Configure route tables.

Enabling restricted internet browsing for Amazon WorkSpaces Secure Browser

The recommended network setup of a WorkSpaces Secure Browser portal is to use private subnets with NAT gateway, so that the portal can browse both public internet and private content. For more information, see the section called "Unrestricted internet browsing". However, you might be required to control outbound communication from a WorkSpaces Secure Browser portal to the internet by using a web proxy. For example, if you use a web proxy as the gateway to the internet, you can implement preventive security controls, such as domain allow-listing and content filtering. This can also reduce bandwidth usage and improve network performance by caching frequently accessed resources, such as web pages or software updates locally. For some use cases, you might have private content that is only accessible by using a web proxy.

You might already be familiar with configuring proxy settings on managed devices, or on the image of your virtual environments. But this poses challenges if you aren't in control of the device (for example, when users are on devices not owned or managed by the enterprise), or if you need to manage the image for your virtual environment. With WorkSpaces Secure Browser, you can set proxy settings using Chrome's policies built into the web browser. You can do this by setting up an HTTP outbound proxy for WorkSpaces Secure Browser.

This solution is based on a recommended outbound VPC proxy setup. The proxy solution is based on the open source HTTP proxy Squid. Then, it uses WorkSpaces Secure Browser browser settings to configure WorkSpaces Secure Browser portal to connect to the proxy endpoint. For more information, see How to set up an outbound VPC proxy with domain whitelisting and content filtering.

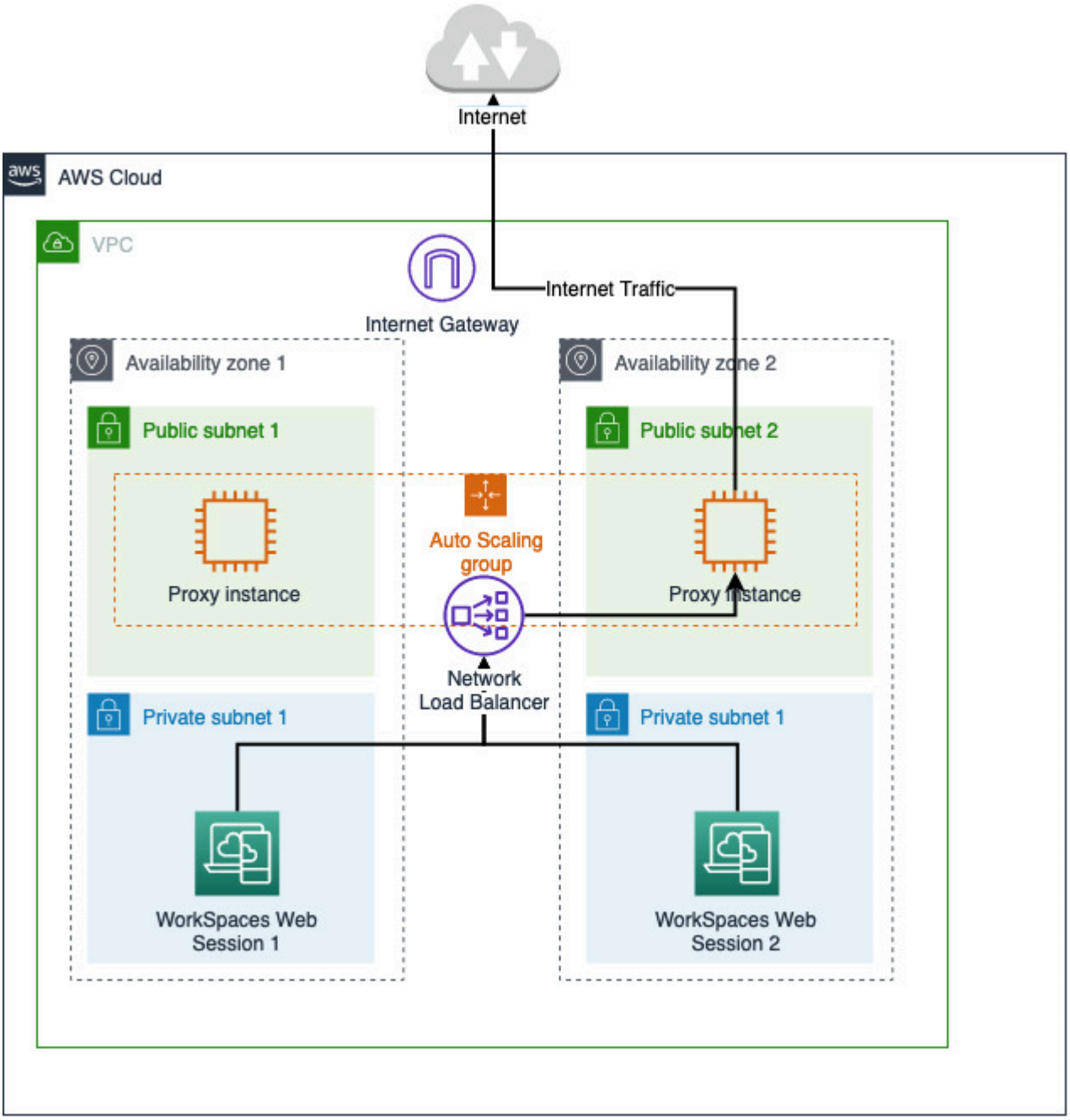
This solution provides you with the following benefits:

- An outbound proxy that includes a group of auto-scaling Amazon EC2 instances, hosted by a network load balancer. Proxy instances live in a public subnet, and each of them is attached with an Elastic IP, so they can have access to the internet.
- A WorkSpaces Secure Browser portal deployed to private subnets. You don't need to configure NAT gateway to enable internet access. Instead, you configure your browser policy, so all internet traffic goes through the outbound proxy. If you want to use your own proxy, the WorkSpaces Secure Browser portal setup will be similar.

Topics

Restricted internet browsing architecture for Amazon WorkSpaces Secure Browser

The following is an example of a typical proxy setup in your VPC. The proxy Amazon EC2 instance is in public subnets and associated with Elastic IP, so they have access to internet. A network load balancer hosts an auto scaling group of proxy instances. This ensures that proxy instances can scale up automatically, and the network load balancer is the single proxy endpoint, which can be consumed by WorkSpaces Secure Browser sessions.



Restricted internet browsing prerequisites for Amazon WorkSpaces Secure Browser

Before you get started, make sure that you meet the following prerequisites:

- You need an already deployed VPC, with public and private subnets spreading over several Availability Zones (AZs). For more information about how to set up your VPC environment, see Default VPCs.

- You need one single proxy endpoint that is accessible from private subnets, where WorkSpaces Secure Browser sessions live (for example, the network load balancer DNS name). If you want to use your existing proxy, make sure it also has a single endpoint that is accessible from your private subnets.

HTTP outbound proxy for Amazon WorkSpaces Secure Browser

To set up an HTTP outbound proxy for WorkSpaces Secure Browser, follow these steps.

1. To deploy an example outbound proxy to your VPC, follow the steps in How to set up an outbound VPC proxy with domain whitelisting and content filtering.
   a. Follow the steps in "Installation (one-time setup)" to deploy the CloudFormation template to your account. Make sure to choose the right VPC and subnets as the CloudFormation template parameters.
   b. After deployment, find the CloudFormation output parameter **OutboundProxyDomain** and **OutboundProxyPort**. This is your proxy's DNS name and port.
   c. If you already have your own proxy, skip this step and use your proxy's DNS name and port.
2. In the WorkSpaces Secure Browser, console, select your portal and then choose **Edit**.
   a. In the **Network connection details**, choose the VPC and private subnets that have access to the proxy.
   b. In the **Policy settings**, add the following ProxySettings policy by using a JSON editor. The `ProxyServer` field should be your proxy's DNS name and port. For more details about ProxySettings policy, see ProxySettings.

```
{
    "chromePolicies":
    {
        ...
        "ProxySettings": {
            "value": {
                "ProxyMode": "fixed_servers",
                "ProxyServer": "OutboundProxyLoadBalancer-0a01409a46943c47.elb.us-
west-2.amazonaws.com:3128",
                "ProxyBypassList": "https://www.example1.com,https://
www.example2.com,https://internalsite/"
            }
        },
    }
```

```
}
```

3. In your WorkSpaces Secure Browser session, you will see the proxy is applied to Chrome setting **Chrome is using proxy settings from your administrator**.
4. Go to chrome://policy and the **Chrome policy** tab to confirm that the policy is applied.
5. Verify that your WorkSpaces Secure Browser session can successfully browse internet content without NAT gateway. In the CloudWatch Logs, verify that Squid proxy access logs are recorded.

Troubleshooting restricted internet browsing for Amazon WorkSpaces Secure Browser

After Chrome policy is applied, if your WorkSpaces Secure Browser session still can't access the internet, follow these steps to try to resolve your issue:

- Verify that the proxy endpoint is accessible from the private subnets where your WorkSpaces Secure Browser portal lives. To do you this, create an EC2 instance in the private subnet, and test the connection from the private EC2 instance to your proxy endpoint.
- Verify that the proxy has internet access.
- Verify that the Chrome policy is correct.
  - Confirm the following formatting for the `ProxyServer` field of the policy: `<Proxy DNS name>:<Proxy port>`. There should be no `http://` or `https://` in the prefix.
  - In the WorkSpaces Secure Browser session, use Chrome to navigate to chrome://policy, and make sure that the ProxySettings policy is successfully applied.

Internet connectivity ports for Amazon WorkSpaces Secure Browser

Each WorkSpaces Secure Browser streaming instance has a customer network interface that provides connectivity to the resources within your VPC, as well as to the internet if private subnets with NAT gateway are set up.

For internet connectivity, the following ports must be open to all destinations. If you are using a modified or custom security group, you'll need to add the required rules manually. For more information, see Security group rules.

> **Note**
>
> This applies to egress traffic.

- TCP 80 (HTTP)
- TCP 443 (HTTPS)
- UDP 8433

VPC best practices for WorkSpaces Secure Browser

The following recommendations can help you configure your VPC more effectively and securely.

Overall VPC Configuration

- Make sure that your VPC configuration can support your scaling needs.
- Make sure that your WorkSpaces Secure Browser service quotas (also referred to as limits) are sufficient to support your anticipated demand. To request a quota increase, you can use the Service Quotas console at https://console.aws.amazon.com/servicequotas/. For information about default WorkSpaces Secure Browser quotas, see the section called “Managing service quotas”.
- If you plan to provide your streaming sessions with access to the internet, we recommend that you configure a VPC with a NAT gateway in a public subnet.

Elastic Network Interfaces

- Each WorkSpaces Secure Browser session requires its own elastic network interface during the streaming duration. WorkSpaces Secure Browser creates as many elastic network interfaces (ENIs) as the maximum desired capacity of your fleet. By default, the limit for ENIs per Region is 5000. For more information, see Network interfaces.

  When planning capacity for very large deployments, for example, thousands of concurrent streaming sessions, consider the number of ENIs that might be required for your peak usage. We recommend that you keep your ENI limit at or above the max concurrent usage limit you configure for your web portal.

Subnets

- As you develop your plan to scale up users, keep in mind that each WorkSpaces Secure Browser session requires a unique client IP address from your configured subnets. Therefore, the size of the client IP address space configured on your subnets determines the number of users who can stream concurrently.

- We recommend each subnet is configured with a subnet mask that allows for enough client IP addresses to account for the maximum number of expected concurrent users. In addition, consider adding additional IP addresses to account for anticipated growth. For more information, see VPC and Subnet Sizing for IPv4.
- We recommend that you configure a subnet in each unique Availability Zone that WorkSpaces Secure Browser supports in your desired region for availability and scaling consideration. For more information, see the section called “Creating a new VPC”.
- Make sure that the network resources required for your web applications are accessible through your subnets.

Security Groups

- Use security groups to provide additional access control to your VPC.

  Security groups that belong to your VPC let you control the network traffic between WorkSpaces Secure Browser streaming instances and network resources required by web applications. Make sure that the security groups provide access to the network resources that your web applications require.

Supported Availability Zones for Amazon WorkSpaces Secure Browser

When you are creating a virtual private cloud (VPC) for use with WorkSpaces Secure Browser, your VPC's subnets must reside in different Availability Zones in the Region where you're launching WorkSpaces Secure Browser. Availability Zones are distinct locations that are engineered to be isolated from failures in other Availability Zones. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location. Each subnet must reside entirely within one Availability Zone and cannot span zones. We recommend configuring a subnet for each supported AZ in your desired region for maximum resiliency

An Availability Zone is represented by a Region code followed by a letter identifier; for example, `us-east-1a`. To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to names for each AWS account. For example, the Availability Zone `us-east-1a` for your AWS account might not be the same location as `us-east-1a` for another AWS account.

To coordinate Availability Zones across accounts, you must use the *AZ ID*, which is a unique and consistent identifier for an Availability Zone. For example, `use1-az2` is an AZ ID for the `us-east-1` Region and it has the same location in every AWS account.

Viewing AZ IDs enables you to determine the location of resources in one account relative to the resources in another account. For example, if you share a subnet in the Availability Zone with the AZ ID `use1-az2` with another account, this subnet is available to that account in the Availability Zone whose AZ ID is also `use1-az2`. The AZ ID for each VPC and subnet is displayed in the Amazon VPC console.

WorkSpaces Secure Browser is available in a subset of the Availability Zones for each supported Region. The following table lists the AZ IDs that you can use for each Region. To see the mapping of AZ IDs to Availability Zones in your account, see AZ IDs for Your Resources in the *AWS RAM User Guide*.

Region name	Region code	Supported AZ IDs
US East (N. Virginia)	`us-east-1`	`use1-az1`, `use1-az2`, `use1-az4`, `use1-az5`, `use1-az6`
US West (Oregon)	`us-west-2`	`usw2-az1`, `usw2-az2`, `usw2-az3`
Asia Pacific (Mumbai)	`ap-south-1`	`aps1-az1`, `aps1-az3`
Asia Pacific (Singapore)	`ap-southeast-1`	`apse1-az1` , `apse1-az2` , `apse1-az3`
Asia Pacific (Sydney)	`ap-southeast-2`	`apse2-az1` , `apse2-az2` , `apse2-az3`
Asia Pacific (Tokyo)	`ap-northeast-1`	`apne1-az1` , `apne1-az2` , `apne1-az4`
Canada (Central)	`ca-central-1`	`cac1-az1`, `cac1-az2`, `cac1-az4`
Europe (Frankfurt)	`eu-central-1`	`euc1-az2`, `euc1-az2`, `euc1-az3`

Region name	Region code	Supported AZ IDs
Europe (Ireland)	`eu-west-1`	`euw1-az1`, `euw1-az2`, `euw1-az3`
Europe (London)	`eu-west-2`	`euw2-az1`, `euw2-az2`

For more information about Availability Zones and AZ IDs, see Regions, Availability Zones, and Local Zones in the *Amazon EC2 User Guide*.

Enabling user connections for Amazon WorkSpaces Secure Browser

WorkSpaces Secure Browser is configured to route streaming connections over the public internet. Internet connectivity is required to authenticate users and deliver the web assets that WorkSpaces Secure Browser requires to function. To allow this traffic, you must allow the domains listed in Allowed domains for Amazon WorkSpaces Secure Browser.

The following topics provide information about how to enable user connections to WorkSpaces Secure Browser.

Topics

- IP address and port requirements for Amazon WorkSpaces Secure Browser
- Allowed domains for Amazon WorkSpaces Secure Browser

IP address and port requirements for Amazon WorkSpaces Secure Browser

To access WorkSpaces Secure Browser instances, user devices require outbound access on the following ports:

- Port 443 (TCP)
  - Port 443 is used for HTTPS communication between user devices and streaming instances when using the internet endpoints. Typically, when end users browse the web during streaming sessions, the web browser randomly selects a source port in the high range for streaming traffic. You must ensure that return traffic to this port is allowed.
  - This port must be open to the required domains listed at Allowed domains for Amazon WorkSpaces Secure Browser.

- AWS publishes its current IP address ranges, including the ranges that the Session Gateway and CloudFront domains may resolve to, in JSON format. For information about how to download the .json file and view the current ranges, see AWS IP address ranges. Or, if you are using AWS Tools for Windows PowerShell, you can access the same information by using the **Get-AWSPublicIpAddressRange** PowerShell command. For more information, see Querying the Public IP Address Ranges for AWS.

- (Optional) Port 53 (UDP)
  - Port 53 is used for communication between user devices and your DNS servers.
  - This port is optional if you are not using DNS servers for domain name resolution.
  - The port must be open to the IP addresses for your DNS servers so that public domain names can be resolved.

Allowed domains for Amazon WorkSpaces Secure Browser

For users to be able to access web portals from their local browser, you must add the following domains to the allow list on the network the user is trying to access the service from.

In the following table, replace *{region}* with the code of the operating web portal's Region. For example, s3.*{region}*.amazonaws.com should be s3.eu-west-1.amazonaws.com for a web portal the Europe (Ireland) region. For a list of Region codes, see Amazon WorkSpaces Secure Browser endpoints and quotas.

Category	Domain or IP address
WorkSpaces Secure Browser streaming assets	s3.*{region}*.amazonaws.com s3.amazonaws.com appstream2.*{region}*.aws.amazon.com *.amazonappstream.com *.shortbread.aws.dev
WorkSpaces Secure Browser static assets	*.workspaces-web.com di5ry4hb4263e.cloudfront.net

Category	Domain or IP address
WorkSpaces Secure Browser authentication	*.auth.*{region}*.amazoncognito.com cognito-identity.*{region}*.amazonaws.com cognito-idp.*{region}*.amazonaws.com *.cloudfront.net
WorkSpaces Secure Browser metrics and reporting	*.execute-api.*{region}*.amazonaws.com unagi-na.amazon.com

Depending on your configured identity provider, you might also need to allow list additional domains. Review your IdP's documentation to identify which domains you need to allow list in order for WorkSpaces Secure Browser to use that provider. If you are using IAM Identity Center, see IAM Identity Center prerequisites for more information.

Getting started with Amazon WorkSpaces Secure Browser

Follow these steps to create a WorkSpaces Secure Browser 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 AWS Region.

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

There are no costs associated with setting up a web portal. WorkSpaces Secure Browser 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.

> **Important**
>
> Before you begin, you must complete the necessary prerequisites for a web portal. For more information about web portal prerequisites, see Setting up Amazon WorkSpaces Secure Browser.

Topics

Creating a web portal for Amazon WorkSpaces Secure Browser

Follow these steps to create a web portal.

Topics

Configuring network settings for Amazon WorkSpaces Secure Browser

To configuring network settings for WorkSpaces Secure Browser follow these steps.

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home.
2. Choose **WorkSpaces Secure Browser**, then **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.

    1. For **Networking details**, choose a VPC with a connection to the content you want your users to access with WorkSpaces Secure Browser.
    2. Choose up to three private subnets that meet the following requirements. For more information, see Networking for Amazon WorkSpaces Secure Browser.
        - You must choose a minimum of two private subnets to create a portal.
        - To ensure high availability for your web portal, we recommend you provide the maximum number of private subnets in unique availability zones for your VPC.
    3. Choose a security group.

Configuring portal settings for Amazon WorkSpaces Secure Browser

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

1. Under **Web portal details**, for **Display name**, enter an identifiable name for your web portal.
2. Under **Instance Type**, select the instance type for your web portal from the drop-down menu. Then, enter your **Max concurrent user limit** for the web portal. For more information, see the section called “Managing service quotas”.

> **Note**
>
> Selecting a new instance type will change the cost for each monthly active user. For more information, see Amazon WorkSpaces Secure Browser Pricing.

3. Under **User access logging**, for **Kinesis stream ID**, select the Amazon Kinesis data stream you want to send your data to. For more information, see the section called "Setting up user activity logging".
4. Under **Policy settings**, complete the following:

   - For **Policy options**, select **Visual editor** or **JSON file upload**. You can use either method to provide the policy configuration details for your web portal. For more information, see the section called "Managing browser policy".
     - WorkSpaces Secure Browser includes support for Chrome enterprise policies. You can add and manage policies with 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 can see the available policies in the file in the console. However, you can't edit all policies in the visual editor. The console lists policies in your JSON file that you can't edit with the visual editor under **Additional JSON policies**. To make changes to these policies, you must edit them manually.
   - (Optional) For **Startup URL - optional**, enter a domain to use as the homepage when users launch their browser. Your VPC must have a stable connection to this URL.
   - Select or clear **Private browsing** and **History deletion** to turn these features on or off during a user's session

     > **Note**
     >
     > URLs visited while browsing privately, or before a user deletes their browser history, can't be recorded in user access logging. For more information, see the section called "Setting up user activity logging".

   - Under **URL filtering**, you can configure which URLs users can visit during a session. For more information, see the section called "Setting up URL filtering".
   - (Optional) For **Browser bookmarks - optional**, enter the **Display name**, **Domain**, and **Folder** for any bookmarks you want your users to see in their browser. Then, choose **Add bookmark**.

> **Note**
>
> **Domain** is a required field for browser bookmarks.
> In Chrome, users can find managed bookmarks in the **Managed bookmarks** folder on the bookmarks toolbar.

- (Optional) Add **Tags** to your portal. You can use tags to search for or filter your AWS resources. Tags consist of a key and optional value and are associated with your portal resource.

5. Under **IP Access Control (optional)**, choose whether to restrict access to trusted networks. For more information, see the section called “Managing IP access controls”.
6. Choose **Next** to continue.

Configuring user settings for Amazon WorkSpaces Secure Browser

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. Under **Permissions**, choose whether to enable the extension for single sign-on. For more information, see the section called “Managing the single sign-on extension”.
2. For **Allow users to print to a local device from their web portal**, choose **Allowed** or **Not allowed**.
3. For **Allow users to deeplink to their web portal**, choose **Allowed** or **Not Allowed**. For more information about deep links, see the section called “Deep links”.
4. Under **Toolbar controls**, choose the settings that you want under **Features**.
5. Under **Settings**, manage the toolbar presentation view at start of the session including toolbar state (docked or detached), theme (dark or light mode), icon visibility, and maximum display resolution for the session. Leave these settings unconfigured to grant end users full control over these options. For more information, see the section called “Toolbar controls”.
6. For **Session timeouts**, 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.

Configuring your identity provider for Amazon WorkSpaces Secure Browser

Use the following steps to configure your identity provider (IdP).

Topics

Choosing the identity provider type for Amazon WorkSpaces Secure Browser

WorkSpaces Secure Browser offers two authentication types: **Standard** and **AWS IAM Identity Center**. You choose the authentication type to use with your portal on the **Configure identity provider page**.

- For **Standard** (default option), federate your 3rd party SAML 2.0 identity provider (such as Okta or Ping) directly with your portal. For more information, see the section called "Standard authentication type". The standard type supports both SP-initiated and IdP-initiated authentication flows.
- For **IAM Identity Center** (advanced option), federate the IAM Identity Center with your portal. To use this authentication type, your IAM Identity Center and WorkSpaces Secure Browser portal must both reside in the same AWS Region. For more information, see the section called "IAM Identity Center authentication type".

Topics

- Configuring the standard authentication type for Amazon WorkSpaces Secure Browser
- Configuring the IAM Identity Center authentication type for Amazon WorkSpaces Secure Browser

Configuring the standard authentication type for Amazon WorkSpaces Secure Browser

The *standard* authentication type is the default authentication type. It can support service provider-initiated (SP-initiated) and identity provider-initiated (IdP-initiated) sign-in flows with your SAML 2.0 compliant IdP. To configure the standard authentication type, follow the steps below to federate your third-party SAML 2.0 IdP (such as Okta or Ping) directly with your portal.

Topics

- Configuring your identity provider on Amazon WorkSpaces Secure Browser
- Configuring your IdP on your own IdP
- Finishing IdP configuration on Amazon WorkSpaces Secure Browser
- Guidance for using specific IdPs with Amazon WorkSpaces Secure Browser

Configuring your identity provider on Amazon WorkSpaces Secure Browser

Complete the following steps to configure your identity provider:

1. On the **Configure identity provider page** of the creation wizard, choose **Standard**.
2. Choose **Continue with Standard IdP**.
3. Download the SP metadata file, and keep the tab open for individual metadata values.

- If the SP metadata file is available, choose **Download metadata file** to download the service provider (SP) metadata document, and upload the service provider metadata file to your IdP in the next step. Without this, users won't be able to sign in.
- If your provider doesn't upload SP metadata files, manually enter the metadata values.

4. Under **Choose SAML sign-in type**, choose between **SP-initiated and IdP-initiated SAML assertions**, or **SP-initiated SAML assertions only**.
   - **SP-initiated and IdP-initiated SAML assertions** allow your portal to support both types of sign-in flows. Portals that support IdP-initiated flows allow you to present SAML assertions to the service identity federation endpoint without requiring users to launch a session by visiting the portal URL.
     - Choose this to allow the portal to accept unsolicited IdP-initiated SAML assertions.
     - This option requires a **default Relay State** to be configured in your SAML 2.0 Identity Provider. The Relay state parameter for your portal is in the console under **IdP initiated SAML sign in**, or you can copy it from the SP metadata file under `<md:IdPInitRelayState>`.
     - Note
       - The following is the format of the relay state: `redirect_uri=https%3A%2F%2Fportal-id.workspaces-web.com%2Fsso&response_type=code&client_id=1example23456789&identity_provider=Ex` `Identity-Provider`.
       - If you copy and paste the value from the SP metadata file, make sure that you change `&amp;` to `&`. `&amp;` is an XML escape character.
   - Choose **SP-initiated SAML assertions only** for the portal to only support SP-initiated sign in flows. This option will reject unsolicited SAML assertions from IdP-initiated sign-in flows.

> **Note**
>
> Some third-party IdPs allow you to create a custom SAML application that can deliver IdP-initiated authentication experiences leveraging SP-initiated flows. For example, see Add an Okta bookmark application.

5. Choose whether you want to enable **Sign SAML requests to this provider**. SP-initiated authentication allows your IdP to validate that the authentication request is coming from the portal, which prevents accepting other third-party requests.

a. Download the signing certificate and upload it to your IdP. The same signing certificate can be used for single logout.

b. Enable signed request in your IdP. The name might be different, depending on the IdP.

> **Note**
>
> RSA-SHA256 is the only request and default request signing algorithm supported.

6. Choose whether you want to enable **Require encrypted SAML assertions**. This allows you to encrypt the SAML assertion that comes from your IdP. It can prevent data from being intercepted in SAML assertions between the IdP and WorkSpaces Secure Browser.

> **Note**
>
> The encryption certificate is not available at this step. It will be created after your portal launches. After you launch the portal, download the encryption certificate and upload it to your IdP. Then, enable assertion encryption in your IdP (the name might be different, depending on the IdP.

7. Choose whether you want to enable **Single Logout**. Single logout allows your end users to sign out of both their IdP and WorkSpaces Secure Browser session with a single action.

   a. Download the signing certificate from WorkSpaces Secure Browser and upload it onto your IdP. This is the same signing certificate used for **Request Signing** in the previous step.

   b. Using **Single Logout** requires you to configure a **Single Logout URL** in your SAML 2.0 identity provider. You can find the **Single Logout URL** for your portal in the console under **Service provider (SP) details - Show individual metadata values**, or from the SP metadata file under `<md:SingleLogoutService>`.

   c. Enable **Single Logout** in your IdP. The name might be different, depending on the IdP.

Configuring your IdP on your own IdP

To configure your IdP on your own IdP, follow these steps.

1. Open a new tab in your browser.
2. Add your portal metadata to your SAML IdP.

Either upload the SP metadata document that you downloaded in the previous step to your IdP, or copy and paste the metadata values into the correct fields in your IdP. Some providers do not allow file upload.

The details of this process can vary between providers. Find your provider's documentation in the section called “Guidance for specific IdPs” for help on how to add the portal details to your IdP configuration.

3. Confirm the **NameID** for your SAML assertion.

   Make sure your SAML IdP populates **NameID** in the SAML assertion with the user email field. **NameID** and user email are used for uniquely identifying your SAML federated user with the portal. Use the persistent SAML Name ID format.

4. Optional: Configure the **Relay State** for IdP-initiated authentication.

   If you chose **Accept SP-initiated and IdP-initiated SAML assertions** in the previous step, follow steps in step 2 of the section called “IdP configuration on WorkSpaces Secure Browser” to set the default **Relay State** for your IdP application.

5. Optional: Configure **Request signing**. If you chose **Sign SAML requests to this provider** in the previous step, follow steps in step 3 of the section called “IdP configuration on WorkSpaces Secure Browser” to upload the signing certificate onto your IdP and enable request signing. Some IdPs such as Okta might require your **NameID** to belong to the “persistent” type to use **Request signing**. Make sure to confirm your **NameID** for your SAML assertion by following the steps above.

6. Optional: Configure **Assertion encryption**. If you chose **Require encrypted SAML assertions from this provider**, wait until portal creation is complete, then follow step 4 in "Upload metadata" below to upload the encryption certificate onto your IdP and enable assertion encryption.

7. Optional: Configure **Single Logout**. If you chose **Single Logout**, follow the steps in step 5 of the section called “IdP configuration on WorkSpaces Secure Browser” to upload the signing certificate onto your IdP, fill in **Single Logout URL**, and enable **Single Logout**.

8. Grant access to your users in your IdP to use WorkSpaces Secure Browser.

9. Download a metadata exchange file from your IdP. You will upload this metadata to WorkSpaces Secure Browser in the next step.

Finishing IdP configuration on Amazon WorkSpaces Secure Browser

To finish IdP configuration on WorkSpaces Secure Browser follow these steps.

1. Return to the WorkSpaces Secure Browserconsole. On the **Configure identity provider page** of the creation wizard, under **IdP metadata**, either upload a metadata file, or enter a metadata URL from your IdP. The portal uses this metadata from your IdP to establish trust.
2. To upload a metadata file, under **IdP metadata document**, choose **Choose file**. Upload the XML-formatted metadata file from your IdP that you downloaded in the previous step.
3. To use a metadata URL, go to your IdP that you set up in the previous step and obtain its **Metadata URL**. Go back to the WorkSpaces Secure Browser console, and under **IdP metadata URL**, enter the metadata url that you obtained from your IdP.
4. When you are done, choose **Next**.
5. For portals where you have enabled the **Require encrypted SAML assertions from this provider** option, you need to download the encryption certificate from the portal IdP details section and upload it onto your IdP. Then, you can enable the option there.

> **Note**
>
> WorkSpaces Secure Browser 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 aren't configured correctly, your users can't sign in to the web portal and start a session.
> WorkSpaces Secure Browser requires the following claims to be present in the SAML response. You can find *<Your SP Entity ID>* and *<Your SP ACS URL>* from your portal's service provider details or metadata document, either through the console or the CLI.
>
> - An `AudienceRestriction` claim with an `Audience` value that sets your SP Entity ID as the target of the response. Example:
>
>   ```
>   <saml:AudienceRestriction>
>       <saml:Audience><Your SP Entity ID></saml:Audience>
>   </saml:AudienceRestriction>
>   ```
>
> - A `Response` claim with an `InResponseTo` value of the original SAML request ID. Example:

```
<samlp:Response ... InResponseTo="<originalSAMLrequestId>">
```

- A `SubjectConfirmationData` claim with a `Recipient` value of your SP ACS URL, and an `InResponseTo` value that matches the original SAML request ID. Example:

```
<saml:SubjectConfirmation>
    <saml:SubjectConfirmationData ...
        Recipient="<Your SP ACS URL>"
        InResponseTo="<originalSAMLrequestId>"
        />
</saml:SubjectConfirmation>
```

WorkSpaces Secure Browser validates your request parameters and SAML assertions. For IdP-initiated SAML assertions, the details of your request must be formatted as a `RelayState` parameter in the body of an HTTP POST request. The request body must also contain your SAML assertion as a `SAMLResponse` parameter. Both of these should be present if you have followed the previous step.
The following is an example POST body for an IdP-initiated SAML provider.

```
SAMLResponse=<Base64-encoded SAML assertion>&RelayState=<RelayState>
```

Guidance for using specific IdPs with Amazon WorkSpaces Secure Browser

To make sure you correctly configure the SAML federation for your portal, see the links below for documentation from commonly used IdPs.

IdP	SAML application setup	User management	IdP-initiated auth	Request signing	Assertion encryption	Single logout
Okta	Create SAML app integrations	User management	Application Integration Wizard SAML field reference	Application Integration Wizard SAML field reference	Application Integration Wizard SAML field reference	Application Integration Wizard SAML field reference

IdP	SAML application setup	User management	IdP-initiated auth	Request signing	Assertion encryption	Single logout
Entra	Create your own application	Quickstart: Create and assign a user account	Enable single sign-on for an enterprise application	SAML Request Signature Verification	Configure Microsoft Entra SAML token encryption	Single Sign-Out SAML Protocol
Ping	Add a SAML application	Users	Enabling IdP-initiated SSO	Configuring authentication request signing in PingOne for Enterprise	Does PingOne for Enterprise support encryption?	SAML 2.0 single logout
One Login	SAML Custom Connector (Advanced) (4266907)	Add Users to OneLogin Manually	SAML Custom Connector (Advanced) (4266907)	SAML Custom Connector (Advanced) (4266907)	SAML Custom Connector (Advanced) (4266907)	SAML Custom Connector (Advanced) (4266907)
IAM Identity Center	Set up your own SAML 2.0 application	Set up your own SAML 2.0 application	Set up your own SAML 2.0 application	N/A	N/A	N/A

Configuring the IAM Identity Center authentication type for Amazon WorkSpaces Secure Browser

For the **IAM Identity Center** type (advanced), you federate IAM Identity Center with your portal. Only select this option if the following applies to you:

- Your IAM Identity Center is configured in the same AWS account and AWS Region as your web portal.
- If you are using AWS Organizations, you are using a management account.

Before creating a web portal with the IAM Identity Center authentication type, you must set up IAM Identity Center as a standalone provider. For more information, see Get started with common tasks in IAM Identity Center. Or, you can connect your SAML 2.0 IdP to IAM Identity Center. For more information, see Connect to an external identity provider. Otherwise, you won't have any users or groups to assign to your web portal.

If you are already using IAM Identity Center, you can choose IAM Identity Center as a provider type and follow the steps below to add, view, or remove users or groups from your web portal.

> **Note**
>
> In order to use this authentication type, your IAM Identity Center needs to be in the same AWS account and AWS Region as your WorkSpaces Secure Browser portal. If your IAM Identity Center is in a separate AWS account or AWS Region, follow the instructions for the **Standard** authentication type. For more information, see the section called “Standard authentication type”.
>
> If you're using AWS Organizations, you can only create WorkSpaces Secure Browser portals integrated with IAM Identity Center using a management account.

Topics

Creating a web portal with IAM Identity Center

To create a web portal with IAM Identity Center, follow these steps.

To create a web portal with IAM Identity Center

1. During portal creation at **Step 4: Configure identity provider**, choose **AWS IAM Identity Center**.
2. Choose **Continue with IAM Identity Center**.
3. On the **Assign users and groups** page, choose the **Users** and/or **Groups** tab.
4. Check the box next to the user(s) or group(s) that you want to add to the portal.
5. After you create your portal, the users that you associated can sign into WorkSpaces Secure Browser with their IAM Identity Center user name and password.

Managing your web portal with IAM Identity Center

To manage your web portal with IAM Identity Center, follow these steps.

To manage your web portal with IAM Identity Center

1. After you create your portal, it is listed in the IAM Identity Center console as a configured application.
2. To access this application's configuration, choose **Applications** in the sidebar, and look for a configured application with a name that matches the display name for your web portal.

> **Note**
>
> If you haven't entered a display name, your portal's GUID is shown instead. The GUID is the ID that is prefixed to your web portal's endpoint URL.

Adding additional users and groups to a web portal

To add additional users and groups to an existing web portal, follow these steps.

To add additional users and groups to an existing web portal

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

3. Choose **Identity provider settings** and **Assign additional users and groups**. From here, you can add users and groups to your web portal.

> **Note**
>
> You can't add users or groups from the IAM Identity Center console. You must do this from the edit page of your WorkSpaces Secure Browser portal.

Viewing or removing users and groups for your web portal

To view or remove users and groups for your web portal, use the actions available in the **Assigned users** table. For more information, see Manage access to applications

> **Note**
>
> You can't view or remove users and groups from the edit page of the WorkSpaces Secure Browserportal. You must do this from the edit page of your IAM Identity Center console.

Changing the identity provider type for Amazon WorkSpaces Secure Browser

You can change the authentication type of your portal at any time. To do this, follow these steps.

- To change from **IAM Identity Center** to **Standard**, follow the steps at the section called "Standard authentication type".
- To change from **Standard** to **IAM Identity Center**, follow the steps at the section called "IAM Identity Center authentication type".

Changes to the identity provider type may take up to 15 minutes to deploy, and will not automatically terminate in-progress sessions.

You can view identity provider type changes to your portal through AWS CloudTrail by inspecting `UpdatePortal` events. The type is visible in the request and response payloads of the event.

Launching a web portal with Amazon WorkSpaces Secure Browser

When you are finished configuring your web portal, you can follow these steps to launch it.

1. On the **Step 5: Review and launch** page, review the settings you selected for your web portal. You can choose **Edit** to changes settings within a given section. You can also change these settings later on from the **Web portals** tab of the console.
2. When you're done, choose **Launch web portal**.
3. To view the status of your web portal, choose **Web portals**, choose your portal, and then choose **View details**.

   A web portal has 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.
4. Wait up to 15 minutes for your portal to become **Active**.

Testing your web portal in Amazon WorkSpaces Secure Browser

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

If you already completed these steps in the section called “Identity provider configuration”, you can skip this section and go to Distributing your web portal in Amazon WorkSpaces Secure Browser.

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. Choose **WorkSpaces Secure Browser**, **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 Secure Browser 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 Secure Browser 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.

Distributing your web portal in Amazon WorkSpaces Secure Browser

When you are ready for your users to begin using WorkSpaces Secure Browser, you choose from the following options to distribute the portal:

- Add your portal to your SAML application gateway to enable users to launch a session from their IdP directly. You can do this through the IdP-initiated sign-in flow with your SAML 2.0 compliant IdP. For more information, see **SP-initiated and IdP-initiated SAML assertions** in the section called "Standard authentication type". Alternatively, you can create a custom SAML application that can deliver IdP-initiated authentication experiences by using SP-initiated flows. For more information, see Create a Bookmark App integration.
- Add the portal URL to a website that you own, and use a browser redirect to direct users to the web portal.
- Email the portal URL to your users, or push down to a device you manage as a browser home page or bookmark.

Managing your web portal in Amazon WorkSpaces Secure Browser

After you set up your web portal, you can perform the following actions to manage it.

Topics

Viewing web portal details in Amazon WorkSpaces Secure Browser

To view web portal details, follow these steps.

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.

2. Choose **WorkSpaces Secure Browser**, **Web portals**, choose your web portal, and then choose **View details**.

Editing a web portal in Amazon WorkSpaces Secure Browser

To edit a web portal, follow these steps.

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

> **Note**
>
> Changes to networking settings or timeout settings immediately end any active portal sessions. Users are disconnected and must reconnect to begin a new session. Changes to **Clipboard permissions**, **File transfer permissions**, or **Print to local device** apply beginning with the first new session. Currently active sessions aren't disconnected. Users connected to active sessions aren't affected by the changes until they disconnect and connect to a new session.

Deleting a web portal in Amazon WorkSpaces Secure Browser

To delete a web portal, follow these steps.

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

Managing service quotas for your portal in Amazon WorkSpaces Secure Browser

When you create your AWS account, we automatically set default service quotas (also referred to as limits) for resource usage with AWS services. Administrators must be aware of two quotas that

might need to be increased to support their use case. These two quotas are the number of web portals you can create in each region, and the number of maximum concurrent sessions you can support with each available instance type in each region. You can request an increase for these from the Service Quotas page in the AWS Console.

The following table lists the default service quotas limits.

Default quotas within an AWS Region by account	Value
Web portals	3
Maximum concurrent sessions - standard.regular	25
Maximum concurrent sessions - standard.large	10
Maximum concurrent sessions - standard.xlarge	5

To view the service quotas allocated to your account for each region at any time, see the Service Quotas page.

Important

Service quotas affect one AWS Region at a time. You must request service quota increases in each AWS Region where you need more resources. For more information, Amazon WorkSpaces Secure Browser endpoints and quotas.

Topics

- Requesting a service quota increase in Amazon WorkSpaces Secure Browser
- Requesting a portal increase in Amazon WorkSpaces Secure Browser
- Requesting a maximum concurrent sessions increase in Amazon WorkSpaces Secure Browser
- Limit example for Amazon WorkSpaces Secure Browser
- Other service quotas in Amazon WorkSpaces Secure Browser

Requesting a service quota increase in Amazon WorkSpaces Secure Browser

To request a service quota increase follow these steps.

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

> **Important**
>
> WorkSpaces Secure Browser 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**.

Requesting a portal increase in Amazon WorkSpaces Secure Browser

A *portal* is the service's foundational resource. Each portal is an association between your SAML 2.0 identity provider and your networking connection to the internet and any private web content. Each portal can have a separate portal browser policy and user settings, so administrators will commonly create multiple portals in the same region to address different use cases. For example, you can provide Group A with access to a specific website with restrictive policies (e.g., Clipboard and File transfer disabled), and Group B with access to the general internet without URL filtering. You can create a portal in any supported AWS Region. To view current service availability, see AWS Services by Region.

To request a service quota increase

1. Open the Service Quotas page in your desired region.
2. Choose **Number of Web Portals**.
3. Choose **Request an increase at account level**.
4. Under **Increase quota value**, enter in the total amount that you want the quota to be.

Requesting a maximum concurrent sessions increase in Amazon WorkSpaces Secure Browser

The *maximum concurrent sessions* quota is the highest amount of users that can be connected at the same time to a portal. If the service quota limit for maximum concurrent sessions is not set appropriately, users may find that a session is not available when they sign in. In addition to increasing this service quota, customers must also ensure that their VPC and subnets have sufficient IP space to support the maximum concurrent sessions.

To request a maximum concurrent session increase

1. Open the Service Quotas page in your desired region.
2. Choose **Number of Maximum Concurrent Sessions per Portal** for the instance type you want to increase.
3. Choose **Request an increase at account level**.
4. Under **Increase quota value**, enter in the total amount that you want the quota to be.

> **Note**
>
> For large or urgent increases, go to your Service Quotas history page, select the link in the status column of your request, link to your support case, and add a reply with details about your use case and/or the urgency. This information helps the service team prioritize requests and ensure sufficient capacity is allocated for your account.

Limit example for Amazon WorkSpaces Secure Browser

As an example, assume an administrator is configuring two web portals in US East (N. Virginia) for 125 total users. Before creating the web portal, the administrator identifies the first web portal (Portal A) will support 100 users. When testing the workflow for these users, the administrator determines they will need the XL instance type to support streaming of audio and video during

the session. The second web portal (Portal B) needs to be available for up to 25 users to support access to a single static webpage hosted in the customers VPC. When testing this use case, the administrator determines that the standard instance type can support this use case.

For portal A, the administrator must submit a service quota increase request to raise the limit for XL instances from the regions default (i.e., 5) to 100. Once fulfilled, the administrator can allocate the capacity by editing the web portal. For portal B, the administrator can move forward without requesting a quota increase (i.e., since the region has a default quota of 25 for standard instance type).

Other service quotas in Amazon WorkSpaces Secure Browser

You can view and request increases for other quotas listed on the Service Quotas page. In practice, most customers will find it unnecessary to request increases for these limits. These quotas are broadly grouped into two types: *Number* and *Rate*.

For Number quotas, when you submit a service quota increase for Number of web portals, you will automatically receive an increase in the number of sub-resources required to create a unique portal. This will be reflected on the Service Quotas page. For example, if you request an increase in portals from 3 to 5, you will automatically receive a service quota increase from 3 to 5 for both browser and user settings. You have the option to re-use or create new sub-resources as desired.

On rare occasion, customers may find a use case for increasing the number or rate of other resource quotas. For example, administrators may want to increase the number of browser settings for testing additional portal configurations. These service quota requests will be reviewed and fulfilled on a case-by-case basis.

For Rate quotas, the rate limits exposed in Service Quotas should not need to be adjusted, regardless of the account portal limit.

Controlling the interval for re-authenticating a SAML IdP token in Amazon WorkSpaces Secure Browser

When a user visits a WorkSpaces Secure Browser 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 sign-in credentials.

If a user has a valid SAML IdP token, they can access WorkSpaces Secure Browser. 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 IAM Identity Center, see Set session duration.
2. Set your WorkSpaces Secure Browser portal's inactivity and idle timeout values. These values controls the amount of time between a user's last interaction and when a WorkSpaces Secure Browser 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 "Web portal creation".

   > **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 Secure Browser session. To control how tokens are re-authenticated, follow the guides in the previous step.

Setting up user activity logging in Amazon WorkSpaces Secure Browser

WorkSpaces Secure Browser offers two options for logging user activity and security-related events:

- Session Logger captures a wide range of session events. These logs are delivered to an Amazon S3 bucket in your account, enabling easy integration with your preferred SIEM platform.
- User Access Logging captures the most critical session events. These logs are streamed to an Amazon Kinesis stream for real-time processing and analysis.

Both logging options are configured at the portal level. You must set up each option individually for every portal where you want logging to be active. You can enable either option or both, depending on your requirements for each portal.

You are responsible for complying with any requirements that apply to the logging or monitoring of user activity when using this feature, including logging or monitoring of employee activity.

Topics

Setting up Session Logger for Amazon WorkSpaces Secure Browser

> **Warning**
>
> Enabling Session Logger disables the following Chrome features:
>
> - Incognito mode
> - Developer Tools
> - Chrome Profile Switching

To activate session logger for a WorkSpaces Secure Browser portal, you must first identify the Amazon S3 bucket where session events will be collected. You can use an existing bucket that already stores similar logs or create a new one specifically for this purpose.

The Amazon S3 bucket must have a bucket policy that grants WorkSpaces Secure Browser permission to write logs to it. We recommend placing the Amazon S3 bucket in the same AWS account and region as your WorkSpaces Secure Browser portal.

There is no naming requirement for the Amazon S3 bucket. To create a new bucket, follow the steps below or see Creating a general purpose bucket in the *Amazon Simple Storage Service User*

Guide. For guidance on configuring permissions, see Bucket policies for Amazon S3 in the *Amazon Simple Storage Service User Guide*.

Below is an example of a policy for your Amazon S3 bucket. Make sure to update the policy with the name of your Amazon S3 bucket. Note that the Principal is "workspaces-web.amazonaws.com".

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
          "Sid": "AllowSessionLogger",
          "Effect": "Allow",
          "Principal": {
              "Service": "workspaces-web.amazonaws.com"
          },
          "Action": [
              "s3:PutObject"
          ],
          "Resource": [
              "arn:aws:s3:::bucket-name/*"
          ]
      }
  ]
}
```

Activating Session Logger on your WorkSpaces Secure Browser portal might result in charges from Amazon S3. For information, see Amazon S3 pricing.

For more information about the session-related events that Session Logger captures, see the section called “Session events in Session Logger”.

S3 buckets with KMS encryption (optional)

WorkSpaces Secure Browser Session Logger fully supports Amazon S3 buckets with AWS KMS encryption enabled. To ensure proper logging functionality with your encrypted Amazon S3 bucket, you must grant Session Logger the necessary permissions to use your AWS KMS key.

Add the following policy to your AWS KMS key configuration:

```
{
  "Sid": "Session Logger",
  "Effect": "Allow",
  "Principal": {
    "Service": "workspaces-web.amazonaws.com"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*"
},
```

In the AWS console, select the WorkSpaces Secure Browser portal you will collect events from, and choose the **Session logger** tab and **Edit**.

Enter the following information to configure Session Logger for the portal:

- **S3 Location (required)**: The name of your Amazon S3 bucket where events will be delivered.
- **Key Prefix (optional)**: The folder where events are delivered. If the folder does not exist, it will be created. If the field is left blank, Session Logger will write events at the root of the Amazon S3 bucket.

Under **Advanced**, you can configure the following fields:

- **Event filter**: This is the list of events monitored by Session Logger.
  - **All**: Selecting this option means all current and future events will be monitored
  - **Include**: This allows you to manually select specific events to monitor. Only the events explicitly selected will be logged. New events introduced in future updates will not be monitored, unless they are manually added to the selection.
- **File format**
  - **JSON (default)**: This is a file format where each log file is presented as an array of events. We recommend this format for most use cases.
  - **JSONLines**: This is a file format that is optimized for Amazon Athena.
- **Folder structure**: This determines how the log files are stored.
  - **Flat (default)**: All log files are in a single folder.

- **Nested By date**: The log files are organized into folders by date and time. Partitioned for Amazon Athena, and optimized for Amazon Athena queries.

You can test the Session Logger setup and ensure that session logger is functioning correctly. Once the configuration is complete, the system attempts to write a test file named `_workspaces_secure_browser.tmp` to the specified Amazon S3 bucket and folder. This serves as a validation of both logging functionality and permission setup.

You can also run a test session by starting a Secure Browser session in the portal and using the browser as you normally would. Session Logger writes log files to your configured Amazon S3 bucket every 15 minutes during an active session, or when the session ends.

After ending the session or waiting for the next logging interval, check the Amazon S3 bucket to confirm that log files for your session have been generated and stored as expected.

Setting up User Access logging for Amazon WorkSpaces Secure Browser

To activate user access logging in the WorkSpaces Secure Browser console, under **User access logging**, select the **Kinesis Stream ID** that you want to use to receive data. The data recorded will be delivered directly to that stream.

For more information about how to create an Amazon Kinesis Data Stream, see What Is Amazon Kinesis Data Streams?

In order to receive logs from WorkSpaces Secure Browser, you must have an Amazon Kinesis Data Stream that starts with "amazon-workspaces-web-*". Your Amazon Kinesis data stream must either have server-side encryption turned off, or must use AWS managed keys for server-side encryption.

For more information about setting server-side encryption in Amazon Kinesis, see How Do I Get Started with Server-Side Encryption?.

Managing browser policy in Amazon WorkSpaces Secure Browser

With WorkSpaces Secure Browser, you can set a custom browser policy using Chrome policies available for the latest stable version. There are more than 300 policies you can apply to a web portal. For more information, see the section called “Tutorial: Setting a custom browser policy” and Chrome Enterprise policy list.

By using the console view to create a web portal, you can apply the following policies:

- `StartURL`
- Bookmarks and bookmark folders
- Turning private browsing on and off
- History deletion
- URL filtering with `AllowURL` and `BlockURL`

For more information about using console view policies, see *Getting started*.

WorkSpaces Secure Browser applies a baseline browser policy configuration to all portals along with any policies that you specify. You can edit some of these policies with your custom JSON file. For more information, see the section called "Editing the baseline browser policy".

Topics

Tutorial: Setting a custom browser policy in Amazon WorkSpaces Secure Browser

You can set any supported Chrome policy for Linux by uploading a JSON file. To learn more about Chrome policies, see Chrome Enterprise policy list and select the Linux platform. Then, search and review the policies for the most recent stable version.

In the following tutorial, you create a web portal with the following policy controls:

- Set up bookmarks
- Set up default startup pages
- Prevent the user from installing other extensions
- Prevent the user from deleting history
- Prevent the user from accessing incognito mode
- Pre-install the Okta plug-in extension for all sessions.

Topics

Step 1: Create a web portal

In order to upload your Chrome policy JSON file, you must create a WorkSpaces Secure Browser portal. For more information, see the section called “Web portal creation”.

Step 2: Gather policies

Search for and locate policies you want from Chrome Policy. You then use the policies to create a JSON file in the next step.

1. Go to Chrome Enterprise policy list.
2. Choose the platform **Linux**, and then choose the most recent Chrome version.
3. Search for the policies you want to set. For this example, search for extensions to find policies for managing them. Each policy includes a description, Linux preference name, and sample value.
4. From the search results, there are 3 policies that meet the business requirements if used together:
   - **ExtensionSettings** – Installs an extension at browser start.
   - **ExtensionInstallBlocklist** – Prevents specific extensions from being installed.
   - **ExtensionInstallAllowlist** – Allows certain extensions to be installed.
5. Additional policies satisfy the remaining requirements;
   - **ManagedBookmarks** – Adds bookmarks to webpages.
   - **RestoreOnStartupURLs** – Configures which webpages are opened whenever a new browser window is launched.
   - **AllowDeletingBrowserHistory** – Configures whether users can delete their browsing history.
   - **IncognitoModeAvailability** – Configures whether users can access incognito mode.

Step 3: Create a custom JSON policy file

Create a JSON file using a text editor, template, and the policies you found in the previous step.

1. Open a text editor.
2. Copy and paste the following template into your text editor:

```
{
  "chromePolicies":
    {
        "ManagedBookmarks":
        {
            "value":
            [
                {
                    "name": "Bookmark 1",
                    "url": "bookmark-url-1"
                },
                {
                    "name": "Bookmark 2",
                    "url": "bookmark-url-2"
                },
            ]
        },
        "RestoreOnStartup":
        {
            "value": 4
        },
        "RestoreOnStartupURLs":
        {
            "value":
            [
                "startup-url"
            ]
        },
        "ExtensionInstallBlocklist": {
            "value": [
                "insert-extensions-value-to-block",
            ]
        },
        "ExtensionInstallAllowlist": {
            "value": [
```

```
                "insert-extensions-value-to-allow",
            ]
        },
        "ExtensionSettings":
        {
            "value":
            {
                "insert-extension-value-to-force-install":
                {
                    "installation_mode": "force_installed",
                    "update_url": "https://clients2.google.com/service/update2/crx",
                    "toolbar_pin": "force_pinned"
                },
            }
        },
        "AllowDeletingBrowserHistory":
        {
            "value": should-allow-history-deletion
        },
        "IncognitoModeAvailability":
        {
            "value": incognito-mode-availability
        }
    }
}
```

Step 4: Add your policies to the template

Add your custom policies to the template for each business requirement.

1. Set up bookmark URLs.
   a. Under the `value` key, add pairs of `name` and `url` keys for each bookmark you want to add.
   b. Set `bookmark-url-1` to `https://www.amazon.com`.
   c. Set `bookmark-url-2` to `https://docs.aws.amazon.com/workspaces-web/latest/adminguide/`.

```
    "ManagedBookmarks":
        {
```

```
            "value":
            [
                {
                    "name": "Amazon",
                    "url": "https//www.amazon.com"
                },
                {
                    "name": "Bookmark 2",
                    "url": "https://docs.aws.amazon.com/workspaces-web/latest/
adminguide/"
                },
            ]
        },
```

2. Set up the startup URLs. This policy allows administrators to set the webpages displayed when a user launches a new browser window.

   a. Set the `RestoreOnStartup` to 4. This sets the `RestoreOnStartup` action to open a list of URLs . You can also use other actions on your startup URLs. For more information, see Chrome Enterprise policy list.

   b. Set `RestoreOnStartupURLs` to https://www.aboutamazon.com/news.

```
    "RestoreOnStartup":
       {
           "value": 4
       },
   "RestoreOnStartupURLs":
       {
           "value":
           [
               "https://www.aboutamazon.com/news"
           ]
       },
```

3. To prevent the user from deleting their browser history, set `AllowDeletingBrowserHistory` to `false`.

```
    "AllowDeletingBrowserHistory":
       {
```

```
            "value": false
        },
```

4. To turn off access to Incognito mode access for your users, set `IncognitoModeAvailability` to 1.

```
    "IncognitoModeAvailability":
        {
            "value": 1
        }
```

5. Set and enforce the Okta plug-in with the following policies:

   - `ExtensionSettings` – Installs an extension at browser start. The extension value is available from the Okta plug-in help page.
   - `ExtensionInstallBlocklist` – Prevents specific extensions from being installed. Use a * value to prevent all extensions by default. Administrators can control which extensions to allow on the `ExtensionInstallAllowlist`.
   - `ExtensionInstallAllowlist` allows you to install certain extensions. Since `ExtensionInstallBlocklist` is set to *, add the Okta plug-in value here to allow it.

   The following shows an example policy to turn on the Okta plug-in:

```
        "ExtensionInstallBlocklist": {
            "value": [
                "*",
                ]
        },
        "ExtensionInstallAllowlist": {
            "value": [
                "glnpjglilkicbckjpbgcfkogebgllemb",
               ]
        },
        "ExtensionSettings": {
            "value": {
                "glnpjglilkicbckjpbgcfkogebgllemb": {
                    "installation_mode": "force_installed",
                    "update_url": "https://clients2.google.com/service/update2/crx",
```

```
                    "toolbar_pin": "force_pinned"
            }
        }
```

Step 5: Upload your policy JSON file to your web portal

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. Choose **WorkSpaces Secure Browser**, then choose **Web portals**.
3. Choose your web portal, and then choose **Edit**.
4. Choose **Policy settings**, then choose **JSON file upload**.
5. Choose **Choose File**. Navigate to, select, and upload your JSON file.
6. Choose **Save**.

Editing the baseline browser policy in Amazon WorkSpaces Secure Browser

In order to deliver the service, WorkSpaces Secure Browser applies a baseline browser policy to all portals. This baseline policy is applied in addition to those you specify from the console view or JSON upload. The following is the list of policies applied by the service in JSON format:

```
{
    "chromePolicies":
    {
        "DefaultDownloadDirectory": {
            "value": "/home/as2-streaming-user/MyFiles/TemporaryFiles"
        },
        "DownloadDirectory": {
            "value": "/home/as2-streaming-user/MyFiles/TemporaryFiles"
        },
        "DownloadRestrictions": {
            "value": 1
        },
        "URLBlocklist": {
            "value": [
```

```
                "file://",
                "http://169.254.169.254",
                "http://[fd00:ec2::254]",
            ]
        },
        "URLAllowlist": {
            "value": [
                "file:///home/as2-streaming-user/MyFiles/TemporaryFiles",
                "file:///opt/appstream/tmp/TemporaryFiles",
            ]
        }
    }
}
```

Customers can't make changes to the following policies:

- `DefaultDownloadDirectory` – This policy can't be edited. The service overwrites any changes to this policy.
- `DownloadDirectory` – This policy can't be edited. The service overwrites any changes to this policy.

Customers can update the following policies for their web portal:

- `DownloadRestrictions` – The default is set to `1` to prevent downloads identified as malicious by Chrome Safe Browsing. For more information, see Prevent users from downloading harmful files. You can set the value from `0` to `4`.
- The `URLAllowlist` and `URLBlocklist` policies can be extended by using the console view URL Filtering feature or JSON upload. However, the baseline URLs can't be overwritten. These policies aren't visible from a JSON file downloaded from your web portal. However, if you visit “chrome://policy” during a session, the remote browser displays the applied policies.

Configuring the Input Method Editor for Amazon WorkSpaces Secure Browser

An Input Method Editor (IME) is a utility that provides options to the end user to input text in languages that use a keyboard layout other than a QWERTY keyboard. IMEs help users enter text in languages with larger, more complex language sets, such as Japanese, Chinese, and Korean.

WorkSpaces Secure Browser sessions include IME support by default. Users can select alternative languages from the IME toolbar in the session or by using keyboard shortcuts.

The following languages are currently supported by WorkSpaces Secure Browser's IME:

- English
- Simplified Chinese (Pinyin)
- Traditional Chinese (Bopomofo)
- Japanese
- Korean

To select a language from the IME toolbar, do the following:

1. Select the language selector drop-down located on the right side of the black, top panel bar. By default, the selector will show **en**, for English.
2. In the dropdown menu, choose the desired language.
3. In the submenu that appears after choosing a language, choose additional language details.

To select a language using keyboard shortcuts, use the following:

- All IMEs
  - To cycle the IME forward (or move to the right keyboard layout), press **Shift**+**Control**+**Left Alt**.
- Japanese
  - To choose Hiragana, press **F6**.
  - To choose Katakana, press **F7**.
  - To choose Latin, press **F10**.
  - To choose Wide Latin, press **F9**.
  - To choose Direct Input, press ALT +, ALT+@, Zenkaku Hankaku.
- Korean
  - To choose Hangul, press **Shift**+**Space**.
  - To choose Hanja, press **F9**.

To remove the IME toolbar and menu, or to turn off the on-screen keyboard from your WorkSpaces Secure Browser sessions, contact Support.

Configuring in-session localization for Amazon WorkSpaces Secure Browser

When a user starts a session, WorkSpaces Secure Browser detects the user's local browser language and time zone settings and applies them to the session. This affects the display language during the session, and helps ensure that the displayed time matches the current time in the user's location.

The session language is determined in the following priority order:

1. The **ForcedLanguages** policy in the web portal's browser settings. For more information, see ForcedLanguages.
2. The end user's local browser language setting.
3. The default value, **English (en-US)**.

The time zone is determined by the local time zone settings specified in the end user's browser. If the time zone setting isn't valid, UTC is used.

The following components in WorkSpaces Secure Browser support localization:

- WorkSpaces Secure Browser sign-in page
- WorkSpaces Secure Browser portal status messages (including loading messages and errors)
- Chrome browser
- System **Context** menu and **Save as** window

Topics

Supported language codes for Amazon WorkSpaces Secure Browser

The following list shows the language codes currently supported by WorkSpaces Secure Browser. If the user's local browser is set to use a language code that isn't supported, the session defaults to English (en-US).

- German

- de – German
- de-AT – German (Austria)
- de-DE – German (Germany)
- de-CH – German (Switzerland)
- de-LI – German (Liechtenstein)

- English
  - en – English
  - en-AU – English (Australia)
  - en-CA – English (Canada)
  - en-IN – English (India)
  - en-NZ – English (New Zealand)
  - en-ZA – English (Southern Africa)
  - en-GB – English (United Kingdom)
  - en-US – English (United States)
- Spanish
  - es – Spanish
  - es-AR – Spanish (Argentina)
  - es-CL – Spanish (Chile)
  - es-CO – Spanish (Colombia)
  - es-CR – Spanish (Costa Rica)
  - es-HN – Spanish (Honduras)
  - es-419 – Spanish (Latin America)
  - es-MX – Spanish (Mexico)
  - es-PE – Spanish (Peru)
  - es-ES – Spanish (Spain)
  - es-US – Spanish (United States)
  - es-UY – Spanish (Uruguay)
  - es-VE – Spanish (Venezuela)
- French
  - fr – French

  - fr-CA – French (Canada)
  - fr-FR – French (France)
  - fr-CH – French (Switzerland)
- Indonesian
  - id – Indonesian
  - id-ID – Indonesian (Indonesia)
- Italian
  - it – Italian
  - it-IT – Italian (Italy)
  - it-CH – Italian (Switzerland)
- Japanese
  - ja – Japanese
  - ja-JP – Japanese (Japan)
- Korean
  - ko – Korean
  - ko-KR – Korean (Korea)
- Portuguese
  - pt – Portuguese
  - pt-BR – Portuguese (Brazil)
  - pt-PT – Portuguese (Portugal)
- Chinese
  - zh – Chinese
  - zh-CN – Chinese (China)
  - zh-HK – Chinese (Hong Kong)
  - zh-TW – Chinese (Taiwan)

Selecting languages in user browser settings

To set a user's local browser settings, follow the appropriate steps.

- In Chrome, choose **Settings**, choose **Languages**, and then order the languages based on preference.

- In Firefox, choose **Settings**, **General**, **Language**, and select the language from the drop-down menu.
- In Edge, choose **Settings**, choose **Languages**, and then order the languages based on preference.

Managing IP access controls in Amazon WorkSpaces Secure Browser

WorkSpaces Secure Browser allows you to control which IP addresses your web portal can be accessed from. By using IP access settings, you can define and manage groups of trusted IP addresses, and only allow users to access their portal when they're connected to a trusted network.

By default, WorkSpaces Secure Browser allows users to access their web portal from anywhere. An IP access control group acts as a virtual firewall that filters which IP address a user can use to connect to the web portal. When associated with your web portal, IP access settings will detect the user IP before authentication to determine whether they are eligible to connect. Once connected, WorkSpaces Secure Browser continuously monitors a user's IP address to ensure they remain connected from a trusted network. If a user's IP changes, WorkSpaces Secure Browser will detect and terminate the session.

To specify the CIDR address ranges, add rules to your IP access control group, and then associate the group with your web portal. You can associate each IP access setting with one or more web portals. To specify the public IP addresses and ranges of IP addresses for your trusted networks, add rules to your IP access control groups. If your users access their web portal through a NAT gateway or VPN, you must create rules that allow traffic from the public IP addresses for the NAT gateway or VPN.

> **Note**
>
> Customers are responsible for understanding the potential legal issues that arise with their use of WorkSpaces Secure Browser, and must ensure that their use of WorkSpaces Secure Browser complies with all applicable laws and regulations. This includes laws that regulate an employer's ability to monitor an employee's use of WorkSpaces Secure Browser, including activities performed within the application.

Topics

- Associating an IP access setting with a web portal in Amazon WorkSpaces Secure Browser
- Editing an IP access control group in Amazon WorkSpaces Secure Browser
- Deleting an IP access control group in Amazon WorkSpaces Secure Browser

Creating an IP access control group in Amazon WorkSpaces Secure Browser

To create an IP access control group, follow these steps.

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. In the navigation pane, choose **IP access controls**.
3. Choose **Create IP access control group**.
4. In the **Create IP access control group** dialog box, enter a name (required) and description (optional) for the group.
5. Enter the IP address or CIDR IP range that will be associated to **Source**, and a **Description** (optional).
6. Under **Tags**, choose whether to tag a key value pair for each IP access control group.
7. When you are done adding rules and tags, choose **Save**.

Associating an IP access setting with a web portal in Amazon WorkSpaces Secure Browser

To associate an IP access control group with an existing web portal, follow these steps.

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. In the navigation pane, choose **Web portals**.
3. Select the web portal, and choose **Edit**.
4. Under **IP access control group,** and select the IP access control groups for the web portal.
5. Choose **Save**.

To associate an IP access control group when creating a new web portal, follow these steps.

1. Complete steps 1 through 4 in the section called “Portal settings” to access **IP Access Control (optional)**.
2. Choose **Create IP access controls**.
3. In the **Create IP Group** dialog box, enter a name (required) and description (optional) for the group.
4. Enter the IP address or CIDR IP range that will be associated to **Source**, and a **Description** (optional).
5. Under **Tags**, choose whether to tag a key value pair for each IP access control group.
6. When you are done adding rules and tags, choose **Create IP access control**.
7. Your IP access control group will be associated to this web portal when launched.

Editing an IP access control group in Amazon WorkSpaces Secure Browser

You can delete a rule from an IP access setting at any time. If you remove a rule that was used to allow a connection to a web portal, any users with a current session will be disconnected from the web portal.

To edit an IP access control group, follow these steps.

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. In the navigation pane, choose **IP access controls**.
3. Select the group and choose **Edit**.
4. Edit the existing rules **Source** and **Description** (optional), or add additional rules.
5. Under **Tags**, choose whether to tag a key value pair for each IP access control group.
6. When you are done adding rules and tags, choose **Save**.
7. If you updated an existing IP access setting, wait up to 15 minutes for the new or edited rule to take effect.

Deleting an IP access control group in Amazon WorkSpaces Secure Browser

You can delete a rule from an IP access control group at any time. If you remove a rule that was used to allow a connection to a web portal, any users with a current session will be disconnected from the web portal.

To delete an IP access control group, follow these steps.

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. In the navigation pane, choose **IP access control group**.
3. Select the group and choose **Delete**.

Managing the single sign-on extension in Amazon WorkSpaces Secure Browser

You can enable an extension for your end users to have a better portal sign-on experience. For example, if you use Okta as your portal's SAML 2.0 identity provider (IdP), and you also use it as the IdP for the websites you want users to visit during a session, you can pass the Okta sign-in cookie to the session with the extension. Afterwards, when users visit a website that requires the Okta domain cookie, they can access the website without having to sign in during the session.

The extension is supported in Chrome and Firefox browsers. The extension enables cookie synchronization for the allowed domains from the users sign-in to the session. The extension does not require the user to sign in, and it works behind the scenes to enable cookie synchronization without requiring the user to take any actions after installation. No data is stored by the extension.

By default, extensions are not enabled in Chrome in Incognito windows or Firefox Private Browsing windows. Users can enable them manually. For more information about Chrome, see Extensions in Incognito mode. For more information about Firefox, see Extensions in Private Browsing.

Users are prompted to install the extension when they sign into a portal. For details about the user experience with the extension, see the section called "Single sign-on extension".

Topics

- Identifying domains for the single sign-on extension in Amazon WorkSpaces Secure Browser

- Adding the single sign-on extension to a new web portal in Amazon WorkSpaces Secure Browser
- Adding the single sign-on extension to an existing web portal in Amazon WorkSpaces Secure Browser
- Editing or removing the single sign-on extension in Amazon WorkSpaces Secure Browser

Identifying domains for the single sign-on extension in Amazon WorkSpaces Secure Browser

First, determine which domains you need for your SAML IdP and websites. You can add up to 10 domains.

You are responsible for testing and identifying the appropriate domain for the cookies to be synchronized. Changes might be required at the IdP or website authentication level to ensure single sign-on works as expected.

To see which domains to use with most common IdP, refer to the following table:

IdP and domains

IdP	Domain
Okta	okta.com
Entra ID	microsoftonline.com
AWS Identity Center	awsapps.com
One Login	onelogin.com
Duo	duosecurity.com

Adding the single sign-on extension to a new web portal in Amazon WorkSpaces Secure Browser

To allow the extension when creating a new web portal, follow these steps.

1. Follow the steps in the section called “Web portal creation” until you get to the section called “User settings”.

2. For step 1 of the section called “User settings”, under **User permissions**, choose **Allowed** to enable the extension for your web portal.
3. Enter the domain for cookie synchronization, and choose **Add new domain**.
4. Complete the steps in the section called “User settings” and the remaining sections in the section called “Web portal creation” to create your web portal.

Adding the single sign-on extension to an existing web portal in Amazon WorkSpaces Secure Browser

To add the extension to an existing web portal, follow these steps.

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home.
2. Select the web portal to edit.
3. Choose **User settings**, **Users permissions**, and **Allowed** to enable the extension for your web portal.
4. Enter the domain for cookie synchronization, choose **Add new domain**.
5. Save your portal changes. The portals will prompt users to install the extension within 15 minutes.

Editing or removing the single sign-on extension in Amazon WorkSpaces Secure Browser

To edit domains or remove the extension, follow these steps.

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home.
2. Select the web portal to edit.
3. Choose **User settings**, **Users permissions**, and **Not allowed** to remove the extension for your web portal.
4. Remove or edit individual domains.
5. Once removed, sessions will no longer synchronize cookies, even if the user has the WorkSpaces Secure Browser extension installed in their browser.

Setting up URL filtering in Amazon WorkSpaces Secure Browser

You can use Chrome Policy to filter which URLs users can access from their remote browser. Chrome Policy provides two mechanisms to filter URLs: URLAllowlist and URLBlocklist. You can use the WorkSpaces Secure Browser console interface to configure URL filtering as a portal setting, or you can add it as part of your custom JSON statement (either in the inline editor, or as a JSON file upload).

Topics

Setting up URL filtering using the console in Amazon WorkSpaces Secure Browser

To set up URL filtering using the console, follow these steps.

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

   - **Allow access to all URLs**: By default, a web portal allows access to all URLs. You can add specific websites to the **BlockURL** list to prevent users from visiting those sites during a session. For example, adding **www.anycorp.com** to the **BlockURL** list will prevent user from navigating to www.anycorp.com during their session.
   - **Block access to all URLs**: By default, the web portal blocks access to all URLS. You can add specific websites to the URL allowlist to curate a list of websites users can visit, and block traffic to any other websites. Consider adding each URL as a bookmark to enable 1-click access for users during their session.
   - **Advanced configuration**: Choose this option to create **allowURL** and **blockURL** lists in parallel. The **URL allowlist** has priority over **URL blocklist**. This option enables URL filtering by path. For example, you can add **www.anycorp.com** to the blocklist, and then add

www.anycorp.com/hr to the allow list. This allows users to visit www.anycorp.com/hr, but they won't be able to access other URL paths, such as www.anycorp.com/finance.

For more guidance about using block and allow URLs, see Allow or block access to websites. Add URLs to these lists following Chrome's blocklist filter format for the best results. For more information, see URL blocklist filter format.

Setting up URL filtering using the JSON editor or file upload for Amazon WorkSpaces Secure Browser

To set up URL filtering using the JSON editor or file upload, follow these steps.

1. From the **Policy settings** module, choose **JSON Editor** and bypass the console UI module for either **Editor** or **File Upload** view.
   - **Editor** allows customers to create custom policy statements inline in the console. Editor highlights errors in the JSON statement during policy creation.
   - **File upload** allows customers to add a JSON file created outside the console (such as exported from an existing Chrome browser).
2. See Chrome Policy details for URLAllowlist and URLBlocklist to properly format an allow/denyURL list for your web portal. For more information, see URLAllowlist and URLBlocklist.

Deep links in Amazon WorkSpaces Secure Browser

When a user signs into WorkSpaces Secure Browser, they start the session on a home page set by the administrator. You can also allow portals to receive deep links that connect users to a specific website during a session. When a deep link is selected, the portal displays the URL specified in the deep link. The link is displayed alongside the home page(s) configured for session start, or by itself if a session is already in progress. This feature allows administrators to create more dynamic user experiences with WorkSpaces Secure Browser.

Deep links open pages in a WorkSpaces Secure Browser session. If a session is already running, it will open the deep link in a new tab. If a session is not already running, it will open the deep link URL in a new tab, and the portal default home page in a separate tab. If a deep link contains more than one URL, it will display the deep link URL listed first in focus, with each subsequent URL (including the default home page) opened in separate tabs.

Topics

Setting up deep links in Amazon WorkSpaces Secure Browser

To allow permission for deep links, choose **Allowed** when creating user settings. The site you want to deep link to must be URL-encoded. For example, to link a user to “https://www.example.com/?query=true”, update the link to https%3A%2F%2Fwww.example.com%2F%3Fquery%3Dtrue.

A deeplink can contain up to 10 URLs, delineated by comma. For example:

https://<uuid>.workspaces-web.com/?deepLinks=https%3A%2F%2Fwww.example.com%2F%3Fquery%3Dtrue,https%3A%2F%2Fwww.example.com%2F%3Fquery%3Dtrue2,https%3A%2F%2Fwww.example.com%2F%3Fquery%3Dtrue3,https%3A%2F%2Fwww.example.com%2F%3Fquery%3Dtrue4.

For more information about allowing deep links, see the section called “User settings”.

Using URL filtering for deep links in Amazon WorkSpaces Secure Browser

Any user you share this portal link with can manipulate the deep link value to visit a website, if that domain is reachable from the portal and not on the URL blocklist. To create a restrictive allowlist or blocklist to prevent users from visiting unintended domains with your portal, use URL filtering.

The allowlist and blocklist for a portal can be edited with URL filtering in your portal’s browser settings. To do this, append the URL to an allow-listed portal URL in the following format, where UUID is the portal id: https://<uuid>.workspaces-web.com/?deepLinks=https%3A%2F%2Fwww.example.com%2F%3Fquery%3Dtrue

For more information, see the section called “Setting up URL filtering” and Allow or block access to websites.

Using the session management dashboard in Amazon WorkSpaces Secure Browser

Use the session management dashboard on your WorkSpaces Secure Browser console to monitor and manage active and complete sessions.

Dashboard access

To access the dashboard, follow these steps.

To access the dashboard

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. Choose **WorkSpaces Secure Browser**, **Web portals**, and choose your web portal.
3. Choose the **Session** tab or choose **View sessions** to open the dashboard in a split panel below.

Dashboard filters

In the sessions panel, you can filter sessions by the following properties or values:

- **Status**
  - **Active** – Indicates a session is currently running. To terminate the session, see below.
  - **Terminated** – Indicates a session is no longer active.
- **Session ID**
- **Username**
- **Session start time**

Terminate sessions

To terminate a session, follow these steps.

To terminate a session

1. On the sessions dashboard, select the session you want to stop.

2. Choose **Terminate**.
3. Disconnected users lose all state from the session. All open tabs, browser history, and files downloaded to the secure browser are recycled.

Session history

The dashboard contains sessions from the last 35 days. You can use the CLI to list sessions, with or without a filter. The session history is delivered as JSON, which administrators can process, manage, and store in a separate repository.

The following are sample CLI commands for managing sessions in the US-West-2 (Oregon) region.

To list all sessions for a web portal, run the following command:

aws workspaces-web list-sessions --portal-arn arn:aws:workspaces-web:us-west-2:<accountId>:portal/<portalId>

To list all sessions for a specific user of a web portal, run the following command:

aws workspaces-web list-sessions --portal-arn arn:aws:workspaces-web:us-west-2:<accountId>:portal/<portalId> --username <username>

Protecting data in transit with FIPS endpoints and Amazon WorkSpaces Secure Browser

By default, when you communicate with the WorkSpaces Secure Browser service as an administrator using the console, the AWS Command Line Interface (AWS CLI), or an AWS SDK, or during a user's session, all data in transit is encrypted using TLS 1.2.

If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. When you use a FIPS endpoint, all data in transit is encrypted using cryptographic standards that comply with Federal Information Processing Standard (FIPS) 140-3. For information about FIPS endpoints, including a list of WorkSpaces Secure Browser endpoints, see https://aws.amazon.com/compliance/fips.

After a portal is created with FIPS endpoints, all user sessions and administrative changes are automatically made using FIPS 140-3 endpoints. You can use the

AWS_USE_FIPS_ENDPOINT=true environment variable to locate FIPS endpoints and send requests with the SDK. The following is an example.

```
$ export AWS_USE_FIPS_ENDPOINT=true
$ aws workspaces-web list-portal
```

You can also use the –endpoint-url option to send requests directly to FIPS endpoints. The following is an example calling list portals in the US-West-2 (Oregon) Region:

```
$ aws workspaces-web list-portal --endpoint-url https://workspaces-web-fips.us-
west-2.amazonaws.com
```

Managing data protection settings in Amazon WorkSpaces Secure Browser

Data Protection Settings are used to help protect data from being shared during a session. Settings can be created and applied to multiple portals.

Topics

Inline data redaction in Amazon WorkSpaces Secure Browser

By adding inline data redaction to a portal, you can automatically predict and redact certain data from a string of text displayed in web pages. You can create redaction policies by choosing from

built-in patterns (such as social security numbers or credit card numbers), or create their own custom data types using regular expression and keywords. Policies include configurable levels of enforcement and controls for the URLs where redaction should be enforced.

The following components determine when data is redacted:

- **Data Protection Settings** - Data Protection Settings is the name of the resource that includes your data types and enforcement criteria. To use this resource, first create your settings, then associate them to a portal. When users launch a session, your settings are enforced during the session.
- **In-session browser extension** - When you associate redaction settings with your portal, the session browser will launch with a system-enforced browser extension that enforces your settings. Data Protection Settings enforce redaction through pattern matching (Regular Expressions) and keyword searching following your confidence level and URL enforcement configuration. Content is predicted from text strings and redacted before displayed on the screen. The extension also sets related browser policies that govern users' ability to bypass redaction (such as disabled private browsing, access to developer tools, and network inspection).

The following Chrome browser policy changes are enforced by the in-session browser extension. For more information, see Chrome Enterprise policy list.

- Enforce browser policy to prevent users from viewing the session without redaction:
  - IncognitoModeAvailability = 1
  - DeveloperToolsAvailability = 2
  - BrowserAddPersonEnabled = false
  - BrowserGuestModeEnabled = false
- The extension also prevents users from downloading HTML files from URLs that are enforcing data protection settings by canceling the download event.

In general, you should use redaction with private, structured websites (such as your customer management tools, ticketing systems, or wikis), and not for unstructured public browsing (such as Facebook or Google). You can choose from built-in data types (see below for the full list), or define custom data types using your own regular expression values and keywords. Administrators are responsible for testing and validating that each data type, confidence level, and URL enforcement are working as expected. AWS cannot guarantee compatibility with custom websites or applications provided by third parties.

WorkSpaces Secure Browser does not currently support redaction of supported or custom data types in non-text formats, including text in the following formats:

- Images, such as JPEG, PNG, or GIF
- Web pages that enable users to use dynamic word processing or editing, such as Google Docs or Sheets
- Audio or video streams accessed in the browser, such as a YouTube videos
- PDFs viewed by the Chrome browser

Do not use redaction for content in an unsupported format. Administrators are responsible for validating site and content compatibility prior to granting users access to content they intend to be redacted.

Default redaction configuration in Amazon WorkSpaces Secure Browser

The default redaction configuration will automatically apply a confidence level and URL enforcement for all built-in data types in the data protection settings. You have the option to override the default configuration when adding a built-in data type.

Confidence levels allow you to fine-tune the redaction logic for built-in data types using a combination of format, keywords, and unformatted text. Choose the level of strictness for how redaction is applied, including High, Medium, or Low. The default value will apply to all data types, unless an override is applied at the data type level. In general, start with a default configuration of Medium, and refine by validating that the redaction is enforced as expected on your sites.

Confidence level	Description	Example
High	Requires a formatted text pattern match in order for content to be redacted.	SSN of 123-45-6798 would be redacted, while 123456789 would not.
Medium	Redaction considers both formatted and unformatt ed text, and adds keyword associate to the logic.	SSN of 123-45-6798 would be redacted. 123456789 would be redacted if detected nearby a keyword (such as "social security number").

Confidence level	Description	Example
Low	Redaction enforced for both formatted pattern + unformatted pattern without keyword.	SSN in either format - 123-45-6798 and 123456789 - are redacted without requiring keyword.

You must set the default redaction configuration for all data types. You can choose from the following options:

- All URLs
- Specific URLs
- Advanced configuration

The default value will apply to all data types, unless an override is applied at the data type level. URL enforcement uses similar logic to Chrome policy for managing allow and blocklists. For guidance using block and allow URLs, see Allow or block access to websites. For the best results, add URLs to these lists following Chrome's blocklist filter format. For more information, see URL blocklist filter format.

Base inline redaction in Amazon WorkSpaces Secure Browser

Inline data redaction has support for built-in patterns (such as social security numbers and credit card numbers), which you can find listed under **Base inline redaction**. Choose the data type(s) from the drop-down menu, and specify the replacement value for each data type. All data types follow the default configuration enforcement pattern above, but you can choose to override the confidence level, and fine-tune the domain enforcement pattern for each data type.

To enter an alternative value from the default configuration, choose **Confidence level override**. For example, with the default configuration set to Medium, you might notice during testing that one of your data types is not being redacted reliably. You can set the override to Low to increase the chance of redaction, without adjusting the logic used for your other data types.

To fine-tune the way redaction is applied across URLs without changing the default configuration, apply **URL enforcement overrides**. For example, you can set use URL overrides to enforce email address redaction in your customer relationship management system, without breaking user access to email addresses in the company directory website or web based email.

The following is a list of data types and their corresponding built-in pattern IDs:

builtInPatternId	Data type
awsAccessKey:	AWS Access Key
awsSecretKey:	AWS Secret Key
cardNumbers:	Credit Card Numbers
crypto:	Cryptocurrency Addresses
cusipNum:	CUSIP Number
date:	Date
deaNum:	US DEA Numbers
dob:	Date of Birth
driversLicense:	US Driver's Licenses
emailAddress:	Email Address
ein:	US Employer Identification Number
expDate:	Credit Card Expiration Date
healthInsuranceNum:	Medicare Health Insurance Claim Number
hipaaCode:	HIPAA ICD-10 Code
indivTaxId:	US Individual Tax Id
ipAddr:	IP Address
isin:	International Securities Identification Numbers
jwt:	JSON Web Token
locationCoord:	Location Coordinates

builtInPatternId	Data type
macAddr:	MAC Address
medicareBeneficiaryId:	Medicare Beneficiary Number
npi:	National Provider Identification Number
ndc:	National Drug Codes (NDC)
passportNum:	US Passport Number
phoneNum:	Phone Number
routingNumber:	ABA Routing Number
ssn:	US Social Security Number
swiftCode:	SWIFT Code
time:	Time
vin:	US Vehicle Identification Number

Custom inline redaction in Amazon WorkSpaces Secure Browser

Customers can define their own patterns using regular expression, such as custom internal application IDs. To create your custom inline redaction pattern, follow these steps:

1. Go to your data protection setting.
2. Choose **Custom inline redaction** and **add**.
3. Enter a name for the custom data type.
4. Enter your regular expression value.
   - Regular expression values must match the JavaScript regular expression literal syntax. For more information, see Regular expressions. An example regular expression is `/ex[am]+ple/i`.
   - Make sure to test your custom patterns on the websites that you plan on supporting. If custom patterns are written with errors, they can introduce unintended performance issues.

5. Specify the replacement value.
6. Choose **More options** for more optional customizations, including the following:
   - Add **keywords** to fine tune the redaction logic. Keywords can increase the accuracy of enforcement. Add keywords in the Javascript regular expression literal syntax. For more information, see Regular expressions.

     For example, if you are creating a custom redaction pattern for client IDs used in an internal system, you can add `/client name/i` to the keyword field to inform the scanning and detection logic.
   - Apply **URL enforcement overrides** to fine-tune the way redaction is applied across URLs, without changing the default configuration.

     For example, you can set use URL overrides to enforce email address redaction in your customer relationship management system, without breaking user access to email addresses in the company directory website or web-based email.
   - Enter a **description** (optional) for the data type.

Create data protection settings in Amazon WorkSpaces Secure Browser

You can create data protection settings in WorkSpaces Secure Browser.

To create data protection settings

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. In the left-hand navigation pane, choose **Data Protection Settings**.
3. Choose **Create Data Protection Settings**.
4. Enter a display name (required) and description (optional) for the setting.
5. Select the default settings for inline redaction. You can set the following:
   - The level of strictness of all data types
   - The domains on which redaction should be enforced
6. Choose your base inline redaction data types from the supported types, or create a custom data type. You can set overrides for each data type, including the level of strictness and domain exceptions.
7. Add any **Tags** (optional) for reporting.

8. When you are done, choose **Save**.

Associate data protection settings in Amazon WorkSpaces Secure Browser

You can associate data protection settings in WorkSpaces Secure Browser.

To associate a data protection setting with an existing portal

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. In the left hand navigation pane, choose **Web portals**.
3. Select the web portal, and choose **Edit**.
4. Under **Data protection settings**, select the setting for your portal.
5. Choose **Save**.

To associate a data protection setting when creating a new portal, follow these steps.

To associate a data protection setting when creating a new portal

1. Follow the instructions in the section called "Web portal creation" to create a portal, until you get to **data protection setting**.
2. Choose your **data protection setting** from the drop-down menu.
3. Complete the steps in the section called "Web portal creation" to finish creating your portal.

To create a data protection setting when creating a new portal, follow these steps.

To create a data protection setting when creating a new portal

1. Follow the instructions in the section called "Web portal creation" to create a portal, until you get to **data protection setting**.
2. Choose **data protection settings** from the drop-down menu.
3. Enter a display name (required) and description (optional) for the setting.
4. Select the default settings for inline redaction. You can set the following:

   - The level of strictness of all data types

- The domains on which redaction should be enforced

5. Choose your base inline redaction data types from the supported types, or create a custom data type. You can set overrides for each data type, including the level of strictness and domain exceptions.
6. Add any **Tags** (optional) for reporting.
7. When you are done, choose **Save**.
8. Select the refresh button under **data protection settings**, then choose your data protection setting from the drop-down menu.
9. Continue to follow the create portal instructions to finish creating your portal.

Edit data protection settings in Amazon WorkSpaces Secure Browser

You can edit data protection settings in WorkSpaces Secure Browser.

To edit data protection settings

1. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.
2. Choose **data protections settings** and the data protection setting you want to edit from the list view.
3. You can update the name, description, default settings, data types (supported or custom), and apply confidence level or domain overrides.
4. Choose **Save**.

Delete data protection settings in Amazon WorkSpaces Secure Browser

You can delete data protection settings in WorkSpaces Secure Browser.

To delete data protection settings

1. If you have a portal associated with a data protection setting, you must first remove the association before deleting the data protection setting.
2. Open the WorkSpaces Secure Browser console at https://console.aws.amazon.com/workspaces-web/home?region=us-east-1#/.

3. Choose **data protections settings** and the data protection setting you want to delete from the list view.
4. Choose **Delete**.

Managing toolbar controls in Amazon WorkSpaces Secure Browser

With **Toolbar controls**, you can configure the toolbar presentation for end user sessions, including the following options:

- **Features**
  - **Clipboard**: When enabled, allows copy/paste with granular controls (copy only, paste only, or both). When disabled, hides icon and prevents usage from the toolbar.
  - **File transfer**: When enabled, allows file operations with granular controls (upload only, download only, or both). When disabled, hides icon and prevents transfers.
  - **Microphone**: When enabled, allows microphone usage. When disabled, hides icon.
  - **Webcam**: When enabled, allows camera usage. When disabled, hides icon.
  - **Dual monitor**: When enabled, allows dual monitor usage. When disabled, hides icon.
  - **Full screen**: When enabled, allows full screen mode. When disabled, hides icon.
  - **Windows**: When enabled, allows moving between windows. When disabled, hides icon.
- **Settings**
  - **Toolbar theme**: Controls light or dark mode display. Configuration removes end user theme control.
  - **Toolbar state**: Sets docked or detached state of the toolbar. Configuration removes end user control over the toolbar state.
  - **Max resolution**: Defines the highest allowed display resolution. Users can only select resolutions up to this defined limit.

Security in Amazon WorkSpaces Secure Browser

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 Secure Browser, 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 Secure Browser. It shows you how to configure Amazon WorkSpaces Secure Browser 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 Secure Browser resources.

Contents

Data protection in Amazon WorkSpaces Secure Browser

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

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

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

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

Topics

Data encryption in Amazon WorkSpaces Secure Browser

Amazon WorkSpaces Secure Browser collects portal customization data, such as browser settings, user settings, network settings, identity provider information, trust store data, and trust store certificate data. WorkSpaces Secure Browser also collects browser policy data, user preferences (for browser settings), and session logs. Collected data is stored in Amazon DynamoDB and Amazon S3. WorkSpaces Secure Browser 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 Secure Browser actions. Use IAM templates to create a Full Access role or Read Only role. For more information, see AWS managed policies for WorkSpaces Secure Browser.
- Protect data end to end by providing a customer managed key, so WorkSpaces Secure Browser 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 Secure Browser 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 Secure Browser 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 Secure Browser 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 for Amazon WorkSpaces Secure Browser

Encryption at rest is configured by default and all customer data (for example, browser policy statements, usernames, logging, or IP addresses) used in WorkSpaces Secure Browser is encrypted

using AWS KMS. By default, WorkSpaces Secure Browser enables encryption with an AWS-owned key. You can also use a Customer Managed Key (CMK) by specifying your CMK on resource creation. This is currently only supported via the CLI.

If you choose to pass a CMK, the key provided must be a symmetric encryption AWS KMS key and you, as the administrator, must have the following permissions:

```
kms:DescribeKey

kms:GenerateDataKey

kms:GenerateDataKeyWithoutPlaintext

kms:Decrypt

kms:ReEncryptTo
kms:ReEncryptFrom
```

If you use a CMK, you will need to allowlist the WorkSpaces Secure Browser external service principal to access to the key.
For more information, see Example of Scoped CMK Key Policy with aws:SourceAccount

Whenever possible, WorkSpaces Secure Browser will use Forward Access Sessions (FAS) credentials to access your key. For more information about FAS, see Forward access sessions.
There are cases where WorkSpaces Secure Browser may need to access your key asynchronously. By allowlisting the WorkSpaces Secure Browser external service principal in your key policy, WorkSpaces Secure Browser will be able to perform the allowlisted set of cryptographic operations with your key.

After a resource is created, the key can no longer be removed or changed. If you used a CMK, you, as the administrator accessing the resource, must have the following permissions:

```
kms:GenerateDataKey
kms:GenerateDataKeyWithoutPlaintext
kms:Decrypt

kms:ReEncryptTo
kms:ReEncryptFrom
```

If you see an **Access Denied** error when using the console, it is likely that the user accessing the console doesn't have the required permissions to use the CMK on the key that is being used.

Key policy and scoping examples for WorkSpaces Secure Browser

CMKs require the following key policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
  ...,
    {
      "Sid": "Allow WorkSpaces Secure Browser to encrypt/decrypt",
      "Effect": "Allow",
      "Principal": {
        "Service": "workspaces-web.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
       ],
      "Resource": "*",
      }
    ]
}
```

The following permissions are required by WorkSpaces Secure Browser:

- `kms:DescribeKey` — Validates that the provided AWS KMS key is configured correctly.
- `kms:GenerateDataKeyWithoutPlaintext` and `kms:GenerateDataKey` — Request for the AWS KMS key to create data keys used to encrypt objects.
- `kms:Decrypt` — Requests the AWS KMS key to decrypt the encrypted data keys. These data keys are used to encrypt your data.
- `kms:ReEncryptTo` and `kms:ReEncryptFrom` — Request for the AWS KMS key to permit re-encryption from or to a KMS key.

Scoping WorkSpaces Secure Browser permissions on your AWS KMS key

When the principal in a key policy statement is an AWS service principal, we strongly recommend that you use the aws:SourceArn or aws:SourceAccount global condition keys, in addition to the Encryption Context.

The Encryption Context used for a resource will always contain an entry in the format `aws:workspaces-web:RESOURCE_TYPE:id` and the corresponding resource ID.

The source ARN and source account values are included in the authorization context only when a request comes to AWS KMS from another AWS service. This combination of conditions implements least privileged permissions and avoids a potential confused deputy scenario. For more information, see Permissions for AWS services in key policies.

```
   "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "AccountId",
          "kms:EncryptionContext:aws:workspaces-web:resourceType:id": "resourceId"
        },
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:workspaces-web:Region:AccountId:resourceType/resourceId"
          ]
        },
      }
```

> **Note**
>
> Before resource creation, the key policy should only use the `aws:SourceAccount` Condition, as the full resource arn will not exist yet. Following resource creation, the key policy can be updated to include the `aws:SourceArn` and `kms:EncryptionContext` Conditions.

Example of Scoped CMK key policy with `aws:SourceAccount`

```
{
  "Version": "2012-10-17",
  "Statement": [
  ...,
```

```
    {
      "Sid": "Allow WorkSpaces Secure Browser to encrypt/decrypt",
      "Effect": "Allow",
      "Principal": {
        "Service": "workspaces-web.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
       ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "<AccountId>"
        }
      }
    }
  ]
}
```

Example of scoped CMK key policy with `aws:SourceArn` and resource wildcard

```
{
  "Version": "2012-10-17",
  "Statement": [
  ...,
    {
      "Sid": "Allow WorkSpaces Secure Browser to encrypt/decrypt",
      "Effect": "Allow",
      "Principal": {
        "Service": "workspaces-web.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
```

```
       ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:workspaces-web:<Region>:<AccountId>:*/*"
        }
      }
    }
  ]
}
```

Example of scoped CMK key policy with `aws:SourceArn`

```
{
  "Version": "2012-10-17",
  "Statement": [
  ...,
    {
      "Sid": "Allow WorkSpaces Secure Browser to encrypt/decrypt",
      "Effect": "Allow",
      "Principal": {
        "Service": "workspaces-web.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
       ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:workspaces-web:<Region>:<AccountId>:portal/*",
            "arn:aws:workspaces-web:<Region>:<AccountId>:browserSettings/*",
            "arn:aws:workspaces-web:<Region>:<AccountId>:userSettings/*",
            "arn:aws:workspaces-web:<Region>:<AccountId>:ipAccessSettings/*"
          ]
        }
    }
  ]
```

```
}
```

> **Note**
>
> After you create the resource, you can update the wildcard in `SourceArn` for it. If you use WorkSpaces Secure Browser to create a new resource that requires CMK access, ensure you update its key policy accordingly.

Example of scoped CMK key policy with `aws:SourceArn` and resource-specific `EncryptionContext`

```
{
  "Version": "2012-10-17",
  "Statement": [
  ...,
    {
      "Sid": "Allow WorkSpaces Secure Browser to encrypt/decrypt portal",
      "Effect": "Allow",
      "Principal": {
        "Service": "workspaces-web.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
       ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "<AccountId>",
            "kms:EncryptionContext:aws:workspaces-web:portal:id": "<portalId>>"
        }
      }
    },
    {
      "Sid": "Allow WorkSpaces Secure Browser to encrypt/decrypt userSettings",
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "workspaces-web.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
       ],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
            "aws:SourceAccount": "<AccountId>",
            "kms:EncryptionContext:aws:workspaces-web:userSetttings:id":
"<userSetttingsId>"
         }
      }
    },
    {
      "Sid": "Allow WorkSpaces Secure Browser to encrypt/decrypt browserSettings",
      "Effect": "Allow",
      "Principal": {
        "Service": "workspaces-web.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
       ],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
            "aws:SourceAccount": "<AccountId>",
            "kms:EncryptionContext:aws:workspaces-web:browserSettings:id":
"<browserSettingsId>"
         }
      }
    },
    {
      "Sid": "Allow WorkSpaces Secure Browser to encrypt/decrypt ipAccessSettings",
```

```
      "Effect": "Allow",
      "Principal": {
        "Service": "workspaces-web.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
       ],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
            "aws:SourceAccount": "<AccountId>",
            "kms:EncryptionContext:aws:workspaces-web:ipAccessSettings:id":
 "<ipAccessSettingsId>"
         }
      }
    },
  ]
}
```

Note

Ensure you create separate statements when including a resource specific `EncryptionContext` on the same key policy. For more information, see the Using multiple encryption context pairs section under kms:EncryptionContext:*context-key*.

Encryption in transit for Amazon WorkSpaces Secure Browser

WorkSpaces Secure Browser 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 Secure Browser.

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

Key management for Amazon WorkSpaces Secure Browser

You can supply your own Customer Managed AWS KMS Key to encrypt your customer information. If you don't supply one, WorkSpaces Secure Browser will use an AWS Owned Key. You can set your key using the AWS SDK.

Inter-network traffic privacy in Amazon WorkSpaces Secure Browser

To secure connections between WorkSpaces Secure Browser and on-premise applications, you use WorkSpaces Secure Browser 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 Secure Browser.

To secure connections between accounts, WorkSpaces Secure Browser 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 Amazon WorkSpaces Secure Browser.

User access logging in Amazon WorkSpaces Secure Browser

Administrators are able to record WorkSpaces Secure Browser session events, including start, stop, and URL visits. These logs are encrypted and securely delivered to customers through an Amazon Kinesis Data Stream. Browsing information from user access logging is not stored by AWS, or available from sessions without logging configured. URL visits in incognito mode, or deleted URLs from browser history, are not recorded in user access logging.

Identity and Access Management for Amazon WorkSpaces Secure Browser

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

Topics

- Managing access using policies
- How Amazon WorkSpaces Secure Browser works with IAM
- Identity-based policy examples for Amazon WorkSpaces Secure Browser
- AWS managed policies for WorkSpaces Secure Browser
- Troubleshooting Amazon WorkSpaces Secure Browser identity and access
- Using service-linked roles for Amazon WorkSpaces Secure Browser

Audience

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

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

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

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

Authenticating with identities

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

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

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

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

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

AWS account root user

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

Federated identity

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

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using

credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

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

IAM users and groups

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

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

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

IAM roles

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

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

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity

is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Create a role for a third-party identity provider (federation) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.

- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API

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

Managing access using policies

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

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

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

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

Identity-based policies

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

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed

policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

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

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

Access control lists (ACLs)

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

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

Other policy types

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

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.

- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – RCPs are JSON policies that you can use to set the maximum available permissions for resources in your accounts without updating the IAM policies attached to each resource that you own. The RCP limits permissions for resources in member accounts and can impact the effective permissions for identities, including the AWS account root user, regardless of whether they belong to your organization. For more information about Organizations and RCPs, including a list of AWS services that support RCPs, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

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

How Amazon WorkSpaces Secure Browser works with IAM

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

IAM features you can use with Amazon WorkSpaces Secure Browser

IAM feature	WorkSpaces Secure Browser support
Identity-based policies	Yes

IAM feature	WorkSpaces Secure Browser support
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys	Yes
ACLs	No
ABAC (tags in policies)	Partial
Temporary credentials	Yes
Principal permissions	Yes
Service roles	No
Service-linked roles	Yes

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

Topics

- Identity-based policies for WorkSpaces Secure Browser
- Resource-based policies within WorkSpaces Secure Browser
- Policy actions for WorkSpaces Secure Browser
- Policy resources for WorkSpaces Secure Browser
- Policy condition keys for WorkSpaces Secure Browser
- Access control lists (ACLs) in WorkSpaces Secure Browser
- Attribute-based access control (ABAC) with WorkSpaces Secure Browser
- Using Temporary credentials with WorkSpaces Secure Browser
- Cross-service principal permissions for WorkSpaces Secure Browser
- Service roles for WorkSpaces Secure Browser
- Service-linked roles for WorkSpaces Secure Browser

Identity-based policies for WorkSpaces Secure Browser

Supports identity-based policies: Yes

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

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

Identity-based policy examples for WorkSpaces Secure Browser

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

Resource-based policies within WorkSpaces Secure Browser

Supports resource-based policies: No

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

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access

to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for WorkSpaces Secure Browser

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

Policy actions in WorkSpaces Secure Browser 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 Secure Browser identity-based policies, see Identity-based policy examples for Amazon WorkSpaces Secure Browser.

Policy resources for WorkSpaces Secure Browser

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

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

Policy condition keys for WorkSpaces Secure Browser

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

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

Access control lists (ACLs) in WorkSpaces Secure Browser

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 Secure Browser

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.

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

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

Using Temporary credentials with WorkSpaces Secure Browser

Supports temporary credentials: Yes

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

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

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

Cross-service principal permissions for WorkSpaces Secure Browser

Supports forward access sessions (FAS): Yes

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

Service roles for WorkSpaces Secure Browser

Supports service roles: No

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

> **Warning**
>
> Changing the permissions for a service role might break WorkSpaces Secure Browser's functionality. Edit service roles only when WorkSpaces Secure Browser provides guidance to do so.

Service-linked roles for WorkSpaces Secure Browser

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 AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

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

Identity-based policy examples for Amazon WorkSpaces Secure Browser

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

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

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

Topics

Identity-based policy best practices for Amazon WorkSpaces Secure Browser

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

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API

operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

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

Using the Amazon WorkSpaces Secure Browser console

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

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

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

Allowing users to view their own permissions for Amazon WorkSpaces Secure Browser

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 Secure Browser

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*.

Topics

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 Secure Browser to perform actions on your behalf. For more information, see the section called “Using service-linked roles”.

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

Permissions details

This policy includes the following permissions:

- `workspaces-web` – Allows access to AWS services and resources used or managed by WorkSpaces Secure Browser.
- `ec2` – Allows principals to describe VPCs, subnets, and availability zones; create, tag, 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.

- `Kinesis` - Allows principals to describe a summary of Kinesis data streams and put records into Kinesis data streams for user access logging. For more information, see the section called "Setting up user activity logging".

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeNetworkInterfaces",
                "ec2:AssociateAddress",
                "ec2:DisassociateAddress",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeVpcEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/WorkSpacesWebManaged": "true"
                }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkInterface"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "WorkSpacesWebManaged"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/WorkSpacesWebManaged": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/WorkSpacesWeb",
                        "AWS/Usage"
                    ]
                }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord",
                "kinesis:PutRecords",
                "kinesis:DescribeStreamSummary"
            ],
            "Resource": "arn:aws:kinesis:*:*:stream/amazon-workspaces-web-*"
        }
    ]
}
```

AWS managed policy: AmazonWorkSpacesSecureBrowserReadOnly

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

This policy grants read-only permissions that allow access to WorkSpaces Secure Browser and its dependencies through the AWS Management Console, SDK, and CLI. This policy does not include the permissions necessary to interact with portals using `IAM_Identity_Center` as the authentication type. To get these permissions, combine this policy with `AWSSSOReadOnly`.

Permissions details

This policy includes the following permissions.

- `workspaces-web` – Provides read-only access to WorkSpaces Secure Browser 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 Secure Browser to show you your VPCs, subnets, and security groups that are available foruse with the service.
- `Kinesis` - Allows principals to list Kinesis data streams. This is used in the AWS Management Console in WorkSpaces Secure Browser to show you Kinesis data streams that are available for use 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:GetUserAccessLoggingSettings",
                "workspaces-web:ListBrowserSettings",
                "workspaces-web:ListIdentityProviders",
                "workspaces-web:ListNetworkSettings",
                "workspaces-web:ListPortals",
                "workspaces-web:ListTagsForResource",
                "workspaces-web:ListTrustStoreCertificates",
                "workspaces-web:ListTrustStores",
                "workspaces-web:ListUserSettings",
                "workspaces-web:ListUserAccessLoggingSettings"
            ],
            "Resource": "arn:aws:workspaces-web:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "kinesis:ListStreams"
            ],
            "Resource": "*"
        }
    ]
}
```

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 Secure Browser and its dependencies through the AWS Management Console, SDK, and CLI. This policy does not include the permissions necessary to interact with portals using IAM_Identity_Center as the authentication type. To get these permissions, combine this policy with AWSSSOReadOnly.

> **Note**
>
> If you are currently using this policy, switch to the new AmazonWorkSpacesSecureBrowserReadOnly policy.

Permissions details

This policy includes the following permissions.

- workspaces-web – Provides read-only access to WorkSpaces Secure Browser 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 Secure Browser to show you your VPCs, subnets, and security groups that are available foruse with the service.
- Kinesis - Allows principals to list Kinesis data streams. This is used in the AWS Management Console in WorkSpaces Secure Browser to show you Kinesis data streams that are available for use 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:GetUserAccessLoggingSettings",
                "workspaces-web:ListBrowserSettings",
                "workspaces-web:ListIdentityProviders",
                "workspaces-web:ListNetworkSettings",
                "workspaces-web:ListPortals",
                "workspaces-web:ListTagsForResource",
                "workspaces-web:ListTrustStoreCertificates",
                "workspaces-web:ListTrustStores",
                "workspaces-web:ListUserSettings",
                "workspaces-web:ListUserAccessLoggingSettings"
            ],
            "Resource": "arn:aws:workspaces-web:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "kinesis:ListStreams"
            ],
            "Resource": "*"
        }
    ]
}
```

WorkSpaces Secure Browser updates to AWS managed policies

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

Change	Description	Date
AmazonWorkSpacesSecureBrowserReadOnly – New policy	WorkSpaces Secure Browser added a new policy to provide read-only access to WorkSpaces Secure Browser and its dependencies through the AWS Management Console, SDK, and CLI.	June 24, 2024
AmazonWorkSpacesWebServiceRolePolicy – Updated policy	WorkSpaces Secure Browser updated the policy to restrict CreateNetworkInterface to tag with aws:RequestTag/WorkSpacesWebManaged: true and act on subnet and security group resources, as well as restrict DeleteNetworkInterface to ENIs tagged with aws:ResourceTag/WorkSpacesWebManaged: true.	December 15, 2022
AmazonWorkSpacesWebReadOnly – Updated policy	WorkSpaces Secure Browser updated the policy to include read permissions for user access logging and list Kinesis data streams. For more information, see the section called “Setting up user activity logging”.	November 2, 2022
AmazonWorkSpacesWebServiceRolePolicy – Updated policy	WorkSpaces Secure Browser updated the policy to describe a summary of Kinesis data streams and put records into Kinesis data streams for user access logging. For	October 17, 2022

Change	Description	Date
	more information, see the section called "Setting up user activity logging".	
AmazonWorkSpacesWebServiceRolePolicy – Updated policy	WorkSpaces Secure Browser updated the policy to create tags during ENI creation.	September 6, 2022
AmazonWorkSpacesWebServiceRolePolicy – Updated policy	WorkSpaces Secure Browser updated the policy to add the AWS/Usage namespace to the PutMetricData API permissions.	April 6, 2022
AmazonWorkSpacesWebReadOnly – New policy	WorkSpaces Secure Browser added a new policy to provide read-only access to WorkSpaces Secure Browser and its dependencies through the AWS Management Console, SDK, and CLI.	November 30, 2021
AmazonWorkSpacesWebServiceRolePolicy – New policy	WorkSpaces Secure Browser added a new policy to allow access to AWS services and resources used or managed by WorkSpaces Secure Browser.	November 30, 2021
WorkSpaces Secure Browser started tracking changes	WorkSpaces Secure Browser started tracking changes for its AWS managed policies.	November 30, 2021

Troubleshooting Amazon WorkSpaces Secure Browser identity and access

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

Topics

I am not authorized to perform an action in WorkSpaces Secure Browser

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

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `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, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `workspaces-web:`*`GetWidget`* action.

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

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to WorkSpaces Secure Browser.

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 Secure Browser. However, the action requires the service to have permissions that are 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's policies must be updated to allow her to perform the `iam:PassRole` action.

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

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

Using service-linked roles for Amazon WorkSpaces Secure Browser

Amazon WorkSpaces Secure Browser 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

Secure Browser. Service-linked roles are predefined by WorkSpaces Secure Browser 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 Secure Browser easier because you don't have to manually add the necessary permissions. WorkSpaces Secure Browser defines the permissions of its service-linked roles, and unless defined otherwise, only WorkSpaces Secure Browser 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 Secure Browser 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.

Topics

- Service-linked role permissions for WorkSpaces Secure Browser
- Creating a service-linked role for WorkSpaces Secure Browser
- Editing a service-linked role for WorkSpaces Secure Browser
- Deleting a service-linked role for WorkSpaces Secure Browser
- Supported regions for WorkSpaces Secure Browser service-linked roles

Service-linked role permissions for WorkSpaces Secure Browser

WorkSpaces Secure Browser uses the service-linked role named `AWSServiceRoleForAmazonWorkSpacesWeb` – WorkSpaces Secure Browser 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 Secure Browser to complete the following actions on the specified resources. For more information, see the section called “AmazonWorkSpacesWebServiceRolePolicy”.

- Action: ec2:DescribeVpcs on all AWS resources
- Action: ec2:DescribeSubnets on all AWS resources
- Action: ec2:DescribeAvailabilityZones on all AWS resources
- Action: ec2:CreateNetworkInterface with aws:RequestTag/WorkSpacesWebManaged: true on subnet and security group resources
- Action: ec2:DescribeNetworkInterfaces on all AWS resources
- Action: ec2:DeleteNetworkInterface on network interfaces with aws:ResourceTag/WorkSpacesWebManaged: true
- 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: ec2:CreateTags on ec2:CreateNetworkInterface Operation with aws:TagKeys: ["WorkSpacesWebManaged"]
- Action: cloudwatch:PutMetricData on all AWS resources
- Action: kinesis:PutRecord on Kinesis data streams with names that start with amazon-workspaces-web-
- Action: kinesis:PutRecords on Kinesis data streams with names that start with amazon-workspaces-web-
- Action: kinesis:DescribeStreamSummary on Kinesis data streams with names that start with amazon-workspaces-web-

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 Secure Browser

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 Secure Browser 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 Secure Browser creates the service-linked role for you again.

You can also use the IAM console to create a service-linked role with the **WorkSpaces Secure Browser** 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 Secure Browser

WorkSpaces Secure Browser 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 Secure Browser

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 Secure Browser 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 Secure Browser 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, trust stores, and user access logging settings) 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 Secure Browser service-linked roles

WorkSpaces Secure Browser 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 Secure Browser

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 Secure Browser with Amazon CloudWatch.

Compliance validation for Amazon WorkSpaces Secure Browser

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

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

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

- Security Compliance & Governance – These solution implementation guides discuss architectural considerations and provide steps for deploying security and compliance features.
- HIPAA Eligible Services Reference – Lists HIPAA eligible services. Not all AWS services are HIPAA eligible.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.

- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in Amazon WorkSpaces Secure Browser

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 Secure Browser:

- 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 Secure Browser

As a managed service, Amazon WorkSpaces Secure Browser is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Amazon WorkSpaces Secure Browser through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.

- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

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

WorkSpaces Secure Browser 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 Secure Browser

WorkSpaces Secure Browser 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 Secure Browser according to specifications and guidelines, and to monitor WorkSpaces Secure Browser usage by your users. All service-related configs and vulnerability analysis are the responsibility of WorkSpaces Secure Browser.

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

Access APIs using an interface VPC endpoint (AWS PrivateLink)

You can directly call Amazon WorkSpaces Secure Browser API endpoint from within a private cloud (VPC), instead of connecting over the internet. You can do this without the use of an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.

You establish this private connection by creating an *interface VPC endpoint* that's powered by AWS PrivateLink. For each subnet that you specify from your VPC, we create an endpoint network

interface in the subnet. An endpoint network interface is a requester-managed network interface that serves as the entry point for Amazon WorkSpaces Secure Browser API traffic.

For more information, see Access AWS services through AWS PrivateLink.

Topics

Considerations for Amazon WorkSpaces Secure Browser

Before you set up an interface VPC endpoint for Amazon WorkSpaces Secure Browser APIs, make sure to review the "Prerequisites" in Access AWS services through AWS PrivateLink. Amazon WorkSpaces Secure Browser supports making calls to all of its API actions through the interface VPC endpoint.

By default, full access to Amazon WorkSpaces Secure Browser is allowed through the endpoint. For more information, see Controlling access to services with VPC endpoints in the *Amazon VPC User Guide*.

Creating an interface VPC endpoint for Amazon WorkSpaces Secure Browser

You can create an interface VPC endpoint for the Amazon WorkSpaces Secure Browser service using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Creating an interface endpoint in the *Amazon VPC User Guide*.

Create an interface VPC endpoint for Amazon WorkSpaces Secure Browser using the following service name:

- com.amazonaws.*region*.workspaces-web

For FIPS-supported regions, create an interface VPC endpoint for Amazon WorkSpaces Secure Browser using the following service name:

- com.amazonaws.*region*.workspaces-web-fips

Creating an endpoint policy for your interface VPC endpoint

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

An endpoint policy specifies the following information:

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

For more information, see Controlling access to services with VPC endpoints in the *Amazon VPC User Guide*.

Example: VPC endpoint policy for Amazon WorkSpaces Secure Browser actions

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

```
{
     "Statement": [
         {
             "Action": "workspaces-web:*",
             "Effect": "Allow",
             "Resource": "*",
             "Principal": "*"
         }
     ]
}
```

Troubleshooting

If your calls to the Amazon WorkSpaces Secure Browser APIs are hanging, there is likely a misconfiguration in your VPC Endpoint Service security group or IAM role setup. To resolve this, try the following:

- While creating your interface VPC endpoint, it might have automatically attached to your AWS account's default security group. Try using a different security group, and make sure the inbound and outbound permissions allow you to transfer your data appropriately.
- Make sure you are using an IAM role that allows you to call Amazon WorkSpaces Secure Browser APIs.

For more information, see What is AWS PrivateLink? in the *Amazon VPC User Guide*.

Security best practices for Amazon WorkSpaces Secure Browser

Amazon WorkSpaces Secure Browser 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 Secure Browser include the following:

- To detect potential security events associated with your use of WorkSpaces Secure Browser, use AWS CloudTrail or Amazon CloudWatch to detect and track access history and process logs. For more information, see Monitoring Amazon WorkSpaces Secure Browser with Amazon CloudWatch and Logging WorkSpaces Secure Browser API calls using AWS CloudTrail.
- To implement detective controls and identify anomalies, use CloudTrail logs and CloudWatch metrics. For more information, see Monitoring Amazon WorkSpaces Secure Browser with Amazon CloudWatch and Logging WorkSpaces Secure Browser API calls using AWS CloudTrail.
- You can set up user access logging to record user events. For more information, see the section called "Setting up user activity logging".

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

- Implement least privilege access and create specific roles to be used for WorkSpaces Secure Browser actions. Use IAM templates to create a Full Access or Read Only role. For more information, see AWS managed policies for WorkSpaces Secure Browser.

- 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 Secure Browser

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

- *Amazon CloudWatch* monitors your AWS resources 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.

Topics

Monitoring Amazon WorkSpaces Secure Browser with Amazon CloudWatch

You can monitor Amazon WorkSpaces Secure Browser 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 Secure Browser

Metric	Description	Dimensions	Statistics	Units
`SessionAttempt`	The number of Amazon WorkSpaces Secure Browser session attempts.	`[PortalId]`	Average, Sum, Maximum, Minimum	Count
`SessionSuccess`	The number of successful Amazon WorkSpaces Secure Browser session starts.	`[PortalId]`	Average, Sum, Maximum, Minimum	Count
`SessionFailure`	The number of failed Amazon WorkSpaces Secure Browser session starts.	`[PortalId]`	Average, Sum, Maximum, Minimum	Count
`SessionIdleDisconnect`	The number of connections that were closed due to user inactivity.	`[PortalId]`	Average	Count
`ActiveSession`	The number of active sessions on a portal.	`[PortalId]`	Average	Count

Metric	Description	Dimensions	Statistics	Units
GlobalCpuPercent	The CPU usage of the Amazon WorkSpaces Secure Browser session instance.	[PortalId] [PortalId, UserName]	Average, Sum, Maximum, Minimum	Percent
GlobalMemoryPercent	The memory (RAM) usage of the Amazon WorkSpaces Secure Browser session instance.	[PortalId] [PortalId, UserName]	Average, Sum, Maximum, Minimum	Percent
DisplayLatency	The average time in milliseconds between frame capture and presentation.	[PortalId] [PortalId, UserName]	Average, Maximum, Minimum	Miliseconds
InputLatency	The input latency between client and server. For example, the latency between the client mouse click and server mouse click.	[PortalId] [PortalId, UserName]	Average, Maximum, Minimum	Miliseconds
SessionLoggerEventDelivered	The number of events each delivered Session Logger file has.	[PortalId]	Average, Sum, Maximum, Minimum	Count

Metric	Description	Dimensions	Statistics	Units
SessionLoggerTargetNotFoundError	The number of log file deliveries that resulted in bucket not found.	[PortalId]	Average, Sum, Maximum, Minimum	Count
SessionLoggerAccessDeniedError	The number of log file deliveries that resulted in permissions denied.	[PortalId]	Average, Sum, Maximum, Minimum	Count

Note

The metric data points are collected by each session once per minute and published to CloudWatch once every 5 minutes. Session Logger metrics are emitted immediately, for each Log File delivery.

Dimensions for Amazon WorkSpaces Secure Browser metrics

Dimension	Description
PortalId	Filters the metric data for Amazon WorkSpaces Secure Browser for a specified portal.
UserName	Filters the metric data for Amazon WorkSpaces Secure Browser for a specified portal and user.

You can use the **SessionLoggerEventDelivered** metric to monitor the aggregate number of events from your portal, or see the number of log files that were delivered by counting the number of data points rather than summing values. We recommend configuring alarms on the **SessionLoggerTargetNotFoundError** and SessionLoggerAccessDeniedError metrics to detect accidental resource or permissions deletion.

Logging WorkSpaces Secure Browser API calls using AWS CloudTrail

WorkSpaces Secure Browser 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 Secure Browser. CloudTrail captures all API calls for Amazon WorkSpaces Secure Browser as events. These include calls from the Amazon WorkSpaces Secure Browser console and code calls to Amazon WorkSpaces Secure Browser 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 Secure Browser. 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 Secure Browser, 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.

Topics

WorkSpaces Secure Browser information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon WorkSpaces Secure Browser, 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 Secure Browser, 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 Secure Browser 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 IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity element.

Understanding WorkSpaces Secure Browser 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"
    }]
}
```


User activity logging in Amazon WorkSpaces Secure Browser

Amazon WorkSpaces Secure Browser enables customers to log session events related to user activities in the Secure browser sessions.

WorkSpaces Secure Browser offers two options for logging user activity and security-related events:

- Session Logger captures a wide range of session events. These logs are delivered to an Amazon S3 bucket in your account, enabling easy integration with your preferred SIEM platform.
- User Access Logging captures the most critical session events. These logs are streamed to an Amazon Kinesis stream for real-time processing and analysis.

For more information about how to set up these options, see the section called “Setting up Session Logger” and the section called “Setting up User Access logging”.

Topics

Session events in Session Logger for Amazon WorkSpaces Secure Browser

Session Logger captures various session-related events for monitoring and auditing purposes.

You can configure Session Logger to collect all session events or a selected subset, depending on the needs of the WorkSpaces Secure Browser portal. For more information about configuration, see the section called “Setting up Session Logger”.

To maintain user privacy, Session Logger does not record sensitive content, such as clipboard data, or the contents of uploaded or downloaded files.

The following fields are included in all events:

- **Time**
- **Username**
- **Portal ID**
- **Portal IP**
- **Client IP**
- **Session ID**

Name	Description	Additional fields included in the event
SessionStart	A secure browser session was launched, but the user has not connected yet.	
SessionConnect	The user is connected to the secure browser session.	
TabOpen	In their secure browser session, the user opened a new tab, or they opened a link in a new tab.	Hostname, path, URL (if the user opens a link in a new tab), none (if the user opens a new tab)
UrlVisit	In their browser session, the user navigated to a URL.	Hostname, path, URL
WebsiteInteract	The user changed a standard HTML element on a website (e.g., clicks a checkbox, radio-	Hostname, path, URL

Name	Description	Additional fields included in the event
	button, or button, or selects an item in the drop-down).	
TabClose	In their browser session, the user closed a tab.	Hostname, path, URL (if the user closes a tab they navigated to), none (if the user closes a new tab)
ContentTransferFromLocalToRemoteClipboard	The user updated the clipboard within the secure browser using content from their local browser (outside the secure environment). This update can occur either by copying content through the in-session toolbar or by transferring data via keyboard shortcuts (Ctrl+C / Ctrl+V).	
ContentCopyFromWebsite	The user updated the clipboard within the secure browser using content from the secure browser (inside the secure environment).	Hostname, path, URL
ContentPasteToWebsite	Clipboard content was pasted into a webpage within the browser. (This event does not capture instances where clipboard content is pasted into the browser's URL bar.)	Hostname, path, URL

Name	Description	Additional fields included in the event
PrintJobSubmit	The user submitted a request job to the browser's virtual printer ("DCV Printer"). The content is saved as PDF on the user's local machine.	Filename, size, extension
FileDownloadFromSecureBrowserToRemoteDisk	A file was saved from the session to the remote instance's local disk.	Hostname, path, URLfilename, size, extension
FileTransferFromRemoteToLocalDisk	A file was downloaded from the remote instance's disk to the user's local device.	Filename, size, extension
FileUploadFromRemoteDiskToSecureBrowser	A file stored on the remote instance's local disk was uploaded to a file-sharing SaaS platform (e.g., Google Drive, Box, or File.io) via the browser session.	
FileTransferFromLocalToRemoteDisk	A file was uploaded from the user device to the secure browser session.	Filename, size, and extension
SessionDisconnection	The user is disconnected from the secure browser session.	

Name	Description	Additional fields included in the event
SessionEnd	The secure browser session has terminated. Terminati on can occur in one of three ways: the administrator ends the session via the User Session Manager in the console, the user manually ends the session using End Session in the toolbar, or the session times out after exceeding a duration set by the administrator.	

Each event follows the OCSF standard and includes a list of attributes that are common to all events:

```
{
    activity_name : String | A human readable name of the event | eg. UrlLoad
    activity_id : Integer | OCSF standard value 99 for 'others'
    category_name : "WorkSpacesSecureBrowser" | The category name where the event
 belongs to.
    category_id : 2 | Numerical identifier for category,
    metadata : link | Required {
        product : link {
            vendor_name : "wsb",
            name : "WorkSpacesSecureBrowser"
        }
        version : String | Version of the schema | eg. 1.0.0
    },
    severity_id : 1 | The severity of the event. All events will have a severity of 1,
 meaning 'Informational',
    type_id : class_uid * 100 + activity_id
    time : The time the event happened (RFC3339 format),
    observables : link [
        {
```

```
            name : "session_detail.portal_id",
            type_id : 10 //Resource UID
            value : //Generated value
        },
        {
            name : "session_detail.session_id",
            type_id : 10 //Resource UID
            value : //Generated value
        },
        {
            name : "session_detail.client_ip",
            type_id : 2 //IP Address
            value : //Generated value
        },
        {
            name : "session_detail.portal_ip",
            type_id : 2 //IP Address
            value : //Generated value
        },
        {
            name : "session_detail.username",
            type_id : 10 //Resource UID
            value : //Generated value
        }
    ],

    // New Events
    session_detail : {
        portal_id : String | UUID of the Portal | eg.
 1ebe42de-86bb-4073-88a4-34284bc5bcbb,
        session_id : String | SessionId of the user session | eg. 17be80fa-7bc2-4675-
b17a-791243938cdf
        client_ip : String | IP Address from which user LoggedIn From | eg. 31.65.180.9
        portal_ip : String | IP Address of the AWS AppStream Instance that is running
 the Portal | eg.240.62.100.169
        username : String | The logged-in username | eg. bobross
    }
}
```

Below is an example of the URLVisit event:

```
{
    activity_id : 99,
    activity_name : "URLVisit",
    ...
    observables : [
        ...
        {
            name : "url",
            type_id : 23 //Unified Resource Locator
        }
    ]
    ...
    url : {
        url_string : String | Full URL path,
        hostname : String | The hostname in the URL
        path : String | Path in the domain
    }
}
```

Below is an example of the PrintJobSubmit event:

```
{
    activity_id : 99,
    activity_name : "PrintJobSubmitted",
    observable : [
        ...
        {
            name : "file.name",
            type_id : 24 // File
        }
    ]
    ...
    file : {
        name : String | The file name,
        type_id : 1 //Regular file
        size : Long | Size in bytes
        ext : String | File extension
    }
}
```

Session Logger metrics for Amazon WorkSpaces Secure Browser

Session Logger emits the following Amazon CloudWatch metrics.

You can use the **SessionLoggerEventDelivered** metric to monitor the aggregate number of events from your portal, or see the number of log files that were delivered by counting the number of data points rather than summing values. We recommend configuring alarms on the **SessionLoggerTargetNotFoundError** and SessionLoggerAccessDeniedError metrics to detect accidental resource or permissions deletion.

> **Note**
>
> Metric data points are collected by each session once per minute and published to Amazon CloudWatch once every 5 minutes. Session Logger metrics are emitted immediately, for each Log File delivery.

Session Logger metrics

Metric	Description	Dimension	Statistics	Unit
SessionLoggerEventDelivered	The number of events each delivered Session Logger file has.	[PortalId]	Average, Sum, Maximum, Minimum	Count
SessionLoggerTargetNotFoundError	The number of log file deliveries that resulted in bucket not found.	[PortalId]	Average, Sum, Maximum, Minimum	Count
SessionLoggerAccessDeniedError	The number of log file deliveries that resulted in permissions denied.	[PortalId]	Average, Sum, Maximum, Minimum	Count

Session events in User Access logging for Amazon WorkSpaces Secure Browser

The following session events are available for User Acess logging:

- **Validation**: The event is sucessfully put to the Kinesis data stream.
- **StartSession**: The user has started a session and is connected to the secure browser session.
- **VisitPage**: The user is visiting a page in the session.
- **EndSession**: The user has terminated the session.

URL navigation logs are recorded from the browser history. URLs not recorded in browser history (either visited in incognito mode or deleted from browser history) are not recorded in logs. It's up to customers to determine whether to turn off incognito mode or history deletion with their browser policy.

Below is an example of each available event. The following fields are always included for each event:

- **timestamp** is included as epoch time in milliseconds.
- **eventType** is included as a string.
- **details** is included as another json object.
- **portalArn** and **userName** are included for every event except for **Validation**.

```
{
  "timestamp": "1665430373875",
  "eventType": "Validation",
  "details": {
    "permission": "Kinesis:PutRecord",
    "userArn": "userArn",
    "operation": "AssociateUserAccessLoggingSettings",
    "userAccessLoggingSettingsArn": "userAccessLoggingSettingsArn"
  }
}

{
  "timestamp": "1665179071723",
  "eventType": "StartSession",
```

```
  "details": {},
  "portalArn": "portalArn",
  "userName": "userName"
}

{
  "timestamp": "1665179084578",
  "eventType": "VisitPage",
  "details": {
    "title": "Amazon",
    "url": "https://www.amazon.com/"
  },
  "portalArn": "portalArn",
  "userName": "userName"
}

{
  "timestamp": "1665179155953",
  "eventType": "EndSession",
  "details": {},
  "portalArn": "portalArn",
  "userName": "userName"
}
```

Guidance for Amazon WorkSpaces Secure Browser users

Administrators use WorkSpaces Secure Browser to create web portals that connect to company websites, such as internal websites, software-as-a-service (SAAS) web applications, or the internet. End users use their existing web browsers to access these web portals in order to launch a session and access content.

The following content helps guide end users who want to learn more about accessing WorkSpaces Secure Browser, launching and configuring a session, and using the toolbar and web browser.

Topics

Browser and device compatibility for Amazon WorkSpaces Secure Browser

Amazon WorkSpaces Secure Browser is powered by the Amazon DCV web browser client, which runs inside a web browser, so no installation is required. The web browser client is supported by common web browsers, such as Chrome and Firefox, and by major desktop operating systems, such as Windows, macOS, and Linux.

For the most up-to-date details on web browser client support, see Web browser client.

> **Note**
>
> Support for webcam is currently only available in Chromium-based browsers, such as Google Chrome and Microsoft Edge. Currently, Apple Safari and Mozilla FireFox do not support webcam.

Web portal access for Amazon WorkSpaces Secure Browser

Your administrator can provide access to your web portal with the following options:

- You can select a link from an email or website, and then sign in with your SAML identity credentials.
- You can sign into your SAML identity provider (such as Okta, Ping, or Azure), and launch a session with one click from your SAML provider's application home page (such as the Okta End User Dashboard or the Azure Myapps portal) .

Session guidance for Amazon WorkSpaces Secure Browser

After you sign into the web portal, you can launch a session and perform various actions during your session.

Topics

Starting a session in Amazon WorkSpaces Secure Browser

After you sign in to launch a session, you will see the **Launching session** message and progress bar. This indicates that Amazon WorkSpaces Secure Browser is creating a session for you. Behind the scenes, Amazon WorkSpaces Secure Browser is creating the instance, launching the managed web browser, and applying administrator settings and browser policies.

If this is your first time signing into your web portal, you will see blue **+** icons in the toolbar. This icon indicates that a tutorial is available, which will guide through the available features in the toolbar. You can use these icons to learn how to:

- Allow browser permissions for the microphone, webcam, and clipboard, by selecting the lock icon next to your local browser, and setting the switch to **On** next to the clipboard, microphone, and camera.

> **Note**
>
> When you enable webcam permissions at the start of your first session, the webcam is briefly enabled and a light on your computer will flash. This grants local browser access to your webcam.

- Enable Amazon WorkSpaces Secure Browser to launch additional monitor windows, by selecting the lock icon in your browser and the setting to **Always allow pop-ups**.

If you ever want to re-launch a tutorial, you can choose **Profile** from the toolbar, **Help**, and **Launch tutorial**.

Using the toolbar in Amazon WorkSpaces Secure Browser

To learn how to use the toolbar, follow these steps.

To move the toolbar, select the lighter bar in the top section of the tool bar, drag it to your desired location, then release it to drop it.

To collapse the toolbar, hover over it, and select the up-arrow button, or double-click the lighter bar in the top section. The collapsed view provides you with more screen real estate, and one-click access to the most commonly used icons.

To increase the size of the display, select the browser window and zoom in. To increase the display size of the toolbar icons and text, select the toolbar and zoom in.

To zoom in or out on a Windows device, follow these steps:

1. Select the toolbar or web content.
2. Press **Ctrl** + **+** to zoom in, or press **Ctrl** + **-** to zoom out.

To zoom in or out on a Mac device, follow these steps:

1. Select the toolbar or web content.
2. Press **Cmd** + **+** to zoom in, or press **Cmd** + **-** to zoom out.

To dock the toolbar to the top of the screen, choose **Preferences**, **General**, and **Docked** under **Toolbar mode**.

The following table includes a description of all the available icons in the toolbar:

Icon	Title	Description
	**Windows**	Move between windows or launch additional browser windows.
	**Launch additional monitor window**	Launch an additional monitor window with a separate browser window. Then drag to your secondary monitor.
	**Full screen**	Launch a full screen experience view.
	**Microphone**	Activate mic input for the session. Use the down arrow to select from a list of available microphones.
	**Webcam**	Activate webcam for the session. Use the down arrow to select from a list of available webcams.
	**Preferences**	Access the **General** and **Keyboard** menus. From the **General** menu, toggle between light and dark mode, activate the keyboard input selector (for changing the keyboard language), and switch between streaming mode or display resolution. From the **Keyboard** menu, change the option and command key settings (on Mac devices), or activate **Functions** (see below).
	**Profile**	End your session, view performance metrics, access **Feedback** and **Help**, and learn about Amazon WorkSpaces Web. **End Session** ends the Amazon WorkSpaces Web session. **Performance metrics** displays the frame rate, network latency, and bandwidth usage graph. This information is useful for administrators when investigating issues with the service. **Feedback** provides you with an email address to share feedback to the Amazon WorkSpaces Web team. **Help** provides you with access to Frequently Asked Questions, such as how to use the clipboard, microphone, and webcam during the session, or how to troubleshoot launching an additional monitor window. From help, you can also launch the tutorial or user guide. **About** provides more information about Amazon WorkSpaces Web.
	**Notifications**	Get one-click access to session notifications.
	**Clipboard**	Access clipboard shortcut descriptions, links to set the command key preference, and troubleshoot clipboard permissions from the local web browser. You can use the content preview text box to test clipboard functionality. This icon only displays if clipboard permission is granted by your administator.
	**Files**	From the files menu, you can upload content to the remote browser. Once uploaded, you can rename, download, or delete, as well as create folders in the temporary file menu. All files and data in **Files** are deleted at the end of the session. This icon only displays if **Files** permission is granted by your administator.

> **Note**
>
> The Clipboard and Files icons are hidden by default, unless your administrator grants these permissions. Only administrators can enable or disable clipboard and files on a web portal. If these icons are hidden and you need to access them, contact your administrator.

Using the browser in Amazon WorkSpaces Secure Browser

When you start your session, the browser displays the **Startup URL**, which is a URL chosen by your administrator. If administrator hasn't chosen a **Startup URL**, you will see the default new tab experience from Google Chrome.

From the browser, you can open tabs, launch additional browser windows (from the Windows toolbar icon or the browser's triple dot menu), enter a URL or search in the URL bar, or go to websites from managed bookmarks. To access bookmarks for the web portal, open the **Managed Bookmarks** folder on the bookmarks bar (beneath the URL bar), or open the bookmarks manager from the triple dot menu on the right side of the URL bar.

To resize or move the browser window, drag down the Chrome tab strip. This allows more screen real estate for multiple browser windows during the session.

> **Note**
>
> Browser features, such as Incognito mode, might not be available during your session if your administrator has turned them off.

Ending a session in Amazon WorkSpaces Secure Browser

To end a session, choose **Profile** and **End session**. After a session ends, Amazon WorkSpaces Secure Browser deletes all of the data from the session. No browser data, such as open websites or history, or files or data from File Explorer are available after a session ends.

If you close a tab during an active session, the session ends after a period of time set by your administrator. If you close the tab and revisit the web portal before this timeout takes effect, you can join the current session and see all of your previous session data, such as open websites and files.

Troubleshooting user issues in Amazon WorkSpaces Secure Browser

If you encounter any of the following issues while using WorkSpaces Secure Browser, try the following resolutions.

*My Amazon WorkSpaces Secure Browser portal won't let me sign in. I received an error message that says "**Your web portal isn't set up yet. Contact you IT administrator for help.**"*

Your administrator needs to complete portal creation with a SAML 2.0 identity provider to enable you to sign in. Contact your administrator for help.

*My portal won't launch a session. I received an error message that says"**Failed to reserve session. There was an internal error. Please retry.**"*

There was a problem with your web portal session launch. Try to launch the session again. If this continues, contact your administrator for help.

I can't use the clipboard, microphone, or webcam.

To allow browser permissions, select the lock icon next to the URL, and toggle the blue switch next to **Clipboard**, **Microphone**, **Camera**, and **Pop-ups and redirects** to turn these features on.

> **Note**
>
> If your web browser doesn't support video or audio input, these options won't appear on the toolbar.

Amazon WorkSpaces Secure Browser real-time audio-video (AV) redirects your local webcam video and microphone audio input to the browser streaming session. This way, you can use your local devices for video and audio conferencing within your streaming session with Chromium-based web browsers, such as Google Chrome or Microsoft Edge. Webcam is not currently supported in non-Chromium browsers.

For information about how to configure Google Chrome, see Use your camera & microphone.

My web portal won't launch an additional monitor window.

If you try to launch dual monitors and see a **Pop-ups blocked** icon at the end of the address bar on the top browser, select the icon and the radio button next to **Always allow pop-ups and redirects**.

With pop-ups allowed, select the **Dual monitor** icon on the toolbar to launch a new window, reposition the window on your monitor, and drag a browser tab to the window.

*When I try to download files from the **Files** pane, nothing happens.*

If you try to download files from the **Files** pane and see a **Pop-ups blocked** icon at the end of the address bar on the top browser, select the icon and the radio button next to **Always allow pop-ups and redirects**. With pop-ups allowed, try downloading the files again.

How can I tell which microphone and/or webcam is being used, and how can I change it?

Click the down arrow icon next to the microphone or camera. The menu displays available devices, with a checkmark indicating your current device. Select a different device to change the device you want to use for your session.

Single sign-on extension for Amazon WorkSpaces Secure Browser

Amazon WorkSpaces Secure Browser offers an extension for single sign-on with Chrome and Firefox browsers on desktop computers. If your administrator has enabled the extension, the web portal will ask you to install the extension when you sign in.

Amazon WorkSpaces Secure Browser built the extension to enable single sign-on to websites during your session. For example, if you sign into your web portal using a SAML 2.0 identity provider (such as Okta or Ping), and you visit a website during your session that uses the same identity provider, the extension can make it easier to access the website by removing additional sign-in prompts.

You aren't required to install the extension to access your web portal, but it can improve your experience by reducing the number of times you are asked to enter your username and password.

When you sign in, the extension locates the cookies your administrator listed for your session. All of the data that the extension locates is encrypted at rest and during transit. None of this data is stored in your local browser. When you end your session, all of your session data (such as open tabs, files downloaded, and cookies delivered to or created during the session) is deleted.

Topics

- Troubleshooting the single sign-on extension for Amazon WorkSpaces Secure Browser

Single sign-on extension compatibility for Amazon WorkSpaces Secure Browser

The single sign-on extension works with the following devices and browsers:

- Devices
  - Laptops
  - Desktop computers
- Browsers
  - Google Chrome
  - Mozilla Firefox

Installing the single sign-on extension for Amazon WorkSpaces Secure Browser

To install the single sign-on extension, follow these steps.

When you sign into the portal, follow the prompt to install the extension for your Chrome or Firefox browser. You only have to do this one time for each web browser.

If you switch devices, switch to a different browser on the same device, or delete the extension from your local browser, you'll see a prompt to install the extension when you start your next session.

To ensure that the extension works as expected, use the extension in a normal browsing window, instead of Incognito (Chrome) or Private Browsing (Firefox).

Troubleshooting the single sign-on extension for Amazon WorkSpaces Secure Browser

While using the single sign-on extension, you might encounter the following issue.

If you have the extension installed, but you're still being asked to sign in during your session, follow these steps:

1. Make sure that you have the Amazon WorkSpaces Secure Browser extension installed on your browser. If you deleted your browser data, you might have removed the extension by accident.
2. Make sure that you are not Incognito (Chrome) or Private Browsing (Firefox). These modes can cause issues with extensions.
3. If the issue persists, contact your portal administrator for additional help.

Document history for the Amazon WorkSpaces Secure Browser Administration Guide

The following table describes the documentation releases for Amazon WorkSpaces Secure Browser.

Change	Description	Date
Session Logger	Set up Session Logger to capture a wide range of session events.	August 1, 2025
CloudWatch metrics	Updated CloudWatch metrics	July 21, 2025
Toolbar controls	With toolbar controls, you can configure the toolbar presentation for end user sessions.	February 21, 2025
Access APIs using an interface VPC endpoint (AWS PrivateLink)	Directly call the Amazon WorkSpaces Secure Browser API endpoint from within a private cloud (VPC), instead of connecting over the internet.	January 10, 2025
Data Protection Settings	Add Data Protection Settings to help protect data from being shared during a session.	November 20, 2024
FIPS endpoints	Protect data in transit with FIPS endpoints.	October 7, 2024
Session management dashboard	Use the session management dashboard to monitor and manage active and complete sessions.	September 19, 2024
Allow deep links	Allow portals to receive deep links that connect users to	June 25, 2024

	a specific website during a session.	
Managed policy update	Added AmazonWorkSpacesSecureBrowserReadOnly managed policy	June 24, 2024
Use the toolbar to zoom	You can increase the size of the display, icons, and text with the toolbar.	May 1, 2024
New web portal settings	You can now specify **Instance type** and **Max concurrent user limit** for your web portal.	April 22, 2024
CloudWatch metrics	Added GlobalCpuPercent and GlobalMemoryPercent metrics.	February 26, 2024
Set up URL filtering	You can use Chrome Policy to filter which URLs users can access from their remote browser.	February 21, 2024
IdP authentication types	You can choose either the standard or IAM Identity Center authentication type.	February 5, 2024
Enable extension for single sign-on	You can enable an extension for your end users to have a better portal sign-on experience.	August 28, 2023

User guidance for Amazon WorkSpaces Secure Browser	Added content to help guide end users, who want to learn more about accessing Amazon WorkSpaces Secure Browser, launching and configuring a session, and using the toolbar and web browser.	July 17, 2023
IP access controls	WorkSpaces Secure Browser allows you to control which IP addresses your web portal can be accessed from.	May 31, 2023
Managed policy update	Updated AmazonWorkSpacesWebReadOnly managed policy	May 15, 2023
Configure identity provider update	WorkSpaces Secure Browser offers two authentication types: **Standard** and **AWS IAM Identity Center**	March 15, 2023
Browser policy update	Updated and restructured browser policy section	January 31, 2023
Managed policy update	Updated AmazonWorkSpacesWebServiceRolePolicy managed policy	December 15, 2022
Allowlist and blocklist	Specify the **Allowlist** and **Blocklist** to specify a list of domains that your users can or cannot access.	November 14, 2022
Managed policy update	Updated AmazonWorkSpacesWebReadOnly managed policy	November 2, 2022

Managed policy update	Updated AmazonWorkSpacesWebServiceRolePolicy managed policy	October 24, 2022
User access logging	Set up user access logging to record user events	October 17, 2022
Networking updates	Various updates to "Networking and access" section	September 22, 2022
Managed policy update	Updated AmazonWorkSpacesWebServiceRolePolicy managed policy	September 6, 2022
Configure user sessions	Configure the Input Method Editor (IME) and in-session localization	July 28, 2022
Networking updates	Various updates to "Networking and access" section	July 7, 2022
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 Secure Browser Administration Guide	November 30, 2021