# Table of Contents

What Is AppStream 2.0? ............................................................................................................................. 1
   Features ................................................................................................................................................. 1
   Key Concepts ......................................................................................................................................... 2
   How to Get Started ............................................................................................................................... 2
   Accessing AppStream 2.0 .................................................................................................................... 3

Setting Up .................................................................................................................................................. 4
   Sign Up for AWS ................................................................................................................................. 4

Getting Started ........................................................................................................................................ 5
   Step 1: Set Up a Sample Stack, Choose an Image, and Configure a Fleet ............................................ 5
   Step 2: Provide Access to Users ......................................................................................................... 6
   Resources ............................................................................................................................................. 6

Network Settings ....................................................................................................................................... 8
   Port Requirements for User Connections to AppStream 2.0 ............................................................... 8
      Ports for AppStream 2.0 User Devices ............................................................................................... 8
      Whitelisted Domains ......................................................................................................................... 9
   Port Requirements for AppStream 2.0 Connections ........................................................................... 9
      Network Interfaces ............................................................................................................................. 9
      Management Network Interface IP Address Range and Ports ....................................................... 9
      Customer Network Interface Ports ................................................................................................. 10
   Network Setup Guidelines ................................................................................................................. 10
      Fleets ............................................................................................................................................ 10
      Image Builders ............................................................................................................................... 11
   Security Groups .................................................................................................................................... 11
   Home Folders and VPC Endpoints ....................................................................................................... 12
   Enabling Internet Access Using a Public Subnet ................................................................................... 12
      Enabling Internet Access for a Fleet ............................................................................................... 13
      Enabling Internet Access for an Image Builder ............................................................................. 13
   Enabling Internet Access Using a NAT Gateway ................................................................................ 14
      Enabling Internet Access for a Fleet Using a NAT Gateway .......................................................... 14
      Enabling Internet Access for an Image Builder Using a NAT Gateway ........................................ 15

Image Builders ......................................................................................................................................... 16
   Actions ............................................................................................................................................... 16
   Tutorial: Create a Custom Image ......................................................................................................... 17
      Step 1: Create an Image Builder ....................................................................................................... 17
      Step 2: Install Applications to an Image ........................................................................................... 19
      Step 3: Add Applications to an Image ............................................................................................. 19
      Step 4: Optimize Applications ......................................................................................................... 20
      Step 5: Create an Image .................................................................................................................... 20
      Step 6: Clean Up ............................................................................................................................... 21

Images ....................................................................................................................................................... 22
   Windows Image Versions ................................................................................................................... 22
   Amazon AppStream 2.0 Agent Versions ............................................................................................ 24

Stacks and Fleets ....................................................................................................................................... 25
   Fleet Type ......................................................................................................................................... 25
   Session Context .................................................................................................................................... 25
   Instance Families ............................................................................................................................... 26
   Create Stacks and Fleets .................................................................................................................... 27
      Create a Fleet ................................................................................................................................. 28
      Create a Stack ............................................................................................................................... 29
      Provide Access to Users ................................................................................................................ 29
      Clean Up Resources ........................................................................................................................ 29
   Fleet Auto Scaling ............................................................................................................................ 30
   Scaling Concepts .............................................................................................................................. 30
   Managing Fleet Scaling Using the Console ....................................................................................... 31
<table>
<thead>
<tr>
<th>Tagging Your Resources</th>
<th>Working with Tags in the AppStream 2.0 Console</th>
<th>Troubleshooting Image Builder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag Restrictions</td>
<td>Working with Tags by Using the AppStream 2.0 API, an AWS SDK, or AWS CLI</td>
<td>Troubleshooting Active Directory Domain Join</td>
</tr>
<tr>
<td>Working with Tags by Using the AppStream 2.0 API, an AWS SDK, or AWS CLI</td>
<td>Troubleshooting Active Directory Domain Join</td>
<td>Troubleshooting Active Directory Domain Join</td>
</tr>
<tr>
<td>Tagging Basics</td>
<td>Troubleshooting Active Directory Domain Join</td>
<td>Troubleshooting Active Directory Domain Join</td>
</tr>
<tr>
<td>Tag Restrictions</td>
<td>Trubleshooting Active Directory Domain Join</td>
<td>Troubleshooting Active Directory Domain Join</td>
</tr>
<tr>
<td>Tagging Basics</td>
<td>Troubleshooting Active Directory Domain Join</td>
<td>Troubleshooting Active Directory Domain Join</td>
</tr>
<tr>
<td>Tag Restrictions</td>
<td>Troubleshooting Active Directory Domain Join</td>
<td>Troubleshooting Active Directory Domain Join</td>
</tr>
</tbody>
</table>

**Troubleshooting Active Directory Domain Join**

- My image builders and fleet instances are stuck in "pending" status
- My users aren't able to log in with the SAML app
- My fleet instances work for one user but don't cycle correctly
- My user Group Policy Objects aren't applying successfully
- My AppStream 2.0 streaming instances aren't joining the Active Directory domain
- User login is taking a long time to complete on a domain-joined streaming session
- The changes I made in the image builder aren't reflected in end user streaming sessions
- Unable to access a domain resource in a domain joined streaming session but could access the resource from a domain-joined image builder

**Troubleshooting Notification Codes**

- **General Troubleshooting**
  - SAML federation is not working. The user is not authorized to view AppStream 2.0 applications.
  - After federating from an ADFS portal, my streaming session doesn't start. I am getting the error "Sorry connection went down".
  - I get an invalid redirect URL error.
  - My stack's Home Folders aren't working correctly.

- **Troubleshooting Image Builder**
  - I cannot connect to the internet from my image builder.
  - When I tried installing my application, I see an error that the operating system version is not supported.
  - When I connect to my image builder, I see a login screen asking me to enter Ctrl+Alt+Delete to log in. However, my local machine intercepts the key strokes.
  - When I switched between admin and test modes, I saw a request for a password. I don't know how to get a password.
  - I get an error when I add my installed application.
  - I accidentally quit a background service on the image builder and got disconnected. I am now unable to connect to that image builder.
  - The application fails to launch in test mode.
  - The application could not connect to a network resource in my VPC.
  - I customized my image builder desktop, but my changes are not available when connecting to a session after launching a fleet from the image I created.
  - My application is missing a command line parameter when launching.
  - I am unable to use my image with a fleet after installing an antivirus application.
  - My image creation failed.

- **Troubleshooting Active Directory Domain Join**
  - My image builders and fleet instances are stuck in "pending" status
  - My users aren't able to log in with the SAML app
  - My fleet instances work for one user but don't cycle correctly
  - My user Group Policy Objects aren't applying successfully
  - My AppStream 2.0 streaming instances aren't joining the Active Directory domain
  - User login is taking a long time to complete on a domain-joined streaming session
  - The changes I made in the image builder aren't reflected in end user streaming sessions
  - Unable to access a domain resource in a domain joined streaming session but could access the resource from a domain-joined image builder

<table>
<thead>
<tr>
<th>Limits</th>
<th>Document History</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Directory Domain Join</td>
<td>Document History</td>
</tr>
</tbody>
</table>
What Is Amazon AppStream 2.0?

Amazon AppStream 2.0 is a fully managed application streaming service that enables you to stream desktop applications from Amazon Web Services (AWS) to any device running a web browser, without rewriting them. AppStream 2.0 manages the AWS resources required to host and run your applications, scales automatically, and provides access to your users on demand. AppStream 2.0 provides users access to the applications they need on the device of their choice, with a responsive, fluid user experience that is indistinguishable from natively installed applications. There are no files to download and no time-consuming installations.

With AppStream 2.0, you can easily add your existing desktop applications to AWS and instantly start streaming them to an HTML5 compatible browser. You can maintain a single version of each of your apps, which makes application management easier. Your users always access the latest versions of their applications. Your applications run on AWS compute resources, and data is never stored on users' devices, which means they always get a high performance, secure experience.

Unlike traditional on-premises solutions for desktop application streaming, AppStream 2.0 offers pay-as-you-go pricing, with no upfront investment and no infrastructure to maintain. You can scale instantly and globally, ensuring that your users always have the best possible experience.

For more information, see AppStream 2.0.

Features

Using Amazon AppStream 2.0 provides the following advantages:

**Run desktop applications on any device**

Your desktop applications run securely in an HTML5 web browser on Windows and Linux PCs, Macs, and Chromebooks.

**Secure applications and data**

Applications and data remain on AWS — only encrypted pixels are streamed to end users. Applications run on an AppStream 2.0 instance dedicated to each user so that compute resources are not shared. Applications can run inside your own virtual private cloud (VPC), and you can use Amazon VPC security features to control access. This enables you to isolate your applications and deliver them in a secure way.

**Consistent, scalable performance**

AppStream 2.0 runs on AWS with access to compute capabilities not available on local devices, which means that your applications run with consistently high performance. You can instantly scale locally and globally, and ensure that your users always get a low-latency experience. Unlike on-premises solutions, you can quickly deploy your applications to the AWS region that is closest to your users, and start streaming with no incremental capital investment.

**Integrate with your IT environment**

Integrate with your existing AWS services and your on-premises environments. By running applications inside your VPCs, your users can access data and other resources that you have in AWS. This reduces the movement of data between AWS and your environment and provides a faster user experience.

Integrate with your existing Microsoft Active Directory environment. This enables you to use existing Active Directory governance, user experience, and security policies with your streaming applications.
Configure identity federation, which allows your users to access their applications using their corporate credentials. You can also allow authenticated access to your IT resources from applications running on AppStream 2.0.

**Choose the fleet type that meets your needs**

There are two types of fleets:

- **Always-On** — Your instances run all the time, even when no users are streaming applications. Use an Always-On fleet to provide your users with instant access to their applications.

- **On-Demand** — Your instances run only when users are streaming applications. Idle instances that are available for streaming are in a stopped state. Use an On-Demand fleet to optimize your streaming charges and provide your users with access to their applications after a 1-2 minute wait.

For more information, see Amazon AppStream 2.0 Pricing.

**Key Concepts**

To get the most out of AppStream 2.0, be familiar with the following concepts:

**image builder**

An *image builder* is a virtual machine that you use to create or modify an image. You can launch and connect to an image builder by using the AWS Management Console. After you are connected to an image builder, you can install, add, and test your apps, and then use the image builder to publish an image.

**image**

An *image* contains applications that are streamed to users. AWS provides base images that you can use to create images that include your own applications.

**fleet**

A *fleet* consists of streaming instances that run the image that you specify. You can set the desired number of streaming instances for your fleet and configure policies to scale your fleet automatically based on demand. Note that one user requires one instance.

**stack**

A *stack* consists of a fleet, user access policies, and storage configurations. You set up a stack to start streaming applications to users.

**user pool**

Use the *user pool* to manage users and their assigned stacks.

**How to Get Started**

If you are using AppStream 2.0 for the first time, you can use the Try it Now feature or follow the Getting Started with Amazon AppStream 2.0 (p. 5) tutorial (both are available in the AppStream 2.0 console).

- **Try it Now** provides you with a free trial experience that allows you to easily start desktop applications from your desktop browser.

- The Getting Started tutorial enables you to set up application streaming by using sample applications or your own applications. If you decide to start by using sample applications, you can always add your own applications later.
For more information about these two options, see Amazon AppStream 2.0 FAQs.

When you use the service for the first time, AppStream 2.0 creates an AWS Identity and Access Management (IAM) role to create and manage AppStream 2.0 resources on your behalf.

To use the Try It Now feature
2. Choose Try it now.
3. Sign in using your AWS account credentials, if requested.
4. Read the terms and conditions and choose Agree and Continue.
5. From the list of applications shown, select one to try.

To run the Getting Started tutorial
2. Choose Get Started.
3. Select the option to learn more about AppStream 2.0 resources.

Accessing AppStream 2.0

You can work with AppStream 2.0 using any of the following interfaces:

AWS Management Console
The console is a browser-based interface to manage AppStream 2.0 resources. For more information, see Getting Started with Amazon AppStream 2.0 (p. 5).

AWS command line tools
AWS provides two sets of command line tools: the AWS Command Line Interface (AWS CLI) and the AWS Tools for Windows PowerShell. To use the AWS CLI to run AppStream 2.0 commands, see Amazon AppStream 2.0 Command Line Reference.

AWS SDKs
You can access AppStream 2.0 from a variety of programming languages. The SDKs automatically take care of tasks such as the following:
• Setting up an AppStream 2.0 stack or fleet
• Getting an application streaming URL to your stack
• Describing your resources

For more information, see Tools for Amazon Web Services.
Setting Up for Amazon AppStream 2.0

Complete the following tasks to get set up for Amazon AppStream 2.0.

Sign Up for AWS

When you sign up for AWS, your AWS account is automatically signed up for all services, including AppStream 2.0. You are charged only for the services that you use.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

1. Open https://aws.amazon.com/, and then choose Create an AWS Account.

   Note
   This might be unavailable in your browser if you previously signed into the AWS Management Console. In that case, choose Sign in to a different account, and then choose Create a new AWS account.

2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.
Getting Started with Amazon AppStream 2.0

To stream your applications, Amazon AppStream 2.0 requires an environment consisting of a stack, a fleet, and at least one application image. This tutorial describes how to configure a sample AppStream 2.0 environment for application streaming and give users access to that stream.

Tasks
- Step 1: Set Up a Sample Stack, Choose an Image, and Configure a Fleet (p. 5)
- Step 2: Provide Access to Users (p. 6)
- Resources (p. 6)

Step 1: Set Up a Sample Stack, Choose an Image, and Configure a Fleet

Before you can stream your applications, you need to set up a stack, choose an image that has applications installed, and configure a fleet. In this step, you use a template to help simplify these tasks.

To set up a sample stack, choose an image, and configure a fleet

2. Choose Get Started if you are new to the console, or Quick Links from the left navigation menu. Choose Set up with sample apps.
3. For Step 1: Stack Details, keep the default sample stack name and description, or type new ones. You can also specify a redirect URL. If you specify a redirect URL, after users' streaming sessions end, the users are redirected to that URL. Choose Next.
4. For Step 2: Choose Image, choose an image, and then choose Next. The sample image contains pre-installed open source applications for evaluation purposes. For more information, see Amazon AppStream 2.0 Windows Image Version History (p. 22).
5. For Step 3: Configure Fleet, we recommend that you keep the default values and choose Next. You can change most of these values after fleet creation.

- Choose instance type — Choose the instance type that matches the performance requirements of your applications. All streaming instances in your fleet launch with the instance type that you select. For more information, see AppStream 2.0 Instance Families (p. 26).
- Fleet type — Choose the fleet type that suits your use case. The fleet type determines its immediate availability and how you pay for it.
- Maximum session duration — Choose the maximum amount of time that a streaming session can remain active. If users are still connected to a streaming session five minutes before this limit is reached, they are prompted to save any open documents before being disconnected.
- Disconnect timeout — Choose the time that a streaming instance should remain 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 the previous session. Otherwise, they are connected to a new session with a new instance. If you associate a stack with a fleet for which
a redirect URL is specified, after users' streaming sessions end, the users are redirected to that URL.

If a user ends the session by choosing **End Session** on the streaming session toolbar, the disconnect timeout does not apply. Instead, the user is prompted to save any open documents, and then immediately disconnected from the streaming instance.

- **Minimum capacity** — Choose a minimum number of instances for your fleet based on the minimum number of expected concurrent users. Every unique user session is served by an instance. For example, to have your stack support 100 concurrent users during low demand, specify a minimum capacity of 100. This ensures that 100 instances are running even if there are fewer than 100 users.

- **Maximum capacity** — Choose a maximum number of instances for your fleet based on the maximum number of expected concurrent users. Every unique user session is served by an instance. For example, to have your stack support 500 concurrent users during high demand, specify a maximum capacity of 500. This ensures that up to 500 instances can be created on demand.

6. For **Step 4: Configure Network**, choose a VPC and two subnets with access to the network resources that your application needs. If you don't have a VPC or subnets, you can create them using the links provided and then click the refresh icons. For **Security groups**, you can select up to five security groups. Otherwise, the default security group is used. For more information, see Network Settings for Amazon AppStream 2.0 (p. 8).

7. For **Step 5: Enable Storage**, keep the default setting, **Enable Home Folders**, and choose **Review**. Home Folders offer persistent storage for AppStream 2.0 streaming sessions. For more information, see Persistent Storage with AppStream 2.0 Home Folders (p. 36).

8. Choose **Create**.

   After the service sets up resources, the **Stacks** page appears. The status of your new stack is **Active** when it is available to work with from the console.

   Optionally, you can apply one or more tags to help manage the stack. Choose **Tags**, choose **Add/Edit Tags**, choose **Add Tag**, specify the key and value for the tag, and then choose **Save**. For more information, see Tagging Your Amazon AppStream 2.0 Resources (p. 71).

---

**Step 2: Provide Access to Users**

After you create a stack, each user needs an active URL for access. The AppStream 2.0 User Pool feature enables you to create and manage users, using a permanent login portal URL. For more information, see Manage Access Using the AppStream 2.0 User Pool (p. 41). To quickly test application streaming without setting up users, create a temporary URL as shown below.

**To provide access to users with a temporary URL**

1. In the navigation pane, choose **Stacks**. Select the radio button for the stack, and then choose **Actions, Create Streaming URL**.
2. For **User id**, type the user ID. Select an expiration time, which determines how long the generated URL is valid.
3. To view the user ID and URL, choose **Get URL**.
4. To copy the link to the clipboard, choose **Copy Link**.

---

**Resources**

For more information, see the following:
• Learn how to use the AppStream 2.0 image builder to add your own apps and create images you can stream. For more information, see Tutorial: Create a Custom Image (p. 17).
• Provide persistent storage for your session users using AppStream 2.0 Home Folders. For more information, see Persistent Storage with AppStream 2.0 Home Folders (p. 36).
• Integrate your AppStream 2.0 streaming resources with your Microsoft Active Directory environment. For more information, see Using Active Directory Domains with AppStream 2.0 (p. 52).
• Control who has access to your AppStream 2.0 streaming instances. For more information, see Controlling Access to Amazon AppStream 2.0 with IAM Roles and Policies (p. 67), Manage Access Using the AppStream 2.0 User Pool (p. 41) and Single Sign-on Access to AppStream 2.0 Using SAML 2.0 (p. 45).
• Monitor your AppStream 2.0 resources using Amazon CloudWatch. For more information, see AppStream 2.0 Metrics and Dimensions (p. 64).
• Troubleshoot your AppStream 2.0 streaming experience. For more information, see Troubleshooting (p. 75).
Network Settings for Amazon AppStream 2.0

The following sections contain information about enabling users to connect to AppStream 2.0 streaming instances and enabling your AppStream 2.0 fleets and image builders to access network resources and the internet.

Contents

- Port Requirements for User Connections to Amazon AppStream 2.0 (p. 8)
  - Ports for AppStream 2.0 User Devices (p. 8)
  - Whitelisted Domains (p. 9)
- Port Requirements for Amazon AppStream 2.0 Connections to Network Resources and the Internet (p. 9)
  - Network Interfaces (p. 9)
  - Management Network Interface IP Address Range and Ports (p. 9)
  - Customer Network Interface Ports (p. 10)
- Network Setup Guidelines (p. 10)
  - Fleets (p. 10)
  - Image Builders (p. 11)
- Security Groups (p. 11)
- Home Folders and VPC Endpoints (p. 12)
- Enabling Internet Access Using a Public Subnet (p. 12)
  - Enabling Internet Access for a Fleet (p. 13)
  - Enabling Internet Access for an Image Builder (p. 13)
- Enabling Internet Access Using a NAT Gateway (p. 14)
  - Enabling Internet Access for a Fleet Using a NAT Gateway (p. 14)
  - Enabling Internet Access for an Image Builder Using a NAT Gateway (p. 15)

Port Requirements for User Connections to Amazon AppStream 2.0

For AppStream 2.0 users to connect to streaming instances and stream applications, the network that the users' devices are connected to must allow access to certain IP addresses and ports.

Ports for AppStream 2.0 User Devices

AppStream 2.0 users' devices require outbound access on port 443 (TCP), and if you are using DNS servers for domain name resolution, port 53 (UDP).

- Port 443 is used for HTTPS communication between AppStream 2.0 users' devices and streaming instances. 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.
Port 53 is used for communication between AppStream 2.0 users' devices and your DNS servers. The port must be open to the IP addresses for your DNS servers so that public domain names can be resolved. This port is optional if you are not using DNS servers for domain name resolution.

**Whitelisted Domains**

For AppStream 2.0 users to access streaming instances, you must whitelist the following domains on the network from which users are trying to access the streaming instances.

- Session Gateway: *.amazonappstream.com
- CloudFront: *.cloudfront.net

Amazon Web Services (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 in the Amazon Web Services General Reference. Or, if you are using AWS Tools for Windows PowerShell, you can access the same information by using the Get-AWSPublicIpAddressRange cmdlet. For more information, see Querying the Public IP Address Ranges for AWS.

**Port Requirements for Amazon AppStream 2.0 Connections to Network Resources and the Internet**

To enable AppStream 2.0 connectivity to network resources and the internet, configure your streaming instances as follows.

**Network Interfaces**

Each AppStream 2.0 streaming instance has the following network interfaces:

- The customer network interface provides connectivity to the resources within your VPC, as well as the internet, and is used to join the streaming instance to your directory.
- The management network interface is connected to a secure AppStream 2.0 management network. It is used for interactive streaming of the streaming instance to a user's device, and to allow AppStream 2.0 to manage the streaming instance.

AppStream 2.0 selects the IP address for the management network interface from the following private IP address range: 198.19.0.0/16. Do not use this range for your VPC CIDR or peer your VPC with another VPC with this range, as this might create a conflict and cause streaming instances to be unreachable. Also, do not modify or delete any of the network interfaces attached to a streaming instance, as this might also cause the streaming instance to become unreachable.

**Management Network Interface IP Address Range and Ports**

The management network interface IP address range is 198.19.0.0/16. The following ports must be open on the management network interface of all streaming instances:
• Inbound TCP on port 8300. This is used for establishment of the streaming connection.
• Inbound TCP on port 8443. This is used for management of the streaming instance by AppStream 2.0.

Limit the inbound range on the management network interface to 198.19.0.0/16.

Under normal circumstances, AppStream 2.0 correctly configures these ports for your streaming instances. If any security or firewall software is installed on a streaming instance that blocks any of these ports, the streaming instance may not function correctly or may be unreachable.

Customer Network Interface Ports

• For internet connectivity, the following ports must be open to all destinations. If you are using a modified or custom security group, you need to add the required rules manually. For more information, see Security Group Rules in the Amazon VPC User Guide.
  • TCP 80 (HTTP)
  • TCP 443 (HTTPS)
• If you join your streaming instances to a directory, the following ports must be open between your AppStream 2.0 VPC and your directory controllers.
  • TCP/UDP 53 - DNS
  • TCP/UDP 88 - Kerberos authentication
  • UDP 123 - NTP
  • TCP 135 - RPC
  • UDP 137-138 - Netlogon
  • TCP 139 - Netlogon
  • TCP/UDP 389 - LDAP
  • TCP/UDP 445 - SMB
  • TCP 1024-65535 - Dynamic ports for RPC

For a complete list of ports, see Active Directory and Active Directory Domain Services Port Requirements in the Microsoft documentation.

• All streaming instances require that port 80 (HTTP) be open to IP address 169.254.169.254 to allow access to the EC2 metadata service. Any HTTP proxy assigned to your streaming instances must exclude 169.254.169.254.

Network Setup Guidelines

There are some network setup guidelines to consider for fleets and image builders. If your fleets and image builders require internet access, you can use the Default Internet Access feature. You could also manually control internet access using an advanced networking configuration, such as a VPC with NAT gateways. For more information, see Enabling Internet Access Using a Public Subnet (p. 12) and Enabling Internet Access Using a NAT Gateway (p. 14).

Fleets

You can provide subnets to establish network connections from your fleet instances to your VPC. We recommend that you specify two private subnets from different Availability Zones for high availability and fault tolerance. Also, ensure that the network resources for your applications are accessible through both of the specified private subnets.

AppStream 2.0 creates as many elastic network interfaces as the maximum desired capacity of your fleet. The following guidelines will help you set up a VPC to support scaling behavior for your fleet.
• Make sure that your AWS account has sufficient elastic network interface capacity to support the scaling requirements of your fleet. If you are planning to launch a large fleet of streaming instances, contact AWS Support and request a higher ENI limit to match the maximum number of instances that you plan to launch.

• Specify subnets with a sufficient number of elastic IP addresses to match the maximum desired capacity of your fleet.

• Use security groups to provide your VPC with specific security settings. For more information, see Security Groups (p. 11).

Image Builders

You can choose one subnet while launching an image builder. Ensure the subnet accessibility of the network resources, with which your applications may interact. The typical resources required for the successful execution of your apps may include licensing servers, database servers, file servers, and so on.

Security Groups

You can provide additional access control to your VPC from streaming instances in a fleet or an image builder in Amazon AppStream 2.0 by associating them with VPC security groups. Security groups that belong to your VPC allow you to control the network traffic between AppStream 2.0 streaming instances and VPC resources such as license servers, file servers, and database servers. For more information, see Security Groups for your VPC in the Amazon VPC User Guide.

The rules that you define for your VPC security group are applied when the security group is associated with a fleet or image builder. The security group rules determine what network traffic is allowed from your streaming instances. For more information, see Security Group Rules in the Amazon VPC User Guide.

You can associate up to five security groups while launching a new image builder or while creating a new fleet. You can also associate security groups to an existing fleet or change the security groups of a fleet. For more information, see Working with Security Groups in the Amazon VPC User Guide.

If you don't select a security group, your image builder or fleet is associated with the default security group for your VPC. For more information, see Default Security Group for Your VPC in the Amazon VPC User Guide.

Use these additional considerations when using security groups with AppStream 2.0.

• All end user data, such as internet traffic, Home folder data, or application communication with VPC resources, are affected by the security groups associated with the streaming instance.

• Streaming pixel data is not affected by security groups.

• If you have enabled default internet access for your fleet or image builder, the rules of the associated security groups must allow internet access.

You can create or edit rules for your security groups or create new security groups using the Amazon VPC console.

• To associate security groups with an image builder — Follow the instructions at Step 1: Create an Image Builder (p. 17).

• To associate security groups with a fleet
  • While creating the fleet — Follow the instructions at Create a Fleet (p. 28).
  • For an existing fleet — Edit the fleet settings using the AWS Management Console.

You can also associate security groups to your fleets using the AWS CLI and SDKs.
Home Folders and VPC Endpoints

To support Home Folders on a private network, AppStream 2.0 needs access permissions to the VPC endpoint. To enable AppStream 2.0 access to your private Amazon S3 endpoint, attach a custom policy, as defined below, to your VPC endpoint for Amazon S3. For more information about private Amazon S3 endpoints, see VPC Endpoints and Endpoints for Amazon S3 in the Amazon VPC User Guide.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow-AppStream-to-access-specific-bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:sts::account-id-without-hyphens:assumed-role/AmazonAppStreamServiceAccess/AppStream2.0"
            },
            "Action": [
                "s3:ListBucket",
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:GetObjectVersion",
                "s3:DeleteObjectVersion"
            ],
            "Resource": "arn:aws:s3:::appstream2-36fb080bb8-*"
        }
    ]
}
```

Enabling Internet Access Using a Public Subnet

AppStream 2.0 can provide your fleets with a default internet connection by using your Amazon VPC public subnet. This subnet has a route to the internet through an internet gateway.

AppStream 2.0 enables internet connectivity by associating an Elastic IP address to the network interface that is attached from the streaming instance to your VPC public subnet. You can have a VPC with a public subnet in several ways:

**Default VPC**

Your AWS account, if it was created after 2013-12-04, has a default VPC that has public subnets. You can use this default VPC to enable internet access from your streaming instances. For more information, see Default VPC and Default Subnets in the Amazon VPC User Guide.

**New VPC**

If your AWS account was created before 2013-12-04 or to manage a new VPC, you can create a new VPC with a public subnet using the VPC creation wizard. For more information, see Implementation of VPC with a single public subnet in the Amazon VPC User Guide.
Existing VPC

To use an existing VPC that does not have a public subnet, you can add a new public subnet using the following steps.

To add a new public subnet to an existing VPC

1. Follow the steps in Creating a Subnet in the Amazon VPC User Guide, using the existing VPC you intend to use with AppStream 2.0.
2. To add an internet gateway to your VPC, follow the steps in Attaching an Internet Gateway in the Amazon VPC User Guide.
3. To configure your subnets to route internet traffic through the internet gateway, follow the steps in Creating a Custom Route Table in the Amazon VPC User Guide. Use IPv4 format (0.0.0.0/0) for Destination.

Enabling Internet Access for a Fleet

After you have a public subnet available on a VPC, you can enable internet access for your fleet. This can be performed either when you create the fleet, or by editing the fleet details after creation.

To enable internet access at fleet creation

1. Follow the instructions at Create a Fleet (p. 28) up to the Network access section.
2. Choose Default Internet Access.
3. If the subnet fields are empty, select a subnet for Subnet 1 and, if desired, Subnet 2.
4. Continue with the instructions at Create a Fleet (p. 28).

To enable internet access after fleet creation

1. In the navigation pane, choose Fleets.
2. Select a fleet and check that its state is Stopped.
3. Choose Fleet Details, Edit, Default Internet Access.
4. Choose a subnet for Subnet 1 and, if desired, Subnet 2. Choose Update.

You can test internet connectivity by starting your fleet, creating a stack, associating the fleet to a stack, and browsing the internet within a streaming session for stack. For more information, see Create AppStream 2.0 Stacks and Fleets (p. 27).

Enabling Internet Access for an Image Builder

After you have a public subnet available on a VPC, and can enable internet access for your image builder. This can be performed when you create the image builder.

To enable internet access for an image builder

1. Follow the instructions at Step 1: Create an Image Builder (p. 17) up to the Network Access section.
2. Choose Default Internet Access.
3. If Subnet is empty, select a subnet.
4. Continue with the instructions at Step 1: Create an Image Builder (p. 17).
Enabling Internet Access Using a NAT Gateway

You can control internet access for your users using an advanced networking configuration such as NAT gateways. To manage your own VPC and VPC NAT gateway, launch your AppStream 2.0 image builders and fleets in private VPC subnets that provide a route to the internet. Use the instructions below to quickly create a network setup for enabling internet access. For more information, see NAT Gateways and VPC with Public and Private Subnets (NAT) in the Amazon VPC User Guide.

To create and configure a new VPC to use with a VPC NAT gateway

1. Navigate to Implementing VPC with Public and Private Subnets (NAT) in the Amazon VPC User Guide, and follow the steps given in the section To create a VPC, leaving out the optional IPv6 step.
2. For Availability Zone, leave the public subnet zone as the default, and select a specific zone for the private subnet. Make a note of the zones you chose.
3. For Elastic IP Allocation ID, choose an existing Elastic IP address. If you don't have one, create an Elastic IP address from the Elastic IPs section on the Amazon VPC console.
4. Leave the other fields as their default values, making a note of the value for Private subnet's IPv4 CIDR, and then choose Create VPC. This may take some time to complete.
5. If you want to add another private subnet to your VPC, perform the following steps.
   a. In the left navigation pane, choose Subnets, Create Subnet. Be sure to choose a different name than the ones specified in step 3.
   b. For VPC, enter the VPC that you created earlier. For Availability Zone, enter a different value than the one noted earlier.
   c. For IPv4 CIDR block, provide a unique for the new subnet. For example, if you noted that the first subnet has a IPv4 CIDR block range of 10.0.1.0/24, the new subnet could have a valid CIDR block range of 10.0.2.0/24.
6. Choose Yes, Create.

To add a NAT gateway to an existing VPC

1. Follow the instructions in Creating a NAT Gateway in the Amazon VPC User Guide.
2. To update the route tables of your private subnets and route internet traffic through the NAT gateway, follow the instructions in Updating Your Route Table in the Amazon VPC User Guide.
3. Check your VPC to be sure it has at least one private subnet and, if needed, create a new private subnet. For more information, see Creating a Subnet in the Amazon VPC User Guide.

Enabling Internet Access for a Fleet Using a NAT Gateway

After you have a NAT gateway available on a VPC, you can enable internet access for your fleet. This can be performed either when you create it, or by editing the fleet details after creation.

To enable internet access at fleet creation using a NAT gateway

1. Follow the instructions at Create a Fleet (p. 28) up to the Network access section.
2. Choose a VPC with a NAT gateway.
3. If the subnet fields are empty, select a private subnet for Subnet 1 and, if desired, another private subnet for Subnet 2. If one is not already present for your VPC, you may need to create a second private subnet.
4. Continue with the instructions at Create a Fleet (p. 28).
To enable internet access after fleet creation using a NAT gateway

1. In the navigation pane, choose Fleets.
2. Select a fleet and check that the state is Stopped.
3. Choose Fleet Details, Edit, and choose a VPC with a NAT gateway.
4. Choose a private subnet for Subnet 1 and, if desired, another private Subnet 2. You may need to create a second private subnet if one is not already present for your VPC.
5. Choose Update.

You can test your internet connectivity by starting your fleet, and then connecting to your streaming instance and browsing to the internet.

Enabling Internet Access for an Image Builder Using a NAT Gateway

After you have a NAT gateway available on a VPC, and can enable internet access for your image builder. This can be performed when you create the image builder.

To enable internet access for an image builder using a NAT gateway

1. Follow the instructions at Step 1: Create an Image Builder (p. 17), up to the Network Access section.
2. Choose the VPC with a NAT gateway.
3. If Subnet is empty, select a subnet.
4. Continue with the instructions at Step 1: Create an Image Builder (p. 17).
AppStream 2.0 Image Builders

AppStream 2.0 provides virtual machines, or instances, that are used to install and add applications into and create your image. These instances are called image builders. You can launch an image builder from a base image provided by AWS, or from an image that you create. After your image builder instance is available (running), you can connect to the image builder to start a desktop session, install your applications, add your applications to an image, and create an image.

While launching a new image builder, you can choose from different instance types with various compute, memory, and graphics configurations. Note that the instance type must align with the instance family you need. For more information, see AppStream 2.0 Instance Families (p. 26).

You also provide a VPC subnet so that AppStream 2.0 can establish a network interface to the image builder. This connection provides your image builder with access to resources that might be needed while you install and add applications; for example, file servers, licensing servers, database servers, and so on. For more information, see Tutorial: Create a Custom Image (p. 17).

Actions

The following actions can be performed on an image builder, depending on the current state (status) of the image builder instance.

Delete

Permanently delete an image builder.

The instance must be in a Stopped state.

Connect

Connect to a running image builder. This action starts a desktop streaming session with the image builder to install and add applications to the image, and create an image.

The instance must be in a Running state.

Start

Start a stopped image builder. A running instance is billed to your account.

The instance must be in a Stopped state.

Stop

Stop a running image builder. A stopped instance is not billed to your account.

The instance must be in a Running state.

None of these actions can be performed on an instance in any of the following intermediate states:

- Pending
- Snapshooting
- Stopping
Tutorial: Create a Custom Image

Before you can stream your applications, Amazon AppStream 2.0 requires at least one image that you create by using an image builder. This tutorial describes how to create custom images by using an image builder.

**Important**
After you create an image builder and it is running, your account may incur nominal charges. For more information, see AppStream 2.0 Pricing.

**Important**
This tutorial contains details that apply to the latest base image release. For more information, see Amazon AppStream 2.0 Windows Image Version History (p. 22).

If you are using images that are created from base images dated prior to 2017-07-24, you can view a compatible version of this tutorial by downloading the PDF file appstream2-dg-2017-07-23.pdf.

**Contents**

- Step 1: Create an Image Builder (p. 17)
- Step 2: Install Applications to an Image (p. 19)
- Step 3: Add Applications to an Image (p. 19)
- Step 4: Optimize Applications (p. 20)
- Step 5: Create an Image (p. 20)
- Step 6: Clean Up (p. 21)

**Step 1: Create an Image Builder**

In this step, you create a new image builder so that you can add applications and create images for streaming.

**To create an image builder for adding applications**

2. You can launch the image builder in the following ways:
   - If a welcome screen appears displaying two options (Try it now and Get started), choose Get started, Custom set up.
     For information about these two options, see Amazon AppStream 2.0 FAQs.
   - If a welcome screen does not appear, choose Quick links in the left navigation pane, then Custom set up.
   - Alternatively, choose Images in the left navigation pane, then the Image Builder tab, Launch Image Builder.
3. For **Step 1: Choose Image**, select a base image. If you are launching the image builder for the first time, you can use one of the latest base images released by AWS (selected by default). For a list of the latest versions of base images released by AWS, see Amazon AppStream 2.0 Windows Image Version History (p. 22). If you have already created images, or you want to update applications in an existing image, you can select one of your existing images. Be sure to select an image that
Step 1: Create an Image Builder

aligns with the instance family that you need. For more information, see AppStream 2.0 Instance Families (p. 26).

Choose Next.

4. For **Step 2: Configure Image Builder**, configure the image builder by accepting the default values or providing inputs for the following fields:

   **Name**
   
   Type a unique name identifier for the image builder.

   **Instance Type**
   
   Select the instance type for the image builder. Choose a type that matches the performance requirements of the applications that you plan to install. For more information, see AppStream 2.0 Instance Families (p. 26).

   **Important**
   
   The AppStream 2.0 agent software runs on your streaming instances, enabling your users to connect to and stream their applications. Starting December 7, 2017, your streaming instances can be automatically updated with the latest AppStream 2.0 agent software. This capability helps to ensure that your image builder includes the latest features, performance improvements, and security updates that are available from AWS. You can enable automatic updates of the AppStream 2.0 agent by creating a new image from any base image published by AWS on or after December 7, 2017. If the image from which you are launching your image builder is not using the latest version of the AppStream 2.0 agent, we recommend that you select the option to launch your image builder with the latest agent. This option is not displayed if you are already using the latest base image from AWS or if you are using a custom image that uses the latest version of the agent.

   Choose Next.

5. For **Step 3: Configure Network**, choose a virtual private cloud (VPC) subnet in which to launch your image builder. Your image builder has access to any of the network resources that are accessible from within this VPC subnet.

   For internet access on the image builder, choose Default Internet Access, select a VPC that has public subnets on your default VPC, and then select one of the public subnets listed for Subnet. If you are controlling internet access using a NAT gateway, leave Default Internet Access unselected and use the VPC with the NAT gateway. For more information, see Network Settings for Amazon AppStream 2.0 (p. 8).

   For Security group(s), select up to five security groups to associate with this image builder. If needed, choose Create new security group. If you do not choose a security group, the image builder is associated with the default security group for your VPC. For more information, see Security Groups (p. 11).

   For Active Directory Domain (Optional), expand this section to choose which Active Directory and organizational unit in which to place your streaming instance computer objects. Ensure that the selected network access settings enable DNS resolvability and communication with your directory. For more information, see Using Active Directory Domains with AppStream 2.0 (p. 52).

6. Choose Review and confirm the details for the image builder. To change the configuration for any section, choose Edit and make the needed changes. After you finish reviewing the configuration details, choose Launch.

   After the service prepares the needed resources, the image builder instance list appears. The status of your new image builder appears as Running when the image builder is ready to use.
Optional, you can apply one or more tags to help manage the image builder. Choose Tags, choose Add/Edit Tags, choose Add Tag, specify the key and value for the tag, and then choose Save. For more information, see Tagging Your Amazon AppStream 2.0 Resources (p. 71).

### Step 2: Install Applications to an Image

In this step, you connect to the image builder that you created and launched, then install the applications to be included in the image.

#### To install applications

1. On the left navigation pane, choose Images, Image Builder.
2. Select the image builder to use, check to be sure it has a Running status, and choose Connect.
   
   For this step to work, you may need to configure your browser to allow pop-ups from https://stream.<aws-region>.amazonappstream.com/.
3. Sign in by choosing one of the following options:

   **Administrator**
   
   This mode has full administrator permissions on the image builder instance. Use this mode to install your applications, add applications to the image, and create an image.

   **Test User**
   
   This mode has the same limited permissions as your end users have on their streaming instances. Use this mode to test applications for proper function as an end user.

   **Directory User**
   
   If your image builder is joined to an Active Directory domain, this mode allows you to log in as a user in your domain to access resources that are managed by Active Directory. Provide the username and password of the user to log in as. The user must have local administrator permissions to install applications. For more information, see Providing Local Administrator Permissions for Image Builders (p. 59).

   At any point after logging in, you can switch between users by selecting Switch Users from the Admin Commands menu. This disconnects your current session and brings up the login menu.
4. Install applications by browsing to an application website or other download source. Complete the application's own installation process before moving to the next step. If an application requires the Windows operating system to be restarted, let the operating system restart. Before the operating system restarts, you will be disconnected from your image builder. Wait a few minutes, connect to the image builder again, open Image Assistant, and then finish installing the application.

### Step 3: Add Applications to an Image

In this step, you can add applications (.exe), batch scripts (.bat), and application shortcuts (.lnk) to the image.

#### To add your applications

1. From the image builder desktop, start the Image Assistant application.
2. Choose Add Application and navigate to the location of the application, script, or shortcut to add. Choose Open.
3. In the Application Properties dialog box, enter a display name to be shown to the users in the catalog, change the icon, and enter launch parameters (additional arguments passed to the application when it is launched). Repeat this step for each application that you add to the image.
4. When you finish adding applications, choose Next.

To test your applications

- Verify that the applications you’ve added start correctly. To do this, start a new Windows session as a user who has similar access rights as your end users.
  
  a. From the Admin Commands menu, choose Switch user. This disconnects you from the current session and shows the login menu.
  
  b. To log in as a local test user, choose Test User. To log in as an Active Directory user, choose Directory User, and provide the username and password of the user to log in as. Choose Log in.
  
  c. Launch Image Assistant from the shortcut on the desktop. Choose Launch next to the application to launch, and test your application.
  
  d. Repeat the previous step for each application in the image.
  
  e. To return to the admin mode, choose Switch user, and select the user used to add applications to the image.

Note
Do not exit the Image Assistant application, as you need to use it in the next section.

Step 4: Optimize Applications

In this step, you optimize your applications and create the image. The image builder optimizes your applications for startup performance. This is a required step that is performed on all applications in the list. All applications must be launched before optimization.

To optimize your applications

1. Choose Launch and the service automatically launches the first application in your list. When the application is running, choose Continue.

2. Provide any interactions or inputs that are required by the application to bring it to a usable state. For example, a web browser may prompt you to import settings before it is completely up and running.

3. After you bring the application to a usable state, choose Continue. The application helper launches the next application automatically.

4. Repeat the previous steps until all applications are launched, and leave them running. After you launch all applications, in the Image Assistant application, choose Next.

Step 5: Create an Image

In this step, you choose an image name and create the image.

To create the image

1. Enter a unique image name and image display name (a description is optional), and choose Next. The name you choose cannot begin with "Amazon", "AWS", or "AppStream".

2. Review the image details.

   Note
   If you choose a base image that is published by AWS on or after December 7, 2017, the option Always use the latest agent version appears, and it is selected by default. We recommend that you leave this option selected so that streaming instances that are launched from the image always use the latest version of the agent. If you deselect this option, the application will use the agent version that was installed at the time the image was created.
option, you cannot re-select it after you finish creating the image. For information about the latest release of the AppStream 2.0 agent, see Amazon AppStream 2.0 Agent Version History (p. 24).

Choose Disconnect and Create Image. After your new image is created and the session is disconnected, you can close the session window. Your image builder transitions into a Snapshotting state while the image is being created. After the image is created, your image builder transitions into the Stopped state. You might need to refresh the console listing to see the state change.

3. Return to the console and navigate to Images, Image Registry. Verify that your new image appears in the list.

The new image first appears with a status of Pending in the image registry of your console. After the image is successfully created, the status of the image changes to Available, which means that you can use the image to launch a stack and stream your applications.

Optionally, you can apply one or more tags to help manage the image. Choose Tags, choose Add/Edit Tags, choose Add Tag, specify the key and value for the tag, and then choose Save. For more information, see Tagging Your Amazon AppStream 2.0 Resources (p. 71).

To continue creating images, you can start the image builder and connect to it from the console, or create a new image builder. There is a limit of five image builders per account.

### Step 6: Clean Up

Finally, stop your running image builders to free up resources and avoid unintended charges to your account. We recommend stopping any unused, running image builders. For more information, see AppStream 2.0 Pricing.

#### To stop a running image builder

1. In the navigation pane, choose Images, Image Builders, and select the running image builder instance.
2. Choose Actions, Stop.
AppStream 2.0 Images

An Amazon AppStream 2.0 image contains applications that can be streamed to users. The image is used to launch streaming instances that are part of an AppStream 2.0 fleet. All images available to you are listed under the Image Registry section in the AWS Management Console. Note that the image’s instance family must align with the instance type you need. For more information, see AppStream 2.0 Instance Families (p. 26).

The images in your image registry are differentiated by these visibility attributes:

- **Public Images** — Base images that are made available by AWS to help you create images with your own applications.
- **Private Images** — Images that are created and owned by you.

You can use either public or private images to launch an image builder and set up your AppStream 2.0 fleet. For more information, see Tutorial: Create a Custom Image (p. 17).

You can also delete your private images. Note that a private image cannot be deleted if there are active fleets using it. You must stop all associated fleets before deleting the image.

Amazon AppStream 2.0 Windows Image Version History

AWS publishes base images to help you create images that include your own applications. Base images include the latest Windows operating system and the AppStream 2.0 agent software. For information about the latest AppStream 2.0 software, see Amazon AppStream 2.0 Agent Version History (p. 24).

The following are the latest images:

- **Base** — Base-Image-Builder-1-24-2018
- **Graphics Design** — Graphics-Design-Image-Builder-01-24-2018
- **Graphics Desktop** — Graphics-Desktop-Image-Builder-01-24-2018
- **Graphics Pro** — Graphics-Pro-Image-Builder-01-24-2018
- **Sample apps** — Amazon-AppStream2-Sample-Image-06-20-2017

The latest base image released on January 24, 2018 includes the following software components:

- **Amazon SSM Agent** — 2.2.93.0
- **Amazon WDDM Hook Driver** — 1.0.0.50
- **EC2Config service** — 4.9.2262.0

The following table describes all released images.

<table>
<thead>
<tr>
<th>Release</th>
<th>Image</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release</td>
<td>Image</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>01-01-2018</td>
<td>Base, Graphics Design, Graphics Desktop, Graphics Pro</td>
<td>Includes Microsoft Windows updates for the Spectre and Meltdown vulnerabilities. Enables a default profile to be created on image builders and used for the AWS Command Line Interface (CLI) during streaming sessions. Resolves an issue with connectivity to AppStream 2.0 instances</td>
</tr>
<tr>
<td>11-13-2017</td>
<td>Base</td>
<td>Resolves an issue with Microsoft Office 365 applications not working during streaming sessions Includes Microsoft Windows updates up to October 11, 2017</td>
</tr>
<tr>
<td>09-05-2017</td>
<td>Base, Graphics Design, Graphics Desktop, Graphics Pro</td>
<td>New Graphics Design instance family Support for On-Demand fleets Updated approach for session context Includes Microsoft Windows updates up to August 9, 2017 Resolves an intermittent issue with applications not coming to the foreground Resolves an intermittent issue with applications not appearing in tile view</td>
</tr>
<tr>
<td>07-24-2017</td>
<td>Base</td>
<td>Includes Microsoft Windows updates up to July 13, 2017 Adds support for Microsoft Active Directory domains</td>
</tr>
<tr>
<td>06-20-2017</td>
<td>Base, Sample apps</td>
<td>Optimizes application launch performance Resolves an issue with applications not displaying in tile view Resolves an issue with applications displaying in tile view only Resolves an issue with applications displaying multiple times in tile view Resolves an issue with recently launched application windows not appearing in the foreground Resolves an issue with page margins when printing</td>
</tr>
</tbody>
</table>
Amazon AppStream 2.0 Agent Version History

The Amazon AppStream 2.0 agent software runs on your streaming instances, enabling end users to connect to and start their streaming applications. Starting December 7, 2017, your streaming instances can be automatically updated with the latest features, performance improvements, and security updates that are available from AWS. Before December 7, 2017, agent updates were included with new base image releases.

To use the latest AppStream 2.0 agent software, you need to rebuild your images by using new base images published by AWS on or after December 7, 2017. When you do this, the option to enable automatic updates of the agent is selected by default in the Image Assistant. We recommend that you leave this option selected so that any new image builder or fleet instance that is launched from your image always uses the latest version of the agent. For more information, see Tutorial: Create a Custom Image (p. 17).

The following table describes the latest updates that are available in released versions of the AppStream 2.0 agent.

<table>
<thead>
<tr>
<th>Amazon AppStream 2.0 agent version</th>
<th>Description</th>
</tr>
</thead>
</table>
| 01-24-2018                         | • Resolves an issue with the Alt Graph key not working on certain keyboard layouts.  
  • Works with these software components:  
    • Amazon SSM Agent — 2.2.93.0  
    • Amazon WDDM Hook Driver — 1.0.0.50  
    • EC2Config service — 4.9.2262.0 |
| 12-07-2017                         | • Resolves issues with using ALT key combinations  
  • Resolves an issue with file uploads from local computers to streaming sessions  
  • Works with these software components:  
    • Amazon SSM Agent — 2.2.93.0  
    • Amazon WDDM Hook Driver — 1.0.0.21  
    • EC2Config service — 4.9.2218.0 |
Amazon AppStream 2.0 Stacks and Fleets

With Amazon AppStream 2.0, you create stacks and fleets as part of the process of streaming applications. A fleet consists of streaming instances that run the image that you specify. A stack consists of a fleet, user access policies, and storage configurations.

Contents

- Fleet Type (p. 25)
- Session Context (p. 25)
- AppStream 2.0 Instance Families (p. 26)
- Create AppStream 2.0 Stacks and Fleets (p. 27)
- Fleet Auto Scaling for Amazon AppStream 2.0 (p. 30)

Fleet Type

The fleet type determines when your instances run and how you pay for them. You can specify a fleet type when you create a fleet. You cannot change the fleet type after you create the fleet.

The following are the possible fleet types:

- Always-On
  Instances run all the time, even when no users are streaming applications.

- On-Demand
  Instances run only when users are streaming applications. Idle instances that are available for streaming are in a stopped state.

Use an Always-On fleet to provide your users with instant access to their applications. Use an On-Demand fleet to optimize your streaming charges and provide your users with access to their applications after a 1-2 minute wait. For more information, see Amazon AppStream 2.0 Pricing.

To create an On-Demand fleet, you must use a base image starting with 09-05-2017.

Session Context

You can pass parameters to your streaming application using session context. The format is a string with parameters separated by commas. Session context is supported using the AWS CLI and the AWS SDKs, but is not supported using the AWS Management Console.

Starting with the images released on 09-05-2017, the parameters are passed using the AppStream_Session_Context environment variable. This environment variable is accessible only through .NET, and we provide an executable file, SessionContextRetriever.exe, that you can use to access it. With images released prior to 09-05-2017, parameters are passed to the application.
The following example uses session context to launch a specific website using Google Chrome.

**To use session context to launch a website**

1. Connect to your image builder in Administrator mode. For this example, install Google Chrome on the image builder.
2. Create a child folder of C:\. For this example, use C:\Scripts.
3. For images released on or after 09-05-2017, download SessionContextRetriever.exe.
4. Create a Windows batch file in the new folder. For this example, create C:\Scripts\session-context-test.bat and add a script that launches Chrome with the URL from session context, and then waits for keyboard input.

For images released on or after 09-05-2017, use the following script:

```bash
for /f "tokens=* USEBACKQ" %%f in (`SessionContextRetriever.exe`) do (
    set var=%%f
    chrome.exe %var%
    pause
)
```

For images released prior to 09-05-2017, use the following scripts:

```bash
chrome.exe %1
pause
```

5. In Image Assistant, add session-context-test.bat and change the working directory to C:\Program Files (x86)\Google\Chrome\Application.
6. Create an image, fleet, and stack. For this example, use a fleet name of session-context-test-fleet and a stack name of session-context-test-stack.
7. After the fleet is running, you can call create-streaming-url with the session-context parameter, as shown in this example.

```bash
aws appstream create-streaming-url --stack-name session-context-test-stack \
    --fleet-name session-context-test-fleet \ 
    --user-id username -validity 10000 \ 
    --application-id chrome --session-context "www.google.com"
```

**AppStream 2.0 Instance Families**

Amazon AppStream 2.0 users stream applications from stacks created by an administrator. Each stack is associated with a fleet. When you create a fleet, the instance type that you specify determines the hardware of the host computers used for your fleet. Each instance type offers different compute, memory, and GPU capabilities. Instance types are grouped into instance families based on these capabilities.

When you create a fleet or image builder, you must select an image that is compatible with the instance family on which you intend to run your fleet.

- When launching a new image builder, you are presented with a list of the images in your image registry. Select the appropriate base image.
- When launching a fleet, ensure that the private image you select was created from the appropriate base image.
The following table summarizes the available instance families and provides the base image naming format for each. Select an instance type from an instance family based on the requirements of the applications that you plan to stream on your fleet, and match the base image according to the following table.

<table>
<thead>
<tr>
<th>Instance Family</th>
<th>Description</th>
<th>Base Image Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Purpose</td>
<td>Basic computing resources for running web browsers and most business applications.</td>
<td>Base-Image-Builder-MM-DD-YYYY</td>
</tr>
<tr>
<td>Memory Optimized</td>
<td>Optimized for memory-intensive applications that process large amounts of data.</td>
<td>Base-Image-Builder-MM-DD-YYYY</td>
</tr>
<tr>
<td>Compute Optimized</td>
<td>Optimized for compute-bound applications that benefit from high performance processors.</td>
<td>Base-Image-Builder-MM-DD-YYYY</td>
</tr>
<tr>
<td>Graphics Design</td>
<td>Uses AMD FirePro S7150x2 Server GPUs and AMD Multiuser GPU technology to support graphics applications that use DirectX, OpenGL, or OpenCL.</td>
<td>Graphics-Design-Image-Builder-MM-DD-YYYY</td>
</tr>
<tr>
<td>Graphics Desktop</td>
<td>Uses NVIDIA GRID K520 GPU to support applications that benefit from or require graphics acceleration. This instance family supports DirectX, OpenGL, OpenCL, and CUDA.</td>
<td>Graphics-Desktop-Image-Builder-MM-DD-YYYY</td>
</tr>
<tr>
<td>Graphics Pro</td>
<td>Uses NVIDIA Tesla M60 GPUs and provide a high-performance, workstation-like experience for graphics applications that use DirectX, OpenGL, OpenCL, or CUDA.</td>
<td>Graphics-Pro-Image-Builder-MM-DD-YYYY</td>
</tr>
</tbody>
</table>

For more information, see the following:

- Amazon AppStream 2.0 Windows Image Version History (p. 22)
- Amazon AppStream 2.0 Service Limits (p. 84)
- AppStream 2.0 Pricing

Create AppStream 2.0 Stacks and Fleets

To stream your applications, Amazon AppStream 2.0 requires an environment consisting of a stack, an associated fleet and at least one application image. This tutorial walks through the steps to set up a stack and a fleet, and how to give users access to the stack. If you haven’t already done so, we recommend that you try the procedures in Getting Started with Amazon AppStream 2.0 (p. 5) first.

If you want to create an image to use, see Tutorial: Create a Custom Image (p. 17).

If you intend to join a fleet to an Active Directory domain, configure your Active Directory before following the steps below. For more information, see Using Active Directory Domains with AppStream 2.0 (p. 52).

Tasks
Create a Fleet

Set up and create a fleet from which user applications are executed and streamed.

To set up and create a fleet

2. Choose Get Started if you are new to the console, or Fleets from the left navigation pane. Choose Create Fleet.
3. For Step 1: Provide Fleet Details, provide a fleet name, optional display name, and optional description. Choose Next.
4. For Step 2: Choose an Image, choose an image that meets your needs and then choose Next.
5. For Step 3: Configure Fleet, do the following:
   a. For Choose instance type, choose the instance type that meets the performance requirements of your applications.
   b. For Fleet type, choose the fleet type that suits your use case. The fleet type determines its immediate availability and how you pay for it.
   c. For Maximum session duration — Choose the maximum amount of time that a streaming session can remain active. If users are still connected to a streaming session five minutes before this limit is reached, they are prompted to save any open documents before being disconnected.
   d. For Disconnect timeout, choose the time that a streaming instance should remain 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 the previous session. Otherwise, they are connected to a new session with a new instance. If you associate a stack with a fleet for which a redirect URL is specified, after users' streaming sessions end, the users are redirected to that URL.
      
      If a user ends the session by choosing End Session on the streaming session toolbar, the disconnect timeout does not apply. Instead, the user is prompted to save any open documents, and then immediately disconnected from the streaming instance.
   e. For Minimum capacity, choose a minimum number of instances for your fleet based on the minimum number of expected concurrent users.
   f. For Maximum capacity, choose a maximum number of instances for your fleet based on the maximum number of expected concurrent users.
   g. For Scaling details, specify the scaling policies that AppStream 2.0 uses to increase and decrease the capacity of your fleet. Note that the size of your fleet is limited by the minimum and maximum capacity that you specified. For more information, see Fleet Auto Scaling for Amazon AppStream 2.0 (p. 30).
6. For Step 4: Configure Network, do the following:
   a. To add internet access for fleet instances in a VPC with a public subnet, choose Default Internet Access. If you are providing internet access using a NAT gateway, leave Default Internet Access unselected. For more information, see Network Settings for Amazon AppStream 2.0 (p. 8).
   b. Choose a VPC and two subnets with access to the network resources that your application needs. If you don't have a VPC or subnets, you can create them using the links provided and then click the refresh icons.
To set up and create a stack

1. In the left navigation pane, choose Stacks, and then choose Create Stack.
2. For Step 1: Stack Details, provide a stack name. Optionally, you can also provide a display name, description, and redirect URL. If you specify a redirect URL, after users’ streaming sessions end, the users are redirected to that URL. For Fleet, choose the fleet to associate with your stack. If you need to create a fleet, use the link provided and then click the refresh icon. Choose Next.
3. For Step 2: Enable Storage, you can enable persistent storage for the stack users by selecting Enable Home Folders. Choose Review.
4. Choose Create.

The status of your new stack is Active when it is available to work with from the console.

Optionally, you can apply one or more tags to help manage the stack. Choose Tags, choose Add/Edit Tags, choose Add Tag, specify the key and value for the tag, and then choose Save. For more information, see Tagging Your Amazon AppStream 2.0 Resources (p. 71).

Provide Access to Users

After you create a stack with an associated fleet, you can provide access to users through the AppStream 2.0 user pool. For more information, see User Pool Administration (p. 42).

Note that user pool users cannot be assigned to stacks with fleets that are joined to an Active Directory domain.

Clean Up Resources

You can stop your running fleet and delete your active stack to free up resources and to avoid unintended charges to your account. We recommend stopping any unused, running fleets.

Note that you cannot delete a stack with an associated fleet.

To clean up your resources

1. In the navigation pane, choose Stacks.
2. Select the stack and choose Actions, Disassociate Fleet.
3. From Stack Details, open the Associated Fleet link to select the fleet.
4. Choose Actions, Stop. It takes about 5 minutes to stop a fleet.
5. When the status of the fleet is Stopped, choose Actions, Delete.
6. In the navigation pane, choose Stacks.
7. Select the stack and choose Actions, Delete.

Fleet Auto Scaling for Amazon AppStream 2.0

Fleet Auto Scaling allows you to automatically change the size of your AppStream 2.0 fleet to match the supply of available instances to user demand. Because each instance in a fleet can be used by only one user at a time, the size of your fleet determines the number of users who can stream concurrently. You can define scaling policies that adjust the size of your fleet automatically based on a variety of utilization metrics, and optimize the number of available instances to match user demand. You can also choose to turn off automatic scaling and make the fleet run at a fixed size.

AppStream 2.0 scaling is provided by Application Auto Scaling. For more information, see the Application Auto Scaling API Reference.

Before you can use Fleet Auto Scaling, Application Auto Scaling needs permissions to access Amazon CloudWatch alarms and AppStream 2.0 fleets. For more information, see Application Auto Scaling IAM Role (p. 67) and Application Auto Scaling Required IAM Permissions (p. 68).

For a walk-through of AppStream 2.0 scaling, see Scaling Your Desktop Application Streams with Amazon AppStream 2.0 in the AWS Compute Blog.

Scaling Concepts

To use Application Auto Scaling effectively, there are a few terms and concepts that you should be familiar with and understand.

Minimum Capacity

The minimum size of the fleet. Scaling policies do not scale your fleet below this value. For example, if you specify 2, your fleet will never have less than 2 instances available. Note that if Desired Capacity (set by editing Fleet Details and not Scaling Policies) is set below the value of Minimum Capacity and a scale-out activity is triggered, Application Auto Scaling scales the Desired Capacity value up to the value of Minimum Capacity and then continues to scale out as required, based on the scaling policy. However, in this example, a scale-in activity does not adjust Desired Capacity, because it is already below the Minimum Capacity value.

Maximum Capacity

The maximum size of the fleet. Scaling policies do not scale your fleet above this value. For example, if you specify 10, your fleet will never have more than 10 instances available. Note that if Desired Capacity (set by editing Fleet Details and not Scaling Policies) is set above the value of Maximum Capacity and a scale-in activity is triggered, Application Auto Scaling scales Desired Capacity down to the value of Maximum Capacity and then continues to scale in as required, based on the scaling policy. However, in this example, a scale-out activity does not adjust Desired Capacity, because it is already above the Maximum Capacity value.

Scaling Policy Action

The action that scaling policies perform on your fleet when the Scaling Policy Condition is met. You can choose an action based on % capacity or number of instance(s). For example, if Desired
**Capacity** is 4 and **Scaling Policy Action** is set to "Add 25% capacity", **Desired Capacity** is increased by 25% to 5 when **Scaling Policy Condition** is met.

**Scaling Policy Condition**

The condition that triggers the action set in **Scaling Policy Action**. This condition includes a scaling policy metric, a comparison operator, and a threshold. For example, to scale a fleet if the utilization of the fleet is greater than 50%, your scaling policy condition should be "If Capacity Utilization > 50%".

**Scaling Policy Metric**

This is the metric on which your scaling policy is based. The following metrics are available for scaling policies:

- **Capacity Utilization**
  
  Percentage of instances in a fleet that are being used. You can use this metric to scale your fleet based on usage of the fleet. For example, **Scaling Policy Condition**: "If Capacity Utilization < 25%" perform **Scaling Policy Action**: "Remove 25 % capacity".

- **Available Capacity**
  
  Number of instances in your fleet that are available for user sessions. You can use this metric to maintain a buffer in your capacity available for users to start streaming sessions. For example, **Scaling Policy Condition**: "If Available Capacity < 5" perform **Scaling Policy Action**: "Add 5 instance(s)".

- **Insufficient Capacity Error**
  
  Number of session requests rejected due to lack of capacity. You can use this metric to provision new instances for users that are unable to get sessions because of lack of capacity. For example, **Scaling Policy Condition**: "If Insufficient Capacity Error > 0" perform **Scaling Policy Action**: "Add 1 instance(s)".

### Managing Fleet Scaling Using the Console

You can set up and manage fleet scaling using the AWS Management Console in two ways: during fleet creation, or anytime using the **Fleets** tab. Two default scaling policies are associated with newly created fleets after launch and can be edited via the console from the **Scaling Policies** tab. For more information, see Create a Fleet (p. 28).

For user environments that vary in number, define scaling policies to control how scaling responds to demand. If you expect a fixed number of users or have other reasons for disabling scaling, you can set your fleet with a fixed number of instances.

**To set a fleet scaling policy using the console**

2. In the navigation pane, choose **Fleets**.
3. Select the fleet and then choose **Scaling Policies**.
4. Edit existing policies by choosing the edit icon next to each value. Set the desired values in the edit field and choose **Update**. The policy changes go into effect within a few minutes.
5. Add (create) new policies using the **Add Policy** link. Set the desired values in the edit field and choose **Create**. The new policy goes into effect within a few minutes.

You can use the **Fleet Usage** tab to monitor the effects of your scaling policy changes. The following is an example usage graph of scaling activity when five users connect to the fleet and then disconnect. This example is from a fleet using the following scaling policy values:
• Minimum Capacity = 1
• Maximum Capacity = 5
• Scale Out = Add 2 instances if Capacity Utilization > 75%
• Scale In = Remove 1 instance if Capacity Utilization < 25%

To set a fixed capacity fleet using the console

2. In the navigation pane, choose Fleets.
3. Select the fleet.
4. For Scaling Policies, remove all policies associated with the fleet.
5. For Fleet Details, edit the fleet to set Desired Capacity.

The fixed fleet has constant capacity based on the value that you specified as Desired Capacity. Note that a fixed fleet has the desired number of instances available at all times and the fleet must be stopped to stop billing costs for that fleet.
Managing Fleet Scaling Using the AWS CLI

You can set up and manage fleet scaling using the AWS Command Line Interface.

CLI Examples

Before running scaling policy commands, you must register your fleet as a scalable target. Use the following `register-scalable-target` command:

```
aws application-autoscaling register-scalable-target
    --service-namespace appstream \
    --resource-id fleet/fleetname \
    --scalable-dimension appstream:fleet:DesiredCapacity \
    --min-capacity 1 --max-capacity 5 \
    --role-arn arn:aws:iam::account-number-without-hyphens:role/service-role/ApplicationAutoScalingForAmazonAppStreamAccess
```

Example 1: Applying a Scaling Policy Based on Capacity Utilization

This CLI example sets up a scaling policy that scales out a fleet by 25% if Utilization >= 75%.

The following `put-scaling-policy` command defines a utilization-based scaling policy:

```
aws application-autoscaling put-scaling-policy --cli-input-json file://scale-out-utilization.json
```

The contents of the file `scale-out-utilization.json` are as follows:

```
{
    "PolicyName": "policyname",
    "ServiceNamespace": "appstream",
    "ResourceId": "fleet/fleetname",
    "ScalableDimension": "appstream:fleet:DesiredCapacity",
    "PolicyType": "StepScaling",
    "StepScalingPolicyConfiguration": {
        "AdjustmentType": "PercentChangeInCapacity",
        "StepAdjustments": [
            {
                "MetricIntervalLowerBound": 0,
                "ScalingAdjustment": 25
            }
        ],
        "Cooldown": 1500
    }
}
```

If the command is successful, the output looks something like the following, although some details are unique to your account and region. In this example, the policy identifier is e3425d21-16f0-d701-89fb-12f98dac64af.

```
```

Now, set up a CloudWatch alarm for this policy. Use the names, region, account number, and policy identifier from your information. You can use the policy ARN returned by the previous command for the `--alarm-actions` parameter.
Example 2: Applying a Scaling Policy Based on Insufficient Capacity Errors

This CLI example sets up a scaling policy that scales out the fleet by 1 if the fleet throws an InsufficientCapacityError error.

The following command defines an insufficient capacity-based scaling policy:

```
aws application-autoscaling put-scaling-policy --cli-input-json file://scale-out-capacity.json
```

The contents of the file `scale-out-capacity.json` are as follows:

```
{
    "PolicyName": "policyname",
    "ServiceNamespace": "appstream",
    "ResourceId": "fleet/fleetname",
    "ScalableDimension": "appstream:fleet:DesiredCapacity",
    "PolicyType": "StepScaling",
    "StepScalingPolicyConfiguration": {
        "AdjustmentType": "ChangeInCapacity",
        "StepAdjustments": [
            { "MetricIntervalLowerBound": 0, "ScalingAdjustment": 1 },
        ],
        "Cooldown": 1500
    }
}
```

If the command is successful, the output looks something like the following, although some details are unique to your account and region. In this example, the policy identifier is f4495f21-0650-470c-88e6-0f393adb64fc.

```
{"PolicyARN": "arn:aws:autoscaling:us-west-2:123456789012:scalingPolicy:f4495f21-0650-470c-88e6-0f393adb64fc:resource/appstream/fleet/SampleFleetName:policyName/SamplePolicyName"}
```

Now, set up a CloudWatch alarm for this policy. Use the names, region, account number, and policy identifier from your information. You can use the policy ARN returned by the previous command for the `--alarm-actions` parameter.

```
aws cloudwatch put-metric-alarm
--alarm-name alarmname
--alarm-description "Alarm when out of capacity is > 0"
--metric-name CapacityUtilization
--namespace AWS/AppStream
--statistic Average
--period 300
--threshold 75
--comparison-operator GreaterThanThreshold
--dimensions "Name=FleetName,Value=fleetname"
--evaluation-periods 1 --unit Percent
```
```bash
--metric-name InsufficientCapacityError \
--namespace AWS/AppStream \
--statistic Maximum \
--period 300 \
--threshold 0 \ 
--comparison-operator GreaterThanThreshold \
--dimensions "Name=FleetName,Value=fleetname" \ 
--evaluation-periods 1 --unit Count \
```
Persistent Storage with AppStream 2.0 Home Folders

AppStream 2.0 supports persistent storage for your end users with Home Folders. When this option is enabled for an AppStream 2.0 stack, end users of the stack are presented with a persistent storage folder in their AppStream 2.0 sessions. No further configuration is required on your users' part to access their Home Folder. Data stored by users in their Home Folder is automatically backed up to an Amazon S3 bucket in your AWS account and is made available in subsequent sessions for those users.

Home Folder End User Experience

For end users, the Home Folder behaves like a normal Windows folder. Users experience the following Home Folder access features when they are working in an application during a streaming session.

- Users can save their documents and project files to their Home Folder. AppStream 2.0 continuously checks for the most recently modified files and backs them up to each user's folder in persistent storage.
- Data content in each user's Home Folder is specific to that user and cannot be accessed by other users.
- Users can easily access their Home Folder when they are working in an application by choosing File Open or File Save from the application interface.
- A web view is available to access the Home Folder by choosing My Files from the web view session toolbar.

Using the web view, users can navigate to Home Folder files, create new folders within their Home Folder, upload files from their local computer to their Home Folder, download files from their Home Folder to their local computer, and rename any of the existing files or folders within their Home Folder.

To upload and download files between your local computer and your Home Folder

1. In the AppStream 2.0 web view session, choose the My Files icon at the top left of your browser.
2. Navigate to an existing folder, or choose Add Folder to create a new folder.
3. When the desired folder is displayed, choose Upload to upload a local file to the currently displayed location in the session's Home Folder. To download a specific file, choose the down arrow on the right of the file name and choose Download.
Home Folder Administration

AppStream 2.0 administrators can enable or disable Home Folders for a stack by using the AWS Management Console for AppStream 2.0, AWS SDK, or AWS CLI. For each region, Home Folders are backed up by an S3 bucket.

Tasks
- Before Enabling Home Folders (p. 37)
- Enabling Home Folders (p. 38)
- Disabling Home Folders (p. 38)
- Amazon S3 Bucket Storage (p. 39)
- Home Folder Formats (p. 39)
- Using the AWS Command Line Interface or AWS SDKs (p. 40)

Before Enabling Home Folders

Refer to the following guidelines before enabling Home Folders for a stack:

- Check that you have the correct IAM permissions for Amazon S3 actions. For more information, see IAM Policies and the Amazon S3 Bucket for Home Folders (p. 69).
- Use an image that was created from an AWS base image released on or after May 18, 2017. For a current list of released AWS images, see Amazon AppStream 2.0 Windows Image Version History (p. 22).
- Enable network connectivity to Amazon S3 from your VPC by configuring internet access or a VPC endpoint for Amazon S3. For more information, see Network Settings for Amazon AppStream 2.0 (p. 8) and Home Folders and VPC Endpoints (p. 12).
Enabling Home Folders

An AppStream 2.0 administrator can enable or disable Home Folders while creating a stack (see Create a Stack (p. 29)), or after the stack is created.

The first time you enable Home Folders for an AppStream 2.0 stack in an AWS region, the service creates an S3 bucket in your account in that same region. The same bucket is used to store the content of Home Folders for all users and all stacks in that region. For more information, see Amazon S3 Bucket Storage (p. 39).

To enable Home Folders while creating a stack

- Follow the instructions at Create a Stack (p. 29), and ensure that Enable Home Folders is selected.

To enable Home Folders for an existing stack

2. On the left navigation pane, choose Stacks, and select the stack for which to enable Home Folders.
3. Below the stack list, choose Storage and select Enable Home Folders.
4. Confirm the enable action in the resulting dialog box by choosing Enable.

Disabling Home Folders

You can disable Home Folders for a stack without losing user content already stored in Home Folders. Disabling Home Folders for a stack has the following effects:

- If the stack has sessions that are in use, an error message is shown to any users currently using the session, and these users can no longer store content to Home Folders in the session.
- Any new sessions using the stack with Home Folders disabled do not present Home Folders.
- Other stacks that have Home Folders enabled are not affected by this stack's disable operation, and continue to provide users access to their data within those stacks' streaming sessions. Only the specific stack for which Home Folders is disabled is affected.
- Even if Home Folders are disabled for all stacks, AppStream 2.0 does not delete the user content.

To restore access to Home Folders for the stack, enable Home Folders again by following the steps described earlier in this topic.

To disable Home Folders while creating a stack

- Follow the instructions at Create a Stack (p. 29) and ensure that Enable Home Folders is cleared.

To disable Home Folders for an existing stack

2. On the left navigation pane, choose Stacks, and select the stack.
3. Below the stack list, choose Storage and clear Enable Home Folders.
4. Confirm the disable action in the resulting dialog box by typing CONFIRM (case-sensitive) and choosing Disable.
Amazon S3 Bucket Storage

AppStream 2.0 manages user content stored in Home Folders by using S3 buckets created in your account. For every region, AppStream 2.0 creates a bucket in your account and stores all user content generated from streaming sessions of stacks in that region in that bucket. The buckets are fully managed by the service without any admin inputs or configuration. The buckets are named in a specific format as follows:

\[
\text{appstream2-36fb080bb8-\textit{region-code}\textit{-account-id-without-hyphens}}
\]

Where \textit{region-code} is the AWS region code in which the stack is created and \textit{account-id-without-hyphens} is your AWS account ID. The first part of the bucket name, appstream2-36fb080bb8-, does not change across accounts or regions.

For example, if you enable Home Folders for stacks in region us-west-2 on account number 123456789012, the service creates an S3 bucket in the us-west-2 region with the name shown. This bucket name cannot change or be deleted without manual modification by an administrator.

\[
\text{appstream2-36fb080bb8-us-west-2-123456789012}
\]

As mentioned, disabling Home Folders for stacks does not delete any user content stored in the S3 bucket. To permanently delete user content, an administrator with adequate access must do so from the Amazon S3 console. AppStream 2.0 adds a bucket policy that prevents accidental deletion of the bucket. For more information, see IAM Policies and the Amazon S3 Bucket for Home Folders (p. 69).

Additional Resources

To learn more about managing S3 buckets and best practices, see the following topics in the Amazon Simple Storage Service Developer Guide:

- You can provide offline access to user data for your users with Amazon S3 policies. For more information, see Allow Users to Access a Personal "Home Directory" in Amazon S3.
- You can enable file versioning for content stored in S3 buckets used by AppStream 2.0. For more information, see Using Versioning.

Home Folder Formats

When Home Folders are enabled, users are provided with a unique folder in which to store their content (one folder per user). The folder is created and maintained as a unique Amazon S3 object within the bucket for that region. The hierarchy of a user folder depends on how the user launches a streaming session.

AWS SDKs and AWS CLI

For sessions created using CreateStreamingURL or create-streaming-url the user folder structure is as follows:

\[
\text{bucket-name/user/custom/user-id-SHA-256-hash/}
\]

Where \textit{bucket-name} is in the format shown in Amazon S3 Bucket Storage (p. 39) and \textit{user-id-SHA-256-hash} is the user-specific folder name created using a lower case SHA-256 hash hexadecimal string generated from the \textit{UserId} value passed to the CreateStreamingURL API operation or create-
streaming-url command. For more information, see CreateStreamingURL in the Amazon AppStream 2.0 API Reference and create-streaming-url in the AWS CLI Command Reference.

The following example folder structure applies to session access using the API or CLI with a UserId testuser@mydomain.com, account id 123456789012 in region us-west-2:

```
appstream2-36fb080bb8-us-west-2-123456789012/user/custom/
a0bc1da11f480d9b5b3e90f91243143ec04ccfd8d77e740f64828a0cd13/
```

Administrators can identify the folder for a user by generating the lower case SHA-256 hash value of the UserId using websites or open source coding libraries available online.

**SAML**

For sessions created using SAML federation, the user folder structure is as follows:

```
bucket-name/user/federated/user-id-SHA-256-hash/
```

In this case, `user-id-SHA-256-hash` is the folder name created using a lower case SHA-256 hash hexadecimal string generated from the `NameID` SAML attribute value passed in the SAML federation request. To differentiate users with the same name belonging to two different domains, send the SAML request with `NameID` in the format `domainname\username`. For more information, see Single Sign-on Access to AppStream 2.0 Using SAML 2.0 (p. 45).

The following example folder structure applies to session access using SAML federation with a `NameID` SAMPLEDOMAIN\testuser, account ID 123456789012 in region us-west-2:

```
appstream2-36fb080bb8-us-west-2-123456789012/user/
federated/8dd9a642f511609454d53cb861a71190e44fed2b8af9fde0c507012a9901
```

Administrators can identify the folder for a user by generating the lower case SHA-256 hash value of the `NameID` using websites or open source coding libraries available online.

**Using the AWS Command Line Interface or AWS SDKs**

You can enable and disable Home Folders for a stack using the AWS CLI or AWS SDKs.

Use the following `create-stack` command enable Home Folders while creating a new stack:

```
aws appstream create-stack --name ExampleStack --storage-connectors type=HOMEFOLDERS
```

Use the following `update-stack` command to enable Home Folders for an existing stack:

```
aws appstream update-stack --name ExistingStack --storage-connectors type=HOMEFOLDERS
```

Use the following command to disable Home Folders for an existing stack. This command does not delete any user data.

```
aws appstream update-stack --name ExistingStack --delete-storage-connectors
```
Manage Access Using the AppStream 2.0 User Pool

The AppStream 2.0 user pool offers a simplified way to manage access to applications for your end users through a persistent portal for each region. This feature is offered as a built-in alternative to user management through Active Directory and SAML 2.0 federation. To use external identity providers for user management, see Single Sign-on Access to AppStream 2.0 Using SAML 2.0 (p. 45). To join your Active Directory domain to AppStream 2.0, see Using Active Directory Domains with AppStream 2.0 (p. 52).

Note
User pool users cannot be assigned to stacks with fleets that are joined to an Active Directory domain.

The AppStream 2.0 user pool offers the following key features:

- Users can access application stacks through a persistent URL and login credentials using their email address and a password that they choose.
- Administrators can assign a user multiple stacks, offering multiple application catalogs to the user when they log in.
- When an administrator creates a new user, a welcome email is automatically sent to the end user with a login portal link and instructions.
- After being created, a user in the pool remains valid and usable unless an administrator specifically disables that user.
- Administrators can control which users have access to which application stacks, or disable access completely.

User Pool End User Experience

With the user pool, the following flow of actions summarizes the initial connection experience for the end user.

1. An administrator creates a new user in the desired region using the end user’s email address.
2. AppStream 2.0 sends a welcome email with instructions and a temporary password.
3. An administrator assigns the user one or more stacks.
4. AppStream 2.0 sends an optional notification email to the end user with information and instructions for the stacks to which the user is newly assigned.
5. Using the information in the welcome email, the end user connects to the login portal and uses their temporary password to set a permanent password. The login portal link never expires and can be used anytime.
6. Using the email address and permanent password they set up earlier, the end user signs in and is presented with their application catalogs.

The login portal link provided in the welcome email should be saved for future use, as it does not change and is valid for all user pool users. Note that the login portal URL and user pool user are managed on a per-region basis.
Resetting a Forgotten Password

If a user forgets their password, they can connect to the login portal link (provided in the welcome email) to choose a new password.

**To choose a new password**

1. Open the AppStream 2.0 login portal using the login link provided in the welcome email.
2. Choose **Forgot Password?**.
3. Type the email address used to create your user pool user. Choose **Next**.
4. Check your email for the password reset request message. If you are having difficulty finding the email, check your spam email folder. Type the verification code from the email in **Verification Code**.
   
   **Note**
   
   The verification code is valid for 24 hours. If a new password is not chosen within this time, request a new verification code.
5. Following the password rules shown, type and confirm your new password. Choose **Reset Password**.

User Pool Administration

To perform administrator actions, sign in to the AppStream 2.0 console in the AWS Management Console for the desired region and select User Pool in the left navigation pane. The User Pool dashboard supports bulk operations on a list of users for some actions. An administrator can select multiple users on which to perform the same action from the **Actions** list. Bulk user creation or disable is not supported. User pool users are created and managed on a per-region basis.

**Note**

AppStream 2.0 sends email to users on your behalf, such as when a new user is created or a user is assigned to a stack. To ensure that email is delivered, add no-reply@accounts.aws-region-code.amazonaws.com to your whitelist, where *aws-region-code* is a valid AWS region code in which you are working. If users are having difficulty finding the emails, ask them to check their " spam" email folder.

**Tasks**

- Creating a User (p. 42)
- Assigning Stacks to Users (p. 43)
- Unassigning Stacks from Users (p. 43)
- Disabling Users (p. 44)
- Enabling Users (p. 44)
- Re-Sending Welcome Email (p. 44)

Creating a User

Users are managed on a per-region basis. You must use a valid and unique email address for each new user within a region. However, you can reuse an email address for a new user in another region.

When you create a new user, be aware of the following:

- There is no limit on the number of users in the user pool.
- You can enable or disable a user, but you cannot delete a user.
• You cannot change the email address, first name, or last name for a user that you have already created. To change this information for a user, disable the user. Then, recreate the user (as a new user) and specify the updated information as needed.
• You can assign one or more stacks to the user after the user is created.

To create a new user
2. On the left navigation pane, choose User Pool, Create User.
3. For Email, type the unique email address for this user.
4. For First name and Last name, type values. These fields need not be unique.
5. Choose Create User.

After the user is created, AppStream 2.0 sends a welcome email to the user. This email has the login portal link, the login email address to be used, and a temporary password. By browsing to the login portal and using the temporary password, the user can set a permanent password to access their applications.

By default, the new user’s status is Enabled, meaning you can assign one or more stacks to the user, or perform other actions.

Assigning Stacks to Users

An AppStream 2.0 administrator can assign one or more stacks to one or more user pool users. After being assigned at least one stack, the user can log in and launch applications. If users are assigned more than one stack, they are presented with a list of stacks as catalogs to choose from before launching applications. User Pool users cannot be assigned to stacks with fleets that are joined to an Active Directory domain.

To assign a stack to users
2. On the left navigation pane, choose User Pool and select the users.
3. Choose Actions, Assign stack. Note that users cannot be assigned to stacks that have a fleet joined to an Active Directory domain. For more information, see Using Active Directory Domains with AppStream 2.0 (p. 52).
4. Confirm the list of users in the resulting dialog box. For Stack, choose the desired stack.
5. By default, Send email notification to user is enabled. Clear this option if you do not want to send the notification email to the user at this time.
6. Choose Assign stack.

Unassigning Stacks from Users

An AppStream 2.0 administrator can unassign stacks from one or more user pool users. After being unassigned a stack, the user can no longer launch applications from that stack.

To unassign a stack from users
2. On the left navigation pane, choose User Pool and select the users.
3. Choose Actions, Unassign stack.
4. Confirm the list of users in the resulting dialog box. For **Stack**, choose the desired stack. This list includes all stacks, assigned or unassigned.

5. Choose **Unassign stack**.

### Disabling Users

An AppStream 2.0 administrator can disable one or more user pool users, one at a time. After being disabled, the user can no longer log in until they are re-enabled. This action does not delete the user. If the user is currently connected when an administrator disables them, their session remains active until the session cookie expires (about one hour). Stack assignments for the user are retained. If the user is re-enabled, the stack assignment becomes active again.

**To disable a user**

2. On the left navigation pane, choose **User Pool** and select the user.
3. Choose **Actions**, **Disable user**.
4. Confirm the user in the resulting dialog box and choose **Disable User**.

### Enabling Users

An AppStream 2.0 administrator can enable one or more user pool users, one at a time. After being enabled, the user can log in and launch applications from the stacks to which they are assigned. If the user was disabled, these assignments are retained.

**To enable users**

2. On the left navigation pane, choose **User Pool** and select the users.
3. Choose **Actions**, **Enable user**.
4. Confirm the user in the resulting dialog box and choose **Enable User**.

### Re-Sending Welcome Email

An AppStream 2.0 administrator can re-send the welcome email with connection instructions to user pool users. Unused passwords expire after seven days. To provide a new temporary password, the administrator must re-send the welcome email. This option is only available until the user sets their permanent password. If they've already set a password and have forgotten it, they can set a new one. For more information, see Resetting a Forgotten Password (p. 42).

**To resend the welcome email for a user**

2. On the left navigation pane, choose **User Pool** and select a user.
3. For **User Details**, choose **Resend welcome email**.
4. Confirm the success message at the top of the dashboard.
Amazon AppStream 2.0 supports identity federation to AppStream 2.0 stacks through Security Assertion Markup Language 2.0 (SAML 2.0). You can use an identity provider (IdP) that supports SAML 2.0—such as Active Directory Federation Services (AD FS) in Windows Server, Ping One Federation Server, or Okta—to provide an onboarding flow for your AppStream 2.0 users.

This feature offers your users the convenience of one-click access to their AppStream 2.0 applications using their existing identity credentials. You also have the security benefit of identity authentication by your IdP. By using your IdP, you can control which users have access to a particular AppStream 2.0 stack.

Example Authentication Workflow

The following diagram illustrates the authentication flow between AppStream 2.0 and a third-party IdP. In this example, the administrator has set up a sign-in page to access AppStream 2.0, called applications.exampleco.com. The webpage uses a SAML 2.0–compliant federation service to trigger a sign-on request. The administrator has also set up a user to allow access to AppStream 2.0.
1. The user browses to https://applications.exampleco.com. The sign-on page requests authentication for the user.

2. The federation service requests authentication from the organization's identity store.

3. The identity store authenticates the user and returns the authentication response to the federation service.

4. On successful authentication, the federation service posts the SAML assertion to the user's browser.

5. The user's browser posts the SAML assertion to the AWS Sign-In SAML endpoint (https://signin.aws.amazon.com/saml). AWS Sign-In receives the SAML request, processes the request, authenticates the user, and forwards the authentication token to AppStream 2.0.

6. Using the authentication token from AWS, AppStream 2.0 authorizes the user and presents applications to the browser.
From the user's perspective, the process happens transparently: The user starts at your organization's internal portal and lands at an AppStream 2.0 application portal, without ever having to supply any AWS credentials.

## Setting Up SAML

To enable users to sign in to AppStream 2.0 by using their existing credentials, and start streaming applications, you can set up identity federation using SAML 2.0. To do this, use an IAM role and a relay state URL to configure your SAML 2.0-compliant identity provider (IdP) and enable AWS to permit your federated users to access an AppStream 2.0 stack. The IAM role grants users the permissions to access the stack. The relay state is the stack portal to which users are forwarded after successful authentication by AWS.

### Contents

- Prerequisites (p. 47)
- Step 1: Create a SAML Identity Provider in AWS IAM (p. 47)
- Step 2: Create a SAML 2.0 Federation IAM Role (p. 47)
- Step 3: Embed an Inline Policy for the IAM Role (p. 48)
- Step 4: Configure Your SAML-Based IdP (p. 49)
- Step 5: Create Assertions for the SAML Authentication Response (p. 49)
- Step 6: Configure the Relay State of Your Federation (p. 50)

### Prerequisites

Complete the following prerequisites before configuring your SAML 2.0 connection.

1. Configure your SAML-based IdP to establish a trust relationship with AWS.
   - Inside your organization's network, configure your identity store to work with a SAML-based IdP. For configuration resources for using Ping Identity, Okta, Active Directory Federation Services (AD FS) in Windows Server, or Shibboleth as your SAML-based IdP, see AppStream 2.0 Integration with SAML 2.0 (p. 50).
   - Use your SAML-based IdP to generate and download a federation metadata document that describes your organization as an IdP. This signed XML document is used to establish the relying party trust. Save this file to a location that you can access from the IAM console later.
2. Use the AppStream 2.0 management console to create an AppStream 2.0 stack. You need the stack name to create the IAM policy and to configure your IdP integration with AppStream 2.0, as described later in this topic.

   You can create an AppStream 2.0 stack by using the AppStream 2.0 management console, AWS CLI, or AppStream 2.0 API. For more information, see Create AppStream 2.0 Stacks and Fleets (p. 27).

### Step 1: Create a SAML Identity Provider in AWS IAM

First, create a SAML IdP in AWS IAM. This IdP defines your organization's IdP-to-AWS trust relationship using the metadata document generated by the IdP software in your organization. For more information, see Creating and Managing a SAML Identity Provider (AWS Management Console) in the IAM User Guide.

### Step 2: Create a SAML 2.0 Federation IAM Role

Next, create a SAML 2.0 federation IAM role. This step establishes a trust relationship between IAM and your organization's IdP, which identifies your IdP as a trusted entity for federation.
To create an IAM role for the SAML IdP

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose Roles, Create role.
3. For Role type, choose SAML 2.0 federation.
4. For SAML Provider, select the SAML IdP that you created.
   **Important**
   Do not choose either of the two SAML 2.0 access level methods.
5. For Attribute, choose SAML:sub_type.
6. For Value, type persistent. This step restricts role access to only SAML user streaming requests that include a SAML subject type assertion with a value of persistent. If the SAML:sub_type is persistent, your IdP sends the same unique value for the NameID element in all SAML requests from a particular user. For more information about the SAML:sub_type assertion, see the Uniquely Identifying Users in SAML-Based Federation section in Using SAML-Based Federation for API Access to AWS.
7. Review your SAML 2.0 trust information, confirming the correct trusted entity and condition, and then choose Next: Permissions.
8. On the Attach permissions policies page, choose Next: Review. You create and embed an inline policy for this role later.
9. For Role name, type a name that helps you identify the purpose of this role. Because various entities might reference the role, you cannot edit the name of the role after it has been created.
10. (Optional) For Role description, type a description for the new role.
11. Review the role details and choose Create role.

Step 3: Embed an Inline Policy for the IAM Role

Next, embed an inline IAM policy for the role that you created. When you embed an inline policy, the permissions in the policy cannot be inadvertently attached to the wrong principal entity. The inline policy provides federated users with access to the AppStream 2.0 stack that you created. For information about how to embed the inline policy in JSON, see Create a Policy on the JSON Tab.

As you follow the steps in the procedure for embedding an inline policy for a user or role, note that you’ll create a policy on the JSON tab. To do this, copy and paste the following JSON policy into the JSON window and modify the resource by entering your AWS Region Code, account ID, and stack name. In the following policy, "Action": "appstream:Stream" is the action that provides your AppStream 2.0 users with permissions to connect to streaming sessions on the stack that you created.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "appstream:Stream",
         "Condition": {
            "StringEquals": {
               "appstream:userId": ":saml:sub",
               "saml:sub_type": "persistent"
            }
         }
      }
   ]
}
```
Choose a value for `REGION-CODE` that corresponds to the AWS Region where your AppStream 2.0 stack exists. Replace `STACK-NAME` with the name of the stack. Note that this value is case-sensitive, so the case in the stack name that you specify in this policy must match the case in the AppStream 2.0 stack name as it appears in the Stacks dashboard of the AppStream 2.0 management console.

**Note**
After you copy and paste the JSON policy, you may see an error message that indicates the validation failed. Ignore the error and continue with policy creation. You may also see an error on the Review policy page that indicates the policy does not grant any permissions. Ignore this error. The JSON policy is valid and provides the needed permissions.

**Step 4: Configure Your SAML-Based IdP**

Next, depending on your SAML-based IdP, you may need to manually update your IdP to trust AWS as a service provider by uploading the `saml-metadata.xml` file at https://signin.aws.amazon.com/static/saml-metadata.xml to your IdP. This step updates your IdP's metadata. For some IdPs, the update may already be configured. If this is the case, proceed to the next step.

If this update is not already configured in your IdP, review the documentation provided by your IdP for information about how to update the metadata. Some providers give you the option to type the URL, and the IdP obtains and installs the file for you. Others require you to download the file from the URL and then provide it as a local file.

**Step 5: Create Assertions for the SAML Authentication Response**

Next, depending on your SAML-based IdP, you may need to configure the information that the IdP passes as SAML attributes to AWS as part of the authentication response. For some IdPs, this information may already be configured. If this is the case, proceed to the next step.

If this information is not already configured in your IdP, provide the following:

- **SAML Subject NameID** – The unique identifier for the user who is signing in.
  
  **Note**
  For stacks with domain-joined fleets, the NameID value for the user must be provided in the format of "domain\username" using the `sAMAccountName` or "username@domain.com" using `userPrincipalName`. If you are using the `sAMAccountName` format, you can specify the `domain` by using either the NetBIOS name or the fully qualified domain name (FQDN). For more information, see Using Active Directory Domains with AppStream 2.0 (p. 52).

- **SAML Subject Type** (with a value set to `persistent`) – Setting the value to `persistent` ensures that your IdP sends the same unique value for the NameID element in all SAML requests from a particular user. Make sure that your IAM policy includes a condition to only allow SAML requests with a SAML sub_type set to `persistent`, as described in the section called “Step 2: Create a SAML 2.0 Federation IAM Role” (p. 47).

- **Attribute element with the Name attribute set to https://aws.amazon.com/SAML/Attributes/Role** – This element contains one or more `AttributeValue` elements that list the IAM role and SAML IdP to which the user is mapped by your IdP. The role and IdP are specified as a comma-delimited pair of ARNs.

- **Attribute element with the SessionDuration attribute set to https://aws.amazon.com/SAML/Attributes/SessionDuration (optional)** – This element contains one `AttributeValue` element that specifies the maximum amount of time that a federated streaming session for a user can remain active before reauthentication is required. The default value is 60 minutes. For more information, see the An optional Attribute element with the SessionDuration attribute set to https://aws.amazon.com/SAML/Attributes/SessionDuration section in Configuring SAML Assertions for the Authentication Response.
Step 6: Configure the Relay State of Your Federation

Finally, use your IdP to configure the relay state of your federation to point to the AppStream 2.0 stack relay state URL. After successful authentication by AWS, the user is directed to the AppStream 2.0 stack portal, defined as the relay state in the SAML authentication response.

The format of the relay state URL is as follows:

https://relay-state-region-endpoint?stack=stackname&accountId=aws-account-id-without-hyphens

Construct your relay state URL from your AWS account ID, stack name, and the relay state endpoint associated with the region in which your stack is located.

<table>
<thead>
<tr>
<th>Region</th>
<th>Relay state endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>us-east-1 (N.Virginia)</td>
<td><a href="https://appstream2.us-east-1.aws.amazon.com/saml">https://appstream2.us-east-1.aws.amazon.com/saml</a></td>
</tr>
<tr>
<td>eu-west-1 (Ireland)</td>
<td><a href="https://appstream2.eu-west-1.aws.amazon.com/saml">https://appstream2.eu-west-1.aws.amazon.com/saml</a></td>
</tr>
<tr>
<td>ap-northeast-1 (Tokyo)</td>
<td><a href="https://appstream2.ap-northeast-1.aws.amazon.com/saml">https://appstream2.ap-northeast-1.aws.amazon.com/saml</a></td>
</tr>
<tr>
<td>ap-southeast-1 (Singapore)</td>
<td><a href="https://appstream2.ap-southeast-1.aws.amazon.com/saml">https://appstream2.ap-southeast-1.aws.amazon.com/saml</a></td>
</tr>
</tbody>
</table>

AppStream 2.0 Integration with SAML 2.0

The following links help you configure third-party SAML 2.0 identity provider solutions to work with AppStream 2.0.

<table>
<thead>
<tr>
<th>IdP solution</th>
<th>More information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ping Identity</td>
<td>Configuring an SSO connection to Amazon AppStream 2.0 — Describes how to set up single sign-on (SSO) to AppStream 2.0.</td>
</tr>
<tr>
<td>Okta</td>
<td>How to Configure SAML 2.0 for Amazon AppStream 2.0 — Describes how to use Okta to set up SAML federation to AppStream 2.0. For stacks that are joined to a domain, the &quot;Application username format&quot; must be set to &quot;AD user principal name&quot;.</td>
</tr>
</tbody>
</table>
### IdP solution

<table>
<thead>
<tr>
<th>IdP solution</th>
<th>More information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Directory Federation Services (AD FS) for Windows Server</td>
<td>Enabling Identity Federation with AD FS 3.0 and Amazon AppStream 2.0 — Describes how to provide users with SSO access to AppStream 2.0 by using their existing enterprise credentials. You can configure federated identities for AppStream 2.0 by using AD FS 3.0.</td>
</tr>
<tr>
<td>Shibboleth</td>
<td>Single Sign-On: Integrating AWS, OpenLDAP, and Shibboleth — Describes how to set up the initial federation between the Shibboleth IdP and the AWS Management Console. You must complete the following additional steps to enable federation to AppStream 2.0. Step 4 of the AWS Security whitepaper describes how to create IAM roles that define the permissions that federated users have to the AWS Management Console. After you create these roles and embed the inline policy as described in the whitepaper, modify this policy so that it provides federated users with permissions to access only an AppStream 2.0 stack. To do this, replace the existing policy with the policy noted in Step 3: Embed an Inline Policy for the IAM Role, in Setting Up SAML (p. 47). When you add the stack relay state URL as described in Step 6: Configure the Relay State of Your Federation, in Setting Up SAML (p. 47), add the relay state parameter to the federation URL as a target request attribute. For information about configuring relay state parameters, see the SAML 2.0 section in the Shibboleth documentation.</td>
</tr>
</tbody>
</table>

For solutions to common problems you may encounter, see Troubleshooting (p. 75).

For more information about additional supported SAML providers, see Integrating Third-Party SAML Solution Providers with AWS in the IAM User Guide.
Using Active Directory Domains with AppStream 2.0

You can join your Amazon AppStream 2.0 fleets and image builders to Microsoft Active Directory domains. You can use your existing Microsoft Active Directories, either cloud-based or on-premises, for launching domain-joined streaming instances. You can also use AWS Directory Service to create an Active Directory and use that to support your AppStream 2.0 resources. For more information about using AWS Directory Service, see Microsoft Active Directory in the AWS Directory Service Administration Guide.

Joining AppStream 2.0 to your Active Directory domain offers the following key benefits:

• Allow your users and applications to access your Active Directory resources such as printers and file shares from streaming sessions.

• Apply directory-based group policies to your streaming instances to configure and control the end user experience.

• Stream applications such as Microsoft SharePoint or Microsoft Outlook that require users to be authenticated using their Active Directory login credentials.

• Apply your enterprise compliance and security policies to your AppStream 2.0 streaming instances.

Contents

• Overview of Active Directory Domains (p. 52)
• Before You Begin Using Active Directory with AppStream 2.0 (p. 54)
• Tutorial: Setting Up the Active Directory (p. 55)
• AppStream 2.0 Active Directory Administration (p. 57)
• More Info (p. 63)

Overview of Active Directory Domains

Using Active Directory domains with AppStream 2.0 requires some understanding of how they work together, and the tasks required. The following tasks must be completed by an administrator:

1. Set the group policies controlling the end user experience and security requirements for applications.
2. Create the domain-joined application stack in AppStream 2.0.
3. Create the AppStream 2.0 app in the SAML 2.0 identity provider and assign it to end users either directly or with Active Directory groups.

To authenticate users for proper access to the joined domain from within their streaming apps, there are several steps that occur when a user initiates an AppStream 2.0 streaming session. The following diagram illustrates the end-to-end user authentication flow from the initial browser request through all the SAML and Active Directory authentication steps.
User Authentication Flow

1. The user browses to https://applications.exampleco.com. The sign-on page requests authentication for the user.
2. The federation service requests authentication from the organization's identity store.
3. The identity store authenticates the user and returns the authentication response to the federation service.
4. On successful authentication, the federation service posts the SAML assertion to the user's browser.
5. The user's browser posts the SAML assertion to the AWS Sign-In SAML endpoint (https://signin.aws.amazon.com/saml). AWS Sign-In receives the SAML request, processes the request, authenticates the user, and forwards the authentication token to the AppStream 2.0 service.
6. Using the authentication token from AWS, AppStream 2.0 authorizes the user and presents applications to the browser.
7. The user chooses an app and is prompted to enter login information for the domain.
8. The domain controller is contacted for user authentication.
9. After being authenticated with the domain, the user's session starts with domain connectivity.

From the user's perspective, the process happens transparently: The user starts at your organization's internal portal and lands at an AppStream 2.0 application portal, without ever having to supply any AWS credentials. Only Active Directory domain login credentials are required.

The administrator's tasks, such as to set up the Active Directory with entitlements and group policies, and to create a domain-joined application stack, must be completed before the user can initiate this authentication process.

Before You Begin Using Active Directory with AppStream 2.0

Before you use Microsoft Active Directory with AppStream 2.0, be aware of the following requirements.

Requirements

- You need an Active Directory—compliant domain to which to join your streaming instances. If you don't have an Active Directory or want to use your on-premises Active Directory, see Active Directory Domain Services on the AWS Cloud: Quick Start Reference Deployment.
- You need a domain service account with permissions to create and manage computer objects in the domain that you intend to use with AppStream 2.0.
- To create a service account using a domain controller, see Creating a service account in the Microsoft documentation.
- While associating this Active Directory with AppStream 2.0, provide the service account name and password. AppStream 2.0 uses this account to create and manage computer objects within the directory. For more information, see Granting Permissions to Create and Manage Active Directory Computer Objects (p. 57).
- When you register your Active Directory with AppStream 2.0, you must provide an organizational unit (OU) distinguished name. Create an OU for this purpose. The default Computers container is not an OU and cannot be used by AppStream 2.0. For more information, see Finding the Organizational Unit Distinguished Name (p. 58).
- The directories to use with AppStream 2.0 must be accessible through their fully qualified domain names (FQDNs) via the VPC in which your streaming instances are launched. For more information, see Active Directory and Active Directory Domain Services Port Requirements in the Microsoft documentation.
- SAML 2.0-based user federation is required to stream from domain-joined fleets. You cannot launch sessions to domain-joined instances using CreateStreamingURL, or the AppStream 2.0 User Pool.
- You must use an image that supports joining image builders and fleets to an Active Directory domain. All public images published on or after July 24, 2017 support joining an Active Directory domain. For more information, see Amazon AppStream 2.0 Windows Image Version History (p. 22) and Tutorial: Setting Up the Active Directory (p. 55).
Tutorial: Setting Up the Active Directory

To use Active Directory domain support in AppStream 2.0, you must first register your directory configuration by creating a Directory Config object using the AppStream 2.0 management console or using AWS SDK or AWS CLI. You can then use the directory configuration to launch domain-joined fleets and image builders.

Tasks

- Step 1: Create a Directory Configuration (p. 55)
- Step 2: Create an Image Using a Domain-Joined Image Builder (p. 55)
- Step 3: Create a Domain-Joined Fleet (p. 56)
- Step 4: Configure SAML 2.0 (p. 56)

Step 1: Create a Directory Configuration

This step creates the necessary Active Directory configuration in AppStream 2.0. The directory configuration is used in later steps.

If you are using the AWS SDK, you can use the CreateDirectoryConfig operation. If you are using the AWS CLI, you can use the create-directory-config command.

To create a directory configuration using the console

2. On the left navigation pane, choose Directory Configs, Create Directory Config.
3. For Directory Name, provide the fully qualified domain name (FQDN) of the Active Directory. Each region can have only one Directory Config value with a specific directory name.
4. For Service Account Name, enter the service account name with the required permissions. For more information, see Granting Permissions to Create and Manage Active Directory Computer Objects (p. 57). The service account name must be in the format DOMAIN\username.
5. For Password and Confirm Password, type the directory password for the specified service account.
6. For Organizational Unit (OU), type the distinguished name of at least one OU for streaming instance computer objects. The default Computers container is not an OU and cannot be used by AppStream 2.0. For more information, see Finding the Organizational Unit Distinguished Name (p. 58).
7. To add more than one OU, select the plus sign (+) next to Organizational Unit (OU). To remove OUs, choose the x icon.
8. Choose Next.
9. Review the configuration information and choose Create.

Step 2: Create an Image Using a Domain-Joined Image Builder

Using the AppStream 2.0 image builder, create a new image with Active Directory domain-join capabilities. Note that the fleet and image can have different domains. You join the image builder to a domain to enable domain join and to install apps. Fleet domain join is discussed in the next section.

To create an image for launching domain-joined fleets

1. Refer to the procedures in Tutorial: Create a Custom Image (p. 17).
2. For the base image selection step, use an AWS base image released on or after July 24, 2017. For a current list of released AWS images, see Amazon AppStream 2.0 Windows Image Version History (p. 22).

3. For Step 3: Configure Network, select a VPC and subnets with network connectivity to your Active Directory environment. Select the security groups that are set up to allow access to your directory via your VPC subnets.

4. Also in Step 3: Configure Network, expand the Active Directory Domain (Optional) section, and select values for Directory Name and Directory OU to which the image builder should be joined.

5. Review the image builder configuration and choose Create.

6. Wait for the new image builder to reach a Running state, and choose Connect.

7. Log in to the image builder as Administrator or as a directory user with local administrator permissions. For more information, see Providing Local Administrator Permissions for Image Builders (p. 59).

8. Complete the steps provided in Tutorial: Create a Custom Image (p. 17) to install apps and create the new image.

**Step 3: Create a Domain-Joined Fleet**

Using the private image created in the previous step, create an Active Directory domain-joined fleet for streaming apps. The domain can be different than the one that was used for the image builder to create the image.

To create a domain-joined fleet

1. Follow the procedures in Create a Fleet (p. 28).

2. For the image selection step, use the image that was created in the previous step, Step 2: Create an Image Using a Domain-Joined Image Builder (p. 55).

3. For Step 4: Configure Network, select a VPC and subnets with network connectivity to your Active Directory environment. Select the security groups that are set up to allow communication to your domain.

4. Also in Step 4: Configure Network, expand the Active Directory Domain (Optional) section, and select values for Directory Name and Directory OU to which the fleet should be joined.

5. Review the fleet configuration and choose Create.

6. Complete the remaining steps provided in Create AppStream 2.0 Stacks and Fleets (p. 27) so that your fleet is associated with a stack and running.

**Step 4: Configure SAML 2.0**

Your users must use your SAML 2.0-based user federation to launch streaming sessions from your domain-joined fleet.

To configure SAML 2.0 for single sign-on access

1. Refer to the procedures in Setting Up SAML (p. 47).

2. AppStream 2.0 requires that the SAML_Subject "NameID" attribute specifies the domain username for the user logging in. The username must be provided in the format of "domain\username" using the sAMAccountName or "username@domain.com" using userPrincipalName. If using the sAMAccountName format, the domain can be specified using either the NetBIOS name or the fully qualified domain name (FQDN).

3. Provide access to your Active Directory users or groups to enable access to the AppStream 2.0 stack from your identity provider application portal.
4. Complete the remaining steps provided in Setting Up SAML (p. 47).

To log in a user with SAML 2.0

1. Log in to your SAML 2.0 provider’s application catalog and launch the AppStream 2.0 SAML app created in the previous procedure.
2. When you see the AppStream 2.0 app catalog, select an application to launch.
3. When you see a loading icon, you are prompted to provide a password. The domain username provided by your SAML 2.0 identity provider appears above the password field. Provide your password, and choose log in.

The streaming instance performs the Windows login procedure, and the selected app launches.

AppStream 2.0 Active Directory Administration

When setting up and using Active Directory with AppStream 2.0, refer to the following administrator tasks.

Tasks
- Granting Permissions to Create and Manage Active Directory Computer Objects (p. 57)
- Finding the Organizational Unit Distinguished Name (p. 58)
- Providing Local Administrator Permissions for Image Builders (p. 59)
- Updating the Service Account Used for Joining the Domain (p. 60)
- Locking the Streaming Session When the User is Idle (p. 61)
- Editing the Directory Configuration (p. 61)
- Deleting a Directory Configuration (p. 61)
- Configuring AppStream 2.0 to Use Domain Trusts (p. 62)
- Managing AppStream 2.0 Computer Objects in the Active Directory (p. 62)

Granting Permissions to Create and Manage Active Directory Computer Objects

To allow AppStream 2.0 to perform Active Directory computer object operations, you need an account with the right permissions. As a best practice, you should use an account that has only the minimum privileges necessary. The minimum Active Directory organizational unit (OU) permissions are as follows:

- Create Computer Object
- Change Password
- Reset Password
- Write Description

Prerequisites

Before setting up permissions, you must complete the following tasks:

- Get access to a computer or EC2 instance that is joined to your domain.
Finding the Organizational Unit Distinguished Name

To set up minimum permissions

1. Launch **Active Directory Users and Computers** in your domain or on your domain controller.
2. In the navigation tree on the left, select the first OU on which to provide domain join privileges, open the context menu (right-click), and choose **Delegate Control**.
3. On the **Delegation of Control Wizard** page, choose **Next, Add**.
4. For **Select Users, Computers, or Groups**, select the pre-created user account, service account, or group, and choose **OK**.
5. On the **Tasks to Delegate** page, choose **Create a custom task to delegate**, **Next**.
6. Choose **Only the following objects in the folder, Computer objects**.
7. Choose **Create selected objects in this folder**, **Next**.
8. For **Permissions**, choose **Read, Write, Change Password, Reset Password**, **Next**.
9. On the **Completing the Delegation of Control Wizard** page, verify the information and choose **Finish**.
10. Repeat steps 2-10 for any additional OUs that require these permissions.

If you delegated permissions to a group, create a user or service account with a strong password and add that account to the group. This account will then have sufficient privileges to connect your streaming instances to the directory. Use this account when creating your AppStream 2.0 directory configuration.

Finding the Organizational Unit Distinguished Name

When you register your Active Directory with AppStream 2.0, you must provide an organizational unit (OU) distinguished name. Create an OU for this purpose. The default Computers container is not an OU and cannot be used by AppStream 2.0. The following procedure shows how to obtain this name.

**Note**

The distinguished name must start with `OU=` or it cannot be used for computer objects.

**Prerequisites**

- Get access to a computer or EC2 instance that is joined to your domain.
- Install the Active Directory User and Computers MMC snap-in. For more information, see Installing or Removing Remote Server Administration Tools for Windows 7 in the Microsoft documentation.
- Log in as a domain user with appropriate permissions to read the OU security properties.

**To find the distinguished name of an OU**

1. Launch **Active Directory Users and Computers** in your domain or on your domain controller.
2. Under **View**, ensure that **Advanced Features** is enabled.
3. In the navigation tree on the left, select the first OU to use for AppStream 2.0 streaming instance computer objects, open the context menu (right-click), and choose **Properties**.
4. Choose **Attribute Editor**.
5. Under **Attributes**, for **distinguishedName**, choose **View**.
Providing Local Administrator Permissions for Image Builders

By default, Active Directory domain users do not have local administrator permissions on image builder instances. You can provide these permissions using Group Policy preferences in your directory, or manually using the local administrator account. This allows a domain user to install applications and create images in an AppStream 2.0 image builder.

Contents
- Using Group Policy Permissions (p. 59)
- Using Local Administrator (p. 60)

Using Group Policy Permissions

Group Policy can be used to grant local administrator permissions to Active Directory users or groups and automatically to all computer objects in the specified OU. The Active Directory users or groups to which to grant local administrator permissions must already exist.

Prerequisites
- Get access to a computer or EC2 instance that is joined to your domain.
- Install the Group Policy Management MMC snap-in. For more information, see Installing or Removing Remote Server Administration Tools for Windows 7 in the Microsoft documentation.
- Log in as a domain user with appropriate permissions to create Group Policy Objects (GPO) and link them to the appropriate OUs.

To provide permissions using Group Policy

1. Launch Group Policy Management in your directory or on your domain controller.
2. In the navigation tree on the left, select the organizational unit (OU) in which to create the GPO, open the context (right-click) menu, and choose Create a GPO in this domain, Link it here....
3. For Name, provide a descriptive name for this GPO.
4. Select the newly created GPO, open the context (right-click) menu, and choose Edit.
5. In the left navigation tree, choose Computer Configuration, Preferences, Windows Settings, Control Panel Settings, and Local Users and Groups.
6. Select Local Users and Groups selected, open the context (right-click) menu, and choose New, Local Group.
7. For Action, choose Update.
8. For Group name, choose Administrators (built-in).
9. Under Members, choose Add... and specify the Active Directory user accounts or groups to which to assign local administrator rights on the streaming instance. For Action, choose Add to this group, and choose OK.
10. To apply this GPO to other OUs, select the additional OU, open the context (right-click) menu and choose Link an Existing GPO.
11. Using the name specified in step 3, scroll to find the created GPO, and choose OK. The GPO now shows up in the navigation tree on the left within the OU.
12. Repeat steps 10-11 for additional OUs that should have this policy.
13. Choose OK to close the New Local Group Properties window, Choose OK again to close the Group Policy Management Editor window.

Any running image builders or fleets must be stopped then started to apply the new policy. Streaming image builders and fleets created in the OU in which the specified GPO is linked automatically provide local administrator rights to the specified users or groups.

### Using Local Administrator

Use the pre-created local administrator account to manually add Active Directory users or groups to the local administrator group to grant administrator rights. New image builders created from this image builder maintain the permissions.

The Active Directory users or groups to which to grant local administrator rights must already exist.

**To provide permissions using local admin rights**

2. Select the image builder, and choose Connect. The image builder must be running and domain-joined. For more information, see Tutorial: Setting Up the Active Directory (p. 55).
3. Log in to the administrator account.
5. In the left navigation pane, choose Local Users and Groups and open the Groups folder.
6. Open the Administrators group and choose Add....
7. Select all Active Directory users or groups to which to assign local administrator rights and choose OK. Choose OK again to close the Administrator Properties window.
9. To log in as a Active Directory user and test local administrator access rights, choose Admin Commands, Switch user.

### Updating the Service Account Used for Joining the Domain

To update the service account that AppStream 2.0 uses for joining the domain, we recommend using two service accounts for joining image builders and fleets to your Active Directory. Using two separate service accounts ensures that you have no disruption in service when a service account needs to be updated, for example when a password expires.

**To update a service account**

1. Create an Active Directory group and delegate the correct permissions to the group.
2. Add your service accounts to the new Active Directory group.
3. When needed, edit your AppStream 2.0 directory configuration to provide the other service account username and password.

After you've set up the Active Directory group with the new service account, any new streaming instance operations use the new service account, while in-process streaming instance operations continue using the old account without interruption.

The service account overlap time while the in-process streaming instance operations complete is very short, no more than a day. The overlap time is needed because you shouldn't delete or change the password for the old service account during the overlap period, or existing operations can fail.
Locking the Streaming Session When the User is Idle

AppStream 2.0 relies on Microsoft Active Directory Group Policies to lock the streaming session when your user goes idle.

To automatically lock the streaming instance when your user is idle

1. Follow the instructions at Customizing the Desktop in the Microsoft documentation.
2. For Enable screen saver, choose Enabled.
3. For Force specific screen saver, choose Enabled and type `scrnsave.scr`. This setting shows a black screen when the screen saver comes on.
4. For Password protect the screen saver, choose Enabled.
5. For Screen saver timeout, enter the appropriate amount of time, in seconds. This is the duration of user inactivity before the screen saver applies. For example, to set an idle duration of 10 minutes idle, specify a value of 600 seconds.

Additionally, set the following policy to Disabled: User Configuration > Administrative Templates > System > Ctrl+Alt+Del Options > Remove Lock Computer.

Editing the Directory Configuration

After a AppStream 2.0 directory configuration has been created, you can edit it to add, remove, or modify organizational units, update the service account username, or update the service account password.

To update a directory configuration

2. In the left navigation pane, choose Directory Configs and select the directory configuration to edit.
3. Choose Actions, Edit.
4. Update the fields to be changed. To add additional OUs, select the plus sign (+) next to the topmost OU field. To remove an OU field, select the x next to the field.
   
   **Note**
   
   At least one OU is required. OUs that are currently in use cannot be removed.
5. To save changes, choose Update Directory Config.
6. The information in the Details tab should now update to reflect the changes.

Changes to the service account username and password do not impact in-process streaming instance operations. New streaming instance operations use the updated credentials. For more information, see Updating the Service Account Used for Joining the Domain (p. 60).

Deleting a Directory Configuration

You can delete a AppStream 2.0 directory configuration that is no longer needed. Directory configurations that are associated with any image builders or fleets cannot be deleted.

To delete a directory configuration

2. In the left navigation pane, choose Directory Configs and select the directory configuration to delete.
3. Choose **Actions, Delete**.
4. Verify the name in the pop-up message, and choose **Delete**.
5. Choose **Update Directory Config**.

**Configuring AppStream 2.0 to Use Domain Trusts**

AppStream 2.0 supports Active Directory domain environments where network resources such as file servers, applications, and computer objects reside in one domain, and the user objects reside in another. The domain service account used for computer object operations does not need to be in the same domain as the AppStream 2.0 computer objects.

When creating the directory configuration, specify a service account that has the appropriate permissions in the server Active Directory domain to manage computer objects.

Your end user Active Directory accounts must have the "Allowed to Authenticate" permissions for the following:

- AppStream 2.0 computer objects
- Domain controllers for the domain

For more information, see [Granting Permissions to Create and Manage Active Directory Computer Objects](p. 57).

**Managing AppStream 2.0 Computer Objects in the Active Directory**

AppStream 2.0 does not delete computer objects from your Active Directory. These computer objects can be easily identified in your directory. Each computer object in the directory is created with the Description attribute specifying a fleet or an image builder instance and the name.

**Computer Object Description Examples**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fleet</td>
<td>ExampleFleet</td>
<td>AppStream 2.0 - fleet:ExampleFleet</td>
</tr>
<tr>
<td>Image builder</td>
<td>ExampleImageBuilder</td>
<td>AppStream 2.0 - image-builder:ExampleImageBuilder</td>
</tr>
</tbody>
</table>

You can identify and delete inactive computer objects created by AppStream 2.0 with the following `dsquery` and `dsrm` commands. For more information, see `Dsquery` and `Dsrm` in the Microsoft documentation.

The `dsquery` command is used to identify inactive computer objects over a certain period of time, and uses the following format. The `dsquery` command should also be run with the parameter `-desc "AppStream 2.0*"` to show only AppStream 2.0 objects.

```
dsquery computer "OU-distinguished-name" -desc "AppStream 2.0*" -inactive number-of-weeks-since-last-login
```

- `OU-distinguished-name` is the distinguished name of the organizational unit. For more information, see [Finding the Organizational Unit Distinguished Name](p. 58). If you don’t provide the `OU-distinguished-name` parameter, the command searches the entire directory.
• **number-of-weeks-since-last-log-in** is the desired value based on how you'd like to define what's "inactive".

For example, the following command displays all computer objects in the OU=ExampleOU,DC=EXAMPLECO,DC=COM organizational unit that have not logged in within the past two weeks.

```
dsquery computer OU=ExampleOU,DC=EXAMPLECO,DC=COM -desc "AppStream 2.0*" -inactive 2
```

If any matches are found, the result is one or more object names. The `dsrm` command deletes the specified object and uses the following format:

```
dsrm objectname
```

Where `objectname` is the full object name from the output of the `dsquery` command. For example, if the `dsquery` command above results in a computer object named "ExampleComputer", the `dsrm` command to delete it would be as follows:

```
dsrm "CN=ExampleComputer,OU=ExampleOU,DC=EXAMPLECO,DC=COM"
```

You can chain these commands together using the pipe (|) operator. For example, to delete all AppStream 2.0 computer objects, prompting for confirmation for each, use the following format. Add the `-noprompt` parameter to `dsrm` to disable confirmation.

```
dsquery computer OU-distinguished-name -desc "AppStream 2.0*" -inactive number-of-weeks-since-last-log-in | dsrm
```

**More Info**

For more information related to this topic, see the following resources:

- **Troubleshooting Notification Codes (p. 81)**—Resolutions to notification code errors.
- **Troubleshooting Active Directory Domain Join (p. 78)**—Help with common difficulties.
- **Microsoft Active Directory**—Information about using AWS Directory Service.
Monitoring Amazon AppStream 2.0 Resources

AppStream 2.0 publishes metrics to Amazon CloudWatch to enabled detailed tracking and deep dive analysis. These statistics are recorded for an extended period so you can access historical information and gain a better perspective on how your fleets are performing. For more information, see the Amazon CloudWatch User Guide.

Viewing Fleet Usage Using the Console

You can monitor Amazon AppStream 2.0 using the AppStream 2.0 console or the CloudWatch console.

To view fleet usage in the AppStream 2.0 console
2. In the left pane, choose Fleets.
3. Select a fleet and choose its Fleet Usage tab.
4. By default, the graph displays the following metrics: ActualCapacity, InUseCapacity, and CapacityUtilization. You can select additional metrics to graph or change the period.

To view fleet usage in the CloudWatch console
2. In the left pane, choose Metrics.
3. Choose the AppStream namespace and then choose Fleet Metrics.
4. Select the metrics to graph.

AppStream 2.0 Metrics and Dimensions

Amazon AppStream 2.0 sends the following metrics and dimension information to Amazon CloudWatch.

Amazon AppStream 2.0 Metrics

AppStream 2.0 sends metrics to CloudWatch one time every minute. The AWS/AppStream namespace includes the following metrics.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
</table>
| ActualCapacity       | The total number of instances that are available for streaming or are currently streaming.  
|                      | ActualCapacity = AvailableCapacity + InUseCapacity                          |
|                      | Units: Count                                                                |
|                      | Valid statistics: Average, Minimum, Maximum                                  |
### Metric | Description
--- | ---
**AvailableCapacity** | The number of idle instances currently available for user sessions.  
\[ \text{AvailableCapacity} = \text{ActualCapacity} - \text{InUseCapacity} \]  
Units: Count  
Valid statistics: Average, Minimum, Maximum

**CapacityUtilization** | The percentage of instances in a fleet that are being used, using the following formula.  
\[ \text{CapacityUtilization} = \left( \frac{\text{InUseCapacity}}{\text{ActualCapacity}} \right) \times 100 \]  
Monitoring this metric helps with decisions about increasing or decreasing the value of a fleet's desired capacity.  
Units: Percent  
Valid statistics: Average, Minimum, Maximum

**DesiredCapacity** | The total number of instances that are either running or pending. This represents the total number of concurrent streaming sessions your fleet can support in a steady state.  
\[ \text{DesiredCapacity} = \text{ActualCapacity} + \text{PendingCapacity} \]  
Units: Count  
Valid statistics: Average, Minimum, Maximum

**InUseCapacity** | The number of instances currently being used for streaming sessions. One InUseCapacity count represents one streaming session.  
Units: Count  
Valid statistics: Average, Minimum, Maximum

**PendingCapacity** | The number of instances being provisioned by AppStream 2.0. Represents the additional number of streaming sessions the fleet can support after provisioning is complete. When provisioning starts, it usually takes 10-20 minutes for an instance to become available for streaming.  
Units: Count  
Valid statistics: Average, Minimum, Maximum

**RunningCapacity** | The total number of instances currently running. Represents the number of concurrent streaming sessions that can be supported by the fleet in its current state.  
This metric is provided for Always-On fleets only, and has the same value as the ActualCapacity metric.  
Units: Count  
Valid statistics: Average, Minimum, Maximum
<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
</table>
| InsufficientCapacity  | The number of session requests rejected due to lack of capacity.  
                          | You can set alarms to use this metric to be notified of users waiting for streaming sessions.  
                          | Units: Count                           
                          | Valid statistics: Average, Minimum, Maximum, Sum |

**Dimensions for Amazon AppStream 2.0 Metrics**

To filter the metrics provided by Amazon AppStream 2.0, use the following dimension.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fleet</td>
<td>The name of the fleet.</td>
</tr>
</tbody>
</table>
Controlling Access to Amazon AppStream 2.0 with IAM Roles and Policies

By default, IAM users don't have permission to create or modify AppStream 2.0 resources, use Fleet Auto Scaling, or perform tasks using the AppStream 2.0 API. To allow IAM users to create or modify resources and perform tasks, you must create IAM policies that grant permissions on specific resources and API actions, and then attach those policies to the IAM users or groups that require those permissions.

While creating AppStream 2.0 resources, AppStream 2.0 makes API calls to other AWS services on behalf of the user. This authentication is accomplished by the service assuming specific IAM roles available in the user's account. These IAM roles are created by the service when the user gets started with the service in an AWS region. These IAM roles are created by the service when the user gets started with the service in an AWS region.

**Note**
IAM roles and policies control which AppStream 2.0 resources can be accessed, while the user pool controls access to AppStream 2.0 itself. For more information, see Manage Access Using the AppStream 2.0 User Pool (p. 41).

Contents
- IAM Service Role (p. 67)
- Identity Based Policies (p. 67)
- Application Auto Scaling IAM Role (p. 67)
- Application Auto Scaling Required IAM Permissions (p. 68)
- IAM Policies and the Amazon S3 Bucket for Home Folders (p. 69)

IAM Service Role

The AmazonAppStreamServiceAccess service role is available for AppStream 2.0.

Identity Based Policies

AppStream 2.0 provides managed policies that you can attach to your IAM users and allow users to control your AppStream 2.0 resources or perform API actions through the AWS CLI, SDK, or console.

**AmazonAppStreamFullAccess**

Provides full administrator access to AppStream 2.0 resources.

**AmazonAppStreamReadOnlyAccess**

Provides read-only access to AppStream 2.0 resources.

Application Auto Scaling IAM Role

Before you can use AppStream 2.0 Fleet Auto Scaling, Application Auto Scaling needs permissions to access Amazon CloudWatch alarms and AppStream 2.0 fleets. The service role
Application Auto Scaling Required IAM Permissions

To use AppStream 2.0 Fleet Auto Scaling, the IAM user accessing fleet creation and settings must have appropriate permissions for the services that support scaling. AppStream 2.0 requires the following permissions:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["appstream:*"],
            "Resource":[
                "*"
            ]
        }
    ]
}
```
IAM Policies and the Amazon S3 Bucket for Home Folders

Access to the Amazon S3 bucket for Home folders is managed using IAM permissions and policies.

Examples
- Deleting the Amazon S3 Bucket for Home Folders (p. 69)
- Restricting Administrator Access to the Amazon S3 Bucket for Home Folders (p. 69)

Deleting the Amazon S3 Bucket for Home Folders

AppStream 2.0 adds an Amazon S3 bucket policy that prevents accidental deletion of the S3 bucket, shown below. To delete the S3 bucket, delete the S3 bucket policy first, and then delete the S3 bucket. For more information, see Deleting or Emptying a Bucket in the Amazon Simple Storage Service Developer Guide.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PreventAccidentalDeletionOfBucket",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:DeleteBucket",
      "Resource": "arn:aws:s3:::appstream2-36fb080bb8-region-code-account-id-without-hyphens"
    }
  ]
}
```

Restricting Administrator Access to the Amazon S3 Bucket for Home Folders

By default, administrators who can access the Amazon S3 bucket created by AppStream 2.0 can view and modify content that is part of users' Home folders. To restrict administrator access to the S3 bucket containing user files, we recommend applying the S3 bucket access policy based on the following policy template:
This policy only allows Home folder S3 bucket access to the users specified and to the AppStream 2.0 service. For every IAM user who should have access, replicate the following line:

```
"arn:aws:iam::account:user/IAM-user-name"
```

For example, the following policy restricts access to the Home folder S3 bucket for anyone other than IAM users bobsmith and terrypratt, and the AppStream 2.0 service, in region us-west-2 for account ID 123456789012.

```
{  
  "Sid": "RestrictedAccess",  
  "Effect": "Deny",  
  "NotPrincipal":  
  {  
    "AWS": [  
      "arn:aws:iam::123456789012:role/service-role/AmazonAppStreamServiceAccess",  
      "arn:aws:sts::123456789012:assumed-role/AmazonAppStreamServiceAccess/PhotonSession",  
      "arn:aws:iam::123456789012:user/bobsmith",  
      "arn:aws:iam::123456789012:user/terrypratt"
    ]
  },  
  "Action": "s3:*",  
  "Resource": "arn:aws:s3:::appstream2-36fb080bb8-region-account"
}
```
Tagging Your Amazon AppStream 2.0 Resources

AWS enables you to assign metadata to your AWS resources in the form of tags. You can use these tags to help manage your AppStream 2.0 image builders, images, fleets, and stacks, and also organize data, including billing data.

You can:

- Logically group resources in different ways (for example, by purpose, owner, or environment).
  This is useful when you have many resources of the same type.
- Quickly identify a specific resource based on the tags you’ve assigned to it
- Identify and control AWS costs

For example, you can identify and group AppStream 2.0 fleets that are in different environments (such as Development or Production) or that are assigned to different business units (such as HR or Marketing). You can then track the associated AWS costs for these fleets on a detailed level. To do this, sign up to get your AWS account bill with tag key values included. For more information about setting up a cost allocation report with tags, see Monthly Cost Allocation Report in the AWS Billing and Cost Management User Guide.

Contents

- Tagging Basics (p. 71)
- Tag Restrictions (p. 72)
- Working with Tags in the AppStream 2.0 Console (p. 72)
- Working with Tags by Using the AppStream 2.0 API, an AWS SDK, or AWS CLI (p. 72)

Tagging Basics

Tags consist of a key-value pair, similar to other AWS services tags. To tag a resource, you specify a key and a value for each tag. A key can be a general category, such as “project”, “owner”, or “environment”, with specific associated values, and you can share the same key and value across multiple resources. You can tag an AppStream 2.0 resource immediately after you create it or later on. If you delete a resource, the tags are removed from that resource on deletion. However, other AppStream 2.0 and AWS resources that have the same tag key are not impacted.

You can edit tag keys and values, and you can remove tags from a resource at any time. You can set the value of a tag to an empty string, but you can’t set the name of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. If you delete a resource, any tags for the resource are also deleted.

Note

If you plan to set up a monthly cost allocation report to track AWS costs for AppStream 2.0 resources, keep in mind that tags added to existing AppStream 2.0 resources appear in your cost allocation report on the first of the following month for resources that are renewed in that month.
Tag Restrictions

- The maximum number of tags per AppStream 2.0 resource is 50.
- The maximum key length is 128 Unicode characters in UTF-8.
- The maximum value length is 256 Unicode characters in UTF-8.
- Tag keys and values are case-sensitive.
- Do not use the "aws:" prefix in your tag names or values because it is a system tag that is reserved for AWS use. You cannot edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per resource limit.
- You can only use the following special characters: + - = . _ : /
  @.
- Although you can share the same key and value across multiple resources, you cannot have duplicate keys on the same resource.
- Tags can only be added to resources that are already created (you cannot specify tags on resource creation).

Working with Tags in the AppStream 2.0 Console

You can add, edit, and delete tags for existing resources by using the AppStream 2.0 console.

To add, edit, or delete tags for an existing AppStream 2.0 resource

2. From the navigation bar, select the region that contains the resource for which you want to add, edit, or delete tags.
3. In the navigation pane, select the resource type. The resource type can be an image builder, image, fleet, or stack.
4. Select the resource from the resource list.
5. Choose **Tags, Add/Edit Tags**, and then do one or more of the following:
   - To add a tag, choose **Add Tag**, and then specify the key and value for each tag.
   - To edit a tag, modify the key and value for the tag as needed.
   - To delete a tag, choose the **Delete** icon (X) for the tag.
6. Choose **Save**.

Working with Tags by Using the AppStream 2.0 API, an AWS SDK, or AWS CLI

If you’re using the AppStream 2.0 API, an AWS SDK or the AWS Command Line Interface (CLI), you can use the following AppStream 2.0 actions to add, edit, remove, or list tags for your resources:

<table>
<thead>
<tr>
<th>Task</th>
<th>AWS CLI</th>
<th>API Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add or overwrite one or more tags for a resource.</td>
<td><code>tag-resource</code></td>
<td><code>TagResource</code></td>
</tr>
</tbody>
</table>
### Working with Tags by Using the AppStream 2.0 API, an AWS SDK, or AWS CLI

<table>
<thead>
<tr>
<th>Task</th>
<th>AWS CLI</th>
<th>API Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove one or more tags for a resource.</td>
<td>untag-resource</td>
<td>UntagResource</td>
</tr>
<tr>
<td>List one or more tags for a resource.</td>
<td>list-tags-for-resource</td>
<td>ListTagsForResource</td>
</tr>
</tbody>
</table>

When you use the AppStream 2.0 API, an AWS SDK, or AWS CLI actions to add, edit, remove, or list tags for an AppStream 2.0 resource, specify the resource by using its Amazon Resource Name (ARN). An ARN uniquely identifies an AWS resource and uses the following general syntax.

```
arn:aws:appstream:region:account:resourceType/resourceName
```

- **region**
  - The AWS Region in which the resource was created (for example, us-east-1).
- **account**
  - The AWS account ID, with no hyphens (for example, 123456789012).
- **resourceType**
  - The type of resource. You can tag the following AppStream 2.0 resource types: image-builder, image, fleet, and stack.
- **resourceName**
  - The name of the resource.

For example, you can obtain the ARN for an AppStream 2.0 fleet by using the AWS CLI `describe-fleets` command. Copy the following command.

```
aws appstream describe-fleets
```

For an environment that contains a single fleet named `TestFleet`, the ARN for this resource would appear in the JSON output similar to the following.

```
"Arn": "arn:aws:appstream:us-east-1:123456789012:fleet/TestFleet"
```

After you obtain the ARN for this resource, you can add two tags by using the `tag-resource` command:

```
aws appstream tag-resource --resource arn:aws:appstream:us-east-1:123456789012:fleet/TestFleet --tags Environment=Test,Department=IT
```

The first tag, `Environment=Test`, indicates that the fleet is in a test environment. The second tag, `Department=IT`, indicates that the fleet is in the IT department.

You can use the following command to list the two tags that you added to the fleet.

```
aws appstream list-tags-for-resource --resource arn:aws:appstream:us-east-1:123456789012:fleet/TestFleet
```

For this example, the JSON output appears as follows:
{
    "Tags": {
        "Environment": "Test",
        "Department": "IT"
    }
}
Troubleshooting

If you encounter difficulties when working with Amazon AppStream 2.0, consult the following troubleshooting resources.

Contents
- General Troubleshooting (p. 75)
- Troubleshooting Image Builder (p. 76)
- Troubleshooting Active Directory Domain Join (p. 78)
- Troubleshooting Notification Codes (p. 81)

General Troubleshooting

The following are possible general issues you might have while using Amazon AppStream 2.0.

Issues
- SAML federation is not working. The user is not authorized to view AppStream 2.0 applications. (p. 75)
- After federating from an ADFS portal, my streaming session doesn't start. I am getting the error "Sorry connection went down". (p. 75)
- I get an invalid redirect URI error. (p. 75)
- My stack’s Home Folders aren’t working correctly. (p. 76)

SAML federation is not working. The user is not authorized to view AppStream 2.0 applications.

This may happen because the policy of the IAM role, which is assumed by the federated user who is accessing an AppStream 2.0 stack, does not include permissions to the stack ARN. Edit the role permissions to include the stack ARN. For more information, see Single Sign-on Access to AppStream 2.0 Using SAML 2.0 (p. 45) and Troubleshooting SAML 2.0 Federation with AWS in the IAM User Guide.

After federating from an ADFS portal, my streaming session doesn't start. I am getting the error "Sorry connection went down".

Set the claim rule’s Incoming Claim Type for the NameID SAML attribute to UPN and try the connection again.

I get an invalid redirect URI error.

This error happens due to a malformed or invalid AppStream 2.0 stack relay state URL. Make sure that the relay state configured in your federation setup is the same as the stack relay state available in stack details through the AppStream 2.0 console. If they are the same and the problem still persists, contact AWS Support. For more information, see Single Sign-on Access to AppStream 2.0 Using SAML 2.0 (p. 45).
My stack's Home Folders aren't working correctly.

Home Folder backup to the S3 bucket can have problems under the following scenarios:

- There is no internet connectivity from the streaming instance, or there is no access to the private Amazon S3 VPC endpoint, if applicable.
- Network bandwidth consumption is too high. For example, multiple large files being downloaded or streamed by the user while the service is trying to back up the Home Folder containing large files to Amazon S3.
- The administrator deleted the bucket created by the service.
- The administrator incorrectly edited the Amazon S3 permissions of the AmazonAppStreamServiceAccess service role.

For more information, see the Amazon Simple Storage Service Console User Guide and Amazon Simple Storage Service Developer Guide.

Troubleshooting Image Builder

The following are possible issues you might have while using Amazon AppStream 2.0 image builder.

Issues

- I cannot connect to the internet from my image builder. (p. 76)
- When I tried installing my application, I see an error that the operating system version is not supported. (p. 77)
- When I connect to my image builder, I see a login screen asking me to enter Ctrl+Alt+Delete to log in. However, my local machine intercepts the key strokes. (p. 77)
- When I switched between admin and test modes, I saw a request for a password. I don't know how to get a password. (p. 77)
- I get an error when I add my installed application. (p. 77)
- I accidentally quit a background service on the image builder and got disconnected. I am now unable to connect to that image builder. (p. 77)
- The application fails to launch in test mode. (p. 77)
- The application could not connect to a network resource in my VPC. (p. 78)
- I customized my image builder desktop, but my changes are not available when connecting to a session after launching a fleet from the image I created. (p. 78)
- My application is missing a command line parameter when launching. (p. 78)
- I am unable to use my image with a fleet after installing an antivirus application. (p. 78)
- My image creation failed. (p. 78)

I cannot connect to the internet from my image builder.

Image builders cannot communicate to the internet by default. To resolve this issue, launch your image builder in a VPC subnet that has internet access. You can enable internet access from your VPC subnet using a NAT gateway. Alternatively, you can configure an internet gateway in your VPC, and attach an Elastic IP address to your image builder. For more information, see Network Settings for Amazon AppStream 2.0 (p. 8).
When I tried installing my application, I see an error that the operating system version is not supported.

Only applications that can be installed on Windows Server 2012 R2 can be added to an AppStream 2.0 image. Check if your application is supported on Microsoft Windows Server 2012 R2.

When I connect to my image builder, I see a login screen asking me to enter Ctrl+Alt+Delete to log in. However, my local machine intercepts the key strokes.

Your client may intercept certain key combinations locally instead of sending them to the image builder session. To reliably send the Ctrl+Alt+Delete key combination to the image builder, choose Admin Commands, Send Ctrl+Alt+Delete. The Admin Commands menu is available on the top right corner of the image builder session toolbar.

When I switched between admin and test modes, I saw a request for a password. I don't know how to get a password.

AppStream 2.0 usually logs you into the user mode that you choose automatically. On some occasions, the switch may not happen automatically. If a password is requested, choose Admin Commands, Log me in. This sends a one-time password, securely, to your image builder and pastes it into the Password field.

I get an error when I add my installed application.

Check if your application type is supported. You can add applications of the types .exe, .lnk, and .bat.

Check if your application is installed under the C:\Users folder hierarchy. Any application installed under C:\Users is not supported. Select a different installation folder under C:\ when installing the application.

I accidentally quit a background service on the image builder and got disconnected. I am now unable to connect to that image builder.

Try stopping the image builder, restarting it and connecting to it again. If the problem persists, you need to launch (create) a new image builder. Do not stop any background services running on the image builder instance. Doing so may interrupt your image builder session or interfere with the image creation.

The application fails to launch in test mode.

Check if your application requires elevated user privileges or any special permissions that are usually available only to an administrator. The Image Builder Test mode has the same limited permissions on the image builder instance as your end users have on an AppStream 2.0 test fleet. If your applications require elevated permissions, they do not launch in the Image Builder Test mode.
The application could not connect to a network resource in my VPC.

Check if the image builder was launched in the correct VPC subnet. You may also need to verify that the route tables in your VPC are configured to allow a connection.

I customized my image builder desktop, but my changes are not available when connecting to a session after launching a fleet from the image I created.

Changes that are saved as part of a local user session, such as wallpaper, are not persisted when creating an image. To persist any local user session changes, add them to the local group policy on the image builder instance.

My application is missing a command line parameter when launching.

You can provide a command line parameter when using image builder to add an application to an image. If the launch parameters for the application do not change for each user, you can enter them while adding an application to the image in the image builder instance.

If the launch parameters are different for every launch, you can pass them programmatically when using the CreateStreamingURL API. Set the sessionContext and applicationID parameters in the API fields. The sessionContext is included as a command line option when launching the application.

If the launch parameters need to be computed on the fly, you can launch your application using a script. You can parse the sessionContext parameter within your script before launching your application with a computed parameter.

I am unable to use my image with a fleet after installing an antivirus application.

You can install any tools, including antivirus programs, on your AppStream 2.0 stack by using the image builder before creating an image. However, these programs should not block any network ports or stop any processes that are used by the AppStream 2.0 service. We recommend testing your application in Image Builder Test mode before creating an image and attempting to use it with a fleet.

My image creation failed.

Verify that you did not make any changes to AppStream 2.0 services before starting the image creation. Try creating your image again; if it fails, contact AWS Support. For more information, see AWS Support Center.

Troubleshooting Active Directory Domain Join

The following are possible issues you might have while setting up and using Active Directory with Amazon AppStream 2.0. For help troubleshooting notification codes, see Troubleshooting Notification Codes (p. 81).
Issues

- My image builders and fleet instances are stuck in "pending" status (p. 79)
- My users aren't able to log in with the SAML app (p. 79)
- My fleet instances work for one user but don't cycle correctly (p. 79)
- My user Group Policy Objects aren't applying successfully (p. 79)
- My AppStream 2.0 streaming instances aren't joining the Active Directory domain (p. 80)
- User login is taking a long time to complete on a domain-joined streaming session (p. 80)
- The changes I made in the image builder aren't reflected in end user streaming sessions (p. 81)
- Unable to access a domain resource in a domain joined streaming session but could access the resource from a domain-joined image builder (p. 81)

My image builders and fleet instances are stuck in "pending" status

Image builders and fleet instances can take up to 25 minutes to move into a ready state and become available. If your instances are taking longer than 25 minutes to become available, check your Active Directory to see if new computer objects were created in the organizational units (OUs). If there are new objects, the streaming instances will be available soon. If the objects aren't there, check the directory configuration details in your AppStream 2.0 Directory Config: Active Directory fully qualified domain name, service account username and password, and the OU distinguished name.

Image builder and fleet errors are available in the AppStream 2.0 console in the notifications tab when viewing the image builder or fleet details. Fleet errors are also available using the AppStream 2.0 API via the DescribeFleets operation, or the CLI command describe-fleets.

My users aren't able to log in with the SAML app

AppStream 2.0 relies on the SAML_Subject "NameID" attribute from your identity provider to populate the username field to log in your user. The username can either be formatted as "domain\username", or "user@domain.com". If using "domain\username" format, domain can either be the NetBIOS name or the fully qualified domain name. If using "user@domain.com" format, the UserPrincipalName attribute can be used. If you've verified your SAML_Subject attribute is configured correctly and the problem persists, contact AWS Support. For more information, see AWS Support Center.

My fleet instances work for one user but don't cycle correctly

Fleet instances are cycled after a user completes a session, ensuring that each user has a new instance. When the cycled fleet instance is brought online, it joins the domain using the computer name of the previous instance. To ensure that this operation happens successfully, the service account requires Change Password and Reset Password permissions on the organizational unit (OU) to which the computer object is joining. Check the service account permissions and try again. If the problem persists, contact AWS Support. For more information, see AWS Support Center.

My user Group Policy Objects aren't applying successfully

By default, computer objects apply computer-level policies based on the OU in which the computer object resides, while applying user-level policies based on the OU in which the user resides. If your user-level policies aren't being applied, you can do one of the following:
• Move the user-level policies to the OU in which the user Active Directory object resides
• Enable computer-level “loopback processing”, which applies the user-level policies in the computer object OU.

For more information, see Loopback processing of Group Policy at Microsoft Support.

My AppStream 2.0 streaming instances aren't joining the Active Directory domain

The Active Directory domain to use with AppStream 2.0 must be accessible through its fully qualified domain name (FQDN) via the VPC in which your streaming instances are launched.

To test that your domain is accessible

1. Launch an Amazon EC2 instance in the same VPC, subnet, and security groups that you use with AppStream 2.0.
2. Manually join the EC2 instance to your Active Directory domain using the FQDN (for example, yourdomain.exampleco.com) with the service account that you intend to use with AppStream 2.0. Use the following command in a Windows PowerShell console:

   ```
   netdom join computer /domain:FQDN /OU:path /ud:user /pd:password
   ```

   If this manual join fails, proceed to the next step.
3. If you cannot manually join to your domain, open a command prompt and verify that you can resolve the FQDN using the `nslookup` command. For example:

   ```
   nslookup yourdomain.exampleco.com
   ```

   Successful name resolution returns a valid IP address. If you are unable to resolve your FQDN, you may need to update your VPC DNS servers by using a DHCP option set for your domain. Then, come back to this step. For more information, see DHCP Options Sets in the Amazon VPC User Guide.
4. If the FQDN resolves, validate connectivity by using the following `telnet` command.

   ```
   telnet yourdomain.exampleco.com 389
   ```

   A successful connection shows a blank command prompt window without any connection errors. You may need to install the Telnet Client feature on your EC2 instance. For more information, see Install Telnet Client in the Microsoft documentation.

If you were not able to manually join the EC2 instance to your domain, but were successful in resolving the FQDN and testing connectivity with the Telnet Client, your VPC security groups may be preventing access. Active Directory requires certain network port settings. For more information, see Active Directory and Active Directory Domain Services Port Requirements in the Microsoft documentation.

User login is taking a long time to complete on a domain-joined streaming session

AppStream 2.0 performs a Windows login action after the end user provides their domain password, and then launches the application after successful authentication. The login and launch time is impacted by many variables, such as network contention to the domain controllers or time taken to apply group
The changes I made in the image builder aren't reflected in end user streaming sessions

User-specific settings in the image builder are saved in the specific user profile, and do not persist to the streaming instances. Examples include drive mounting, wallpaper changes, browser customizations, or Internet Explorer customizations. You need to manage these settings using the Microsoft Active Directory Group Policy settings that are applied to the OUs under which your streaming instances are created.

To quickly test whether your Group Policy settings are applied to the end user, connect to your image builder, login as a domain user and test the experience. For more information, see Group Policy for Beginners in the Microsoft documentation.

Unable to access a domain resource in a domain joined streaming session but could access the resource from a domain-joined image builder

Confirm that your fleet is created in the same VPC, subnets, and security groups that your image builder is in, and that your user has the appropriate permission to access and use the domain resource.

Troubleshooting Notification Codes

The following are notification codes and resolution steps for notifications you may see while setting up and using Amazon AppStream 2.0. These notifications can be found in the Notifications tab in the AppStream 2.0 console, after selecting an image builder or fleet. Fleet notifications can also be obtained using the AppStream 2.0 API operation DescribeFleets, or using the describe-fleets CLI command.
Active Directory Domain Join

The following are notification codes and resolution steps for codes you might see while setting up and using Active Directory with Amazon AppStream 2.0.

**DOMAIN_JOIN_ERROR_ACCESS_DENIED**

**Message:** Access is denied.

**Resolution:** The service account specified in the directory configuration does not have permission to create the computer object, or reuse an existing one. Validate the permissions and start the image builder or fleet. For more information, see Granting Permissions to Create and Manage Active Directory Computer Objects (p. 57).

**DOMAIN_JOIN_ERROR_LOGON_FAILURE**

**Message:** The username or password is incorrect.

**Resolution:** The service account specified in the directory configuration has an invalid username or password. Update the configuration and re-create the image builder or fleet that had the error.

**DOMAIN_JOIN_NERR_PASSWORD_EXPIRED**

**Message:** The password of this user has expired.

**Resolution:** The password for the service account specified in the AppStream 2.0 directory configuration has expired. Change the password for the service account in your Active Directory domain, then update the configuration, and re-create the image builder or fleet that had the error.

**DOMAIN_JOIN_ERROR_DS_MACHINE_ACCOUNT_QUOTA_EXCEEDED**

**Message:** Your computer could not be joined to the domain. You have exceeded the maximum number of computer accounts you are allowed to create in this domain. Contact your system administrator to have this limit reset or increased.

**Resolution:** The service account specified on the directory configuration does not have permission to create the computer object, or reuse an existing one. Validate the permissions and start the image builder or fleet. For more information, see Granting Permissions to Create and Manage Active Directory Computer Objects (p. 57).

**DOMAIN_JOIN_ERROR_INVALID_PARAMETER**

**Message:** A parameter is incorrect. This error is returned if the LpName parameter is NULL or the NameType parameter is specified as NetSetupUnknown or an unknown nametype.

**Resolution:** This error can occur when the distinguished name for the OU is incorrect. Validate the OU chosen. If you continue to see this error, contact AWS Support. For more information, see AWS Support Center.

**DOMAIN_JOIN_ERROR_MORE_DATA**

**Message:** More data is available.

**Resolution:** This error can occur when the distinguished name for the OU is incorrect. Validate the OU chosen. If you continue to see this error, contact AWS Support. For more information, see AWS Support Center.

**DOMAIN_JOIN_ERROR_NO_SUCH_DOMAIN**

**Message:** The specified domain either does not exist or could not be contacted.

**Resolution:** The streaming instance was unable to contact your Active Directory domain. Confirm your VPC, subnet, and security group settings to ensure network connectivity. For more information, see My AppStream 2.0 streaming instances aren’t joining the Active Directory domain (p. 80).
DOMAIN_JOIN_NERR_WORKSTATION_NOT_STARTED

**Message:** The Workstation service has not been started.

**Resolution:** An error occurred starting the Workstation service. Ensure that the service is enabled in your image. If you continue to see this error, contact AWS Support. For more information, see AWS Support Center.

DOMAIN_JOIN_ERROR_NOT_SUPPORTED

**Message:** The request is not supported. This error is returned if a remote computer was specified in the lpServer parameter and this call is not supported on the remote computer.

**Resolution:** Contact AWS Support for assistance. For more information, see AWS Support Center.

DOMAIN_JOIN_ERROR_FILE_NOT_FOUND

**Message:** The system cannot find the file specified.

**Resolution:** This error occurs when an invalid organizational unit (OU) distinguished name is provided. The distinguished name must start with `OU=`. Validate the OU distinguished name and try again. For more information, see Finding the Organizational Unit Distinguished Name (p. 58).
Amazon AppStream 2.0 Service Limits

By default, AWS limits the resources you can create and the number of users that can use the service. To request a limit increase, use the AppStream 2.0 Limits form.

The following table lists the limits for each AppStream 2.0 resource.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stacks</td>
<td>5</td>
</tr>
<tr>
<td>Fleets</td>
<td>5</td>
</tr>
<tr>
<td>Streaming instances</td>
<td>5 *</td>
</tr>
<tr>
<td>Images</td>
<td>5</td>
</tr>
<tr>
<td>Image builders</td>
<td>5 †</td>
</tr>
<tr>
<td>Users</td>
<td>5</td>
</tr>
</tbody>
</table>

* This is the total limit across all instance families. Certain instance families have additional limits. For the Graphics Desktop and Graphics Pro instance families (p. 26), the default limit is 0. For the Graphics Design instance family, the default limit is 2.

† This is the total limit across all instance families. Certain instance families have additional limits. For the Graphics Desktop and Graphics Pro instance families (p. 26), the default limit is 0. For the Graphics Design instance family, the default limit is 1.
# Document History for Amazon AppStream 2.0

The following table describes important changes to the documentation for Amazon AppStream 2.0.

- **API version:** 2016-12-01

<table>
<thead>
<tr>
<th>Feature</th>
<th>Change</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>New regions</td>
<td>Updated Setting Up SAML (p. 47) to add two new AppStream 2.0 regions: Singapore and Sydney.</td>
<td>January 24, 2018</td>
</tr>
<tr>
<td>Resource tagging</td>
<td>Created Tagging Your Amazon AppStream 2.0 Resources (p. 71) and made other updates.</td>
<td>December 15, 2017</td>
</tr>
<tr>
<td>Managed AppStream 2.0 agent updates</td>
<td>Created Amazon AppStream 2.0 Agent Version History (p. 24) and made other updates.</td>
<td>December 7, 2017</td>
</tr>
<tr>
<td>On-Demand fleets</td>
<td>Created Fleet Type (p. 25) and made other updates.</td>
<td>September 19, 2017</td>
</tr>
<tr>
<td>Instance families</td>
<td>Created AppStream 2.0 Instance Families (p. 26) and made other updates.</td>
<td>July 25, 2017</td>
</tr>
<tr>
<td>Active Directory</td>
<td>Created Using Active Directory Domains with AppStream 2.0 (p. 52) and made other updates.</td>
<td>July 24, 2017</td>
</tr>
<tr>
<td>User pool</td>
<td>Created Manage Access Using the AppStream 2.0 User Pool (p. 41) and made other updates.</td>
<td>June 15, 2017</td>
</tr>
<tr>
<td>Security groups</td>
<td>Created Security Groups (p. 11) and made other updates.</td>
<td>May 26, 2017</td>
</tr>
<tr>
<td>Home Folders</td>
<td>Created Home Folders and VPC Endpoints (p. 12) and made other updates.</td>
<td>May 18, 2017</td>
</tr>
<tr>
<td>Default internet access</td>
<td>Created Network Settings for Amazon AppStream 2.0 (p. 8) and made other updates.</td>
<td>April 21, 2017</td>
</tr>
<tr>
<td>Fleet automatic scaling</td>
<td>Created Fleet Auto Scaling for Amazon AppStream 2.0 (p. 30) and made other updates.</td>
<td>March 23, 2017</td>
</tr>
<tr>
<td>Fleet management</td>
<td>Created Amazon AppStream 2.0 Stacks and Fleets (p. 25) and made other updates.</td>
<td>February 22, 2017</td>
</tr>
<tr>
<td>SAML 2.0 support</td>
<td>Created Single Sign-on Access to AppStream 2.0 Using SAML 2.0 (p. 45) and made other updates.</td>
<td>February 15, 2017</td>
</tr>
<tr>
<td>Image builders</td>
<td>Created AppStream 2.0 Image Builders (p. 16) and made other updates.</td>
<td>January 19, 2017</td>
</tr>
<tr>
<td>Initial release</td>
<td>Created this guide.</td>
<td>December 01, 2016</td>
</tr>
</tbody>
</table>