

Amazon AppStream 2.0: Getting Started Guide

Build an Amazon AppStream 2.0 environment to stream SAP GUI to your users

April 2023

<https://aws.amazon.com/appstream2/>

Welcome

This guide describes how to deploy and stream SAP GUI for your company (in this example, DemoCo) by using [Amazon AppStream 2.0](#), a fully managed, secure application streaming service that runs in the AWS Cloud.

What you'll accomplish:

- Provision an Amazon virtual private cloud ([Amazon VPC](#)) to provide a logically isolated virtual network infrastructure within the AWS Cloud. Your AppStream 2.0 resources will use this environment.
- Use the [AWS Management Console](#) to perform the basic administrative tasks required to build an AppStream 2.0 environment. Specifically, you'll:
 1. Install and configure SAP GUI for streaming using an image builder.
 2. Provision a fleet of instances to stream SAP GUI. The fleet will use the Standard instance type and adhere to scaling policies to match the number of users that you want to be able to stream concurrently.
 3. Provision a stack to create a web portal from which users can stream your application.
 4. Configure persistent storage that users can access across application streaming sessions.
 5. Create a user pool to manage users who access your streaming applications.

What you need before starting:

- **An AWS account:** You need an AWS account to use AppStream 2.0 and other AWS services. For information about how to sign up for and activate an AWS account, see *Appendix A*.
- **A current email address:** During the user configuration process for your AppStream 2.0 environment, AWS sends you two emails. You must use these emails to complete the process.

- **Skill level:** You do not need prior experience with AWS to complete these exercises. A basic understanding of desktop computing is helpful but not required.
- **SAP Product License:** SAP GUI for Windows does not require an own license, because SAP GUI for Windows is not an SAP Product. SAP GUI for Windows is a software component which is part of various SAP Products (like SAP S/4HANA) and it simply inherits the license and terms of the End-user Agreement from these products. This means you can use SAP GUI for Windows as long as you hold a valid license for an SAP Product that contains SAP GUI for Windows.
- **End user client recommendations:** To use SAP GUI delivered through AppStream, your user would need a modern HTML5-capable web browser. Please refer to requirements in [Web Browser for AppStream 2.0](#) for the list of supported browsers.
- **End user network recommendations:** AppStream 2.0 uses [NICE DCV](#), an adaptive streaming protocol to deliver an interactive streaming session to users. The protocol encodes pixels on a remote host, securely transmits them over the network, and renders them on a client device. It also accepts user keyboard and mouse input, enables file transfer between client and remote host, and provides clipboard support to provide an interactive experience for a user when using streamed applications. While the streaming protocol adapts to changes on the screen and only transmits pixels when required, it will use the available bandwidth on the network. Also, since the streaming session is interactive, and the application on the remote host needs to respond to user inputs on a client device, the round-trip latency will influence the responsiveness that a user will experience.

The amount of bandwidth used when transmitting pixels is proportional to the changes on the screen and the resolution of the display monitor(s) used by the client device. The changes on the screen and the resolution are determined by the type of application (3D versus business application) and usage pattern (switching between windows and menus quickly). A 3D application may require a

high-resolution monitor and trigger large changes to the screen when a user is interacting with complex hi-fidelity models. To transmit these changes on the screen quickly and provide a responsive experience to the user, the protocol will use a large amount of bandwidth momentarily. On the other hand, a business application may only involve text input. While changes to text on screen can be transmitted with very small amount of bandwidth, switching quickly between windows or menus within even a text-based application will result in large changes to the screen and hence drive momentary increases in bandwidth used. The round-trip network latency influences the responsiveness that a user perceives when entering input and viewing changes on the screen. While other factors such as quality of network, client device performance, and remote host instance selection can also influence the responsiveness, latency should be considered as one of the primary factors. In general, lower latency connections will deliver more responsive and performant streaming experience. Below are the recommendations for different use-cases.

Use case	Recommended bandwidth available per user	Recommended maximum roundtrip latency
Business applications like SAP GUI	1-2 mbps	< 150 ms
3D applications – Streaming with low fidelity datasets or maps with 2K monitors	5-6 mbps	< 100 ms
3D applications – Streaming high fidelity datasets or maps with 4K monitors	10-12 mbps	< 50 ms

Contents

Welcome	2
Step 1. Sign in to the AWS Management Console and select an AWS Region	7
Step 2: Create network resources	7
AppStream VPC requirements	8
Create a VPC by using the VPC Wizard	8
Step 3: Create an AppStream 2.0 image builder.....	11
Deploy an image builder instance to install applications	11
Step 4: Connect to the image builder and install applications	17
Connect to the image builder instance.....	17
Download application installation files to your image builder instance	18
Install the applications	19
Step 5: Configure applications	22
Before you create an image with SAP GUI for Windows, you must disable Internet Explorer ESC and configure SAP Logon	22
Disable Internet Explorer Enhanced Security Configuration	22
SAP Logon Configuration	23
Installing additional software for use along with SAP GUI for Windows.....	24
Step 6: Use Image Assistant to create an AppStream 2.0 image.....	24
Create your AppStream 2.0 application catalog.....	25
Test your applications by using a local user account.....	26
Optimize the launch performance of your applications	27
Configure the image.....	27
Step 6: Provision a fleet	29
Provide fleet details	29
Configure the fleet	30
Choose an image.....	32
Configure the network.....	33
Step 7: Create an AppStream 2.0 stack and a streaming URL	34
Provide stack details and associate the stack with a fleet.....	34
Enable persistent storage and user settings for the stack.....	35
Create a streaming URL	37
Step 8: Manage user access with an AppStream 2.0 user pool.....	39

Create a user	40
Assign a stack to the user.....	40
Step 9: Test the end user authentication and application streaming experience.....	41
Step 10: Take the next step with AppStream 2.0	43
Appendix A: Create and activate an AWS account	46
Create your AWS account.....	46
On the Payment Information page , type the requested information associated with your payment method. If the address for your payment method is the same as the address you provided for your account, choose Verify and Add	47
Otherwise, choose Use a new address, type the billing address for your payment method, and then choose Verify and Add	47
Verify your phone number.....	47
Choose an AWS Support plan	47
Watch for three AWS account confirmation emails	48
Appendix B. Clean up your AppStream 2.0 resources.....	50
Stop and delete your image builder	50
Revoke stack permissions for users in the user pool.....	50
Disassociate your fleets from your stack and delete your stack	50
Stop and delete your fleet	50

Step 1. Sign in to the AWS Management Console and select an AWS Region

If you do not have an AWS account, you must first complete the steps in *Appendix A*.

1. Sign in to the AppStream 2.0 console at <http://console.aws.amazon.com/appstream2>.
2. Type your email address or your AWS account ID, and choose **Next**.
3. Type your AWS account password, and choose **Sign In**.
4. In the menu in the upper right corner of the console, select the AWS Region for your environment.



Step 2: Create network resources

Amazon AppStream 2.0 Fleet is deployed in Amazon Virtual Private Cloud (VPC) and need access to the SAP Servers. AppStream 2.0 can be deployed in an existing VPC where SAP servers are deployed or in to a new VPC.

In this section, you will create a separate Amazon virtual private cloud (VPC) and other network resources required for your AppStream 2.0 environment.

The topology of your “DemoCo” VPC should look similar to the following diagram:

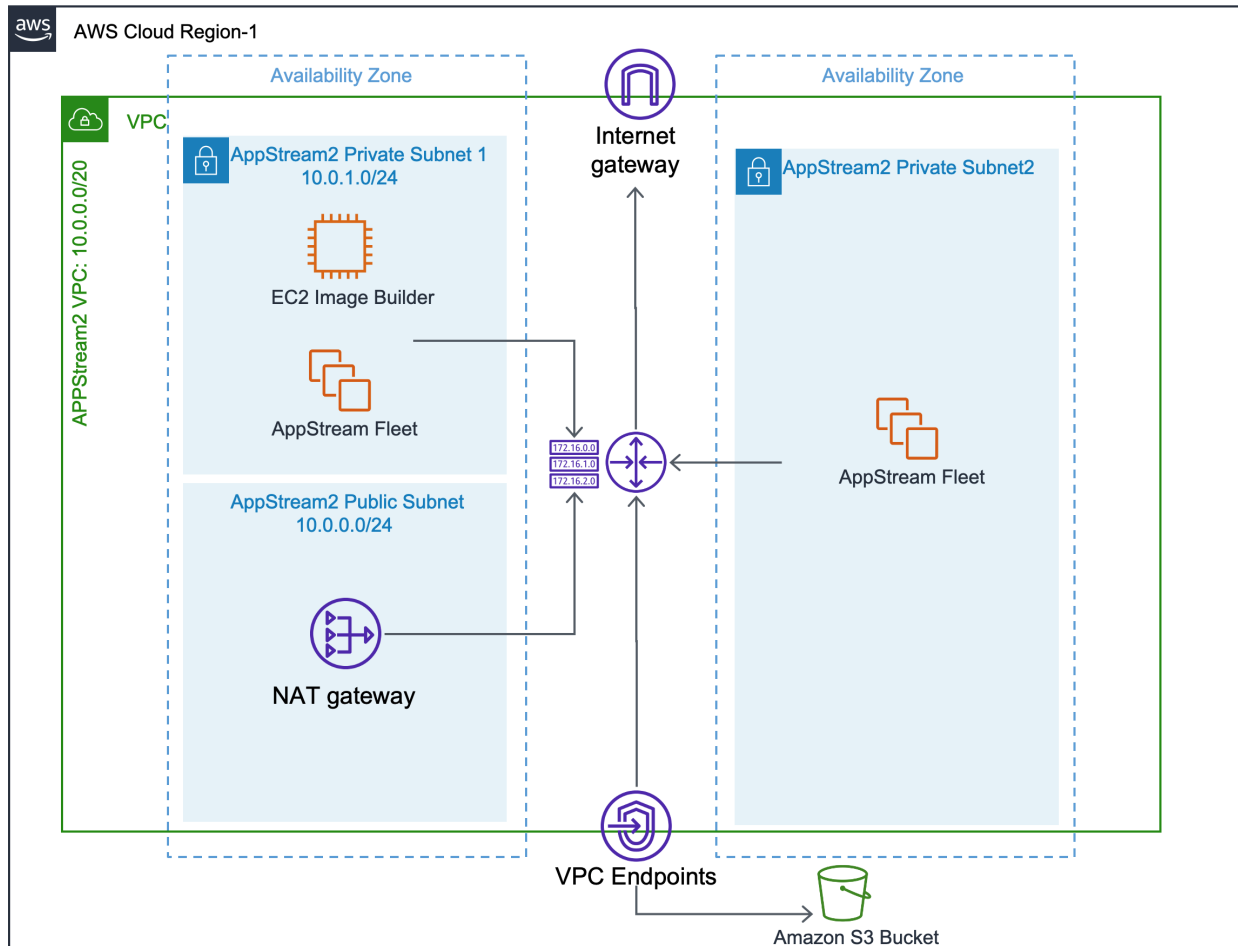


Figure 1: DemoCo VPC architecture.

Note: The CIDR block assignments for the private subnets might be reversed depending on the availability zones used by the VPC wizard.

AppStream VPC requirements

At a minimum, AppStream 2.0 requires a VPC that includes one public subnet and two private subnets. A public subnet has direct access to the internet through an internet gateway. A private subnet requires a Network Address Translation (NAT) gateway or NAT instance to access the internet.

Create a VPC by using the VPC Wizard

The easiest way to start building your VPC environment is to use the VPC Wizard. The wizard guides you through the process of creating a public subnet, private subnet, NAT gateway, and internet gateway, with the correct route table configurations.

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **VPC Dashboard, Create VPC**.
3. **In VPC settings, for Resources to create** choose **VPC and more**.
4. Type the following information and then choose **Create VPC**.

Option	Value
VPC Name (Name Tag)	<i>AppStream2 VPC (Select 'Auto-generate')</i>
IPv4 CIDR block	<i>Select Appropriate CIDR range e.g. 10.0.0.0/20</i>
IPv6 CIDR block	Accept the default value: <i>No IPv6 CIDR Block</i>
Tenancy	Accept the default value: <i>Default</i>
Number of Availability Zones (AZs)	2
Customize AZs	Accept the default value for <i>First Availability Zone (us-east-1a)</i> and <i>Second Availability Zone (us-east-1b)</i>
Number of public subnets	2
Number of private subnets	2
Public subnet's CIDR block in us-east-1a	Accept the default value: <i>10.0.0.0/24</i>
Public subnet's CIDR block in us-east-1b	<i>Adjust the value to 10.0.3.0/24</i>
Private subnet CIDR block in us-east-1a	<i>Adjust the value to 10.0.1.0/24</i>
Private subnet CIDR block in us-east-1b	<i>Adjust the value to 10.0.2.0/24</i>
NAT gateways	<i>Select: In 1 AZ</i>
VPC endpoints	select: <i>S3 Gateway</i>

Option	Value
DNS Options	Accept the default values: select both the check boxes – "Enable DNS hostnames" and "Enable DNS hostnames"

Note: The VPC names and subnet names are for identification purposes only. You can use different names.

The screenshot displays the 'Create VPC' configuration page in the AWS console. The left pane shows the 'VPC settings' section with various options:

- Resources to create:** 'VPC and more' is selected.
- Name tag auto-generation:** 'AppStream2' is entered.
- IPv4 CIDR block:** '10.0.0/16' is selected.
- IPv6 CIDR block:** 'No IPv6 CIDR block' is selected.
- Tenancy:** 'Default' is selected.
- Number of Availability Zones (AZs):** '2' is selected.
- Number of public subnets:** '2' is selected.
- Number of private subnets:** '4' is selected.
- NAT gateways:** '1 per AZ' is selected.
- VPC endpoints:** 'S3 Gateway' is selected.
- DNS options:** 'Enable DNS hostnames' and 'Enable DNS resolution' are checked.

The right pane shows a 'Preview' diagram of the VPC architecture, including subnets, route tables, and network connections.

Figure 2: Creating a VPC.

5. After a few minutes, when a message in the VPC dashboard notifies you that the VPC is created, choose **OK**.
6. You have now successfully created your network resources.

Before proceeding to creating your AppStream 2.0 environment, you have to make your SAP server environment accessible through the Amazon VPC that you just created. This can be accomplished by enabling [VPC Peering](#) between the VPC's where SAP servers are deployed and the APP Stream VPC. To learn more about Amazon VPC configuration for SAP, refer the blog post series from [here](#).

Step 3: Create an AppStream 2.0 image builder

AppStream 2.0 uses EC2 instances to stream applications. You launch instances from base images, called *image builders*, which AppStream 2.0 provides. To create your own custom image, you connect to an image builder instance, install and configure your applications for streaming, and then create your image by creating a snapshot of the image builder instance.

To install and configure applications to stream to your users, you must create an image builder instance as described in the following procedure.

Deploy an image builder instance to install applications

1. Open the AppStream 2.0 console at <https://console.aws.amazon.com/appstream2>.
2. If you have not previously configured any AppStream 2.0 settings, the following page appears:

The screenshot shows the AWS console interface for Amazon AppStream 2.0. On the left is a navigation sidebar with options like Stacks, Fleets, Images, User pool, Directory Configs, Usage Reports, Applications, and Quick Links. The main content area features a dark header with the title "Amazon AppStream 2.0" and the subtitle "Stream desktop applications securely to any web browser". Below this is a "How it works" section containing a video player with a play button and the text "Introducing Amazon AppStream 2.0". To the right of the main content are three informational panels: "Get Started" with a "Try streaming now" and "Get Started" button, "Pricing" with a description of the pay-per-use model and a "Learn more about AppStream 2.0 Pricing" link, and "Documentation" with links for "What is AppStream 2.0?", "Get Started with Amazon AppStream 2.0", and "Guidance for AppStream 2.0 Users".

Figure 3: The AppStream 2.0 first experience page.

Note: If the AppStream 2.0 navigation page appears instead, skip to step 5.

3. Choose **Get started**.
4. In the lower right corner of the page, choose **Skip** (this guide walks you through a different process for getting started with AppStream 2.0).

Quick Links Info

Set up an AppStream 2.0 stack to stream applications

Create an AppStream 2.0 stack to stream applications. You can start with sample applications and add your own applications later.

[Set up with sample apps](#)

Set up an AppStream 2.0 image with your own applications

Create an AppStream 2.0 image to stream your own applications.

[Custom set up](#)

Learn more about AppStream 2.0

Walk through the resources you will create to stream applications using AppStream 2.0.

[Learn more](#)

[Skip](#)

Figure 4: AppStream 2.0 getting started options.

5. In the navigation pane, choose **Images, Image Builder, Launch Image Builder**.
6. In the **Step 1: Choose Image** window, in the list of images, select the image builder with the name *AppStream-WinServerYYYY-mm-dd-yyyy*, where *mm-dd-yyyy* represents the most recent date. Base images include the latest updates to Microsoft Windows and the AppStream 2.0 agent software. You use this base image to create a custom image that includes your own applications.

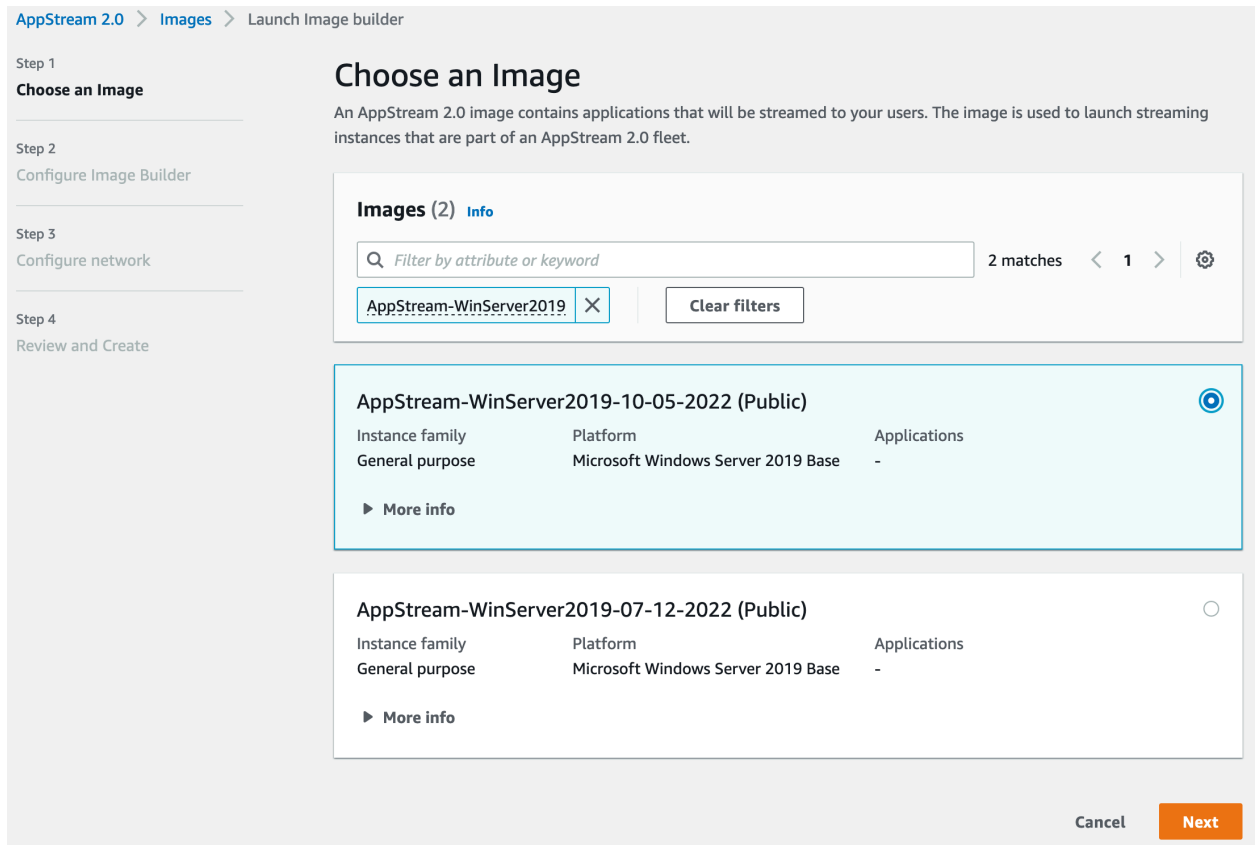


Figure 5: Choosing an image builder.

7. At the bottom of the page, choose **Next**.
8. In **Step 2: Configure Image Builder**, the following image builder configuration options are displayed:

AppStream 2.0 > Images > Launch Image builder

Step 1
Choose an Image

Step 2
Configure Image Builder

Step 3
Configure network

Step 4
Review and Create

Configure Image Builder

Image builder details

Name *
Enter the name of your AppStream 2.0 image builder.

Allowed characters: a-z, A-Z, 0-9, _ - (hyphen)

Display name

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)

You can add up to 40 more tags.

Choose instance type * (21) [Info](#) General Purpose ▾

Select an instance type that matches your applications' requirements.

	Family ▾	Type ▾	vCPUs ▲	Memory (GiB) ▾
<input type="radio"/>	General Purpose	stream.standard.small	1	2
<input type="radio"/>	General Purpose	stream.standard.medium	2	4
<input type="radio"/>	General Purpose	stream.standard.large	2	8
<input type="radio"/>	General Purpose	stream.standard.xlarge	4	16

Figure 6: Image builder configuration options.

9. Type the following information and then choose **Review**.

Configure image builder fields

Option	Value
Name	Provide a unique name identifier for the image builder, such as <i>DemoCo_Image_v1_mmdyyy</i> , using any of the following characters: a-Z,0-9,-,_,.
Display Name	Provide an optional name, such as <i>DemoCo Image v1 April 2018</i> , to be displayed in the console for easier reference and readability.

Option	Value
Choose Instance Type	Accept the default value General Purpose .
Instance Family	Select instance type based on your usage and requirements. Refer to SAP Note 26417
Choose Next to continue to Step 3: Configure Network and then type the following information	
Default Internet Access	Uncheck "Enable default internet access" as we will use access enabled as part of VPC setup
VPC	Select the VPC created as part of VPC setup.
Subnet	Select the subnet. The best practice is to select Private Subnet
Security group(s)	Select Appropriate Security group
Active Directory Domain (Optional)	Select this if you want to integrate with Active Directory Domain
Choose Next to continue to Step 4: Review and Create	

10. Choose **Review**, and confirm the details for the image builder. To change the configuration for any section, choose **Edit** and make your changes.

11. After you finish reviewing the configuration details, choose **Launch Image Builder**.

12. Wait for the status of Image builder to change to **Running**.

13. The image builder is now ready to use and you can create a custom image.

Note: Charges accrue for an image builder instance while it is running, even if no user is actively connected. You can stop or delete the image builder at any time.

User fees are not incurred for administrators connecting to and using image builders to create images. For more information, see [AppStream 2.0 Pricing](#).

Step 4: Connect to the image builder and install applications

Now that you have provisioned an image builder, you can use it to install and configure the applications to stream to users. First, you must establish a remote connection to the instance to install and configure your applications.

Connect to the image builder instance

1. Open the AppStream 2.0 console at <https://console.aws.amazon.com/appstream2>.
2. In the navigation pane, choose **Images, Image Builder**.
3. Select the image builder instance that you created earlier (*DemoCo_Image_v1_mmdyyy*). Verify that its status is **Running** and choose **Connect**.

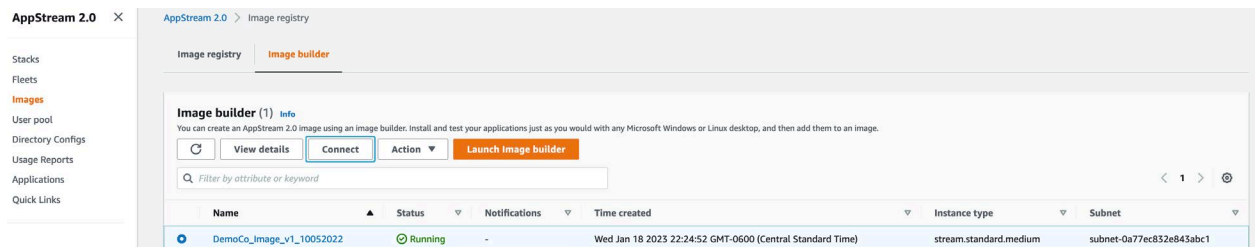


Figure 7: Choosing an image builder instance.

Note: If the status is **Stopped**, select the instance, and choose **Actions, Start**. Click the **Refresh** icon periodically to update the instance list until the status is **Running**.

4. The new browser tab opens, displaying options for logging into the image builder instance. Choose **Local User, Administrator**.

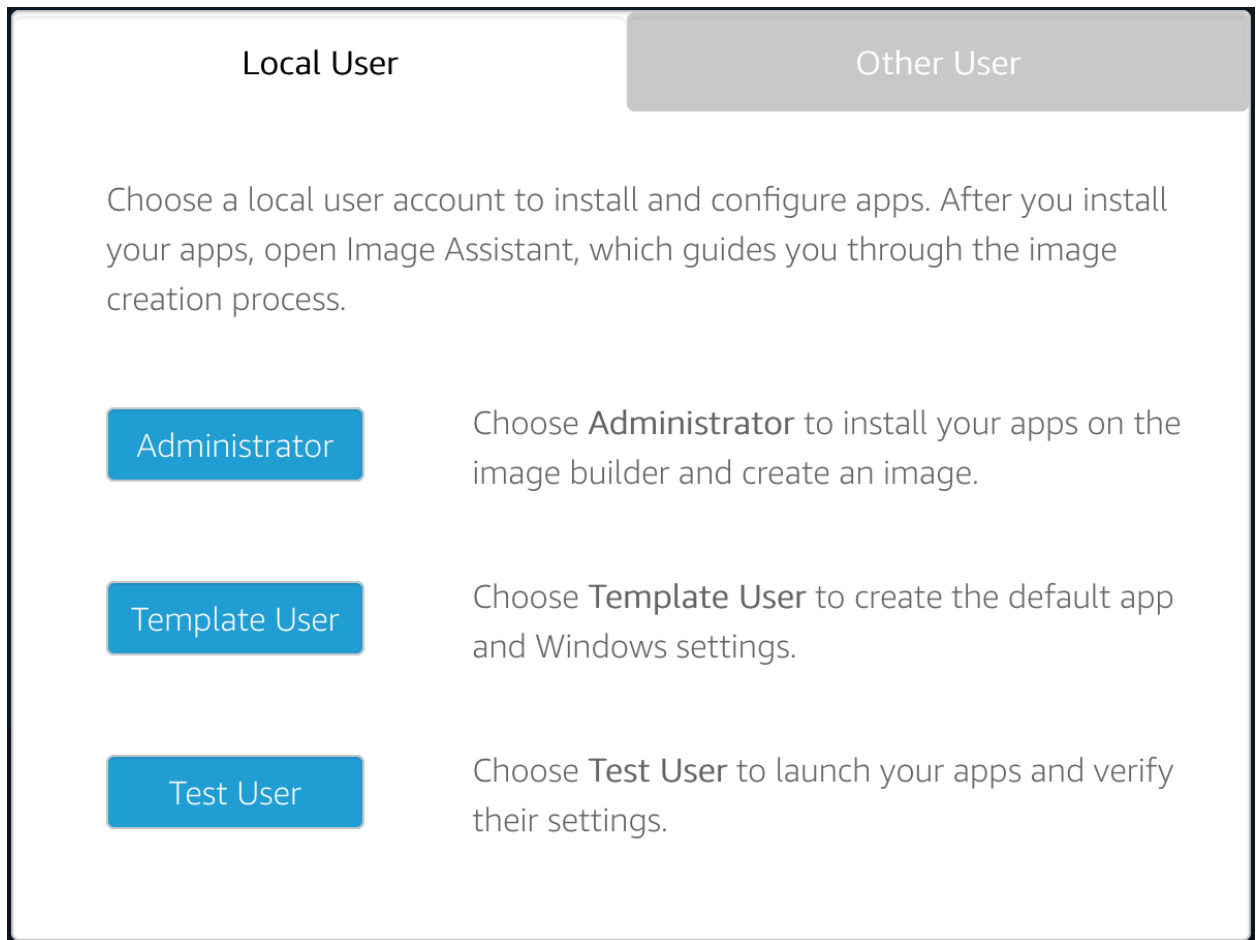


Figure 8: The image builder instance login options window.

Note: If a new browser tab does not open, configure your browser to allow pop-ups from <https://console.aws.amazon.com/>.

5. After a few moments, you are connected to the image builder instance with administrator rights.

Download application installation files to your image builder instance

In this exercise, you use a browser to download and install the following two applications:

- SAP GUI software - <http://support.sap.com/swdc>
1. Launch Mozilla Firefox by opening the Firefox icon on the image builder instance desktop.

- If prompted by the Firefox Import Wizard, choose **Don't import anything, Next**.
 - If prompted to sign in to Firefox, choose **Skip this step**.
 - If Firefox displays a notification that the application is out of date, close the notification.
 - If prompted to set a default browser, choose **Use Firefox as my default browser**.
 - If prompted by Windows to decide how to open HTTP links, choose the Firefox icon.
2. Open Firefox and navigate to the following URL to download the full SAP GUI installer: <http://support.sap.com/swdc>
 3. Login using SAP account and download the latest version of SAP GUI for Windows.
 4. When prompted, choose **Save File**. The SAP GUI will be downloaded to the **Downloads** folder.
 5. Close Firefox, and if prompted, choose **Close tabs**.

Install the applications

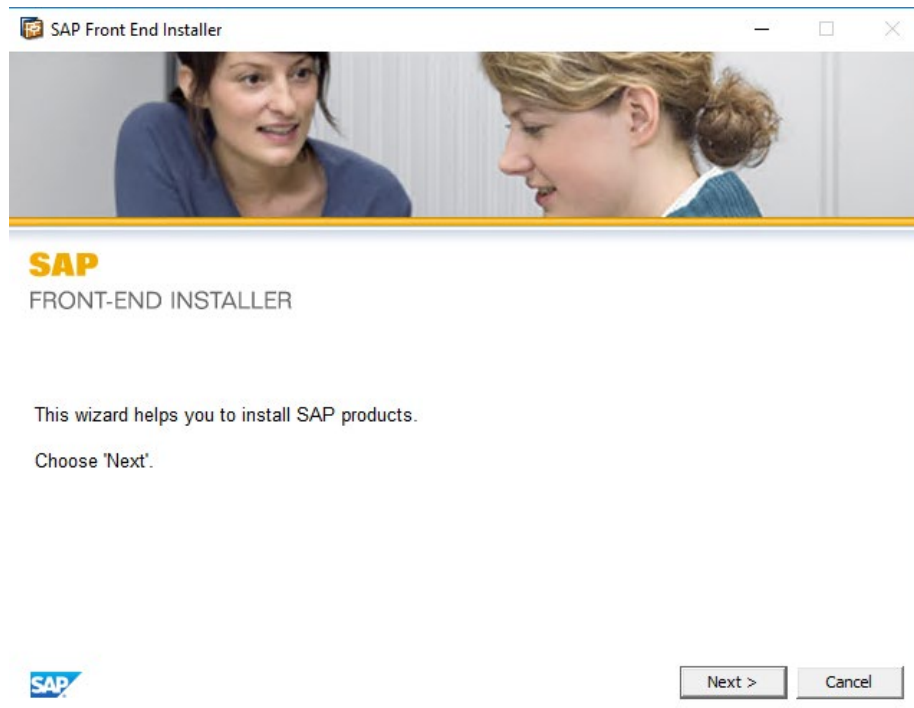
1. Open File Explorer by clicking the folder icon on the taskbar.
2. Select the **Downloads** folder to open it. The SAP installation files are downloaded to this folder.
3. Extract the installation package to a local folder.
4. Navigate to the following path to identify the installation guide for SAP GUI on Windows. Make user you don't extract the installation files to **Temporary Files**. You will have to restart the image builder before starting the installation. **Temporary Files** will be cleared when image builder is restarted.

<extractedfolder>\ <Parent Folder>\PRES1\DOCU\

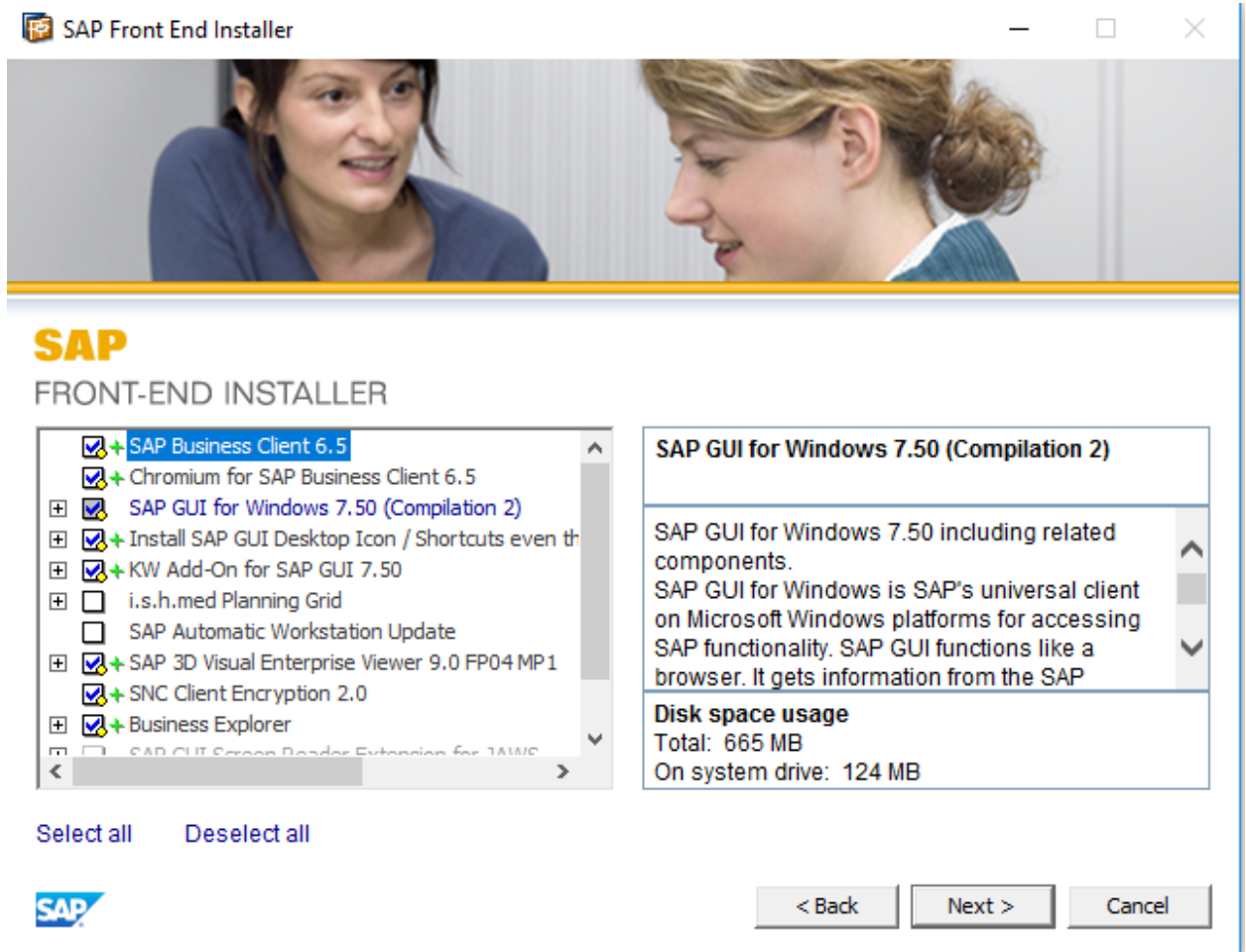
5. To install SAP GUI, navigate to the following folder and launch **SAPGUI<XX>Setup.exe**. If you want to include all the front end components along with the SAP GUI, launch **SetupAll.exe**.

<extractedfolder>\ <Parent Folder>\PRES1\GUI\Windows\WinXX and launch the Setup application **SAPGUI<XX>Setup.exe**

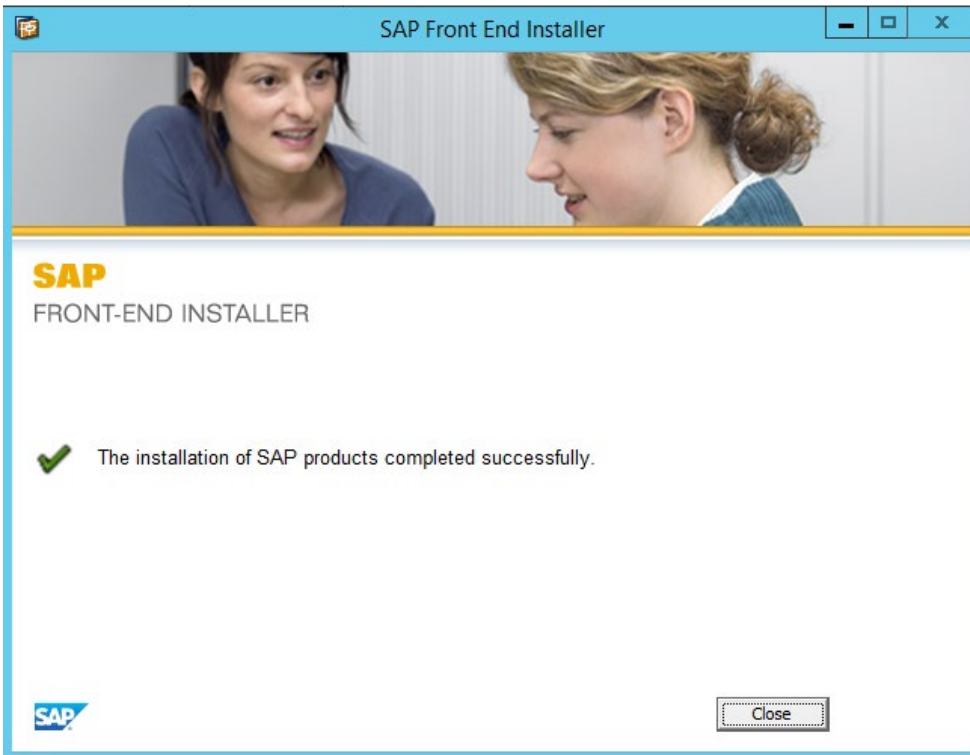
6. On launch, in case the installer recommends you to reboot the image builder before continuing further. Please proceed and restart the image builder either by initiating restart from the instance or by choosing **Actions -> Stop** and **Actions -> Start** from the AWS management console.



7. Once the image builder is in running status again, connect to the image builder as **Administrator**.
8. Relaunch the SAP GUI installer **SAPGUI<XX>Setup.exe** or **SetupAll.exe**.
9. Select all the required components and proceed with the installation by choosing **Next**.



10. The installation wizard might prompt you to enter or change information to customize the installation of the selected products, such as the installation folder. **Choose defaults** and complete the installation.
11. Once SAP GUI is fully installed, you will see a success message as shown below.



12. You have now installed SAP GUI. The next step is to configure SAP Logon and Internet Explorer applications before creating the image with SAP GUI.

Step 5: Configure applications

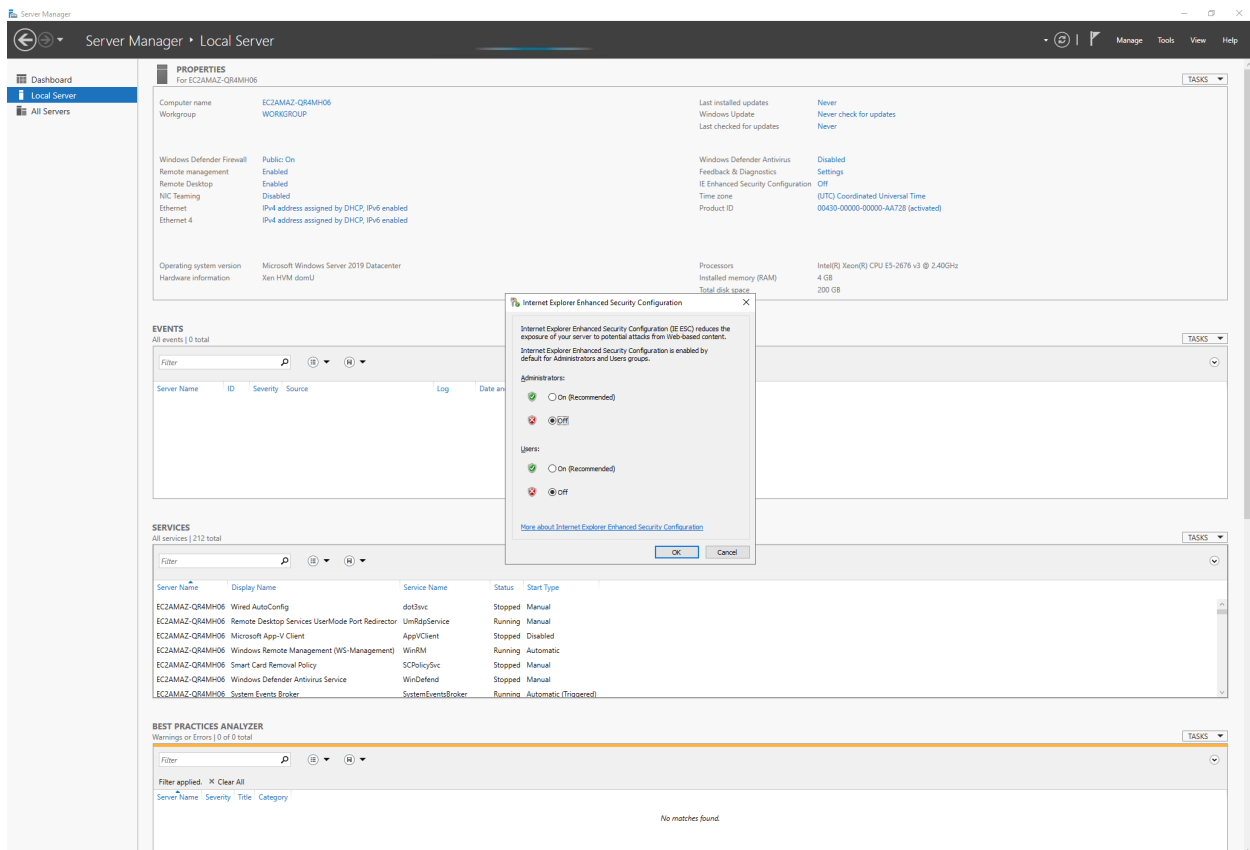
Before you create an image with SAP GUI for Windows, you must disable Internet Explorer ESC and configure SAP Logon

Disable Internet Explorer Enhanced Security Configuration

When you launch the SAP GUI after installation, you will see a warning message. This behavior is due to the Internet Explorer Enhanced Security Configuration, a security setting that decreases your exposure to potential attacks that can occur through web content and application scripts. You can safely disable this setting to enable SAP GUI. To disable this feature for AppStream 2.0 users, do the following.


1. Connect to your image builder as Administrator.

2. Open ServerManager and choose the Windows **Start** button, and then choose **Server Manager**.
3. In the left navigation pane, choose Local Server.
4. In the right properties pane, choose the **On** link next to **IE Enhanced Security Configuration**.
5. In the **Internet Explorer Enhanced Security Configuration** dialog box, choose the **Off** option for both **Administrators and Users**, and then choose **OK**.



SAP Logon Configuration

Refer to the **SAP UI Landscape Configuration Guide**, available within the extracted installation package, for detailed explanation of different options available to configure SAP Logon client to access SAP servers. The steps detailed below are based on SAP OSS Note - **2075150 - SAP Logon (Pad) 740: New format of configuration files as of SAP GUI for Windows 7.40**.

1. While being logged in as **Administrator**, in your image builder, Launch SAP GUI and choose **Create**  and select **Connection**, for adding a new SAP server entry to the client. On the prompted dialog, provide the server details - **IP Address/Network Alias, SAP System No and SAP System ID**. Note that the SAP server must be accessible through the VPC you created earlier. This document does not cover the installation and configuration of SAP backend server.
2. Move the SAP GUI configuration files, **SAPUILandscape.xml and SAPUILandscapeGlobal.xml** to a location under C:\ such that it is accessible to all AppStream 2.0 users. In this case, copy all the files from **C:\Users\ImageBuilderAdmin\AppData\Roaming\SAP\Common** to **C:\Program File (x86)\SAP\FrontEnd\SAPgui**

Note: Before copying the files please enable option to show hidden files, else the AppData folder will not be visible

3. Create a Windows system environment variable as follows.

Env variable name: **SAPLOGON_LSXML_FILE**

Env variable value: **C:\Program File**

(x86)\SAP\FrontEnd\SAPgui\SAPUILandscape.xml

Installing additional software for use along with SAP GUI for Windows

Additional software may be required to use SAP GUI for Windows (like Microsoft Office or a Single-Sign-On product) based on the SAP transactions that you want to execute. Please install the necessary supporting applications before proceeding to the next step.

Step 6: Use Image Assistant to create an AppStream 2.0 image

At this point, you have launched an image builder instance, installed and configured SAP GUI. Now you'll prepare the application for streaming, optimize it for streaming performance, and create your image.

In this section, you'll do the following:

- Create an application catalog by using Image Assistant.
- Test SAP GUI by using a local user account that has the same permissions that end users will have in their streaming sessions.
- Optimize the application's launch performance.
- Configure the image.
- Finish creating the image.

Create your AppStream 2.0 application catalog

The process of creating an AppStream 2.0 application catalog includes specifying the name, display name, executable file to launch, and icon to display for each application that you plan to stream.

1. From the image builder desktop, open Image Assistant.
1. In the **Add Applications to Image** dialog box, on the **Add Apps** tab, choose **Add Application**.

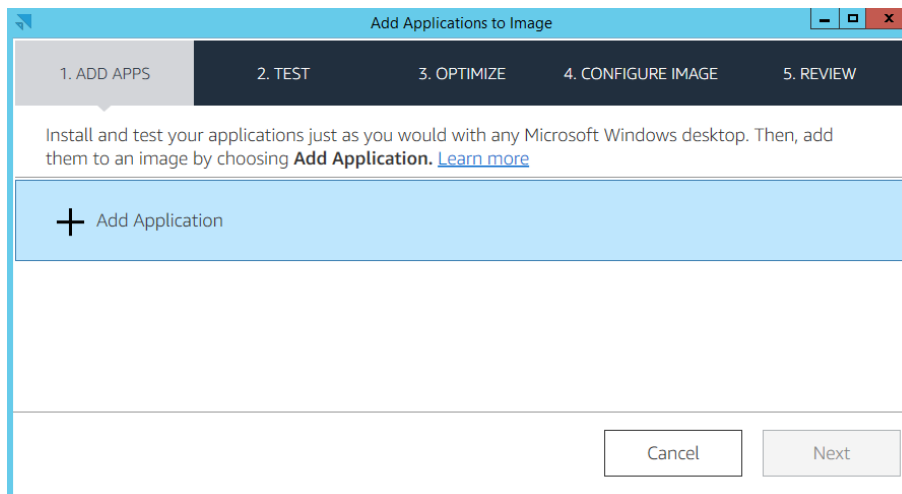


Figure 9: The Add Applications to Image dialog box in Image Assistant.

2. Navigate to the location of the SAP GUI executable file (C:\Program Files (x86)\SAP\FrontEnd\SAPgui), select the SAP GUI executable file, saplogon.exe, and then choose **Open**.
3. Review the application properties displayed and choose **Save**.

Property	Description
Name	A unique identifier for the application
Display name	The name of the application that is displayed to end users. You can change the name to one that is meaningful for your end users.
Launch Path	The location of your application executable
Icon Path	The location of your application icon. Optionally, you can change the default icon to a high-resolution PNG file.
Launch Parameters	Command line arguments that can be passed to the application at launch.
Working Directory	Leave this blank.

4. SAP GUI is now added to the application catalog that will be displayed to the user.

Test your applications by using a local user account

An image builder includes a test user account that enables you to test your applications by using the same policies and permissions as your users. Follow these steps to confirm that your applications open correctly.

1. In the **Test** tab, choose **Switch User, Test User**.

You are now logged into the same Windows Server instance as a local user who has regular (non-administrative) user rights.

2. Open Image Assistant. In **Test Apps**, the SAP GUI application will be displayed.
3. Choose the application to launch SAP GUI. Connect to the SAP backend and test the transactions that you are interested in.
4. Once you have completed the testing, Choose **Switch User** using the **Admin Commands** menu available on the top right corner of the image builder web toolbar.
5. On the **Local User** tab, choose **Administrator**. You will be connected back into the image builder as Administrator.
6. On the Image Assistant **Test** tab, choose **Next**.

Optimize the launch performance of your applications

During this step, Image Assistant opens your applications one after another, identifies their launch dependencies, and performs optimizations to ensure that applications launch quickly.

1. On the **Optimize** tab, choose **SAP GUI, Launch**.
2. Wait for SAP GUI to completely start, as prompted by a message dialog shown by Image Assistant.
3. After you complete the first run experience for the application and verify that it functions as expected, choose **Continue**.

Configure the image

1. On the **Configure Image** tab, type the following information.

Option	Value
Name	<p>The unique name identifier for the image, such as <i>SAPGUI-Demo-Image</i>, using any of the following characters: a-Z,0-9,-,_,.</p> <p>Note: The name cannot begin with "Amazon," "AWS," or "AppStream."</p>

Option	Value
Display Name	A user-friendly name to display in the console, such as <i>SAP GUI Demo Image</i>
Description	An optional description for the image: for example, <i>Image containing the SAP GUI desktop application. Created on DD-MM-YYYY.</i>
Always use latest agent version	Leave this check box selected so that streaming instances that are launched from your image always include the latest AppStream 2.0 features, performance improvements, and security updates. For more information, see Amazon AppStream 2.0 Agent Version History .

2. Then, on the **Configure Image** tab. Choose **Next**.
3. Review the image details and choose **Disconnect and Create Image**.
4. The remote session disconnects within a few moments. When the **Lost Connectivity** message appears, close the browser tab.

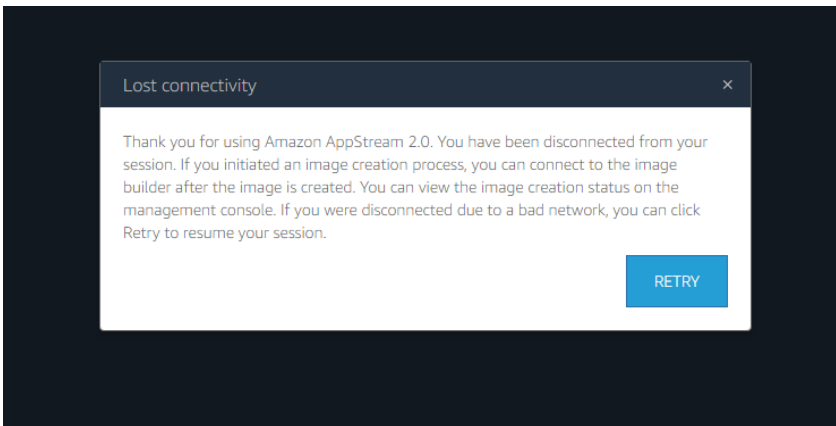


Figure 10: The **Lost connectivity** message indicating that the image creation process has started.

5. Return to the [Amazon AppStream 2.0 console](#) and choose **Images, Image Registry**. While your image is being created, the image status in the image

registry of the console appears as **Pending**. While your image is being created, you cannot connect to it.

6. Click the **Refresh** icon periodically to update the status. Image creation takes about 20 minutes. After your image is created, the image status changes to **Available** and the image builder is automatically stopped.

Note: To make changes to your image, such as adding other applications or updating existing applications, you must create a new image. To do so, restart and reconnect to the image builder, make your changes, and then repeat the Image Assistant process to create a new image that includes the changes.

Step 6: Provision a fleet

An AppStream 2.0 fleet defines the hardware, network, Active Directory (if applicable), and scaling configuration for your application streaming infrastructure. For more information, see [Amazon AppStream 2.0 Stacks and Fleets](#).

In this section, you'll do the following:

- Provide details for your fleet.
- Choose an image.
- Configure the fleet.
- Configure the network.

Provide fleet details

1. Open the AppStream 2.0 console at <https://console.aws.amazon.com/appstream2>.
2. In the navigation pane, choose **Fleets, Create Fleet**.
3. For **Step 1: Select Fleet Type**, choose a fleet type that suits your needs. The fleet type determines the availability of streaming instances and affects your costs. You can choose either of the following:
 - **Always-on:** Instances run all the time, even when no users are streaming applications. When this option is selected, instances are immediately available for the next user to connect to immediately.

- **On-Demand:** Instances run only when users are streaming applications. Idle instances that are available for streaming are in a stopped state. When this option is selected, a user must wait for one to two minutes for an instance to start up.
- **Elastic:** The pool of streaming instances is managed by AppStream 2.0. When your users select their application or desktop to launch, they will start streaming after the app block has been downloaded and mounted to a streaming instance.

For this exercise, select the **On-Demand** option.

Configure the fleet

1. **For Step 2: Provide Fleet Details**, type the following text and choose **Next**.

Option	Value
Name	The unique name identifier for the fleet, such as <i>DemoCo_Fleet_v1_mmdyyyy</i> , using any of the following characters: a-Z,0-9,-,_,. Note: The name cannot begin with "Amazon," "AWS," or "AppStream."
Display Name	The name displayed in the console, such as ' <i>DemoCo Fleet v1 Month Year</i> '.
Description	An optional description for the fleet.

2. **Choose instance type**, you define the hardware configuration for each of the instances that make up your fleet. Because you created the image by using the General Purpose family, the instance type is already populated. However, you can select any instance type options that are presented.

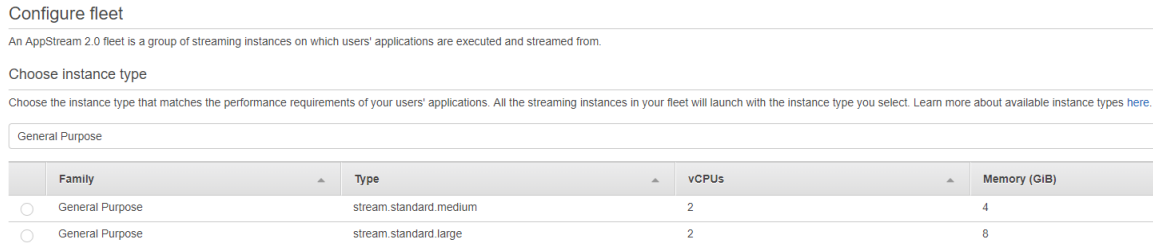


Figure 11: The **Choose instance type** section of the **Configure fleet** page.

3. For this exercise, leave the **General Purpose** instance type selected, and then choose **stream.standard.medium**. For more information, see [Amazon AppStream 2.0 Instance Families](#).
4. Under **User session details**, define the maximum amount of time that users can be connected to streaming sessions and how long streaming sessions should remain active after users disconnect.
 - **Maximum session duration** defines how long user streaming sessions 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. Choose a large value based on your use case, in this demo we select **480 minutes**.
 - **Disconnect timeout** defines how long user streaming sessions can remain active after users are disconnected. 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. After the disconnect timeout expires, the session is terminated, and the user must start a new session to reconnect. Leave the default setting of **15 minutes**.
 - **Idle Disconnect timeout** defines the amount of time that users can be idle (inactive) before they are disconnected from their streaming session and the Disconnect timeout in minutes time interval begins. Users are notified before they are disconnected due to inactivity. If they try to reconnect to the streaming session before the time interval specified in Disconnect timeout in minutes has elapsed, they are connected to their previous session. Otherwise, they are connected to a new session with a

new streaming instance. Setting this value to 0 disables it. When this value is disabled, users are not disconnected due to inactivity.

5. Under **Fleet capacity**, set **Minimum capacity** and **Maximum Capacity** . This defines minimum and maximum number of users that can stream concurrently from this fleet.

Notes:

- Capacity is defined in terms of the number of instances within a fleet and, consequently, every unique user streaming session that is served by a separate instance.

6. In Stream view details, select “Application”

Stream view details [Info](#)

Stream view

Select the stream view you wish your users to have. When you select Application, your users will have an application-specific focus. When you select Desktop, your users will see the standard desktop experience that is available on the operating system.

Application ▼

You can customize Scale-out and Scale-in policies in scaling details and also define Schedule for scaling based on your business requirements.

You don't have to select IAM role for SAP GUI use case.

7. We recommend adding Tags to your Fleet to increase observability and administration.
8. Choose **Next**.

Choose an image

For Step 3: Choose an image, choose the image that you created, scroll to the bottom of the page, and then choose **Next**.

Configure the network

1. For **Step 4: Configure Network**, make sure that the **Default Internet Access** check box is not selected. This option does not need to be selected because you already configured a VPC with a NAT gateway to provide internet access.
2. For **VPC**, select VPC that was created previously (**AppStream2 VPC**).
3. For **Subnet 1**, choose a Private Subnet previously created.
4. For **Subnet 2**, choose second Private Subnet.
5. Choose Security Group
6. Provide Active directory information if you plan to integrate your Active Directory for user login.
7. Choose **Next**.
8. Confirm the fleet configuration details. To change settings for any section, choose **Edit**, and make the needed changes. After you finish reviewing the configuration details, choose **Create fleet**.
9. In the pricing acknowledgement dialog box, select the acknowledgement check box, and choose **Create** to begin provisioning your fleet with the initial set of running instances.

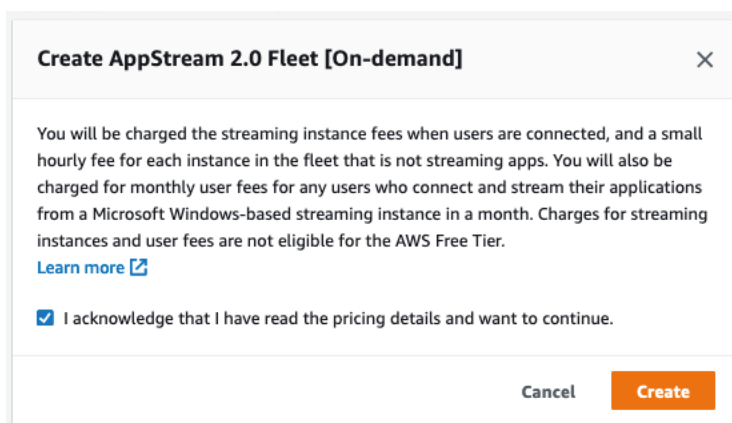


Figure 12: The AppStream 2.0 streaming instance pricing acknowledgement dialog box.

Note: If an error message notifies you that you don't have sufficient limits to create the fleet, submit a limit increase request to the AWS Support Center. For more information, see [Amazon AppStream 2.0 Service Limits](#).

Fleet provisioning usually takes 10 minutes to finish. While your fleet is being created and fleet instances are provisioned, the status of your fleet displays as **Starting** in the **Fleets** list.

10. After the status changes to **Running**, the fleet is available and you can use it to create a stack.

Step 7: Create an AppStream 2.0 stack and a streaming URL

An AppStream 2.0 stack consists of a fleet, user access policies, and storage configurations. You create a stack to start streaming applications to users.

In this section, you'll do the following:

- Provide details for your stack and associate your stack with a fleet.
- Enable persistent storage for the stack.
- Create a streaming URL.

Provide stack details and associate the stack with a fleet

1. Open the AppStream 2.0 console at <https://console.aws.amazon.com/appstream2>.
2. In the navigation pane, choose **Stacks**, **Create Stack**.
3. For **Step1: Stack Details**, type the following information.

Option	Value
Name	The unique name identifier for the stack, such as <i>DemoCo_Stack_mmdyyy</i> , using any of the following characters: a-Z,0-9,-,._.

Option	Value
Display Name	The name displayed in the console, such as <i>DemoCo Stack Month Year</i> .
Description	An optional text box where you can enter details of the stack: for example <i>Stack v1 created by (your initials or name) on mm/dd/20yy</i> .
Redirect URL	An optional URL to which users are redirected at the end of their streaming session. For example: <i>https://aws.amazon.com</i>
Feedback URL	An optional URL, when provided, the users will be displayed the option to submit feedback
Fleet	Select the <i>DemoCo_Fleet_v1_mmddyyy</i> fleet that you created.

4. If you want to ensure your AppStream streaming does not traverse Internet, create VPC endpoint for AppStream Streaming in the VPC and subnet we created previously.
5. For **Streaming Experience Settings**, select a protocol preference for your stack from the dropdown list. For more information, see [System Requirements and Feature Support \(AppStream 2.0 Client\)](#).
6. We recommend adding Tags to your Stack to increase observability and administration.
7. Choose **Next**

Enable persistent storage and user settings for the stack

1. For **Step 2: Enable Storage**, make sure that the **Enable Home Folders** option is selected. When this option is selected for an AppStream 2.0 stack, users of the stack are presented with a persistent storage folder in their AppStream 2.0 sessions. Data stored by users in their Home Folders is backed up to an Amazon S3 bucket that is automatically created in your AWS account. For more information, see [Persistent Storage with AppStream 2.0 Home Folders](#).

Home folders

Your user's files will be saved to an S3 bucket in your AWS account. For this feature to be enabled, the AppStream 2.0 fleet associated with this stack must allow access to S3 via the internet or an Amazon VPC endpoint for S3.

Home folders

Enable home folders

S3 bucket name

appstream2-36fb080bb8-us-east-1-574307099575 [↗](#)

Figure 13: The **Enable Home Folders** page, displaying the Amazon S3 bucket that is automatically created.

You also have options to save files to Google Drive or oneDrive. You can select these options as needed at this step.

2. Choose **Next**.
3. For Step 3, **Edit user settings**, you can select options to specific how users can transfer data between their AppStream 2.0 remote session and their local device.

Edit user settings

Specify how users can transfer data between their AppStream 2.0 remote session and their local device.

Clipboard, file transfer, print to local device, and authentication permissions

Specify how users can transfer data between their AppStream 2.0 remote session and their local device.

Clipboard

Copy and paste ▼

File transfer

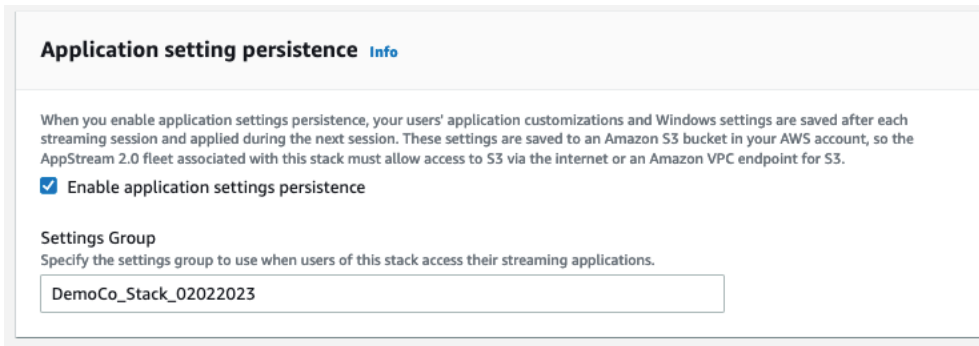
Upload and download ▼

Enable print to local device

Password sign in for Active Directory [Info](#)

Smart card sign in for Active Directory [Info](#)

4. Under **Application setting persistence**, you can enable application setting persistence so that your users' application cusotmizations and Windows settings are saved after each streaming session and used for the next session.



5. Choose **Next**.

6. Confirm the stack configuration details. To change the settings for any section, choose **Edit** and make the needed changes. After you finish reviewing the configuration details, choose **Create Stack**.

After a few moments, the **Stacks** list reappears. Your stack is listed with a status of **Active**.

Create a streaming URL

To quickly test application streaming without setting up users, create a temporary URL that can be pasted into a new browser window.

1. In the navigation pane, choose **Stacks**.
2. For **Stacks**, select the stack that you just created.
3. Choose **Actions**, **Create streaming URL**.

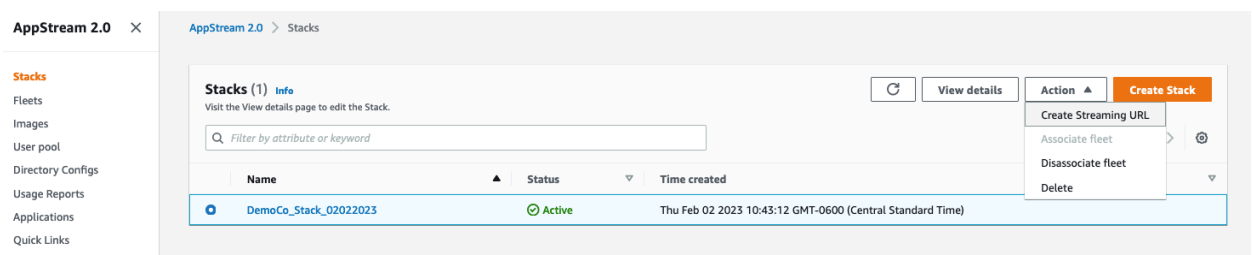


Figure 14: The **Create streaming URL** menu item.

4. In the **Create streaming URL** dialog box, type the following information and choose **Get URL**.

Option	Value
User id	An ID for the user of the streaming URL. For this exercise, type <i>DemoCoTestUser1</i> . Note: This entry is not tied to Active Directory or an LDAP type of directory service. It is simply an identifier for creating a unique URL.
Session Expiration	The length of time that this URL is available to use. For this exercise, choose 1 hour .

- The **Create streaming URL** dialog box displays the user ID that you entered and the URL that AppStream 2.0 generated for the user.
- Choose **Copy Link** to copy the full URL to the clipboard.

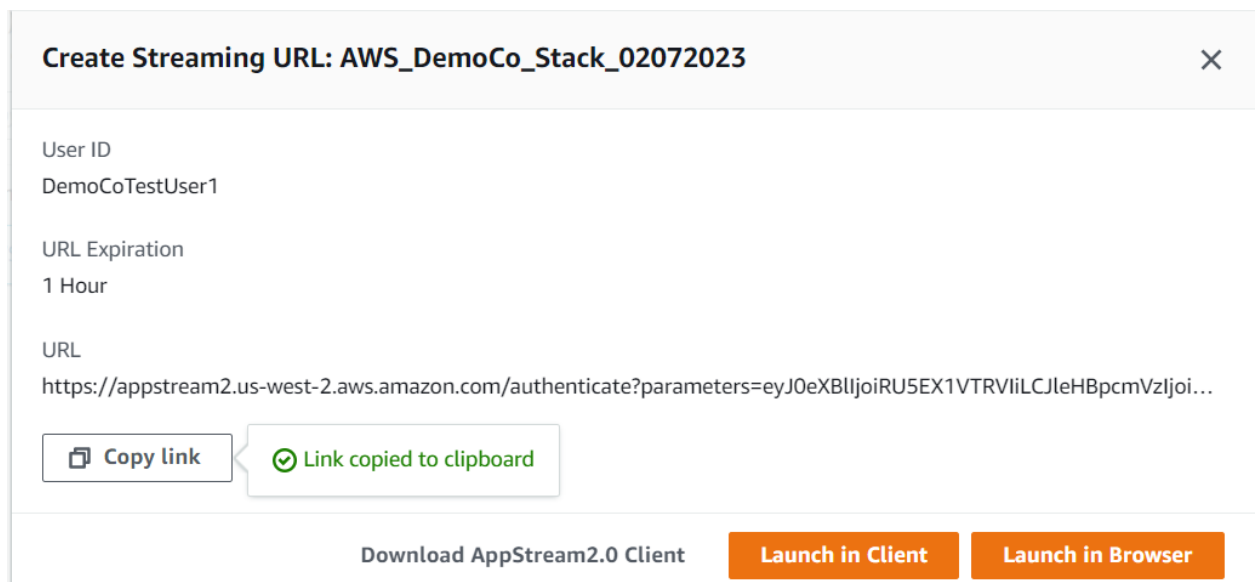


Figure 15: The **Create streaming URL** dialog box.

Notes:

- Providing access to an AppStream 2.0 streaming session through a console-generated link as described in this procedure is for testing only.

- In a production environment, several authentication and authorization options are available to provide your users with access to AppStream 2.0. These options include federation through SAML 2.0, the AppStream 2.0 user pool (a built-in identity management feature), and custom identity solutions. For more information, see [Setting up SAML](#) and [Manage Access with the AppStream 2.0 User Pool](#).
7. If you have [AppStream2.0 client](#) you can click on “Launch in Client”. Or you can click “Launch in Browser” or in a browser, open a new tab, paste the streaming URL into the address bar, and press **Enter**. AppStream 2.0 displays an application catalog page that lists the applications that you have configured for streaming.
 8. Choose the SAP GUI icon to stream the application, and confirm that it functions as expected.
 9. After you confirm that SAP GUI functions as expected when streaming, close the browser tab that you opened to test the AppStream 2.0 environment.

Step 8: Manage user access with an AppStream 2.0 user pool

An AppStream 2.0 user pool is a built-in identity management feature that you can use to enable users to access their streamed applications. Alternatively, you can use SAML 2.0 to federate through Microsoft Active Directory or any other custom identity solution provider that supports SAML 2.0.

Note: This guide describes how to manage user access to AppStream 2.0 with the user pool. For information about configuring third-party SAML 2.0 identity provider solutions to work with AppStream 2.0, see [AppStream 2.0 Integration with SAML 2.0](#).

To enable users in the user pool to open applications after they sign in to the AppStream 2.0 user portal, you must assign each user to at least one stack that contains applications. After you assign the user to a stack, AppStream 2.0 sends an optional notification email to the user with instructions about how to access the stack and a URL. The user can access the stack by using the URL until you delete the stack or unassign the user from the stack.

In this section, you'll configure an AppStream 2.0 user pool and grant a user access to AppStream 2.0 by doing the following:

- Create a user in the user pool. AppStream 2.0 then sends a welcome email with instructions and a temporary password.
- Assign the stack that you created to the user.

Create a user

1. Open the AppStream 2.0 console at <https://console.aws.amazon.com/appstream2>.
2. In the navigation pane, choose **User Pool**, **Create User**.
3. In the **Create User** screen, type the following information and choose **Create User**.

Option	Value
Email	An active email address that you can access.
First Name	The first name of the user.
Last Name	The last name of the user.

4. After a few moments, the **User Pool** list refreshes, and the user is listed and enabled.

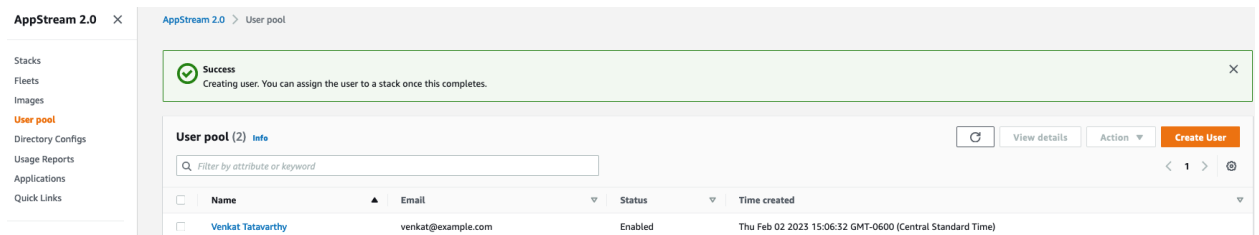


Figure 16: The User Pool dashboard showing the newly created user.

Assign a stack to the user

1. In the navigation pane, choose **User Pool**, and select the user that you created.
2. Choose **Actions**, **Assign Stack**.
3. In the **Assign Stack** dialog box, for **Stack**, select the *DemoCo_Stack_mmddyyyy* stack that you created earlier.

4. Leave the **Send email notification to user** option selected.
5. Choose **Assign Stack**.
6. After a few moments, the **User Pool** list refreshes. The user that you created appears under **User Details** with *DemoCo_Stack_mmddyyyy* as an assigned stack.

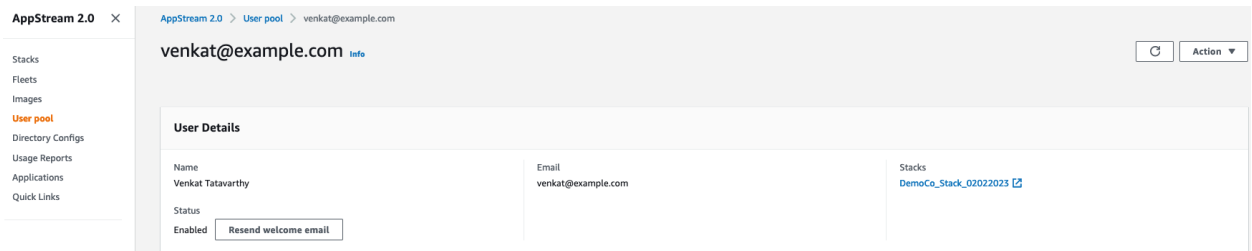


Figure 17: The User Pool dashboard showing the newly created user with a stack now assigned.

Step 9: Test the end user authentication and application streaming experience

In the previous section, you added a user to the user pool by providing a name and an email address and then assigned a stack to the user. AppStream 2.0 sent an email to the email address after each action. To test the end user experience, sign in to AppStream 2.0 as the user that you created and start a streaming session.

1. Open the first notification email that you received, and open the **Login page** link. The AppStream 2.0 portal sign-in page opens in your browser.

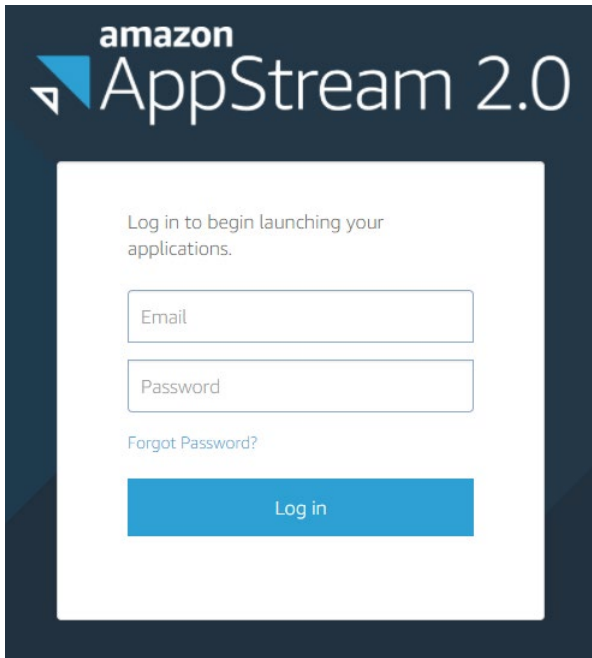


Figure 18: The AppStream 2.0 user login prompt.

2. Type the email address used for the user that you created and the temporary password that was provided in the email, and then choose **Log in**.
3. When prompted, type a new password, confirm it, and then choose **Set Password**. The AppStream 2.0 application catalog page opens, displaying the applications that are available for streaming.
4. Choose an application to begin streaming.

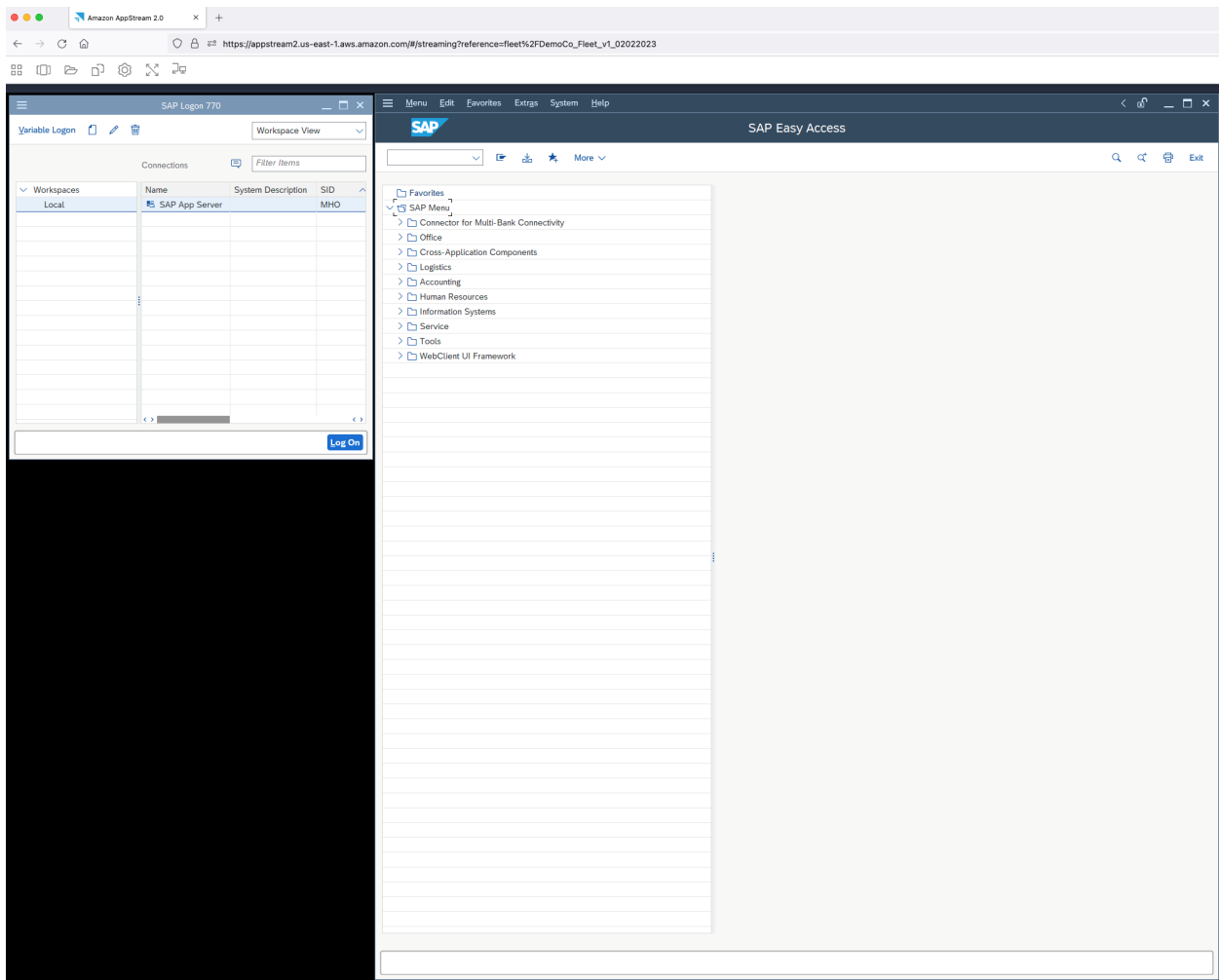


Figure 19: AppStream 2.0 End user experience

Step 10: Take the next step with AppStream 2.0

Congratulations, you have now successfully created an AppStream 2.0 environment to stream SAP GUI desktop application. Below is an architectural diagram illustrating the AppStream 2.0 environment you created:

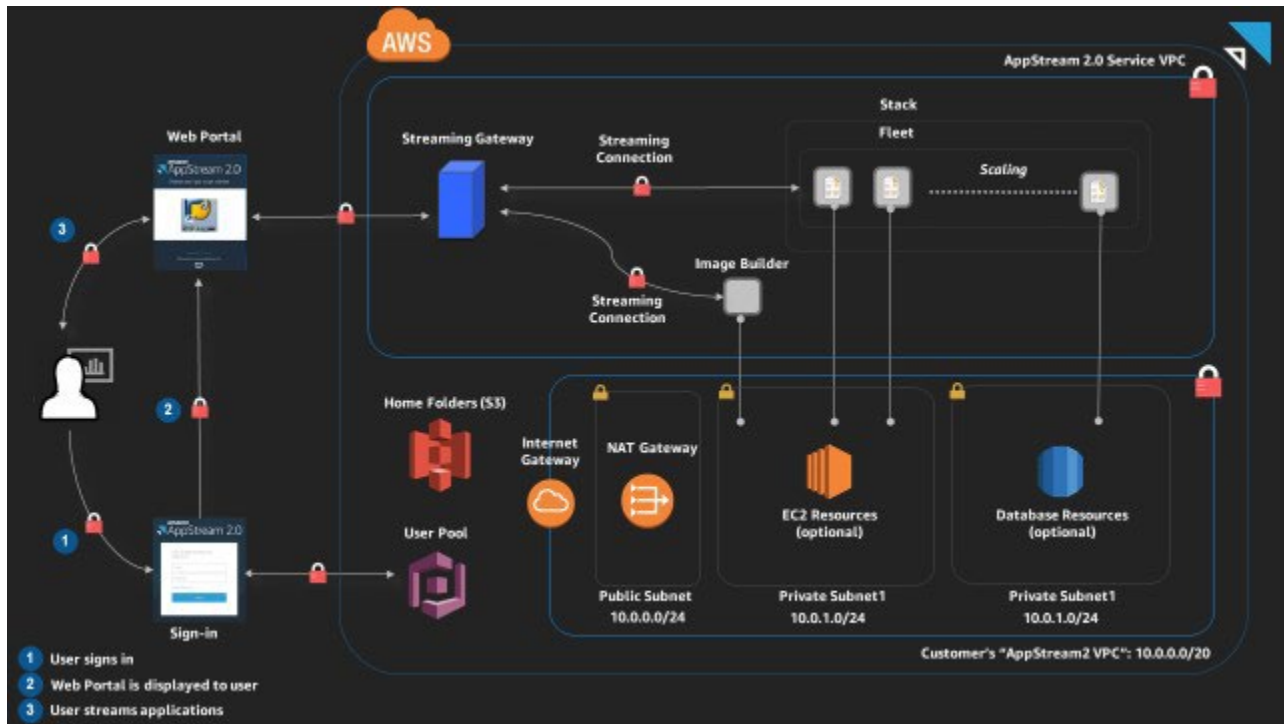


Figure 20: Your AppStream 2.0 environment.

This guide provided an introduction to AppStream 2.0 by walking you through basic configuration and deployment exercises. To increase your understanding of AppStream 2.0 and take advantage of more features, consider doing the following:

1. Try using different instance types and sizes to match your application's requirements. For information about the different instance types and sizes available for AppStream 2.0, and their pricing, see [Amazon AppStream 2.0 Pricing](#).
2. Enable single sign-on (SSO) access to your streamed applications through SAML 2.0. When you do this, your users can use their existing credentials to sign into AppStream 2.0 streaming sessions through your own web portal. For more information, see [Single Sign-on Access to AppStream 2.0 Using SAML 2.0](#).
3. Join your AppStream 2.0 fleets and image builders to domains in Microsoft Active Directory. Your users can then benefit from access to Active Directory network resources such as printers and file shares from within their streaming sessions. You can also apply Group Policy settings to your streaming instances and users to meet the needs of your organization. For more information, see [Using Active Directory with AppStream 2.0](#).

4. Configure your fleet scaling policies to increase or decrease the number of instances available to users in response to changes in user demand or according to time of day. For more information, see [Fleet Auto Scaling for Amazon AppStream 2.0](#).
5. To delete the resources to avoid charges to your account see *Appendix B*.
6. For more information about AppStream 2.0 pricing, see [Amazon AppStream 2.0 Pricing](#).

Appendix A: Create and activate an AWS account

If you do not already have an AWS account, complete the following steps to create and activate one. During this process, you do the following:

- Create your AWS account.
- Add a payment method.
- Verify your phone number.
- Select an AWS Support plan.
- Watch for three account confirmation emails.

Create your AWS account

1. In a browser window, open the [Amazon Web Services](#) webpage.
2. Choose **Create an AWS Account**. If you've signed in to AWS recently, you might see **Sign In to the Console** instead. If **Create a new AWS account** isn't visible, choose **Sign in to a different account, Create a new AWS account**.
3. On the **Create an AWS Account** page, type a valid email address, a password and password confirmation, and an AWS account name.
4. You must note the account name, email address, and password that you choose for your AWS account because you need these credentials to sign in to AWS.
5. Choose **Continue**.
6. On the **Contact Information** page, the option to choose a company account or personal account is available. These two account types function identically. For the exercises in this guide, choose **Personal Account**, and then enter the requested contact information.
7. Review the **AWS Customer Agreement**, and select the corresponding check box.
8. Choose **Create Account and Continue**.

Add a payment method

On the **Payment Information** page, type the requested information associated with your payment method. If the address for your payment method is the same as the address you provided for your account, choose **Verify and Add**. Otherwise, choose Use a new address, type the billing address for your payment method, and then choose **Verify and Add**.

Verify your phone number

1. On the **Phone Verification** page, type a phone number that you can use to accept incoming calls.
2. Type the code displayed in the captcha.
3. When you're ready to receive the call, choose **Call me Now**. In a few moments, you'll receive an automated call from AWS that prompts you to enter your PIN to validate the AWS account.
4. When you receive the call, enter the provided PIN on your phone's keypad.
5. After the process is complete, choose **Continue**.

Choose an AWS Support plan

On the **Select a Support Plan** page, choose appropriate support plan. For information about AWS Support, see [Compare AWS Support Plans](#).

After you choose a Support plan, a confirmation page indicates that your AWS account is being activated. Accounts are usually activated within a few minutes, but the process may take up to 24 hours. If you attempt to sign in to the AWS Management Console before your account is active, the following message appears:

Your service sign-up is almost complete!

Thanks for signing up with Amazon Web Services. Your services may take up to 24 hours to fully activate. If you're unable to access AWS services after that time, here are a few things you can do to expedite the process:

1. Make sure you provided all necessary information during signup. [Complete your AWS registration](#).
2. Check your email to see if you have received any requests for additional information. If you have, please respond to those emails with the information requested.
3. Verify your [credit card information](#) is correct. Also, check your credit card activity to see if there's a \$1 authorization (this is not a charge). You may need to contact your card issuer to approve the authorization.

If the problem persists, please contact Support:

[Contact Support](#)

Figure 21: Message that appears if you sign in before your account activation is complete.

Watch for three AWS account confirmation emails

When you sign up for your account, you receive three account confirmation emails:

- The first email, with a subject line of “Welcome to Amazon Web Services,” confirms the creation of your AWS account and is sent almost immediately after you verify your phone number.
- The second email, with a subject line of “AWS Support (Basic) Sign-Up Confirmation,” confirms the AWS Support option that you selected during the account creation process.
- The third email, with a subject line of “Your AWS Account is Ready - Get Started Now,” is sent after your AWS account ID is ready to use. After you receive this email, you can access AWS services by using the [AWS Management Console](#).

Appendix B. Clean up your AppStream 2.0 resources

Although you can continue to use this AppStream 2.0 environment, keep in mind that you pay for your running resources. For more information, see [Amazon AppStream 2.0 Pricing](#).

Cleaning up the resources that you created frees up resources and helps you avoid unintended charges to your account.

Stop and delete your image builder

1. Open the AppStream 2.0 console at <https://console.aws.amazon.com/appstream2>.
2. In the navigation pane, choose **Images, Image Builder**.
3. Confirm whether the image builder that you created in Step 3 in this guide is in a stopped state. If not, select the image builder and choose **Actions, Stop**. If you created multiple image builders, repeat this step for each image builder that you created.
4. After the image builder has stopped, choose **Actions, Delete**. Repeat this step for each image builder that you created.

Revoke stack permissions for users in the user pool

1. In the navigation pane, choose **User Pool**.
2. Select the user you created in Step 9 in this guide and choose **Actions, Unassign stack**. This action revokes the stack permissions for the user.

Disassociate your fleets from your stack and delete your stack

1. In the navigation pane, choose **Stacks**.
2. Select the stack you created and choose **Actions, Dissociate Fleet**. This action dissociates the fleet from the stack.
3. To delete the stack, choose **Actions, Delete**.

Stop and delete your fleet

1. In the navigation pane, choose **Fleets**.

2. Confirm whether the fleet that you created in Step 6 in this guide is in a stopped state. If not, select the fleet and choose **Actions**, **Stop**.
3. After the fleet has stopped, choose **Actions**, **Delete**.

Appendix F. Additional resources

For more information about AppStream 2.0, visit the following resources:

- [Amazon AppStream 2.0 Product Details](#)
- [Amazon AppStream 2.0 Pricing Details](#)
- [Amazon AppStream 2.0 FAQs](#)
- [Amazon AppStream 2.0 Developer Guide](#)
- [Amazon AppStream 2.0 API Reference](#)
- [Amazon AppStream 2.0 CLI Reference](#)
- [Amazon AppStream 2.0 Try It Now Demo](#)
- [Amazon AppStream 2.0 Resources](#)