NX CAD ON AMAZON APPSTREAM 2.0

Deployment Guide

Build an Amazon AppStream 2.0 environment to stream NX CAD
November 2018

Contents

Introduction

This guide describes how to deploy and stream NX CAD desktop application 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 an isolated virtual network infrastructure within the AWS Cloud. Your AppStream 2.0 resources will use this environment.
- Provision a Siemens PLM License Server in your VPC.
- 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 NX CAD for streaming using an image builder.
    2. Provision a fleet of instances to stream your applications. The fleet will use the Graphics Design 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 applications.
    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.
- **A Siemens PLM Software account (WebKey).** This account is required to log in and download NX installers. If you don't have a WebKey, you can create one here: https://www2.industrysoftware.automation.siemens.com/webkey/.
- **An active NX Network (floating) License.** This license file is required for the NX products that you want to import into AppStream 2.0 and will be deployed along with the Siemens PLM License Server on AWS. Contact your Siemens PLM Software reseller or account executive for more information about NX licenses.
- **End user client recommendations:** To use NX delivered through AppStream, the user would need a modern HTML browser such as Google Chrome, Mozilla Firefox, Microsoft Edge or Internet Explorer 11+. Your local computer should support a minimum display resolution of 1024x768.
- **End user network recommendations:** AppStream 2.0 uses an adaptive streaming protocol (NICE DCV) 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.

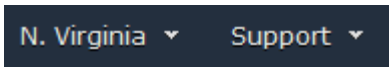
The amount of bandwidth used when transmitting pixels is proportional to the changes on the screen and the resolution of the display monitor(s) used by the client device. The changes on the screen and the resolution are determined by the type of application (3D versus business application) and usage 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 sample NX use-cases.

Use case	**Recommended bandwidth available per user**	**Recommended maximum roundtrip latency**
NX CAD – Streaming with low fidelity datasets or maps with 2K monitors	5-6 mbps	< 100 ms
NX CAD – Streaming high fidelity datasets or maps with 4K monitors	10-12 mbps	< 50 ms

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. AWS currently hosts services in 18 different geographical areas.



5. Select one of the following seven Regions in which AppStream 2.0 is available:

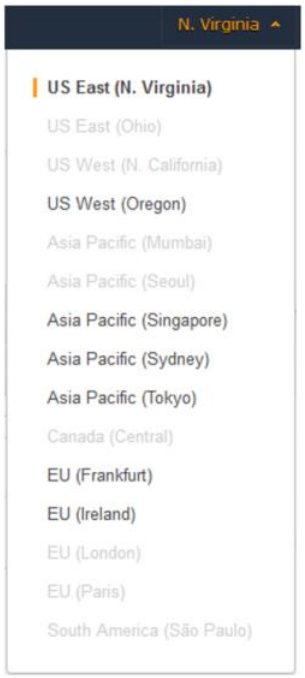


Figure 1: Available AWS Regions for AppStream 2.0.

Step 2: Create network resources

In this section, you will create an Amazon virtual private cloud (VPC) and other network resources required for your AppStream 2.0 environment. The following steps use a template in AWS CloudFormation to automatically create and configure the necessary network resources. To manually create and configure network resources, see *Appendix B*.

1. Make sure that you are signed in to the AWS Management Console.
2. In the following list of regional choices, open the CloudFormation link associated with the AWS Region in which you want to build your AppStream 2.0 environment.

- US East (N. Virginia)
- US West (Oregon)
- EU (Frankfurt)
- EU (Ireland)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)

The AWS CloudFormation console displays the URL of a template that is used to create your network resources and the name of the resulting AWS CloudFormation stack.

3. In the bottom right corner of the window, choose **Create**. AWS CloudFormation starts creating the resources and displays a status message to indicate progress.

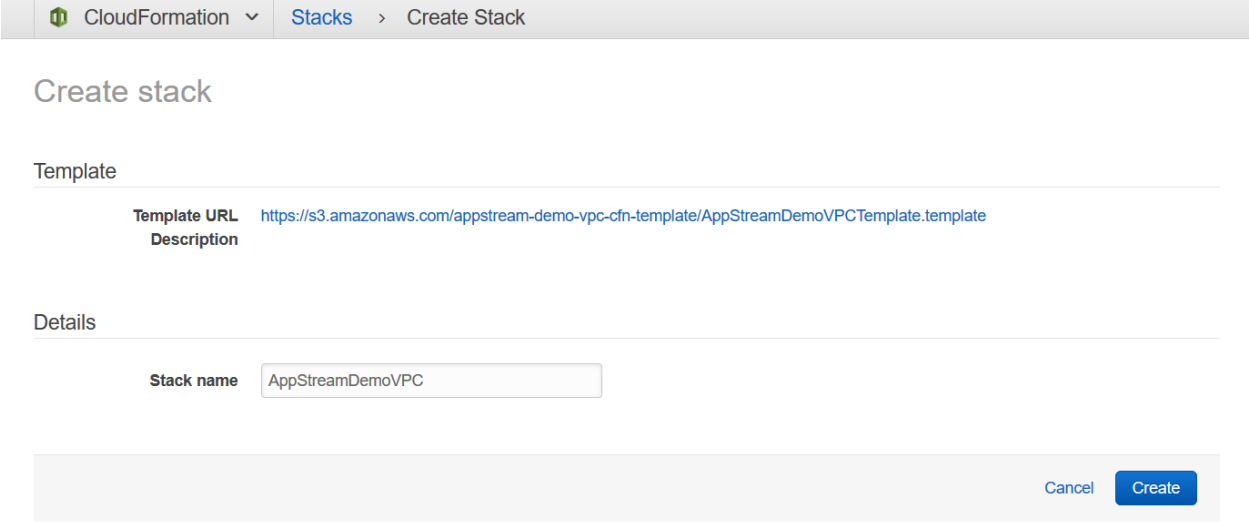


Figure 2: Using a template in AWS CloudFormation to create network resources.

4. When the creation process completes, usually within five minutes, the AWS CloudFormation console displays the status **CREATE_COMPLETE**.

5. Navigate to the Amazon VPC console at https://console.aws.amazon.com/vpc/.

6. In the navigation pane, under **Virtual Private Cloud**, choose **Your VPCs**. In the list of VPCs, you should see the following VPC that was automatically created:

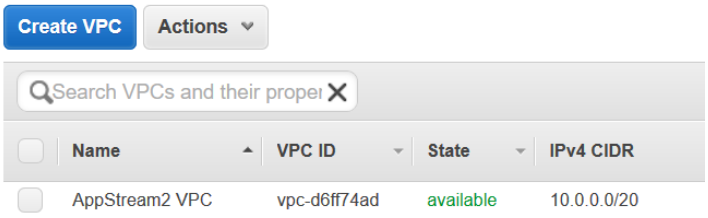


Figure 3: VPC created by AWS CloudFormation.

Note: The VPC ID value will differ for your VPC.

7. In the navigation pane, under **Virtual Private Cloud**, choose **Subnets**. In the list of subnets, you should see the following subnets that were automatically created:

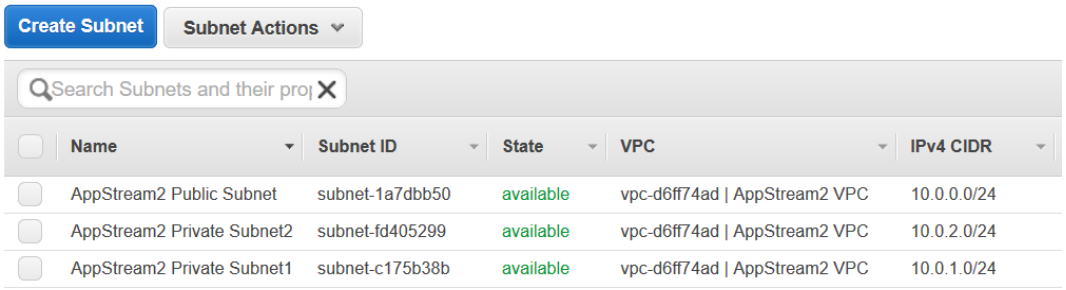


Figure 4: Subnets created by AWS CloudFormation.

Note: The Subnet ID and VPC values will differ for your subnets.

8. You have now successfully created your network resources by using AWS CloudFormation. You can proceed to Step 3.

Step 3: Set up the Siemens PLM License Server for NX

We will now deploy the Siemens PLM License Server by using an EC2 (Elastic Compute Cloud) instance. NX applications running on your AppStream 2.0 streaming instances will contact this license server for license activation.

In this section, you will:

- Configure an EC2 instance that will function as the Siemens PLM License Server.
- Configure VPC Security Group rules for the license server instance.
- Configure another EC2 instance that will be used to remotely administer the license server.
- Install and configure the Siemens PLM License Server.
- Configure Windows Firewall on the license server instance to open up the license communication ports.

Configure an EC2 instance that will function as the Siemens PLM License Server

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Click **Launch Instance**.
3. In **Step 1: Choose an Amazon Machine Image (AMI)**, select the **Microsoft Windows Server 2012 R2 Base** image.
4. In **Step 2: Choose an Instance Type**, select **t2.medium** as the instance type, and then click **Next: Configure Instance Details**.
5. In **Step 3: Configure Instance Details**, choose the following
    a. **Network** – AppStream2 VPC
    b. **Subnet** – AppStream2 Private Subnet1
6. Leave the remaining options set to their default values, and then click **Next: Add Storage**.
7. In **Step 4: Add Storage**, in **Size (GiB**), enter 60, and then click **Review and Launch**.
8. In **Step 7: Review Instance Launch**, confirm that the instance configuration details are correct, and then click **Launch**.
9. In **Select an existing key pair or create a new key pair**, choose **Create a new key pair.** Provide a name for the instance key pair. Click **Download Key Pair**, download the key pair file and store it securely for future reference.
10. Click **Launch Instances**.
11. Click **View Instances** to view the newly launched EC2 instance. This instance will be used as the Siemens PLM License Server and is provisioned in the private subnet.

Configure security group rules for the license server instance

1. On the EC2 Dashboard, in the navigation pane, select **Instances**, and then select the instance that you just launched.

2. On the **Description** tab for the instance, click the name of the security group that is associated with the instance. The name of the security group is formatted as a link. Clicking this link opens the details page for the security group.

3. On the details page for the security group, click the **Inbound** tab, and then click **Edit**. Enter the first rule shown in the following screenshot, and then click **Save**.

Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	28002	sg-0a0f... (Your server security group)	License Server 2
Custom TCP Rule	TCP	28000	sg-0a0f... (Your server security group)	License Server 1
RDP	TCP	3389	sg-02ff... (Your jump host security group)	Remote Desktop

Figure 5: Inbound Security Group rules for the license server

This rule allows NX applications to communicate with the Siemens PLM License Server over TCP ports 28000 and 28002 assuming that the license file was properly edited to configure the server this way.

Note: Do not configure the second security group rule for now. You will configure this rule after you launch the jump host.

4. Click the **Outbound** tab, and confirm that the outbound rule is configured as shown in the following screenshot.

Edit

Type	Protocol	Port Range	Destination	Description
All traffic	All	All	0.0.0.0/0	

Figure 6: Outbound Security Groups rules for the license server

Configure an EC2 instance that will act as the jump host to enable remote license server administration

1. Repeat the steps in the “Configure an EC2 instance that will function as the Siemens PLM License Server” procedure to launch a second instance, but with

the following changes for step 5 of the procedure (which describes how to select the instance subnet on the **Configure Instance Details** page):

- In **Subnet**, select the public subnet (*AppStream2 Public Subnet1)*.
- In **Enable for Auto-assign Public IP**, select **Enable***.*

You will use this instance to connect to the Siemens PLM License Server that is deployed in the private subnet.

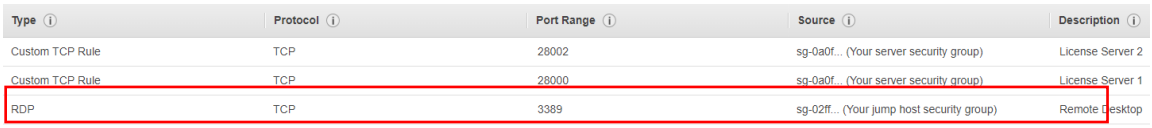
2. On the EC2 Dashboard, in the navigation pane, select **Instances**, and then select the new instance that you just launched.
3. On the **Description** tab for the instance, create a public facing security group that is associated with the instance.
4. Add the IP address of the machine that that you will access the jump host from.

Type	Protocol	Port Range	Source	Description
RDP (3389)	TCP (6)	3389	Your IP ./32	Windows Remote Deskt...

5. On the details page for the security group, note the ID that is displayed in the Group ID column. You will need to enter this ID later in the procedure.
6. On the navigation pane, click **Instances** to return to the list of EC2 instances.
7. Select the Siemens PLM License Server instance.
8. On the **Description** tab for the instance, click the name of the security group that is associated with the instance.
9. On the details page for the security group, click the **Inbound** tab, click **Edit**, and then click **Add Rule**.
10. Enter the second rule shown in the following screenshot.

- In the **Source** box, enter the security group ID that you noted in step 4 (the ID shown in the screenshot is an example ID).

- In the **Source** list, select the security group of the jump host.



Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	28002	sg-0a0f... (Your server security group)	License Server 2
Custom TCP Rule	TCP	28000	sg-0a0f... (Your server security group)	License Server 1
RDP	TCP	3389	sg-02ff... (Your jump host security group)	Remote Desktop

Figure 7: Inbound Security Group rules for the jump host

This rule adds one more security group rule to the security group associated with the license server instance. The rule enables you to use the jump host to establish a remote connection to the license server so that you can remotely administer the server.

11. Use the key pair file that you downloaded to your local computer earlier (Step 3, item 9) to obtain the password for the license server and the jump host. To do this, complete the following steps for each instance.

    a. In the navigation pane, select **Instances**, select the instance, choose **Connect**, and in **Connect To Your Instance**, choose **Get Password**.

    b. In **Key Pair Path**, click **Browse**, and then locate the file that contains the key pair associated with the instance.

    c. Click **Decrypt Password**. Save the decrypted password to your local computer for later use.

    d. Download the RDP connection file to your local computer so that you can use it to start a remote connection

12. On your local computer, open an RDP client to establish a remote connection to the jump host.

13. On the jump host, open an RDP client to establish a remote connection to the Siemens PLM License Server.

Install and configure the Siemens PLM License Server

The basic installation and configuration steps are listed below. For detailed instructions, please refer to the document 'Installing Siemens PLM License Server' (SPLM_Licensing_Install.pdf) available at the Customer Support (GTAC) site:

http://www.siemens.com/gtac →Support Links → Download or Upload Files → Product Filter → Siemens PLM Licensing → Product updates → Documentation

1. Download the latest Siemens PLM Licensing installation kit e.g. SPLMLicenseServer_9.0.1_win_setup.exe from the GTAC site here:

   http://www.siemens.com/gtac →Support Links → Download or Upload Files → Product Filter → Siemens PLM Licensing → Product updates → Installs

   Note: It is recommended to use the latest available license server on the GTAC site. The latest license server supports the current release on NX as well as the older versions. Please refer to the GTAC site for more details about supported versions.

2. Obtain a license file from your Siemens PLM Software reseller or account executive.

3. Edit the Vendor line of the license file and append PORT=28002 to assign a static port for ugslmd.exe. Make sure PORT is in uppercase.

```
24  # Products:        NX,
25  #
26  ##########################################
27  SERVER                                  28000
28  VENDOR ugslmd PORT=28002
29  PACKAGE COMMSITE ugslmd 12.0 COMPONENTS=C
```

4. Run the license server installer executable as an Administrator user. (SPLMLicenseServer_v9.0.1_win6_setup.exe)

5. Select the language for the installer program to use

6. At the intro page click **Next**

7. Select the installation location and click **Next**
8. Navigate to the license file **Choose** → **Next**
9. Review the Summary, **Install**

Configure Windows Firewall to open up the license server ports

To enable the license server to communicate with client computers, complete the following steps to add exception rules to open the required ports on Windows Firewall.

1. Select **Start -> Control Panel -> Security and System -> Windows Firewall**.
2. In the left pane, select **Advanced settings**.
3. Right-click **Inbound Rules**, and then select **New Rule**.
4. Select **Port,** and then click **Next**.
5. Make sure that **TCP** is selected, enter port **28000 and 28002** in **Specific local port**, and then click **Next**.
6. Select **Allow the connection,** and then click **Next**.
7. Enter a name for the rule, and then click **Finish**.

Step 4: 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:

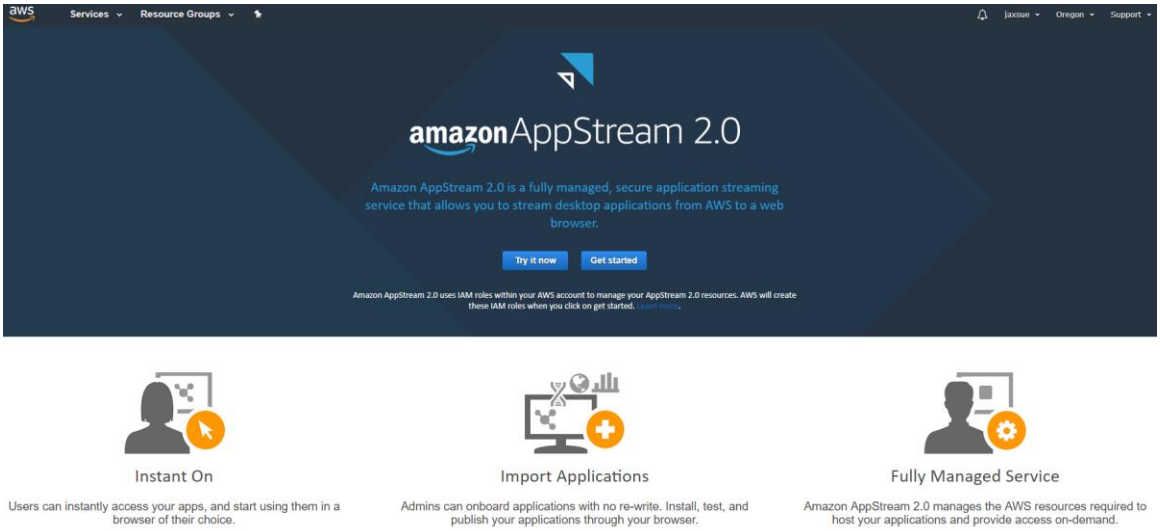


Figure 8: The AppStream 2.0 first experience page.

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).

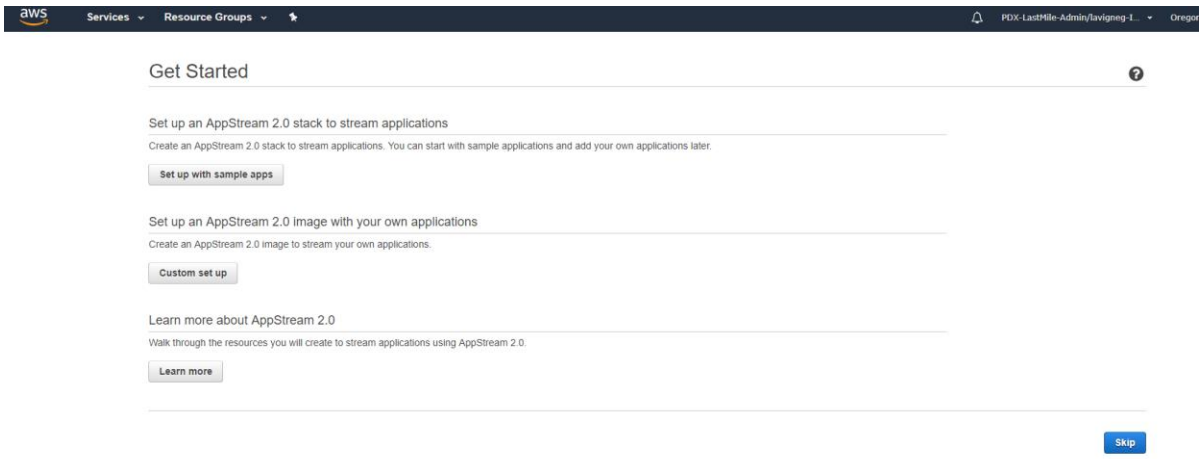


Figure 9: AppStream 2.0 getting started options.

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 *Graphics-Design-Image-Builder-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.
7. At the bottom of the page, choose **Next.**
8. In **Step 2: Configure Image Builder**, the following image builder configuration options are displayed:
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 *DemoCo_Image_v1_mmddyyyy*, using any of the following characters: a-Z,0–9,-,_,.
Display Name	Provide an optional name, such as *DemoCo Image v1 Sept 2018*, to be displayed in the console for easier reference and readability.
Instance Family	Choose **Graphics Design**
Instance Type	Select **stream.graphics-design.xlarge** from the list of available instance types.
Choose **Next** to continue to **Step 3: Configure Network** and then type the following information	
Default Internet Access	Make sure that this option is not selected.
VPC	Select the option corresponding to **AppStream 2 VPC**.
Subnet	Select the subnet with the IP address range 10.0.1.0/24 (**AppStream2 Private Subnet1**).

Option	Value
Security group(s)	Accept the default security group listed.
Active Directory Domain (Optional)	Do **not** configure any options.

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**. If an error message notifies you that you don't have sufficient limits to create the image builder, submit a limit increase request through the AWS Support Center. For more information, see AWS Service Limits.

12. The image builder creation process takes about 15 minutes to complete. During this process, the status of the image builder displays as **Pending** while AppStream 2.0 provisions the necessary resources.

13. Click the **Refresh** icon periodically to update the image builder status. After the status changes to **Running**, the image builder is 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. No user fees are incurred when users connect to an image builder. For more information, see AppStream 2.0 Pricing.

Step 5: 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 (e.g. *DemoCo_Image1_mmddyyyy*). Verify that its status is **Running** and choose **Connect**.

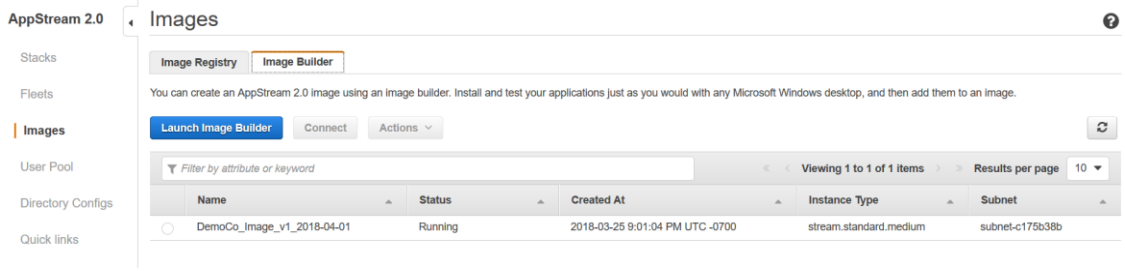


Figure 10: Choosing an image builder instance.

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**.

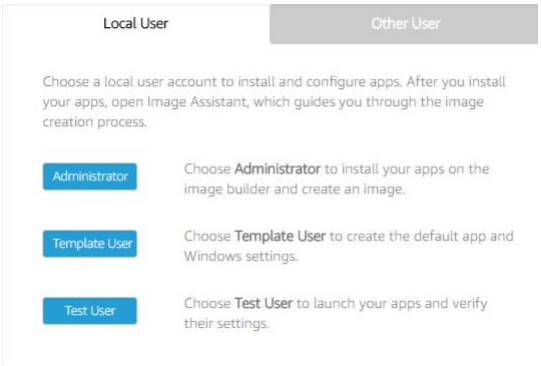


Figure 11: The image builder instance login options window.

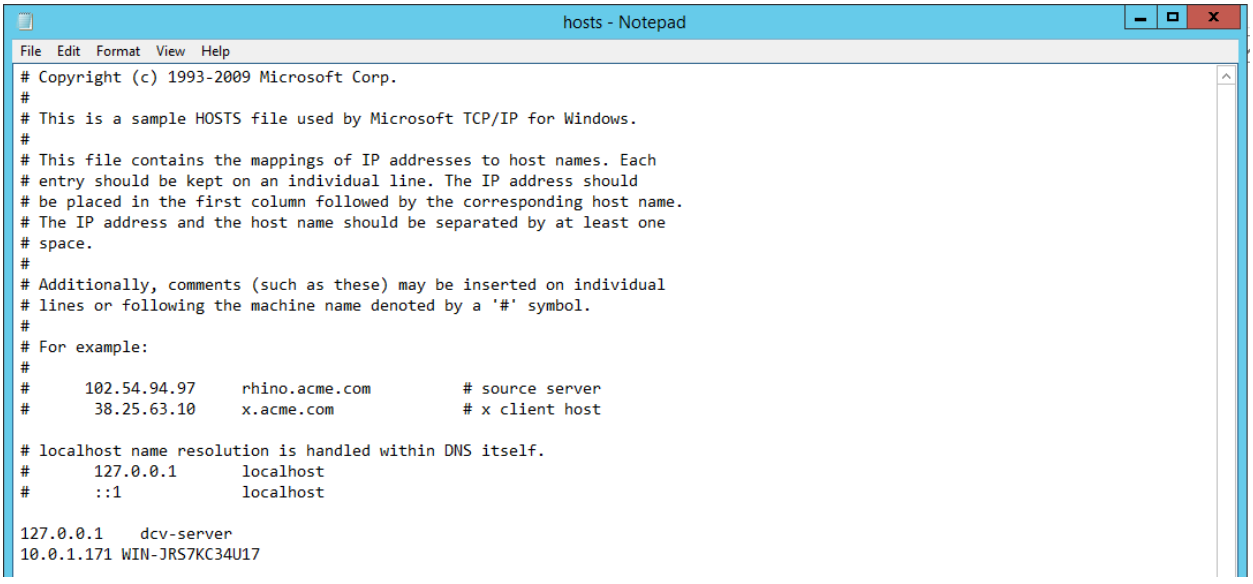
Note: If a new browser tab does not open, configure your browser to allow pop-ups from https://console.aws.amazon.com/.

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

Add the license server hostname to the image builder host list

The next step is to map the IP address of the computer name of the Siemens PLM License Server to the hostname list of the image builder. This enables the image builder to communicate with the license server.

1. Log in into the license server EC2 instance by using the jump host.
2. After you are connected to the license server, open **Command Prompt**, type **ipconfig** and then press ENTER. Note the IP address of the license server.
3. At the command prompt, type **hostname** and then press ENTER. Note the computer name of the license server.
4. Switch to the AppStream 2.0 image builder. On the image builder, launch **Powershell** using the **Run as Administrator** option.
5. Change directory to **C:\Windows\system32\drivers\etc** and open the **hosts** file by using the command **notepad hosts.**
6. Add the IP address and computer name to the hosts file as shown in the following screenshot, and then save the file.



Install and configure NX

As with any application that you want to have available for every streaming instance launched, you will need to install that application in the image builder as the administrator user. To do this, launch a web browser application (e.g. Internet Explorer or Google Chrome), and navigate to the GTAC download site, and enter your WebKey credentials to log in:

https://download.industrysoftware.automation.siemens.com/

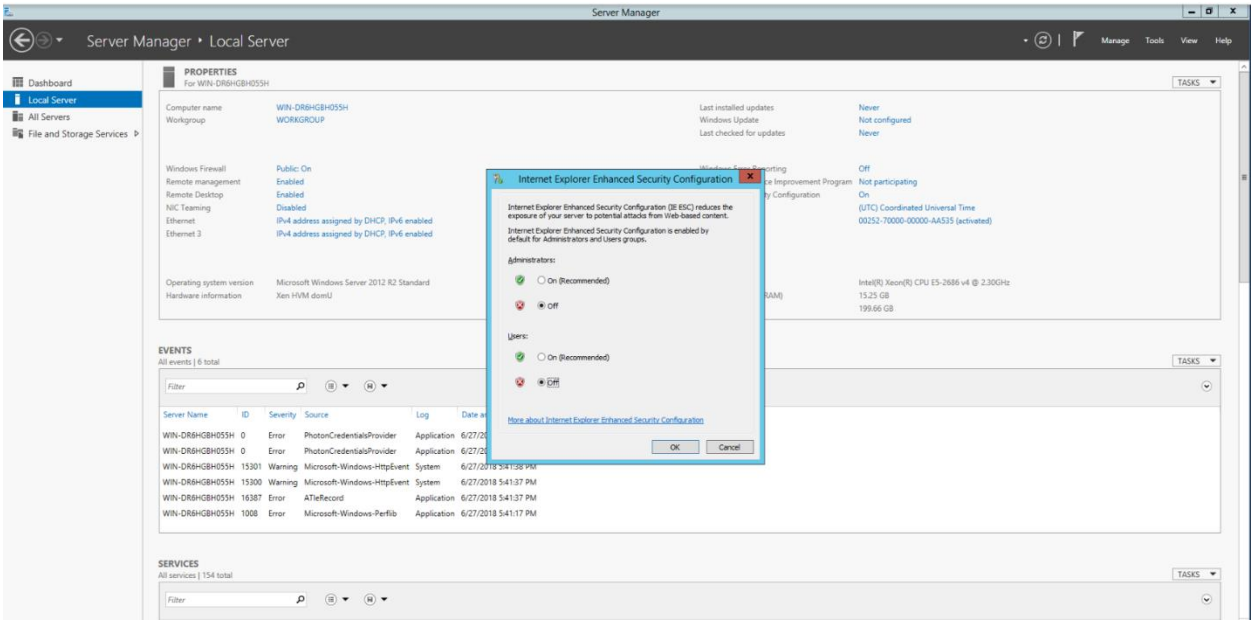
Download the full NX product version that you want every instance to use for your fleet.

The NX Installation Guide (nx_install.pdf) is located on the installation / download media at the top level under the 'docs' folder. Follow the installation prompts and configure the product accordingly.

Disable Internet Explorer Enhanced Security Configuration

NX applications use Internet Explorer to open http links embedded in the applications. When you launch one of these links, Internet Explorer displays a warning message for every webpage that it opens. This behavior is due to the Internet Explorer Enhanced Security Configuration, a security setting of IE that blocks access to web content and application scripts for security reasons. We can safely disable this setting to enable NX function properly. To disable this feature for AppStream 2.0 users, do the following.

1. Connect to your image builder as **Administrator.**
2. Open **Server Manager** from the Windows Task bar.
3. Choose **Local Server -> IE Enhanced Security Configuration.**
4. Choose Off option for both Administrators and Users.



5. Choose **Admin Commands -> Switch User -> Template User** to switch to Template User account.

6. Once you are logged into the Template User account, launch **Internet Explorer.**

7. Choose **Settings** -> **Internet Options** from the top right menu of Internet Explorer. In the prompted dialog, choose **Advanced.**

8. Click the **Reset** button. Click **Reset** again in the prompted dialog. Close **Internet Explorer.**

9. Switch to **Administrator** account. Launch **Image Assistant.** Choose **Next** to proceed to the Configure step. From this tab, click **Save settings.** This will save the template user settings as default user settings.

10. Switch to test user. Launch Internet Explorer. Confirm that the message “Internet Explorer Enhanced Security configuration is not enabled” is displayed in the home page. Browse to any website to confirm that IE is not displaying any blocking prompts.

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

At this point, you have launched an image builder instance and installed NX on the image builder. Now you'll prepare the applications for streaming, optimize them for streaming performance, and create your image.

In this section, you'll do the following:

- Create an application catalog by using Image Assistant.
- Test the applications by using a local user account that has the same permissions that end users will have in their streaming sessions.
- Optimize the applications' 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.
2. In the **Add Applications to Image** dialog box, on the **Add Apps** tab, choose **Add App**.

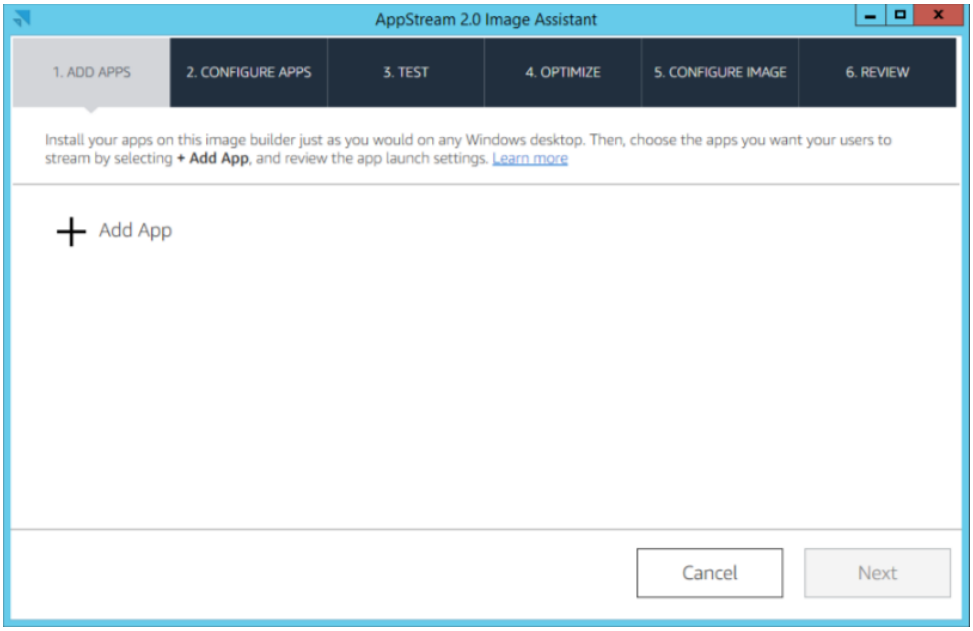


*Figure 12: The **Add Applications to Image** dialog box in Image Assistant.*

3. Navigate to the location of the NX application executable (ugraf.exe), select the application executable, and then choose **Open**.

4. In **Edit Application Setting**, type the following information and choose **Save**.

Option	Value
Name	NX install directory/ugraf.exe
Display Name	The name of the application that is displayed to end users. Type *NX.*
Launch Path	The location of your application executable file. Accept the default value.
Icon Path	Accept the default value of C:\ProgramData\Amazon\Photon\AppCatalogHelper…
Launch Parameters	Leave this blank.
Working Directory	Leave this blank.

5. Now that you have added NX to your catalog, 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 **NX**, **Launch**.
2. Wait for NX to completely start, as prompted by a message in the application.
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	The unique name identifier for the image, such as *DemoCo_Image_Chrome_Notepad_v1_mmddyyyy*, using any of the following characters: a-Z,0–9,-,_,. Note: The name cannot begin with "Amazon," "AWS," or "AppStream."
Display Name	A user-friendly name to display in the console
Description	An optional description for the image: for example, *Image v1 created by (your initials or name) on mm/dd/20yy.*
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.

Finish creating the image

Complete the following steps to disconnect from the remote session and start the image creation process.

1. Review the image details, and choose **Disconnect and Create Image**.
2. The remote session disconnects within a few moments. When the **Lost Connectivity** message appears, close the browser tab.

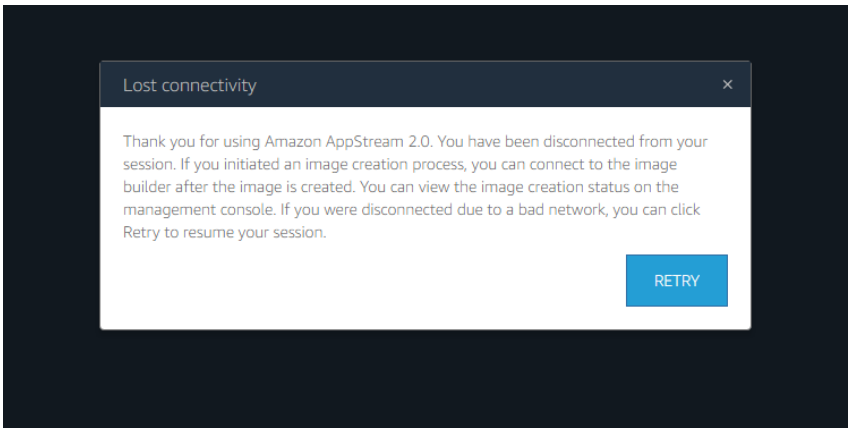


*Figure 13: The **Lost connectivity** message indicating that the image creation process has started.*

3. 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.
4. 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 7: 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: Provide Fleet Details**, type the following text and choose **Next**.

Option	Value
Name	The unique name identifier for the fleet, such as *DemoCo_Fleet_v1_mmddyyyy*, 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 *DemoCo Fleet v1 April 2018*.
Description	An optional description for the fleet. For example, *Fleet v1 created by (your initials or name) on mm/dd/20yy.*

Choose an image

For Step 2: Choose an image, choose the image that you created, scroll to the bottom of the page, and then choose **Next**.

Configure the fleet

1. **For Step 3: Configure fleet**, in **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 Graphics Design family, the instance type is already populated. However, you can select any of the four instance type options that are presented.

2. For this exercise, select the **Graphics Design** instance family, and then choose **stream.graphics-design.xlarge.** For more information, see Amazon AppStream 2.0 Instance Families.

3. Under **Fleet Type details**, 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.

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

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 **8 hours**.

      - **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**.

5. Under **Fleet capacity**, set **Minimum capacity** to 2 and **Maximum Capacity** to 4.

   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.
   - The minimum capacity for your fleet is the minimum number of users who are expected to be streaming at the same time.
   - The maximum capacity for your fleet is the maximum number of users who are expected to be streaming at the same time.

6. 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-xxxxxxxx (AppStream2 VPC)**.
3. For **Subnet 1**, choose **subnet-xxxxxxxx | (10.0.1.0/24)**. This is the AppStream2 Private Subnet1.
4. For **Subnet 2**, choose **subnet-xxxxxxxx | (10.0.2.0/24)**. This is the AppStream2 Private Subnet2.
5. Choose **Next**.
6. 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**.

7. 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.

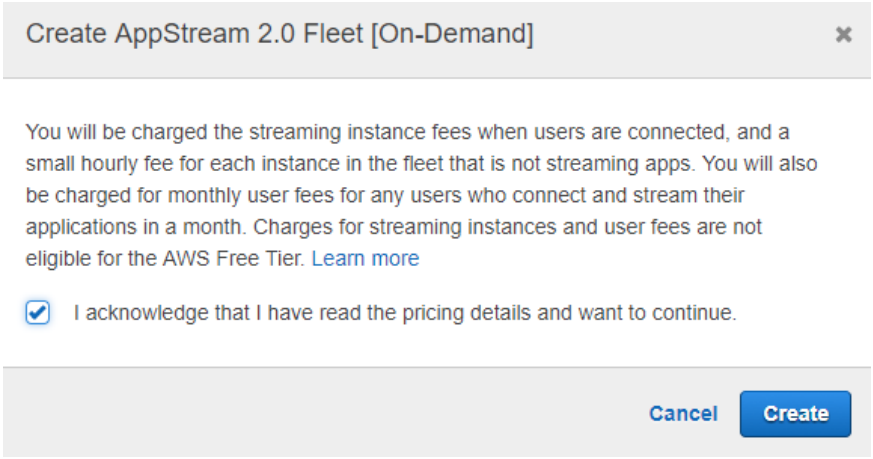


Figure 14: The AppStream 2.0 streaming instance pricing acknowledgement dialog box.

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. Choose the **Refresh** icon periodically to update the fleet status until the status is **Running**.

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

Step 8: 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 and choose **Next**.

Option	Value
Name	The unique name identifier for the stack, such as *DemoCo_Stack_mmddyyyy*, 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 *DemoCo Stack April 2018*.
Description	An optional text box where you can enter details of the stack:
Redirect URL	An optional URL to which users are redirected at the end of their streaming session. Type: *https://aws.amazon.com*
Fleet	Select the *DemoCo_Fleet_v1_mmddyyy* fleet that you created.

Enable persistent storage for the stack

4. 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](#).

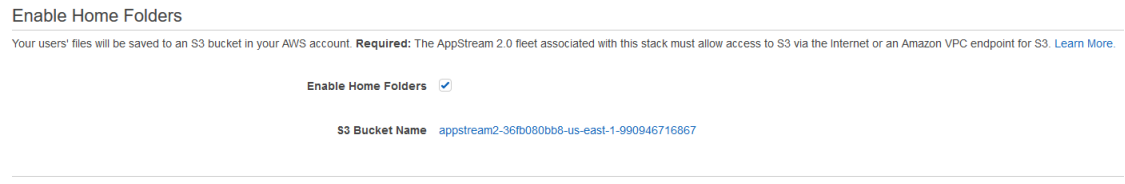


*Figure 15: The **Enable Home Folders** page, displaying the Amazon S3 bucket that is automatically created.*

5. Choose **Review**.
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**.

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**.

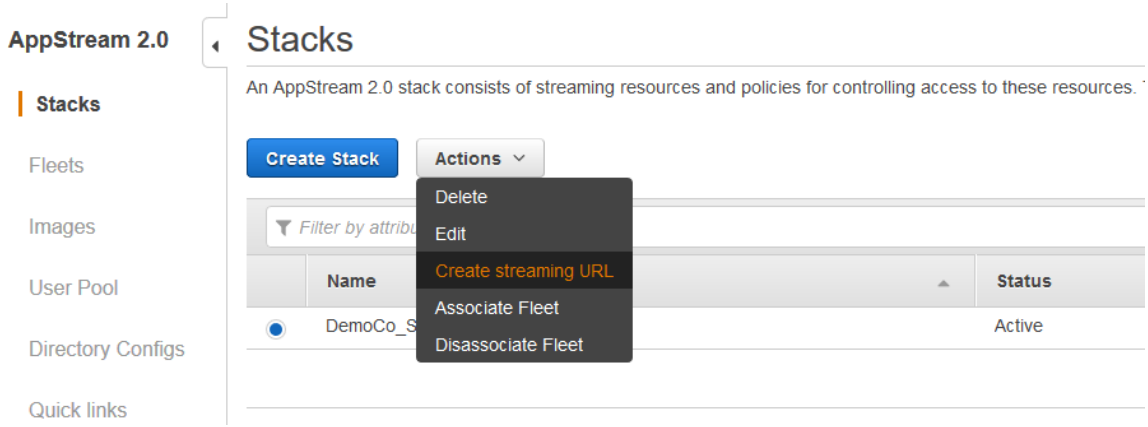


*Figure 16: The **Create streaming URL** menu item.*

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 *DemoCoTestUser1*. 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**.

5. The **Create streaming URL** dialog box refreshes, displaying the user ID that you entered and the URL that AppStream 2.0 generated for the user.

6. Choose **Copy Link** to copy the full URL to the clipboard.

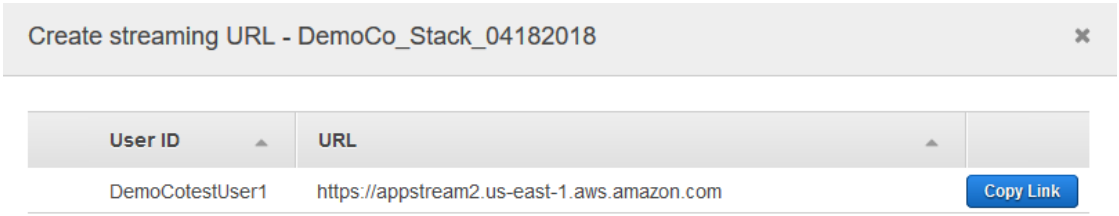


*Figure 17: The **Create streaming URL** dialog box.*

7. The **Create streaming URL** dialog box refreshes again, confirming that the link was successfully copied to the clipboard. Close 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.

8. 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.

9. Choose the NX icon to stream the NX application, and confirm that it functions as expected.

Step 9: 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** dialog box, 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.

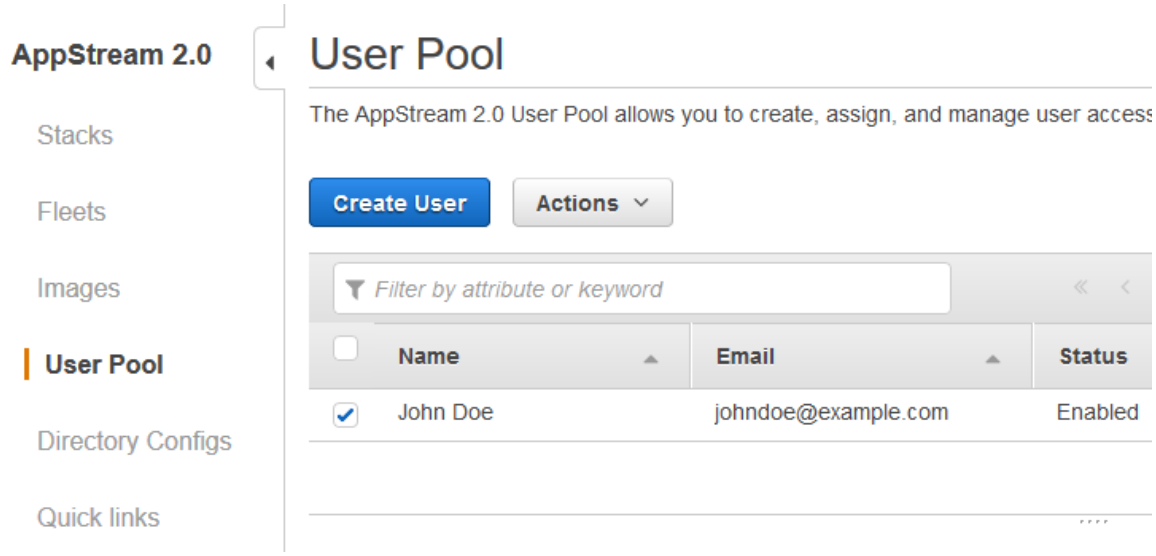


Figure 18: The User Pool dashboard showing the newly created user.

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.

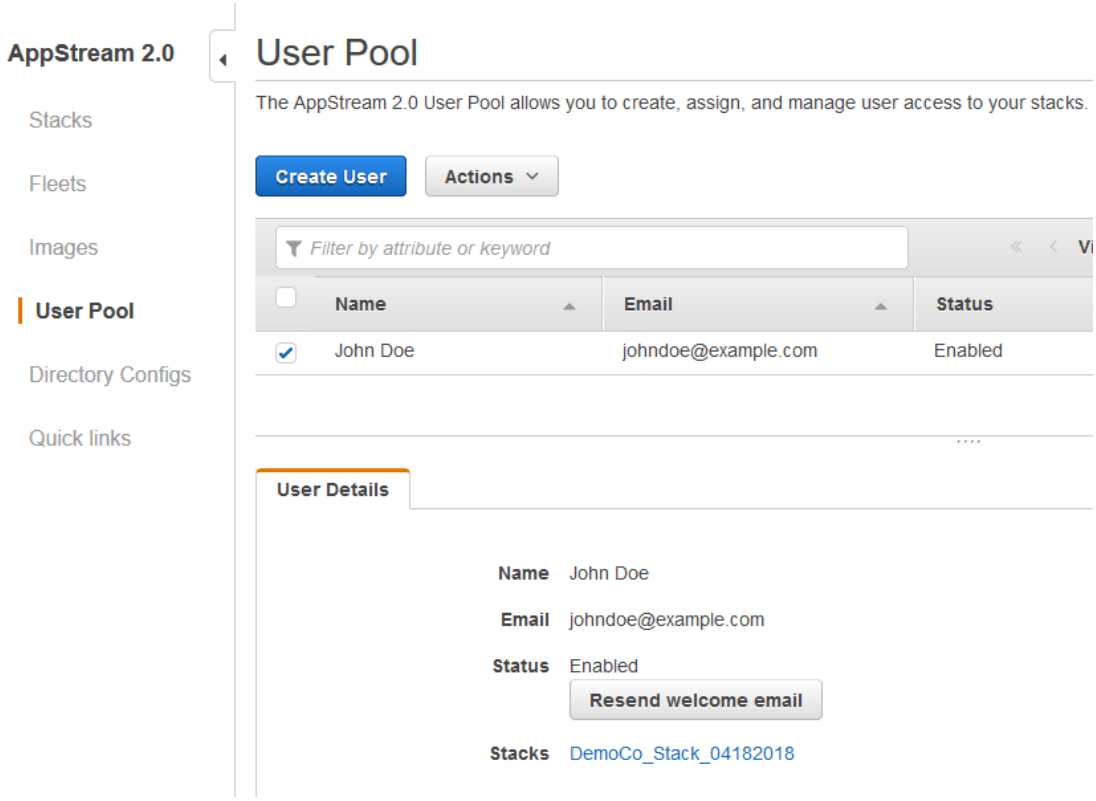


Figure 19: The User Pool dashboard showing the newly created user with a stack now assigned.

Step 10: 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.

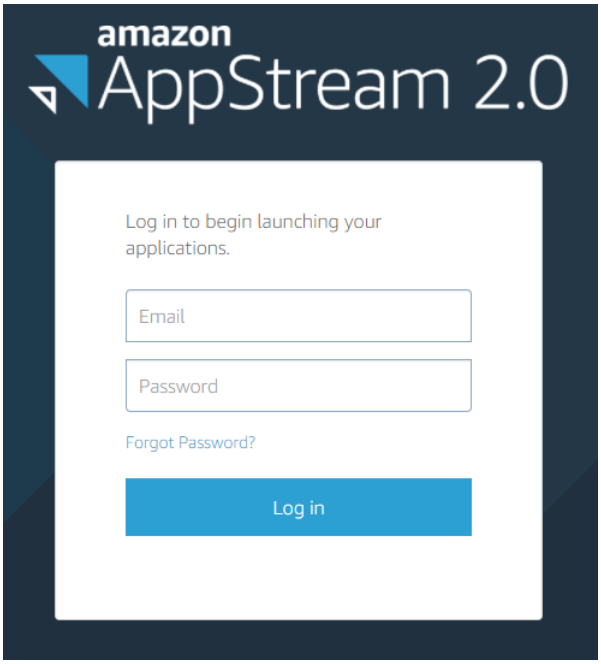


Figure 20: The AppStream 2.0 user login prompt.

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.

Step 11: Take the next step with AppStream 2.0

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

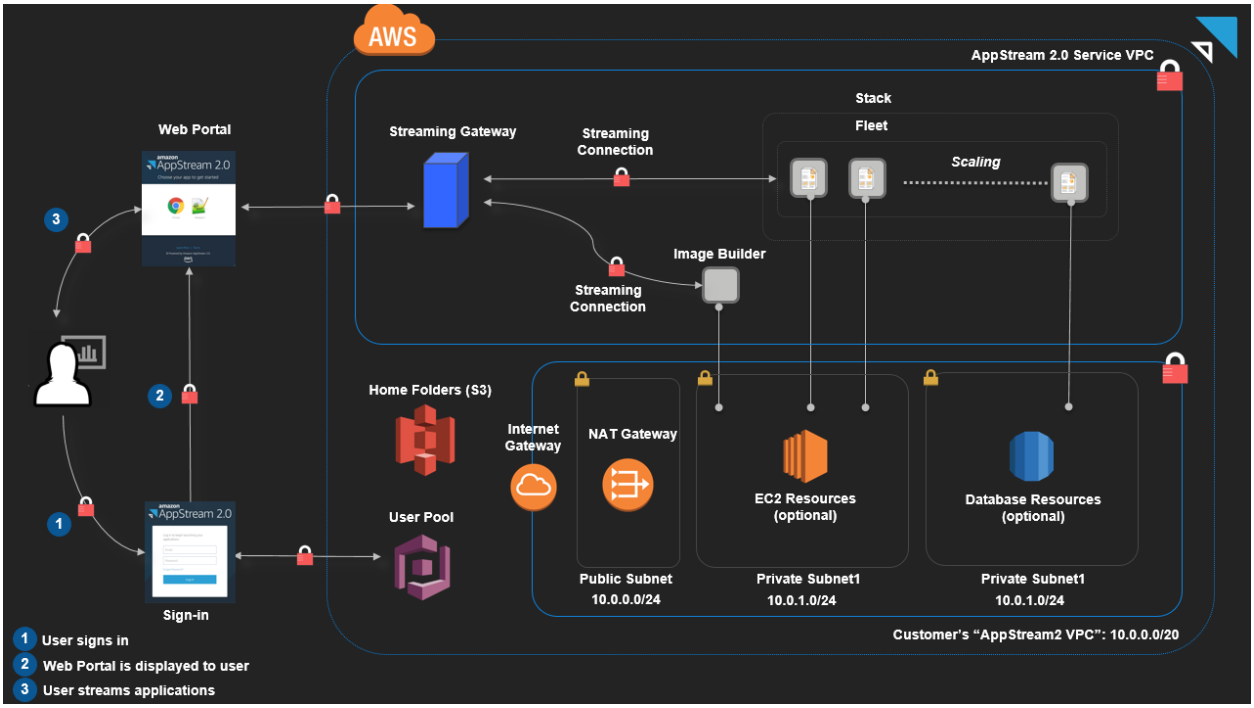


Figure 21: Your AppStream 2.0 environment.

This guide provided an introduction to AppStream 2.0 by walking you through basic configuration and deployment exercises for NX. 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.

Important: Remember to delete the resources that you created in these exercises to avoid further charges to your account. For information about how to delete AppStream 2.0 resources, see *Appendix E*. 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**.

Note: After you receive an email to confirm that your account is created, you can sign in to your new account by using the email address and password that you provided. However, you must continue with the activation process before you can use AWS services.

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 **Secure Submit**.

Otherwise, choose **Use a new address**, type the billing address for your payment method, and then choose **Secure Submit**.

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 **Basic**. For information about AWS Support, see AWS Support Features.

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:

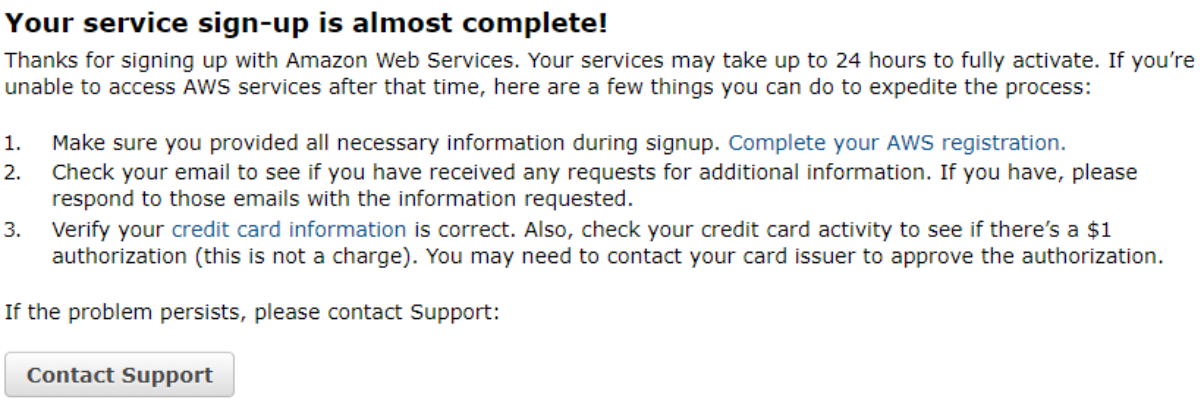


Figure 22: Message that appears if you sign in before your account activation is complete.

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. Manually create and configure network resources

Step 2 of this guide described how to use a CloudFormation template to automatically create and configure the necessary network resources for your AppStream 2.0 environment. To manually create and configure network resources, follow the steps in this appendix. At the end of this appendix, the topology of your “DemoCo” VPC should look similar to the following diagram:

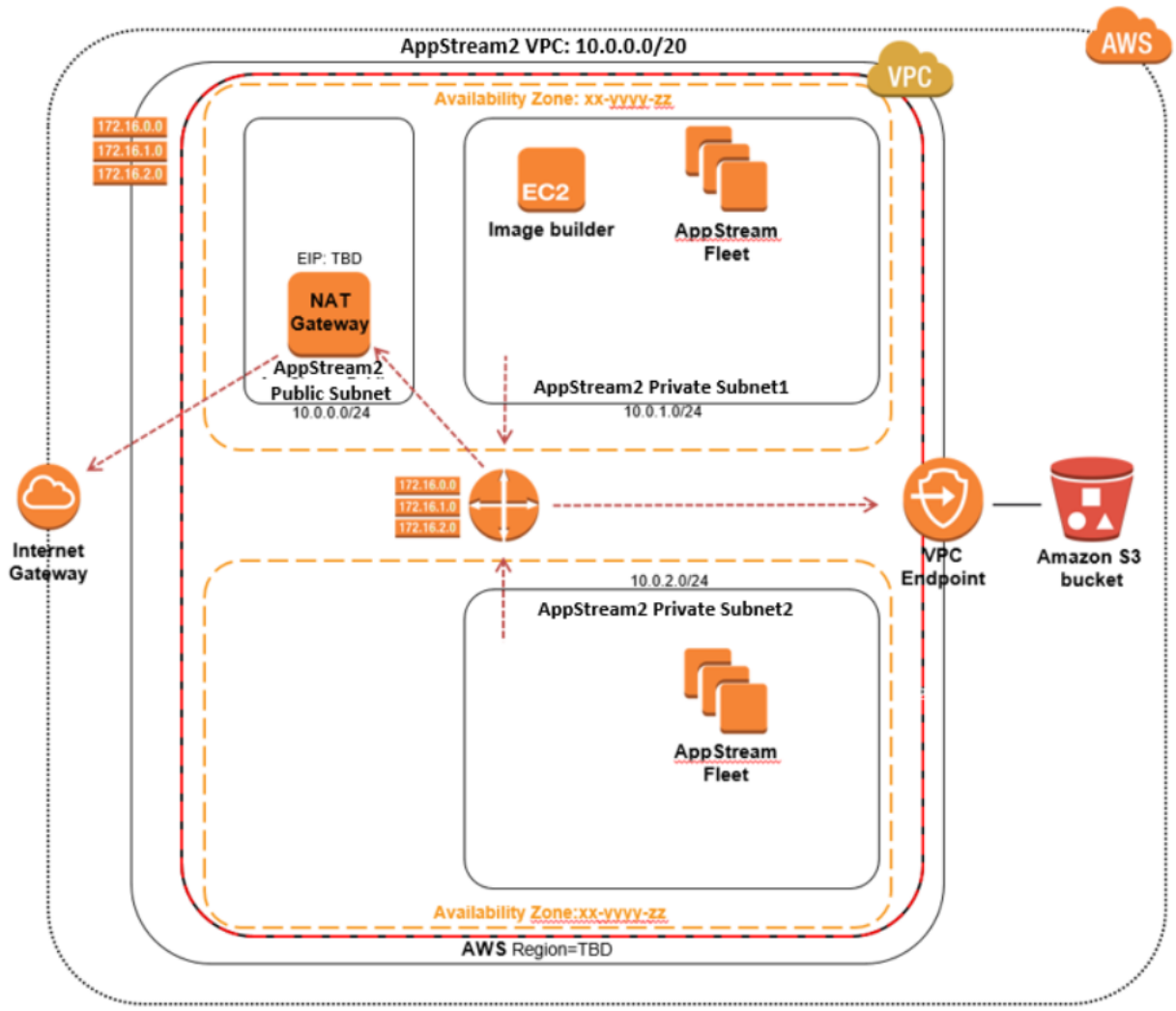


Figure 23: DemoCo VPC architecture.

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.

Allocate an Elastic IP address

Before you create your VPC, you must allocate an Elastic IP address in your AppStream 2.0 region (Note: Instances with a default Ipv4 address may not need this. Instances with a port route through a NAT server also may not need this.). An Elastic IP address enables your streaming instances to be accessible through an internet gateway.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. In the navigation pane, under **Network & Security**, choose **Elastic IPs**.
3. Choose **Allocate New Address** and then choose **Allocate**.
4. Note the Elastic IP address and then choose **Close**.

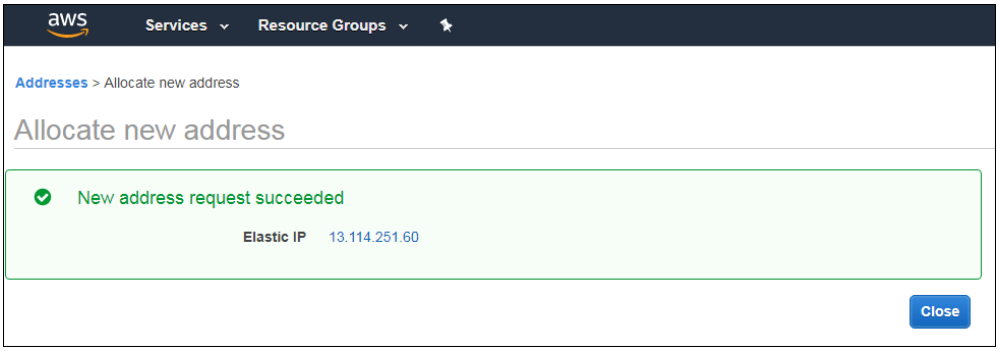


Figure 24: An allocated Elastic IP address.

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**, **Start VPC Wizard**.

3. In **Step 1: Select a VP Configuration**, choose **VPC with Public and Private Subnets**, **Select**.

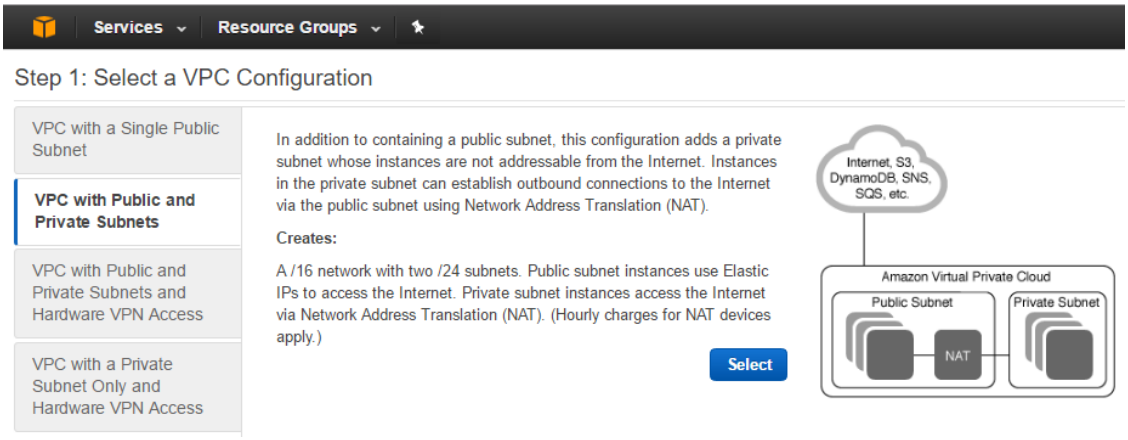


Figure 25: Creating a VPC with the VPC Wizard.

4. In **Step 2: VPC with Public and Private Subnets**, type the following information and then choose **Create VPC**.

Option	Value
IPv4 CIDR block	*10.0.0.0/20*
IPv6 CIDR block	Accept the default value: *No IPv6 CIDR Block*
VPC Name	*AppStream2 VPC*
Public subnet's IPv4 CIDR	Accept the default value: *10.0.0.0/24*
Availability Zone	Accept the default value: *No Preference*
Public subnet name	*AppStream2 Public Subnet*
Private subnet's IPv4 CIDR	Accept the default value: *10.0.1.0/24*
Availability Zone	Accept the default value: *No Preference*
Private subnet name	*AppStream2 Private Subnet1*

Option	Value
Elastic IP Allocation ID	Click in the text box and select the value that corresponds to the Elastic IP address you created. This address is assigned to the NAT gateway.
Service endpoints	Choose **Add Endpoint**.
Service	Select the entry in the list that ends with “s3” (the `com.amazonaws.xx-rrrr-x.s3` service that corresponds to the region in which the VPC is being created). **Note:** This is not the default value.
Subnet	Select *Private subnet.*
Policy	Accept the default value: *Full Access*
Enable DNS hostnames	Accept the default value*: Yes*
Hardware tenancy	Accept the default value: *Default*

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

5. After a few minutes, when a message in the VPC dashboard notifies you that the VPC is created, choose **OK**.

Add a second private subnet

1. In the navigation pane, choose **Subnets**.
2. Select the subnet with the name **AppStream2 Private Subnet1**. On the **Summary** tab, below the list of subnets, make a note of the Availability Zone for this subnet.

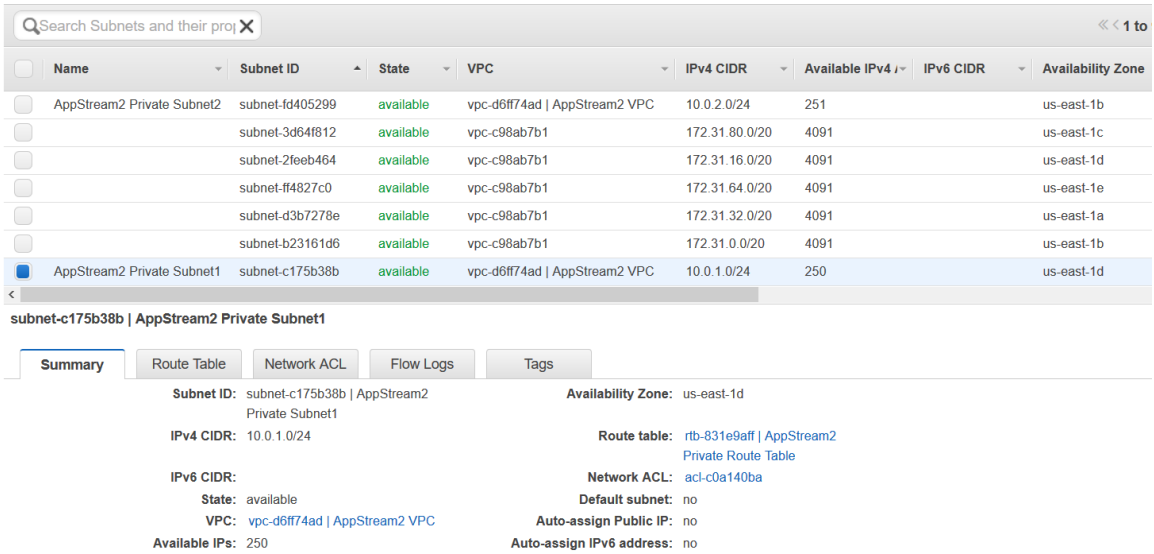


Figure 26: Identifying the Availability Zone for AppStream2 Private Subnet1.

3. At the top of the same page, choose **Create Subnet**. Enter the following information in the **Create Subnet** dialog box and then choose **Yes, Create**.

Option	Value
Name tag	*AppStream2 Private Subnet2*
VPC	Select the VPC with the name *AppStream2 VPC*.
Availability Zone	Select an Availability Zone other than the one you are using for AppStream2 Private Subnet1. Selecting a different Availability Zone increases fault tolerance.
IPv4 CIDR block	*10.0.2.0/24* (This is a subset of the CIDR block for your VPC.)

Modify the subnet route tables

1. In the navigation pane, choose **Subnets**, and then select the subnet with the name **AppStream2 Public Subnet**.

2. On the **Route Table** tab, note the ID of the route table (similar to *rtb-XXXXXXXX*).

3. In the navigation pane, choose **Route Tables** and select the route table with the ID that you noted in the previous step.

4. For **Name**, open the empty field, type *AppStream2 Public Route Table*, and then select the check mark to save your changes.

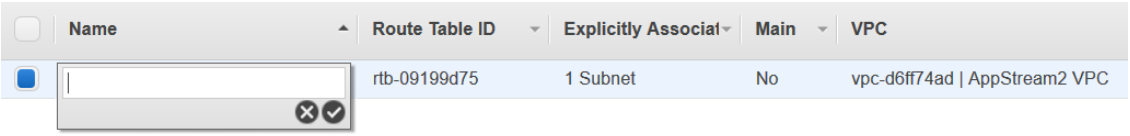


Figure 27: Opening the name field for the route table that serves the AppStream2 Public Subnet.

5. Make sure that **AppStream2 Public Route Table** is still selected. On the **Routes** tab, verify that the route table includes the following two routes:

Destination	Target
10.0.0.0/20	*local*
0.0.0.0/0	*igw-XXXXXXXX*

These two routes function as follows for all resources within a subnet that is associated with the route table:

- Local: All traffic from the resources destined for IPv4 addresses within the 10.0.0.0/20 CIDR block is routed locally within the VPC.
- Outbound: Traffic destined for all other IPv4 addresses is routed to the internet gateway (identified by *igw-XXXXXXXX*) that was created by the VPC Wizard.

To modify the route table, choose **Edit** and make the needed changes. For more information, see Route Tables.

6. In the navigation pane, choose **Subnets** and select the subnet named **AppStream2 Private Subnet1**.
7. On the **Route Table** tab, note the ID of the route table (similar to *rtb-XXXXXXXX*).
8. In the navigation pane, choose **Route Tables** and select the route table with the ID you noted in the previous step.
9. For **Name**, open the empty field, type *AppStream2 Private Route Table*, and then select the check mark to save your changes.

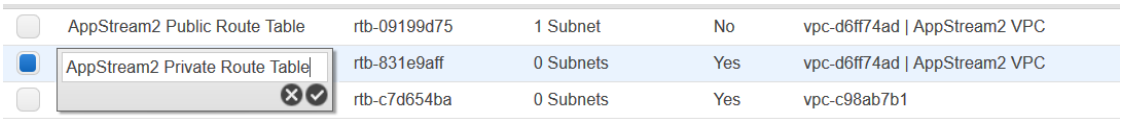


Figure 28: Opening the name field for the route table that serves the AppStream2 private subnets.

10. Make sure that **AppStream2 Private Route Table** is still selected, and on the **Routes** tab, verify that the route table includes the following routes:

Destination	Target
10.0.0.0/20	*local*
0.0.0.0/0	*nat-XXXXXXXXXXXXXXXX*
pl-YYYYYYYY (`com.amazonaws.<region>-<#>.s3`)	*vpce-ZZZZZZZZ*

These three routes function as follows for all resources within a subnet that is associated with the route table:

- Local: All traffic from the resources destined for IPv4 addresses within the 10.0.0.0/20 CIDR block is routed locally within the VPC.
- Storage: Traffic destined for S3 buckets is routed to the S3 endpoint (identified by *vpce-ZZZZZZZZ*).
- Outbound: Traffic destined for all other IPv4 addresses is routed to the NAT gateway (identified by *nat-XXXXXXXX*).

To modify the route table, choose **Edit** and make the needed changes. For more information, see Route Tables.

11. In the navigation pane, choose **Subnets** and select the subnet with the name **AppStream2 Private Subnet2**.

12. On the **Routes** tab, verify that the route table is the one named AppStream2 Private Route Table. If the route table is different, choose **Edit** and select this route table.

Appendix C. Configure Chrome

After you install applications on your image builder instance, you might need to configure them before creating your image. Examples of tasks or configurations to consider at this stage include:

- Applying patches or updates
- Managing auto-update functionality
- Creating startup or `run`/`runonce` commands
- Performance tuning
- Disabling application startup prompts, such as license prompts or first-run experiences.
- Configuring file associations
- Clearing or resetting log files or event logs

Chrome includes an automated update process that you can disable so that users aren't exposed to it. Use Windows Task Scheduler to disable it with the following steps.

Note: Although these steps are specific to Chrome, you may need to perform similar steps to disable updates for other applications that you configure for streaming through AppStream 2.0.

1. In your image builder instance, choose the Windows **Start** menu.
2. Choose **Control Panel**, **System and Security**, **Administrative Tools**.
3. Open **Task Scheduler**.
4. When Task Scheduler opens, expand the window to full screen for easier viewing.
5. On the left side, under **Task Scheduler (Local)**, choose **Task Scheduler Library**. Click the **Name** column in the center pane to sort the task names alphabetically. If necessary, widen the **Name** column to display the full names of the listed items.
6. Select the task **GoogleUpdateTaskMachineUA**. Choose **Action**, **Delete.** When prompted, choose **Yes**.

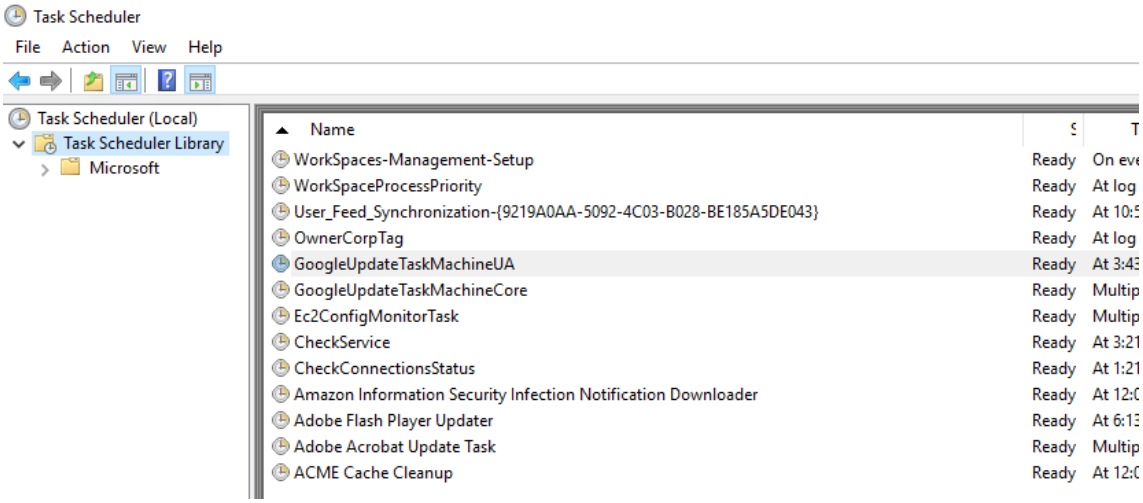


Figure 29: Windows Task Scheduler with the GoogleUpdateTaskMachineUA task selected.

7. Repeat step 6 to delete the **GoogleUpdateTaskMachineCore** task.

8. Close Task Scheduler.

9. Disable any auto update services. In the Administrative Tools window, which should still be open, open **Services**.

10. When the Services window opens, expand the window to full screen for easier viewing. In the list, open the first of two **Google Update Service** entries, **Google Update Service (gupdate)**.

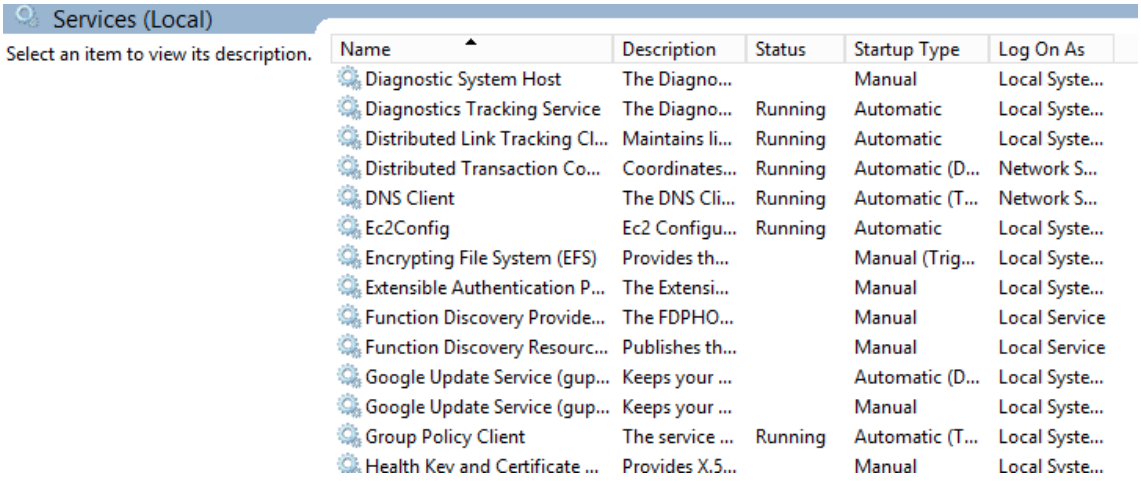


Figure 30: Windows Services, with the two Google Update services listed.

11. For **Startup type,** choose **Manual**.

12. Next to the **Service status** label, verify that the service shows the status **Stopped**. If not, choose **Stop**.
13. Choose **Apply, OK**.
14. Repeat steps 10 to 13 for **Google Update Service (gupdatem)**.
15. Close **Services** and any other open windows.

Appendix D. Install Notepad++

Step 4 in this guide described how to connect to an image builder and download the installer for Notepad++. To complete the Notepad++ installation process, follow these steps.

1. Connect to the *DemoCo_Image1_mmddyyyy* image builder that you created.
2. In the image builder, in File Explorer, open the Downloads folder and open **`npp.x.x.x.Installer.x64.exe`**.
3. In **Installer Language**, choose **English** if it is not already selected, and then choose **OK**.
4. In **Welcome to Notepad++ Setup**, choose **Next**.
5. In **License Agreement**, review the agreement and then choose **I Agree**.
6. In **Choose Install Location**, choose **Next** to accept the default destination folder for the installation and proceed.
7. In **Choose Components**, leave the default components selected, and then choose **Next**.
8. In the next **Choose Components** page, leave all three options unselected, and then choose **Install**.
9. In **Completing Notepad++ Setup**, clear the **Run Notepad++** check box so that it is not selected, and then choose **Finish**.

Appendix E. 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

5. In the navigation pane, choose **User Pool**.
6. 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

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

Stop and delete your fleet

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

11. 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**.
12. 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