Studio in the Cloud

Implementation Guide

Published May 14, 2020
Updated July 28, 2021

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

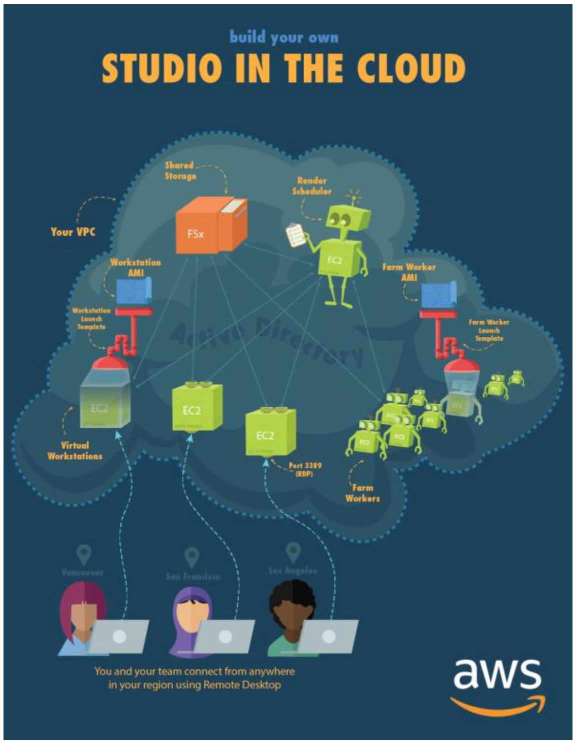
About this Guide

This getting started series is an entry level guide for VFX and Animation studios to help you get started on AWS. These tutorials are designed for small teams who are looking for ways to scale, work remotely, and improve collaboration. In this series we’ll walk through how to set up cloud-based virtual workstations, cloud storage, and cloud rendering.

Tutorial 1: Getting Started with AWS Virtual Workstations

Estimated Time to Complete: 40 minutes

Overview



Welcome to tutorial 1 from our Studio in the Cloud series. This step-by-step series is written for creative studios looking to fully adopt the cloud for producing content. The series has been created by artists including animators and compositors, so it's approach will resonate with an audience less familiar with cloud technology and using the AWS console. This tutorial series walks you through setting up virtual workstations, cloud storage, and cloud rendering. By the end of the seven-part series, you'll have a fully cloud-based studio that leverages the scale, power, and convenience of AWS.

Each tutorial starts with an overview, like this one, as well as an estimate of the time required to complete. There are also exercises throughout to confirm that you have correctly completed the previous steps. Supplemental information about the concepts covered in each tutorial and links to additional websites appear at the end of each tutorial.

Prerequisites

Before you begin, check the following prerequisites to make sure that you'll be able to successfully complete the tutorials.

Region Support

Before you jump into creating your Studio in the Cloud, you should make sure that the geographic region you are in (or the one closest to you) supports all of the services and instance types required.

A Region represents a specific geographic area around the world where AWS maintains a network and servers. Each Region has one or more Availability Zone, which are isolated locations inside the Region. For the best performance, you want AWS services to be located as geographically close to you as possible. With over 20 Regions, there is a good chance that there is a Region located near you. Having multiple Availability Zones in a Region provides additional reliability.

At the time of publication (February 2020), the following AWS Regions fully support all of the services needed for Studio in the Cloud:

- **N. Virginia (us-east-1)**
- **Ohio (us-east-2)**
- **N. California (us-west-1)**
- **Oregon (us-west-2)**
- **Singapore (ap-southeast-1)**
- **Sydney (ap-southeast-2)**
- **Tokyo (ap-northeast-1)**
- **Frankfurt (eu-central-1)**
- **Ireland (eu-west-1)**
- **London (eu-west-2)**
- **Stockholm (eu-north-1)**

If you're not sure which Region is closest to you, see the Regions and Availability Zones webpage. You may also want to use a website such as ping.psa.fun to measure the latency from your location to the different Regions. The Region with the lowest latency value is usually the one you want.

Note: If your nearest Region does not have all of the required services, there may be another nearby Region that does. In addition, Regions are being updated all the time to add support for additional services and instance types. So in time, your closest Region may become supported. See the Appendix for information on the specific services that are required and links to webpages with the most up-to-date information on whether they are available in your nearest Region.

AWS Account

Now that you know whether your nearest AWS Region is supported, create an AWS account.

If you already have an AWS account, is it your personal account or a linked account provided by your employer? If it's your personal account, then be aware that there are costs associated with using the AWS services required for these tutorials. We provide a rough estimate of those costs in the next section.

If your account is linked to your employer or another party, check that they are prepared to accept responsibility for the costs involved. For more information on linked accounts and consolidated billing, see Consolidated Billing Process.

You must also verify that your account has sufficient permissions to access all of the AWS services required to complete the tutorials.

Account Permissions

If you are the owner of your account

If you just created a new AWS account or are the owner of your account, you may be logging in as the root user of your account. (If you're logging in using an email address, rather than a user name, you're using the root account.)

> ⚠ *Note:* Rather than continue to login as the root user, we strongly suggest that you create a new user and login with those credentials instead. If your root user email and password are ever stolen, then the person who stole them gains full access to your account and you will need to close the account. Creating a new user for yourself or others on your account allows you to have more control. You can specify exactly what permissions each user has and if that user information is ever lost, you can simply remove that user from your account.
> If you're unfamiliar with how to create a new user and add permissions to it, see Creating an IAM User.

Accounts provided by a third party

If your account is provided by your employer or a third party, you're likely already logging in as a user and not as the root. In this case, you do not need to create a new user, but you may need to request to have additional permission policies added in order to complete the tutorials. Be aware that the owner of your account may have rules about which policies you can access.

Click the following link to a JSON file which lists of all the permissions needed for the tutorials: Tutorials Permissions Policy. It's best to check with the administrator for your account and request that these permissions be added to your user.

On-Demand G Instance Quota

Many resources in AWS are subject to service quotas, also called limits. Each resource has a quota that represents the maximum value of that resource you can use. These quotas are in place to ensure that AWS can provide highly available and reliable service to all our customers, but also to protect you against accidentally creating too many resources and incurring unexpected charges.

Our tutorials make use of GPU-enabled virtual workstations (G4 and G3 instances). If you are using a new AWS account or have never used GPU instances before, your quota for that instance type may be at the default value of zero. Before starting the tutorials, you should check the value of your G instance quota and request an increase, if necessary. See also the Service Quota Documentation to learn more about service quotas.

Microsoft Remote Desktop

For simplicity, we'll be using Remote Desktop to connect to your Windows instances. There are other options for connecting, such as NICE DCV from AWS or Cloud Access Software from Teradici. These solutions provide performance benefits over Remote Desktop, but require some extra setup which is beyond the scope of these tutorials.

If your local computer is a Windows machine, Remote Desktop should already be installed. But if you're on macOS or Linux, you may need to install it. Here are links to download the latest version for Windows, macOS and Linux.

Estimated Costs

AWS services are charged using a pay-as-you-go model. That means that you only pay for the individual services you use, for the time that you use them. For example, virtual workstations are charged per hour for each hour that they are running (whether you are logged in or not), while storage is charged based on the type and amount of storage you use.

In addition, some services are free to use up to certain limits, while others may be free for a certain amount of time after you initially sign up for an account. After you have exhausted the AWS Free Tier usage, you switch to paying for what you use.

Assuming that you complete the tutorials in a time frame similar to what is estimated for each step, and that you require a minimum amount of storage and compute time, we estimate that the cost to complete the tutorials will not exceed $150-200 USD. The total cost of your Studio in the Cloud depends on how many hours it is running and how much of each service you consume. Items such as shared cloud storage and render hours can vary widely depending on your particular use case. In addition, costs for AWS services differ slightly from Region to Region.

For a more accurate estimate of your cost, we recommend using the AWS Pricing Calculator, which factors in the pricing for your particular Region. We have listed the AWS services used in the tutorials, along with approximate usage amounts, in the Appendix.

Our estimate above also does not account for any existing credits that you may have in your account. In addition, if you decide to keep your Studio in the Cloud infrastructure running after completing the tutorials, you will continue to accrue costs based on how much of each AWS service you consume. Because the costs will vary depending on your individual needs, we do not attempt to estimate those costs here. The tutorials do not make use of any licensed software, so any licensing costs for additional software that you choose to use are also not covered.

Monitoring Your AWS Costs and Usage

As you work through the tutorials, you can keep an eye on your current costs using the AWS Billing Console. In addition, AWS has many other tools for budgeting and tracking usage. See Keeping an Eye on Your AWS Costs and Usage for more information and helpful tips.

Security

Throughout these tutorials we call out security best practices, such as not logging in as the root user, limiting access with security groups, and others. However, the intention of the tutorials is introduce you to the key concepts and services of Studio in the Cloud, rather than dive deep on security.

> ⚠ *Note:* While the security measures implemented here are sufficient for initial setup and testing of a cloud-based production pipeline, we recommend

> implementing extra security after completing the tutorials, according to your individual security compliance goals for production content.

Keeping Track of Important Information

Many times in these tutorials you are asked to refer back to information about components you created in a previous step, including names, IDs, IP addresses and more. To make it easier to keep track of this information, we've created an Important Information Cheat Sheet for you that you can fill out in a PDF editor or print out and fill out by hand.

Starter Exercise: Using the AWS Console

Ready to get started? We're going to begin by introducing you to the **AWS Management Console**, which is your main interface to all AWS services.

For this first exercise, you launch an EC2 instance. Amazon Elastic Compute Cloud (Amazon EC2) is the AWS service that you use to launch the virtual computers that you use for your Studio in the Cloud.

An **instance** is a single virtual computer in EC2. Throughout these tutorials we'll be creating several instances for different purposes, including user management, render management, virtual workstations, and cloud farm workers.

For your first instance, you use some of the default settings that already exist in your account.

Login to Your Account

If you haven't already, login to your AWS account.

As a reminder, make sure you login as an IAM user and not as the root user for your account. If you need to create a new user, see Creating an IAM User.

Fill Out Your Cheat Sheet

In the Important Information Cheat Sheet, record your AWS Account email address and account ID or alias, as well as your IAM user name.

> ⚠ *Note:* You also need to remember your password, but for security reasons we don't recommend writing that down on the cheat sheet.

- If you downloaded your credentials.csv file, note the file location for later reference.

Set Your Region

Next, you should check that your Region is set correctly. After signing in, your current Region is listed in the top right corner of the AWS Management Console navigation bar.

- To change your Region, choose the current Region and then select a different Region from the list.
- As a reminder, for these tutorials you must select a supported Region from the list above.
- *Record the Region you are using (e.g. N. Virginia (us-east-1)) in the Important Information Cheat Sheet*.

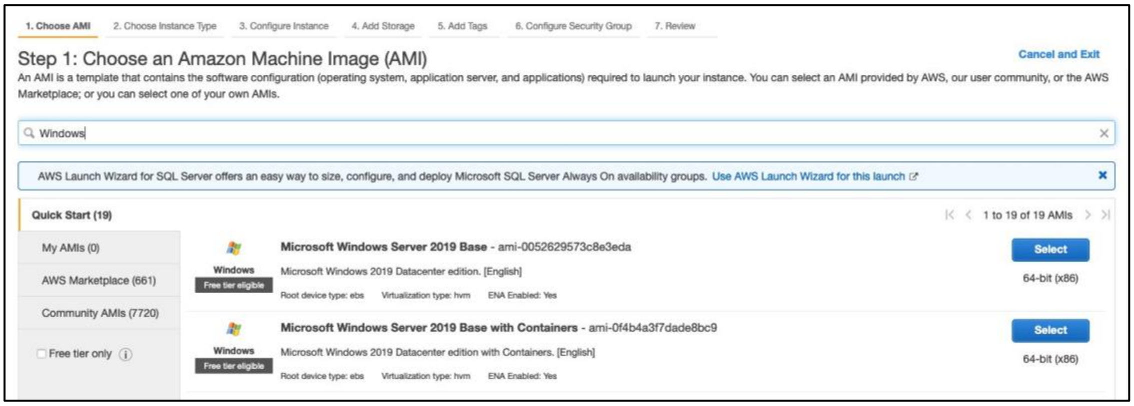
Launch an EC2 Instance

1. In the navigation bar, choose **Services,** then choose **Compute > EC2.**

   (Alternatively, in the search box, type **EC2**.)

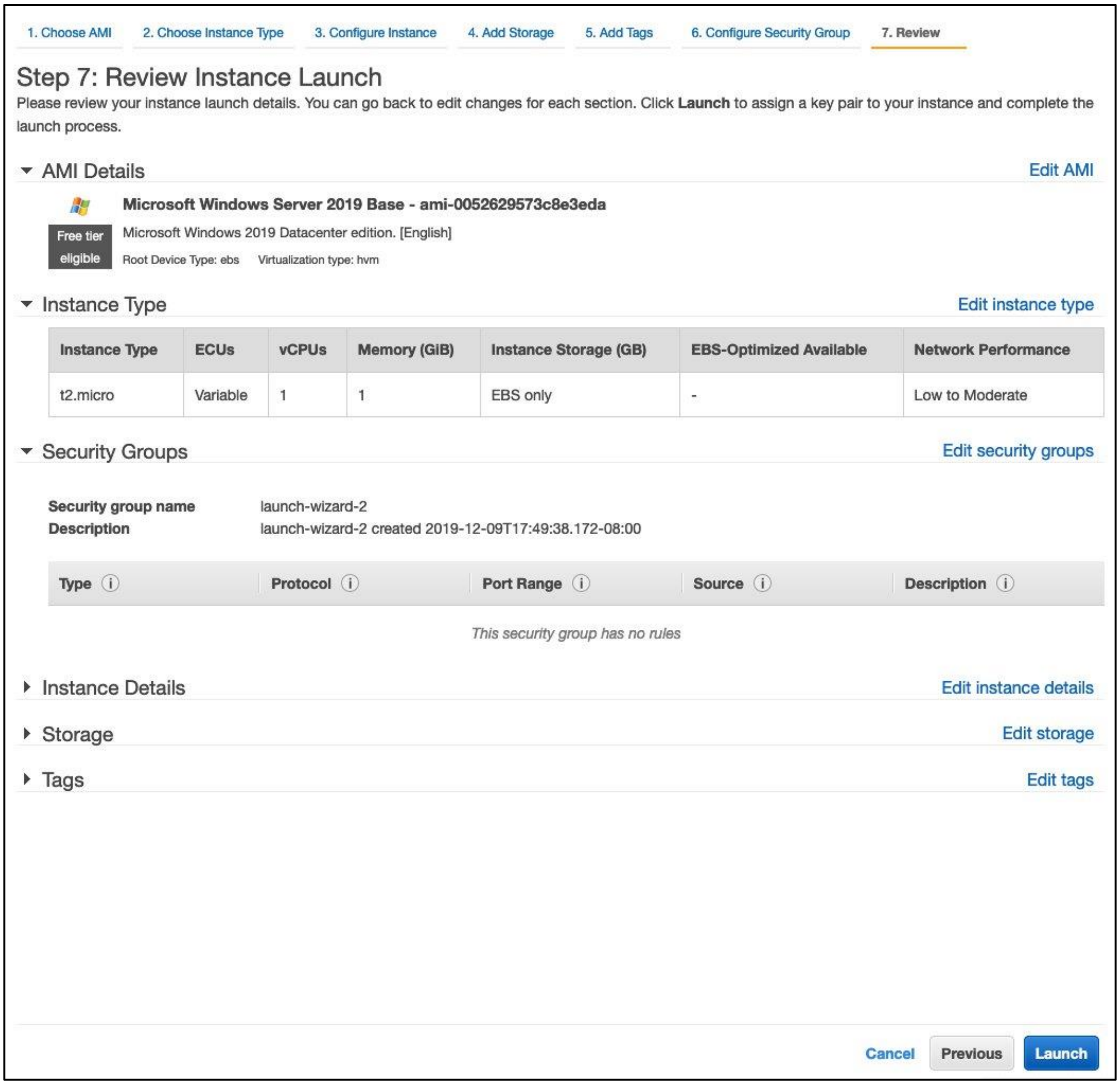
2. In the left navigation pane of the **EC2 Dashboard**, choose **Instances**.
3. On the **Instances** page, choose **Launch Instance**.
4. For **Step 1: Choose an Amazon Machine Image (AMI)**, in the search box, type **Windows**.

In the list of AMIs, find **Microsoft Windows Server 2019 Base**, and choose **Select**.



5. For **Step 2: Choose an Instance Type**, keep the default values and choose **Review and Launch**.

6. For **Step 7. Review Instance Launch**, choose **Launch**.



7. In the **Select an existing key pair or create a new key pair** window, choose **Create a new key pair** from the first drop down menu and then enter a key pair name (e.g., **mystudio-keypair**)

   A **key pair** consists of public and private key files that are used to encrypt data between two computers. AWS stores the public key file, but you need to store the private key file in order to connect to an AWS Cloud workstation.

8. After entering your key pair name, choose **Download Key Pair**.

Important: You must download the private key file when you create a new key pair and store it securely on your local computer. You do not have another chance to download the private key file, so save it in a safe place on your computer where you can find it again!

You should also note the name of the key pair file in the *Important Information Cheat Sheet* for future reference.

9. After downloading the private key file, choose **Launch Instances**.

View the Status of Your Instance

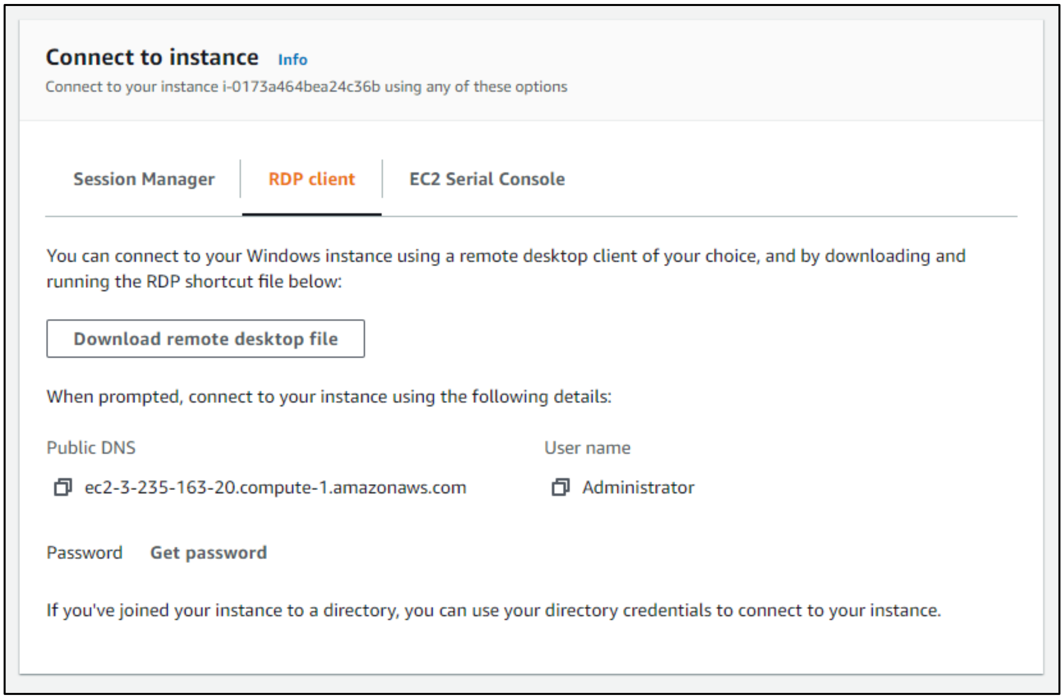
1. On the **Launch Status** page, choose **View Instances**.

   The **Instances** page opens displaying a list of your running instances and their status. When you see the **Instance State** change from **pending** to **running** and the **Status Checks** change from **Initializing** to **2/2 checks passed**, your new instance is ready to be used. It may be between 5 to 10 minutes for the machine to be ready.

2. Choose your instance to see more details in the panel at the bottom of the screen.

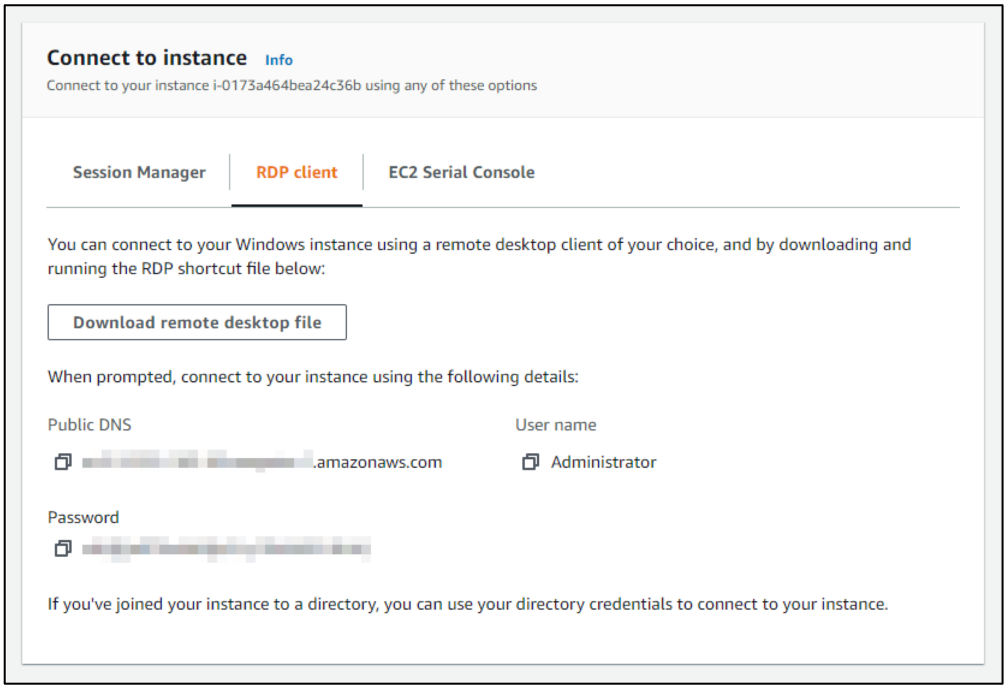
Connect to Your Instance

1. On the instances page, select your instance and then choose **Connect**.



2. Choose the **RDP client** tab and then choose **Get password**.
   a. Click **Browse**, then open the private key file that you created earlier.
   b. Choose **Decrypt Password**.
3. Click the **copy** icon to the left of the password to copy it to your clipboard.

4. Choose **Download remote desktop file**.



   Your browser downloads an .rdp file that the **Remote Desktop Connection** application uses to connect to your instance. Depending on your browser, you must either click the downloaded .rdp file manually or click **OK** in another popup window when you're asked if you want to open it automatically.

5. On Windows, choose **Connect** in the popup window that appears. On a Mac, choose **Continue**.

6. In the prompt for an Administrator password, paste the password that you copied into the password field and choose **OK.**

7. When a window pops up that says **The identity of the remote computer cannot be verified** click **Yes** (or **Continue** if on a Mac) to continue connecting.

   A new window opens that displays the desktop of your cloud workstation/instance.

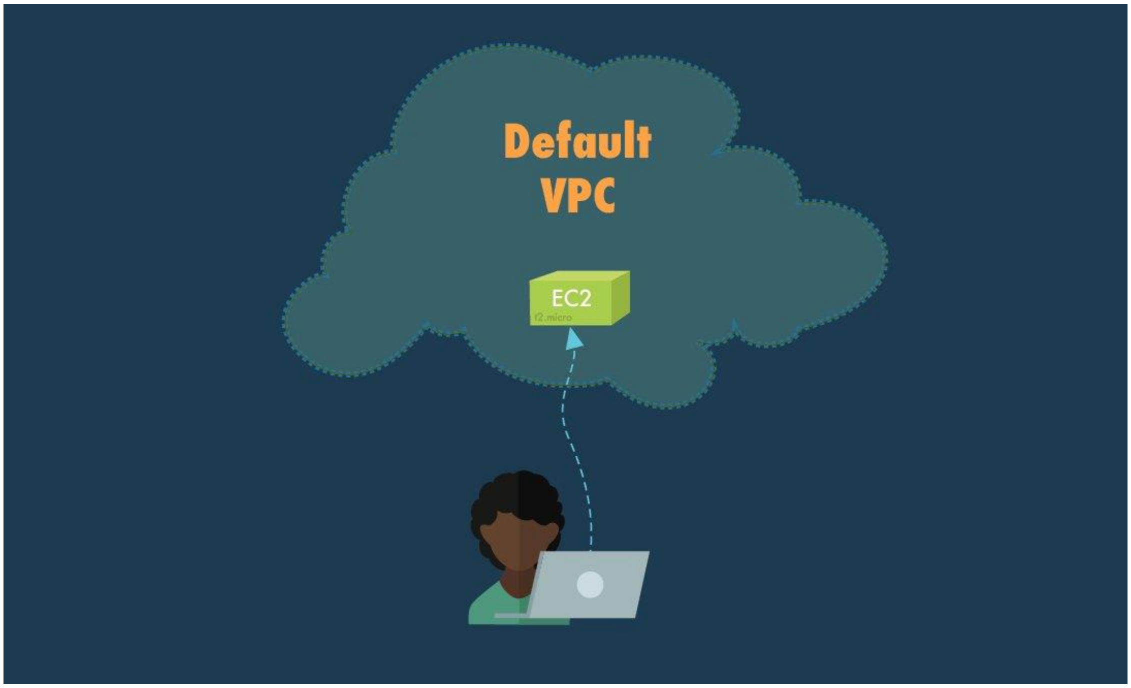
You can interact with the Windows desktop of your instance and run applications just as if it were on your local computer.

Note: By default Remote Desktop tends to fill the entire screen that it runs on. If you have a single monitor, this can make it difficult to view both these tutorial pages and the Remote Desktop at the same time. Click the **Restore** icon and resize the Remote Desktop window as needed:

You can also force Remote Desktop to start up at a resolution smaller than full size. For instructions, see the Appendix.

Your First EC2 Instance

Congratulations! You have launched your first AWS EC2 instance!



Take a moment to marvel at what you have accomplished...but don't marvel too long. Once you've finished taking your instance for a test drive, you will want to disconnect and shut it down. Why is that? Move on to the next section to find out.

Cost Optimization

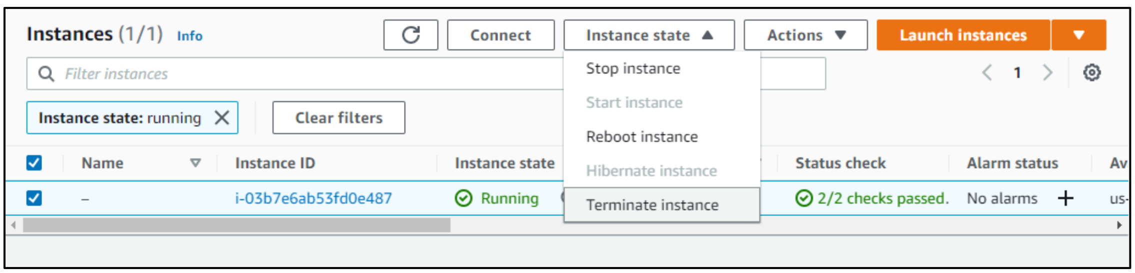
In an effort to keep your costs down as much as possible, at the end of each tutorial we'll provide instructions on which services you can shut down. This may be because they aren't needed anymore, like in the case of the instance you launched during the starter exercise above. Or, it may be because it's going to be a day or two before you're able to move on to the next tutorial and you don't want to have running computers costing you money when you're not using them.

Below we go over the procedure for stopping and/or terminating an instance. *Stopping* an instance is similar to shutting down your home or work computer. All running applications quit, but whatever data was stored on persistent storage (i.e., the instance's hard drive) is preserved. Once the instance is stopped, you are no longer charged any hourly fees for its use (although you are charged a small amount to store the data that was on the instance's hard drive). A stopped instance is still listed in the EC2 instance list in the AWS Management Console and can be restarted at any time.

Terminating the instance is more final than stopping it. When an instance is terminated, it is stopped like above, but all the data that was stored in persistent storage is also deleted, so not only do you not incur any hourly charges, you also do not incur any charges for storage of data. You can think of it as first shutting down your computer, but instead of keeping it around for later, you send it to be recycled. Terminated instances remain visible in the console for a little while, but unlike stopped instances, they cannot be restarted.

Terminate Your Instance

1. On the **Start Menu** of your instance, choose the **Power Icon,** then select **Disconnect**. The Remote Desktop Connection app closes.
2. In the **AWS Management Console**, click **Cancel** in the Connect to instance window. Then select your instance from the list of instances.
3. Choose **Instance state** and then choose **Terminate instance**. Finally, choose **Terminate.**



The steps for stopping your instance are the same as above, except instead of Instance State >Terminate, you select Instance State > Stop. In this case, we don't need the start exercise instance anymore, so we can terminate it.

Throughout these tutorials, you'll be launching and connecting to many different instances, so you'll be starting and stopping instances on a regular basis. This first starter exercise is good practice for the steps yet to come.

In the next tutorial, we'll begin building the basic infrastructure for your Studio in the Cloud. Instead of using the default settings for your account, as we did here, we'll be adjusting the settings to optimize for a production workflow.

Appendix

Links to AWS Documentation

- AWS Services By Region
- AWS Instance Types (all Regions but Beijing & Ningxia)
- AWS Instance Types (Beijing & Ningxia only)
- Consolidated Billing Process
- AWS Free Tier
- AWS Pricing Info
- AWS Pricing Calculator
- AWS Regions and Availability Zones
- What is Service Quotas?
- Launching an Instance Using the Launch Instance Wizard
- Connecting to Your Windows Instance
- Stop and Start Your Instance
- Terminate Your Instance

Links to Other Resources

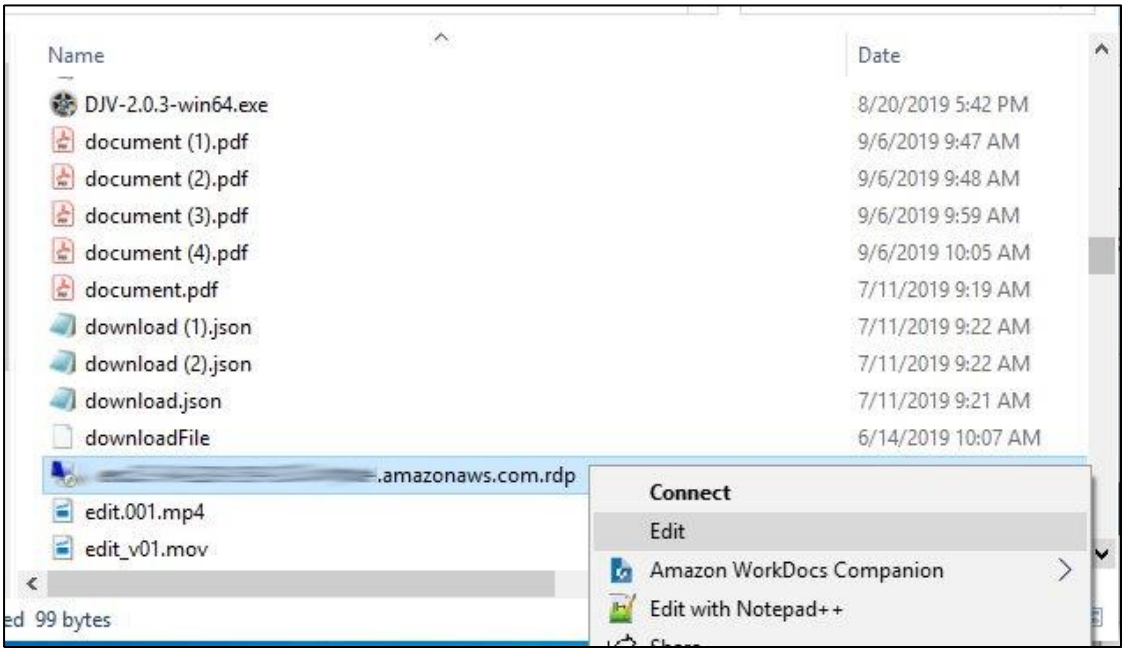
- ping.psa.fun – Useful website for estimating latency to different AWS Regions
- Microsoft Remote Desktop for Windows
- Microsoft Remote Desktop for macOS
- rdesktop Remote Desktop Client for Linux/Unix

Adjusting the Resolution of the Remote Desktop Window

Windows:

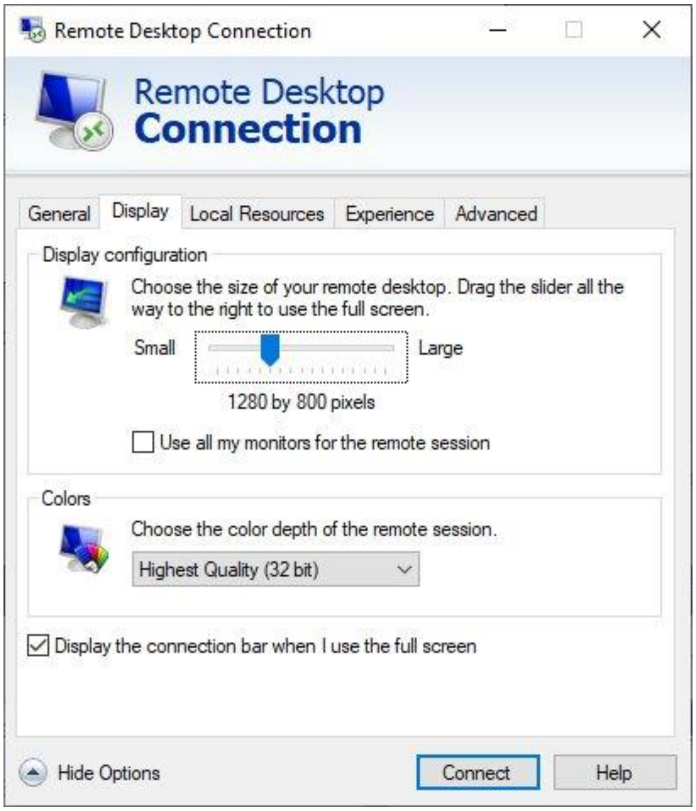
1. Choose the **Download Remote Desktop File** to connect to your instance. If prompted by your browser, save the file.

2. Find the .rdp file in your **Downloads** folder. It will be named something like ec2-35-39-399-854.compute-1.amazonaws.com.rdp



3. Right-click the .rdp file and choose **Edit.**

4. Click the **Display** tab in the Remote Desktop Connection window.



5. Select a resolution smaller than you're monitor's resolution and then click **Connect**

MacOS:

1. Click the **Download Remote Desktop File** to connect to your instance, if prompted by your browser, save the file.
2. Find the .rdp file and open it with your favorite text editor.
3. Add the following two lines to the bottom of the file, where you set the width and height values to something smaller than your monitor's current resolution:

   Desktopwidth:i:1920
   Desktopheight:i:1080

4. Save the .rdp file and the launch Remote Desktop with that file.

Required Services for Studio in the Cloud

AWS Service	Availability (as of 11/26/19)	Purpose	Usage for Tutorials
Amazon EC2 On Demand Instances	all regions	virtual computers in the cloud	*see instance types table below*
Amazon EC2 Spot Instances	all regions	discounted virtual computers ideal for render farm usage	*see instance types table below*
Amazon EC2 Auto Scaling	all regions except South America (Sao Paolo) and AWS GovCloud (US-East & US-West)	allows your render farm to automatically grow/shrink based on usage	*see instance types table below*
Amazon VPC	all regions	virtual network that contains all the parts of your studio	*no usage charges for tutorials*
IAM	all regions	grant/restrict access to AWS services	*no usage charges for tutorials*
Amazon FSx	limited regions	high speed managed storage	32 GB minimum
AWS Directory Service	all regions except Asia Pacific (Osaka-Local)	manage user accounts and profiles for your studio	about 2-5 days
AWS Secrets Manager	all regions except Asia Pacific (Osaka-Local) and China (Beijing & Ningxia)	store confidential information, like administrator passwords	1 secret

Required Instance Types for Studio in the Cloud

AWS Workstation Type	Availability (as of 11/26/19)	Purpose	Usage for Tutorials
m5.xlarge - Windows, 30 GB storage	all regions	studio creation and management	about 5-10 hours
m5.2xlarge - Linux, 300 GB storage	all regions	render farm workers	about 5-10 hours
m5.2xlarge - Windows, 100 GB storage	all regions	render scheduling	about 15-25 hours

AWS Workstation Type	Availability (as of 11/26/19)	Purpose	Usage for Tutorials
g3.4xlarge or g4dn.4xlarge - Windows, 150 GB storage	limited regions	GPU-enabled Windows workstations	about 10-15 hours

Tracking Availability in Your Region

You can track the availability of the required services and instance types in your Region using the websites below:

AWS Services By Region

AWS Instance Types By Region (all regions but Beijing & Ningxia)

AWS Instance Types By Region (Beijing & Ningxia only)

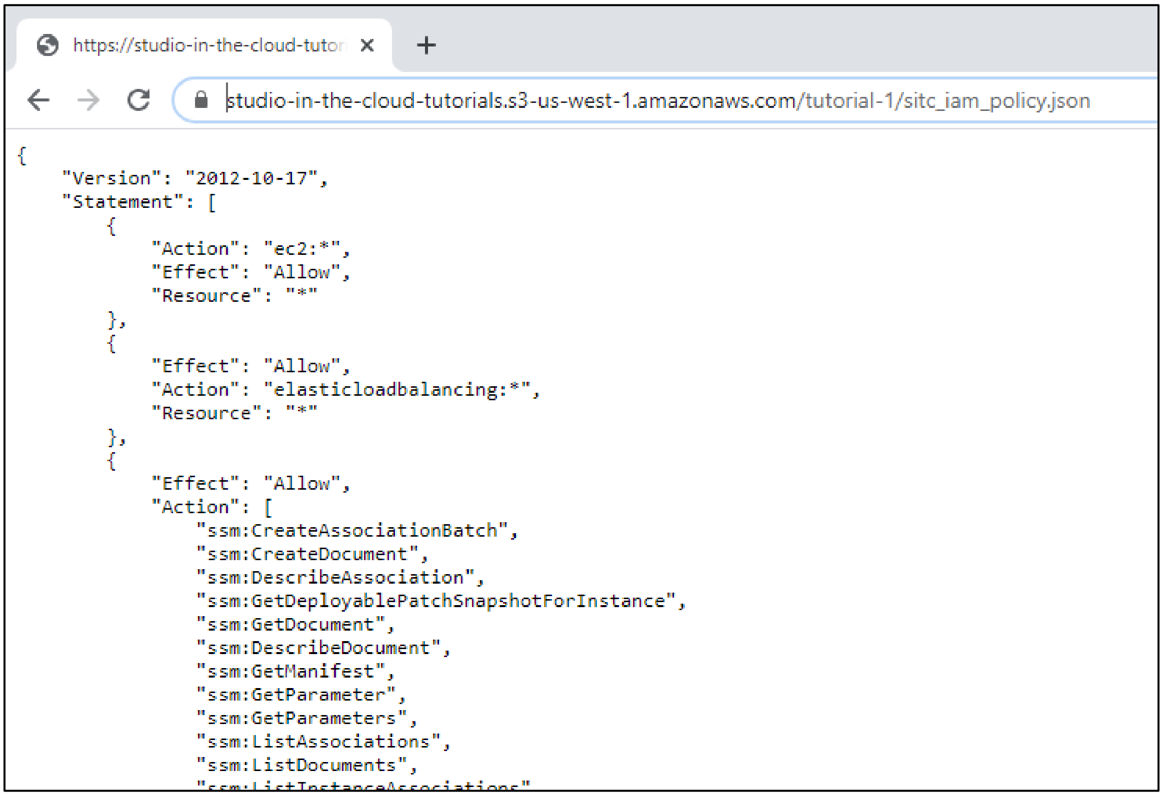
Creating an IAM User

1. Sign in to the AWS Management Console as the root user. Click the **Services** drop down menu, then under **Security, Identity, & Compliance**, choose **IAM** (or search for "IAM" in the search field).

2. In the navigation pane on the left, select **Users**
3. Choose **Add users**.
4. Enter a **User name** for your new IAM user (e.g., bob).
5. Select the check boxes for both **Programmatic access** and **AWS Management Console** access.
6. Select **Custom password** and enter a password in the field.
7. Clear the check box for **Require password reset**
8. Click **Next: Permissions**.
9. Click **Attach existing policies** directly
10. Click **Create policy** .

11. Click the **JSON** tab.

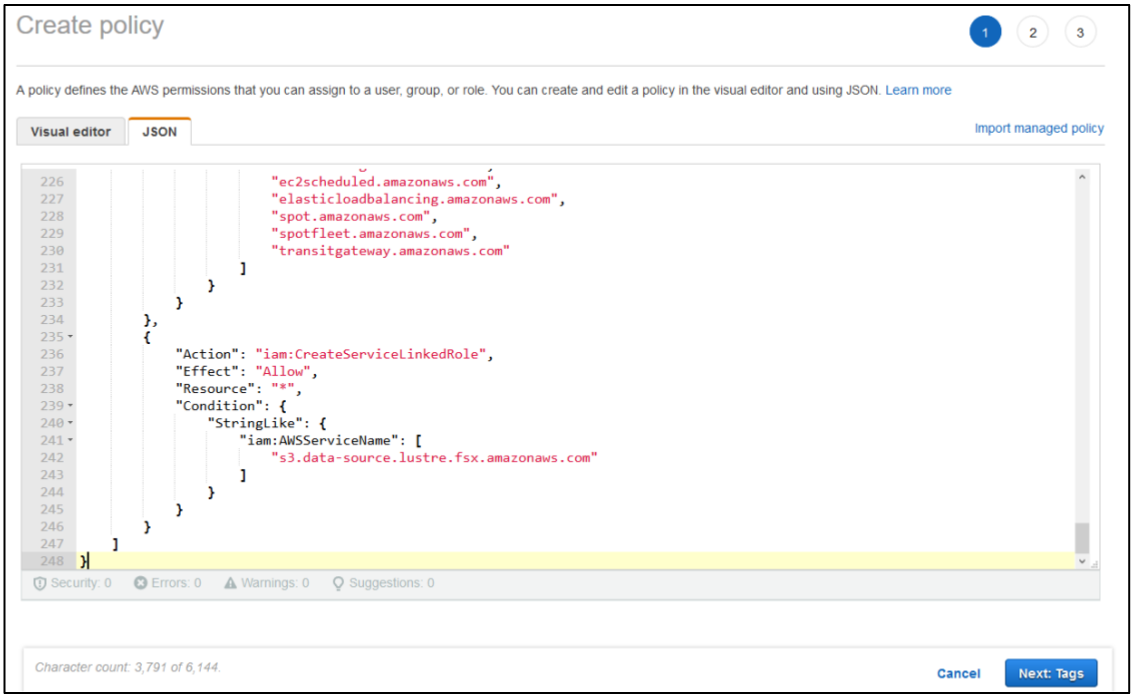
   - <shift>+click the image below to open a new browser tab with the text that needs to be entered into the JSON entry field:



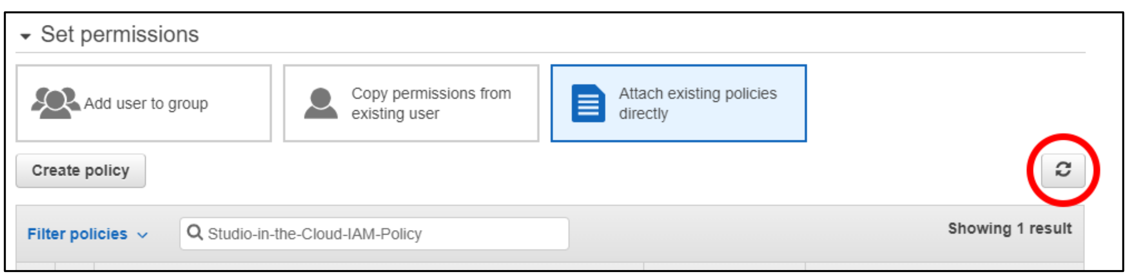

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "ec2:*",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "elasticloadbalancing:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:CreateAssociationBatch",
                "ssm:CreateDocument",
                "ssm:DescribeAssociation",
                "ssm:GetDeployablePatchSnapshotForInstance",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
                "ssm:GetManifest",
                "ssm:GetParameter",
                "ssm:GetParameters",
                "ssm:ListAssociations",
                "ssm:ListDocuments",
```

sitc_iam_policy.json - <shift>+click the image above to open the JSON file in a new tab

   - Replace the existing text in the JSON entry field with the text from the file above. <u>Note</u>: The file is quite long, so make sure you get everything.
   - When you're all done, the JSON entry field should look like this:



12. Choose **Next: Tags**
13. Choose **Next: Review**.
14. For **Name** type *Studio-in-the-Cloud-IAM-Policy.*
15. For **Description** type *Custom IAM policy with all permissions to complete the Studio in the Cloud tutorials.*
16. Choose **Create policy**.
17. Return to the browser tab that you were using to create your user and refresh the policies list by clicking the refresh button.



18. In the policies search field, type **Studio-in-the-Cloud-IAM-Policy** and then select the check box next to it.
19. Choose **Next: Tags**.

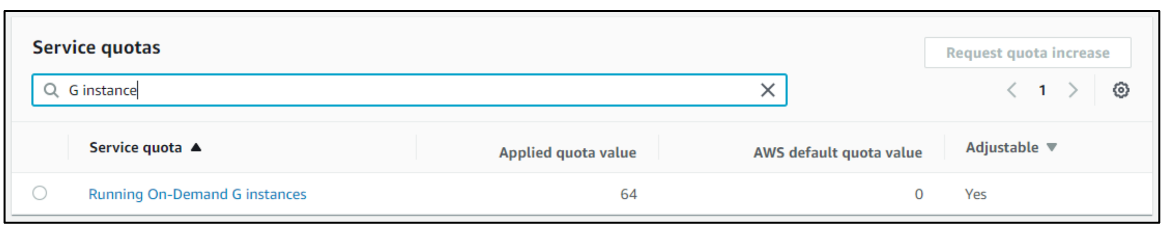
20. Choose **Next: Review**.
21. Review the information for the user and then choose **Create User**.
22. On the next page, you'll be presented with the security credentials for your user. You'll need these credentials later in the tutorials. Choose **Download .csv** to download them now.

    Make note of the location of the .csv file so that you can find it later. *Also enter the location of* the *.csv file on the [Important Information Cheat Sheet]() that we've provided.*

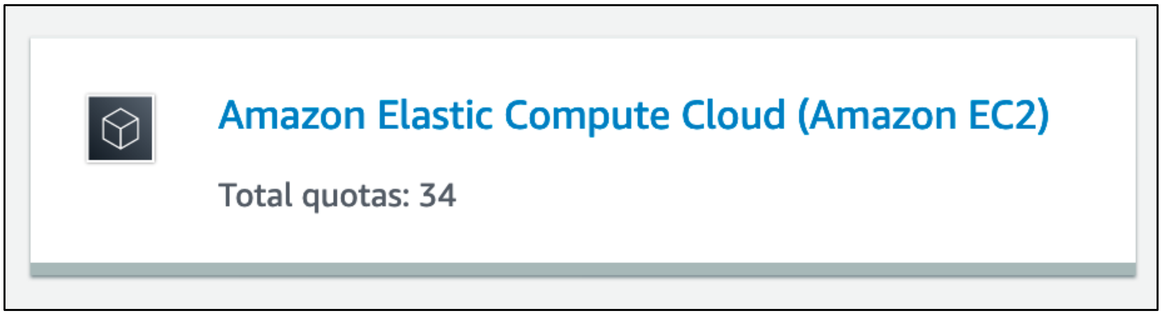
23. When you're done, sign out of the AWS Console and log in as your new user.

Checking Your G Instance Quota

1. Sign in to your AWS Account.
2. At the top of the Console, choose the **Services** drop-down menu and type **service quota.**
3. Select Service **Quotas** to go to the Service Quotas Dashboard.
4. Choose the **Amazon Elastic Cloud Compute (Amazon EC2) card**.
5. In the **Filter by**, type in **G instance.**



Now you can review the quota you have available for your G instances. In order to complete these tutorials, you need an **Applied quota value** of at least **16** because the **g4dn.4xlarge** instance you use in part of the tutorial uses **16 vCPUs**.



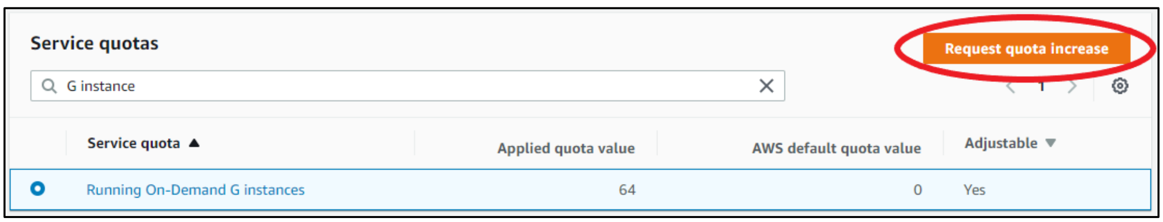
To see how many vCPUs are used by other G4 instance types, see the [G4 Product Details]chart.

Note: After completing the tutorials, you may need more than 16 vCPUs depending on how many additional GPU workstations you need to launch for your artists.

To request an increase in your quota value, follow the instructions below.

Requesting a Quota Increase

1. Choose the option button for **Running On-Demand G instances**, and then choose **Request quota increase**.



2. Enter the new **Quota value** you would like to receive.
3. Click **Request**.

Within a few minutes you will receive a new **Support Case** for your request. It can take anywhere from 12-48 hours to return a limit increase request.

You can track the status of your request in the Service Quotas Dashboard:

1. Choose **Quota request history** in the navigation panel on the left. You will see the status of your request in the list on the right.
2. To see more details, click the **status** value. (Immediately after submitting the request, it will say “Quota requested”). Then in the window that pops up, click the **Support Center case number.**

3. If you have a contact at AWS who supports your account, you should also notify them that you have submitted a quota increase request.

Tutorial 2: Creating a VPC and Active Directory for Your Studio

Estimated Time to Complete: 1 hour

This tutorial will cover creating the basic infrastructure for your Studio in the Cloud. We'll set up your own network, connect it to the internet and configure some options to handle security and user accounts. At the end of the tutorial, you'll launch an instance in your studio and login using your studio's administrator account.

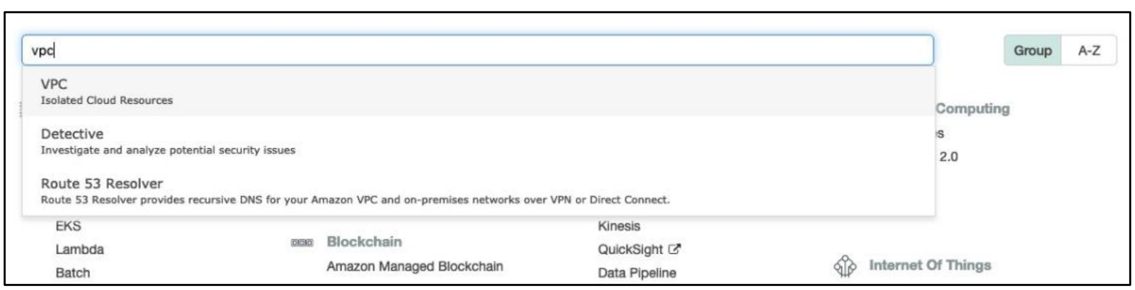
Creating Your Own VPC

The Virtual Private Cloud (VPC) will be the backbone of your cloud-based studio. It's a virtual network that is dedicated to your AWS account. It is isolated from other virtual networks on AWS. You can think of your VPC as the container in which your Studio in the Cloud resides, including workstations for all your users, shared file storage and your render farm. It enables your users to share resources while also keeping your data secure.

Each AWS account already has a default VPC that has been created in each region that AWS serves. In the starter exercise from our first tutorial, you launched an EC2 instance in the default VPC for your region. But for your Studio in the Cloud, we'll need to customize things a bit. We'll begin with creating a VPC of your own.

Launch the VPC

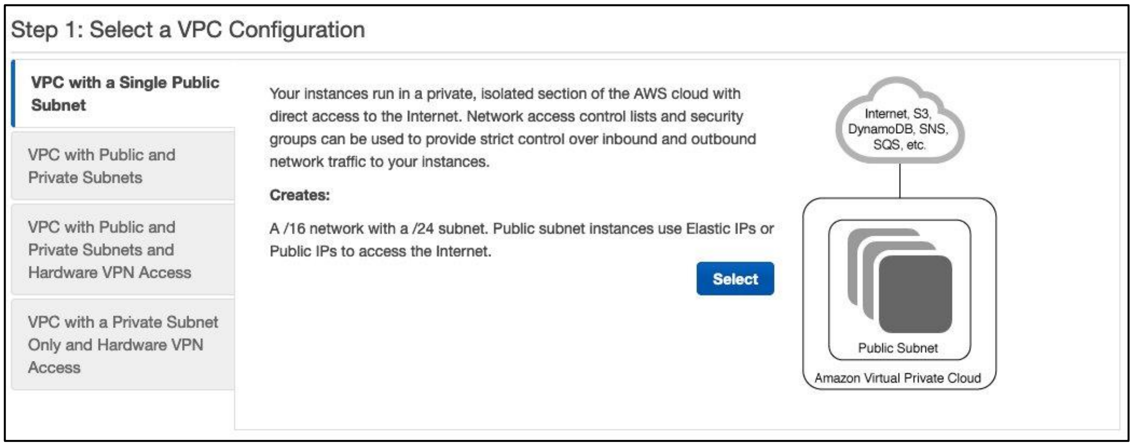
1. Log in to the AWS Console using your AWS account information, if you haven't already.
2. Check that your Region is set correctly. You may have already set your Region as part of the starter exercise in the last tutorial. For a refresher on how to set your Region, go to the Set Your Region section of Tutorial 1.
3. Once you've set your Region, go to the **Services** drop-down menu and in the **Networking & Content Delivery** section choose **VPC** (or search for VPC in the search bar at the top of the page).



4. Click **Launch VPC Wizard**.

5. Choose **VPC with a Single Public Subnet**.

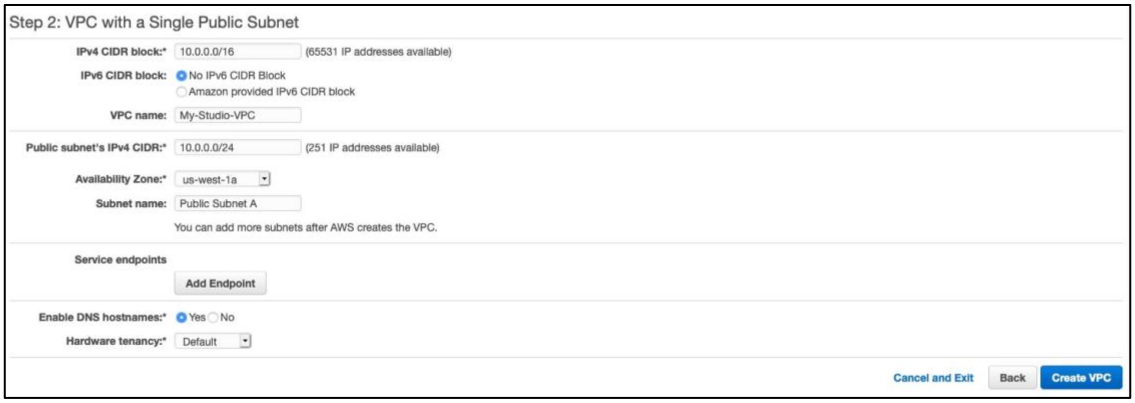
   A subnet represents a portion of the addresses in your network. Having more than one subnet allows you to divide your network into different sections based on usage and for security. By default, the launch wizard creates a subnet with IP addresses from 10.0.0.0 to 10.0.0.255. We'll be creating a second subnet for your VPC in the next section.



⚠ *Note*: We are using a public subnet here for initial setup and testing purposes. After completing the tutorials, we recommend using a private subnet before using your setup for production content requiring a higher level of security.

6. Click **Select**.
7. Fill out the next page, **Step 2: VPC with a Single Public Subnet** (unless otherwise stated, leave as default values).
   - **VPC name**: Give your VPC a descriptive name. If you want to pick a name for your studio, you can use it to name your VPC (e.g., My-Studio-VPC). If you'd like to use our example name of "My-Studio", that's fine too. *Either way, be sure to write down your studio name and VPC name in the [Important Information Cheat Sheet] so you can refer to them later.*
   - **Availability Zone**: Choose the **'a'** Availability Zone for your Region (e.g., us-west-2a).
     - Note: If the Availability Zones you have access to don't include 'a', just set it to one of the other zones.

- **Subnet name**: Set this to **Public Subnet A** (you'll add another public subnet next).



8. Click Create VPC.

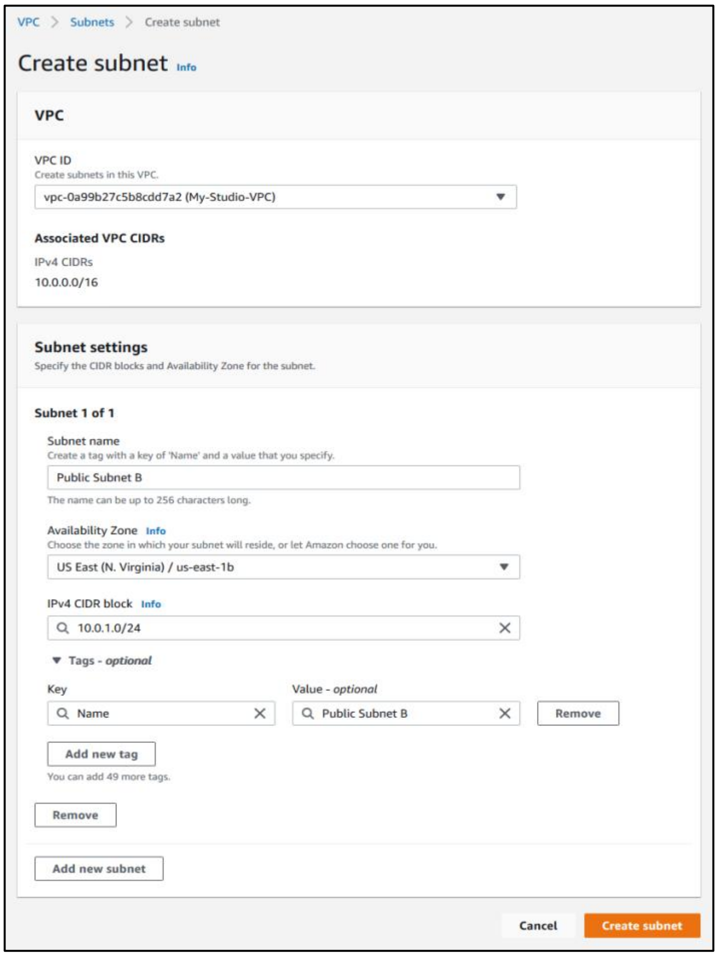
A Note on Naming

Throughout the tutorials, you'll need to name various components of your studio, such as the VPC above. A good rule of thumb is to avoid any spaces or special characters, except for hyphens (-). While some components have more relaxed naming restrictions, some, like directory DNS names, are more restrictive. For that reason, we recommend erring on the side of caution. Each time you need to choose a name for something, we'll provide an example for you that follows the correct naming convention. Feel free to use our example naming or if you choose your own studio name, you can replace any instance of *My-Studio* with your studio's name instead.

Create a Second Subnet

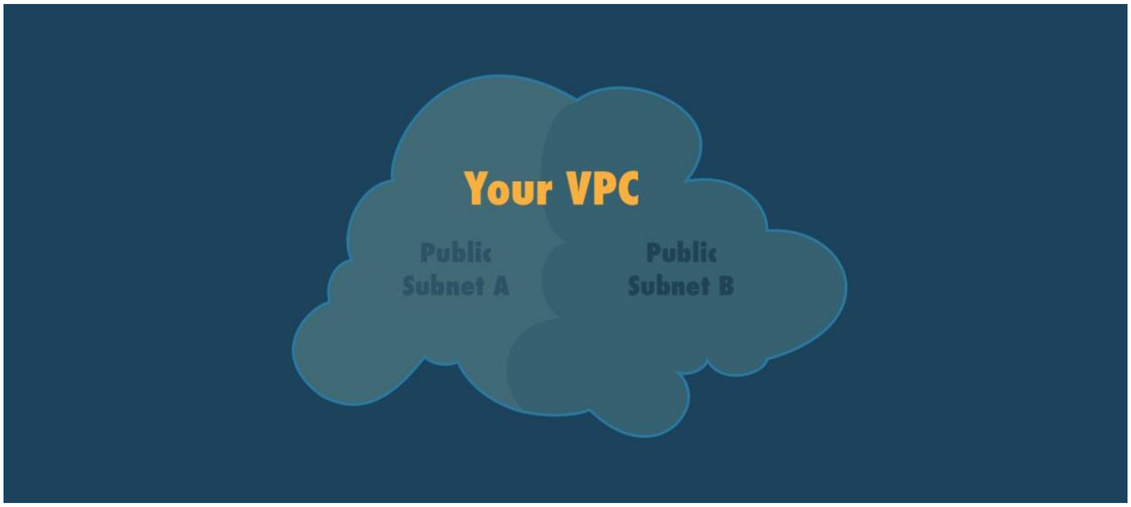
1. Click **Subnets** in the left side panel.
2. Click **Create subnet**
   - VPC ID: Choose the one you just created (e.g., My-Studio-VPC)
   - Subnet name: Public Subnet B
   - Availability Zone: Choose the **'b'** availability zone for your region (e.g., us-west-2b)
   - IPv4 CIDR block: **10.0.1.0/24**

   The "/24" in the CIDR block notation above specifies a range of IPv4 addresses for your subnet. In this case, we are specifying that your second subnet will have 24 bits allocated for the network prefix, and the remaining 8



bits for host addressing, resulting in IP addresses that range from 10.0.1.0 to 10.0.1.255.

3. Click **Create subnet**

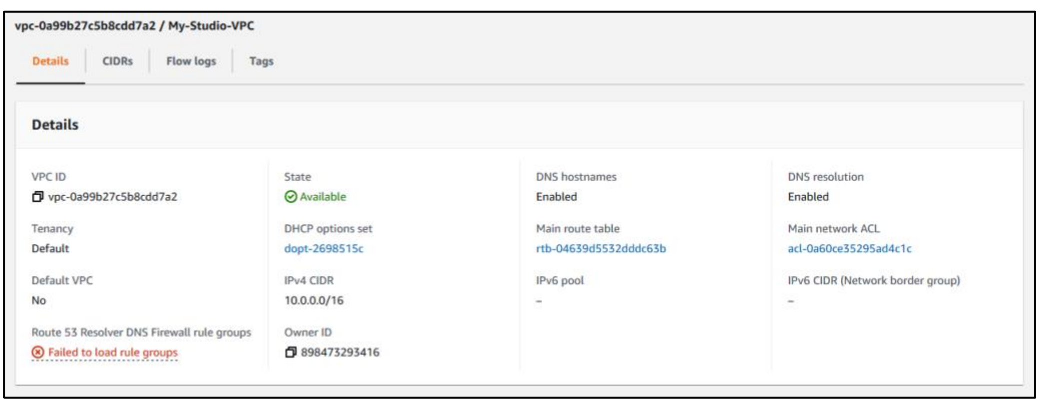


Internet Gateway

Now that you have your own VPC, you'll need to edit the internet gateway. An internet gateway allows for communication between your VPC and the greater internet. You'll need access to the internet in order to access the virtual workstations that we'll create later on.

Edit the Internet Gateway

1. Click **Your VPCs** in the left side panel
2. Select the VPC that you created in the last step
3. Under the **Details** tab in the bottom panel click the **Main route table** item



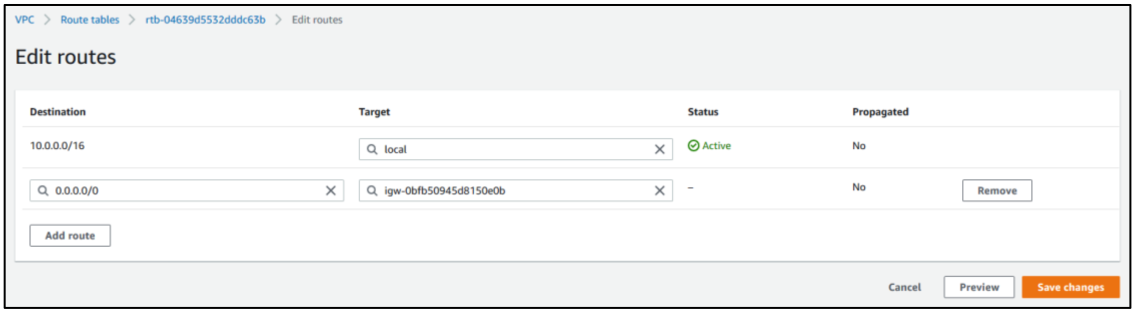
This will show you the Route Table associated with your VPC. By default it doesn't have a name. To make it easier to identify your Route Table in the future, it's a good idea to name it.

4. Hover over the blank field under **Name** in the list of Route Tables until you see the **pencil icon**.

5. Click the icon, and then name your Route Table something that helps you identify it (e.g. My-Studio-Route-Table).

6. Make sure the Route Table you just named is selected, then click the **Routes** tab down below.
7. Click **Edit routes**
   a. Click **Add route**
   b. Under Destination add 0.0.0.0/0

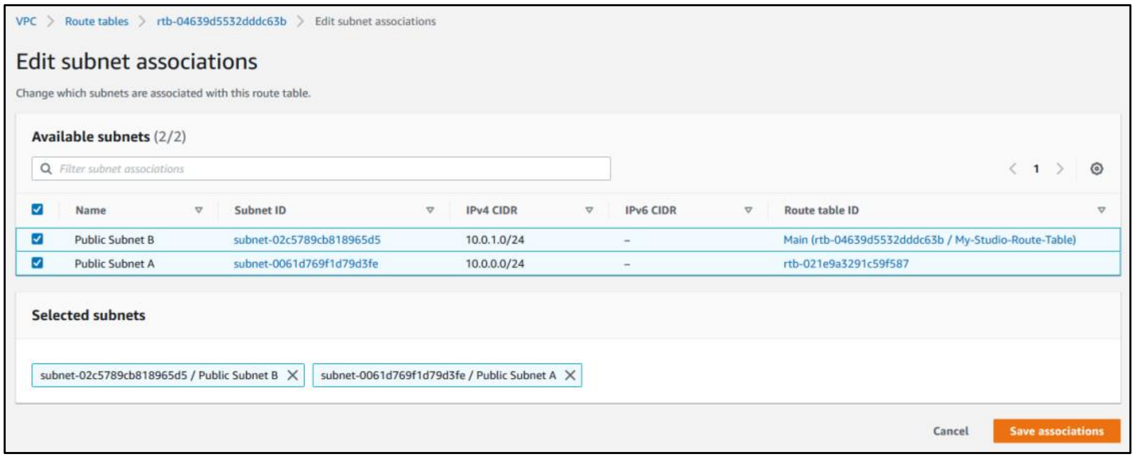
c. For **Target** choose **Internet Gateway** and pick the item that appears.



d. Click **Save changes**

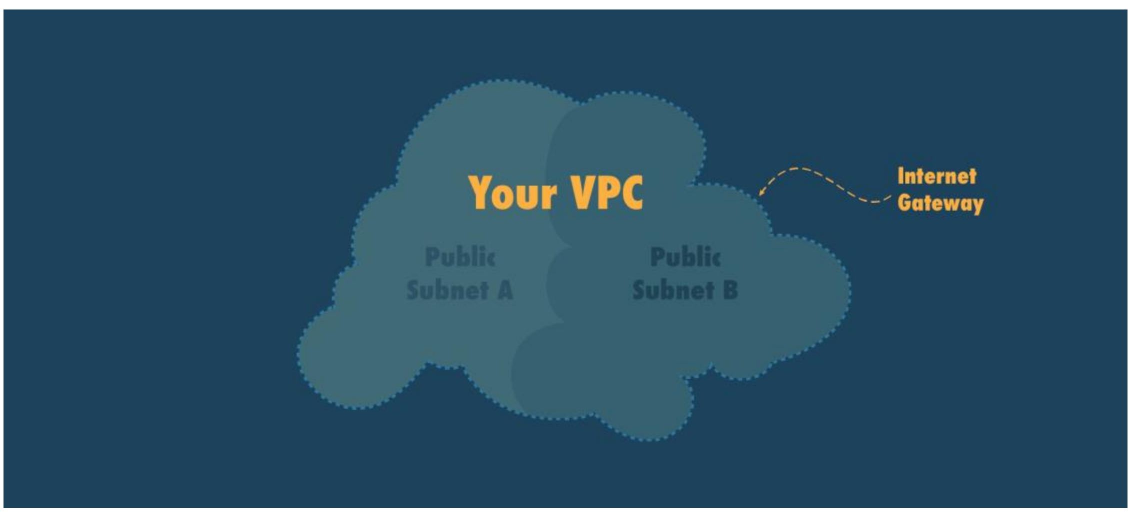
8. Click the **Subnet Associations** tab down at the bottom of the page.
9. If there are no associations listed in the first table, click **Edit subnet associations**

10. Select **ALL** of the subnets



11. Click **Save associations**

You now have your own VPC, which contains two public subnets and is connected to the Internet through your Internet Gateway:

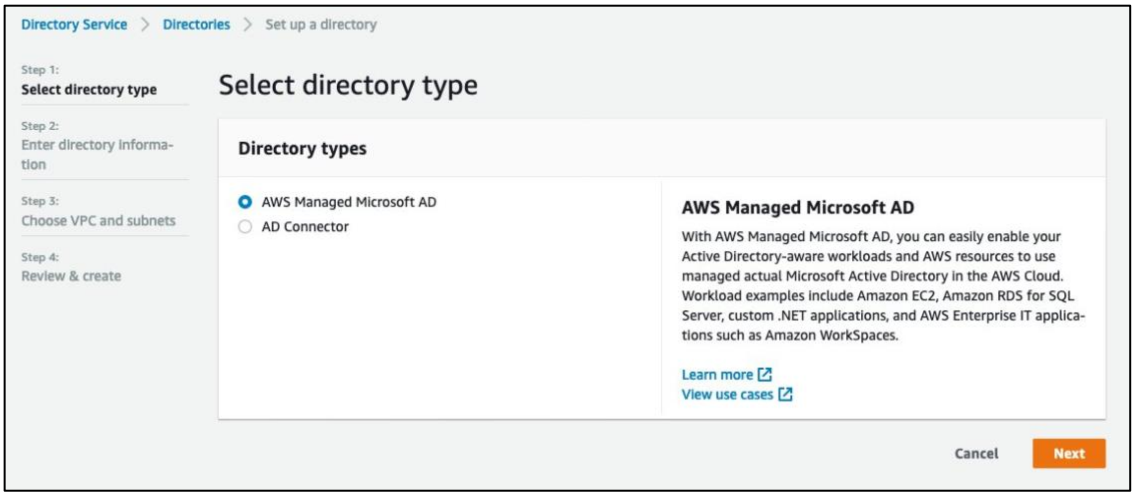


Active Directory

When you created an instance and connected to it in the first [tutorial](), you logged in as Administrator. This is fine for a quick test, but for your studio you will want to have artists log in with their own accounts. In order to manage the different artists and users in your studio you'll set up a **Managed Microsoft Active Directory**. This allows for the storing of user profiles and settings that can follow users from virtual machine to virtual machine. In the next tutorial, we'll be adding users to Active Directory for your artists, but for now we'll just be doing some initial setup.

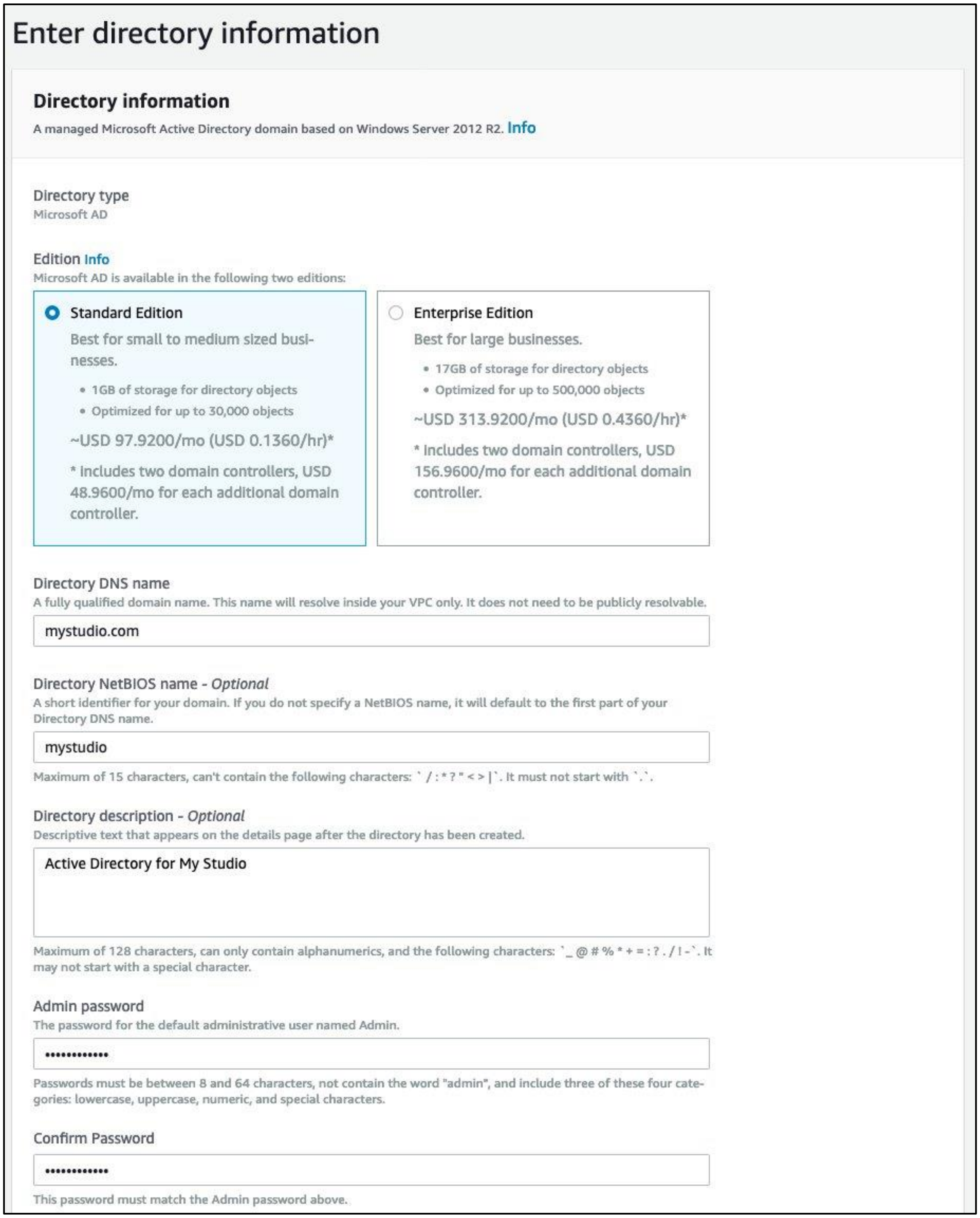
Set up your Active Directory

1. Go to the **Services** drop down menu and search for **Directory Service**
2. Click **Set up directory**
3. Choose **AWS Managed Microsoft AD** and click **Next**



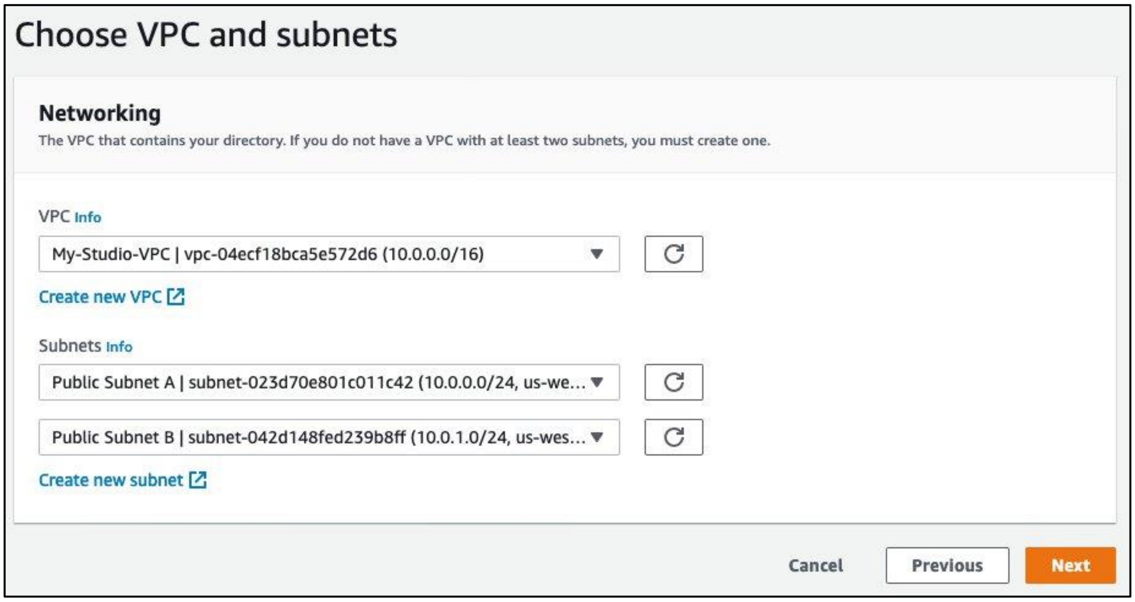
4. Set the Directory Information
   - Edition: Standard Edition
   - Set the Directory DNS Name to **<your studio name>.com** (e.g., mystudio.com). Since DNS names are not case-sensitive, we suggest setting your DNS Name to the name of your studio, but in all lowercase letters. You'll also be typing this in a lot in later steps, so it's also easier to type! *Write this down in the Important Information Cheat Sheet.*
   - Set the directory **NetBIOS** name (e.g., mystudio). Again it's not case-sensitive, so all lowercase here. *Note this in the cheat sheet as well.*
   - Set a description if you like.

- Set a unique Administrator password. Make sure to put this on the cheat sheet as you will be using this password frequently.



5. Click **Next**

6. On the **Networking** page, choose the **VPC** you set up earlier (e.g., My-Studio-VPC) *You can find this on the cheat sheet.*
7. Choose **Subnets** (A, then B - important to keep them in order)



8. Click **Next**
9. Review the information you entered in the steps above, then click **Create directory.**

   This will take some time, about 20-40 minutes. While it is creating, move on to the next step!

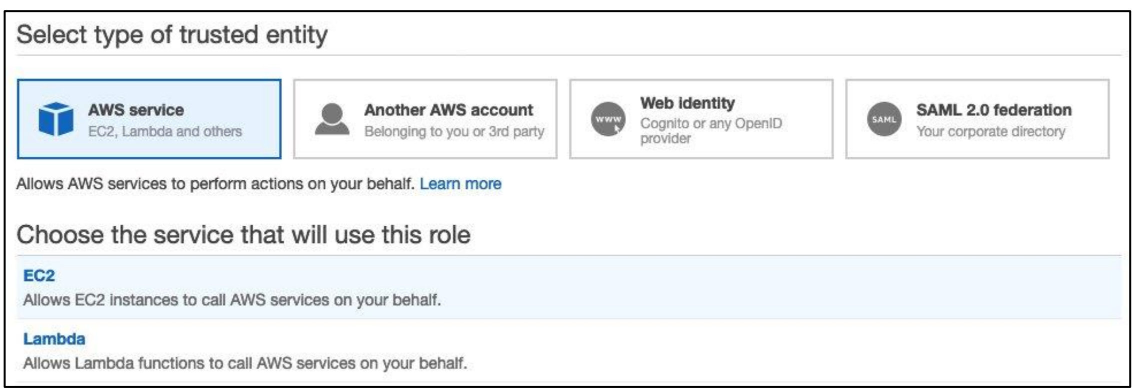
IAM Role

AWS Identity and Access Management (IAM) helps you control access to AWS resources. We will use an IAM role to allow access to the services that your users need and restrict access to things that they don't. In this case, we'll be creating an IAM role to allow access to the Active Directory we just created. Without access to your Active Directory, your artists won't be able to login to instances using the unique usernames and passwords that we'll be creating for them in the next tutorial.

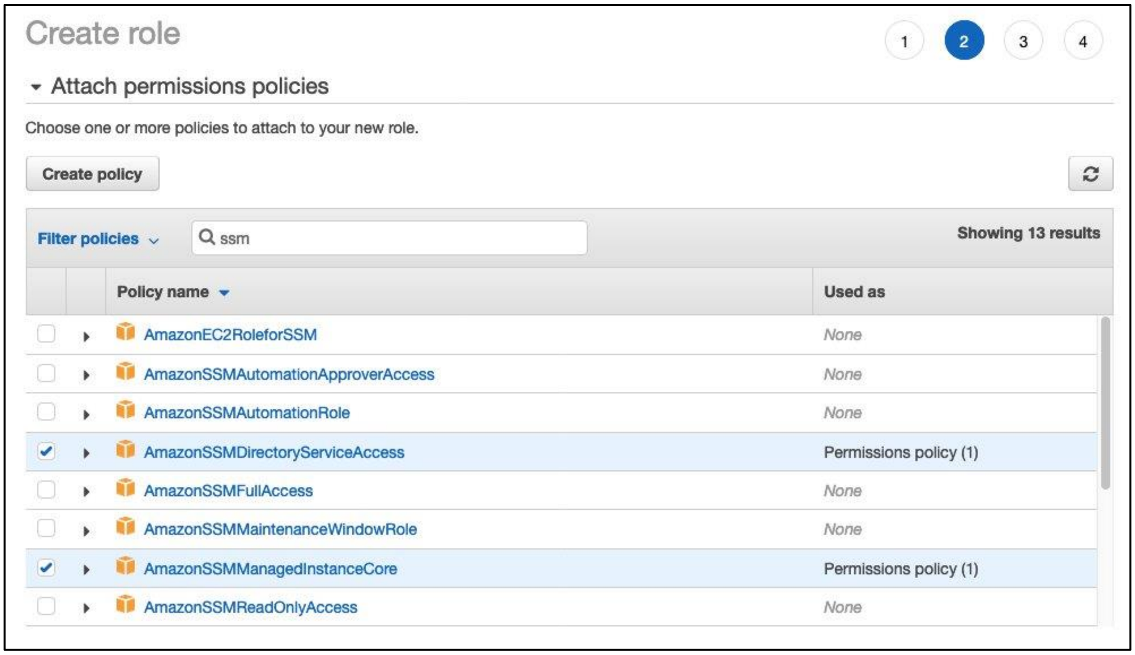
Create IAM Role for Launching Instances

Let's create an IAM Role that we can use when launching virtual workstation instances!

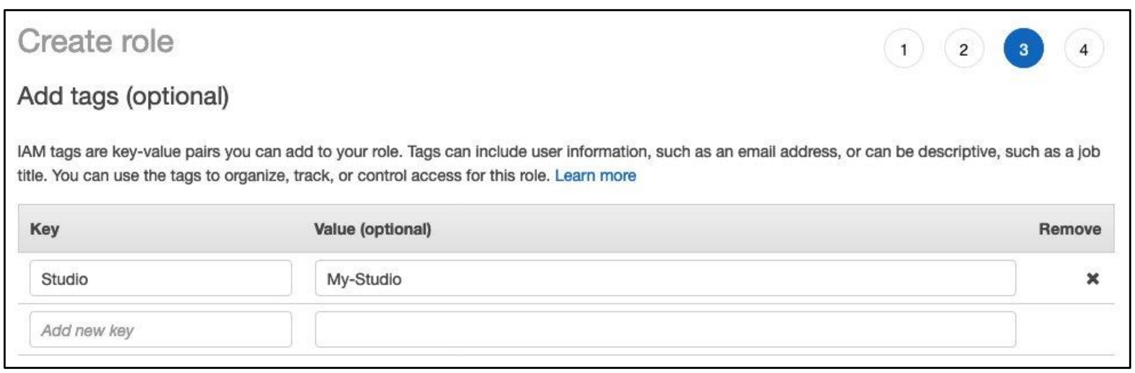
1. In the AWS Management Console, choose the **Services** drop-down menu and search for **IAM.**
2. In the left navigation pane, click **Roles**
3. Click **Create role**
4. Under **Select type of trusted entity**, choose **AWS service**.
5. Under **Choose the service that this role will use**, choose **EC2** and then choose **Next: Permissions**.



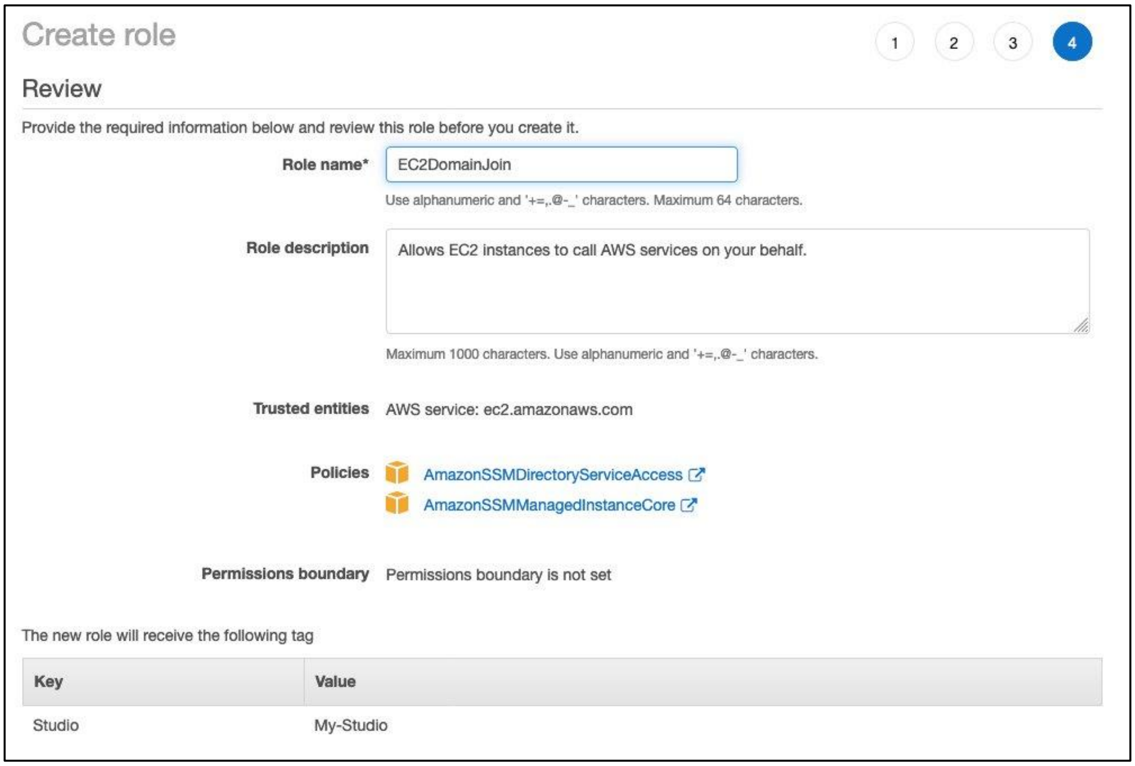
6. In the list of policies, select the **AmazonSSMDirectoryServiceAccess** and **AmazonSSMManagedInstanceCore** policies. (To filter the list, type **SSM** in the search box.)



7. Click **Next: Tags**
8. For **Key** enter **Studio** and for **Value** enter the name of your studio from above (e.g., My-Studio). Then, click **Next: Review**



9. For **Role name**, enter **EC2DomainJoin**
10. (Optional) For **Role description**, enter a description



11. Choose **Create role**

Security Groups

Next, you'll create a new security group. Since your VPC is connected to the Internet, you need some way of controlling the inbound and outbound traffic. In the security group you will define rules to allow only certain kinds of traffic into your VPC. This helps keep your VPC and your data secure. In our tutorials, we'll be creating different security groups for different functions.

The first security group we'll create will be to allow Remote Desktop connections, which we'll be using throughout these tutorials for connecting to EC2 instances. Later on, we'll create more security groups to allow for other things such as connecting to Linux instances and connecting to render farm management software.

Create a Remote Desktop Security Group

1. Go to **Services**→ **EC2**

2. Check that your region is still set correctly by looking at the region drop down menu in the top right of the page.
3. Select **Security Groups** in the left side panel under **Network & Security**.
4. Click **Create security group**.
5. Choose a name for your security group (e.g., My-Studio-Remote-Desktop-SG)
6. Set the Description to **Allows for Remote Desktop Connection**
7. Set the **VPC** to the VPC you created above (e.g. My-Studio-VPC). *You can also find this information on the cheat sheet.*
8. In the **Inbound rules** block click **Add rule**

Next, we'll tell the security group which ports are going to be open for *incoming* connections.

Notice at first there are no rules for incoming connections. This means that nobody can connect to this machine from anywhere. We want to allow your local computer to be able to connect, thus we need to open the port that specifically allows that connection.

9. Under **Type:** choose **RDP**

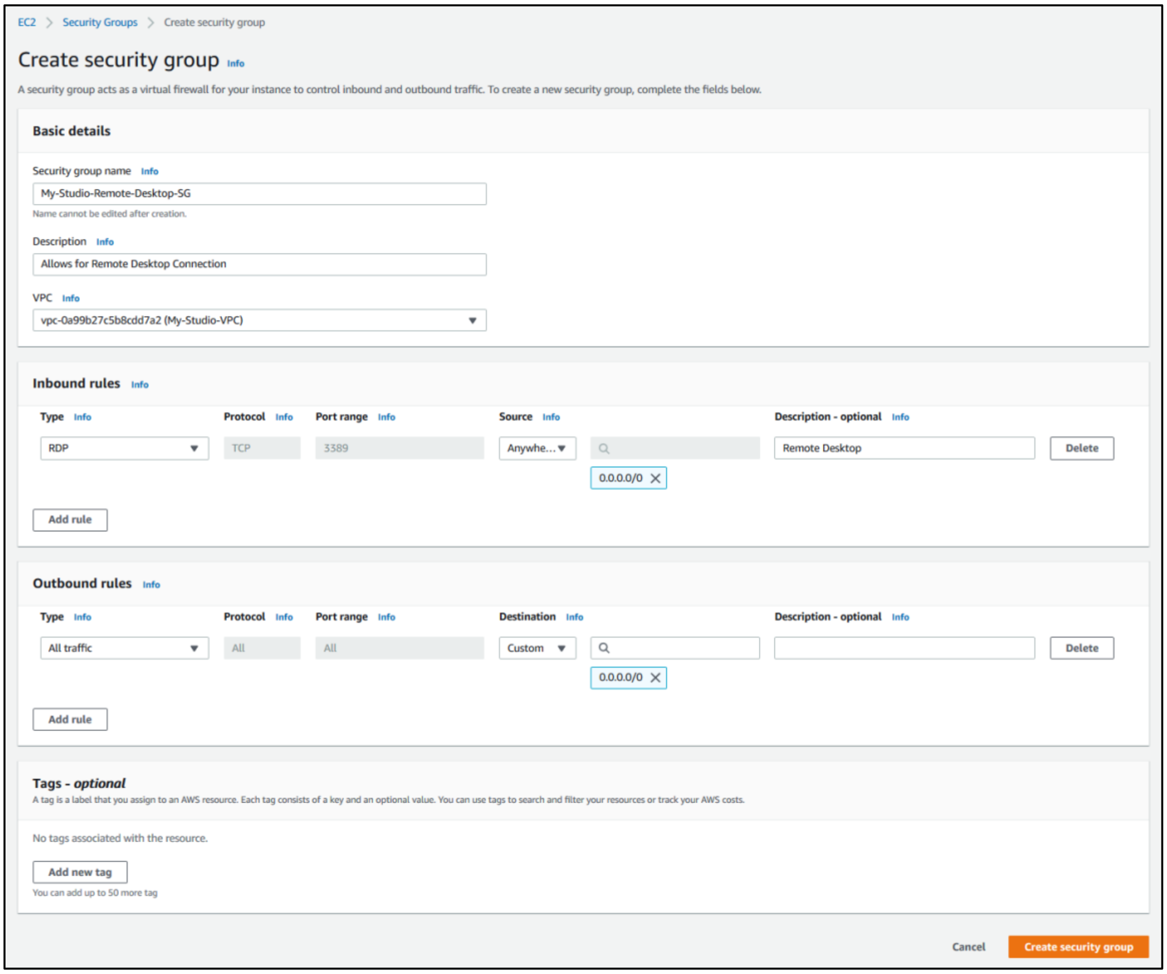
   Notice that it automatically set the port range to **3389** - this is the default incoming port for RDP (Remote Desktop Protocol) connections.

10. Set the **Source**

> ⚠ *Note*: By default, we'll be setting the source to "Anywhere", which allows any computer to attempt Remote Desktop connections to your instances. This is probably acceptable during the initial setup of your studio, but before creating production content, you will want to limit the source to **just the computers** your artists will be using to access your studio. Your network administrator can help you determine the correct range of source IP addresses to enter here.

If you do not wish to limit which IP addresses can connect at this time, then you can set the Source to **Anywhere-IPv4.** It will automatically add the IP address 0.0.0.0/0.

11. Set the Description to **Remote Desktop**



12. Click **Create security group.** The details page for the newly created security group appears.

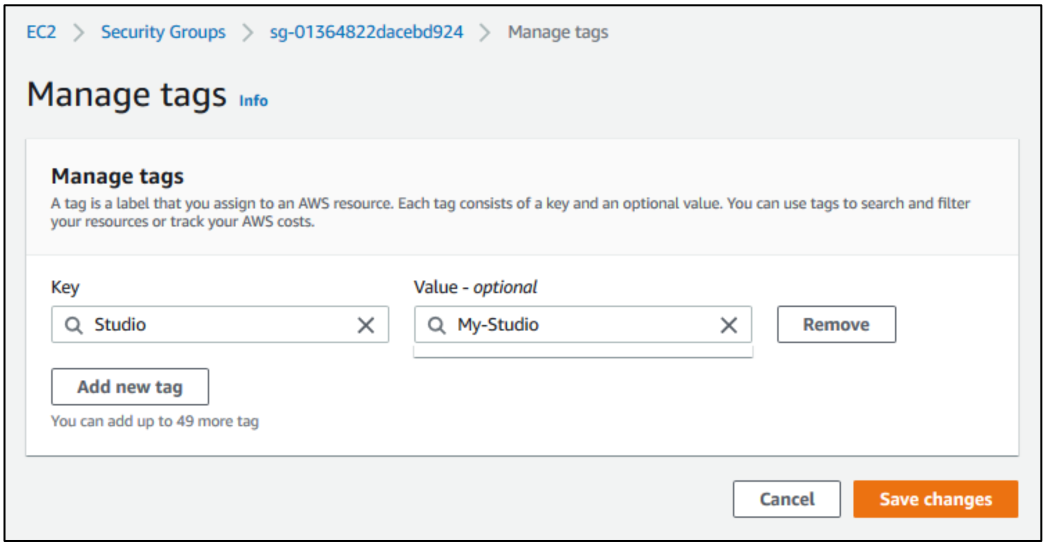
    Take a moment to enter the Remote Desktop Security Group's name and ID on the *Important Information Cheat Sheet*. You can find the ID to the right of the name in the list of security groups. It will look like: sg-928th290koqj8214r.

Add a Tag to Your Security Group

Next, we're going to add a tag to identify this security group as part of your studio. A tag is just a label that you can assign to an AWS resource. They make it easier to manage, search for and filter resources. They can also be used as cost allocation tags to help you understand the cost of running various aspects of the studio.

1. Make sure your new security group is selected

2. On the **Tags** tab in the bottom panel, click **Manage tags**
3. In the **Manage Tags** window, click **Add new tag**
4. For **Key** enter **Studio**
5. For **Value** enter the name you picked for your studio (e.g., My-Studio). *Refer to your studio name in the cheat sheet if needed.*



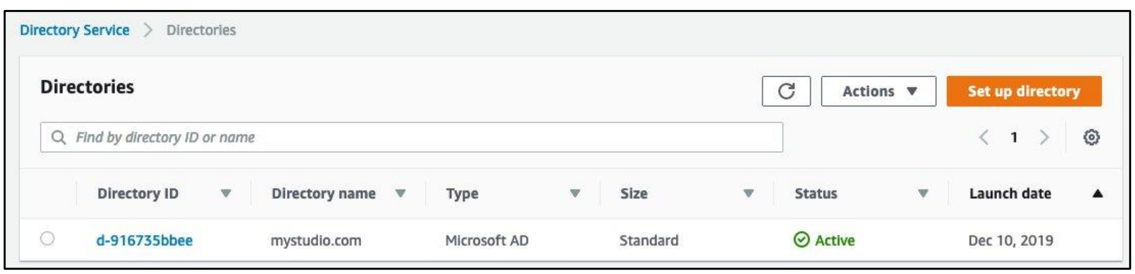
6. When you're done, click **Save changes**

We'll be adding this same Studio:My-Studio tag to each of the resources you'll be creating for your studio in the cloud. That will make it easy to find everything if you need to later on.

Exercise: Test Your Setup So Far

Congratulations, you've completed the basic setup for your Studio in the Cloud! Now you're going launch a virtual workstation and test that Active Directory is working. This instance will be used later to manage your users. We'll call it the **User Management** instance.

<u>Note:</u> Before moving on to this step, make sure that Active Directory has finished creating by going to **Services** → **Directory Service**. When it's done, the status will change from **Creating** to **Active**.



Sometimes the status may not update automatically. If you've already waited 20-40 minutes and the status is still listed as Creating, you can click **Refresh**.

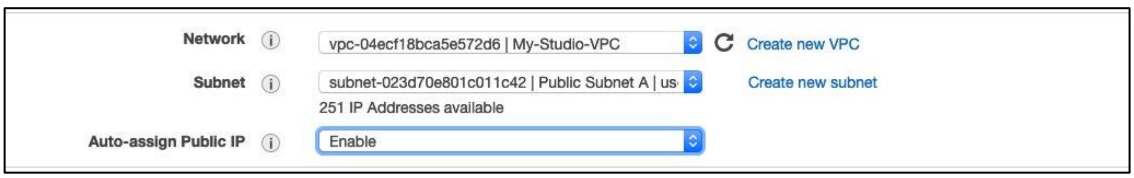
Create an EC2 Instance

1. Go to **Services** → **EC2** and select **Instances** in the left side panel
2. Click **Launch instances**

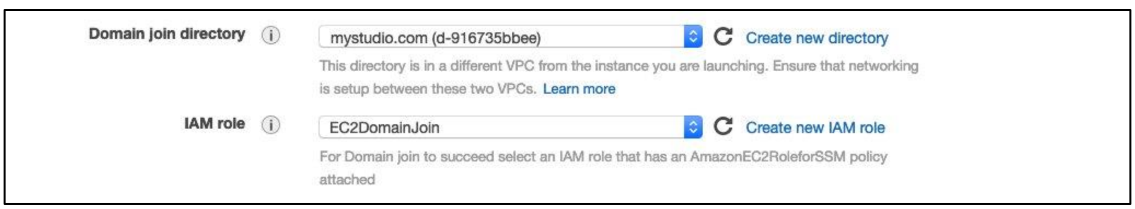
3. Search for **Windows**
4. Select **Microsoft Windows Server 2019 Base**
5. Choose the instance type **m5.xlarge**, then click **Next: Configure Instance**

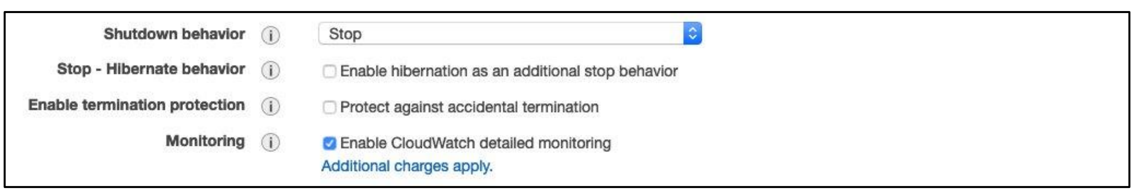
   Details

6. Configure **Instance Details:**
   - Network: **<VPC you created>** (e.g., My-Studio-VPC) Refer to the VPC name your wrote down in the cheat sheet.
   - Subnet: Public Subnet A
   - Auto-assign Public IP: **Enable**



   - Domain join directory: <**your Active Directory DNS name>** (e.g., mystudio.com) *You can also find this on the cheat sheet.*
   - IAM role: **EC2DomainJoin**

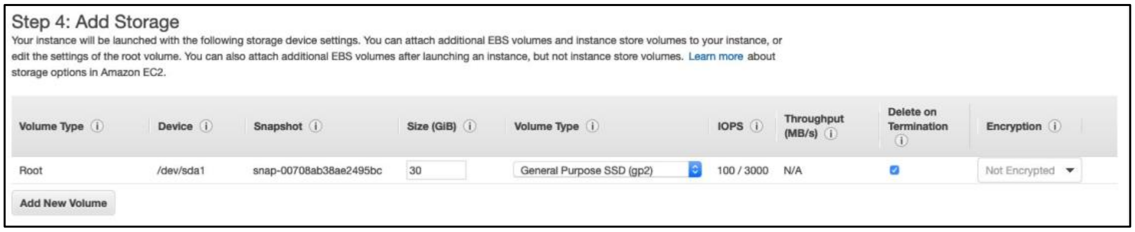




   - Monitoring: **Enable CloudWatch detailed monitoring**

7. Click **Next: Add Storage**
8. Add **Storage.**

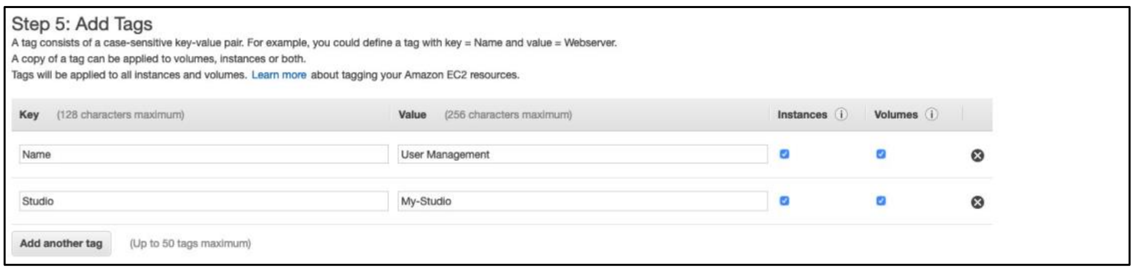
- Leave the storage size at the default value of **30 GB.**



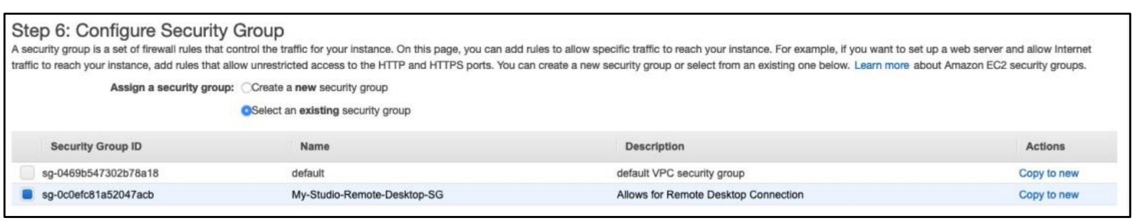
- Click **Next: Add Tags**

• Add **Tags**

- Add a tag with the key **Name** and the value **User Management**
- Add another tag with the key **Studio** and the value the name of your studio you've used in other steps (e.g., My-Studio)

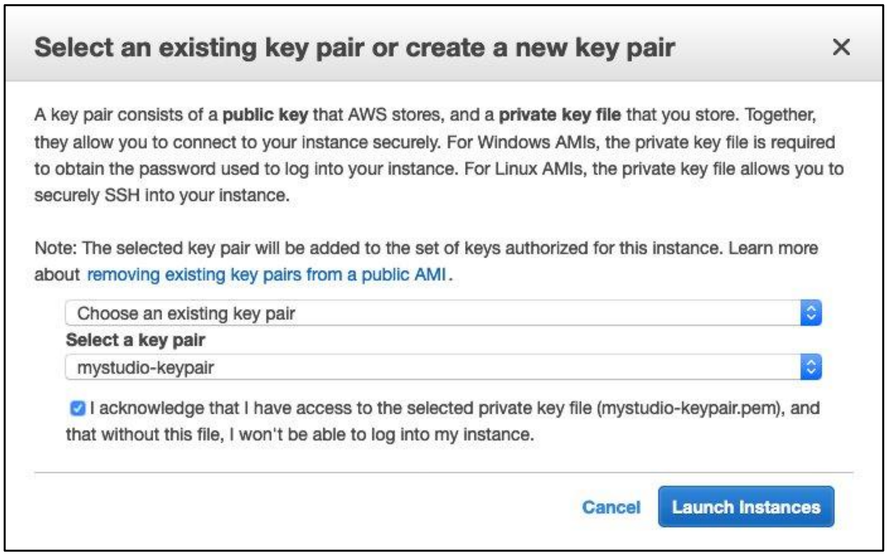


9. Click **Next: Configure Security Group**
10. Configure **Security Group**
    a. Choose **Select existing security group**
    b. Choose the security group that you created earlier (e.g., My-Studio-Remote-Desktop-SG). Selecting that security group allows for Remote Desktop connections to your instance. For now that is all we need, but in later tutorials we'll be adding more security groups to allow other types of connections as well. *Refer to the Remote Desktop Security Group Name and ID your entered on the cheat sheet if necessary.*
11. Click **Review and Launch**



12. Click **Launch**

13. Create a new key pair if you want, or use one that you already have. If you completed the Starter Exercise from Tutorial 1, we recommend re-using the key pair that you already created (e.g., mystudio-keypair.pem). *If you don't remember the name of the key pair you created, you can find it at the bottom of the Tutorial 1 section of the cheat sheet.*
14. Select the check box next to the acknowledgement that you have access to the selected key pair.



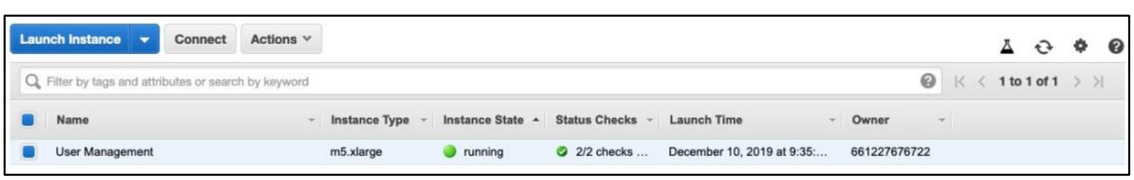
15. Click **Launch Instances**

    Note: If you are trying to launch in a new region, you may get a Launch Failed screen after attempting to launch. If that happens, just wait a few minutes and try again.

16. Click **View Instances** at the bottom right of the screen to go to the list of running instances.

Logging in

1. After a few minutes, when the instance is ready (its state says running and the status checks say 2/2 Status Checks Complete), select it and choose **Connect.**
2. Choose the **RDP client** tab, then click **Download the remote desktop file**

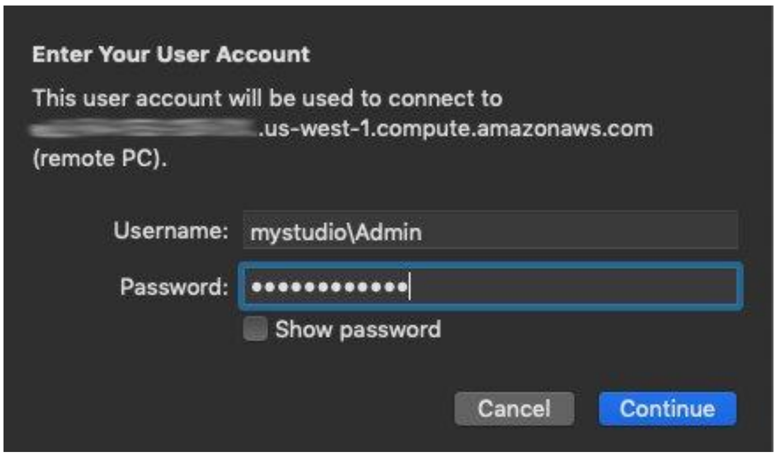


3. Open **Remote Desktop**

   A window asking for the Administrator password will pop up, but this time we're going to use the credentials you created when you set up your Active Directory. Instead, click **More choices** near the bottom of the popup, then click **Use a different account**

   For **User name**: use the <Active Directory NetBios name>\Admin. For example: **mystudio\Admin** *You can find the Active Directory NetBios name on the cheat sheet.*

   The password will be what you entered above when you set up your Active Directory. *This password can also be found on the cheat sheet.*

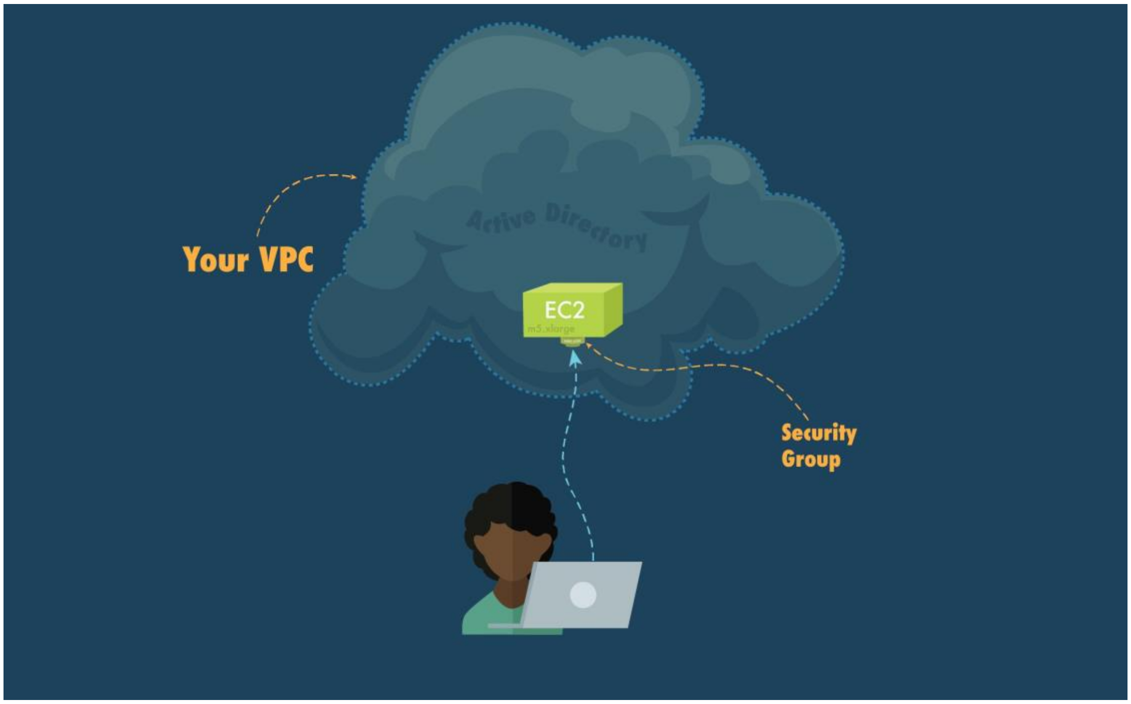


4. Click **OK**
5. Click **Yes** in the next popup window

If all goes well, you should connect to the desktop of your virtual workstation with the Active Directory administrator account!

Your Studio in the Cloud VPC So Far

Let's review what we've done up to this point. Here's an illustration of the current state of your VPC:



You’ve got your custom VPC and inside it are one Active Directory serving two subnets. And inside Public Subnet A you have one EC2 instance currently running, connected to your local machine through an Internet gateway with the traffic controlled by your security group.

In our next tutorial, we’ll continue building on the parts that you’ve already put in place and add new ones. Our next task is to create a shared file system and set up accounts for your users.

If you’ll be moving immediately to the next tutorial, you can stay logged in to your User Management instance and go directly to: [Tutorial 3: Setting Up an FSx File System and User Accounts]

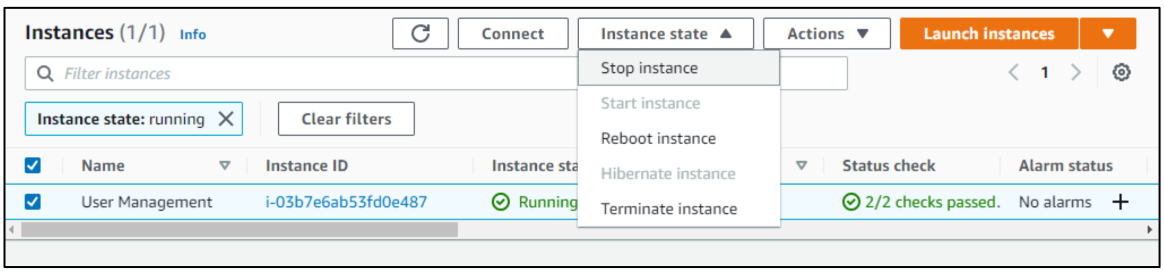
However, if you will be continuing your setup on another day, you’ll want to disconnect from and temporarily stop your instance to save resources (and money!).

Shut Down Notes

At the moment you only have one instance running for user management, so the shutdown notes for this tutorial are short. The procedure is almost identical to the steps you used to terminate the instance you created in the starter exercise in Tutorial 1, except this time instead of terminating your instance you'll be stopping it instead. This way, you'll be able to easily start it up again when you're ready to continue.

Stopping an Instance

1. In the Remote Desktop session for your instance, disconnect by going to **Start Menu→Power→Disconnect**
2. In the AWS Console go to **Services→EC2**
3. Near the top of the page, click **Instances (running)**
4. Select your **User Management** instance from the list
5. Click the **Instance state** button, then select **Stop instance**



Appendix

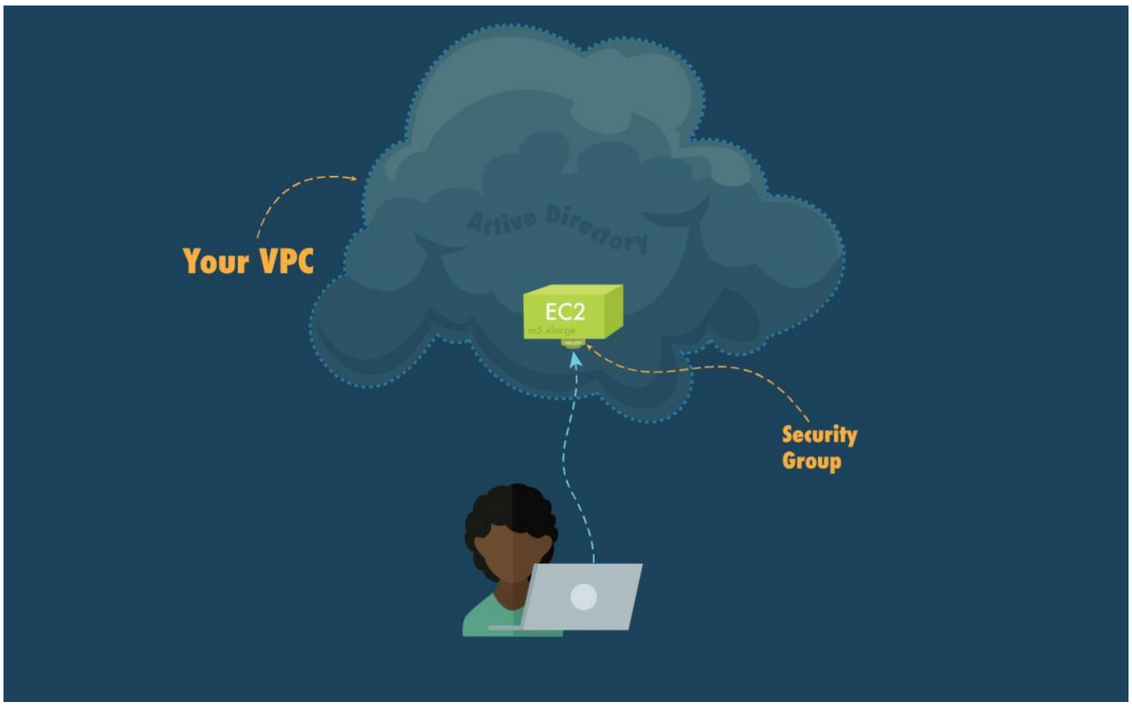
Links to AWS Documentation

- What Is Amazon VPC?
- Internet Gateways
- Security Groups for Your VPC
- Active Directory Prerequisites
- What is IAM?
- Seamlessly Join a Windows EC2 Instance

Tutorial 3: Setting Up an FSx File System and User Accounts

Estimated Time to Complete: 2 hour

In this tutorial, we'll set up accounts for your artists. We'll be creating a file system for storing user profiles and other studio data. By the end of the tutorial, you'll be able to login as a user with your own profile and access your shared storage.



Startup Notes

If you're coming straight from the last tutorial and you already have your User Management instance running, you can skip this section and continue straight to [Create an FSx Security Group](). But if you stopped your User Management instance at the end of the last tutorial, you'll need to start it back up before continuing.

Restarting a Stopped EC2 Instance

1. From the AWS Console go to **Services→EC2**
2. Click the **Instances (running)** link near the top of the page
3. Select your **User Management** instance

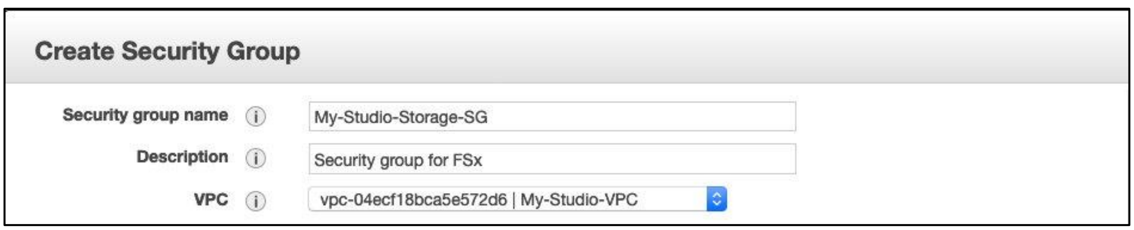
4. Click **Actions**, then select **Instance State→Start**

While the instance is starting up, continue on to the next section to start setting up accounts for your artists.

Create an FSx Security Group

In order to access our shared storage we need to setup a security group that will allow this drive to be mounted onto our virtual workstations. We will attach this security group to the FSx file system that we create in the next step.

1. Go to **Services → EC2**
2. In the left panel, under **Network & Security** click **Security Groups**
3. Click **Create security group**
4. Give your security group a name (e.g., My-Studio-Storage-SG) and a description. *Take a moment to enter storage security name and ID in the* [Important Information Cheat Sheet](#)*.*
5. Choose your VPC that you made (e.g., My-Studio-VPC). *You can find this on the cheat sheet under Tutorial 2.*



6. Click **Add rule**
7. Add these rules on the **Inbound** tab, and click **Create security group** when you are finished:

	Protocol	Port	Source	Description
Custom TCP Rule	TCP	135	10.0.0.0/16	RPC
Custom UDP Rule	UDP	445	10.0.0.0/16	SMB
SMB	TCP	445	10.0.0.0/16	SMB
Custom UDP Rule	UDP	1024-65535	10.0.0.0/16	FSx Ephemeral ports for RPC
Custom TCP Rule	TCP	1024-65535	10.0.0.0/16	FSx Ephemeral ports for RPC

Add a Tag to Your Security Group

Next, let's add the same Studio tag as before to help keep everything organized.

1. On the **Tags** tab, click **Manage tags**.
2. In the **Manage tags** pop-up window, click **Add new tag**.
3. For **Key** enter **Studio**
4. For **Value** enter the name you picked for your studio (e.g., My-Studio) *You can find your studio name on the cheat sheet.*
5. When you're done, click **Save changes**.

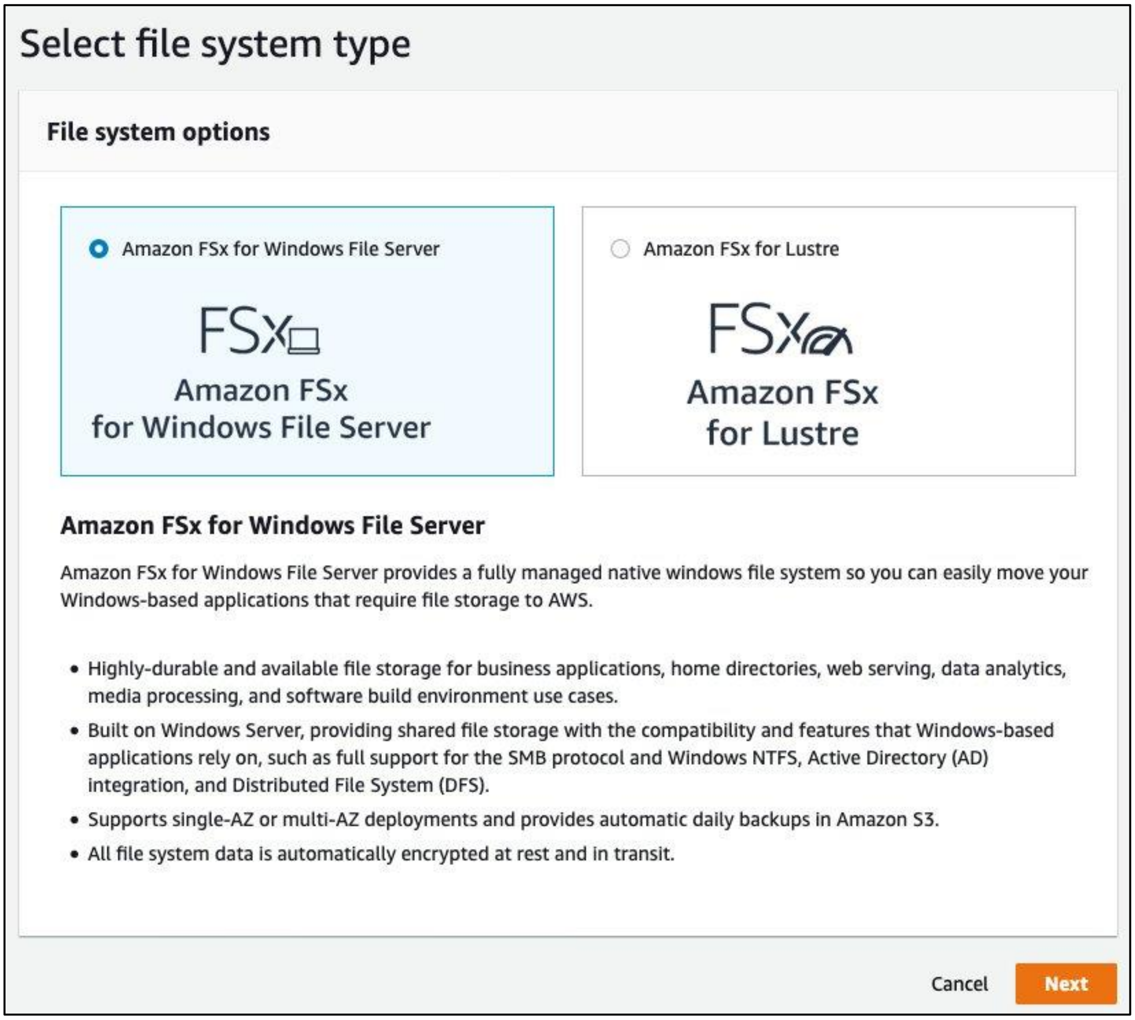
Create an FSx File System

Amazon FSx is a fully managed file system that provides high performance shared file storage. It is well suited to support content creation pipelines, as it can automate time-consuming administration tasks such as hardware provisioning, software configuration, patching, and backups. Although not covered in this tutorial, other options for file systems include Qumulo, and WekaIO.

In this step we'll create an FSx file system for all your shared storage, including user profiles, applications, and project data.

1. Go to **Services** → **Storage** → **FSx**
2. Click **Create file system** to start the file system creation wizard.

3. Choose **Amazon FSx for Windows File Server** and click **Next**.



4. For **File system name** enter **studio**.

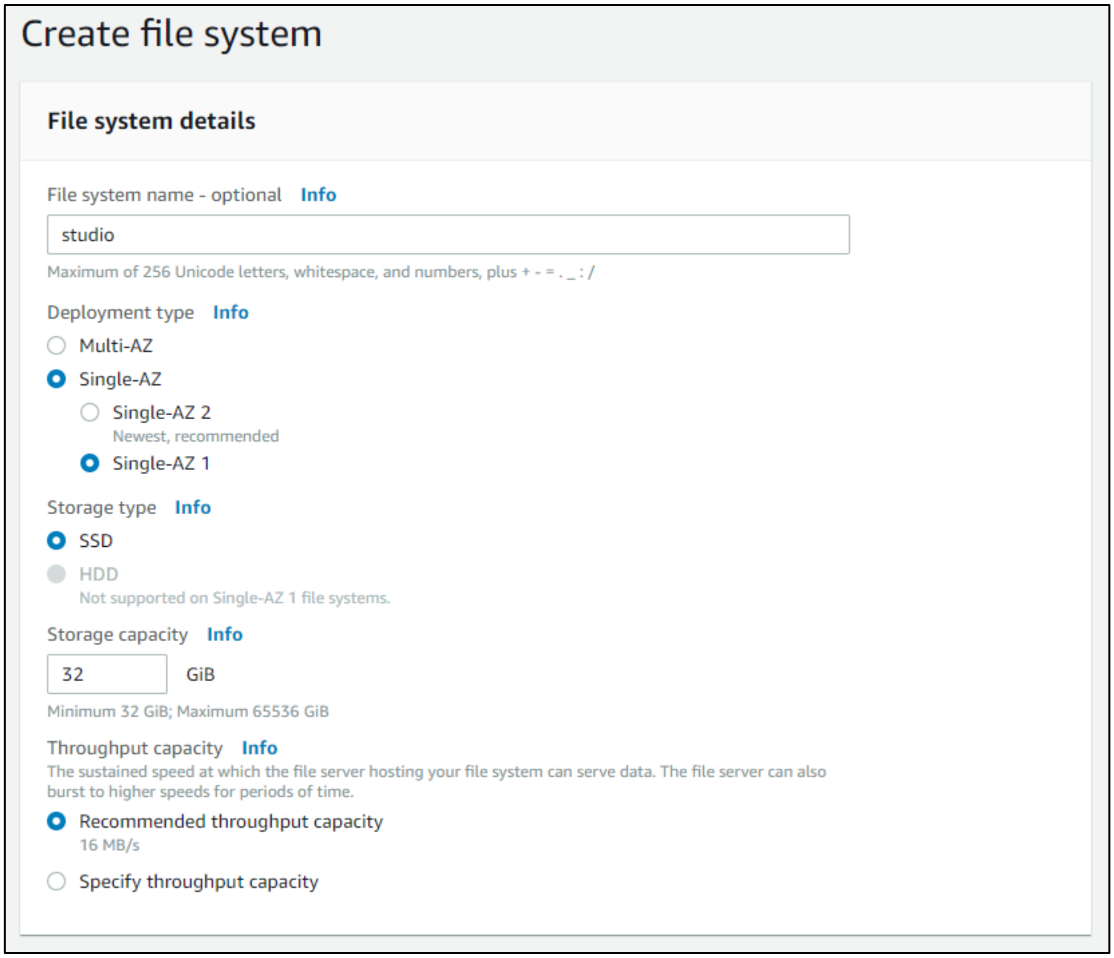
5. For **Deployment type** select **Single-AZ**, then choose**Single-AZ**, then **Single-AZ 1**.

   Note: You select the older **Single-AZ 1** because it has support for custom DNS names, which you use later to create an easy-to-use alias for your file system. Single-AZ 2 is newer, but does not currently have support for custom DNS names.

6. Set the **Storage capacity** to the minimum value listed below the entry field. In most regions the minimum is **32 GiB**.

   If you think you'll need more storage you can increase the capacity at this time.

7. For **Throughput capacity** leave **Recommended throughput capacity** selected.



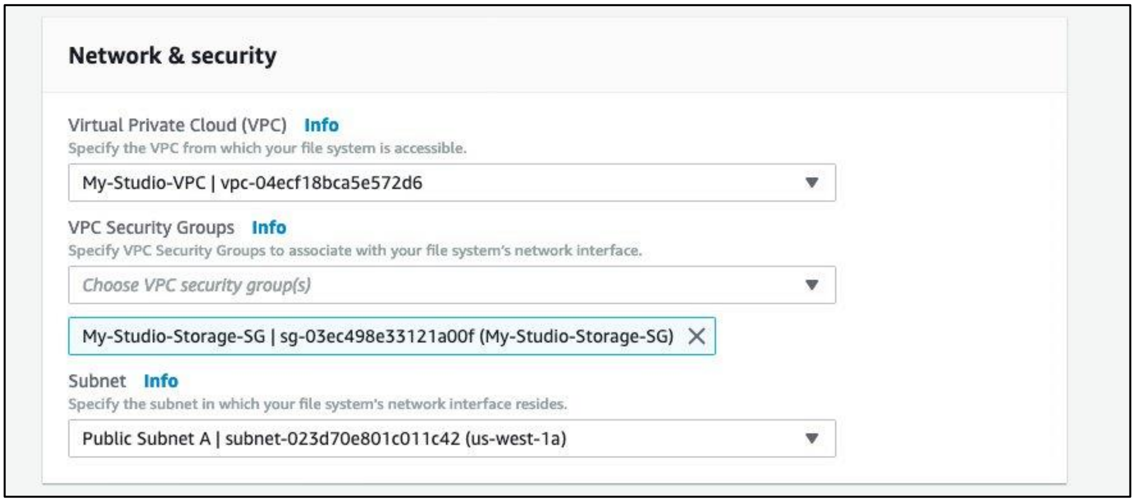
8. Choose the **VPC** that you created in the last tutorial (e.g., My-Studio-VPC). *Refer to the cheat sheet for your VPC name, if needed.*

9. Choose the **Storage Security Group** that you just made (e.g., My-Studio-Storage-SG). Note: Be careful to select the storage security group that you created above and not the remote desktop security group that you created in Step 02. *Refer to the Storage Security Group Name and ID on the cheat sheet.*

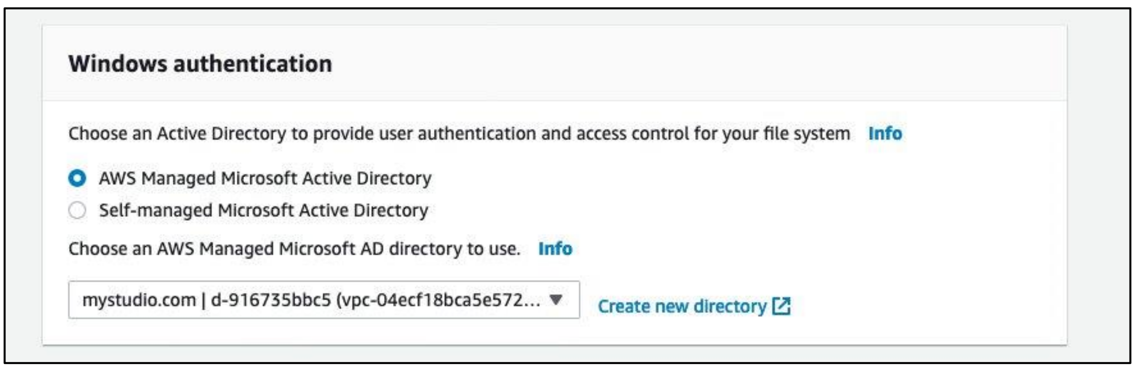
10. You can remove the default security group by clicking the “X” next to its name.

    Don’t worry if you make a mistake with security groups, you can always change them later by following [these directions].

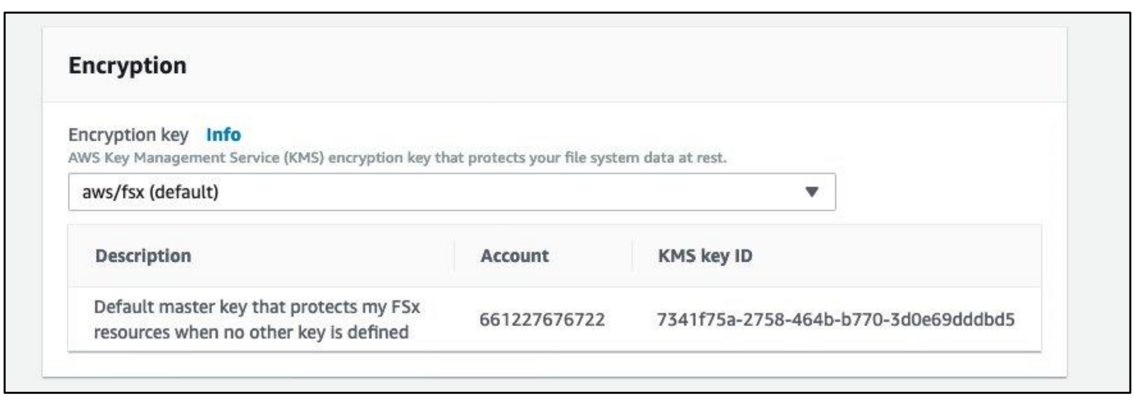
11. Make sure the **Subnet** is set to **Public Subnet A**.



12. For **Windows authentication**, leave **AWS Managed Microsoft Active Directory** selected and choose your **Active Directory DNS Name** from the list. *Refer to the cheat sheet if needed.*

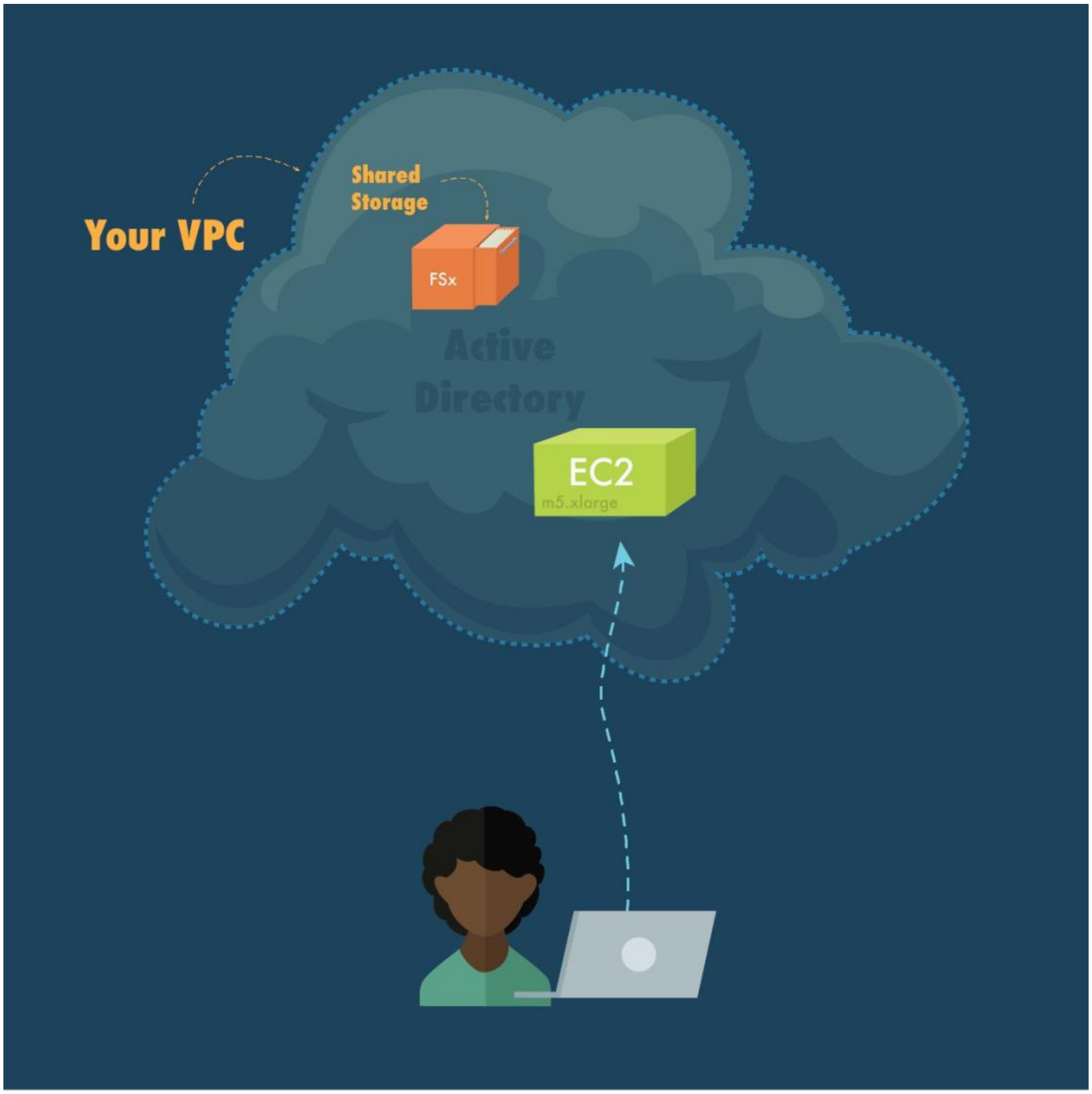


13. Keep **Encryption** at default.



14. Click **Next**.
15. Review the file system configuration on the **Create file system** page, and click **Create file system**.

The file system will take about 20 minutes to create. While you're waiting, move on to the next section.



Install Active Directory Tools

While you're waiting for your FSx file system to create, we'll occupy your time by installing some Active Directory tools on your User Management instance. In the last tutorial, we created an Active Directory to store your user accounts, but we didn't actually add any users, aside from the Admin, which is created by default. Before we can add new users for everyone in your studio, we need to install the proper Active Directory tools.

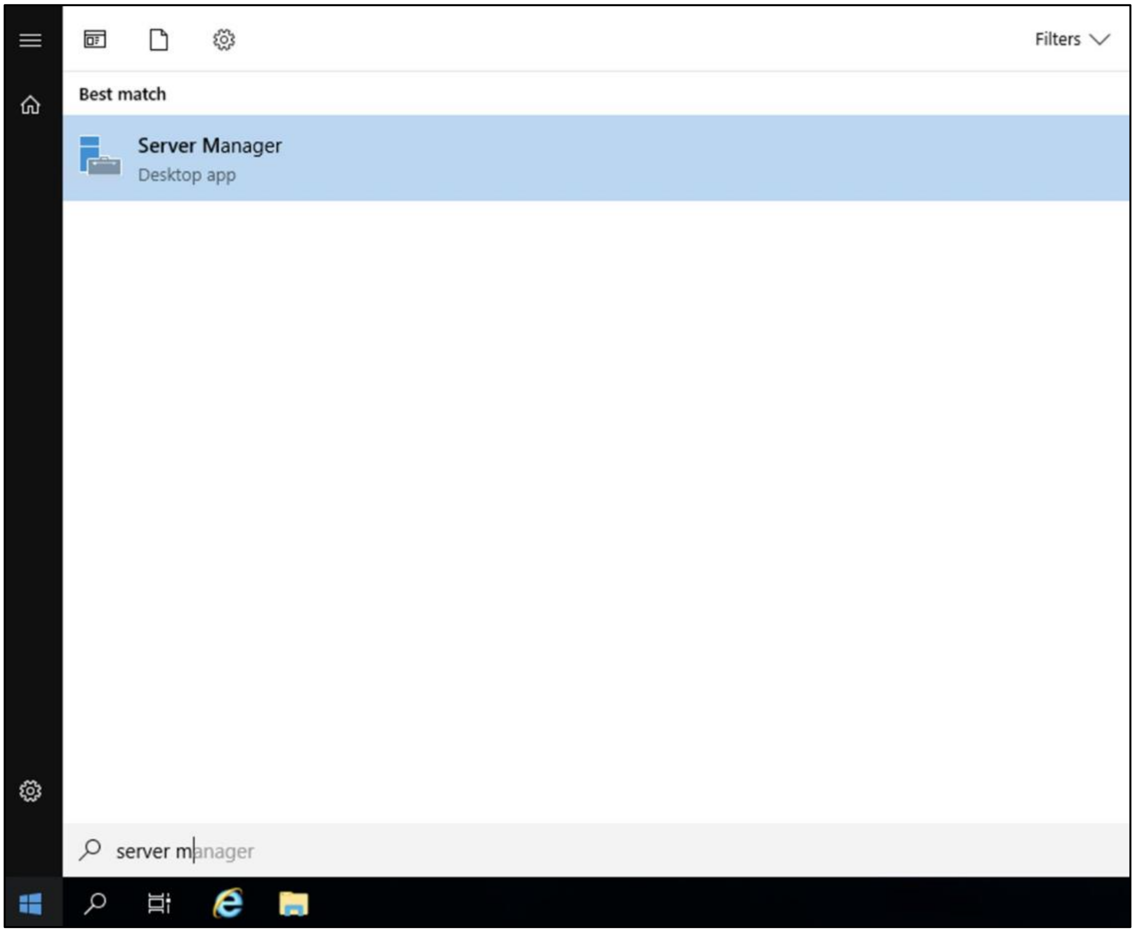
Connect to Your User Management Instance

If you haven't already, you'll need to connect to your **User Management** using **Remote Desktop Connection**. If you're already connected, you can skip to the next section.

1. Go to **Services→EC2**
2. Click **Instances (running)**.
3. Select your **User Management** instance.
4. Click **Connect**.
5. Choose the **RDP client** tab, then click **Download the Remote Desktop File**.
6. Open the **Remote Desktop File**. Click **Connect** or **Continue.**
7. A window asking for the Administrator password will pop up, but like we did at the end of Tutorial 2, we're going to use your Active Directory Admin login information. Click **More choices** near the bottom of the pop-up, then click **Use a different account.**
8. For **User name**: use the <Active Directory NetBios name>\Admin. For example: **mystudio\Admin** *You can find the Active Directory NetBios name on the cheat sheet.*
9. Click **OK**, then click **Yes** in the next pop-up window.

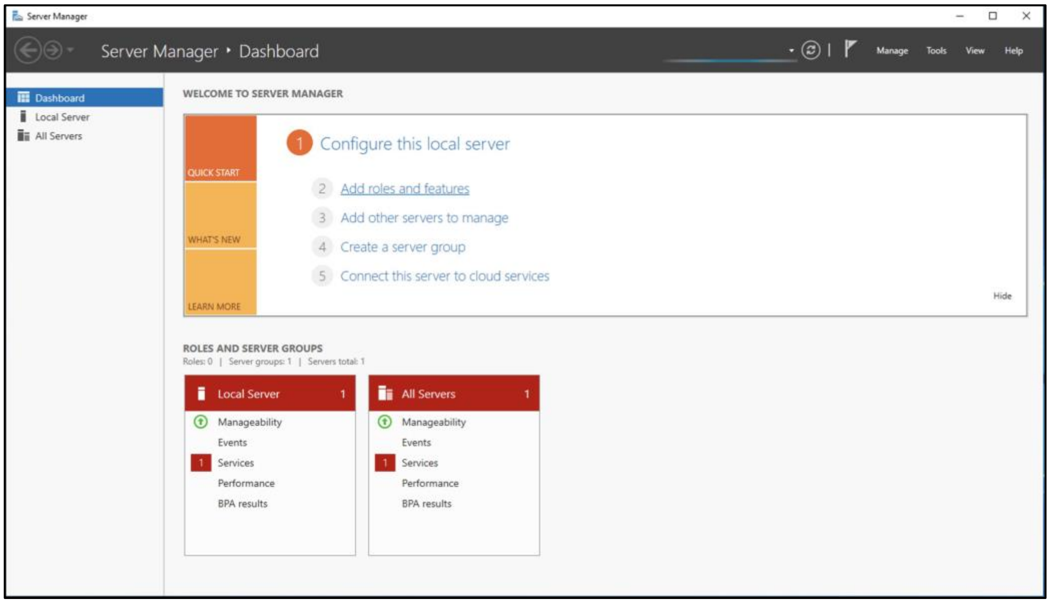
Install Role Administration Tools

1. On your User Management instance click the **Windows start menu** and type in **Server Manager.** Select the **Server Manager** app.

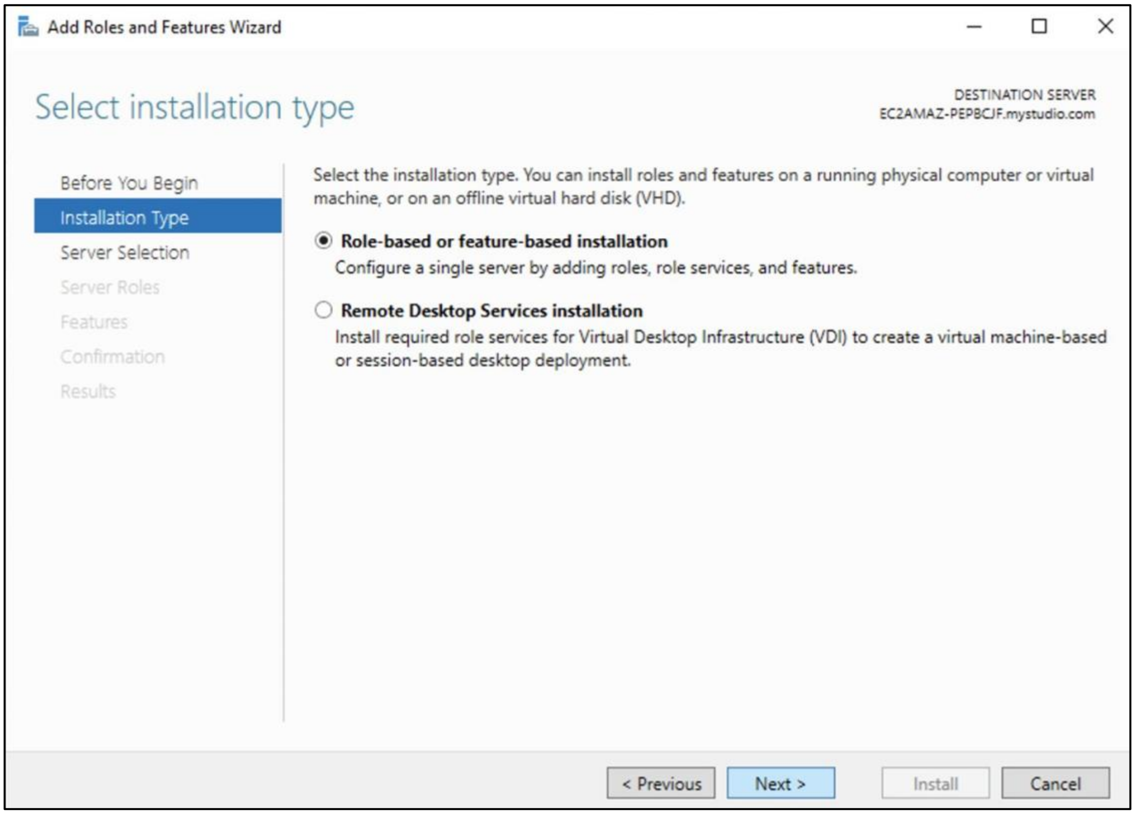


2. In the **Server Manager Dashboard**, choose **Add roles and features**.

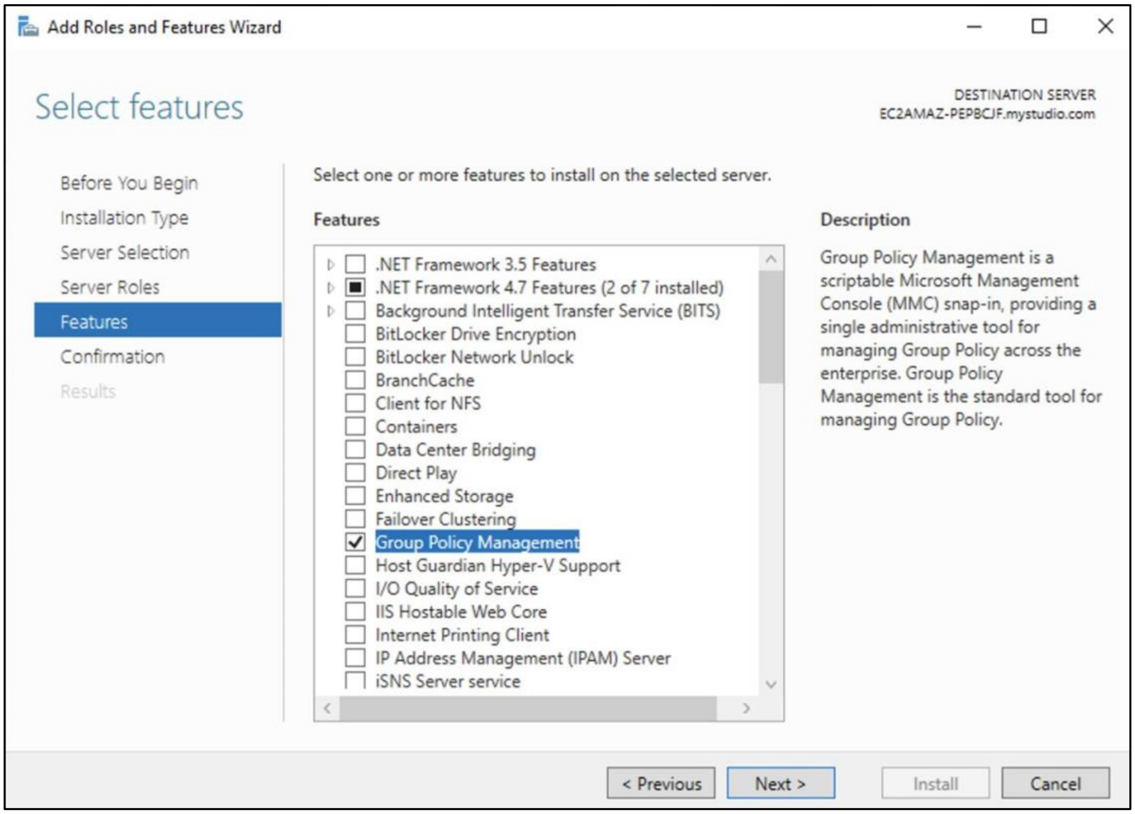
Note: It may take a minute for the server to be done collecting data.



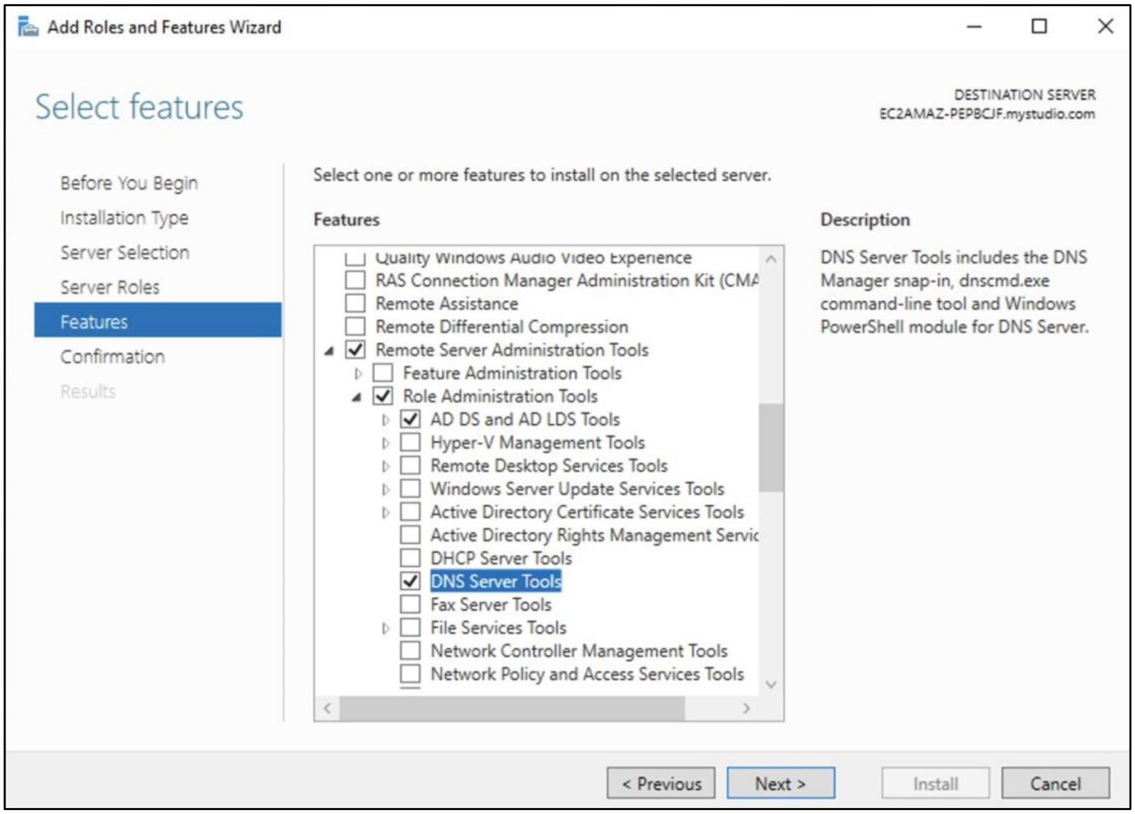
3. In the **Add Roles and Features Wizard** click **Installation Type**, select **Role-based or feature-based installation**, and choose **Next**.



4. Under **Server Selection**, your instance should already be selected in the server pool list. Then click **Features** in the left navigation pane.
5. Then, in the **Features** tree, click the check box next to **Group Policy Management.** This will make it possible to automatically map the FSx drive for every user.



6. Next scroll down in the list and open **Remote Server Administration Tools→ Role Administration Tools**, select **AD DS and AD LDS Tools**, scroll down, and select **DNS Server Tools**, and then choose **Next**.



7. Review the information and choose **Install**. When the feature installation is finished, you can click **Close** and exit Server Manager. The Active Directory tools are now available in the Start menu in the **Windows Administrative Tools** folder.
8. Stay logged into your instance while you continue to the next step.

Create an Alias for Your FSx File System

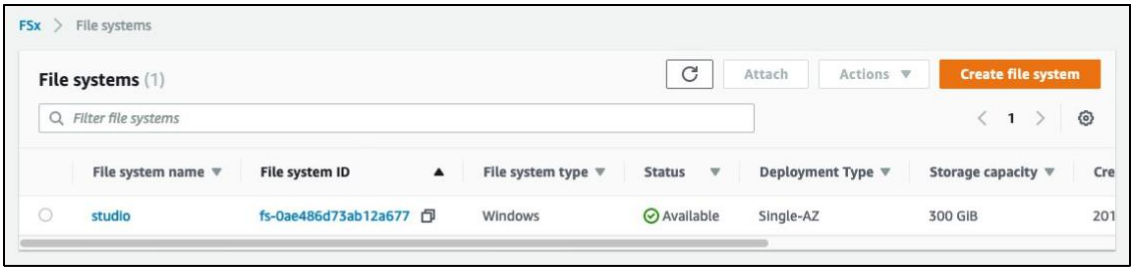
Let's briefly talk about **Fully Qualified Domain Names** (FQDN). The FQDN is what other computers use to identify and connect to a file system. The default FQDN for your FSx file system is really long and hard to remember, so we're going to create a much simpler alias for it that will make it easier to use.

Before creating an alias, we need check that our FSx file system has been successfully created and is available for use.

Check FSX Setup Status

1. Back in the **AWS Console**, go to **Services→FSx**

2. Look at the **Status** for the file system you created above. If that has a green check mark and says **Available**, then you are good to go. If the status still says



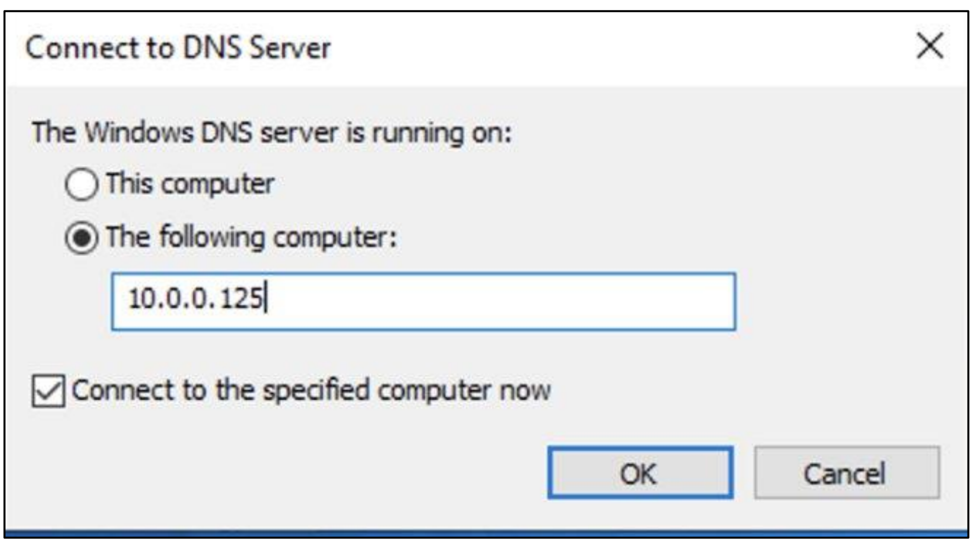
Creating, wait until the status changes before continuing.

Once the file system is available, click its name in the list and then write down two pieces of information on your cheat sheet: under Summary, find the File system ID and under Network & Security, find the DNS name.

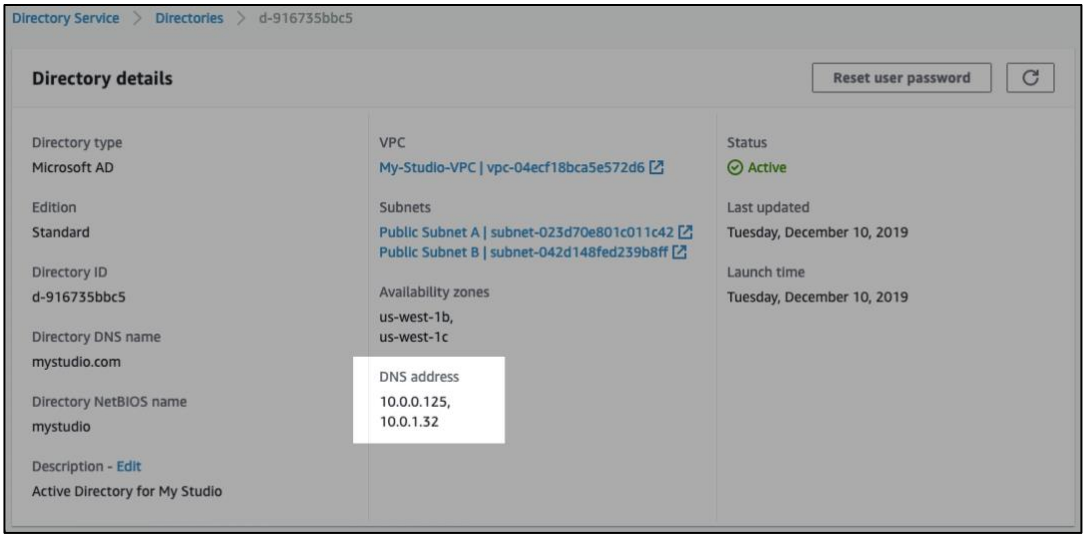
Create the Alias

1. Back on your User Management instance, click the **Start Menu,** type **dns** and then open the **DNS** application.
2. When it asks you to connect, pick **The following computer,** enter the **DNS address** of your **Active Directory,** then click **OK**.

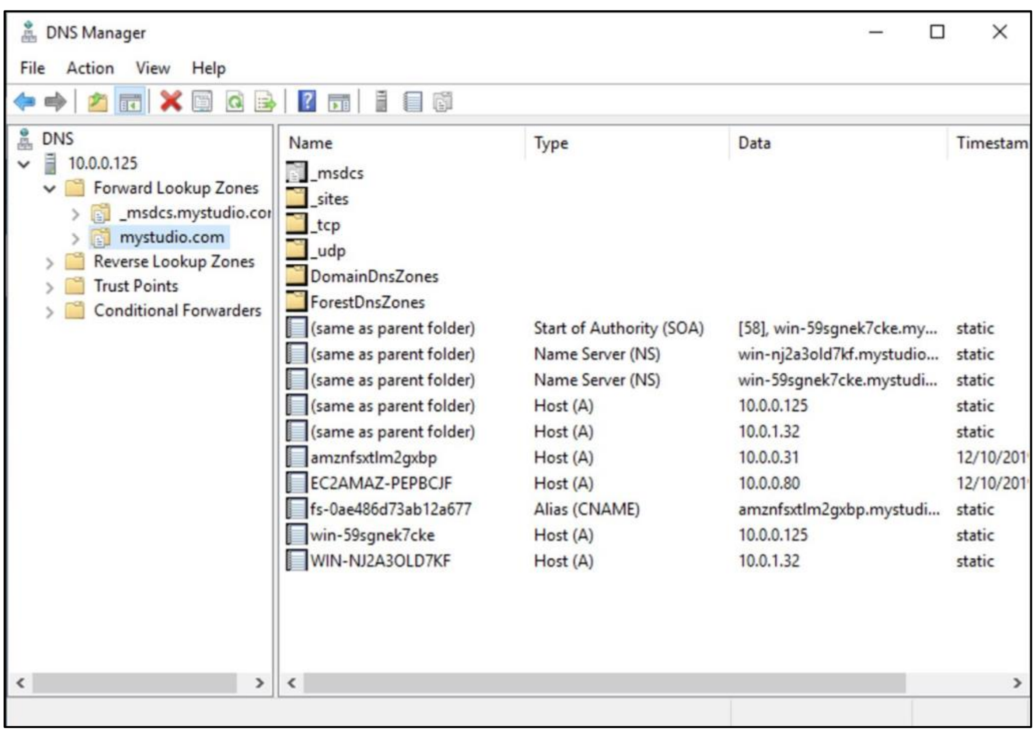
   Note: Your **Active Directory** has two DNS addresses, one that starts with "10.0.0" and one that starts with "10.0.1". You only need to enter the one that starts with "**10.0.0**" here (e.g., 10.0.0.125).



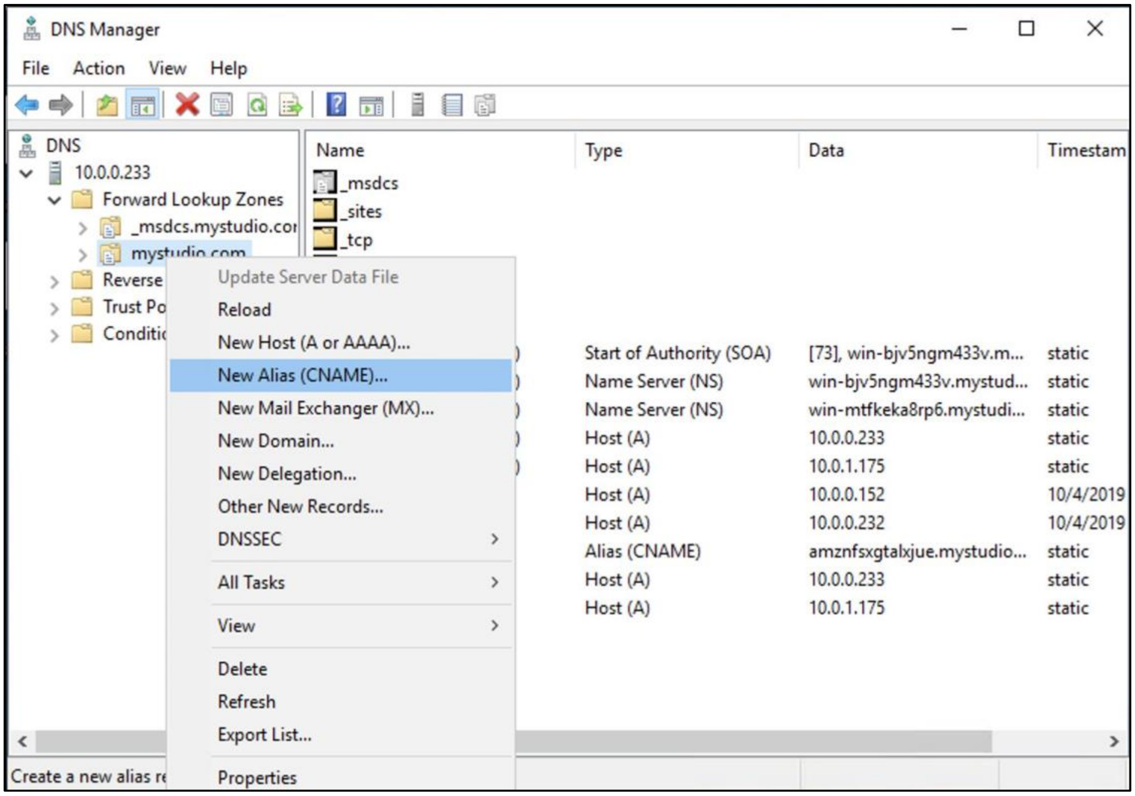
You can find the DNS addresses if you go to the **Services →Directory Service** in your AWS Console, and click the directory ID of your Active Directory. The DNS addresses will appear under Directory details at the bottom of the middle column. *At this time, you should also enter the two Active Directory DNS addresses on your cheat sheet.*



3. In the **DNS Manager**, first select the DNS address of your active directory in the left navigation pane, click the **>** next to it, and then click the **>** in front of **Forward Lookup Zones**.



4. Next, select your Active Directory (e.g., mystudio.com), and click with the **Right Mouse Button** over the directory (or go to the **Action** menu) and choose **New Alias (CNAME)...**



5. In the **New Resource Record** dialog box:

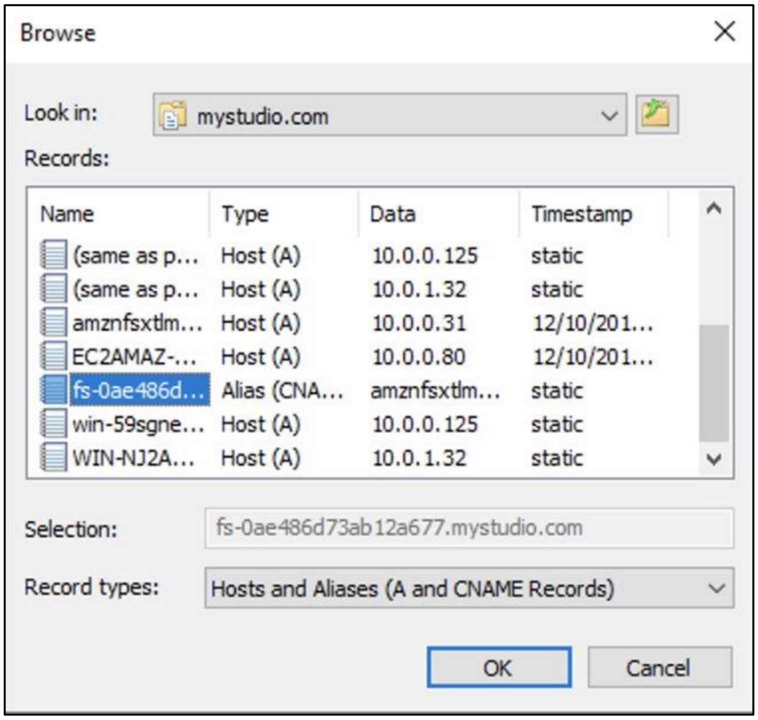
   a. For **Alias name**, enter **studio**.

      After you enter the Alias name, the field below will automatically update with the new FQDN for your file system (e.g., studio.mystudio.com). *Write down the new aliased FQDN on your Important Information Cheat Sheet.*

   b. For **Fully qualified domain name (FQDN) for target host**, choose **Browse**

      - Double-click the IP address (if it's there).
      - Double-click the **Forward Lookup Zone.**
      - Double-click the name of your **Active Directory** (e.g., mystudio.com).

- Scroll down until you find your **FSx file system ID** and then **select** it (it'll be something like fs-0c0b4fab1db1b9a0e). *Refer to your cheat sheet for your FSx file system ID.*



c. Click **OK.**

6. Click **OK** again.

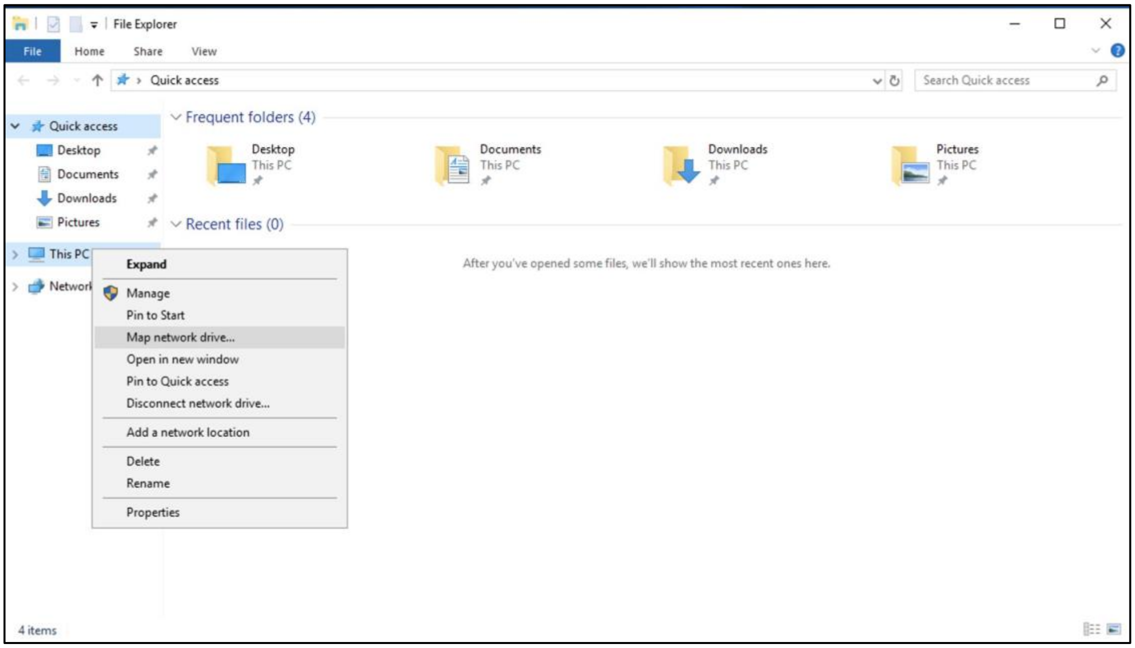
You've now added a much easier name to find your file system! Instead of having to remember that long name, you'll just need to remember the aliased FQDN (e.g., studio.mystudio.com).

Map the File System to Your Instance

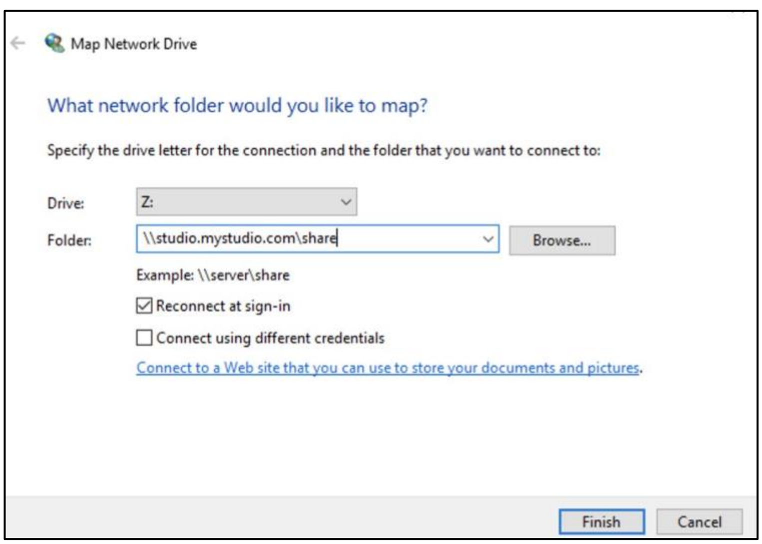
Next, we're going to map (or connect) your FSx file system to your User Management instance. We'll be using the FSx file system to store user profiles that you will create for your artists as well as to store studio tools, which we'll do in our next tutorial.

1. Open a **File Explorer** window.

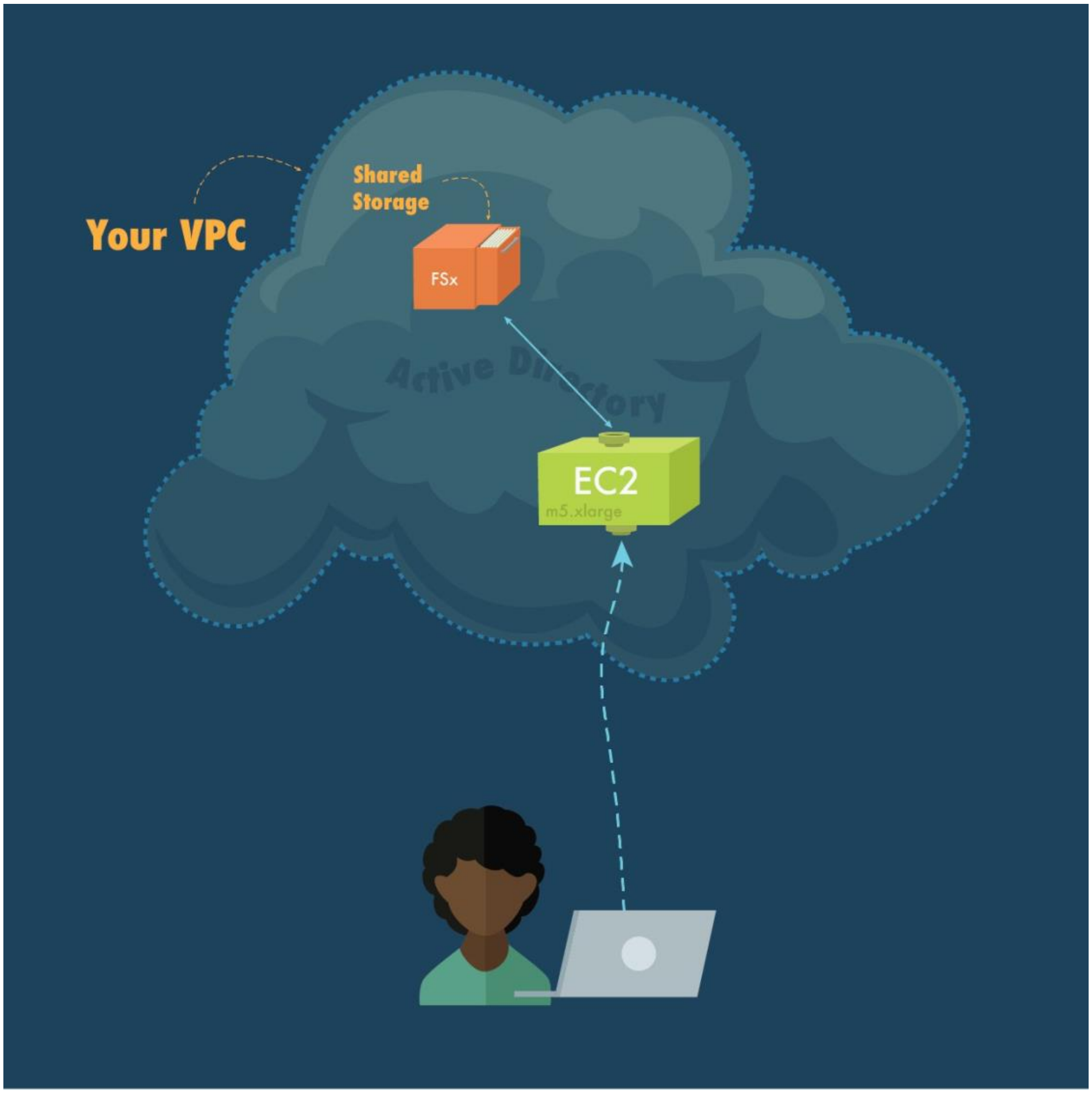
2. From the navigation pane, right-click **Network** and choose **Map Network Drive**.



3. Choose a drive letter of your choice for **Drive** (e.g., **Z:**).
4. For **Folder** enter the FSx aliased FQDN that you noted above (e.g., \\studio.mystudio.com\share\, make sure to add the "\\" at the beginning and the "\share\" at the end). *You can also refer to the cheat sheet for the FSx aliased FQDN.*
5. Click **Finish**.



File Explorer should open a window pointing to that drive location. At the moment, it's empty. However, we'll be filling it with folders and data very quickly!



Automatically Map the File System for All Users

It's great that the FSx drive is mapped for this particular instance, but in order to scale our studio we'll want to make this automatic for all of our users. To do this, we'll create a **Group Policy** to map the network drive.

Create Group Policy Object

1. Open the **Group Policy Management Console** by going to the **Start Menu** and typing **Group Policy**, then select the **Group Policy Management** app.

2. In the **Group Policy Management Console**, open up:

   Forest: <Active Directory DNS Name>

   → Domains

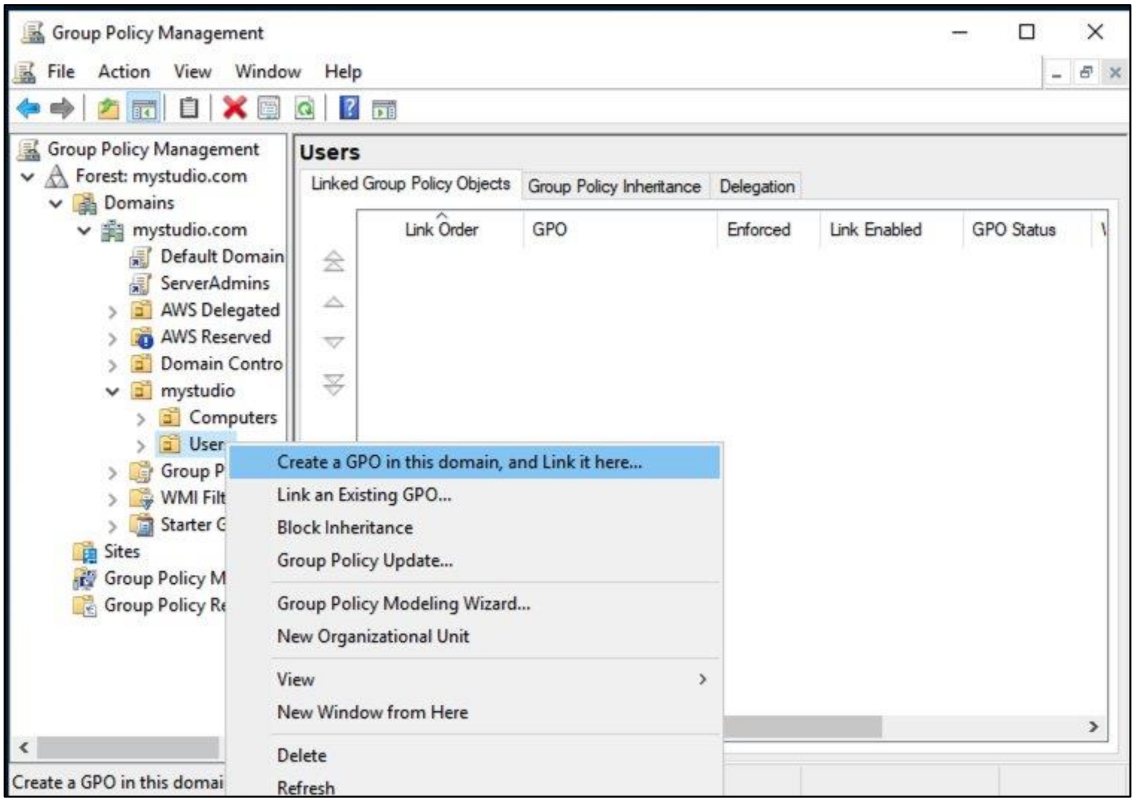
   → <Active Directory DNS Name>.com

   → <Active Directory NetBios Name>

   → Users

   e.g., Forest: mystudio.com → Domains → mystudio.com → mystudio → Users). *Refer to the cheat sheet for your Active Directory DNS and NetBios names.*

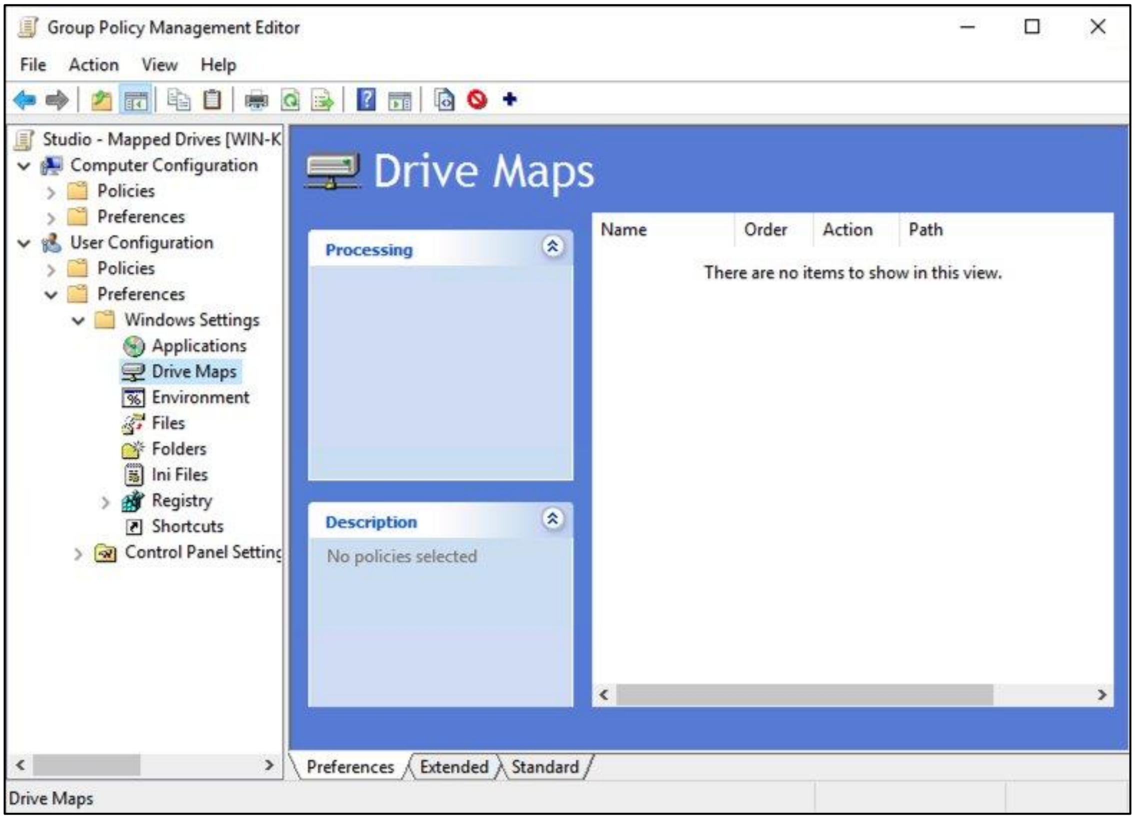
3. Right-click **Users** and choose **Create a GPO in this domain, and Link it here.**



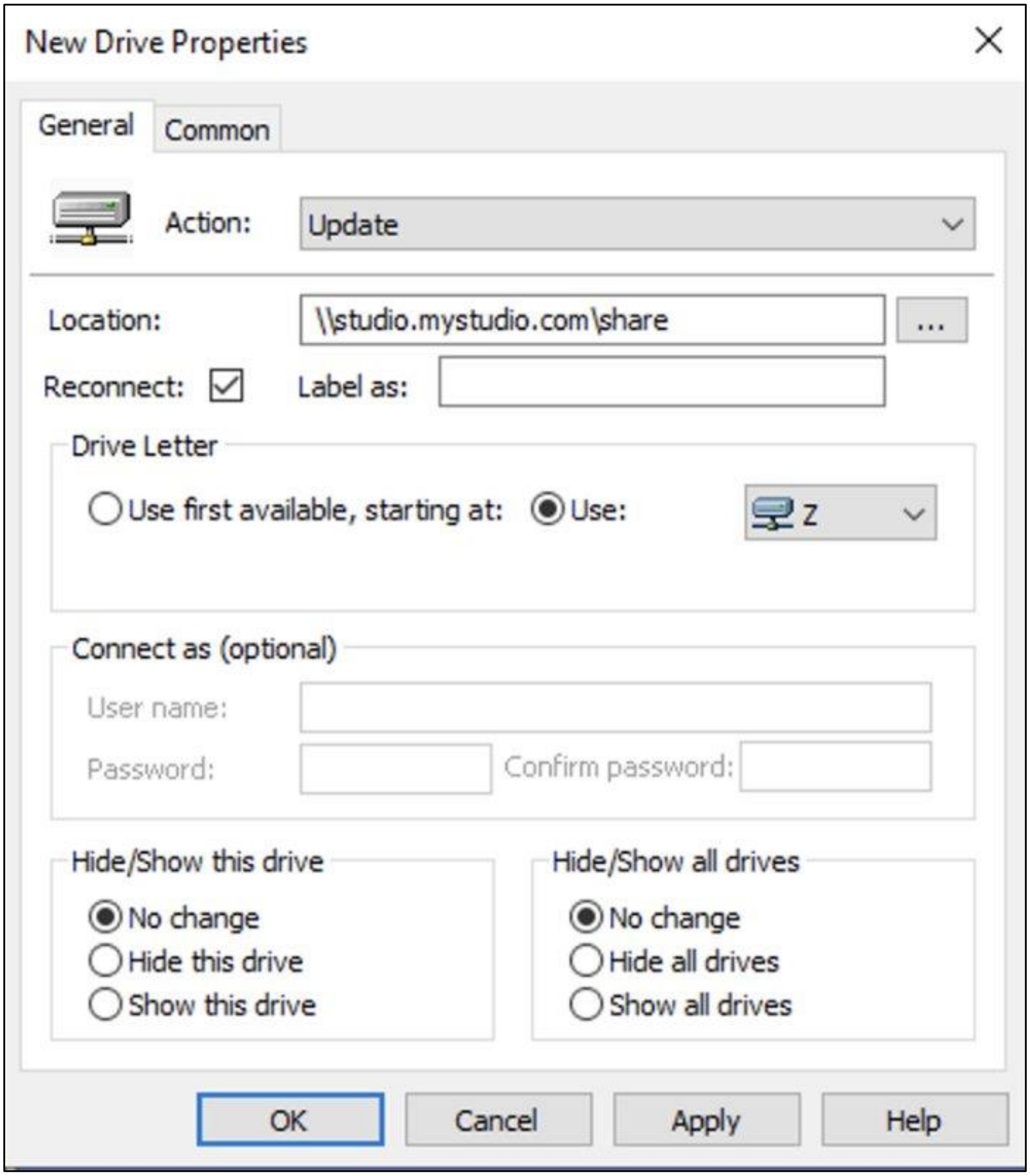
4. For **Name** enter **Studio-Mapped-Drives**.
5. Click **OK.**

Configure the Group Policy Object

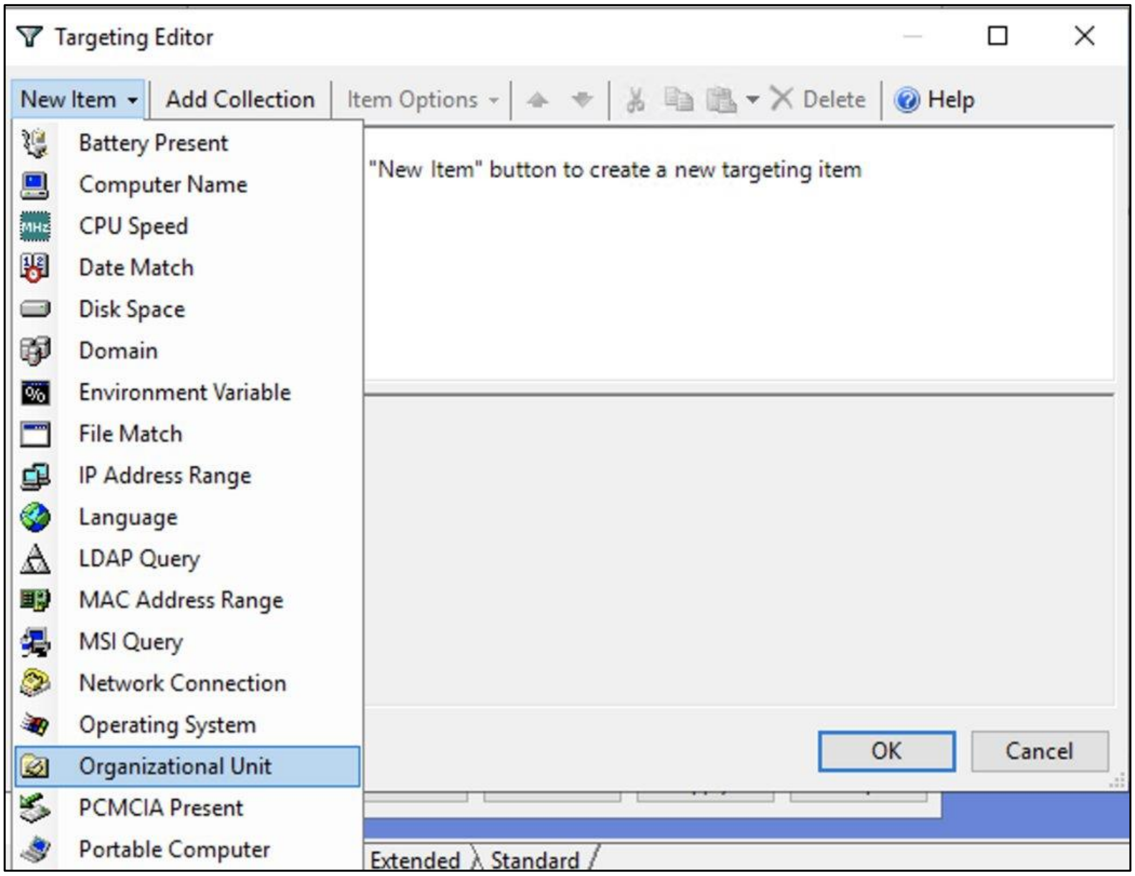
1. Under **Users**, right-click the **Studio-Mapped-Drives Group Policy Object** and choose **Edit.**
2. Navigate to **User Configuration → Preferences → Windows Settings → Drive Maps**



3. Right-click **Drive Maps** and select **New** → **Mapped Drive**
4. On the **General** tab:
   a. Set the **Location** to the location of your drive that you entered when mapping the drive in the last section (e.g., \\studio.mystudio.com\share).
   b. Select the checkbox next to **Reconnect.**
   c. Under **Drive Letter**, make sure “**Use:”** is selected, then open the dropdown menu and select the **Z** drive.

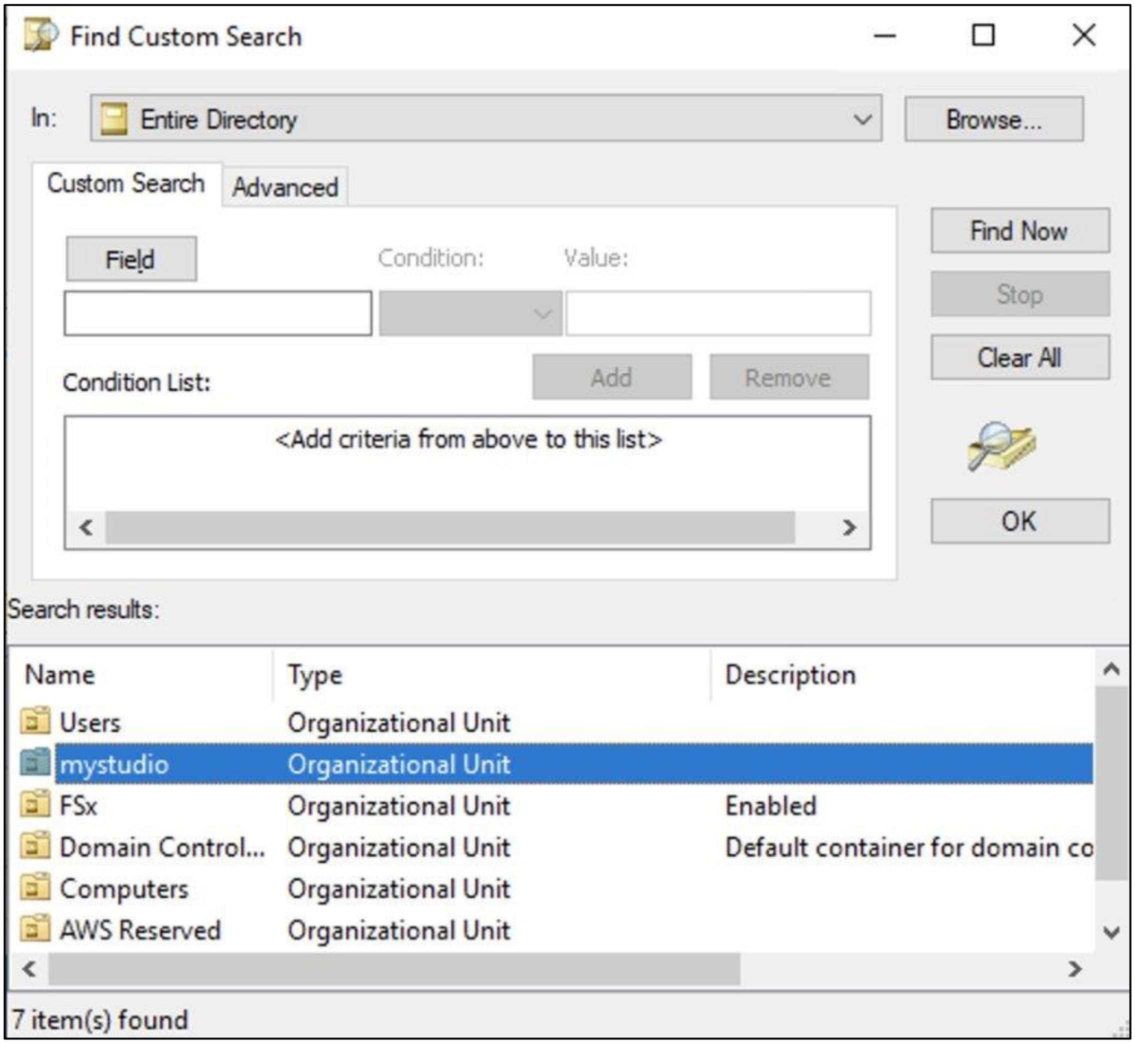


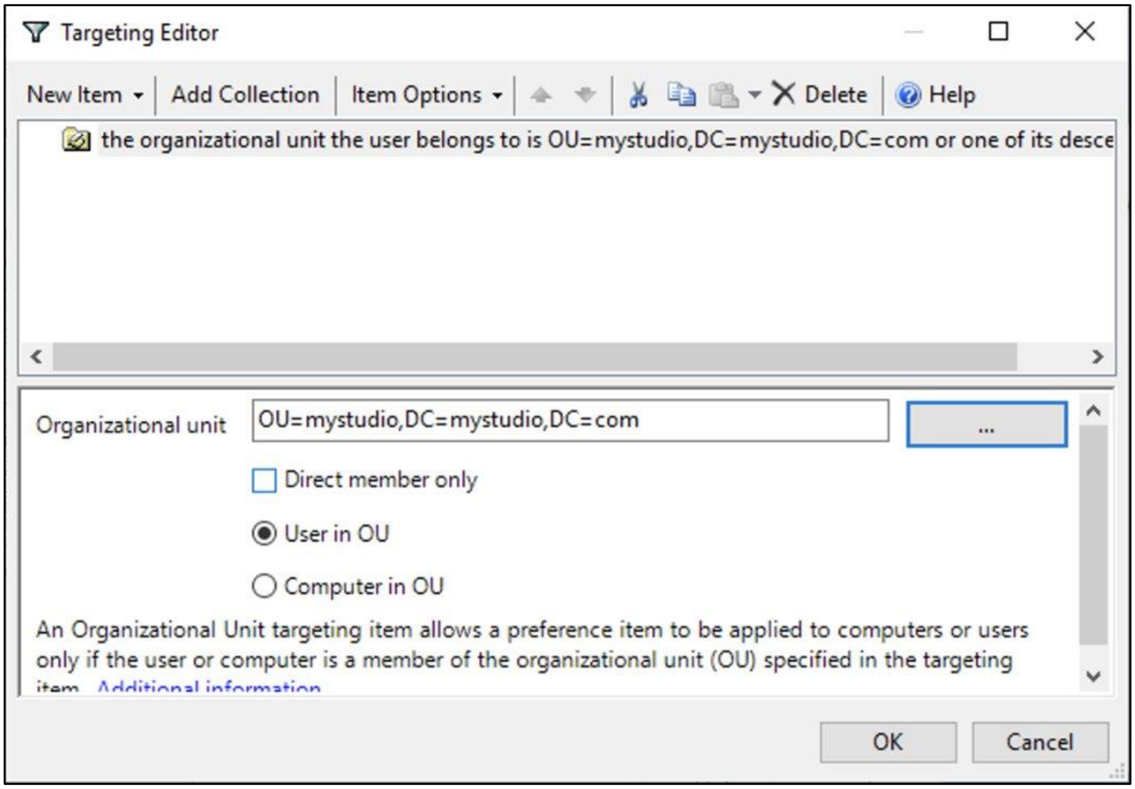
5. On the **Common** Tab:
   a. Select **Run in logged on user's security context**
   b. Select **Item-level Targeting**
   c. Click **Targeting**
   d. Select **New Item** → **Organizational Unit**



6. Click the **browse** button to choose the organization unit you want

7. When the window opens, double-click the name of your studio (e.g., mystudio).





8. Click **OK**, and **OK** again.
9. Close the Group Policy Management windows.

Create Accounts for Your Artists

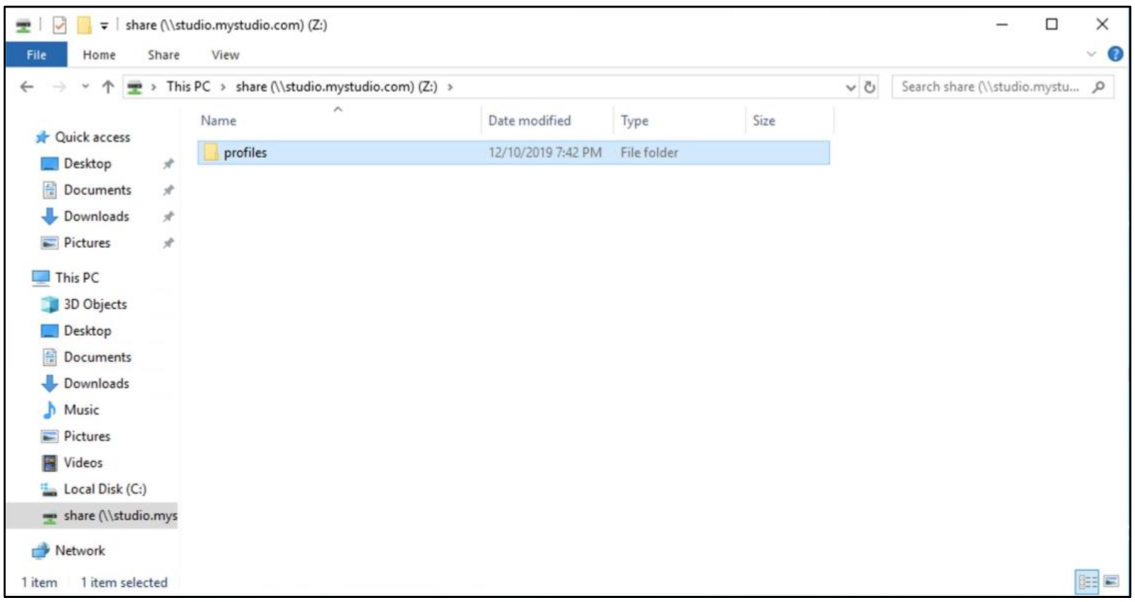
Each one of the artists in your studio is going to need their own user account. Some of the work we've already done has been in preparation for this. The Active Directory we created in the last tutorial provides a centralized location for the list of users in your studio. Each of those will also have profile data which includes environment settings, documents, and other data specific to each user.

One purpose of the FSx file share you created in the last section is to store that user profile data. That way this data can follow users around if they need to login to different instances in your studio.

Once you've mapped your FSx file system, it will appear in a **File Explorer** window as the drive letter you picked (e.g., Z:). Next, we just need to create a folder on that drive for user profiles.

Create a Profiles Folder

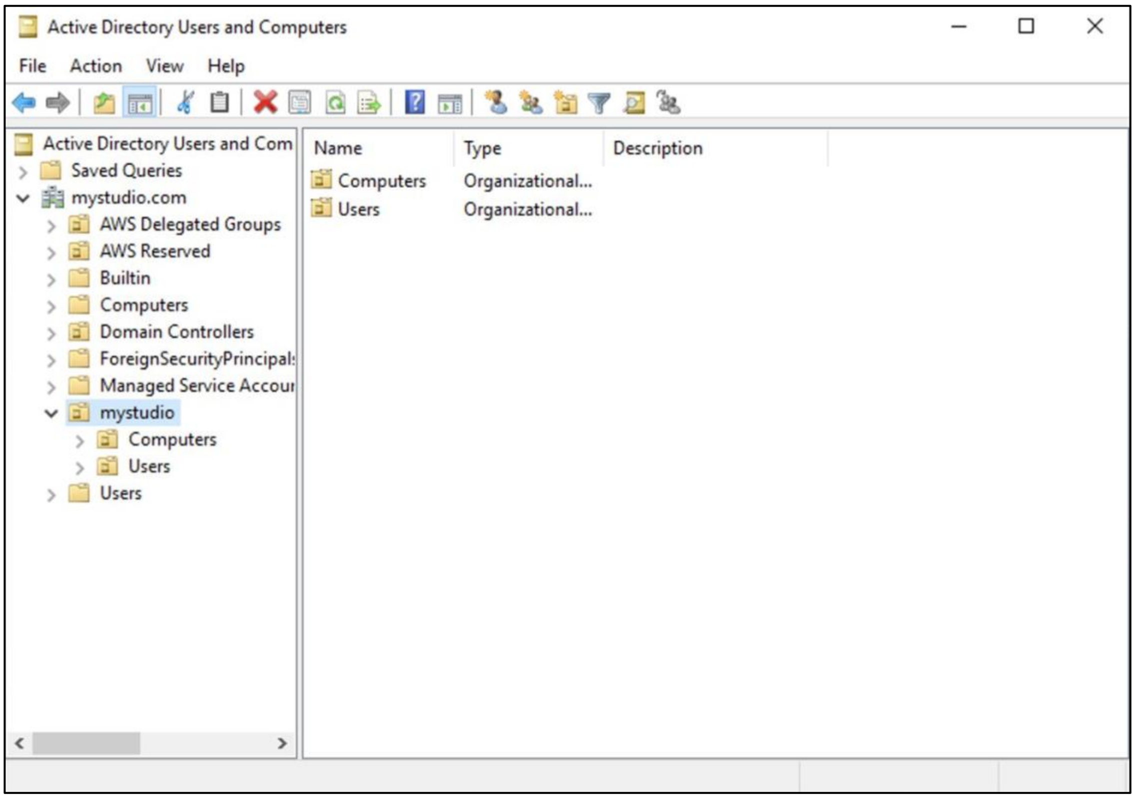
1. In **File Explorer** click **This PC,** then at the bottom double-click your newly-mounted FSx file system (e.g., Z:).
2. Right-click and choose **New→Folder**.
3. Name the new folder **profiles.**



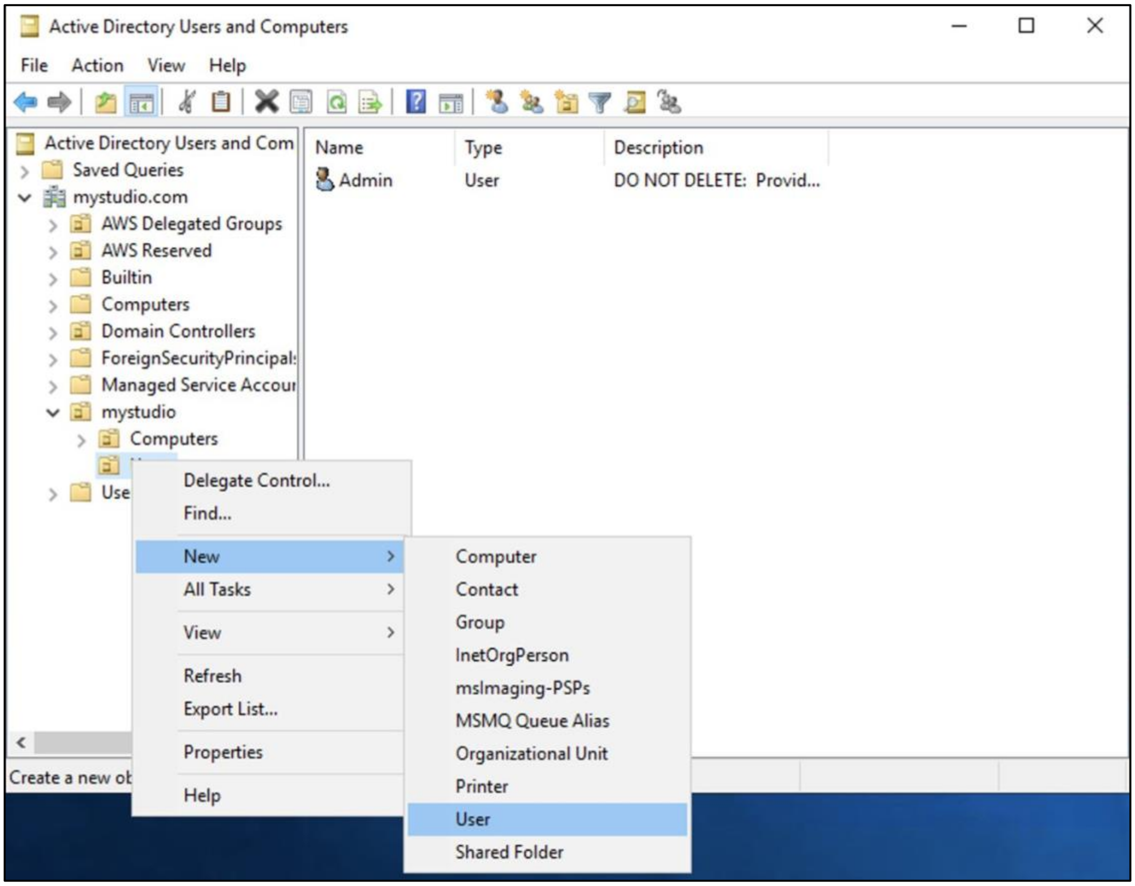
Add New Users

1. Open the Start Menu and type **Administrative Tools**. Select **Windows Administrative Tools**.
2. Double-click **Active Directory Users and Computers**
3. Expand your **Active Directory** (e.g., mystudio.com) by clicking the **>** to the left of its name.

4. Expand the OU (Organizational Unit) with your **AD's NetBios name** (e.g., mystudio).

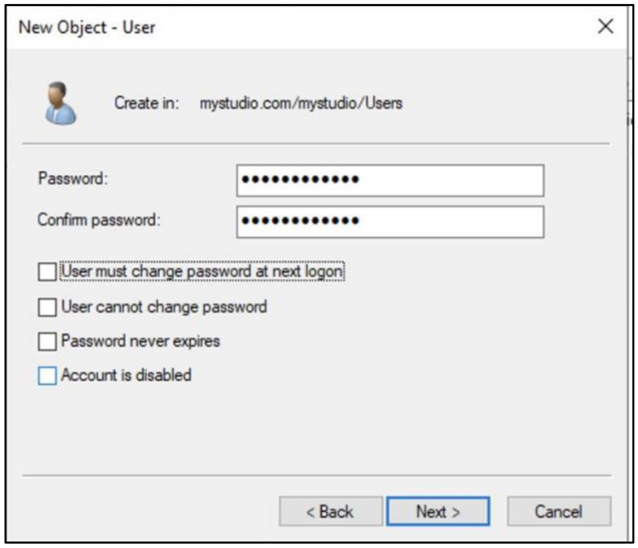


5. Right-click **Users** and select **New→User.**



6. Enter the user's information (name, logon name, etc) and click **Next.**
7. Set their password (use the master password or a unique one).

8. Clear the **User must change password at next logon** check box and



   click **Next**.

9. Click **Finish.**
10. Repeat for all artists on your team.

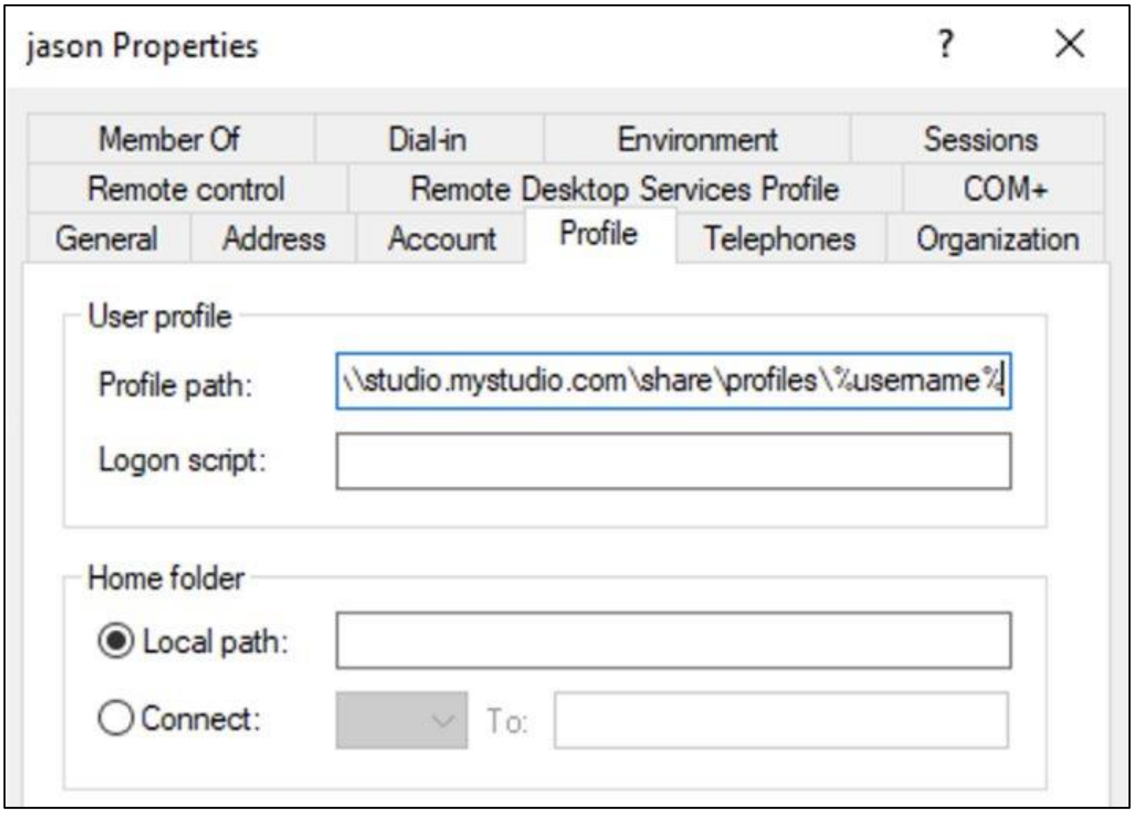
Set Up Profile Path

Now we want to modify the user's profile information to be looking at the FSx drive - this way their profile will follow them for each instance they log into.

1. In the left panel, under the OU with your AD's NetBios name (e.g., mystudio), select **Users**. <u>Note</u>: There is a second Users folder at the bottom, make sure you select the first one.
2. Double-click one of the new users to get the properties panel.
3. Click **Profile** tab
4. For the profile path, enter:

   \\[FSx_aliased FQDN]\share\profiles\%username%

(e.g., \\studio.mystudio.com\share\profiles\%username%)

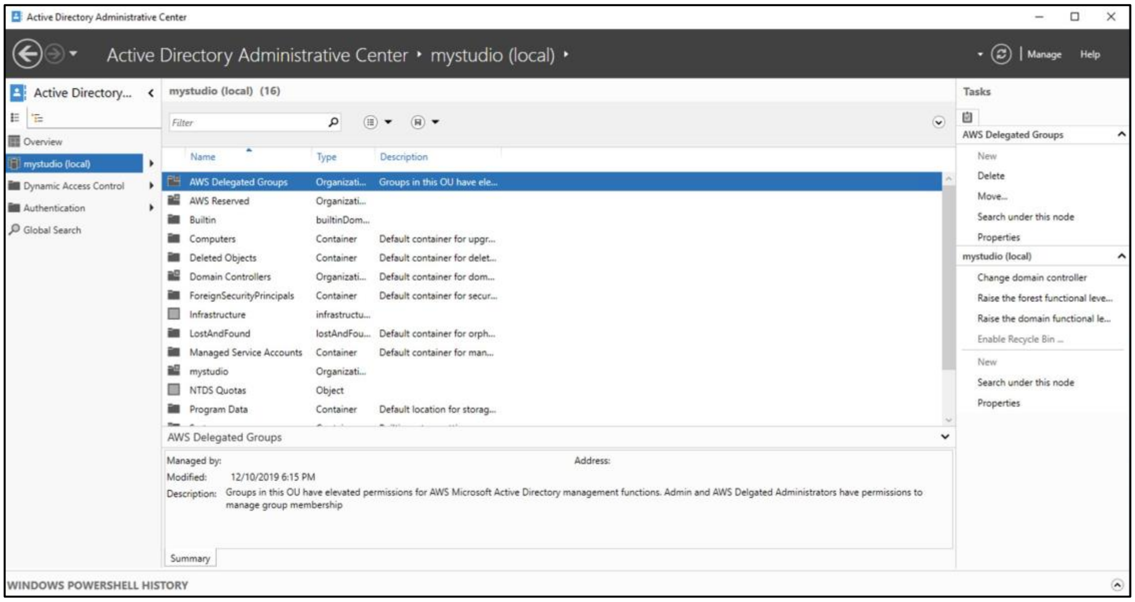


5. Click **OK**.
6. Repeat for all artists on your team.

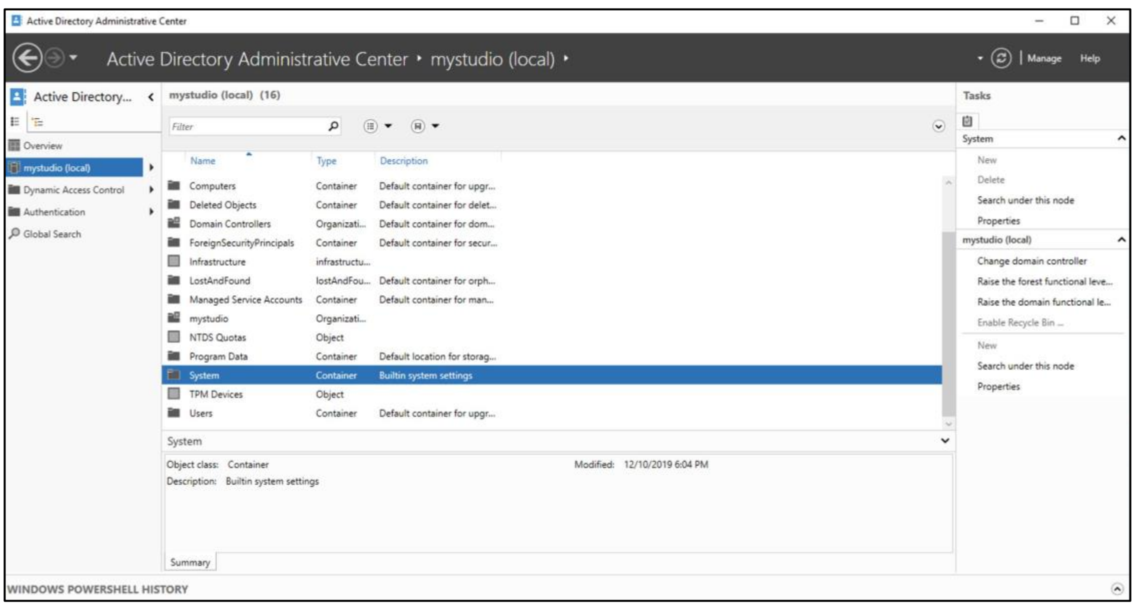
Edit the Password Policy

By default, there is a one day waiting period from the time you set a user's password to when they are allowed to change it. However, we want your users to be able to change their password immediately after logging in for the first time. To do that, we need to change the password policy.

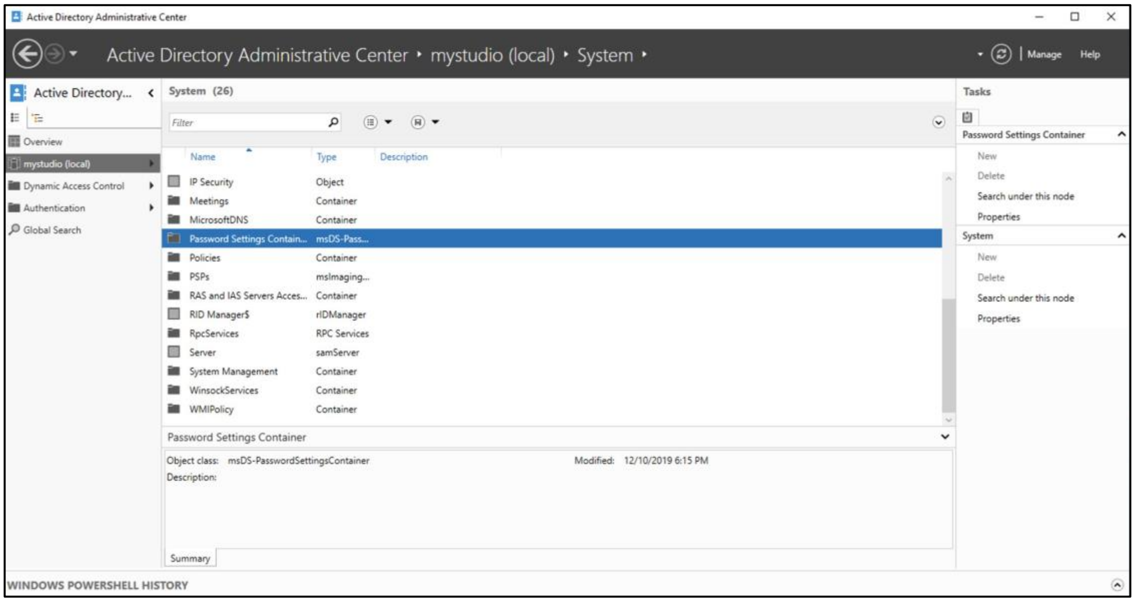
1. Open the **Start Menu**, type **Active Directory Administrative Center** and then select it from the list
2. In the **Active Directory Administrative Center window**, select your Active Directory's NetBios name on the left (e.g. mystudio)



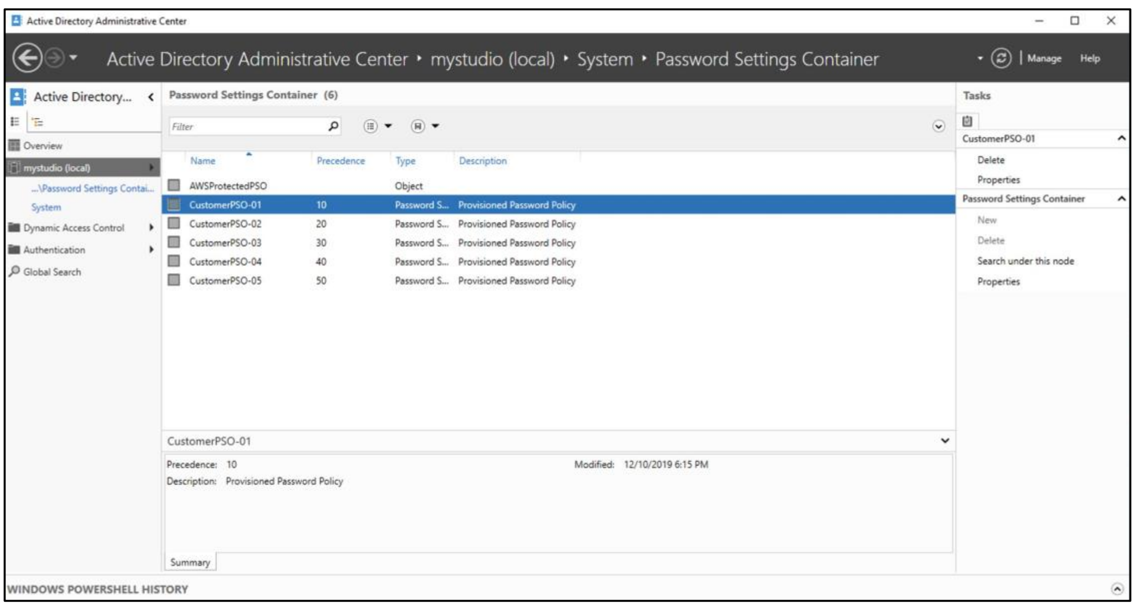
3. In the list that appears, scroll down and double-click **System.**



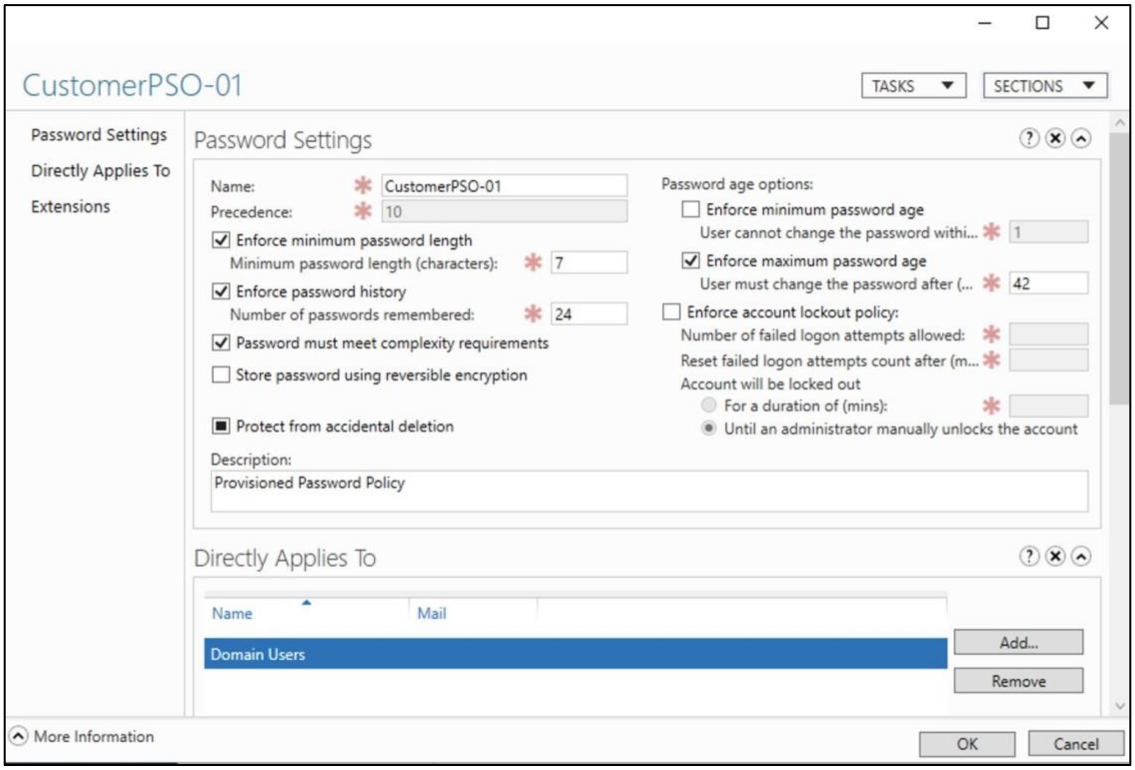
4. Double-click **Password Settings Container**.



5. Double-click **CustomerPSO-01**.



6. On the right, under **Password age options**, clear the **Enforce minimum password age** check box.
7. At the bottom, under **Directly Applies To**, click **Add.**
8. Under **Enter the object names to select**, enter **Domain Users** and click **Check Names**, then click **OK**.



9. Click **OK** again and then close the Active Directory Administrative Center window

Enable Users to Log In

1. Open the **Start menu** and search for **powershell**
2. You may see many different types of PowerShell listed. Right-click **Windows PowerShell** and choose **Run as administrator** (if a prompt comes up click **Yes**).
3. Enter the following command to allow for remote login:

```
Add-LocalGroupMember -Group "Remote Desktop Users" -Member "Domain Users"
```

Reboot

In order for all the work you've just done to take effect, you'll need to reboot your User Management instance: choose **Start Menu** → **Power** → **Restart**.

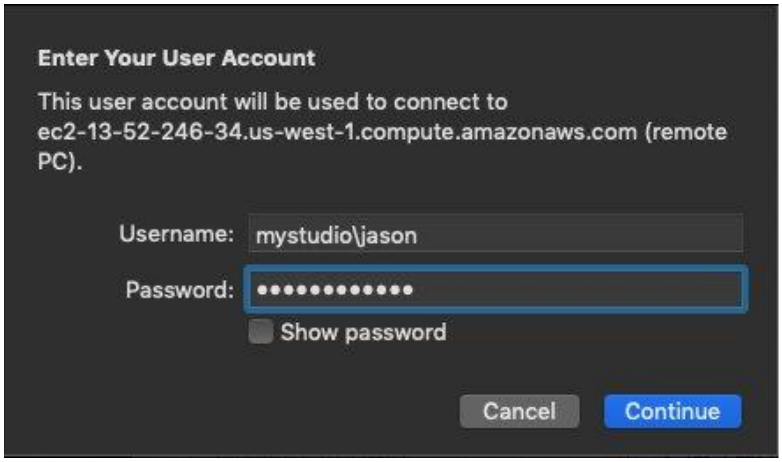
Exercise: Try Logging In as a User

Now that you've done all this work adding your users and setting up profiles, all your users should be able to login to your instance using their own usernames and passwords, instead of the Admin username and password that you used when you launched your User Management instance in the last tutorial.

Be aware that when you are logged in as a user, you will no longer have administrator permissions to alter system settings and make other changes. However, this is exactly what you want for your users, so they don't inadvertently break your setup. At the beginning of the next tutorial, we'll remind you to log back in as an administrator so you can continue building your studio in the cloud.

Login as a User

1. Log back into your User Management instance by reconnecting with Remote Desktop
   a. Go to **Services→EC2**.
   b. Click the **Instances (running)** link near the top of the page.
   c. Select your **User Management** instance.
   d. Click **Connect**.
   e. Choose the **RDP client** tab, then click **Download the remote desktop file** (you don't need to get the password).
   f. Open **Remote Desktop**.
2. This time, instead of logging in as <Active Directory NetBios name>\Admin (e.g., mystudio\Admin), you'll login as <Active Directory NetBios name>\<new user> (e.g., mystudio\jason) with the password you assigned.



If all goes well, you'll successfully be logged in as that user

Note: Sometimes it takes a while for things to propagate. If you have any errors, wait a few moments and try again.

After confirming that you can login as a user, you can close your Remote Desktop session.

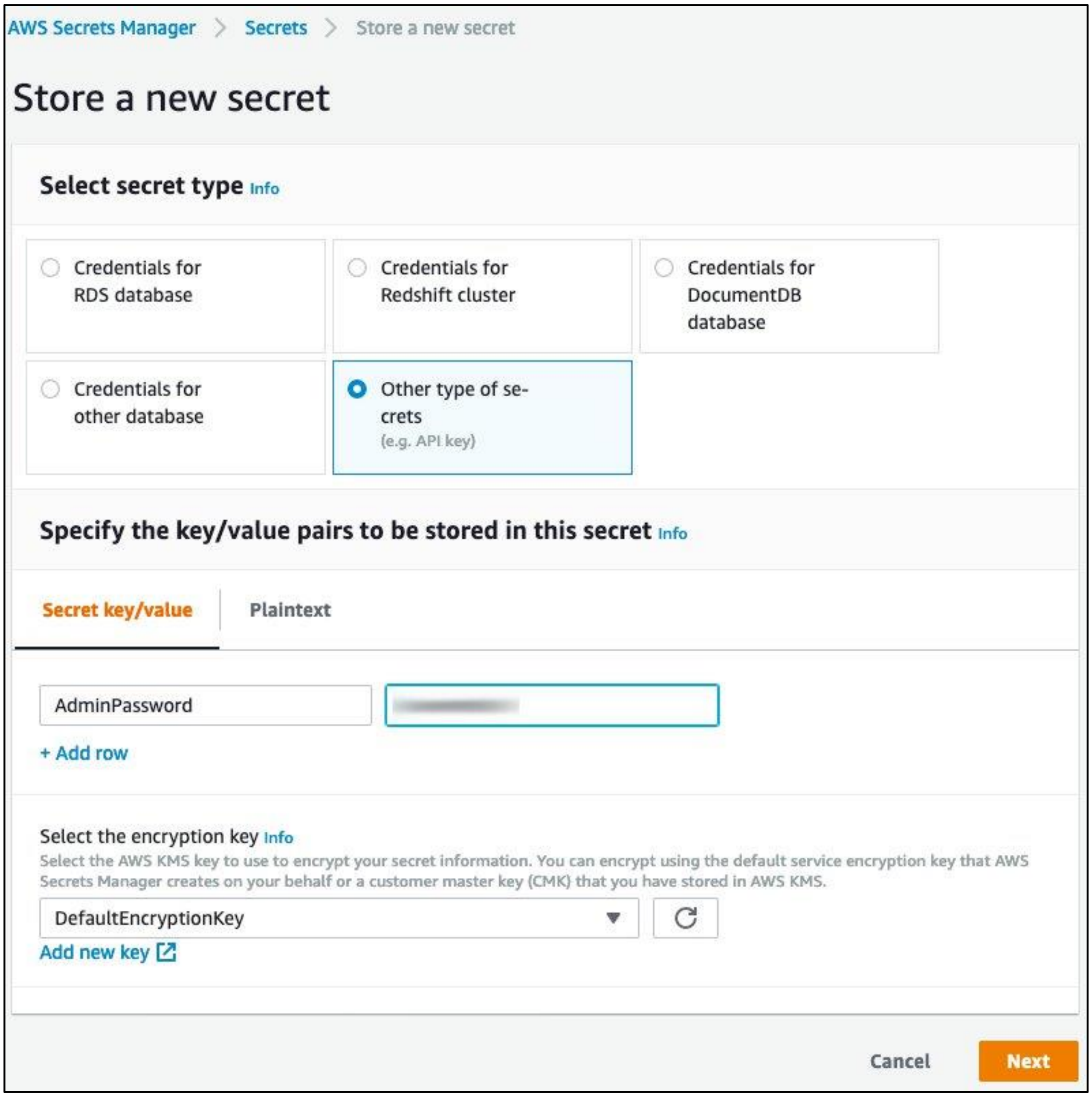
Enable Users to Login to New Instances

The key to enabling users to login to the instance as themselves is the "**Add-LocalGroupMember**" command that we ran right before doing the last exercise. In order to run that command we have to be connected to the Active Directory, and run that command as Admin. Ideally we can create a process that will automatically set up the instances for our artists right when they're launched. We will do this by setting up a **Launch Template** where we can automatically execute a series of commands as Admin every time an instance is created.

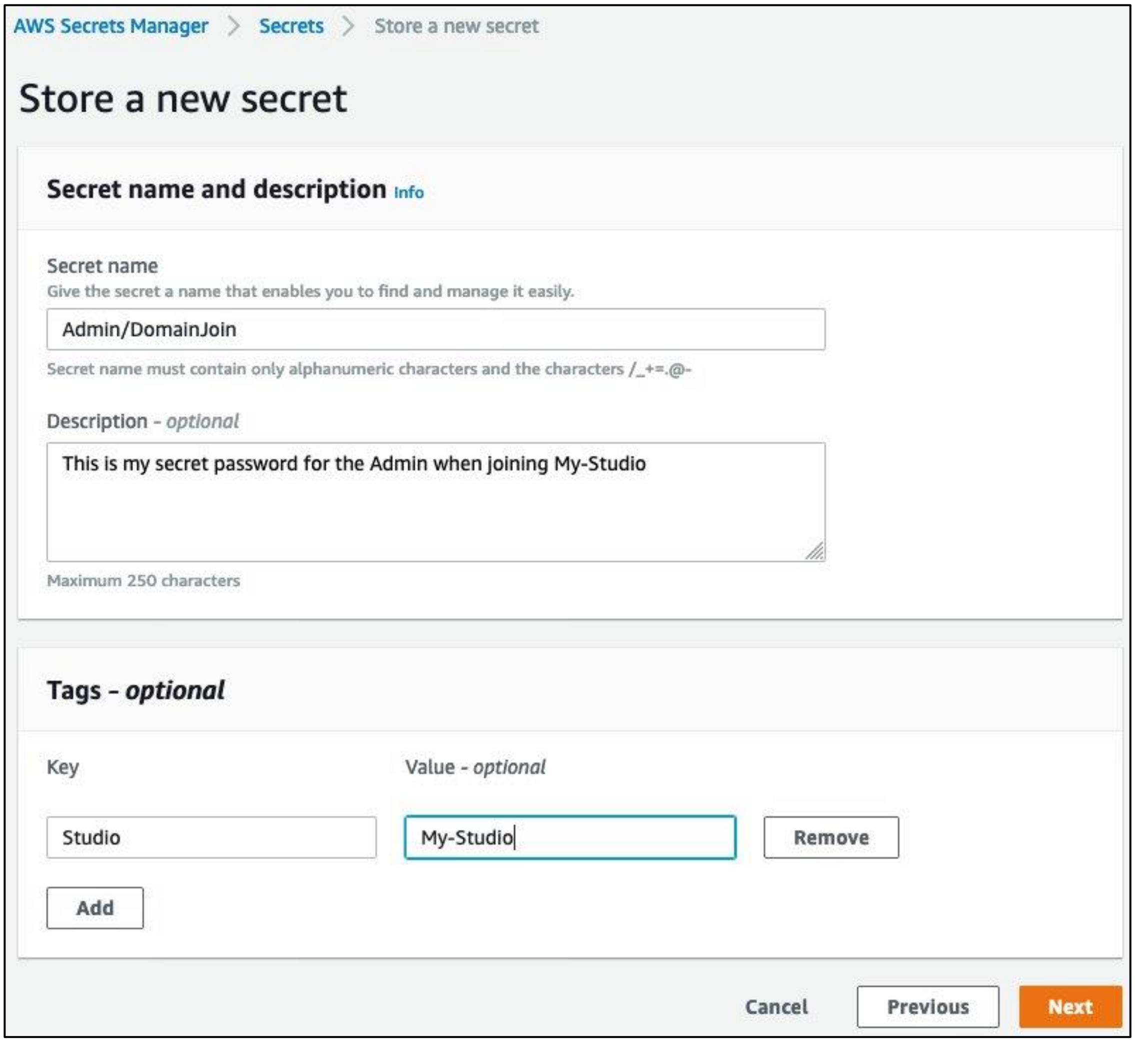
In order to securely set this up, we're going to use AWS Secrets Manager. Secrets Manager is an AWS service that allows you to securely store and retrieve sensitive information. In this case, we'll be storing the Admin password so that new instances can retrieve it later to join the Active Directory and run the "Add-LocalGroupMember" command on their own. The **Add-LocalGroupMember** command adds users or groups to a local security group. All the rights and permissions that are assigned to a group are also assigned to all members of that group.

Set Up Secrets Manager

1. In the Console, go to **Services → Security, Identity, & Compliance→Secrets Manager**.
2. Click **Store a new secret**.
3. Under **Select secret type**, select **Other type of secrets**.
4. Under **Specify the key/value pairs to be stored in this secret** enter a key/value pair for your Active Directory's Admin password.
   a. In the first field, enter a key name of **AdminPassword**
   b. In the field to the right of that, enter the Administrator password that you chose when creating your Active Directory in the last tutorial.
   c. At the bottom of the page, under **Select the encryption key**, check that **DefaultEncryptionKey** is selected.



5. Click **Next**.
6. Under **Secret name**, enter **Admin/DomainJoin**.
7. Enter an optional description.
8. Under **Tags - optional**, enter Key: **Studio** and Value: **<name of your studio>** (e.g., My-Studio).



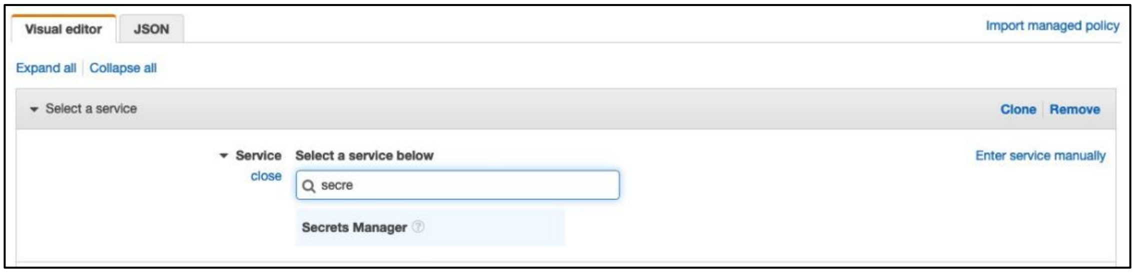
9. Click **Next**.
10. Under **Configure automatic rotation**, check that **Disable automatic rotation** is selected.
11. Click **Next**.
12. Review the information for your secret and if all looks well, click **Store**.

Create a Custom IAM Policy

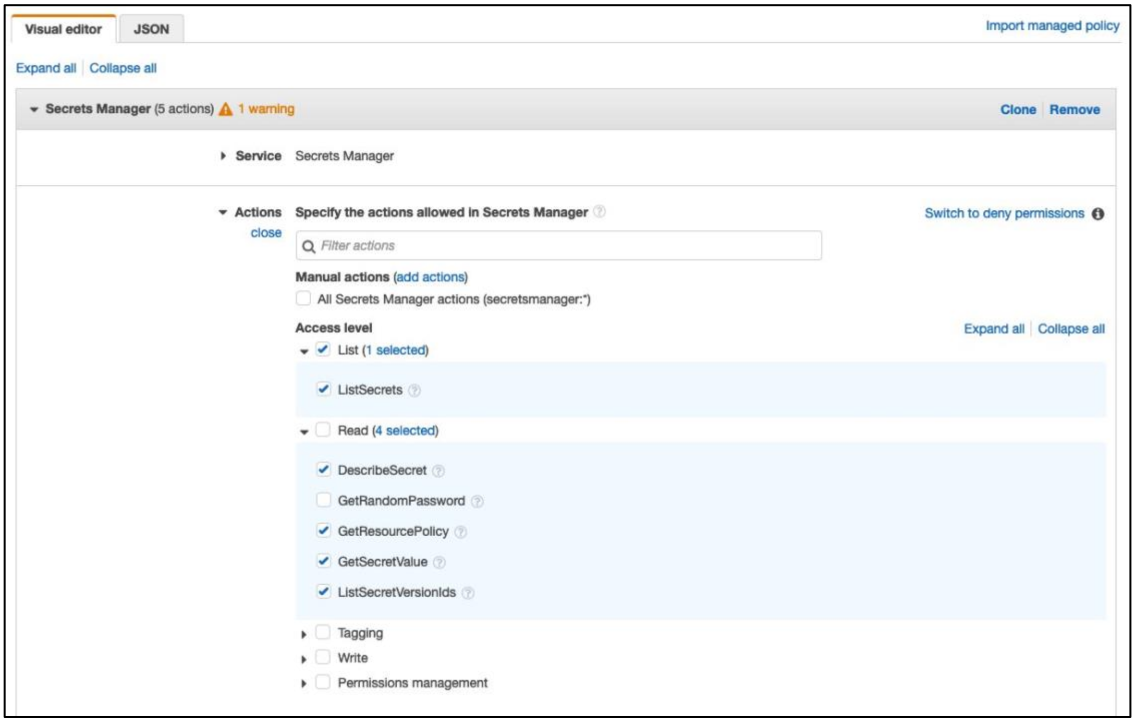
In order to allow your instances to retrieve your stored secret, we need to create a custom IAM policy that allows read, but not write permissions. We don't want anyone accidentally changing the value of the Admin password!

1. Go to **Services→Security, Identity, & Compliance→IAM**.

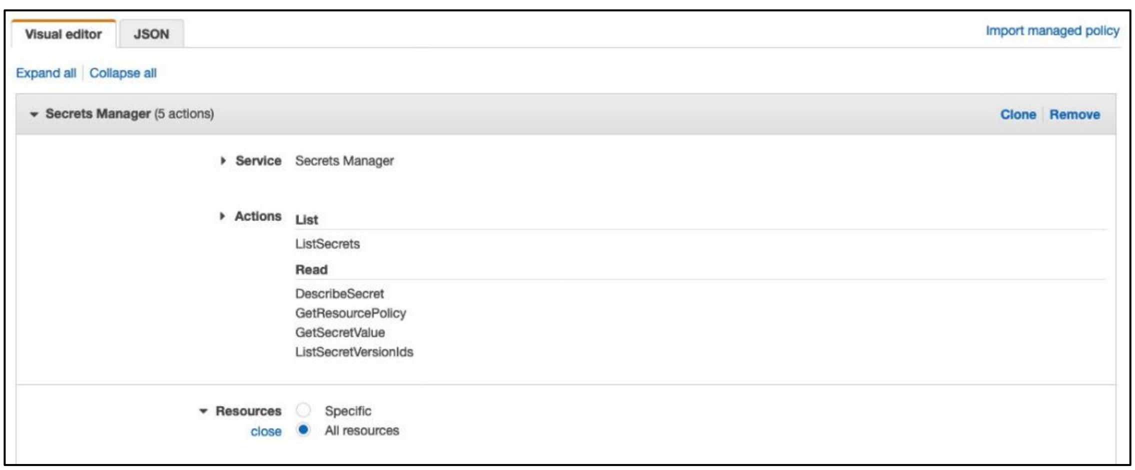
2. In the navigation panel on the left, select **Policies.**
3. Click **Create policy**.
4. On the **Visual editor** tab, click **Choose a service**.
5. Choose **Secrets Manager** from the list of services.



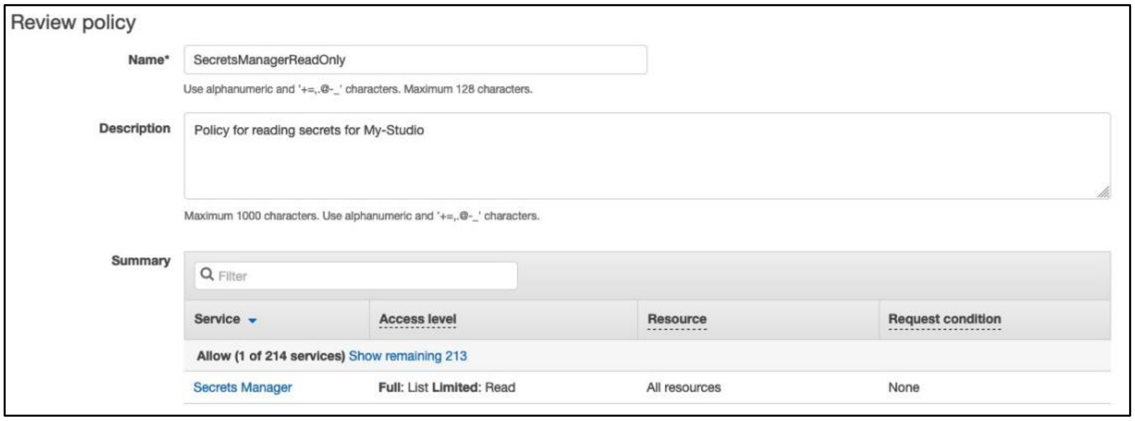
6. In the **Actions** section, expand **List** and click the check box next to **ListSecrets**.
7. Expand **Read** and click the check boxes next to the following actions:
   - **DescribeSecret**
   - **GetResourcePolicy**
   - **GetSecretValue**
   - **ListSecretVersionIds**



8. Expand **Resources** and select **All resources**



9. Click **Next: Tags**.
10. Click **Next: Review**
11. For **Name** enter **SecretsManagerReadOnly**.
12. Enter an optional description for your policy.



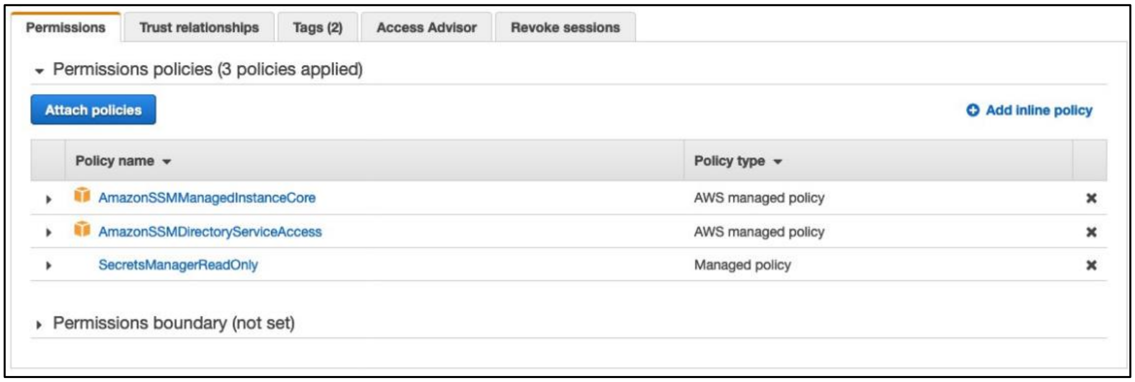
13. Click **Create policy**.

Add Custom Policy to Your IAM Role

Next, we'll add the new policy to the IAM Role that your created in the last tutorial.

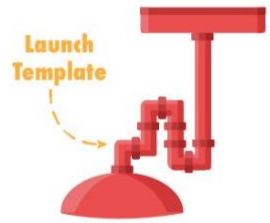
1. In the navigation panel on the left, select **Roles.**
2. Find your **EC2DomainJoin** role in the list and click its name.

3. On the summary page, click **Attach policies**.
4. Type **SecretsManager** into the search field and then select to **SecretsManagerReadOnly.**
5. Click **Attach policy**.



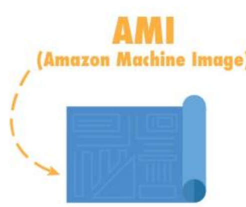
Launch Templates

Now that we've done all this work to create accounts for your users and enable them to login, let's actually get to the part where we launch a new instance and have it automatically setup to allow user logins. We could launch a fresh instance and manually select all the correct settings like when we did for your User Management instance or we could use a **Launch Template**.



A launch template contains all the settings needed to launch an instance. That includes the VPC, subnet, IAM role, security group, and storage settings that we've helped you configure every time you've launched an instance. The launch template also includes the [Amazon Machine Image (AMI)] and instance type.

An AMI contains the operating system and other software for your instance. So when you selected *Microsoft Windows Server 2019 Base* when launching your User Management instance in the last tutorial, you were actually choosing an AMI that contained Windows Server 2019 as well as other software and configuration information.



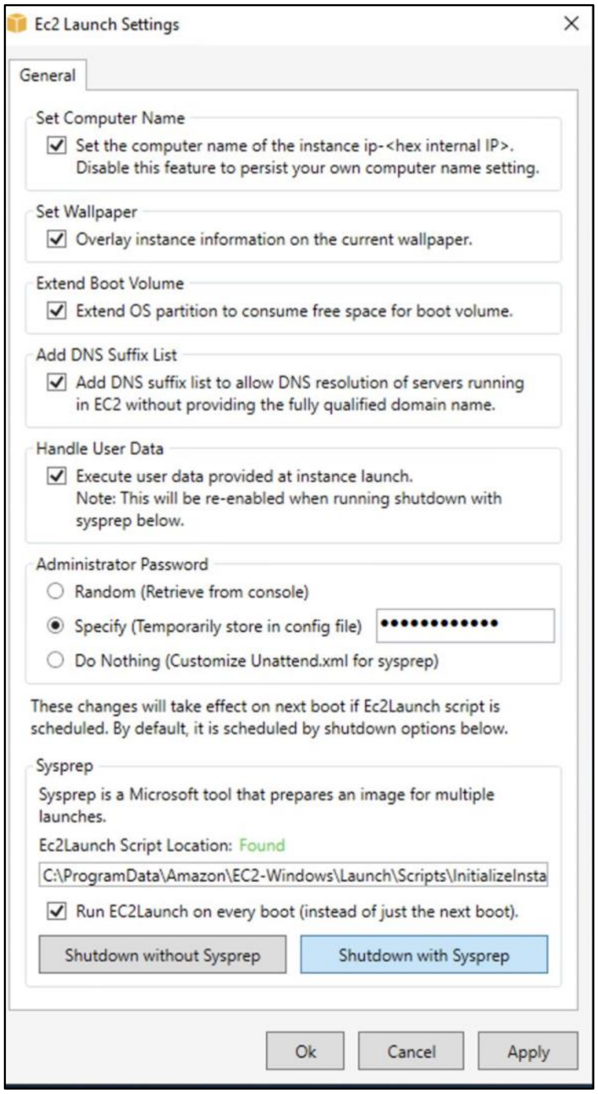
An instance type just refers to the specific combination of CPU, memory, storage, etc. that your virtual workstation is running on. For example, your User Management instance is running on an m5.xlarge, a general purpose instance. But there are many other instance types that are optimized for different uses. For example, there are GPU instances that you'll want to use as artist workstations. You can use the same AMI to launch either a general purpose instance or a GPU instance.

Creating a launch template allows you to package up a particular hardware and software combination so that you don't have remember all the individual settings every time. In this case, we're going to show how you can create a launch template from your User Management instance and then immediately launch a new instance that any user can log into, without having to repeat the setup we did in this tutorial.

Shutdown with Sysprep

1. Connect to your User Management instance again and login as Administrator. Note: In this case we do not want you to use your Active Directory Admin login, but instead we want you to login as Administrator, where you must click the Get Password and select your key pair file to get the Administrator password.
2. Go to the start menu, type **Ec2LaunchSettings** and launch it.
3. Select **Set Computer Name**.
4. Change Administrator Password to **Specify** and input the Administrator Password for your Active Directory (e.g. password for mystudio\Admin). *Your Active Directory Admin password is located under the Tutorial 2 section on your cheat sheet.*
5. Select **Run EC2Launch on every boot**.

6. Click **Shutdown with Sysprep**.

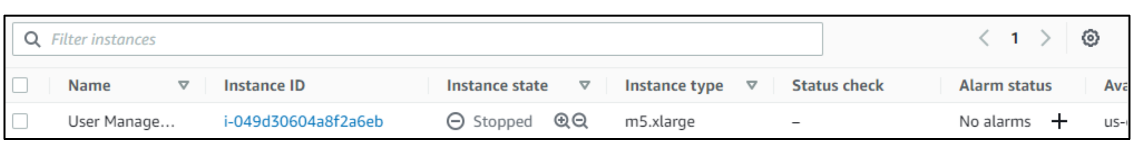


7. Click **Yes.**

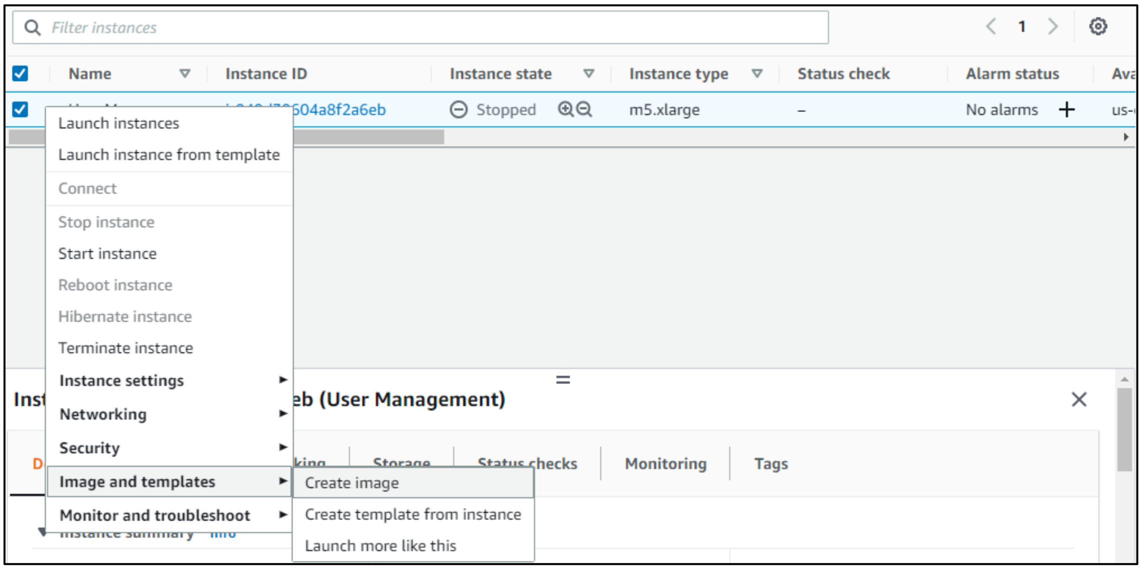
Wait for the instance to shut down, this will take a few minutes. Your Remote Desktop session will be disconnected, but that is OK.

Create an AMI

1. After disconnecting from your Remote Desktop session, go to **Services→EC2.**
2. Click **Instances**.

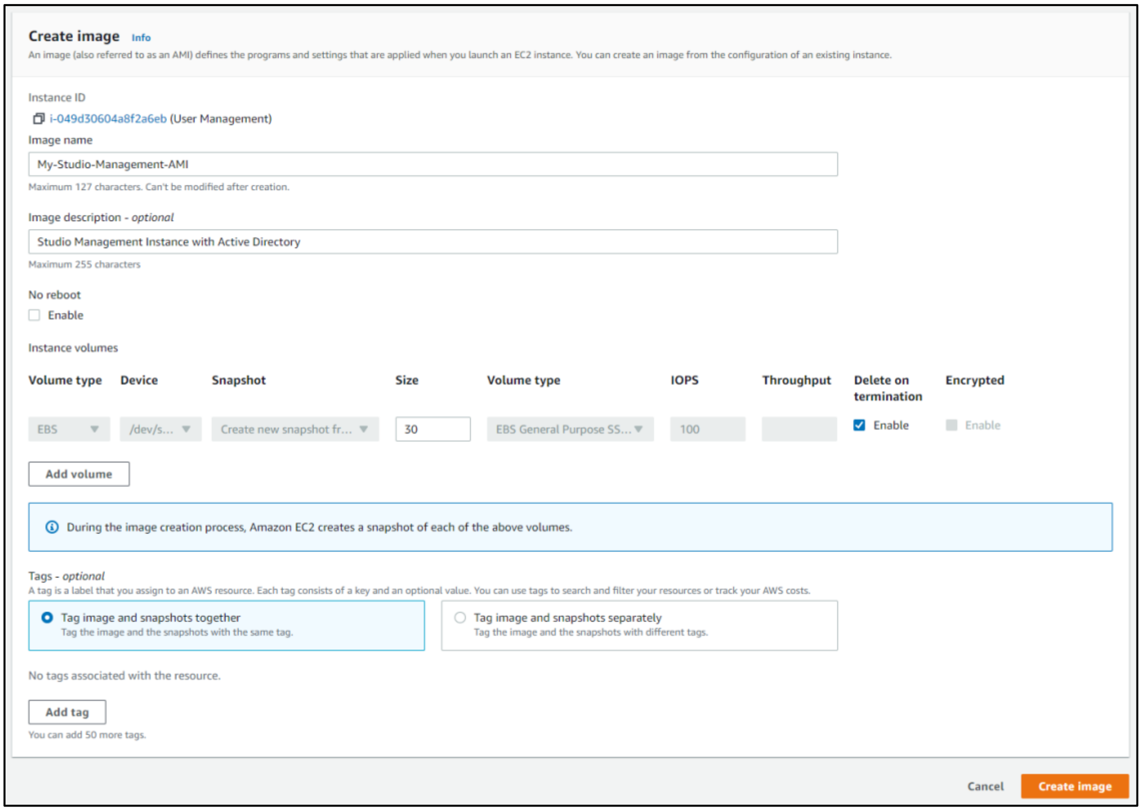


3. First, confirm that the instance state for your **User Management** is listed as stopped, then right-click the instance and select **Image and templates→Create image.**



4. Enter a name for your AMI (e.g., My-Studio-Management-AMI). *Enter the AMI name on your cheat sheet.*

5. Enter an optional description for your AMI.



6. Click **Create Image**.

7. In the panel on the left, under **Images,** click **AMIs**.

In may take a few minutes for the status of the image to change to **available.** Once that has happened, you can move on to the next step.

In the meantime, you may want to add some **Tags** to the image, specifically a Studio tag and Name tag.

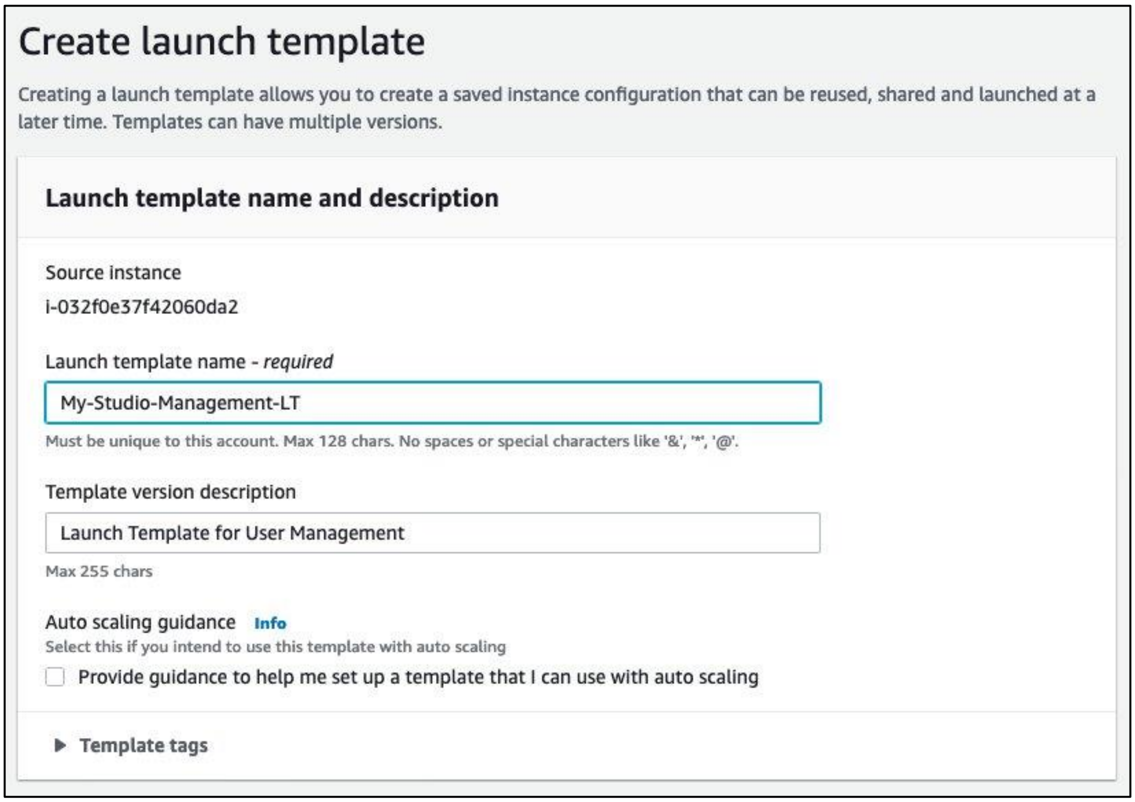
8. Click the **Tags** tab.

9. Add the following Tags:

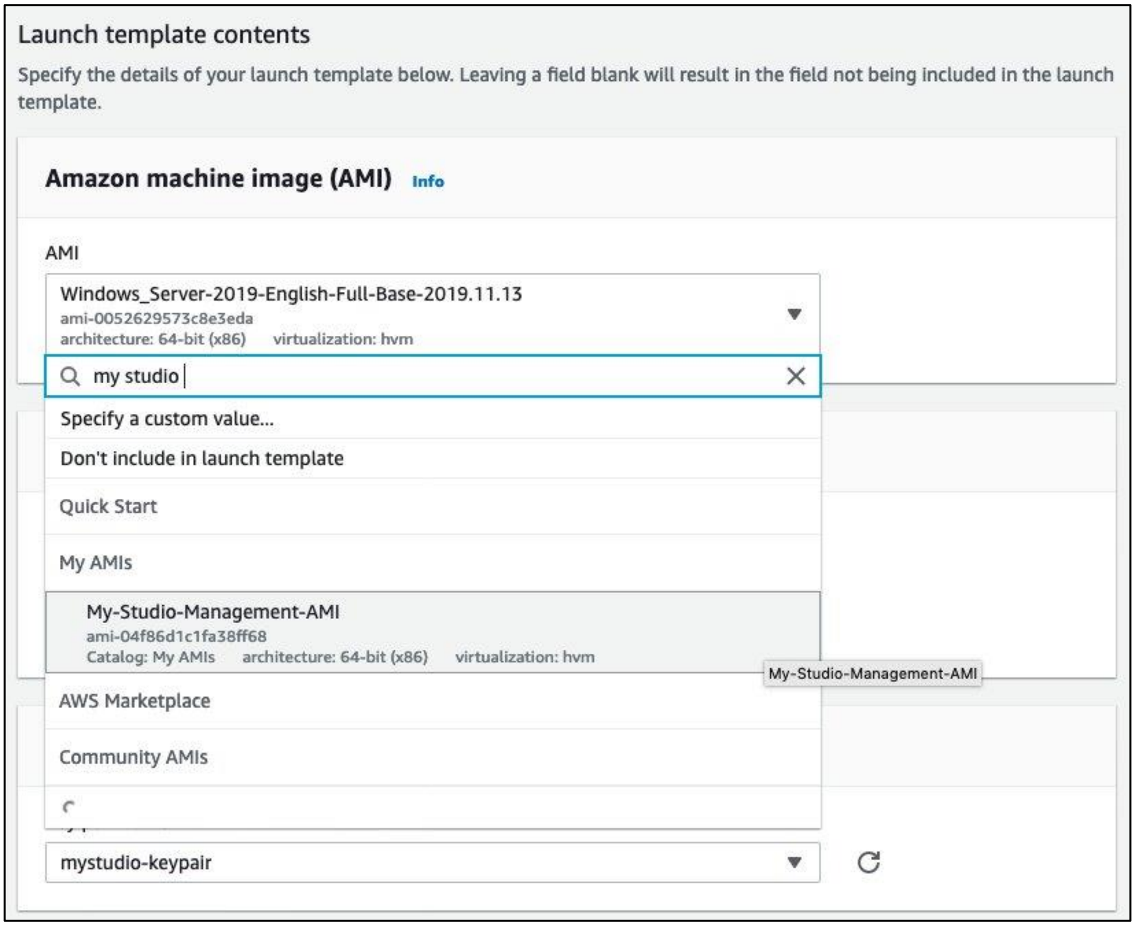
   - **Studio:** My-Studio
   - **Name**: My Studio Management AMI

Create a Launch Template

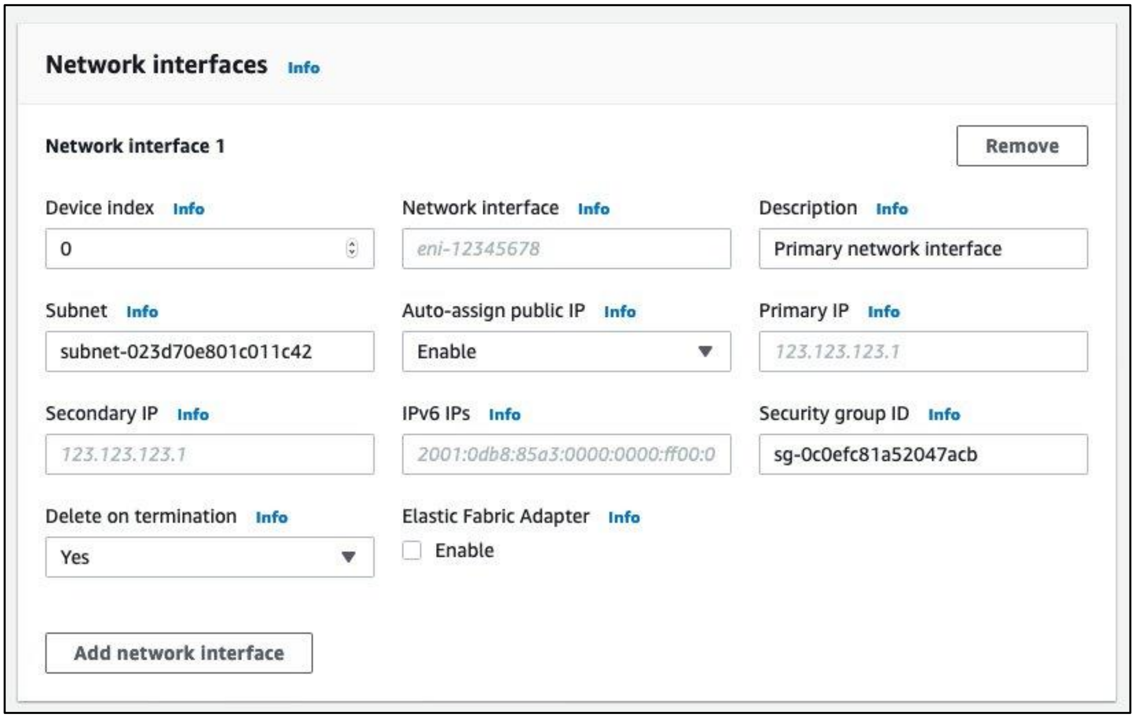
1. In the panel on the left, click **Instances.**
2. Right-click the **User Management** instance.
3. Choose **Image and templates→Create template from instance.**
4. Name your Launch Template (e.g., My-Studio-Management-LT). *Also enter this template name on your cheat sheet at this time*
5. Give it a description of your choice



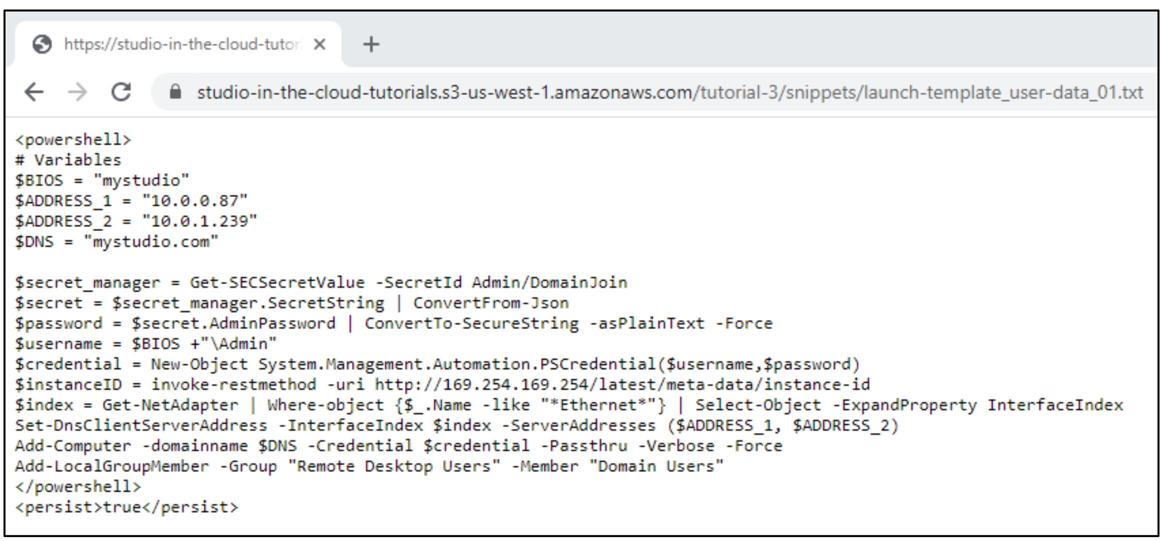
6. Under **AMI**, click the triangle to the right of the currently listed AMI and then scroll down to the My AMIs section to select the AMI you created above (e.g. My-Studio-Management-AMI). You can search for the name of the AMI in the search field. *If you need it, you can find the name of your Management AMI on the cheat sheet*



7. If you get a pop-up that says some of your current settings will be changed, click **Confirm Changes.**
8. Under **Network interfaces** check that **Auto-assign public IP** is set to **Enable.**



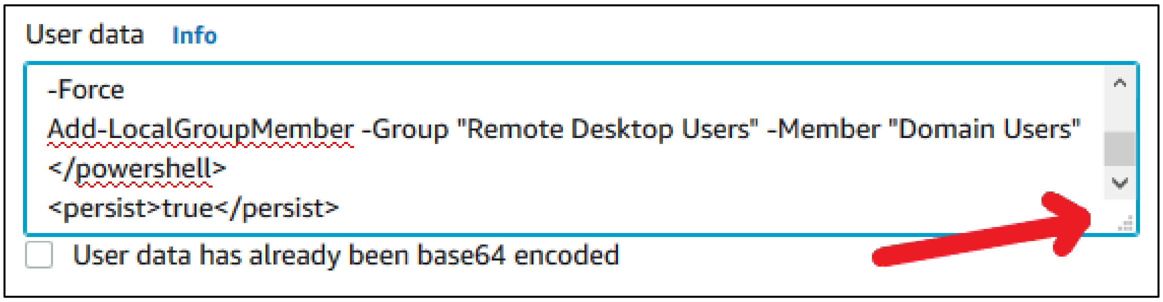
9. Scroll down to the bottom of the page and click **Advanced Details** to expand it.
10. Next, scroll down again until you get to the **User data** entry field
    - <shift>+click the image below to open a new browser tab with the text that needs to be entered into the User data entry field:




```
<powershell>
# Variables
$BIOS = "mystudio"
$ADDRESS_1 = "10.0.0.87"
$ADDRESS_2 = "10.0.1.239"
$DNS = "mystudio.com"

$secret_manager = Get-SECSecretValue -SecretId Admin/DomainJoin
$secret = $secret_manager.SecretString | ConvertFrom-Json
$password = $secret.AdminPassword | ConvertTo-SecureString -asPlainText -Force
$username = $BIOS +"\Admin"
$credential = New-Object System.Management.Automation.PSCredential($username,$password)
$instanceID = invoke-restmethod -uri http://169.254.169.254/latest/meta-data/instance-id
$index = Get-NetAdapter | Where-object {$_.Name -like "*Ethernet*"} | Select-Object -ExpandProperty InterfaceIndex
Set-DnsClientServerAddress -InterfaceIndex $index -ServerAddresses ($ADDRESS_1, $ADDRESS_2)
Add-Computer -domainname $DNS -Credential $credential -Passthru -Verbose -Force
Add-LocalGroupMember -Group "Remote Desktop Users" -Member "Domain Users"
</powershell>
<persist>true</persist>
```

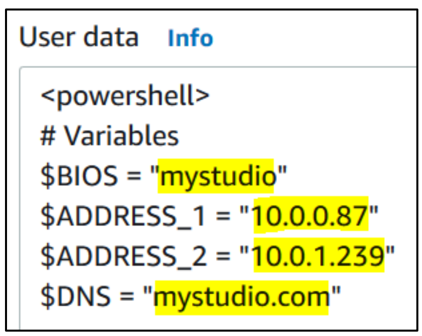
    - **launch-template_user-data_01.txt –** <shift>+click on the image above to open the text file in a new tab

11. Cut and paste the text from the browser tab into the **User data** entry field
12. Expand the size of the **User data** entry field to make it easier to read by clicking and dragging on the bottom right corner:



The code above will be run every time an instance is launched with the template. In this case, it will automatically connect to your Active Directory and run the "Add-LocalGroup-Member" command so that the users you added can login.

IMPORTANT: You will need to update the items highlighted below in yellow with specific information for your studio:



- Update the value in quotes to the right of **$BIOS** with the **Directory NetBios name** (e.g., mystudio). *Refer to your cheat sheet as needed.*
- Update the value in quotes to the right of **$ADDRESS_1** and **$ADDRESS_2** with the **DNS addresses** of your Active Directory. *You should have entered the two DNS addresses on your cheat sheet.*
- Update the value in quotes to the right of **$DNS** with the **Directory DNS name** (e.g., mystudio.com). *Also found on your cheat sheet.*

Your user data should look something like this when you're all done:

User data Info

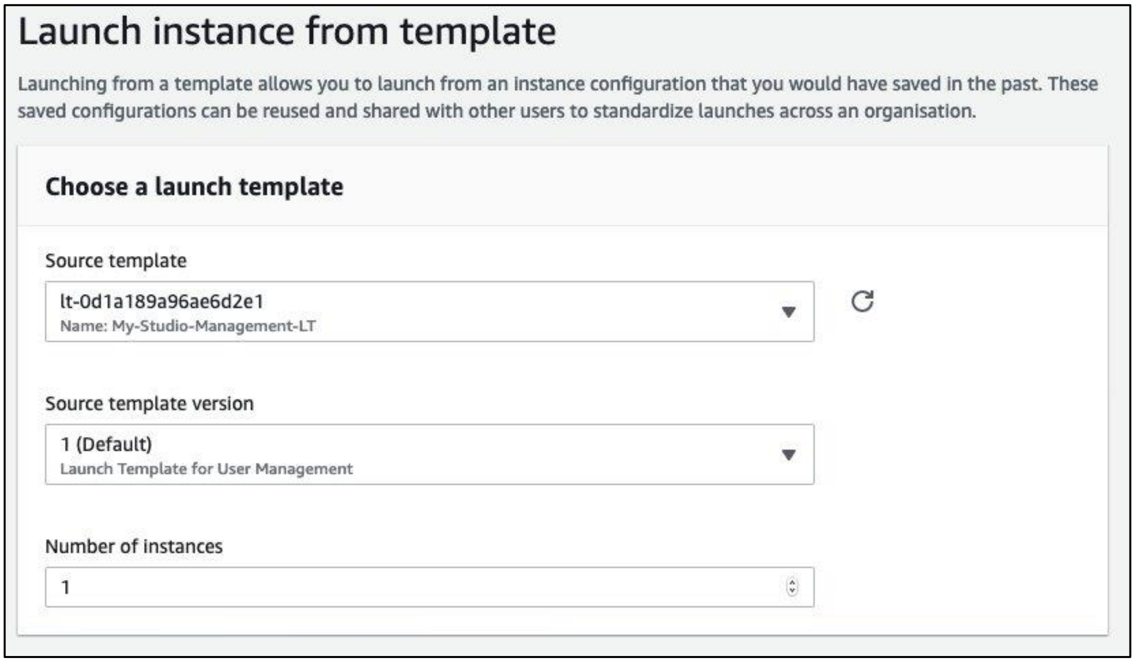
```
<powershell>
# Variables
$BIOS = "mystudio"
$ADDRESS_1 = "10.0.0.87"
$ADDRESS_2 = "10.0.1.239"
$DNS = "mystudio.com"

$secret_manager = Get-SECSecretValue -SecretId Admin/DomainJoin
$secret = $secret_manager.SecretString | ConvertFrom-Json
$password = $secret.AdminPassword | ConvertTo-SecureString -asPlainText -Force
$username = $BIOS +"\Admin"
$credential = New-Object System.Management.Automation.PSCredential($username,$password)
$instanceID = invoke-restmethod -uri http://169.254.169.254/latest/meta-data/instance-id
$index = Get-NetAdapter | Where-object {$_.Name -like "*Ethernet*"} | Select-Object -ExpandProperty InterfaceIndex
Set-DnsClientServerAddress -InterfaceIndex $index -ServerAddresses ($ADDRESS_1, $ADDRESS_2)
Add-Computer -domainname $DNS -Credential $credential -Passthru -Verbose -Force
Add-LocalGroupMember -Group "Remote Desktop Users" -Member "Domain Users"
</powershell>
<persist>true</persist>
```

☐ User data has already been base64 encoded

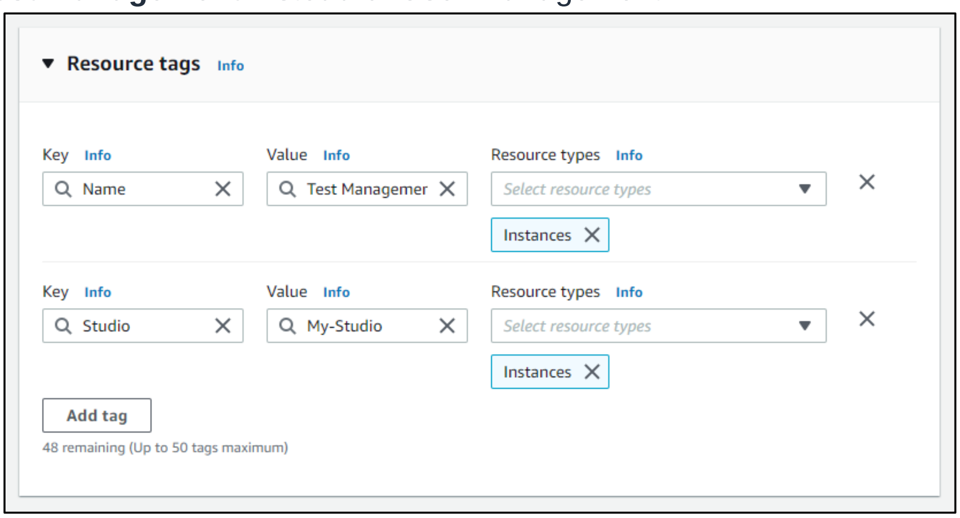
13. Click **Create launch template**.
14. Go to **Services→EC2**.
15. Under **Instances**, click **Launch Templates.**

Launch an Instance with Your Template

1. Select your Launch Template and choose **Actions → Launch instance from template.**
2. Choose the newest Source template version (This should be version 1 since you just created the template).



3. Scroll down to **Resource tags** and change the **Name** tag to **Test Management** instead of User Management.



4. Scroll down to the bottom and launch your instance by clicking **Launch instance from template.**
5. Go to **Services→EC2** and click **Instances (running).**
6. Once the instance has initialized and has passed 2/2 checks, wait an extra minute or two to allow this new instance to join Active Directory.

7. Then try logging in with an Active Directory username like you did above: <Active Directory NetBios name>\<new user> (e.g., mystudio\jason).

   Note: Sometimes it can take a little while for Active Directory to connect. If you get an error when trying to login as a user, wait another few minutes and try again. If that doesn't resolve the issue, try rebooting the instance by going to **Actions→Instance State→Reboot**.

Great job creating your first launch template! We'll be using this template in later tutorials to launch other instances that we'll need, but for now, we only need your original User Management instance. There's no need to keep your Test Management instance running, so we should terminate it to save resources.

Terminate Your Old User Management Instance

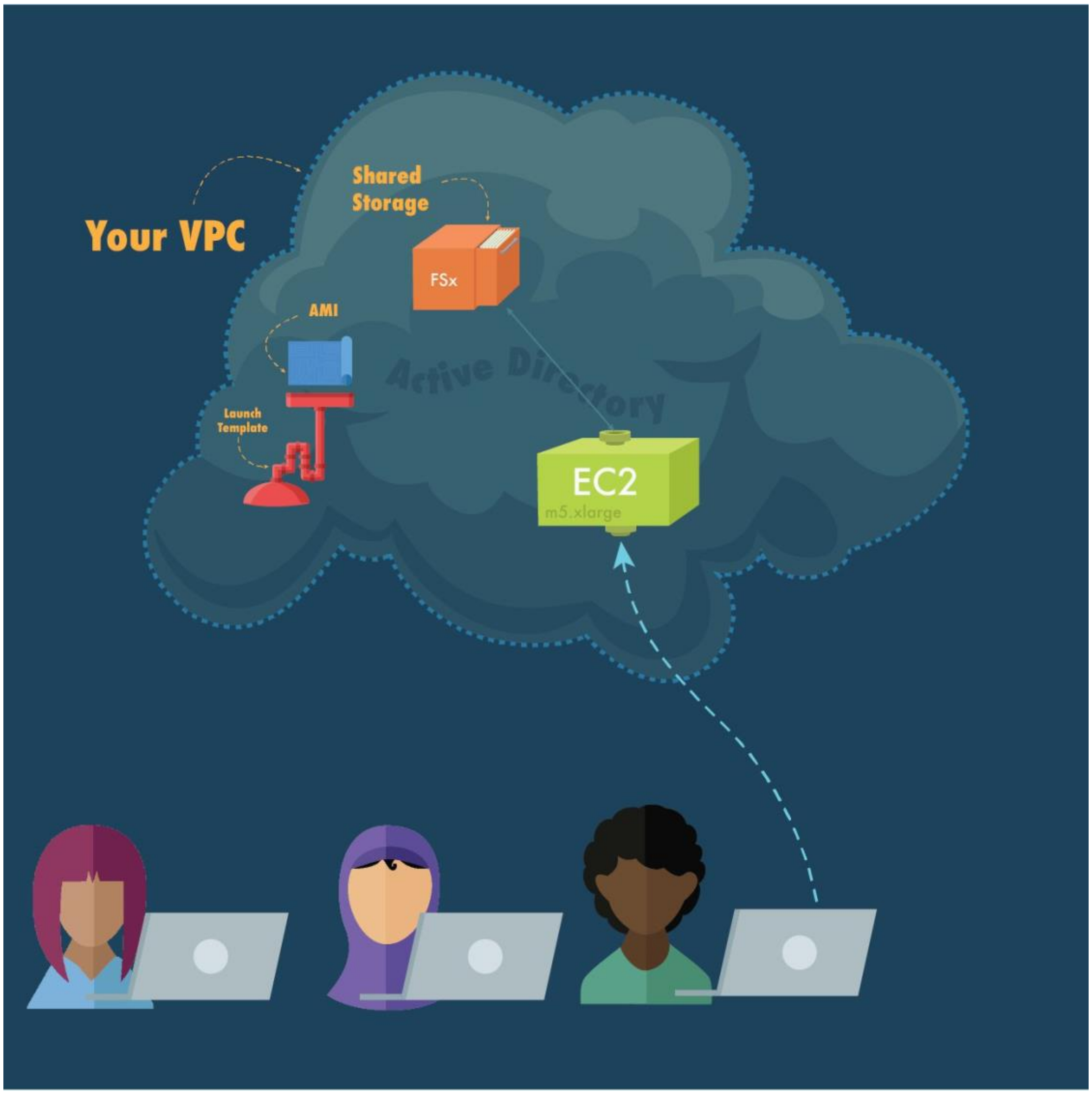
Your old User Management instance is currently stopped as a result of creating an AMI. We're going to terminate this old instance and use your new Test Management instance in its place. Why? The new instance you created is already set up with the correct user data to join Active Directory and allow your users to login. Rather than update the old User Management instance, it's easier to just switch to using this new one.

1. In the **AWS Console**, select your old **User Management** instance.
2. Then click the **Instance state** button at the top of the list of instances and select **Terminate instance**. Finally, click the **Terminate** button.
3. Now select your new **Test Management** instance.
4. Hover over the **Name** field and click the **pencil icon** that appears next to the name.
5. Change the name to **User Management.** From now on, you'll use this instance for any user management needs.

If you've already created all the users that you need for your studio, you can stop your new User Management instance. You can restart it at any time if you need to add more users later.

1. With your new **User Management** instance still selected, click the **Instance state** button at the top of the list of instances and select **Stop instance**. Then click the **Stop** button.

Your VPC Now



There have been a couple of additions to your VPC during this tutorial. We've added an FSx file system in Subnet A of your VPC and connected it up to Active Directory to store your user profiles. We've also created new users in Active Directory so that each of the artists in your studio can login to instances.

In our next tutorial, we'll start installing the applications that your artists will need to do their creative work.

Appendix

Links to AWS Documentation

- What Is Amazon FSx for Windows File Server?
- Using Microsoft Windows File Shares
- Manage Password Policies for AWS Managed Microsoft AD
- Amazon Machine Images (AMI)
- Launching an Instance from a Launch Template
- Amazon EC2 Instance Types

Downloads

- launch-template_user-data_01.txt

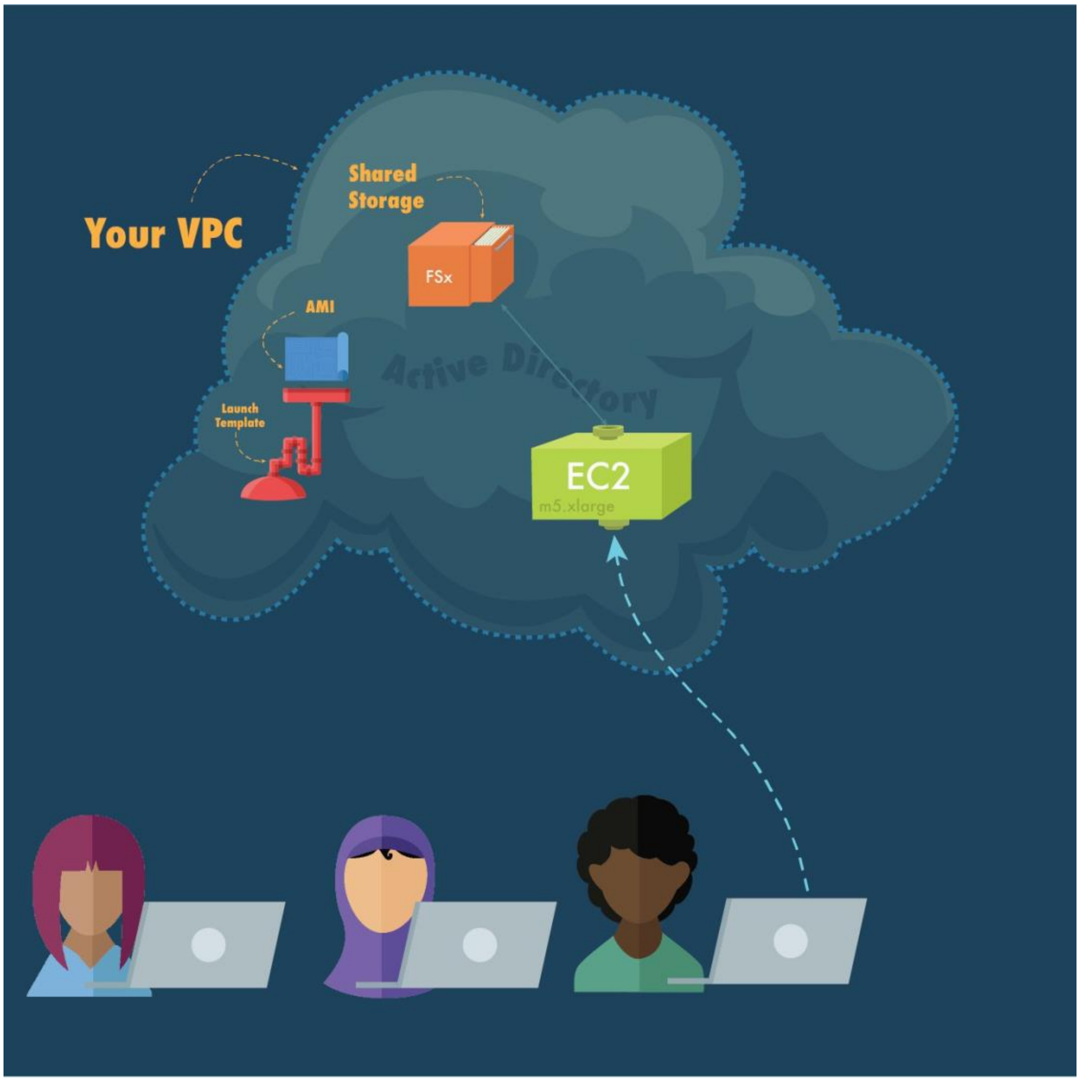
Changing Security Groups After FSX Creation

1. Go to **Services** → **Storage** → **FSx.**
2. Click the **File system name** or **File system ID** for your FSx file system.
3. Next click the **Network & security** tab and then click the link for the **Network interface** for your file system.
4. On the page that appears, click the **Actions** menu, then select **Change Security Groups.**
5. Select one or more security groups that you would like to assign to your file system.

Tutorial 4. Building a Render Scheduler with AWS Thinkbox Deadline

Estimated Time to Complete: 1 hour, 15 minutes

In this tutorial we are going to set up an instance to manage the renders that get sent to the render farm within our studio. We are also going to install AWS Thinkbox Deadline and configure path mapping.



What is AWS Thinkbox Deadline?

Deadline is a compute management toolkit that leverages the scale of the cloud to run compute intensive jobs. Deadline handles provisioning machines to render on, licensing and path mapping so that your artists can send renders off to the farm and continue working while the jobs complete.

What is the Render Scheduler?

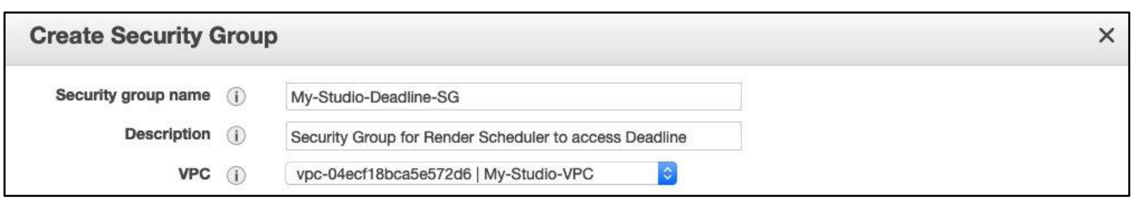
The render scheduler is an instance we use for managing Deadline. The Deadline Repository and Database will reside on this instance, and it is the instance that will provision workers for us when we submit a render.

Allow Render Scheduler to Access Deadline

In order for the render scheduler to talk to your farm workers (which we will create in a later step), we need to open some ports that will allow for Deadline access.

Create Deadline Security Group

1. Go to **Services** → **EC2.**
2. In the left panel click **Security Groups.**
3. Click **Create security group.**
4. Name the security group (e.g., My-Studio-Deadline-SG). *Also, record the name on your cheat sheet.*
5. Give it a description (e.g., Security Group for Render Scheduler to access Deadline).
6. Set the **VPC** to your studio's VPC.



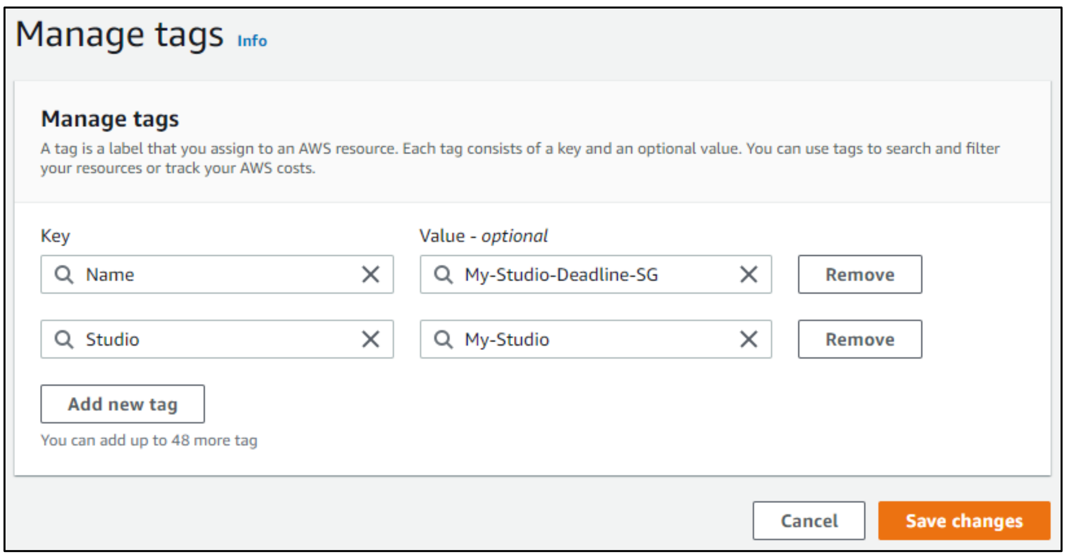
7. Click **Add rule.**
8. Add these rules to the **Inbound rules**:

	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	4433	10.0.0.0/16	Deadline
Custom TCP Rule	TCP	17000-17005	10.0.0.0/16	Deadline
Custom TCP Rule	TCP	8080	10.0.0.0/16	Deadline
Custom TCP Rule	TCP	8082	10.0.0.0/16	Deadline
Custom TCP Rule	TCP	27100	10.0.0.0/16	Deadline
Custom TCP Rule	TCP	443	10.0.0.0/16	Deadline
Custom UDP Rule	UDP	17001	10.0.0.0/16	Deadline
Custom UDP Rule	UDP	123	10.0.0.0/16	Deadline

9. Click **Create security group** when done.

   After the security group is created, you will find that it doesn't have a **Name** visible in the list of security groups. Find the one you just created by locating it under **Group Name**, and then add a **Name** by clicking on the little pencil icon under the name column.

10. You can also add a Tag for your Studio now in the Tags Tab.



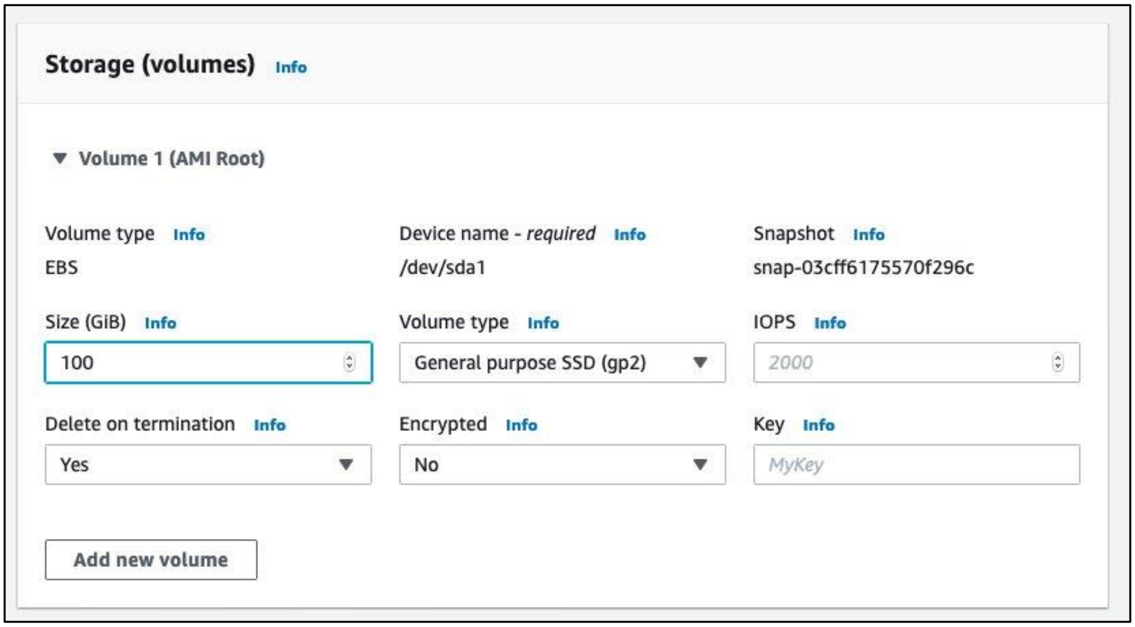
- *Now is also a good time to enter the security group's ID to your cheat sheet. You can find it to the right of the name in the list of security groups. It will look something like: sg-293m3jj3iha2991o4.*

Launch the Render Scheduler Instance

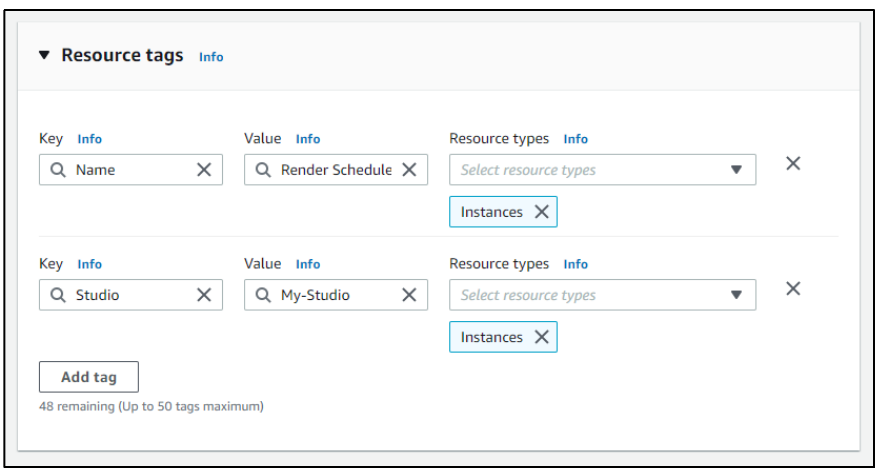
Launch the Instance Using Your Management Launch Template

1. Go to **Services** → **EC2.**
2. Select **Launch Templates** in the left side panel.
3. Select the **Management Launch Template** (e.g., My-Studio-Management-LT). *You can refer to your cheat sheet if you don't remember the launch template's name.*
4. Choose **Actions**→ **Launch instance from template.**
5. Choose the newest version.
6. Change the **Instance type** to **m5.2xlarge.**

7. Increase the storage space for this machine by going to the **Storage (volumes)** section, clicking the triangle to the left of **Volume 1** and changing **Size (GiB)** to **100.**



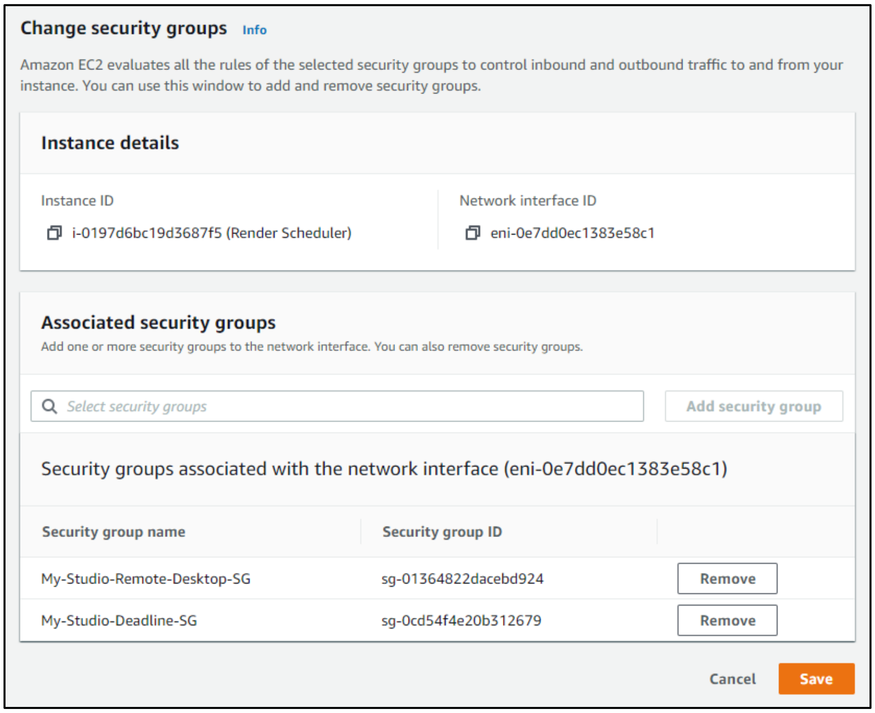
8. Change the **Name** tag to **Render Scheduler** in the Resource tags section.



- At the bottom of the page, click **Launch instance from template.**
- After the instance has launched, go to your list of running instances.

9. Right-click Render Scheduler and choose **Security** → **Change security groups.**

10. Click on the search field, select your Deadline security group (e.g., My-Studio-Deadline-SG) from the list and then click **Add security group**. *The name and ID of your Deadline security group can be found on the cheat sheet.*
11. Click **Save.**



- *Make note the Private IPv4 address of your render scheduler and enter it on your cheat sheet. If you have the render scheduler selected in the instance list, you can find the Private IP in the Details tab at the bottom of the page.*

Log into the Render Scheduler as Administrator

Now we are going to connect to the render scheduler and start installing Deadline! For this task we are going to log onto the machine as **Administrator**.

Note: This is different than logging in as mystudio\Admin as we did in Tutorial 3. In that case, we needed admin privileges for our domain when setting up users, etc. In this case, we need admin privileges for the instance itself, hence the need to login as **Administrator**, the administrator for the instance itself.

To log into the Render Scheduler as Administrator:

1. Go to **Services** → **EC2** and click **Instances (running).**
2. Look for the Render Scheduler instance in the list and verify that the Status Checks column says “2/2 checks passed” before continuing.

3. Select the Render Scheduler instance and click the **Connect** button.
4. Chose the **RDP client** tab and then click **Get password** to get the Administrator password
5. Click **Browse** and find the key pair file (e.g., mystudio-keypair.pem). *You can refer to the Tutorial 1 section of your cheat sheet if you don't remember the key pair file name.*
6. Click **Decrypt Password**, and notice this time that the password is set to your Active Directory Admin password. This is because of the special Sysprep shutdown procedure we did in Tutorial 3, where we specified the Administrator password. Now instead of having to decrypt the password, you can just type in your Admin password.
7. Click **Download remote desktop file.**
8. Open Remote Desktop and connect to your instance.
9. The username should already be set to Administrator. Type in your Active Directory Admin password and click **OK**.

Once logged in as Administrator, you can begin the installation process for your applications! But, first...

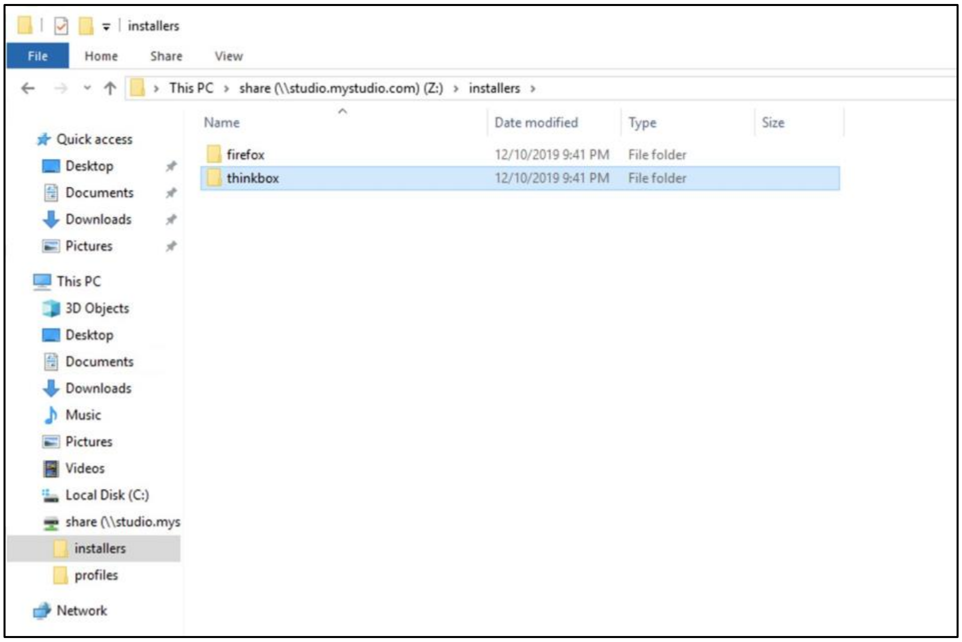
Connect Shared Storage to Render Scheduler

1. Open a **File Explorer** window.
2. From the navigation pane, right-click **Network** and choose **Map Network Drive**.
3. Choose a drive letter of your choice for **Drive** (e.g., **Z:**).
4. Enter the full CNAME Alias for your file share that you noted above for **Folder**.
    - e.g., \\studio.mystudio.com\share\
5. Confirm that **Reconnect at sign-in** is selected and then click **Finish**.

- It will ask for your Active Directory login credentials. For example:
    - Admin
    - Use the password you set up.
    - Toggle on **Remember my credentials.**

6. Create a new folder on your shared drive called **installers.**

7. Create a folder in the installers folder called **firefox** and one called **thinkbox.**

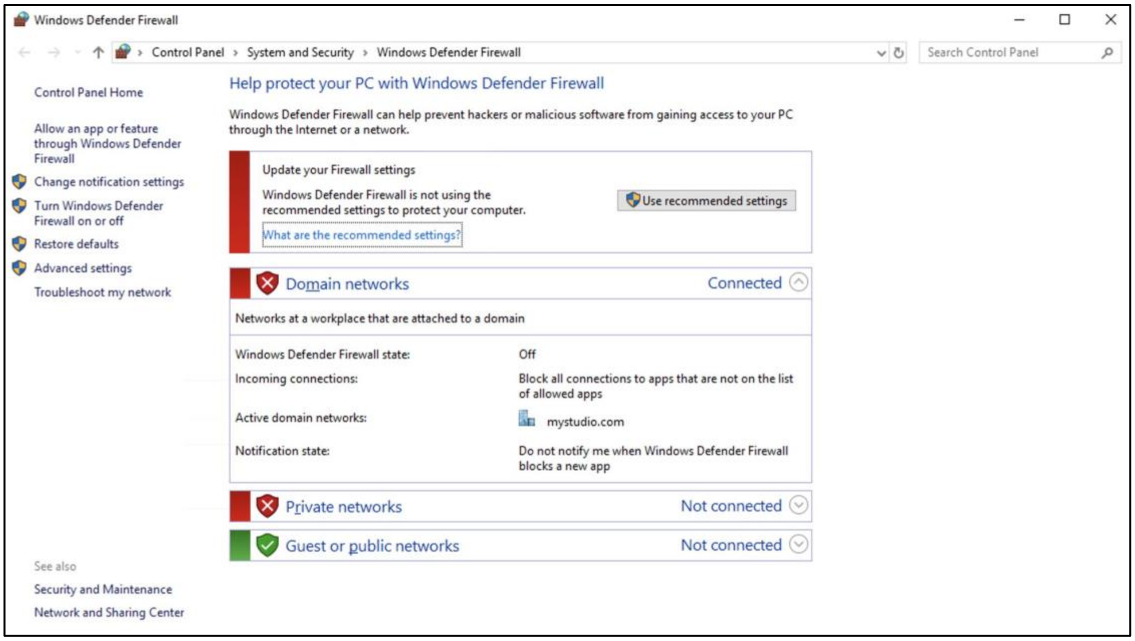


Turn Off Firewall Settings

In order for Deadline to work correctly we need to turn off some of the default Windows firewall settings. This is OK because the AWS security groups we are setting up will handle security for our instances.

1. Go to the **Start Menu** and type **firewall.**
2. Choose **Windows Defender Firewall.**
3. On the left click **Turn Windows Defender Firewall on or off.**
4. Under **Domain network settings** turn **off** Windows defender firewall.
5. Under **Private network settings** turn **off** Windows defender firewall.
6. Leave **Public network settings** firewall **on.**

7. Click **OK.**



Install Firefox

The first thing we are going to install is a web browser that will allow us to download the Deadline installers. In this case we are going to use **Firefox**.

Copy Firefox Installer onto Render Scheduler Instance

1. On your local computer navigate to the Firefox download page.
2. In the **Select your preferred installer** drop down, select **Windows 64-bit MSI.**
3. **Copy** the MSI file from your local downloads folder and **paste** it onto the render scheduler instance in **Z:\installers\firefox .**
   - This will upload the file to the remote instance.
4. Double-click the installer and click **Run** in the popup window.

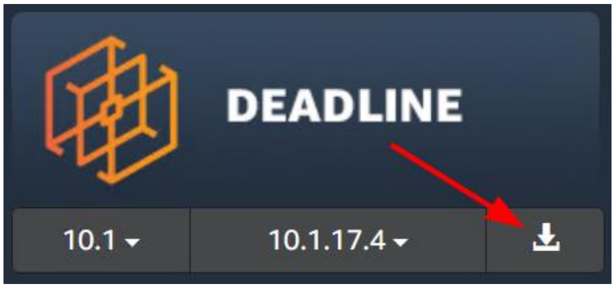
When it's done running you will be able to launch Firefox.

Install Deadline

Download the Installers and Run!

1. In **Firefox** on the remote instance, go to the AWS Thinkbox downloads page and login with your **Amazon.com** login credentials (not your AWS account info).

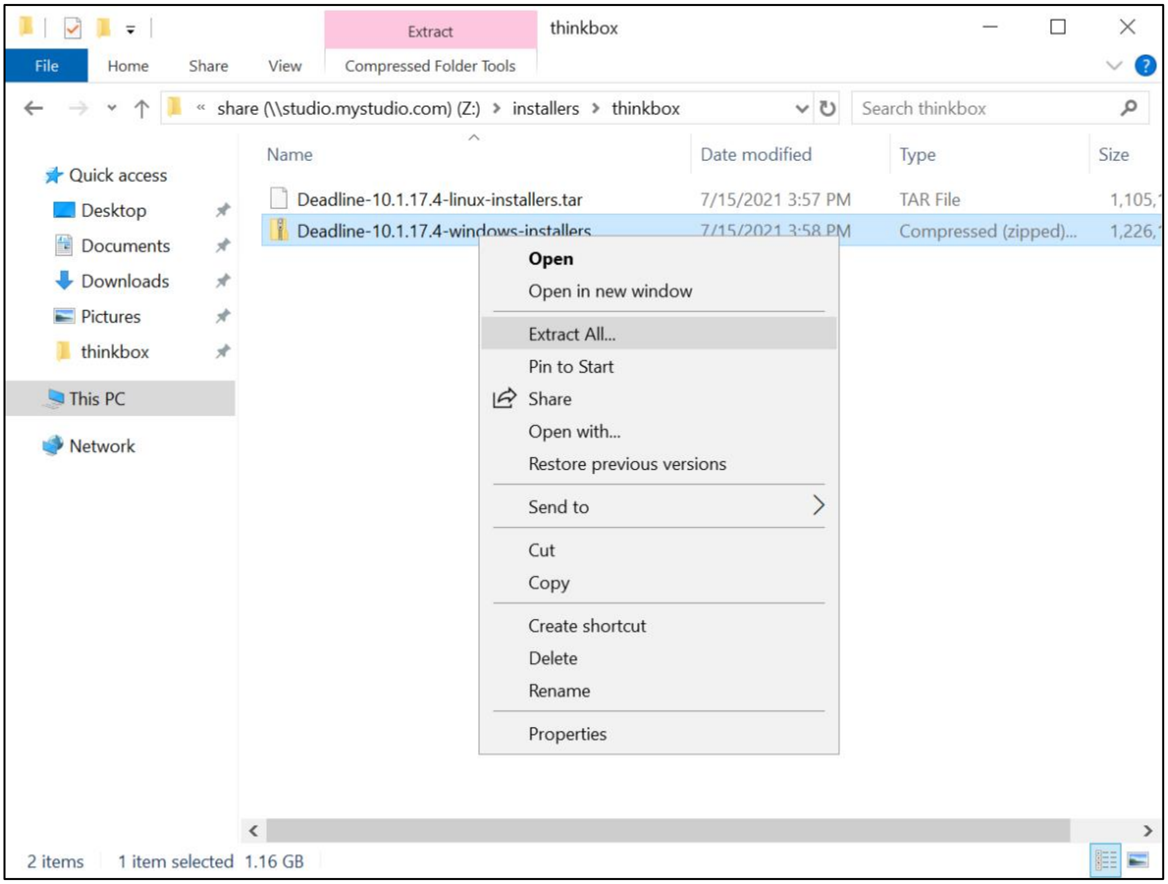
2. Click the **download icon** on Deadline



3. Download the latest version of the Deadline installers for Windows and Linux (we will need the Linux installers later). At the time of writing, this was **Deadline-10.1.17.4-windows-installers.zip** and **Deadline-10.1.17.4-linux-installers.tar**, but the version you see may be higher.

4. **Cut and paste** both installers from the Downloads folder into the **Z:\installers\thinkbox** folder.

5. Unzip the Windows file by right mouse clicking and choosing **Extract all.**



Install the Deadline Repository

1. In **Z:\installers\thinkbox\Deadline-10.1.17.4-windows-installers**, double-click **Deadline Repository** installer.
2. Click **Run.**
3. Click **Next.**
4. Accept the license agreement and click **Next.**
5. Leave the **Repository Directory** at the default (we're going to install the repository on the C:\ drive) and click **Next.**
6. Leave the database type as **MongoDB** and click **Next**.
7. Choose **Install a new MongoDB database on this machine** and click **Next.**
8. Keep option **Download Mongo DB (requires internet connection)** selected and click **Next.**

9. Choose **I accept the agreement** to accept the MongoDB license agreement and click **Next.**
10. Leave **MongoDB Directory, MongoDB Hostname** and **Mongo DB Port** at the default and click **Next.**
11. Using File Explorer, create a new folder in **C:** called **DeadlineCertificates** and choose it as the **Certificate Directory.**
12. For **Certificate Password** and **Confirm Password**, enter your Active Directory admin password. Note: For simplicity, we are re-using the Active Directory admin password here for this tutorial setup. However, if you desire, you can use a different password. *In that case, make sure to enter the Certificate Password you choose in the Notes section of the Important Information Cheat Sheet.*
13. Turn off **Use client certificate for DB user authentication** and click **Next**.
14. Leave **Enable Secrets Management** selected and click **Next**.
15. For **Admin Username** enter your **Active Directory admin username** (e.g., Admin)
16. For **Password** enter your Active Directory admin password and click **Next**. Note: Again, like we did with the Certificate Password, we are re-using the Active Directory admin password. If your Active Directory admin password does not meet the requirements for the Deadline installer, you may need to create a new password. *If so, be sure to enter this Deadline Secrets Management Password in the Notes section of the Important Information Cheat Sheet*.
17. Click **Next** then click **Next** again.
18. Wait a few minutes for the installation to complete.
19. Click **Finish.**

Install the Deadline Client

When the install for the Repository is finished, install the **Deadline Client**

1. Go to **Z:\installers\thinkbox\Deadline-10.1.17.4-windows-installers .**
2. Double-click **Deadline Client** installer.
3. Click **Run.**
4. Click **Next.**

5. **Accept** the license agreement and click **Next.**
6. Leave the **Installation Directory** as default and click **Next.**
7. Select **Remote Connection Server** and click **Next.**
8. Leave the **Repository Directory** as default and click **Next.**
9. In the field next to **Database TLS Certificate**, navigate to **C:\DeadlineCertificates\Deadline10Client.pfx** and click **Open.**
10. For **TLS Certificate Password** enter the Certificate Password you used when installing the Repository above. (This may be the same as your Active Directory admin password. *If not, you can find it in the Notes section of your cheat sheet.*) Click **Next.**
11. Leave **Assign server role and grant master key access** selected and click **Next**.
12. Enter the Deadline Secrets Management **Admin username** and **password** that you entered during the Repository installation instructions above. (Again, this may be the same as your Active Directory admin password. *If not, look in the Notes section of your cheat sheet for the Deadline Secrets Management password you chose.*)
13. Leave **Name of the current master key** at the default value of **defaultKey** and click **Next**.
14. Leave **Launch Worker When Launcher Starts** checked and click **Next.**
15. Leave **Block auto upgrade via a secure setting** selected and click **Next**.
16. Set **User/Group running the RCS** to **Everyone** (from Active Directory).
17. Leave the rest of the Remote Connection Server setup as default and click **Next.**
18. Leave **Generate New Certificates** selected and click **Next.**
19. Change the **Certificate Directory** for the certificates in the HTTPS Server Settings to **C:\DeadlineCertificates .**
    - Leave **Certificate Password** and **Confirm Password** blank
20. Click **Next** and **Next** again.
    - Again, wait a few minutes for the installation to complete.
21. Click **Finish.**

Copy the Deadline Submitter Installers to the FSx File Share

The Deadline submission installers will allow you to install plugins for the digital content creation applications that you are going to be working in so you can launch renders directly from the app.

1. Navigate to the Deadline Repository folder:
   - **C:\DeadlineRepository10**
2. Copy the entire **submission** folder to the FSx share (the Z:\ drive) in this location:
   - **Z:\installers\thinkbox**

   Note: If you get an error that a file could not be copied and a message from Windows Defender Antivirus, you may have to temporarily disable it in order for all the files to copy.

   a. Go to the **Start Menu** and type **windows defender**
   b. Select **Windows Defender** settings

   c. Select **Virus & threat protection**
   d. Under **Virus & threat protection settings** click **Manage settings**
   e. Under **Real-time protection** click the slider to turn it to **Off**

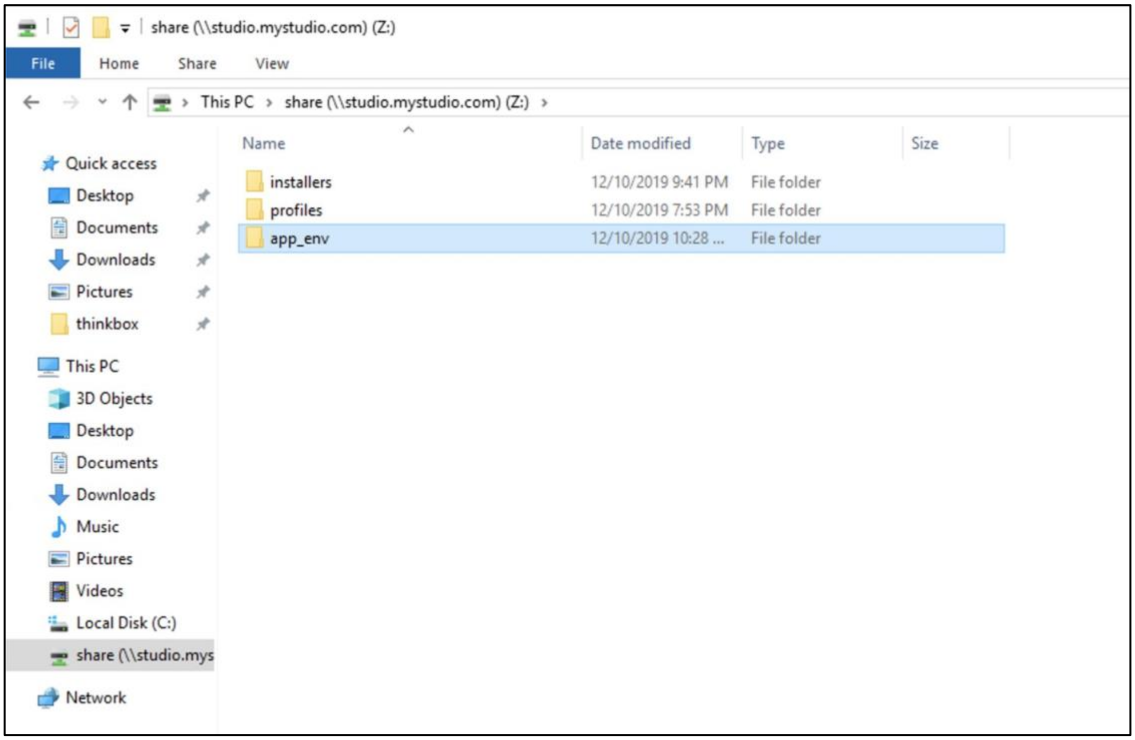
3. Retry the copy of the submissions folder. Once the copy has completed, you can turn real-time protection back on.

Copy the Deadline Certificates to a Shared Space

In order for the workers to connect, they need access to the Deadline Certificates you created in the **C:** drive. However, those need to be copied to the **Z:** drive so they are accessible on the network.

We will start by creating a folder on the **Z:** drive called **app_env** (short for application_environment). You can use this folder to put any application specific scripts, environment settings, etc. For now, we'll just use it for Deadline.

Create a folder called **Z:\app_env .**

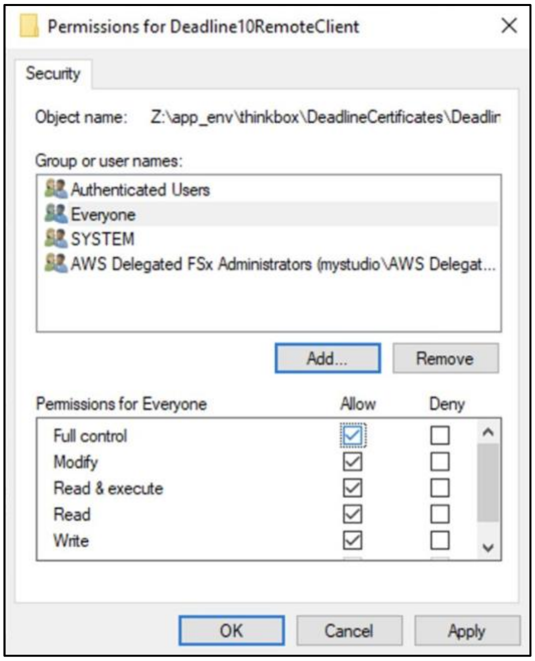


Next, we will create the folder for the Deadline specific files, and copy the certificates there.

1. Create a folder called **thinkbox** in **Z:\app_env .**
2. Copy the folder **C:\DeadlineCertificates** to **Z:\app_env\thinkbox .**
3. You will need to change permissions to allow for Domain Users to have access to **Deadline10RemoteClient .**
   a. Navigate into **Z:\app_env\thinkbox\DeadlineCertificates** and RMB on **Deadline10RemoteClient** and choose **Properties.**
   b. On the **Security** tab, click **Edit** to change permissions.
   c. Click **Add**

      You may need to log in as your *studio* administrator (e.g., mystudio\Admin).
   d. Then under "Enter the object names to select" type in **Everyone**.
   e. Click **Check Names.**
   f. Once Everyone is underlined, click **OK.**

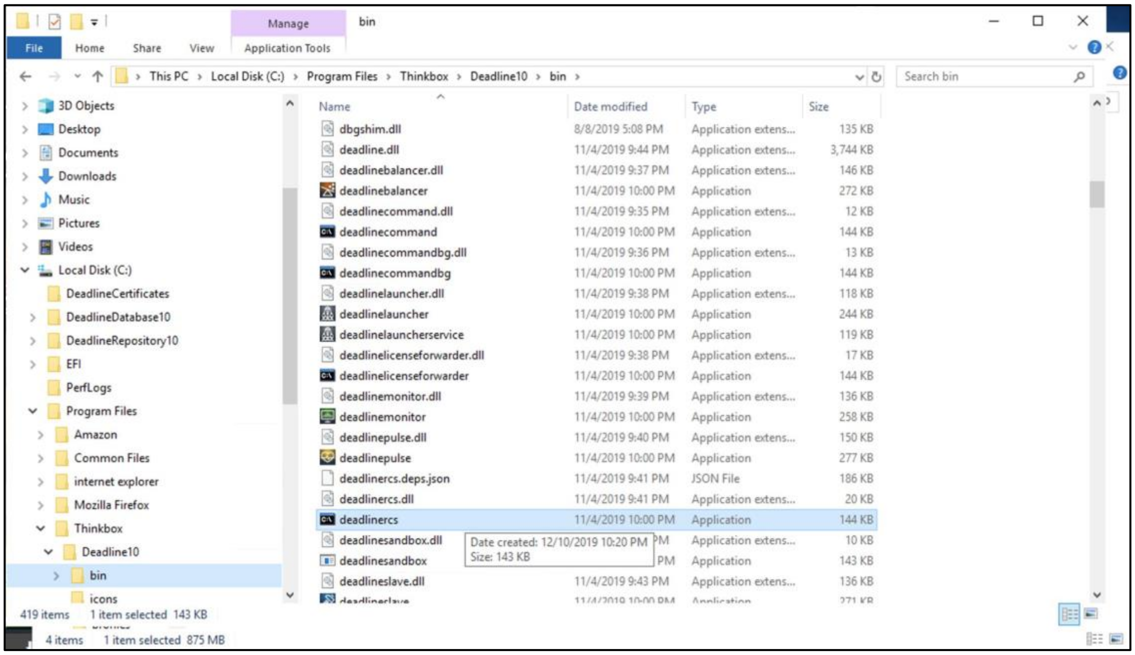
g. Then click the **Full Control** checkbox under **Allow** to add all permissions possible for Everyone.



h. Click **OK** and **OK** again.

Run Deadline Remote Connection Server (RCS)

1. Go to **C:\Program Files\Thinkbox\Deadline10\bin .**
2. Scroll down to **deadlinercs.**



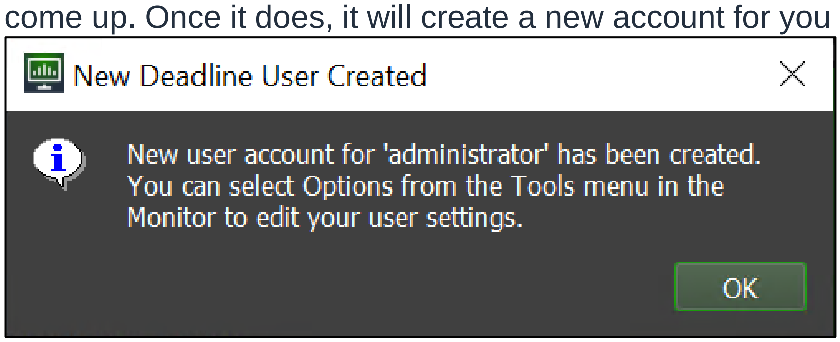
3. Double-click **deadlinercs** and a new window will open up (you might need to wait a minute for it to start).
4. Minimize the window to leave it running in the background.

Setup Deadline Monitor

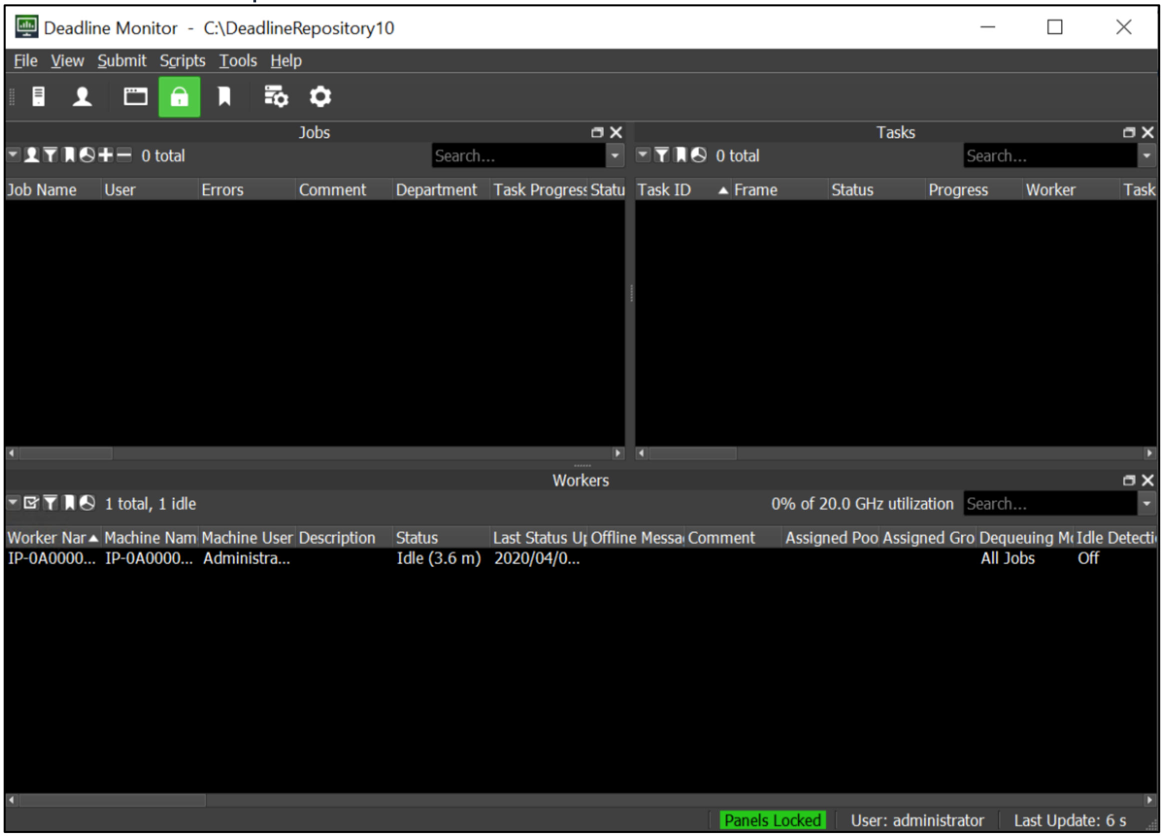
Launch Monitor

1. From the **Start Menu** choose **Thinkbox→Deadline Monitor 10.**

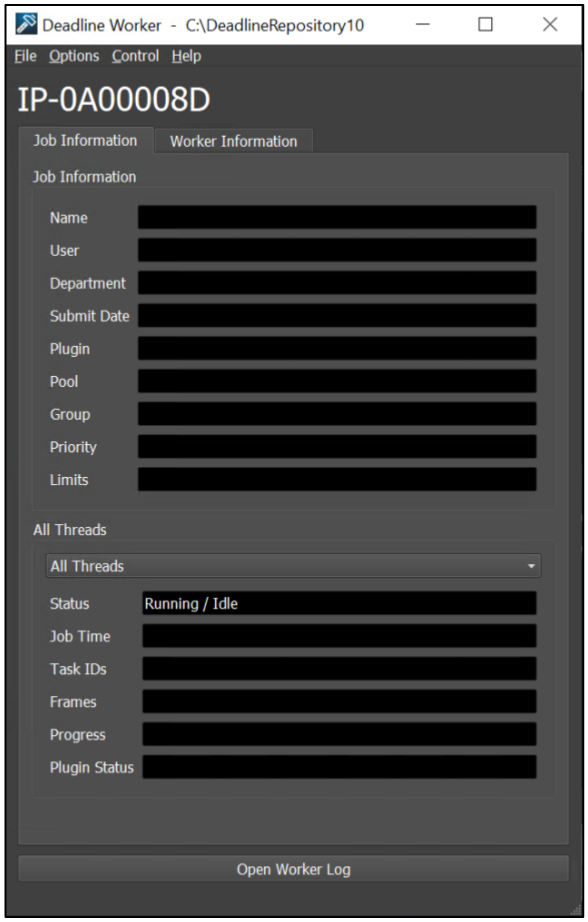
It will take a minute to come up. Once it does, it will create a new account for you called **administrator.**



2. After clicking **OK** in the window above, you should now see the Deadline Monitor window open.



The **Deadline Worker** app will have also automatically launched. The worker window may be behind the monitor window and looks like this:



The worker is the application that actually does the job of rendering on whatever computer it is running on. In exercises later in this tutorial and the next, we'll be using this worker to test our Deadline setup. Without this running worker, our render tests will not get picked up.

Like deadlinercs, make sure that you leave the worker running, but you can minimize the window.

Back in the Deadline Monitor window, you will see this single worker listed in the bottom panel of the monitor. That is the worker running on your Render Scheduler.

If the Deadline Worker didn't start, you can start it manually by going to **C:\Program Files\Thinkbox\Deadline10\bin** and double-clicking on **deadlineworker.exe.**

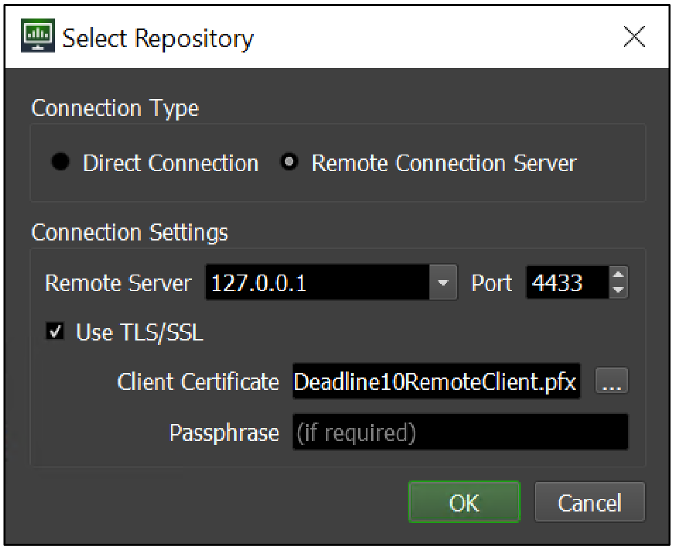
Create an Identity Registry Setting

Deadline securely manages sensitive information such as API keys, Usage Based Licensing activation codes, passwords, etc. using secrets. This is similar to how we used AWS Secrets Manager to store your Active Directory Admin password in the last tutorial.

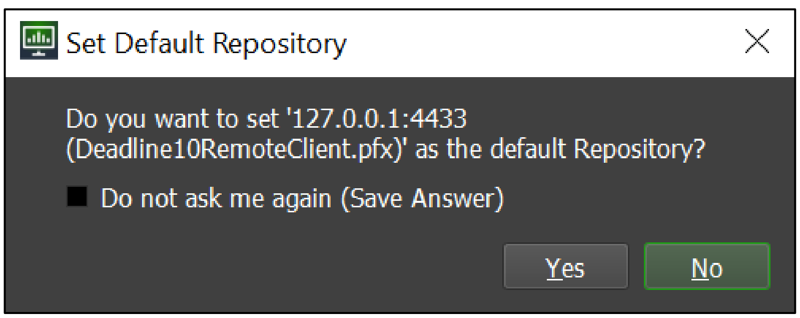
Deadline assigns “server” or “client” roles to each machine that attempts to connect to the Deadline Repository. When we installed the Deadline Client above, your Render Scheduler was automatically assigned to the “server” role since it will be running the Remote Connection Server (RCS). We need to make sure that the workstation and farm worker machines that we create in later tutorials are assigned “client” roles, otherwise they will not have access to the Deadline secrets. To do that, we’re going to create an Identity Registration Setting in the Deadline Monitor.

By default the Render Scheduler connects to the Repository using a direct connection, but in order to manage identity settings, we must connect using a secure remote connection instead. So first we’ll change our repository connection and then move on to creating the identity settings.

1. In the **Deadline Monitor**, choose **File→Change Repository…**
2. For **Connection Type** select **Remote Connection Server**
3. Next, under **Connection Settings** set the following:
    a. **Remote Server** to **127.0.0.1**
    b. **Port** to **4433**
    c. Make sure **Use TLS/SSL** is checked
    d. Set **Client Certificate** to **C:\DeadlineCertificates\Deadline10RemoteClient.pfx**
    e. Leave **Passphrase** blank

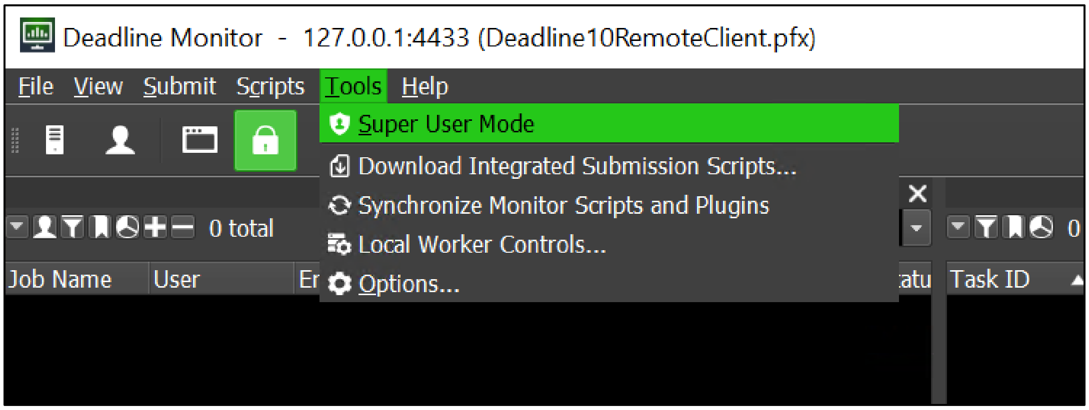


4. Click **OK**
5. Click **No**



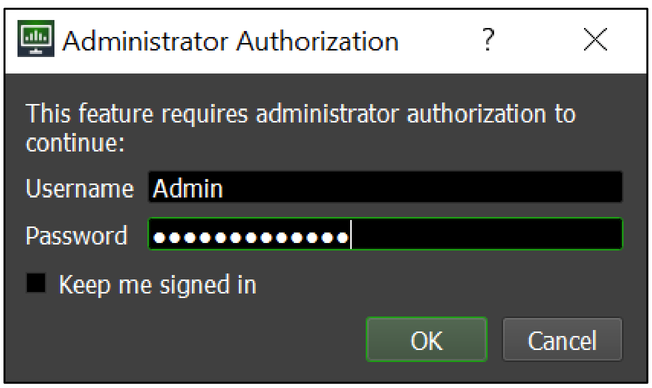
Note: We are only temporarily connecting to the Repository with a Remote Connection. We don't want to make this the default, which is why we're answering "no" above. Later when we restart the Render Scheduler and reopen the Deadline Monitor, it will automatically reconnect using a Direct Connection again.

6. Wait until it reconnects to the Repository, then choose **Tools** → **Super User Mode**



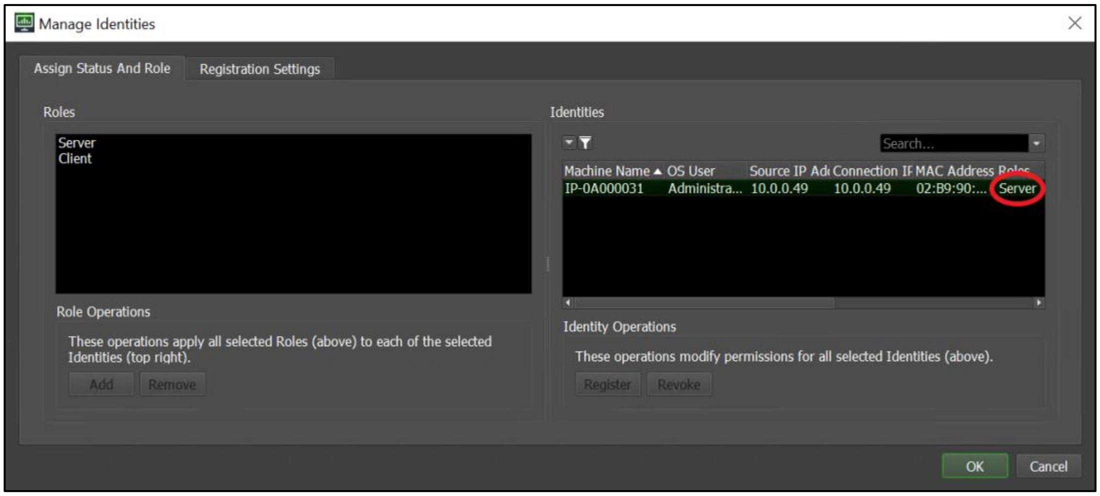
7. Then choose **Tools** → **Manage Identities…**

8. Enter the Secrets Management **Admin Username** and **Password** that you created when you installed the Deadline and click **OK**. (Again, this may be same as your Active Directory admin password. *Refer to the Notes section of your*



cheat sheet if you used a different password.)

9. In the **Assign Status and Role** tab, you'll notice that your Render Scheduler is already assigned to the "server" role. Now we'll create a setting to automatically



assign your workstations and farm workers to the "client" role.

10. Click the **Registration Settings** tab

11. Click **Add** and then enter the following information on the right side of the window under Registration Settings:

    a. Set **Name** to **Workstations_and_Workers**

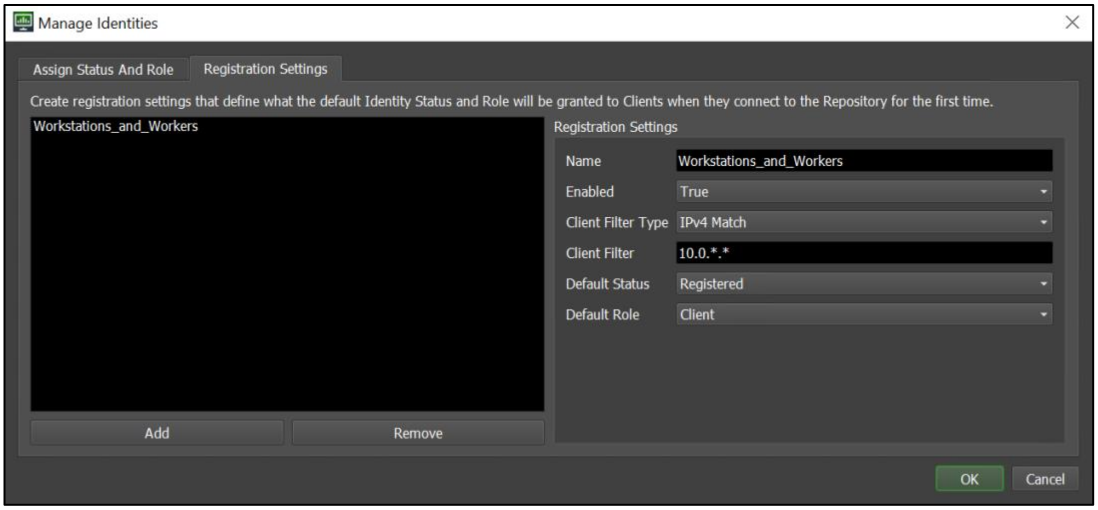
    b. Set **Enabled** to **True**

    c. Leave **Client Filter Type** set to **IPv4 Match**

    d. Set **Client Filter** to **10.0.*.***

    e. Set **Default Status** to **Registered**

    f. Set **Default Role** to **Client**



12. Click **OK**

Set Mapped Paths

Because we will be using Linux workers for rendering, we need to create some mapped pathing to make sure our file paths convert from Windows to Linux correctly. For example, a file located at Z:\project\myshow\shot_01.ma on your Windows machine will need to be mapped to /mnt/studio/project/myshow/shot_01.ma. Deadline can do this automatically when it’s set up correctly.

3. In the **Deadline Monitor**, choose **Tools** → **Configure Repository Options.**
4. Click **Mapped Paths.**
5. Under **Global Rules** click **Add.**
   - For **Replace Path**, enter **Z:**
   - For **Windows Path**, enter **Z:**

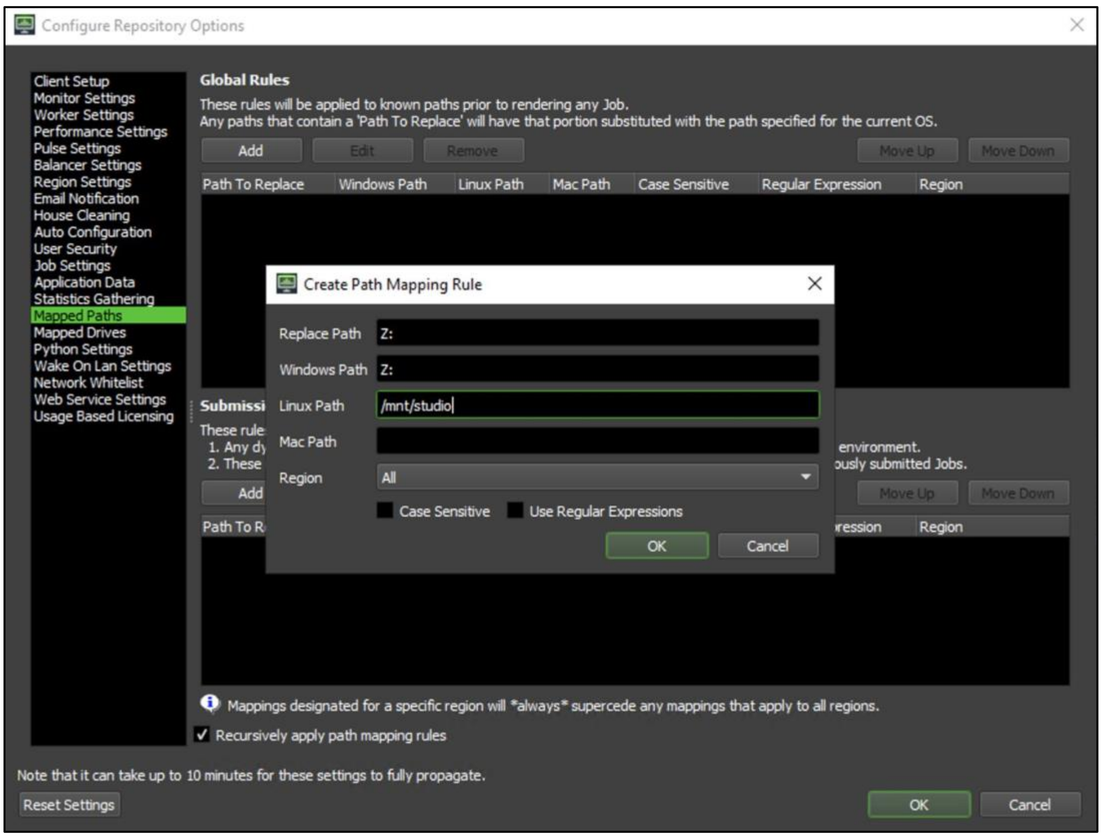
- For **Linux Path**, enter **/mnt/studio**

6. Click **OK.**
7. Click **OK** again to close the window.

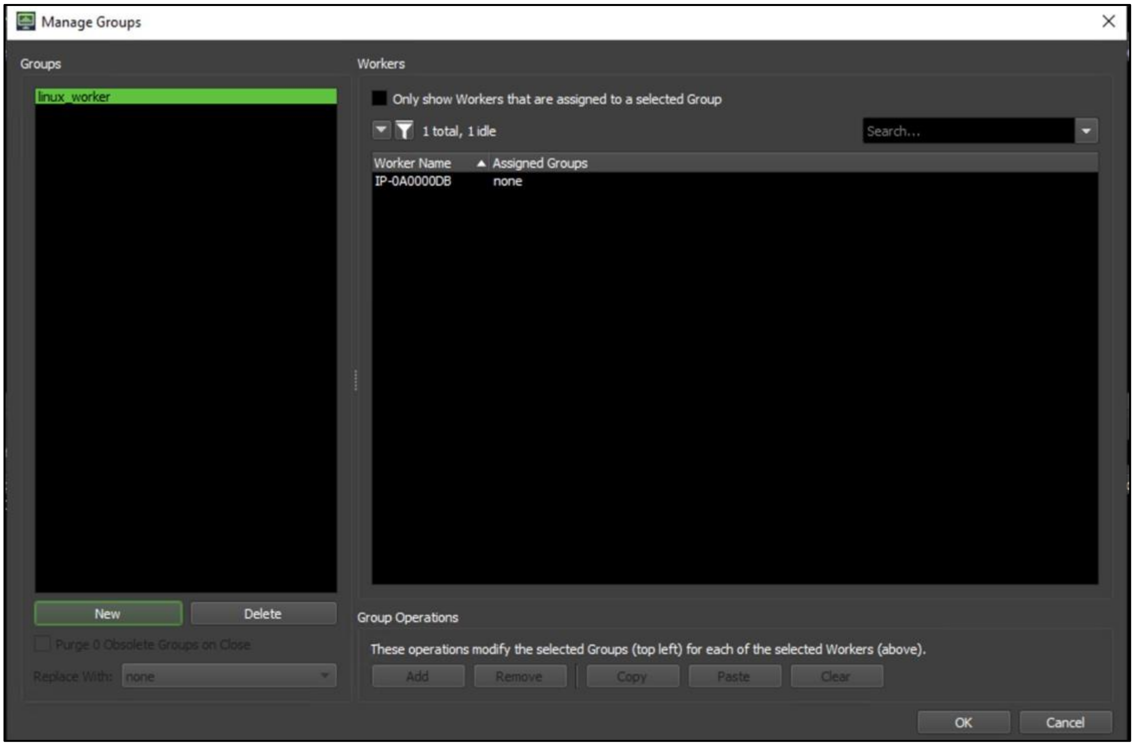
Set up Render Groups

Now we want to create a render group. You can create multiple groups if you want, but for our case a single one will be fine.

1. Go **Tools → Manage Groups...**
2. Click **New.**
3. For **Group name** enter: **linux_worker**



4. Click **OK.**



5. Click **OK** again.

Note: We will use this in a later tutorial, for now we're just setting up the group.

Run Deadline Pulse

Pulse is a mini server application that performs maintenance operations on your farm, and manages some advanced features.

1. Go to the **Start Menu** and select **Thinkbox→Deadline Pulse 10** and leave it running.

2. Minimize Pulse so you don't have to see the window all the time.

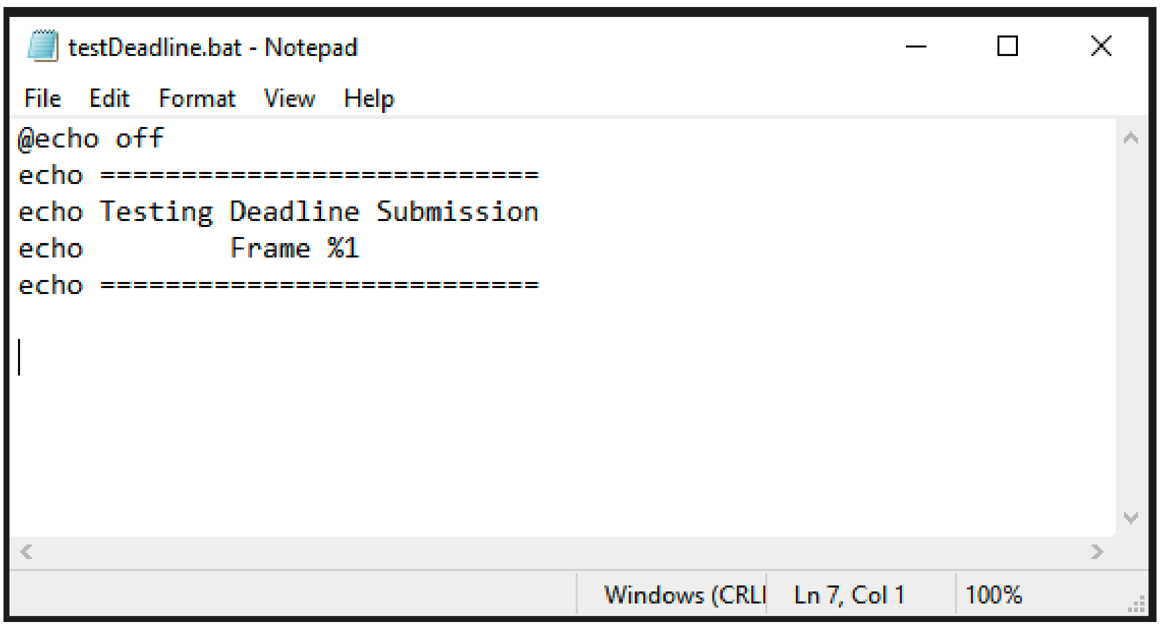
Test Your Setup

To test and make sure that jobs are able to run, you can create a very simple test that just outputs information to the render logs. This won't be how you normally run jobs on the farm, but it will submit a job through the Deadline Monitor, and have the current machine pick it up and execute it.

Create a Test Batch File

First, we'll create a batch file that can execute a simple command. Then, we'll submit this command with Deadline.

1. On your Render Scheduler, open **Notepad** by going to the **Start Menu** and typing **Notepad**.
2. Enter the following code:

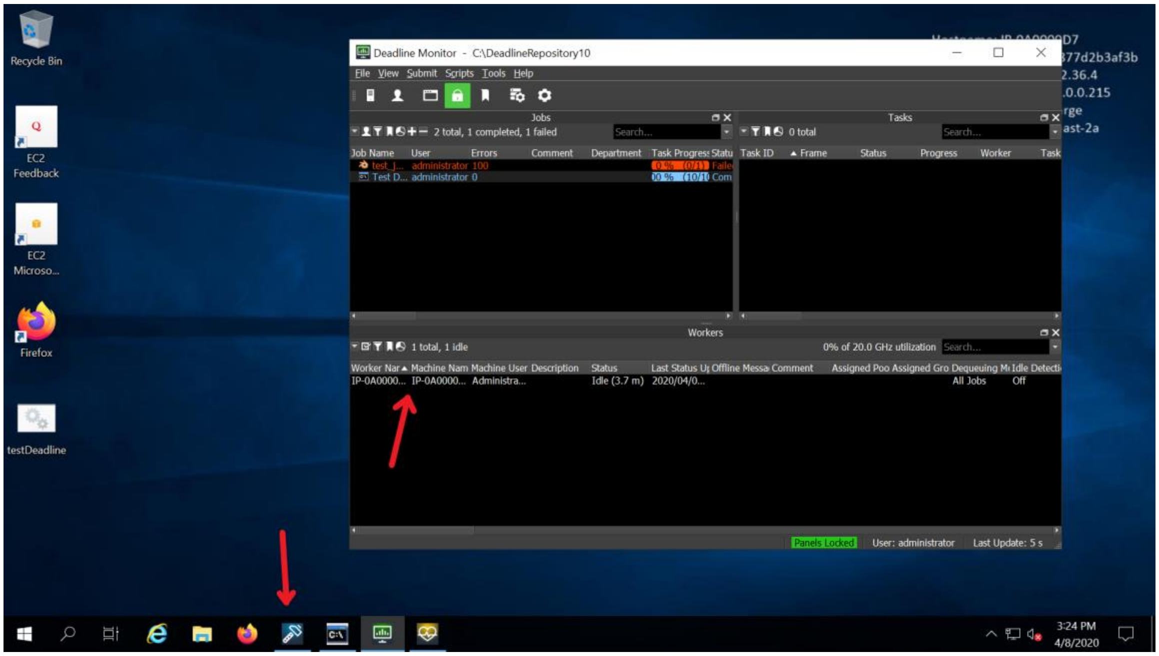


testDeadline – *Note: <shift>+click the image to open the text in a new tab.*

3. Save the file on the Desktop as **testDeadline.bat** (C:\Users\Administrator\Desktop\testDeadline.bat).

Check the Deadline Worker

Before you submit your render, you should check that the Deadline Worker is running. It should have automatically started when you launched the Deadline Monitor above, but it's good to check just in case. If the worker is running, you should see the Deadline Worker icon in the task bar and also see one worker in the list of workers at the bottom of the Deadline Monitor window.



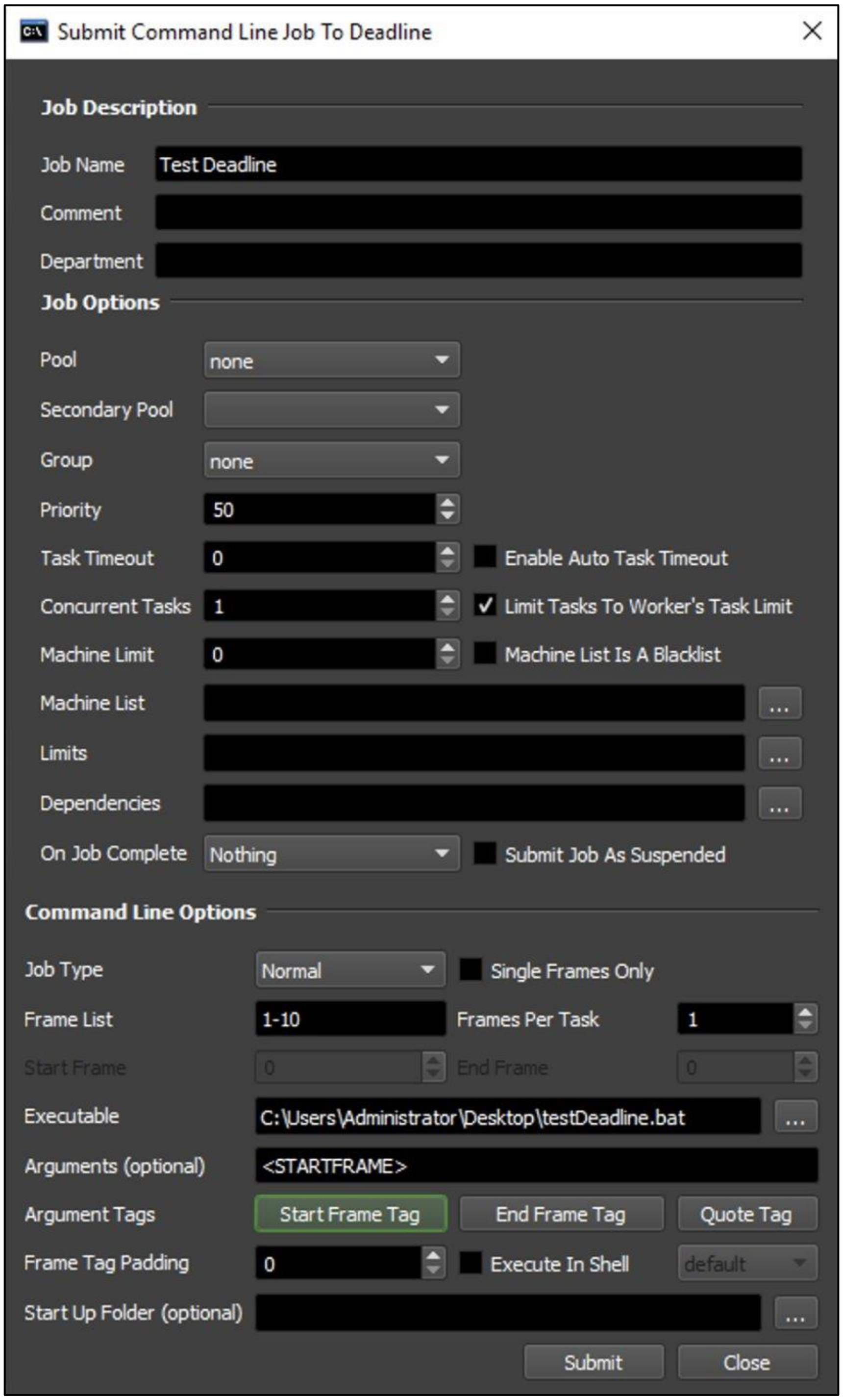
Submit the Test

Now you'll submit the test to Deadline. If everything works as expected, you will run a series of jobs on the current machine. Inside the logs of those jobs, you should see the statements above echoed out.

1. In the **Deadline Monitor** go **Submit→Miscellaneous→Command Line.**
2. Set the **Job Name** to **Test Deadline.**
3. Leave **Group** set to **none**. Note: We won't be using the "linux_worker" group until a later tutorial.
4. Set the **Frame List** to **1-10.**

   This will run the command for 10 frames.
5. Set the **Executable** to the location of your **testDeadline.bat file** (C:\Users\Administrator\Desktop\testDeadline.bat).
6. In **Arguments** enter **<STARTFRAME>**
   - You can do this by just clicking the **Start Frame Tag** button.
   - This will replace the %1 in the batch file with the current frame.

7. Click **Submit.**

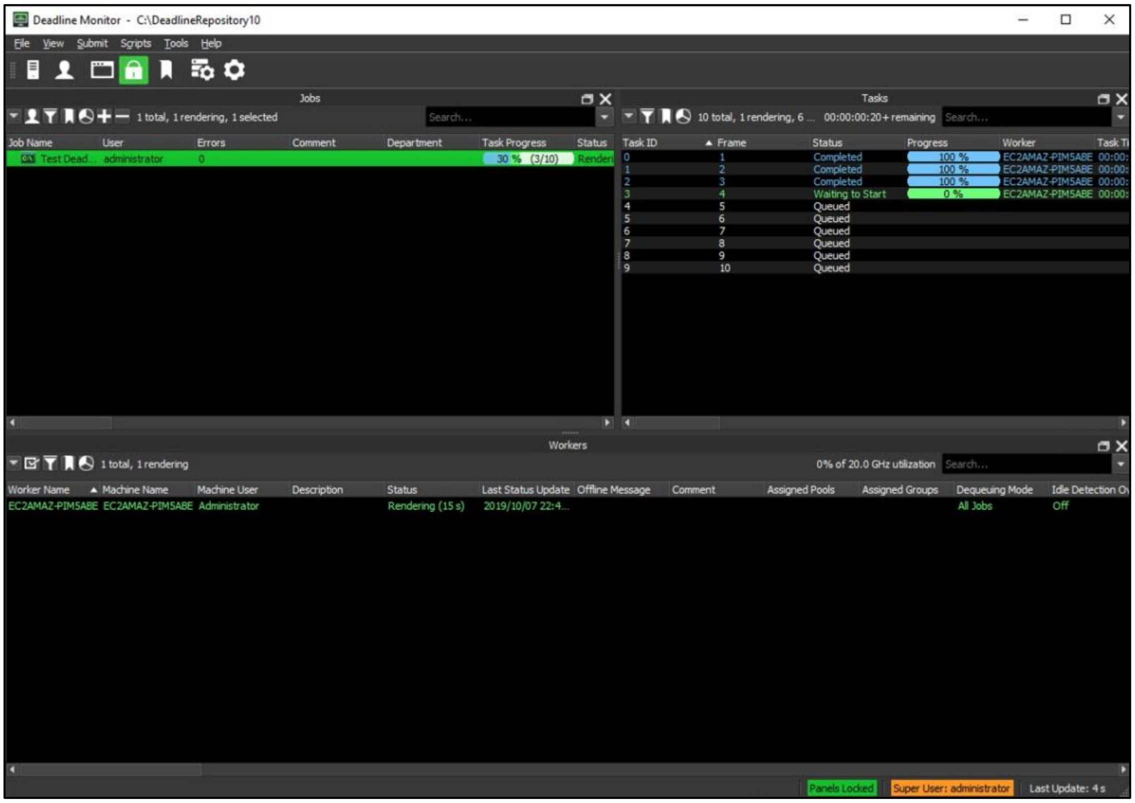


8. Close the Submission dialog.

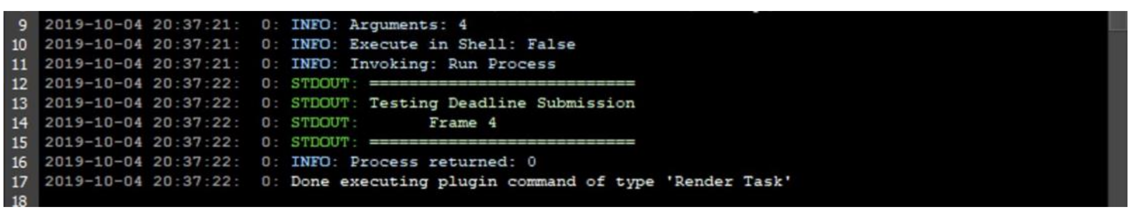
Review Your Submission

Now that the job is submitted, you should see it listed under **Jobs** in the upper left panel of Deadline.

1. Select the **Test Deadline** job, and you will see a list of **Tasks** in the upper right panel of Deadline.



2. **Double-Click** one of the **Completed tasks** to see the **logs**.
3. Select the log and then scroll down until you can see the code we entered.



Congratulations! You've now manually submitted a task and can see Deadline working.

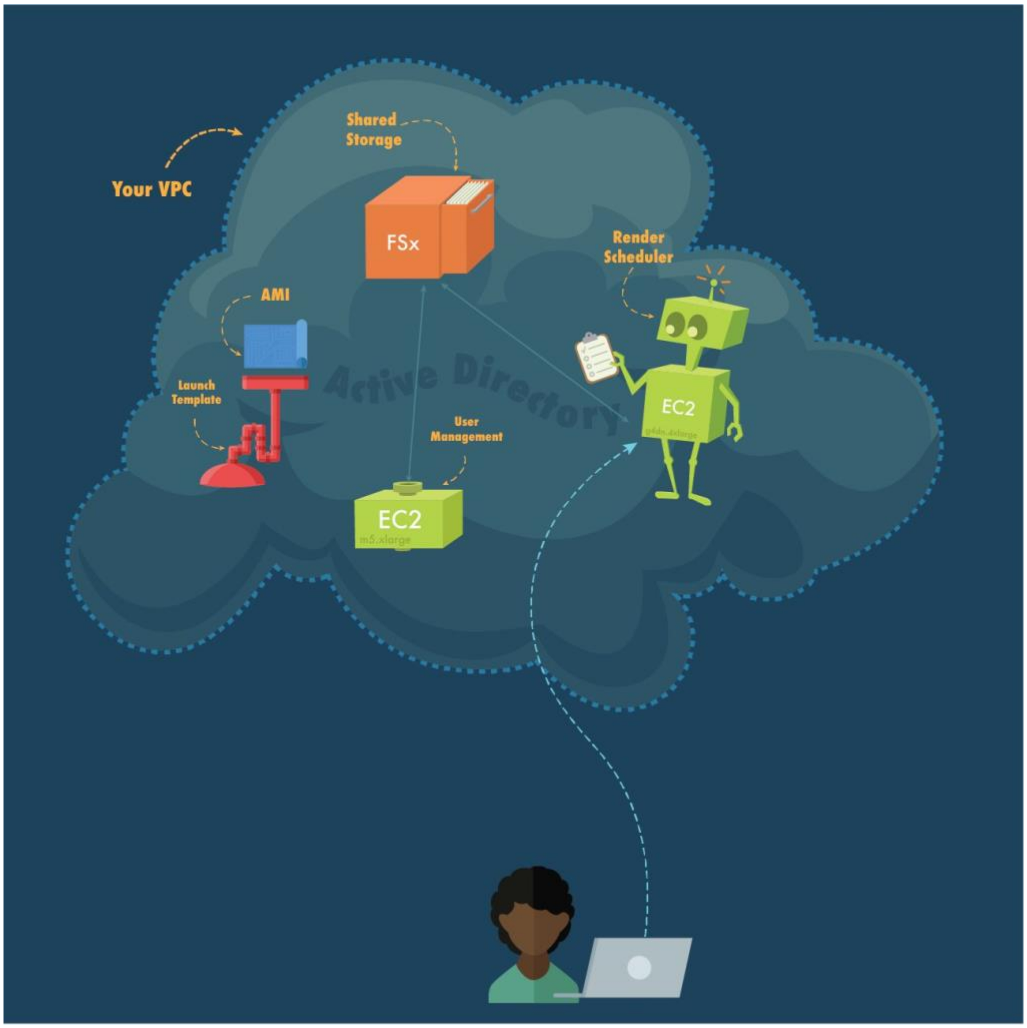
Restart Your Render Scheduler

In order to make sure the Render Scheduler can be connected to via other instances, you'll need to restart the machine.

1. In the **AWS EC2 Console**, select the Render Scheduler instance and choose **Instance state → Reboot instance.**
2. Once it's been restarted, you can connect again by clicking **Connect**
3. Log in as **Administrator.**
4. Then start up the **Deadline Monitor** and **Deadline Pulse**
    - **deadlinercs** should start automatically after you log in, but if it doesn't, you can start it manually by running **C:\Program Files\Thinkbox\Deadline10\bin\deadlinercs.**

Note: If you have trouble when launching any of the Deadline apps above, run **Start→Task Manager** and double-check that they aren't already running. Sometimes you need to force quit them and launch again for them to start correctly.

Your VPC So Far



Shut Down Notes

If you are going to continue to the next tutorial right away then leave your Render Scheduler running. If it is running, you can stop your User Management instance for now. You won't need it until you onboard more artists into your studio. If you are going to continue at another time you can **Stop** your Render Scheduler and restart it when you continue this series.

1. In the AWS Console go to the list of running instances.
2. Select your **User Management** and/or **Render Scheduler** instance from the list
3. Click **Actions**, then select **Instance State→Stop**

Appendix

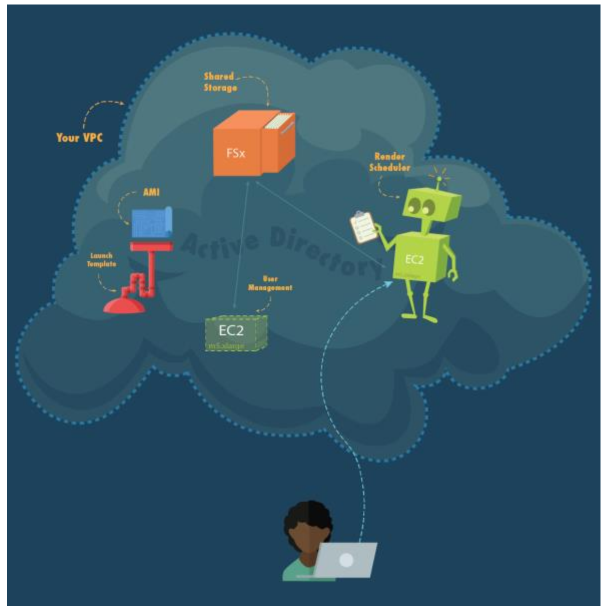
Links to AWS Documentation

- Deadline Repository Installation
- Deadline Client Installation
- Deadline Secrets Management
- Getting Started With Deadline Secrets Management

Tutorial 5. Installing Applications and Creating a Workstation AMI

Estimated Time to Complete: 1 hour, 30 minutes

In this tutorial we'll take you through preparing your **Windows workstation** and installing your applications, such as **Deadline** and **Blender**, as well as the creation of your Workstation **Amazon Machine Image (AMI)**. This will allow you to create a custom template that you can use to launch additional workstations without having to set everything up each time.

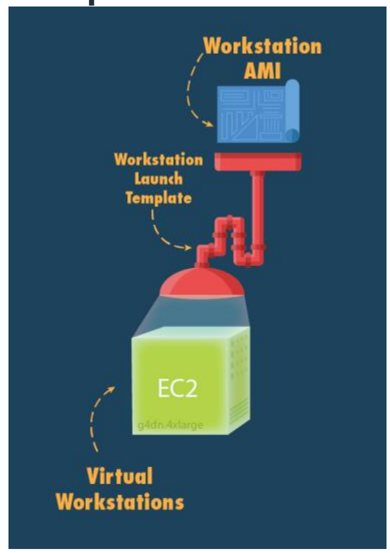


Startup Notes

If you just finished the last tutorial, you already have your Render Scheduler instance running and can continue. However, if you stopped your Render Scheduler when finishing up the last tutorial, navigate to the EC2 dashboard and start it now. Once it's running, make sure to login as Administrator and start up the **Deadline Monitor**, **Deadline Remote Connection Server (RCS)**, and **Deadline Pulse.** The **Deadline Worker** should start automatically after you login, but you should check that it's running and launch manually if needed.

Once you've done that you can continue onto the next step.

Launching a New Instance from Your Launch Template



1. In the **EC2 Dashboard** navigate to your **Launch Templates**
2. Select your Management Launch Template (e.g., My-Studio-Management-LT), click the **Actions** menu and select **Launch instance from template**
3. Select the most recent version in **Source template version**
4. Leave number of instances at **1**
5. In **Instance Type,** change it to a **g4dn.4xlarge** Note: If you don't see the g4dn.4xlarge available, you can use a g3.4xlarge instance instead.
6. In **Storage (volumes)** make sure you have enough space to install all the applications you need, we recommend **150 GiB** if you have multiple applications you will be installing
7. In **Resource Tags,** change the **Name** value to **Workstation_Win.**
8. You can leave the rest of the launch template at its defaults and click **Launch instance from template**

   Note: If you are using a newly-created AWS account or have never launched a G4 or G3 instance before, you may see a warning message saying that you have requested more vCPUs than your current limit. If that happens, see the Appendix in Tutorial 1 for instructions on how to check your current quota value and request an increase to at least 16 vCPUs.

9. After the instance has launched, go to the list of running instances, right mouse click the new instance and choose **Security → Change security groups**
10. Add the Deadline security group (e.g., My-Studio-Deadline-SG) that you created in Tutorial 4. *You can find the name and ID of your Deadline security group in the cheat sheet.*
11. Click **Save**

Create S3 Get Object Privileges

As you can see, we're launching a Windows instance with a GPU - this will let us run powerful 3D applications. For those applications to run correctly, they will need the latest NVIDIA graphics drivers. In a step below, we will download and install those drivers from an S3 bucket. In order to do this, we are going to need to give our instance permission to access them from S3. While waiting for your instance to start, you can enable these permissions.

1. Go to **Services → IAM**
2. Select **Roles** in the left panel
3. Search for and select **EC2DomainJoin**
4. Click **Attach Policies**
5. Search for **S3**
6. Select the check box next to **AmazonS3ReadOnlyAccess**

7. Select **Attach policy**

Log in to Your Workstation

Log in as Administrator

In order to install applications on your workstation instance, you should log in as **Administrator** just as you did in the last tutorial. Once logged in, you can begin the installation process for your applications.

To log into the **Workstation_Win** instance as Administrator:

1. Check that the instance is initialized with 2/2 checked passed

2. Select the instance and hit **Connect**
3. Choose the **RDP client** tab and click **Download the remote desktop file**
4. Open Remote Desktop
5. The username should already be set to **Administrator**. In Tutorial 3, we manually set the Administrator password to match your Active Directory Admin password, so you can just enter that here without having to click Get Password. *You can also find your Active Directory Admin password under the Tutorial 2 section of the cheat sheet.*

Set Up Graphics Drivers

Now it's time to download the drivers mentioned above. As a reminder, it's always important to make sure the drivers for your machines are up to date. The g4dn.4xlarge we are using has the NVIDIA T4 GPU, so let's grab the latest drivers from NVIDIA.

Note: If g4 instances are not available in your region and you launched a g3.4xlarge instead, there are some minor differences in the instructions below to get the proper drivers. Don't worry, we'll call out any changes that you need to make.

We'll be using some PowerShell commands to download the drivers and the NIVIDIA GRID Cloud End User License Agreement to the desktop of your instance.

1. Open a **PowerShell** window.
2. Start by running one of the commands below, depending on the type of instance you launched above:
   - If you launched a **g4dn.4xlarge** instance:

```
$KeyPrefix = "g4/latest"
```

   - If you launched a **g3.4xlarge** instance:

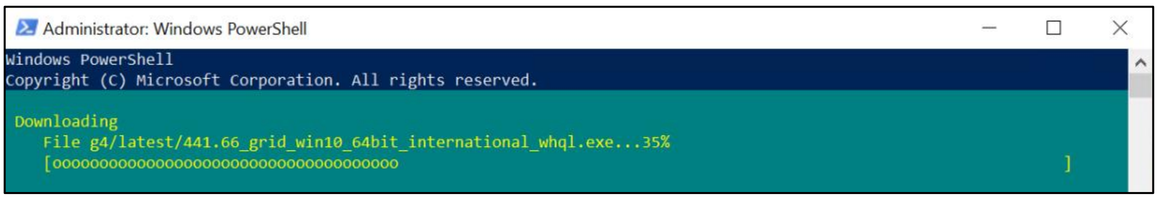
```
$KeyPrefix = "latest"
```

3. Next, copy and paste the following commands to actually download the drivers that you need. These commands are the same for both G4 and G3 instance.

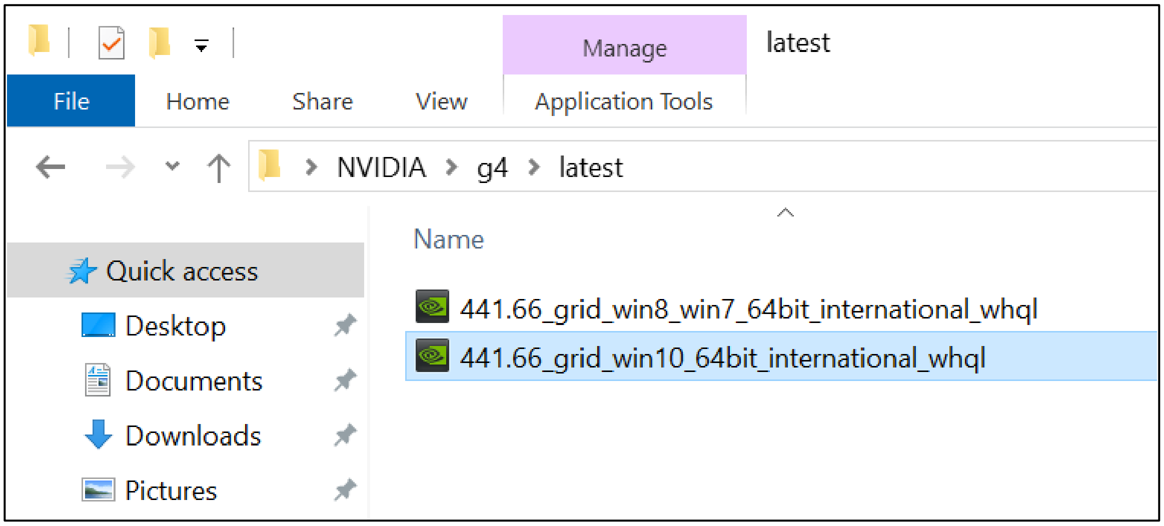
```
$Bucket = "ec2-windows-nvidia-drivers"
$LocalPath = "C:\Users\Administrator\Desktop\NVIDIA"
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region us-east-1
foreach ($Object in $Objects) {
$LocalFileName = $Object.Key
    if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
    $LocalFilePath = Join-Path $LocalPath $LocalFileName
    Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile $LocalFilePath -Region us-east-1
    }
}
```

***Commands** – Note: <shift>+click the image to open the text in a new tab.*

The download will begin, showing you the status in the shell

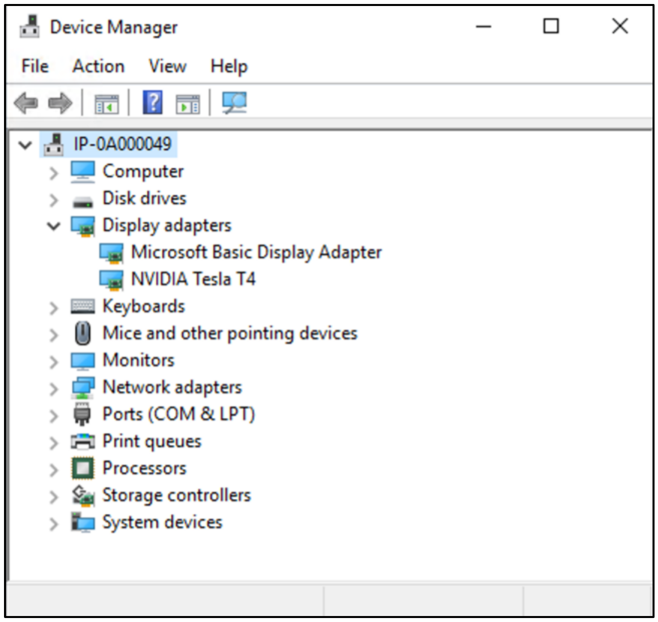


4. Once it's done, navigate to the desktop
5. Click the folder called **NVIDIA**
6. Depending on the type of instance, you'll navigate into a different folder next:
   - For a **g4dn.4xlarge** instance navigate into **g4/latest**
   - For a **g3.4xlarge** instance, navigate into **latest**
7. Double click the installer for **Windows 10** (The file name will look something like 441.66_grid_win10_ 64bit_international_whql)



8. Click **OK** to start the installer. Note: Sometimes the installer window pops up behind the File Explorer window.
9. Click **Agree and Continue** (or **Ok** if you see that instead), then **Next.**
10. When you get to the end of the installation process, click **Close**, then **restart** your instance. (Or click **Restart Now** to restart your instance.)
11. After waiting a minute or so for the instance to restart, **Reconnect** to your workstation and login as Administrator again.
12. To verify that your GPU is working properly, open the **Start Menu**, type **Device Manager** and then select **Device Manager** from the list
13. Click the **>** next to **Display adapters**

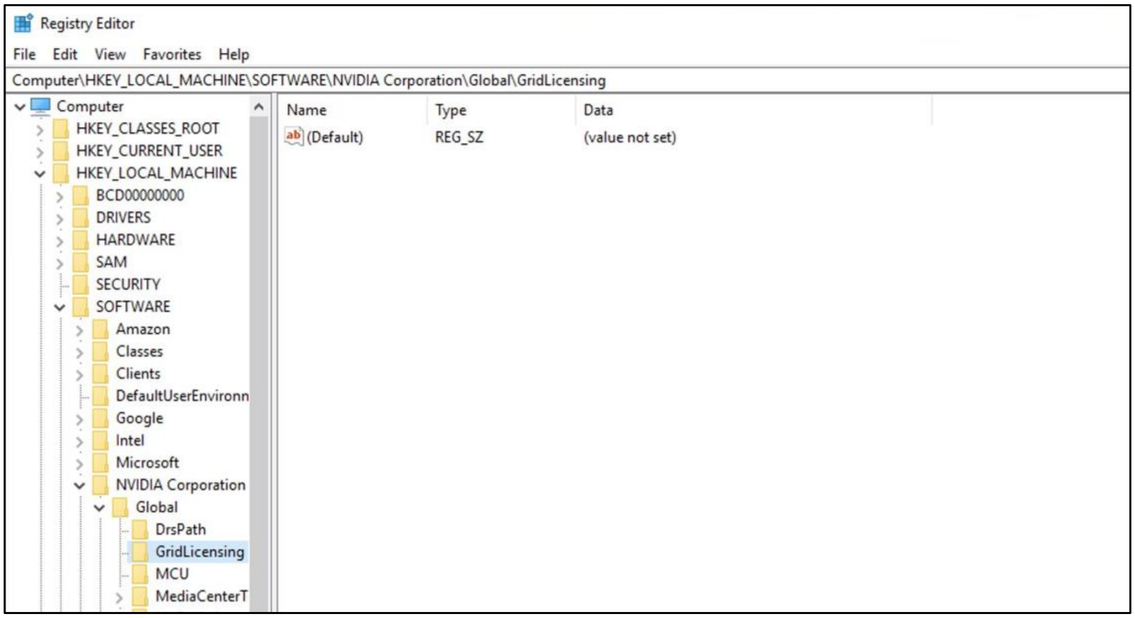
    If you see **NVIDIA Tesla T4** (or NVIDIA Tesla M60 if you're using a g3.4xlarge instance) then you are all set.



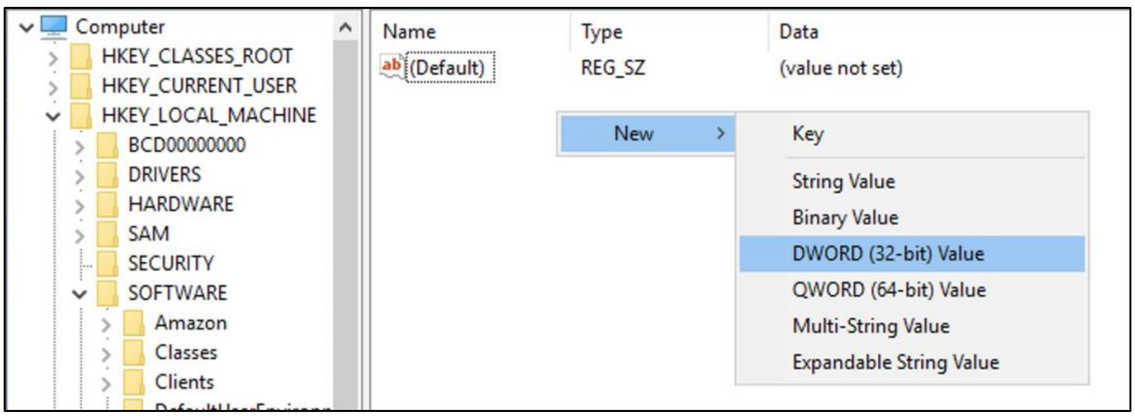
14. Exit Device Manger.

Now, we will disable the licensing page in the control panel to prevent users from accidentally changing the product type (NVIDIA GRID Virtual Workstation is enabled by default). For more information, see the [GRID Licensing User Guide].

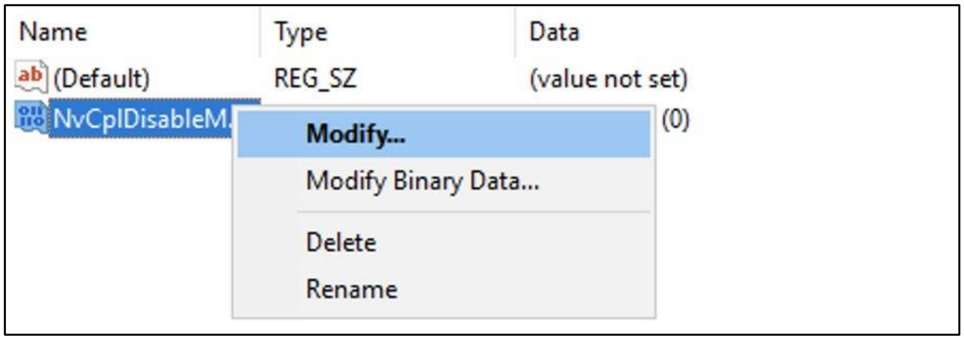
1. Go to the **Start Menu** and type **reg** and select **Registry Editor.**
2. Navigate to **HKEY_LOCAL_MACHINE** → **SOFTWARE** → **NVIDIA Corporation** → **Global** and then select **GridLicensing.**



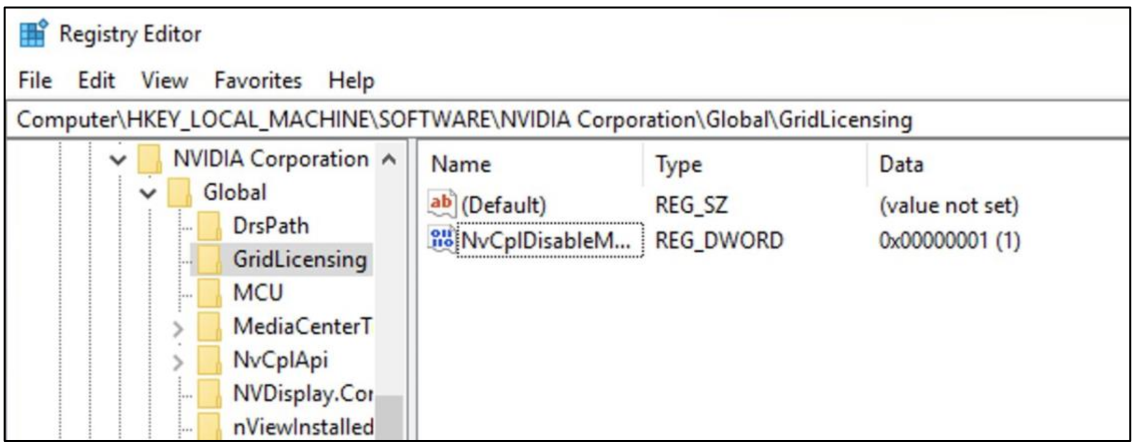
3. Right-click the right pane and choose **New**, **DWORD (32-bit) Value**.



4. For **Name**, enter **NvCplDisableManageLicensePage** and hit **Enter**.
5. Right-click **NvCplDisableManageLicensePage** and choose **Modify**.



6. For **Value data**, type **1** and choose **OK**.



Set Up Application Installers

Setting up your applications to install onto your instance is as simple as downloading the installer for an application as you would regularly on a physical workstation. For the purpose of this setup we're going to stick with **Blender 2.93**, a powerful open source 3D content creation application, since it is free and does not require licensing.

Connect to the Studio FSx Drive

Since we've already created an FSx Drive that has a location for installers to be stored, we can connect to it with this workstation using the same steps we did in Tutorial 3.

1. Open a **File Explorer** window.
2. From the navigation pane, right-click **This PC** and choose **Map Network Drive.**
3. Choose a drive letter of your choice for **Drive** (e.g., **Z:**).

4. Enter the full CNAME Alias for your file share that you noted above for **Folder**.

   e.g., \\studio.mystudio.com\share\

5. Click **Finish.**
6. Use your Active Directory Admin login credentials. For example:
   - Username: Admin
   - Password: Your Admin Password

Install Firefox

We are going to use Firefox as our browser on our workstations and because the installer is already saved onto the FSx Drive, it can be easily accessed.

1. Navigate to **Z:\installers\firefox**
2. Run the installer from that location.

   Note: We already installed Firefox on the User Manager instance, but we haven't installed it yet on this machine. That's why we're going through the process again. This will be the last time we need to install Firefox on one of our instances because we will be saving this as an AMI (Amazon Machine Image).

Install Blender

1. Open up Firefox and navigate to https://www.blender.org
2. Click **Download** in the menu bar at the top and then **Download Blender 2.93.1** (or whatever the latest version is).
3. Download the 64-bit Windows installer and the Linux installer. You'll need both!
4. Create a **blender** folder in **Z:\installers.**
5. Once the download is complete navigate to your **Downloads** folder.
6. Move the installers to **Z:\installers\blender.**
7. Launch the Windows installer.
8. Follow the on screen prompts, accept the license agreement and leave everything else at the default settings.

Blender is now installed on your instance!

Copy Blender Shortcut to the Z: Drive

When you installed Blender, it created a shortcut on the desktop. Unfortunately, this shortcut doesn't carry over when other users (mystudio\Admin, mystudio\jason) login to the machine. But we can fix that by copying the shortcut to the Z: drive.

1. Open **File Explorer** and create a new folder named **applications** in the **Z: drive** (Z:\applications).
2. Click drag the Blender shortcut from the desktop to **Z:\applications.**
3. Now any user who logs in can run Blender by double-clicking on Z:\applications\blender **.**

Additional Applications

There are two additional applications we would recommend installing, **DJV** and **Krita**.

- **DJV** (http://djv.sourceforge.net/) **-** Professional media review software for VFX, animation, and film production.
- **Krita** (https://krita.org/) - Professional open source painting program.

For each of these, we recommend creating a folder inside **Z:\installers** (e.g., Z:\installers\krita, Z:\installers\djv), downloading the installer files from the appropriate websites, and copying them to the installers location. Only the Windows version is needed for these. Then, install the applications directly on the instance just like you did with Blender.

Install AWS Thinkbox Deadline Client

We need to install the Deadline Client on our workstation in order to have access to Deadline Monitor and to connect to the Render Scheduler.

1. Go to **Z:\installers\thinkbox .**
2. Double-click the **Deadline-10.1.17.4-windows-installers** folder (your version may be different).
3. Double-click the **DeadlineClient** installer.
4. Click **Run.**
5. Click **Next.**
6. Accept the license agreement and click **Next**.

7. Leave the **Installation Directory** as **Default.**
8. On the Select Installation Type page, leave **Client** selected and click **Next.**

   Note: There is no need to install the Remote Connection Server because this workstation is simply a client.

9. Leave the **Repository Connection Type** set to **[Recommended] Remote Connection Server** and click **Next.**

   Note: Because we will be connecting to another instance, we must set the connection type to Remote Connection Server.

10. Put the **Private IP address** of the **Render Scheduler** machine into **Server Address**, and point it at port **4433.**

    You should have already written down the Private IP for your render scheduler on the cheat sheet during Tutorial 4. It should look something like (10.0.0.219).

    Enter the private IP with a port number of 4433 into the server address field (e.g., 10.0.0.219:4433).

    Note: This is important, because the default port is 8080 and that won't work.

11. Click **Next**
12. Set the **RCS TLS Certificate** to **Z:\app_env\thinkbox\DeadlineCertificates\Deadline10RemoteClient.pfx .**

    Note: Make sure you select the Deadline10RemoteClient.pfx file and not the Deadline10Client.pfx file, which is also in the same folder. You might have to enter this path manually if the UI won't let you select the Z: drive.

13. Leave **Certificate Password** blank and click **Next.**

14. Uncheck the **Launch Worker When Launcher Starts** check box and click **Next.**

    Note: We are turning this off because we are currently setting up one of our workstation machines. We don't want render workers to launch on these instances.

15. Leave **Block auto upgrade via a secure setting** selected and click **Next**.

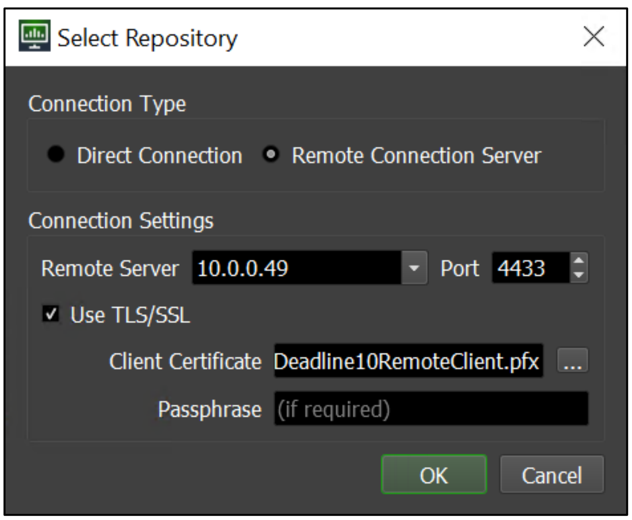
16. Click **Next** again to finish the install.

    Note: You may get an error that says "An error occurred when trying to set the repository connection settings". That's okay, we'll fix it later.

Connect to the Deadline Repository

On the **Start menu**, go to **Thinkbox** → **Deadline Monitor**

- If you get an error saying it can't connect to the Repository make sure you have these options selected in the prompt (Note: your IP address will be different.

  Make sure you have the correct port selected.)

- If it still doesn't work, double-check that your Render Scheduler instance is running, you're logged in as Administrator, and **Deadline Monitor, Deadline RCS,** and **Deadline Pulse** are all running.

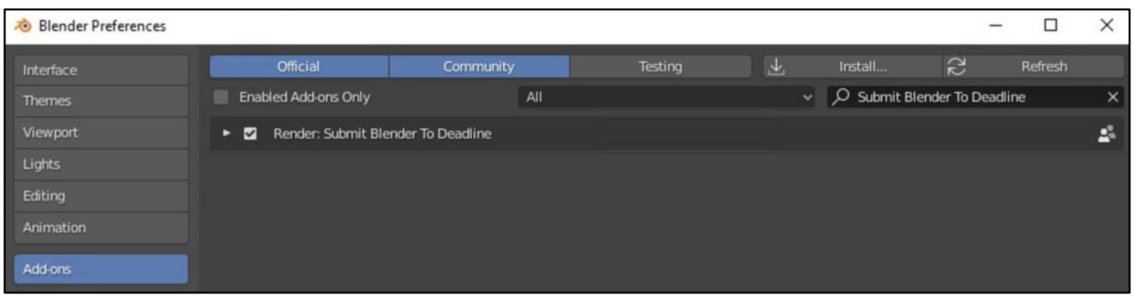
- If it works correctly, you will be able to connect to the Deadline Monitor.

Install Deadline Blender Submitter

In preparation for working with your render farm, we recommend you install the Deadline Submitter add-on inside of Blender.

1. Run **Blender.**
2. Go **Edit → Preferences...** (or **File→Preferences** - for Blender 2.79 & earlier).
3. Click **Add-Ons** in the left panel.
4. Click **Install.**
5. Navigate to **Z:\installers\thinkbox\submission\Blender\Client**

   Note: We copied the submissions folder to Z:\installers\thinkbox in Tutorial 4. If you don't see it there, please refer back to the instructions in that section.
6. Choose **DeadlineBlenderClient.py .**
7. Click **Install Add-on.** Note: Sometimes it can take a minute or two for the add-on to install, so if it looks like nothing is happening, don't worry.
8. Click the check box next to **Render: Submit Blender to Deadline** add-on.



9. Close your preferences window.

Test Submitting to Deadline

1. Open the **Deadline Monitor** and check that there is a single worker in the list of workers at the bottom of the window with a status of "Idle". This is the worker running on your Render Scheduler. If you don't see a worker listed or its status is "Offline", login to your Render Scheduler and launch the Deadline Worker before continuing.
2. Back in Blender, **save** your current scene (this is a requirement to submit to deadline).

3. Choose **Render→Submit To Deadline.**

   The Submit Blender Job To Deadline window will pop up.

4. Leave the **Group** set to **none**.

   Note: We won't be using the "linux_worker" group until the next tutorial.

5. Change the **Output File** path to a location of your choosing.

6. Change the Frame list from 1-250 to **1.**

7. Click **Submit** and then close the submit window

   Note: You'll get warnings if either your Blender file or your output file are set to the C: drive of the instance, but don't worry, for this test that's fine. You can just click **Yes** in any popups you see.

8. Open the **Deadline Monitor.**

Your job will be sitting in the job queue.

Don't worry if the job fails or cycles between "Queued" "Waiting to Start" and "Rendering". We are going to set up workers in the next tutorial that can successfully render jobs for you.

The goal of this exercise is just to make sure that your workstation can connect to the Render Scheduler. As long as your job appears in the job queue in the Deadline Monitor, then you are good to go.

Creating an Amazon Machine Image

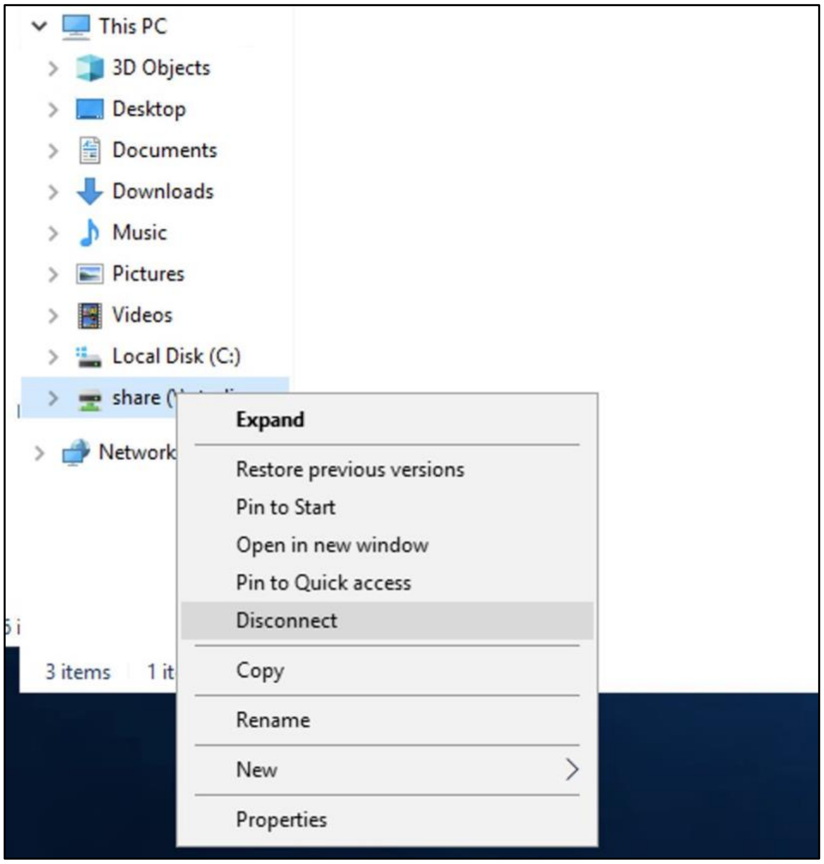
Now that you've installed your applications and set up the graphics drivers, you'll want to make sure you can launch another workstation just like this one but without all the hassle of setting it up again. Luckily, we've already done something similar in Tutorial 3 when we set up the launch template.

What we're going to be doing this time is creating an AMI (Amazon Machine Image) by making an image of the machine you are using that includes all the applications you've installed and updates you've completed on the instance, and allow you to create as many copies of that instance as you need for your studio.

Preparing Your Instance

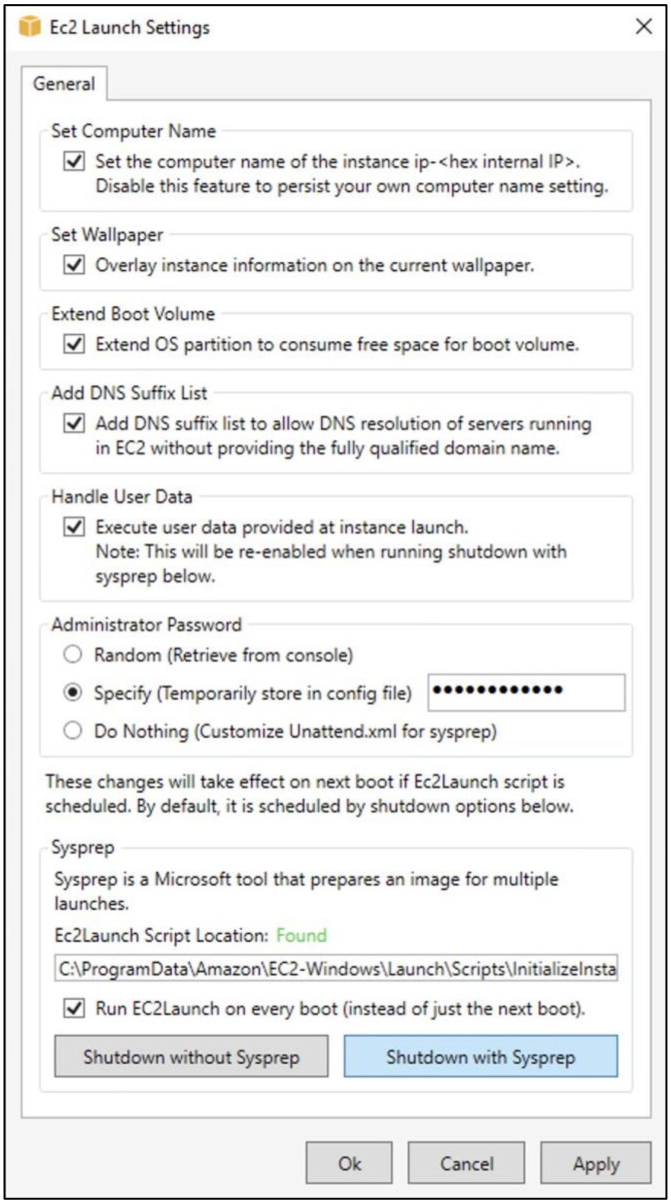
1. Close Blender and the Deadline Monitor

2. Disconnect from your FSx Drive (Z:).
   a. Open the **File Explorer.**
   b. Under **This PC** on the left, click the Z: drive with the **Right Mouse Button** and choose **Disconnect.**



3. Go to the start menu, type **Ec2LaunchSettings** and launch it.
4. Make sure **Set Computer Name** is selected.

   This will ensure each instance you create has a unique name.
5. Check that **Administrator Password** is set to **Specify** and input the Administrator Password for your Active Directory (e.g. password for mystudio\Admin).
6. Select **Run EC2Launch on every boot.**
7. Click **Shutdown with Sysprep.**

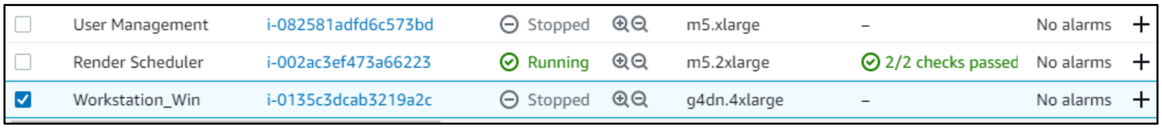


8. Click **Yes.**

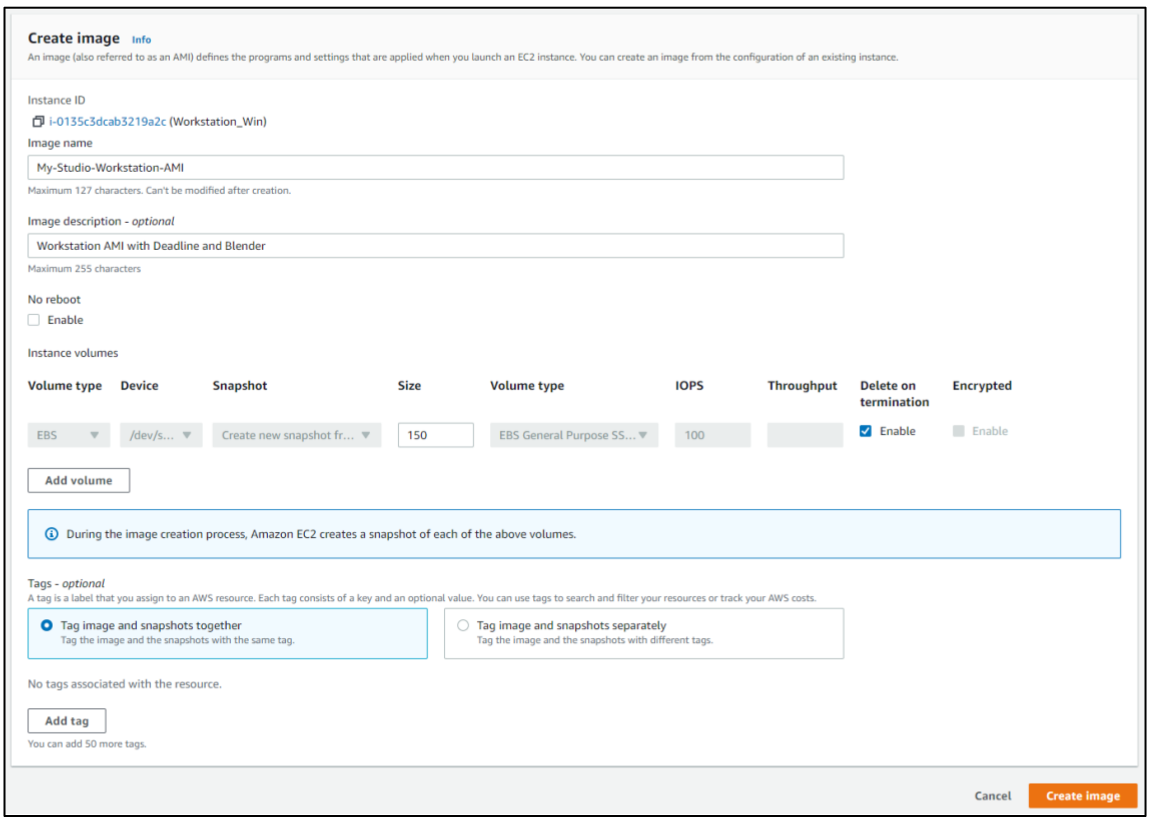
This will shut down your instance after a few processes run. It can take a few minutes, so feel free to grab a coffee (or tea).

Create Workstation AMI

1. Navigate back to the **EC2 Dashboard** and find the Workstation that has just been shut down. If it is not completely stopped yet, wait for that process to finish.



2. Right-click the instance and choose **Images and templates → Create image**
3. Give it an appropriate name (e.g., My-Studio-Workstation-AMI). *Record the AMI name in your cheat sheet.*
4. Give your workstation AMI a description if you want.
5. Increase its storage if necessary.



6. Click **Create image.**

AMI creation can also take 5 to 10 minutes to complete.

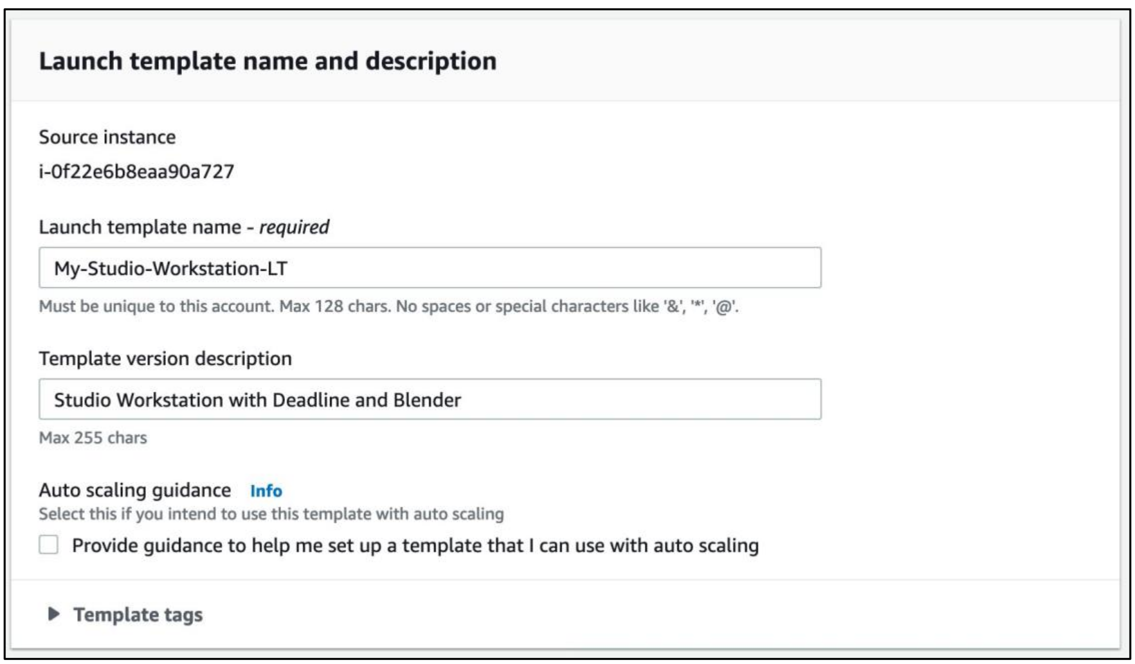
- To see if your AMI is ready click **AMIs** in the left panel. The AMI will need to finish creating before you can create a Launch Template with it.

- While you're waiting, you can also add Tags to your AMI. Again, we recommend creating at least a **Studio** tag and **Name** tag.
- Now is a good time to note the AMI ID in the cheat sheet. You can find the ID by selecting the AMI from the list and looking at the top left of the Details tab.

Create Workstation Launch Template

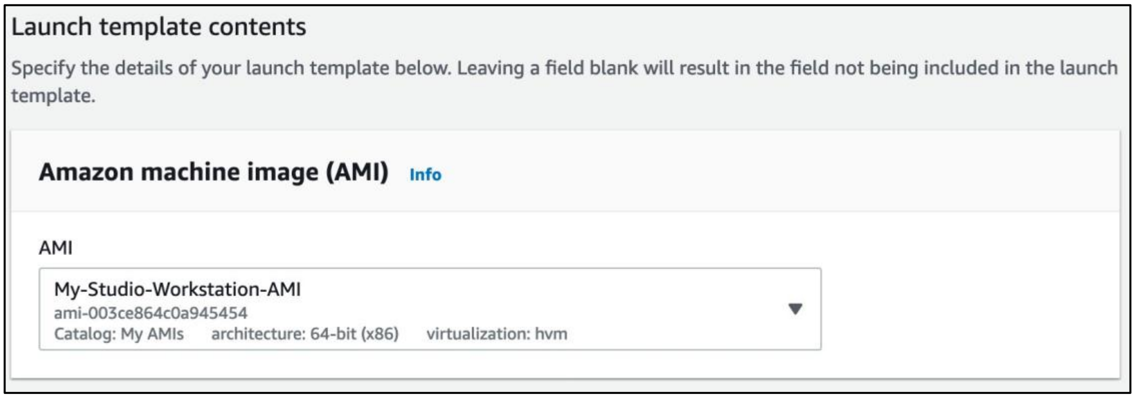
Now that you have an AMI for your workstation, you can create a launch template that will make it really easy to spin up new workstations when you want to collaborate with other artists.

1. Once your new AMI is listed as available, go to **Services** → **EC2** and click **Instances (running)**.
2. Right-click your Workstation instance and choose **Image and templates** → **Create template from instance.**
3. Name your launch template (e.g., My-Studio-Workstation-LT). *Write down the name on your cheat sheet.*
4. Give it a description if you want.



5. You will need to change the **AMI ID** to point to the one that you just created.

6. In the **AMI dropdown**, scroll down to the **My AMIs** section and then select the Workstation AMI (e.g., My-Studio-Workstation-AMI) that you just made. *If necessary, refer to the Workstation AMI ID on your cheat sheet.*

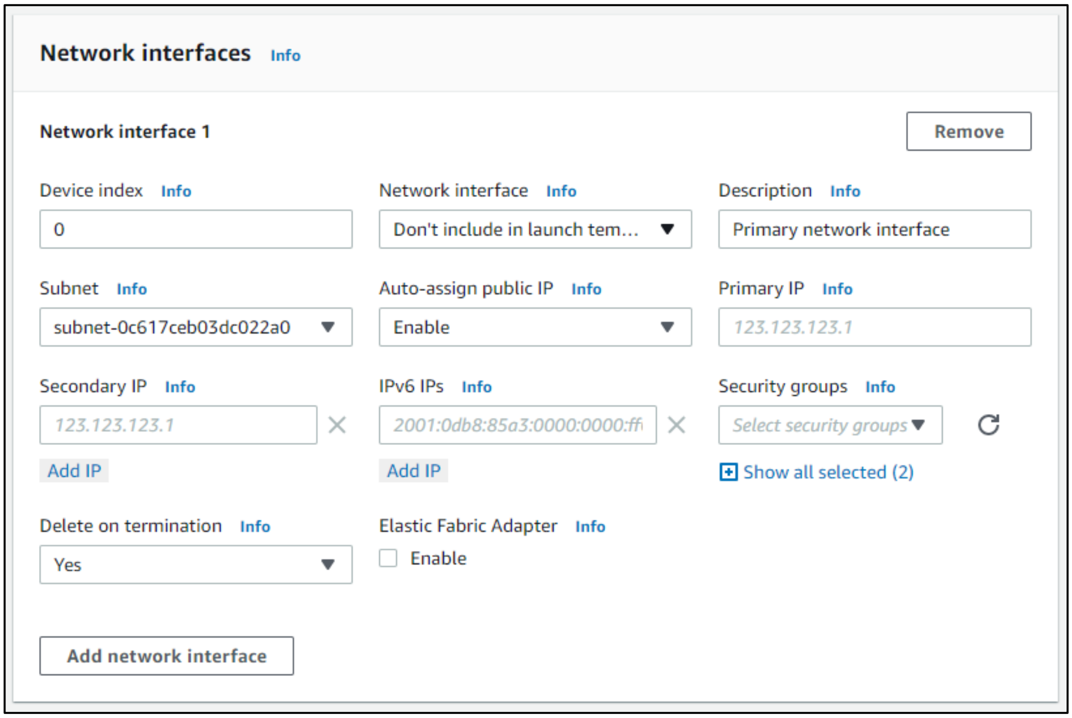


7. Under **Network interfaces,** set **Auto-assign public IP** to **Enable,** otherwise you will not be able to connect to the instances you launch.

8. Also under **Network interfaces**, check that the **Security Group IDs** for both your **Deadline Security Group** (e.g., My-Studio-Deadline-SG) and your **Remote Desktop Security Group** (e.g., My-Studio-Remote-Desktop-SG) are listed. You can find the IDs for both of those security groups on your cheat sheet.

   Click **Show all selected** to view the IDs for the currently selected security groups.

If either security group is missing, click the drop down menu and select the missing group.



9. Click **Create launch template.**

Launch a New Workstation

Now let's test the launch template!

1. Go to **Services** → **EC2** and click **Launch Templates** in the left panel.
2. Select your Workstation launch template (e.g., My-Studio-Workstation-LT). *Refer to your cheat sheet, if needed.*
3. Choose **Actions** → **Launch instance from template.**
4. Select the version (if this is your first time running through the tutorial you'll select version 1).
5. Scroll down to the bottom and click **Launch instance from template.**
6. Go back to **Services** → **EC2,** click **Instances (running).**

When your new instance is done initializing you can log into the instance with your Active Directory username (e.g., mystudio\jason) and password. Note: it may take about 10 minutes for this instance to be available.

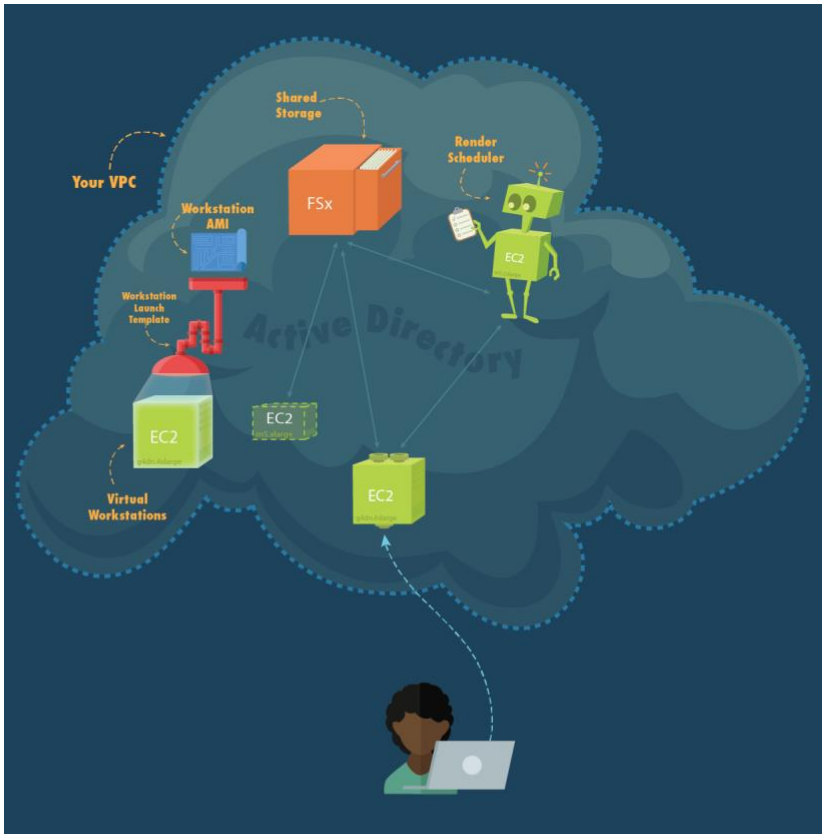
If you are having trouble connecting, wait a few more minutes. If the instance still isn't connecting after 15 minutes, try rebooting the instance to kick-start it.

Note #1: If you do not see the shortcut to launch Blender on the desktop…don't worry, it is still installed. The desktop shortcut we used earlier was created when you logged in as Administrator, so it doesn't exist for the current Active Directory user. Instead, you can use the shortcut that we copied to the Z: drive above. It should be located in **Z:\applications**.

Note #2: Similar to the shortcut, since you originally installed the Deadline Blender submitter as Administrator, you will need to install it again for the current Active Directory user. Follow the directions from earlier in this tutorial to install the submitter.

Once you confirm that your new Workstation_Win instance is running correctly, feel free to terminate the old Workstation_Win instance that you setup at the beginning of this tutorial. It should currently be listed as stopped in the instance list.

Your VPC So Far



In this tutorial, you added a new virtual workstation instance and installed applications on your workstation and your shared storage. You also created a new workstation AMI and launch template that you can use to easily launch new workstations for your artists.

Shut Down Notes

If you are going to continue to the next tutorial then leave your Workstation and Render Scheduler instances running, but if you are going to continue another time then you can stop them both and restart them when you start the next tutorial.

Appendix

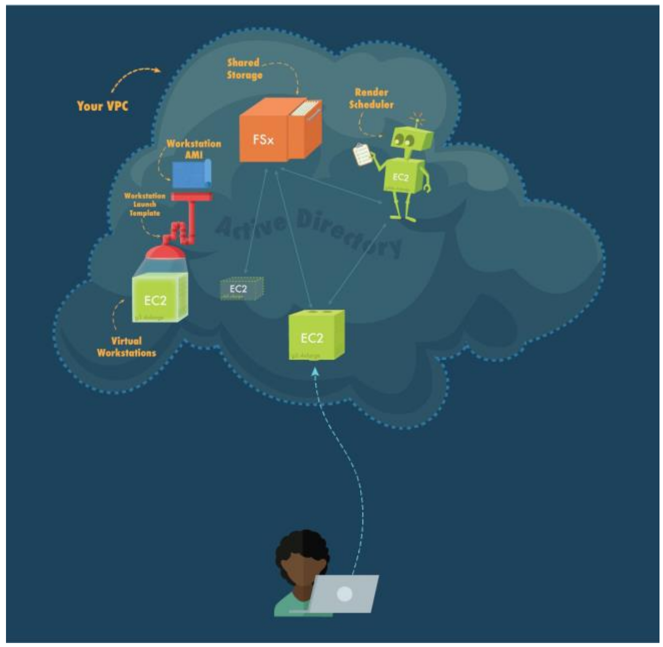
Links to AWS Documentation

- NVIDIA GRID Drivers for G3 and G4 Instances

Tutorial 6. Setting Up a Linux Farm Worker and Spot Fleet Request

Estimated Time to Complete: 3 hours

Next, we are going to create our render farm. This involves launching a Linux instance, mounting our shared storage on it, installing necessary software, creating an AMI, setting up a render fleet, and configuring a Spot request in AWS Thinkbox Deadline so that the render workers will automatically spin up and down as needed. It's a little bit of work, but in the end, you will have a flexible setup for rendering in the cloud.



Startup Notes

If you're coming straight from the last tutorial and already have your Render Scheduler and workstation instances running, you can skip this section and continue straight to launching a Linux instance to set up a worker. But if you stopped your Render Scheduler and workstation instances at the end of the last tutorial, you'll need to start them back up before continuing.

Restarting Your Render Scheduler and Workstation Instances

1. From the **AWS Console** go to **Services→EC2.**
2. Click the **Instances (running)** link near the top of the page.
3. Select your **Render Scheduler** and **Workstation_Win** instances.
4. Click **Actions**, then select **Instance State→Start.**

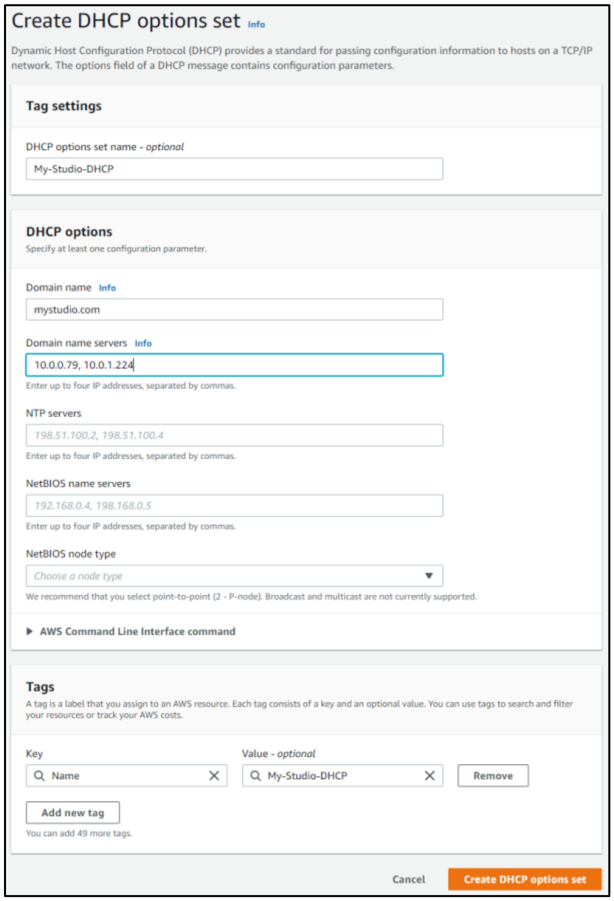
   Important: Once your Render Scheduler is up and running, you will need to log in as Administrator and make sure Deadline Monitor, Deadline RCS, and Deadline Pulse are running.

Update VPC with a DHCP Option Set

In order to connect a Linux instance to the Directory Service, first you need to create a DHCP (Dynamic Host Configuration Protocol) option set. This will help map the Linux instance's IP address to the Directory Service so it can join it.

Create DHCP Settings

1. Go to **Services → VPC.**
2. Click **DHCP Option Sets** in the left panel.
3. Click **Create DHCP options set.**
4. Enter a name for the options set (e.g., My-Studio-DHCP).
5. For **Domain Name** enter your Active Directory DNS Name. (e.g., mystudio.com). As a reminder, you can find yours under the Tutorial 2 section of your cheat sheet.
6. For **Domain Name Servers**, enter the two Active Directory DNS addresses (e.g., 10.0.0.125, 10.0.1.32). *These can be found in the Tutorial 3 section on the cheat sheet.*



7. Leave the rest of the fields blank.

8. Click **Create DHCP options set.**

Make note of the DHCP Option Set ID on your cheat sheet. It should be something like dopt-0857j234nf593hs3u.

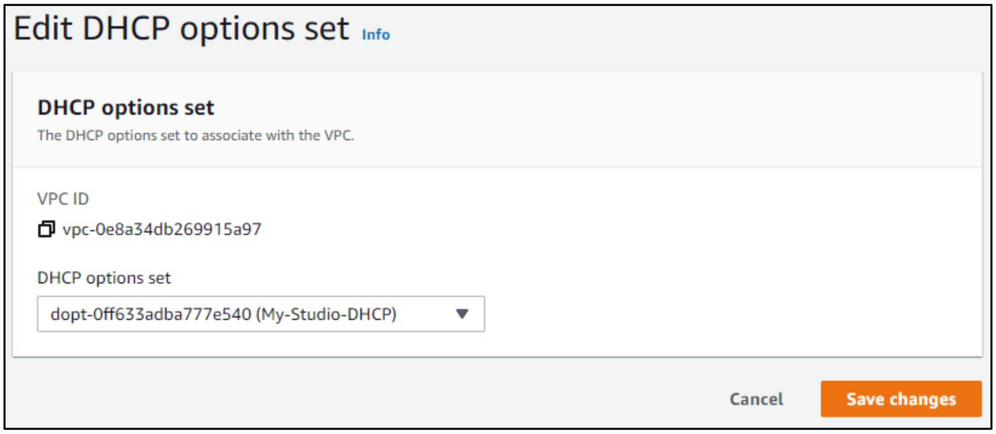
Attach VPC to the DHCP Option Set

1. Select **Your VPCs** in the left panel.

2. Select your studio's VPC (e.g. My-Studio-VPC).

3. Choose **Actions** → **Edit DHCP options set.**

4. From the drop down menu, select the id of the DHCP Option set you just created (e.g., dopt-0857j234nf593hs3u).



5. Click **Save changes.**

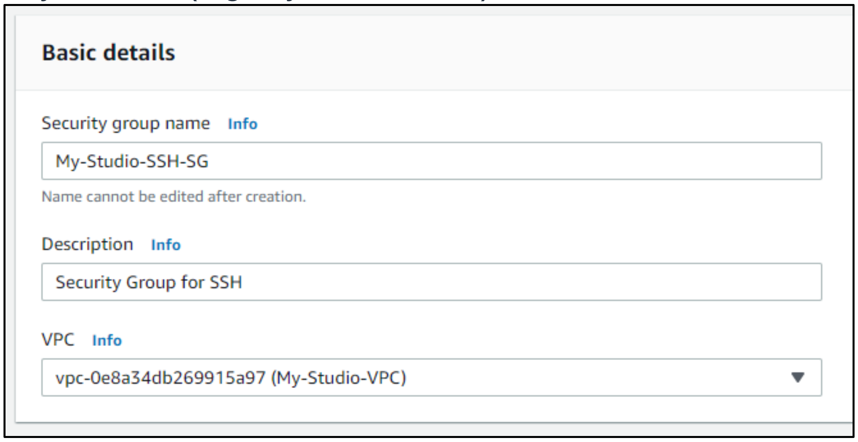
Launch a Linux Instance to Setup a Worker

Create an SSH Security Group

In order to connect to your Linux instance, you'll need to open up an SSH port. This will give you **console** access so you can execute commands. We will create a specific security group allowing for that connection.

1. Go to **Services** → **EC2.**
2. Click **Security Groups** in the left panel.
3. Click **Create security group.**
4. Name your security group (e.g., My-Studio-SSH-SG). *Note the name in your cheat sheet.*
5. Give it a description (required).

6. Choose your **VPC** (e.g. My-Studio-VPC).



7. Add this rule to the **Inbound** tab:

	Protocol	Port	Source	Description
SSH	TCP	22	0.0.0.0/0	SSH - Linux

⚠ *Note*: Like we did for the Remote Desktop security group in Tutorial 1, we are initially opening up your SSH security group to inbound traffic from any IP address. Although this is fine during the initial setup and testing phase, we recommend limiting the range of IP addresses in this security group before working on production content.

8. Click **Create security group**.

Find the security group ID (e.g., sg-0jnb39bfu94hj30da) and write it down on your cheat sheet.

After creating your Security Group, it's recommended to create two new tags: Name, and Studio. By this time, you should be recognizing a pattern - each time you create a new resource, Tag it.

Launch an EC2 Instance

1. Go to **Instances** in the left panel.
2. Click **Launch Instances.**

3. Select **Amazon Linux 2 AMI (HVM), SSD Volume Type** (it should be the first one on the list).

4. Choose **m5.2xlarge** as the instance type.

☐	General purpose	m5.xlarge	4	16	EBS only
☑	General purpose	m5.2xlarge	8	32	EBS only
☐	General purpose	m5.4xlarge	16	64	EBS only

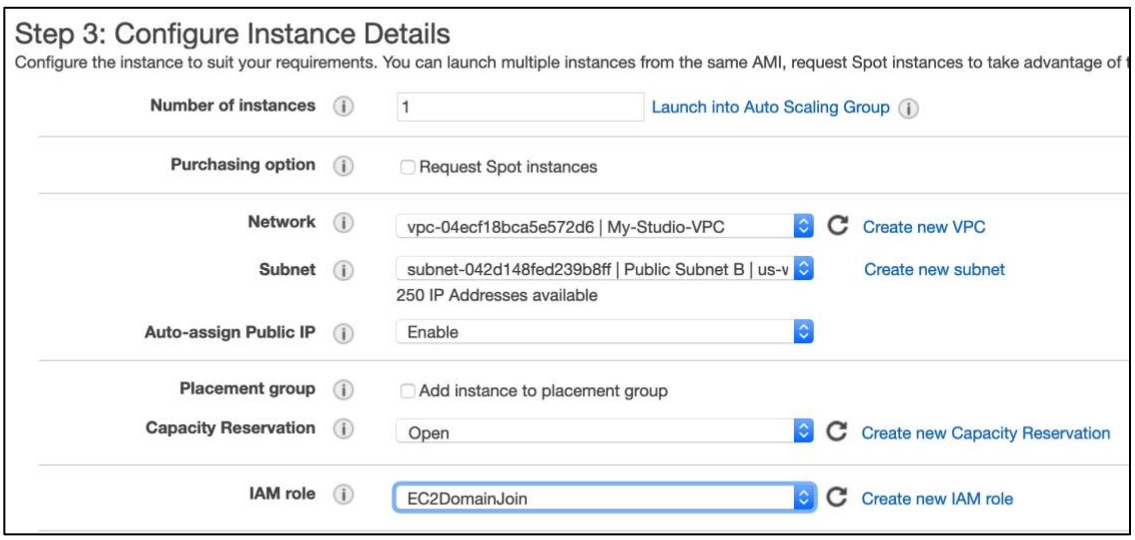
5. Click **Next: Configure Instance Details.**

   a. **Network**: Your VPC (e.g., My-Studio-VPC).

   b. Set subnet, **Public Subnet B.**

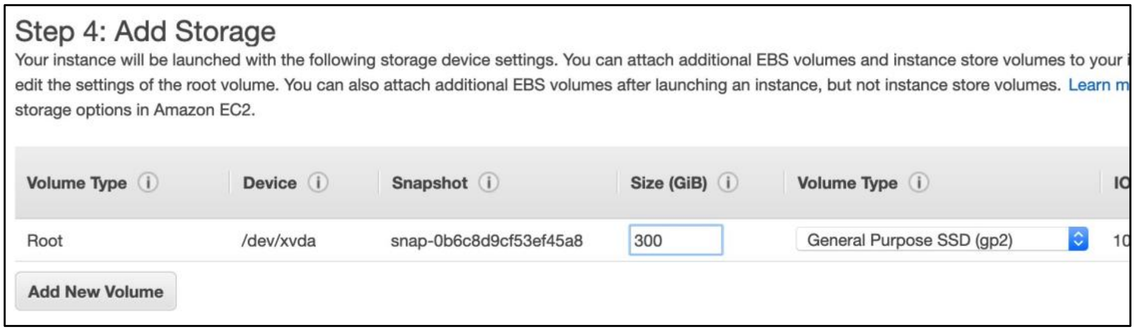
   c. Set Auto Assign Public IPs to **Enable.**

   d. Set IAM role to **EC2DomainJoin.**

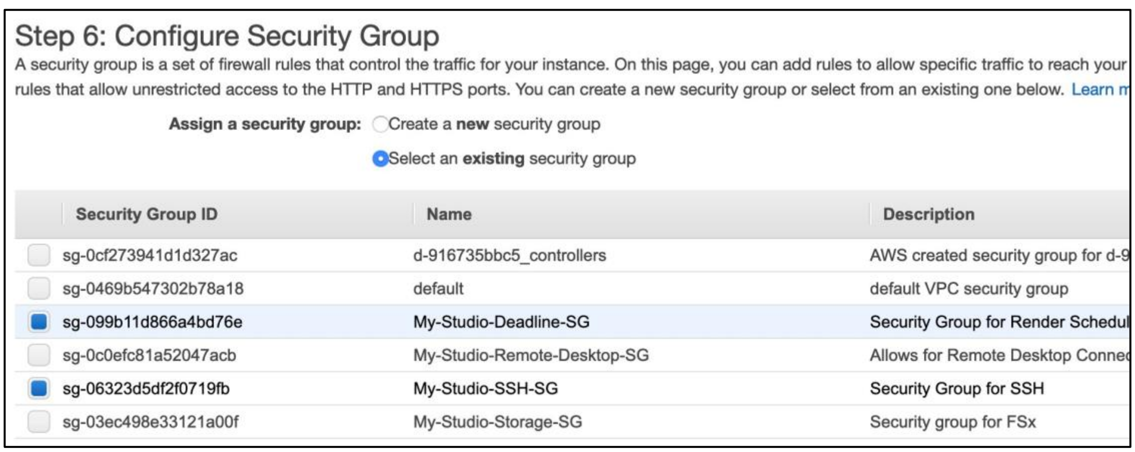


6. Click **Next: Add Storage.**

   a. Set the storage size to **300 GB.**



7. Click **Next: Add Tags** and add these tags as key - value pairs.
   - Key = **Name,** Value = **Worker-Linux.**
   - Key = **Studio,** Value = **My-Studio** (or the name of your studio).
8. Click **Next: Configure Security Group.**
   a. Choose **Select an existing security group.**
   b. Select your **SSH security group** (e.g., My-Studio-SSH-SG) and your **Deadline security group** (e.g., My-Studio-Deadline-SG). *You can find your SSH security group under Tutorial 6 on the cheat sheet and your Deadline security group under Tutorial 4.*



9. Click **Review and Launch** at the bottom of the window.
10. Review the settings for your instance, then click **Launch** at the bottom of the window.
11. Choose the key pair file you have been using (e.g., mystudio-keypair.pem). *This is located under Tutorial 1 on your cheat sheet.*

12. Click **Launch Instances.**
13. Click **View Instances** at the bottom right of the screen.

Select your Worker-Linux instance from the list and write down the Public DNS on your cheat sheet (e.g., ec2-54-187-92-7.us-west-2.compute.amazon.aws.com).

Wait for your Linux instance to initialize and its status checks to switch to 2/2 checks passed, then continue to the next step.

Connect to Active Directory

There are a couple of ways you can SSH into your Linux worker instance. If you have a Mac and have easy access to the keypair you can SSH from the Terminal. If you are on Windows, it may be easier to just use a browser-based SSH connection (this will also work on a Mac).

SSH into Linux Worker Instance - Mac

1. On your local machine open up Terminal.
2. **CD** to the location of your key pair file (e.g. mystudio-keypair.pem).
3. Run this command to change the permissions:

```
chmod 400 mystudio-keypair.pem
```

Make sure you **update the red text** above with the ***name of your key pair***

4. Make sure your instance is done initializing and then run the following command to use SSH to connect:

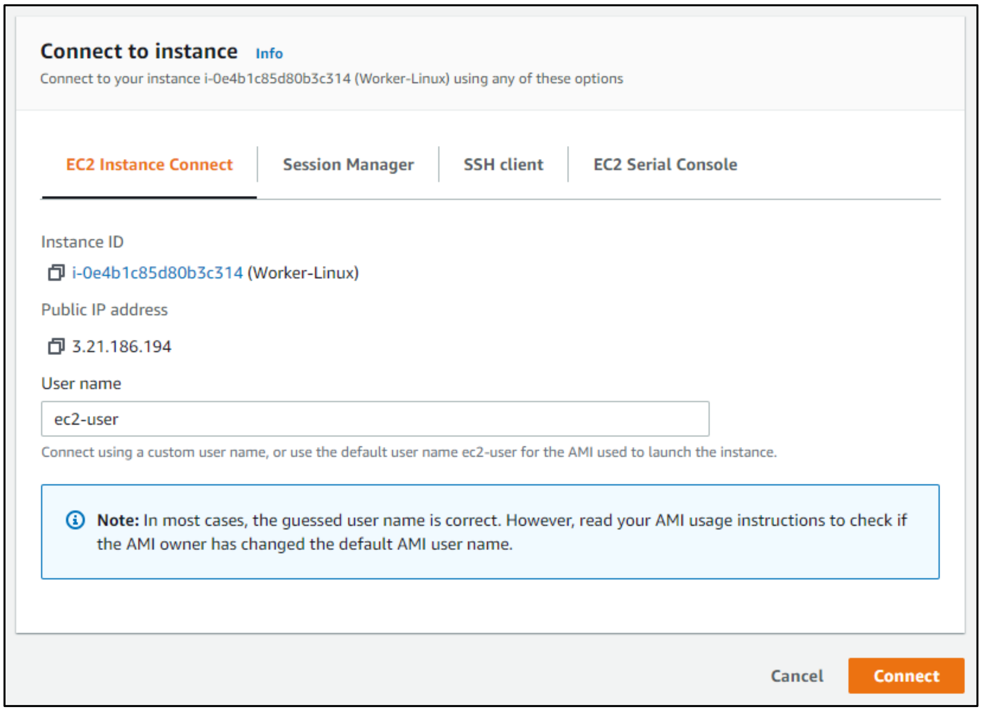
```
ssh -i mystudio-keypair.pem ec2-user@ec2-xx-xxx-xx-x.us-west-
2.compute.amazonaws.com
```

Make sure you **update the red text** above with the ***name of your key pair*** and the ***Public DNS*** of your ***Linux Worker instance.***

Now you are connected to your remote worker instance.

SSH into Linux Worker Instance - Browser-based SSH

1. In the instance list in the AWS Console, select your worker instance and click the **Connect** button.
2. Choose **EC2 Instance Connect.**
3. Leave the **User name** as **ec2-user.**



4. Click **Connect.**

A window will open with access to your instance. Note: The brower-based SSH connection will automatically timeout after a few minutes of inactivity and the window will become unresponsive. If that happens to you, close the connection window and open a new one.

Join Active Directory

1. First, run the command below to update the instance to the latest software:

```
sudo yum -y update
```

2. Next, run these commands to install some specific tools needed to connect to Active Directory:

```
sudo yum -y install sssd realmd krb5-workstation
sudo yum –y install samba-common-tools
```

3. Finally, run the command below to join Active Directory from your Linux Worker. Make sure you **update the red italicized text** to match the **name of your Active Directory**. You'll be replacing the two instances of "mystudio.com" with your Active Directory's DNS Name.

   Note: You'll only have to run these commands once. After this, we'll be creating a Launch Template to connect to your Active Directory.

```
sudo realm join -U Admin@mystudio.com mystudio.com --verbose
```

   You should receive a message that says: **Successfully discovered: mystudio.com.**

4. Type in the password for your Active Directory and you will see a message that says: **Successfully enrolled machine in realm.**

Now we're going to update the sshd_config to allow password authentication.

5. Enter this command:

```
sudo vi /etc/ssh/sshd_config
```

6. To find the correct line type: **/PasswordAuthentication no** and press **<enter>**

   If you are unfamiliar with **vi**, the **/** key puts you in **search** mode. So, the text above will search the file for the exact text of "PasswordAuthentication no" and take you to that line.

7. Move the cursor over using the arrow keys to the word **no.**

8. Type **cw.**

   This puts you in **Change Word** mode.

9. Type the word **yes.**

   Replaces the word **no** with **yes.**

10. Press **esc.**

    Takes you out of **Edit** mode.

11. Type **:wq** and press **<enter>.**
    - ":wq" **Writes** the file and then Quits vi.
    - [Here's]() a cheat sheet for vi commands.
12. Next, restart the sshd service with this command:

```
sudo systemctl restart sshd.service
```

13. Now restart the instance by entering this command:

```
sudo reboot
```

Your instance is now restarting. Note: If you connected using the EC2 Instance Connect browser-based connection, your window may appear to freeze up after running the "sudo reboot" command. Close the window before trying to reconnect.

Wait a minute or two and reconnect to it by using the same method you used before:

MAC

```
ssh -i mystudio-keypair.pem ec2-user@54-187-92-7
```

PC

1. In the AWS Console for EC2, select your worker instance and click the **Connect** button.
2. Choose **EC2 Instance Connect.**
3. Leave **User name** as **ec2-user.**
4. Click **Connect.**

   A window will open with access to your instance.
5. In preparation for the next step, it is recommended to perform a mkdir in /mnt/ to create a directory for the FSx mount.

```
sudo mkdir /mnt/studio
```

Mount FSx File System

1. First run this command to install some utilities:

```
sudo yum -y install cifs-utils
```

2. Next run the command below and enter the Active Directory Admin password when prompted. Make sure to replace MYSTUDIO with your **Active Directory's DNS Name** and type it in all capital letters. *You can find that information on your cheat sheet under Tutorial 2.*

```
kinit Admin@MYSTUDIO.COM
```

3. Finally, run the command below to mount your FSx drive. Again, make sure you update the red italicized text to match your **Active Directory DNS Name** and the **FSx DNS Name**. *You can find this information under Tutorial 2 and Tutorial 3 on your cheat sheet.*

```
sudo mount -t cifs -o user=Admin@MYSTUDIO.COM,cruid=$(id -
u),uid=$(id -u),sec=krb5 //fs-
0c0b4fab1db1b9a0e.mystudio.com/share /mnt/studio -o vers=3.0
```

Note which text is capitalized. These commands ARE case sensitive.

4. To test and see if it mounted correctly, run **ls** of the directory.

```
ls /mnt/studio
```

If you see a few folders returning back, then you've connected correctly!

Note: Whenever you start a new instance or stop and then start an existing one, you'll need to remount your FSx file system. However, later in this tutorial we'll be adding user data to a launch template so this step happens automatically for the farm workers.

Install Deadline Client

1. Run this command:

```
sudo yum -y install lsb
```

2. If asked any questions type **y** and hit enter.
3. Next, navigate to the Thinkbox installers directory

```
cd /mnt/studio/installers/thinkbox
```

4. Run this command to navigate to un-tar the tar file. Note: You may need to replace the version number (e.g., 10.1.7.4) with number of the version that you are using.

```
tar -xvf Deadline-10.1.17.4-linux-installers.tar
```

5. To install the Deadline Client run this command. Note: Again, the Deadline client may be a different version than what is specified here. Adjust as required.

```
sudo ./DeadlineClient-10.1.17.4-linux-x64-installer.run
```

6. Give these answers to the prompts:
   - Press **Enter** to continue (you will need to do this several times).
   - Type **y** and hit enter to agree to the license agreement.
   - Leave Installation Directory at **default** (hit Enter).
   - Set full read/write access for files for all users: **N.**
   - Select Installation Type: **1** to choose **Client.**
   - Connection Type: **1** to choose **Remote Connection Server**
   - Server Address: **[Render Scheduler Private IP Address]**:4433 (e.g., 10.0.0.219:4433). You can find your Render Scheduler's Private IP Address under Tutorial 4 on the cheat sheet.
   - RCS TLS Certificate: (Blank).
   - Certificate Password: (Blank).
   - Launch Worker: **Y.**
   - Install Launcher as Daemon: **Y.**
   - User Name: **root**.
   - Do you wish to continue: **Y.**

- Block Auto Update Override: **1** to choose to **Block auto upgrade**.
- Do you want to continue: **Y.**
  - Wait for the installer to finish. You will see a message that the launcher service has been stopped and then started again.

7. If necessary, press **<enter>** in the SSH session to get the command line back and then navigate to the Deadline10 directory:

```
cd /var/lib/Thinkbox/Deadline10/
```

8. Next, run this command to edit the deadline.ini file:

```
sudo vi deadline.ini
```

9. Put your cursor over the word **False** next to ProxyUseSSL.
10. Press **cw** on your keyboard.
11. Type **True.**
12. Edit the line “ProxySSLCertificate=” so it says (all on one line):

```
ProxySSLCertificate=/mnt/studio/app_env/thinkbox/DeadlineCertificat
es/Deadline10RemoteClient.pfx
```

13. Press **esc.**
14. Type **:wq** and press **Enter.**

Install Blender

1. Navigate to **/mnt/studio/installers/blender:**

```
cd /mnt/studio/installers/blender
```

2. Then extract Blender by running this command:

```
tar -xvf blender-2.93.1-linux-x64.tar.xz
```

<u>Note #1</u>: You may need to change the Blender version (e.g., 2.93.1) to match the version of Blender that you are installing.

<u>Note #2</u>: You may get some symlink errors when running the Blender installer from /mnt/studio, but don't worry you can ignore them.

3. Rename the folder called **blender-2.93.1-linux64** to **blender**. <u>Note</u>: Again, you will need to change the version number to match your version of Blender

```
mv blender-2.93.1-linux-x64 blender
```

4. Move the folder to /usr/local/Blender (the location Deadline expects it to be).

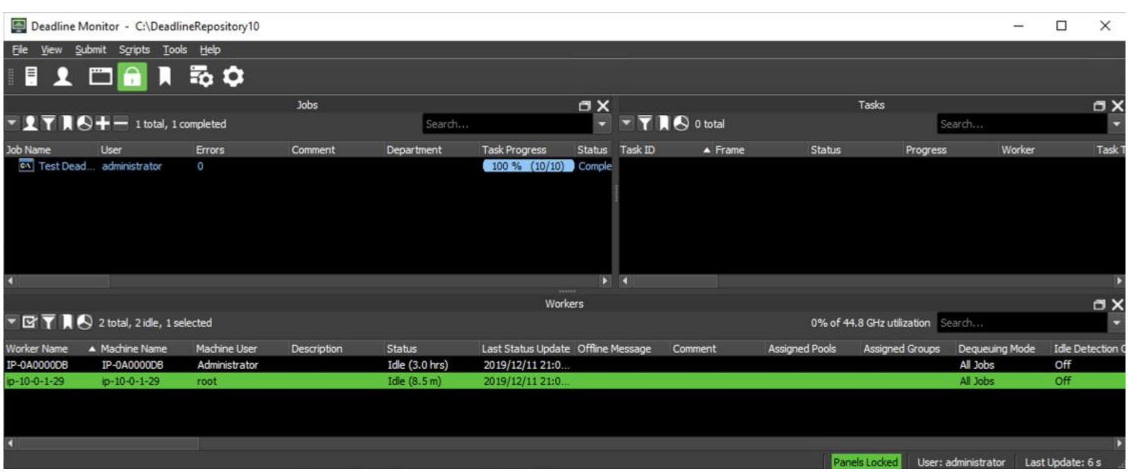
```
sudo mv /mnt/studio/installers/blender/blender /usr/local/Blender
```

Check Deadline Monitor

A great way to check and make sure that your instance has gotten Deadline installed correctly is to jump over to your Render Scheduler instance and look to see if the new worker you created is visible in the list of Workers. If you see it there, then you know Deadline installed correctly, and your worker can see the Render Scheduler.

You may see your Render Scheduler still listed in the worker list, with Machine User set to Administrator.

Your Linux worker will be listed under that with the Machine User set to **root.**



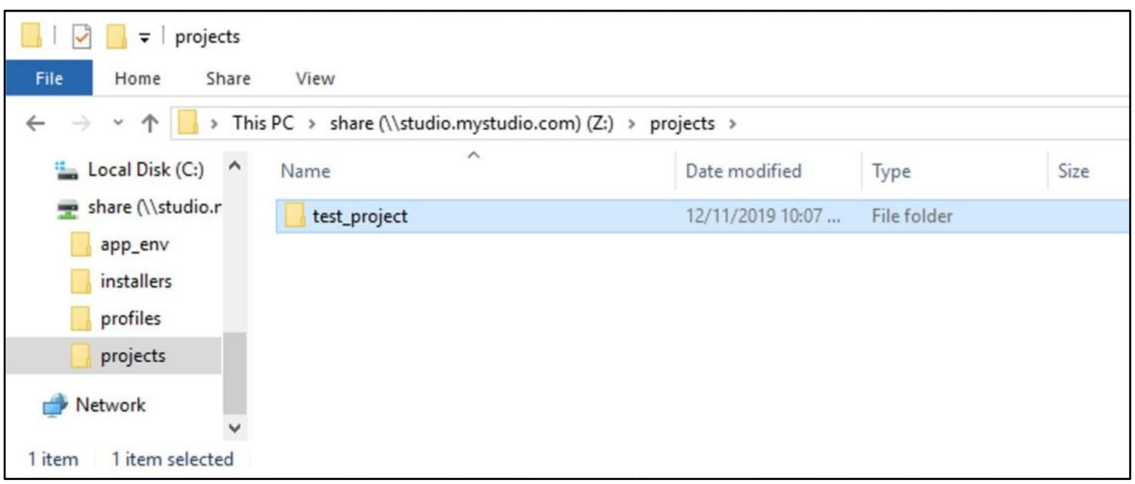
Render Something to Test the Setup

Now that you've got most of the pieces together, it's a good idea to test the setup to make sure you can submit a render from your workstation, have the Render Scheduler pick it up, and run the render on your Worker. To do this, make sure you have three instances running: your Workstation, the Render Scheduler, and your Linux Worker. If any of those aren't running, get them started.

Set Up a Shot to Render

This will involve using the workstation to set up a shot to do a test render.

1. Log into your **Windows Workstation** using one of the user accounts you have created (e.g., mystudio\jason).
2. Create a location on your **Z:** drive where you can store projects (ex: Z:\projects).
3. Now add a **test_project** folder in **Z:\projects**.



4. Launch **Blender** by running the Blender shortcut located in **Z:\applications.**
5. The Deadline Blender submitter needs to be re-installed for each user, so you should do that now:
   a. Go **Edit → Preferences...** (or **File→Preferences** - for Blender 2.79 & earlier)
   b. Click **Add-Ons** in the left panel.

c. Click **Install...**

d. Navigate to **Z:\installers\thinkbox\submission\Blender\Client .**

e. Choose **DeadlineBlenderClient.py .**

f. Click **Install Add-on.**

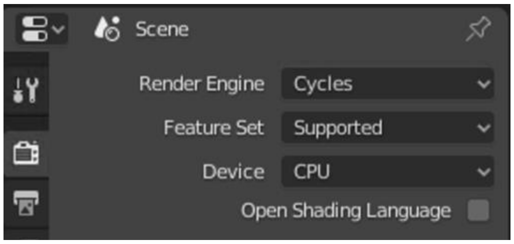
g. Click the checkbox next to **Render: Submit Blender to Deadline** add-on and close the Preferences window.

6. For your test render, you can just use the default blender file with a cube and camera (This is just a test to make sure everything is working correctly. If you'd like to make a super fancy file, you're more than welcome to).

7. Set your output settings

a. In the **Properties** Panel on the right, click **Render Properties** (the icon looks like the back of an SLR camera).

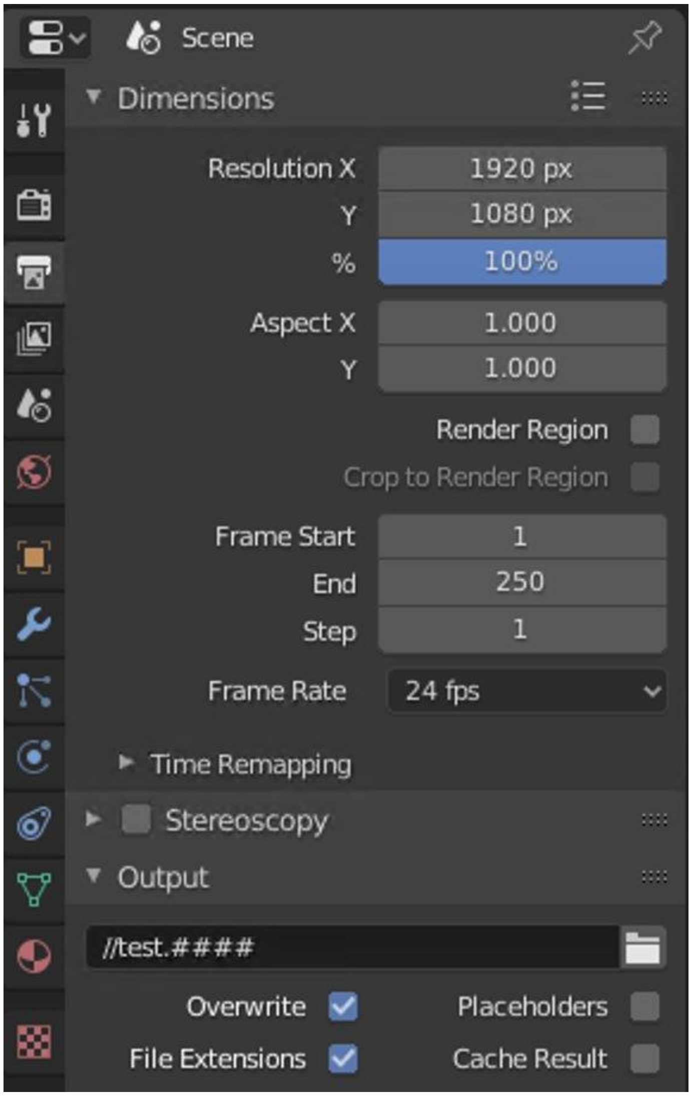
b. For **Render Engine** choose **Cycles.**



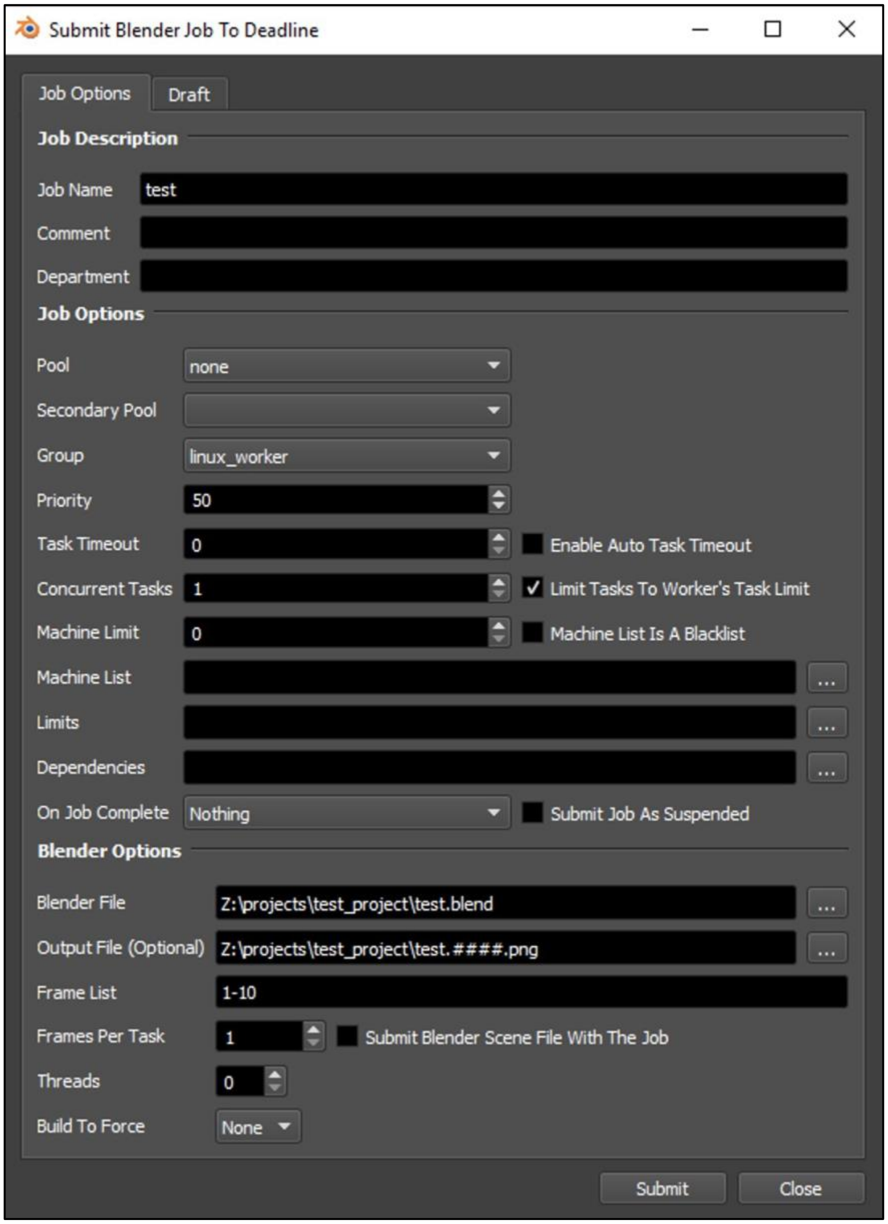
c. Click **Output Properties** (the icon looks like an inkjet printer).

d. Under **Output** change the value to **//test.####** (this will ensure the images are written to the **test_project** folder).

e. Now **save** the file as **Z:\projects\test_project\test.blend .**



8. Submit your render
    a. Choose **Render→Submit to Deadline.**
    b. It may take a few moments for the submit window to appear.
    c. Set the **Group** to **linux_worker.**
    d. Set the **Frame List** to **1-10.**
    e. Make sure the **Blender File** and **Output File** is set properly.



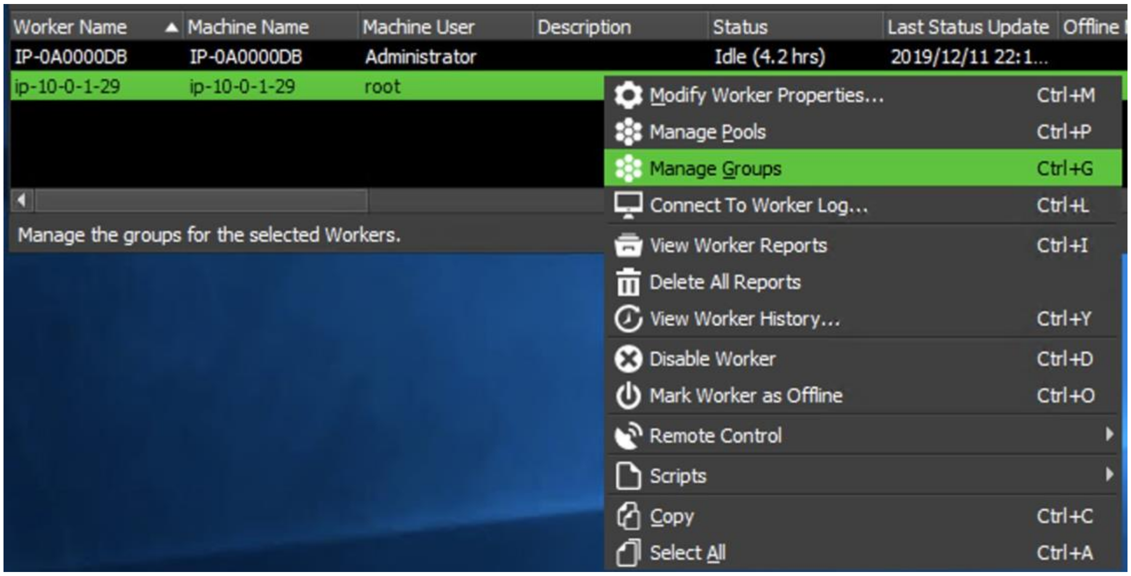
f. Click **Submit.**

Check the Render Scheduler

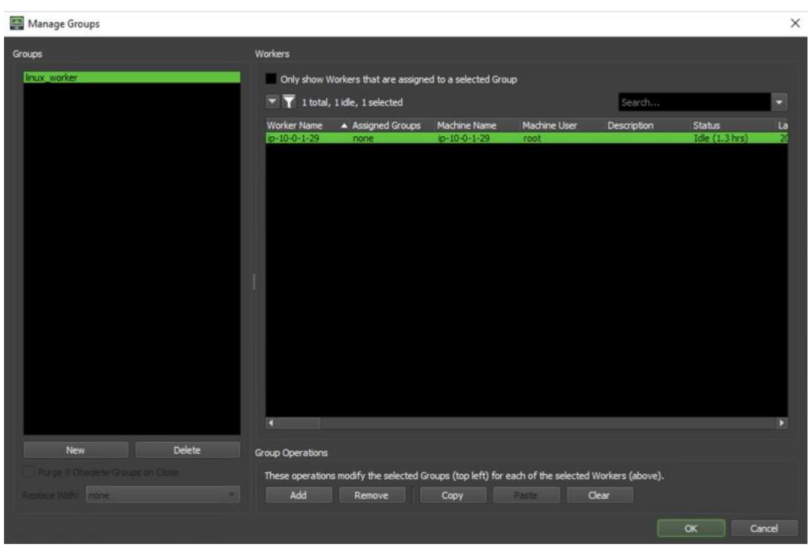
1. Log into your **Render Scheduler** as **Administrator**.

   To get the **worker** to pick up the job, you need to add the **linux_worker** group to it.

2. Make sure you have super user mode enabled by choosing **Tools→Super User Mode**.
3. Select the **worker** in the **list of workers** at the bottom of the monitor window.
4. Right-click and choose **Manage Groups.**



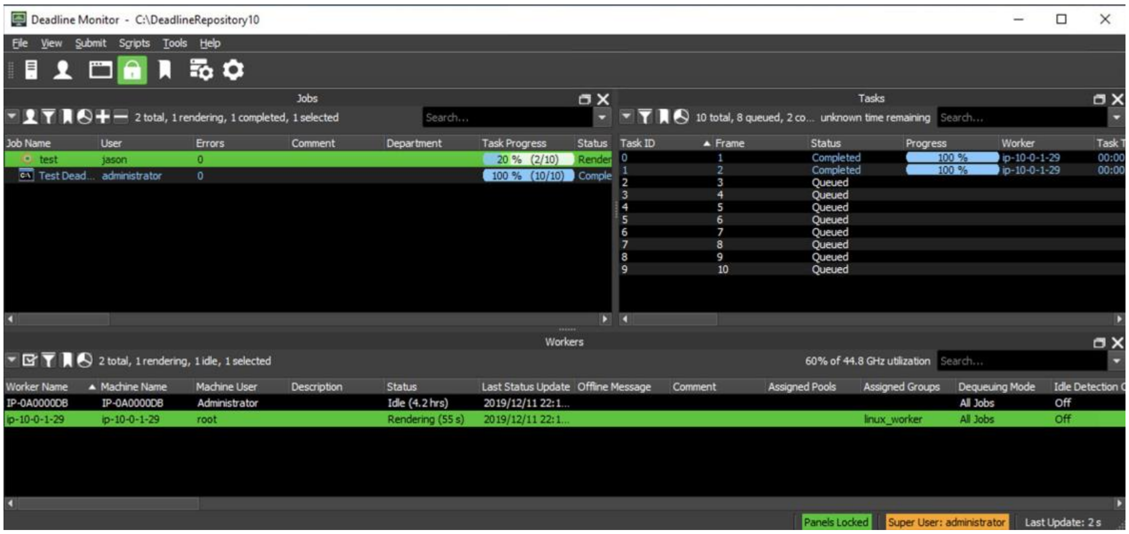
5. Select **linux_worker** on the left, and the machine on the right and click **Add** to add the **linux_worker** group to the machine.



6. Click **OK**.

Once your worker has gotten the **linux_worker** group added to it, it should begin picking up jobs.

You should see your render going in the **Deadline Monitor.**



If the tasks aren't ending up as **Completed,** but instead show up as **Queued** and are colored pink, then there's probably an error. Don't be alarmed, this can happen quite easily if things aren't sent correctly. The best thing to do is double-click the task to bring up the logs. This will let you investigate to find out what's wrong. Some common issues we've found:

- **Error: Blender render executable was not found in the semicolon separated list**

  This means that Deadline can't find the location of the Blender application on the worker. To see where Deadline is looking for the render application, go **Tools→Configure Plugins.** This is a list of all the applications that can be run from Deadline.

a. Click **Blender** and view the list of potential locations for the Blender executable. As you can see, one of the locations it's looking is: **/usr/local/Blender/blender**

b. You can add a new location in here if you installed Blender in a different location. If you think you installed Blender in one of these locations, the best thing to do is log into the **worker** instance and check to see where you installed Blender, then add the location to this list.

c. Re-submit the render and see if it solved the problem.

- **Display Error**

  This means you didn't change the renderer in your Blender file from Eevee to Cycles. Change the renderer and save the file before resubmitting.

- **Render Scheduler In Worker List**

  If you see two workers listed in your monitor, one of them is your Render Scheduler. Right-click the the worker name that starts with IP-0A and choose **Disable Worker.**

  This will ensure that the Linux worker picks up your job.

  To permanently remove your Render Scheduler from the list of workers, connect to it as Administrator and close the DeadlineWorker application that is running.

- **Error: The configured Client Certificate ('/mnt/studio/app_env/thinkbox/DeadlineCertificates/Deadline10RemoteClient.pfx') does not exist.**

  You may run into this if you pause part of the way through this tutorial and leave your Worker-Linux instance running for a few days. In that case, you should connect to Worker-Linux and rerun these commands from before. Make sure to replace the red italicized text to match your **Active Directory DNS Name** and the **FSx DNS Name**. *You can find your Active Directory DNS Name and FSx DNS Name under Tutorial 2 and Tutorial 3 on your cheat sheet.*

As before, also note which text is capitalized. These commands ARE case sensitive.

```
kinit Admin@MYSTUDIO.COM
```

```
sudo mount -t cifs -o user=Admin@MYSTUDIO.COM,cruid=$(id -
u),uid=$(id -u),sec=krb5 //fs-
0c0b4fab1db1b9a0e.mystudio.com/share /mnt/studio -o vers=3.0
```

After running the commands above, you can disconnect from your Worker-Linux and resubmit the render.

If you have a successful test, then congratulations! It's now time to start scaling your render farm by creating an **AMI** (Amazon Machine Image) of the worker so you can create hundreds of duplicate instances, and then set up auto-scaling in order to grow and shrink your farm as necessary.

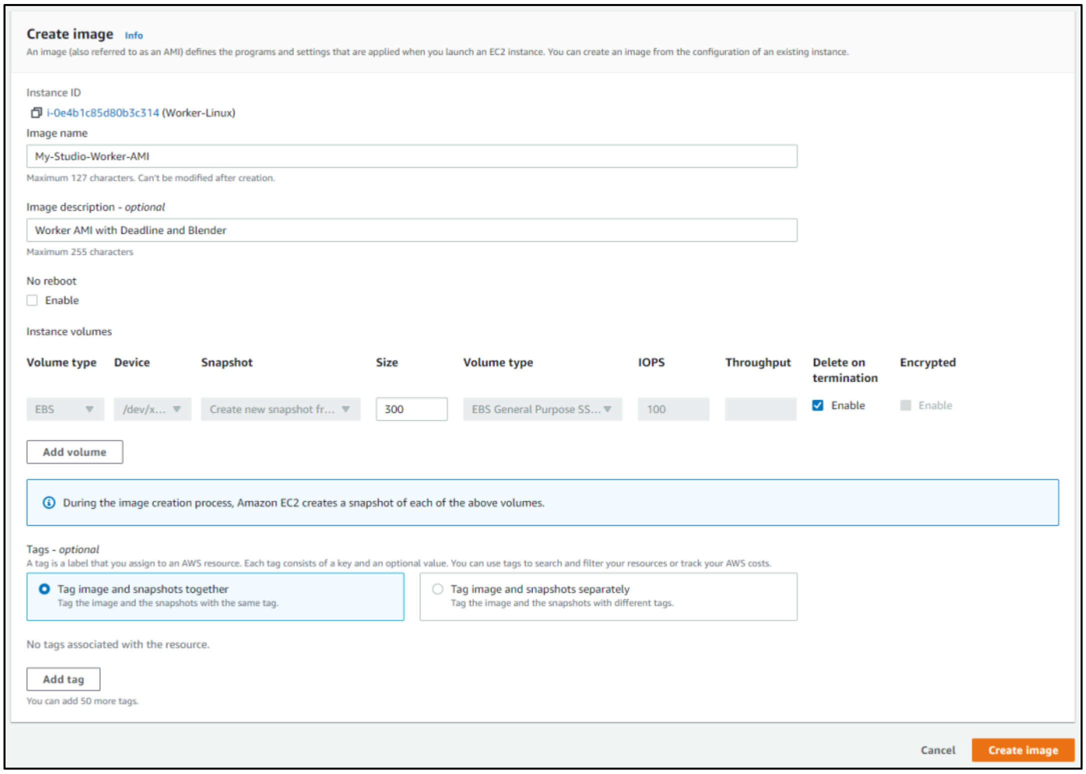
Create a Scalable Workforce

Create an AMI

Shut down your Linux worker instance and create an Image of it.

1. Go to **Services→EC2.**
2. Click **Instances**.
3. Right-click your **Worker-Linux** instance and choose **Stop instance.**
4. Once stopped, right click it again and choose **Image and templates→Create image.**
5. Enter a name for your image (e.g., My-Studio-Worker-AMI) and a description for your image.
6. Check that the **size** of your volume is set to **300 GiB.**

7. Click **Create image.**



8. In the navigation pane on the left, under Images, select **AMIs**.
9. Find the name of your Linux Worker AMI in the list and note the AMI ID as well. *Write down both of these in your cheat sheet.*
10. Add appropriate **Name** and **Studio** tags.

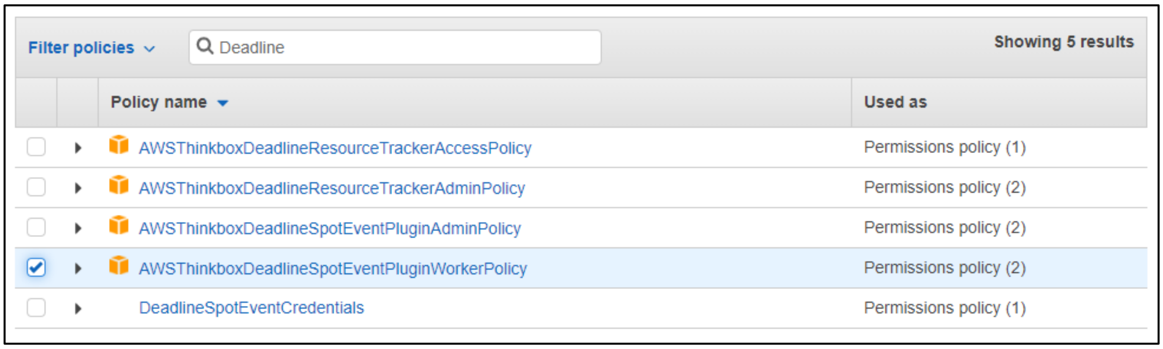
While you are waiting for the AMI to become available, continue to the next section to create a new IAM policy.

Create a Deadline worker role

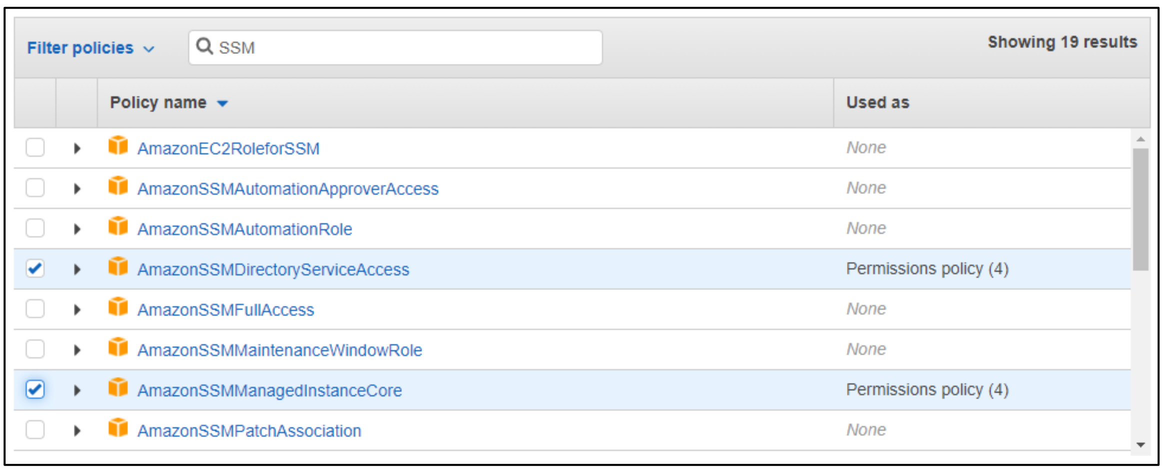
Next you will create a new IAM role that will give your render workers permissions to terminate themselves, determine what Group they are part of, and report their status.

1. Go to **Services** → **IAM** and click **Roles**
2. Click the **Create role** button.

3. Under **Select type of trusted entity**, choose **AWS service**.
4. Under **Choose a use case**, choose **EC2** and then choose **Next: Permissions.**
5. Search for **Deadline.**
6. Click the check box next to **AWSThinkboxDeadlineSpotEventPluginWorkerPolicy.**

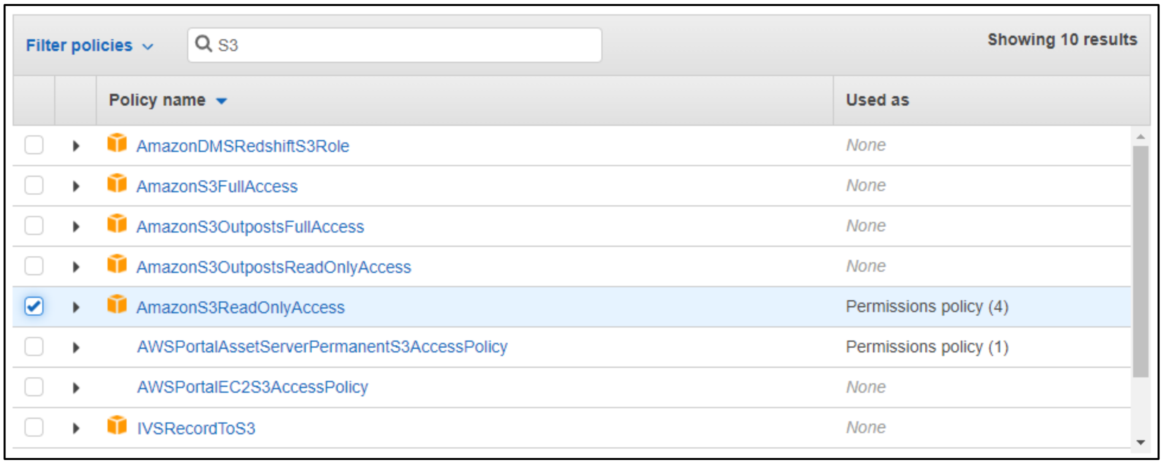


7. Search for **SSM.**
8. Click the check boxes next to **AmazonSSMDirectoryServiceAccess** and **AmazonSSMManagedInstanceCore.**



9. Search for **S3.**

10. Click the check box next to **AmazonS3ReadOnlyAccess.**



Filter policies | S3 | Showing 10 results

Policy name	Used as
AmazonDMSRedshiftS3Role	None
AmazonS3FullAccess	None
AmazonS3OutpostsFullAccess	None
AmazonS3OutpostsReadOnlyAccess	None
AmazonS3ReadOnlyAccess	Permissions policy (4)
AWSPortalAssetServerPermanentS3AccessPolicy	Permissions policy (1)
AWSPortalEC2S3AccessPolicy	None
IVSRecordToS3	None

11. Search for **Secrets.**

12. Click the check box next to **SecretsManagerReadOnly.**

Filter policies | Secrets | Showing 2 results

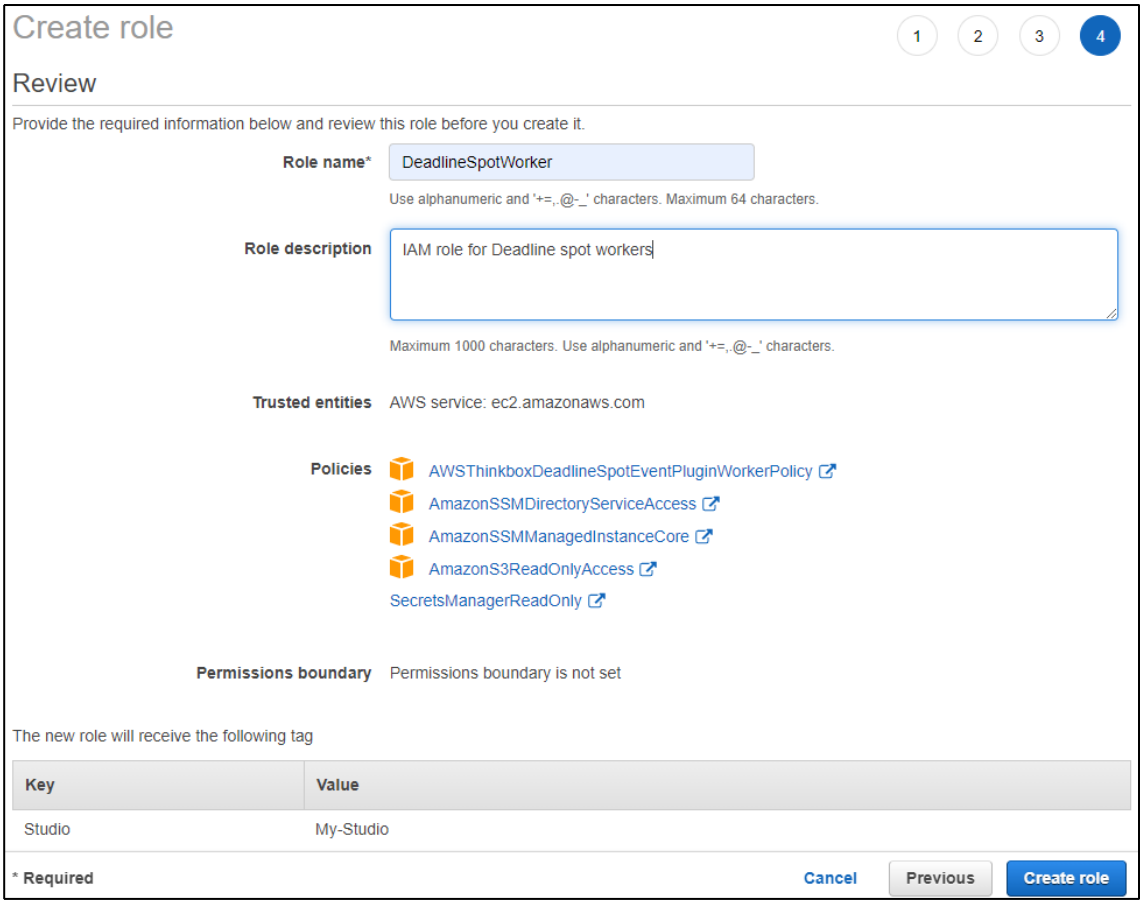
Policy name	Used as
SecretsManagerReadOnly	Permissions policy (6)
SecretsManagerReadWrite	None

13. Click **Next: Tags**

14. For **Key** enter **Studio** and for **Value** enter the name of your studio from above (e.g., My-Studio). Then, click **Next: Review.**

15. For **Role name**, enter **DeadlineSpotWorker**

16. (Optional) For **Role description**, enter a description.

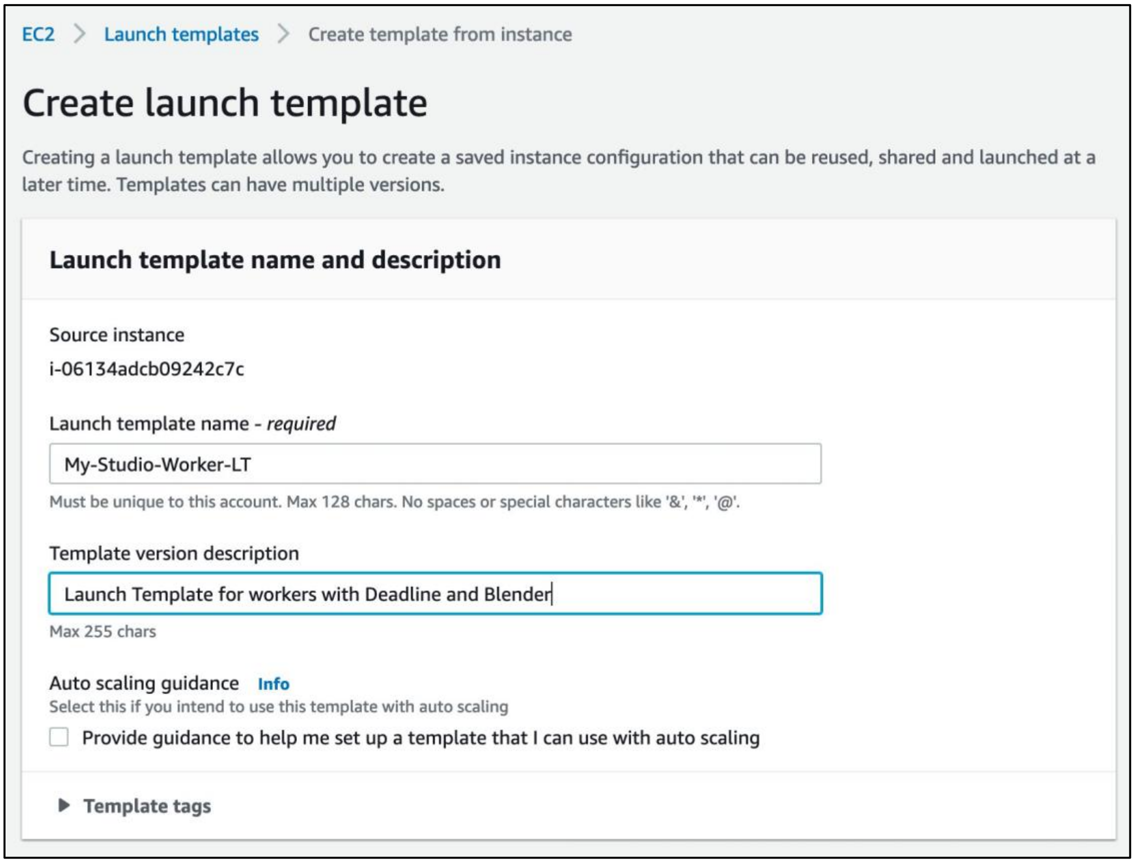


17. Choose **Create role**

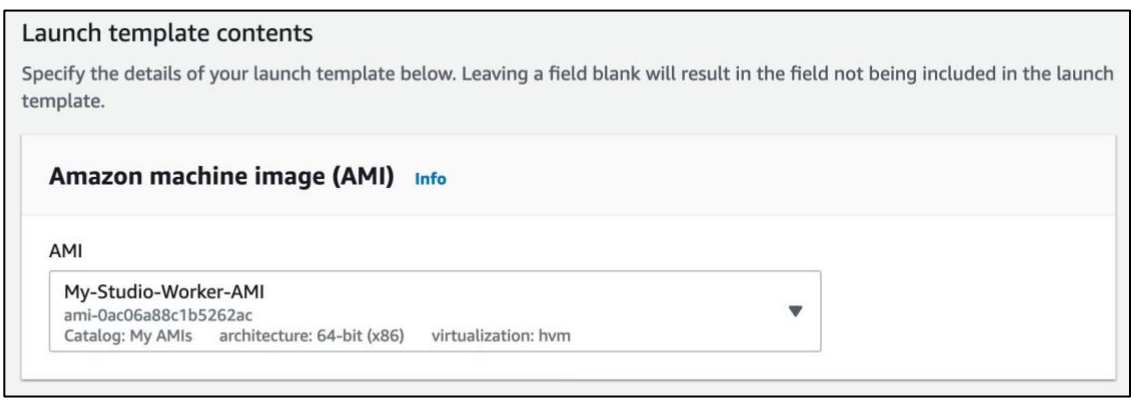
Create a Launch Template

After the Linux Worker AMI has been created, you'll want to create a launch template from that AMI.

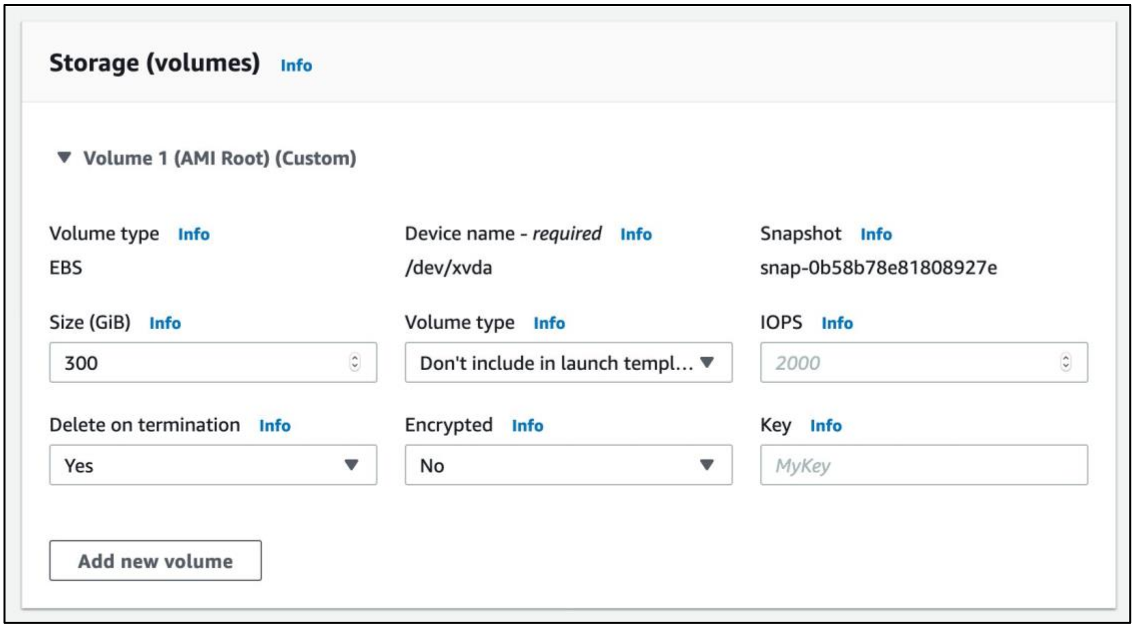
1. Go to **Services** → **EC2** and click **AMIs.**
2. Find the Linux Worker AMI you created earlier. Once the **Status** changes from pending to **available**, in the left panel click **Instances.**
3. Right-click your Linux Worker and choose **Image and templates** → **Create template from instance.**
4. Name your launch template (e.g., My-Studio-Worker-LT).
5. Give your launch template a description if you want.



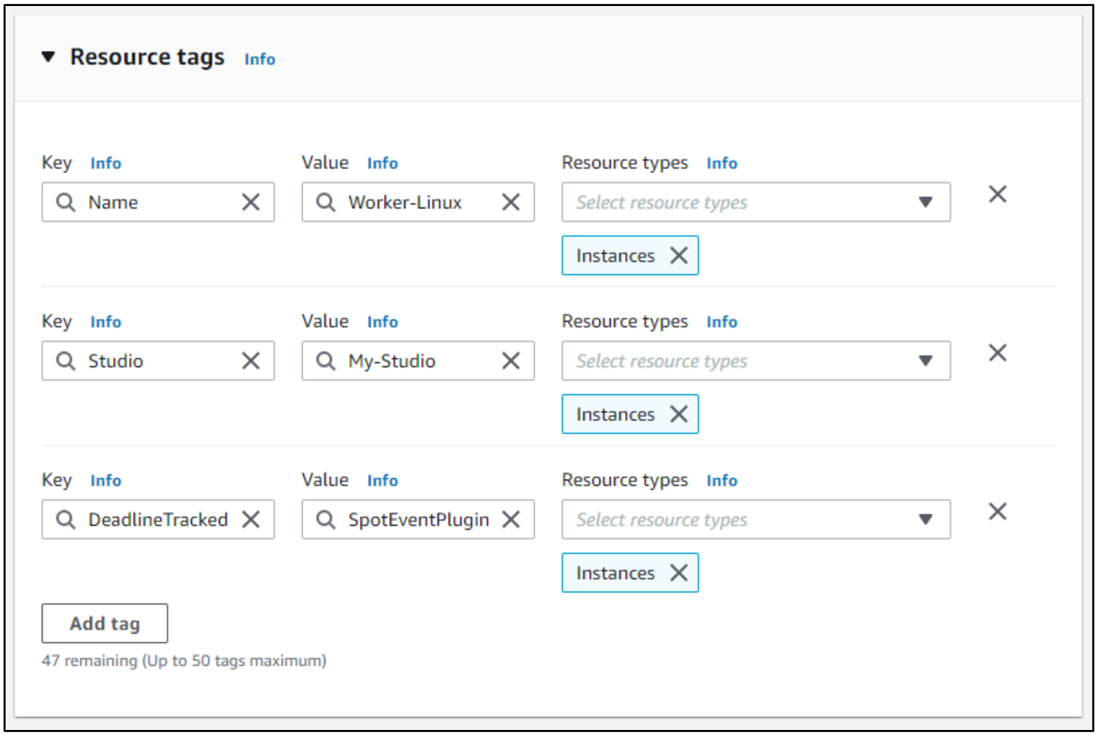
6. Change the **AMI ID** to be the one for your Linux Worker that you just created.



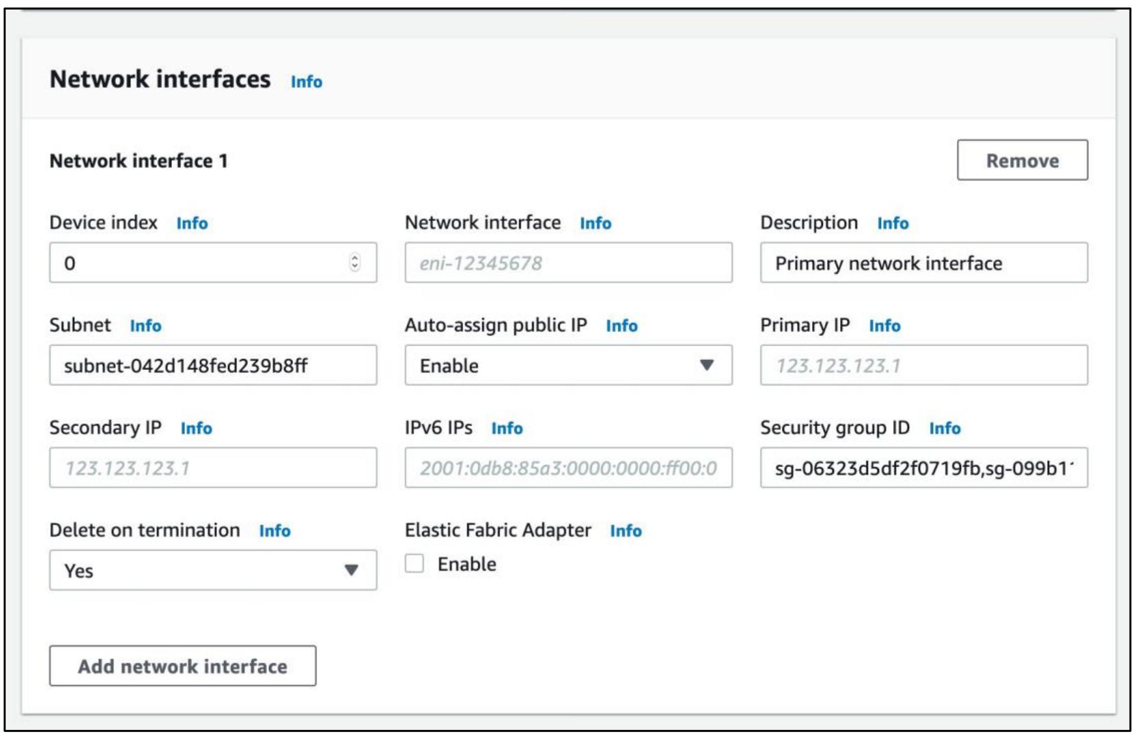
7. Under **Storage (volumes)** change **Volume type** to **Don't include in launch template.**



8. Under **Resource tags**, click **Add tag**
   a. Set **Key** to **DeadlineTrackedAWSResource**
   b. Set **Value** to **SpotEventPlugin**



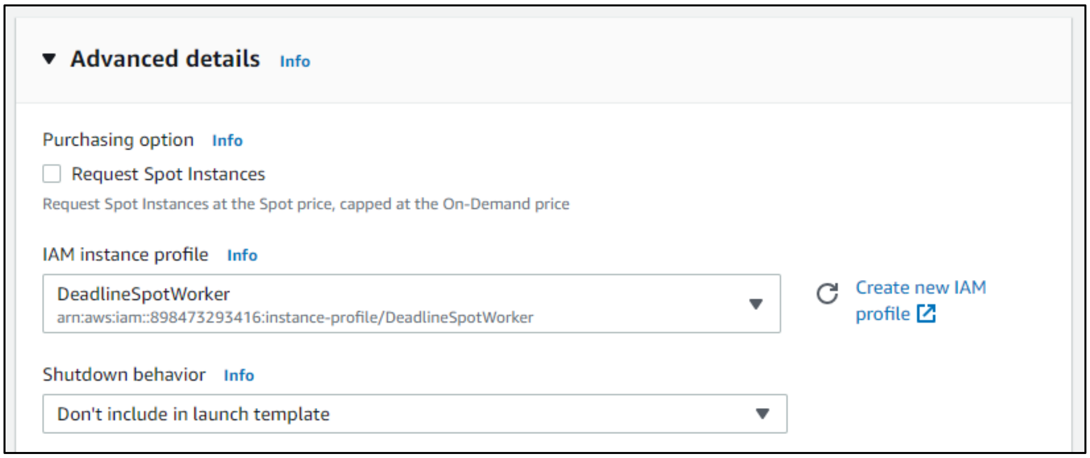
9. Under **Network interfaces** make sure **Auto-assign public IP** is set to **Enable.**



10. Open **Advanced Details**

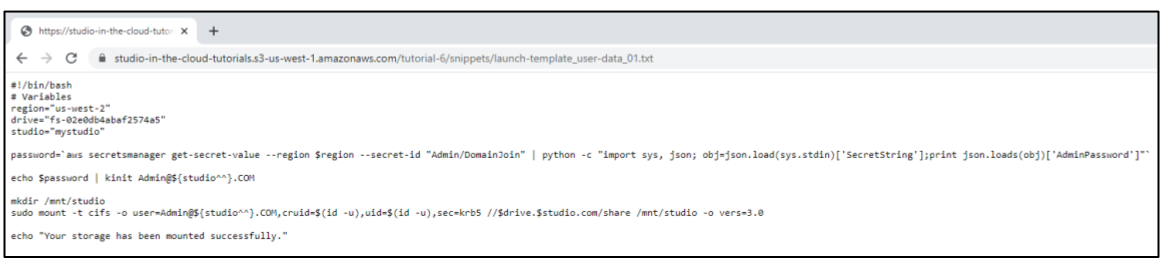
a. Set **IAM instance profile**: DeadlineSpotWorker

b. Set **Shutdown behavior:** Don’t include in launch template.



c. Set **EBS-optimized instance:** Don’t include in launch template.

d. For **User data** <shift>+click the image below to open a new browser tab with the text that needs to be entered into the User data entry field:




```
#!/bin/bash
# Variables
region="us-west-2"
drive="fs-02e0db4abaf2574a5"
studio="mystudio"

password=`aws secretsmanager get-secret-value --region $region --secret-id "Admin/DomainJoin" | python -c "import sys, json; obj=json.load(sys.stdin)['SecretString'];print json.loads(obj)['AdminPassword']"`

echo $password | kinit Admin@${studio^^}.COM

mkdir /mnt/studio
sudo mount -t cifs -o user=Admin@${studio^^}.COM,cruid=$(id -u),uid=$(id -u),sec=krb5 //$drive.$studio.com/share /mnt/studio -o vers=3.0

echo "Your storage has been mounted successfully."
```

launch-template_user-data_01.txt - <shift>+click the image above to open the text file in a new tab

e. Cut and paste the text from the browser tab into the **User data** entry field

f. Expand the size of the **User data** entry field to make it easier to read by clicking and dragging on the bottom right corner:

User data Info

```
sudo mount -t cifs -o user=Admin@${studio^^}.COM,cruid=$(id -u),uid=$(id
-u),sec=krb5 //$drive.$studio.com/share /mnt/studio -o vers=3.0

echo "Your storage has been mounted successfully."
```

☐ User data has already been base64 encoded

- **IMPORTANT: You will need to update the items highlighted below in yellow with specific information for your studio:**

User data Info

```
#!/bin/bash
# Variables
region="us-west-2"
drive="fs-02e0db4abaf2574a5"
studio="mystudio"
```

- Update the value in quotes to the right of **region** with the **region for your studio**. *You can find your region on your cheat sheet.*
- Update the value in quotes to the right of **drive** with your **FSx File System ID.** *This can also be found on the cheat sheet.*
- Update the value in quotes to the right of **studio** with the **Directory NetBios name** for your studio (e.g., mystudio). *Refer to your cheat sheet for this as well.*

Your user data should look something like this when you're all done:

User data Info

```
#!/bin/bash
# Variables
region="us-west-2"
drive="fs-02e0db4abaf2574a5"
studio="mystudio"

password=`aws secretsmanager get-secret-value --region $region --secret-id "Admin/DomainJoin" | python -c "import sys, json; obj=json.load(sys.stdin)['SecretString'];print json.loads(obj)['AdminPassword']"`

echo $password | kinit Admin@${studio^^}.COM

mkdir /mnt/studio
sudo mount -t cifs -o user=Admin@${studio^^}.COM,cruid=$(id -u),uid=$(id -u),sec=krb5 //$drive.$studio.com/share /mnt/studio -o vers=3.0

echo "Your storage has been mounted successfully."
```

☐ User data has already been base64 encoded

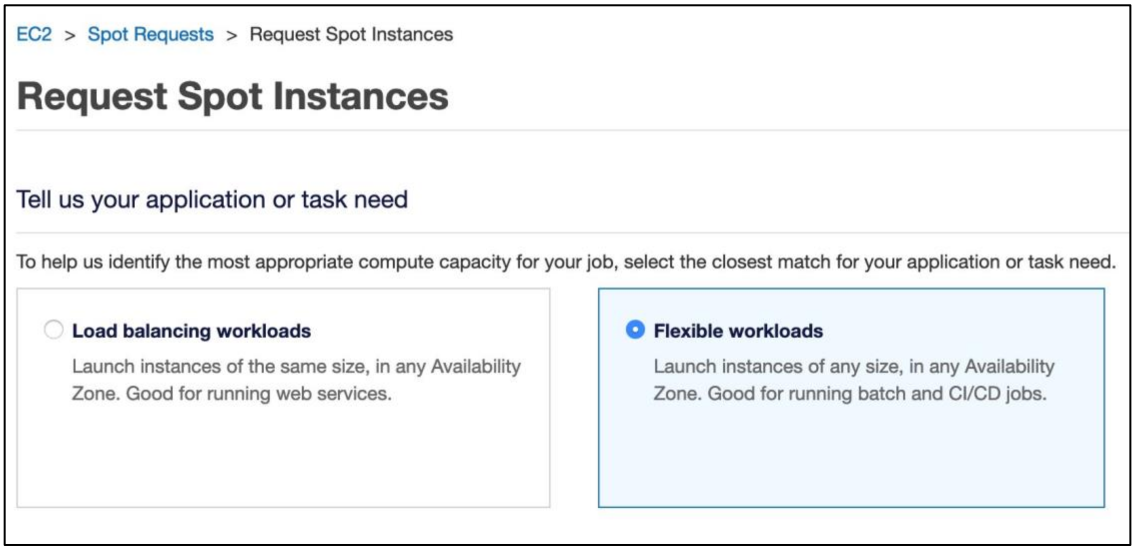
Note: The contents of user data get run every time a new farm worker is launched. In this case, the commands above will mount your FSx drive onto the instances automatically when they launch.

11. Click **Create launch template.**

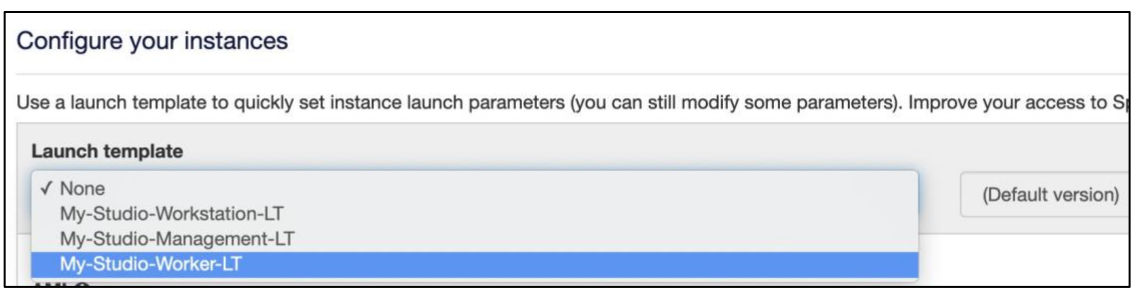
Creating Your Render Fleet

To allow Deadline to automatically spin up Linux workers for us we need to create a spot request.

1. In the **EC2 Console**, click **Spot Requests** on the left hand bar.
2. Click **Request Spot Instances.**
3. Switch to **Flexible workloads.**



4. Select your newly created Linux Worker Launch Template (e.g., My-Studio-Worker_LT) in the **Launch Template** bar, and make sure it is the correct version.



Most of the settings can be left as default.

5. Scroll down and select your **Total Target Capacity**. For now leave it at 1, you can change it later.
6. Select the checkbox next to **Maintain target capacity.**

   Once you are ready to go **DO NOT CLICK Launch.**
7. Scroll down to the bottom and look in the right corner and click **JSON config.** This will download a file called **config.json.**

   We will modify this configuration file to tell Deadline what instances to launch when someone submits a render for a given **group**.

Attaching Render Fleet

1. Go to **Services** → **EC2** and connect to your **Render Scheduler**
2. Log in as Administrator and make sure **Deadline Monitor** is open.
3. Go **Tools**→ **Configure Events.**

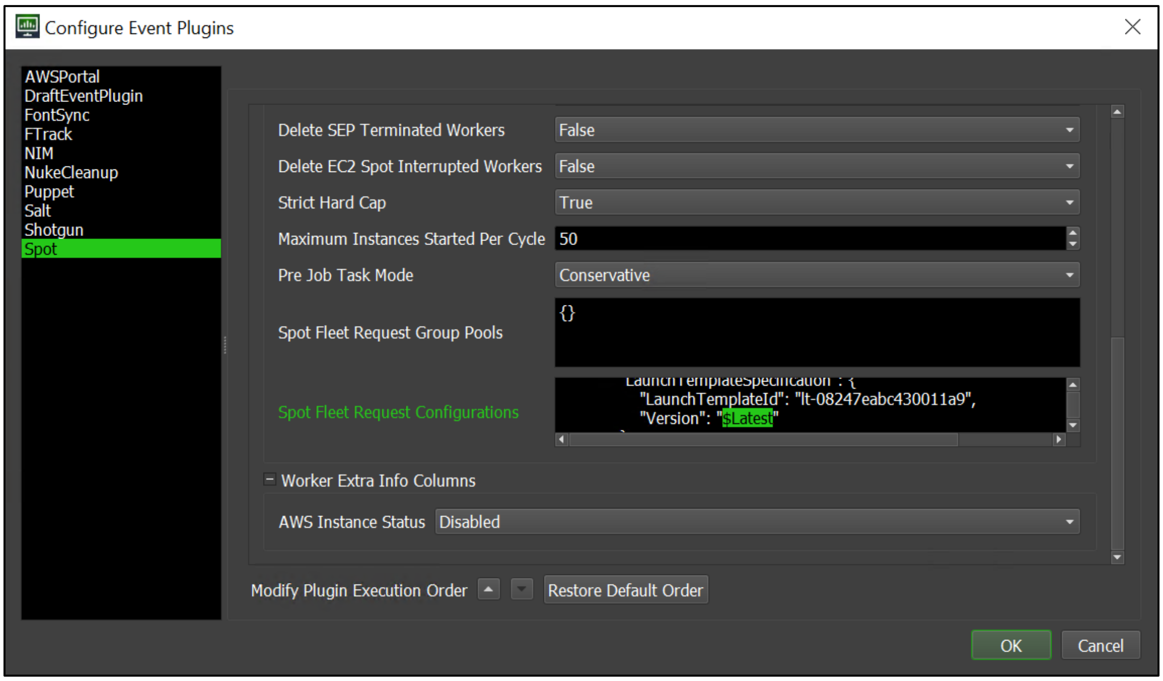
   If you don't see Configure Events, make sure you **Enable Super User** mode in the **Tools** menu.
4. Click **Spot.**
   - Make sure to set the **State** to **Global Enabled.**
   - Enter the **access key ID** and **secret access key** for your IAM user. If you created a new IAM user in Tutorial 1 you can find the access key ID and secret access key in the credentials.csv you downloaded at that time. *You can find the location of your credentials.csv on the cheat sheet.*
   - If you didn't setup the IAM user for your account or do not have the credentials.csv handy, you can create a new access key using these directions: Where's My Secret Access Key?
   - Make sure the **Region** is set correctly.
   - Inside the **Spot Fleet Request Configurations** field, paste the contents of your **config.json**, but be sure to add it with the following line in front of it:

```
{"linux_worker":
```

   - And this line after:

```
}
```

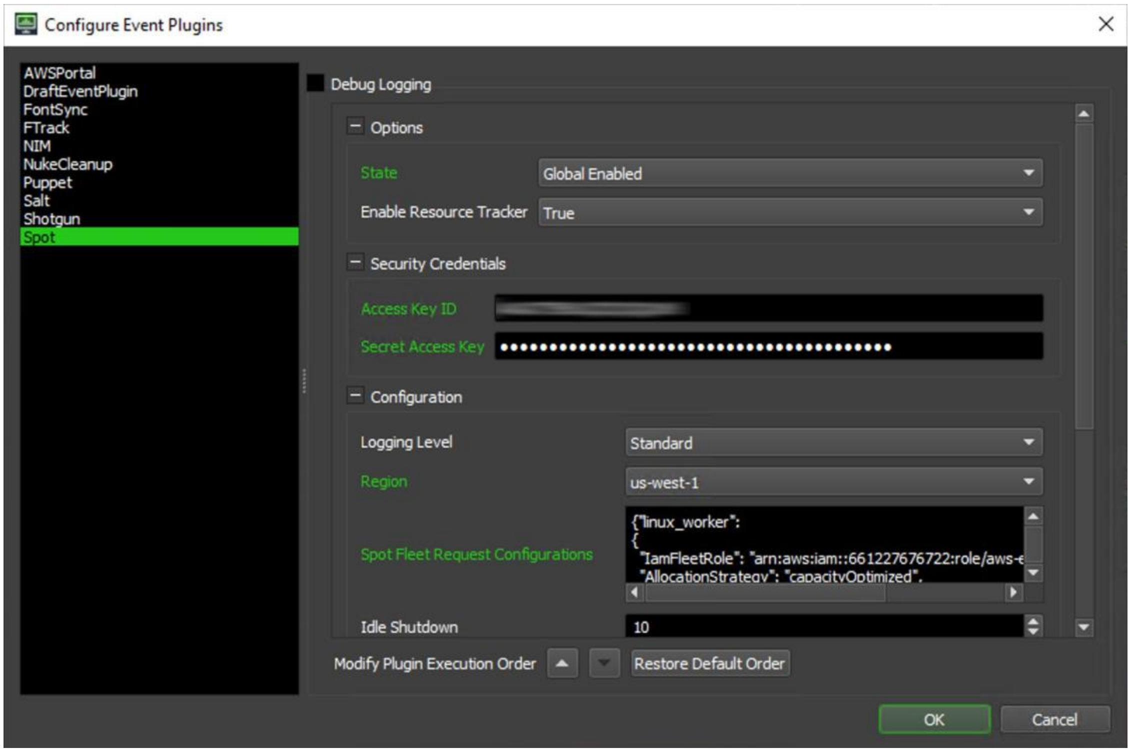
- Also in the **Spot Fleet Request Configurations** field, find the line that refers to the version number of your Linux worker launch template. It is located under the LaunchTemplateId and should currently be set to 1.
- Change 1 to **$Latest**. Using the value of $Latest will ensure that the Spot Fleet Request Configuration is always pointing to the latest version of your template, in case you need to update it later.



The whole thing should look something like below. Now is a good time to double check that you cut and pasted everything correctly, especially the two lines above.

```
{"linux_worker":
{
    "IamFleetRole": "arn:aws:iam::86847329516:role/aws-ec2-spot-fleet-tagging-role",
    "AllocationStrategy": "capacityOptimized",
    "TargetCapacity": 1,
    "ValidFrom": "2021-07-22T19:24:56Z",
    "ValidUntil": "2022-07-22T19:24:56Z",
    "TerminateInstancesWithExpiration": true,
    "LaunchSpecifications": [],
    "Type": "maintain",
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e4a36cd923ad2649",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "m5.2xlarge",
                    "WeightedCapacity": 1,
                    "SubnetId": "subnet-0a93a4e620416d2f4"
                },
                {
                    "InstanceType": "m5ad.2xlarge",
                    "WeightedCapacity": 1,
                    "SubnetId": "subnet-0a93a4e620416d2f4"
                },
                {
                    "InstanceType": "m5d.2xlarge",
                    "WeightedCapacity": 1,
                    "SubnetId": "subnet-0a93a4e620416d2f4"
                },
                {
                    "InstanceType": "m5.4xlarge",
                    "WeightedCapacity": 1,
                    "SubnetId": "subnet-0a93a4e620416d2f4"
                },
                {
                    "InstanceType": "m5.8xlarge",
                    "WeightedCapacity": 1,
                    "SubnetId": "subnet-0a93a4e620416d2f4"
                },
                {
                    "InstanceType": "m5.12xlarge",
                    "WeightedCapacity": 1,
                    "SubnetId": "subnet-0a93a4e620416d2f4"
                },
                {
                    "InstanceType": "m5.16xlarge",
                    "WeightedCapacity": 1,
                    "SubnetId": "subnet-0a93a4e620416d2f4"
                },
                {
                    "InstanceType": "m5.metal",
                    "WeightedCapacity": 1,
                    "SubnetId": "subnet-0a93a4e620416d2f4"
                },
                {
                    "InstanceType": "m5.24xlarge",
                    "WeightedCapacity": 1,
                    "SubnetId": "subnet-0a93a4e620416d2f4"
                }
            ]
        }
    ]
}
}
```

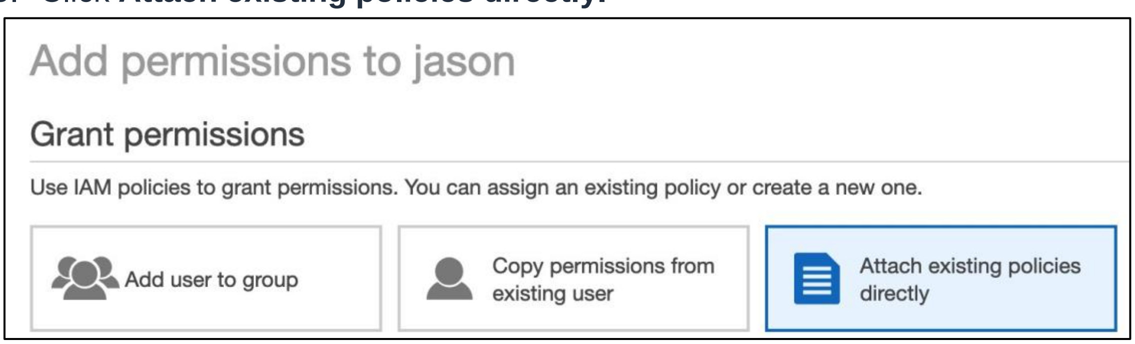

5. Click **OK.**



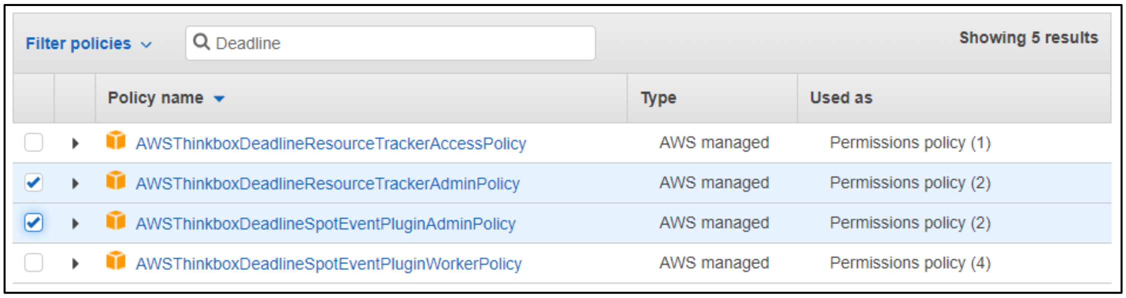
Set Up IAM Permissions

In order for the Spot Event plugin to work correctly, it needs permission to create and modify services in your account. This is done by adding two **AWS managed policies** to your IAM user profile.

1. Go to the **Console** and open your **IAM** settings by going **Services** → **IAM.**
2. In the left navigation pane, click **Users.**
3. Choose the user for your account.
4. Click **Add permissions.**
5. Click **Attach existing policies directly.**



6. Search for **Deadline**
7. Click the check boxes next to **AWSThinkboxDeadlineResourceTrackerAdminPolicy** and **AWSThinkboxDeadlineSpotEventPluginAdminPolicy**

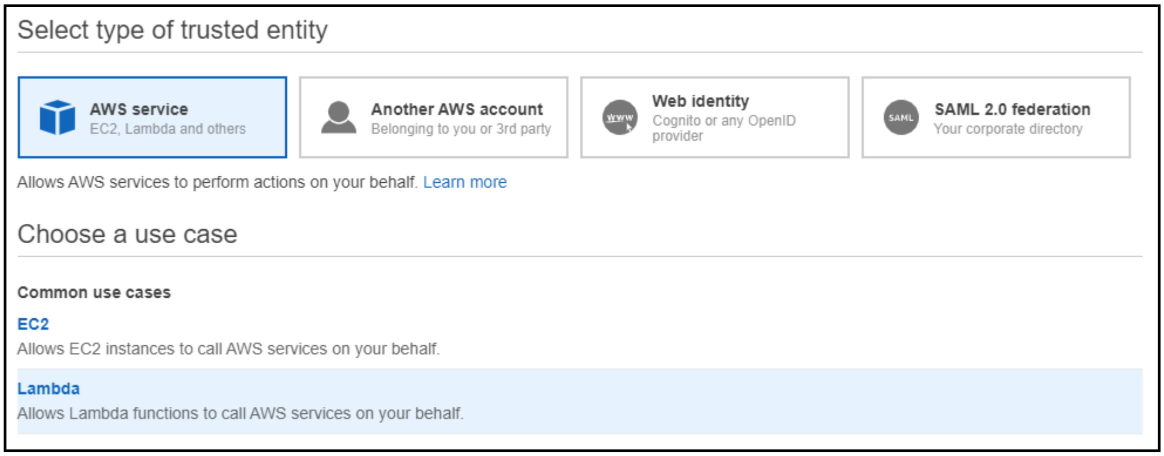


8. Click **Next:Review.**
9. Click **Add permissions.**

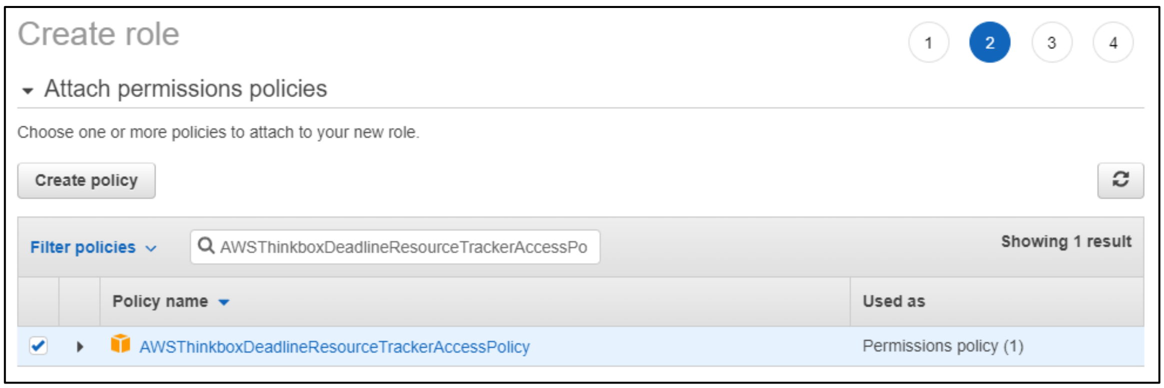
Create a Resource Tracker Role

The Deadline Resource Tracker monitors the health of your render farm workers. In order for it to run properly, you need to create a new IAM Role for it.

1. In the left hand navigation pane, click on **Roles**.
2. Click the **Create role** button.
3. Under **Select type of trusted entity**, choose **AWS service**.
4. Under **Choose a use case**, choose **Lambda** and then choose **Next: Permissions**.



5. Search for **AWSThinkboxDeadlineResourceTrackerAccessPolicy** in the search box, then select the checkbox next to it in the list.



6. Click **Next: Tags**
7. For **Key** enter **Studio** and for **Value** enter the name of your studio (e.g., My-Studio). Then click **Next: Review**
8. For Role name, enter **DeadlineResourceTrackerAccessRole**
9. Check that all the information is correct and then click **Create role**

Test the Fleet

Now that you've set up the event plugin, we should make sure scaling works as expected.

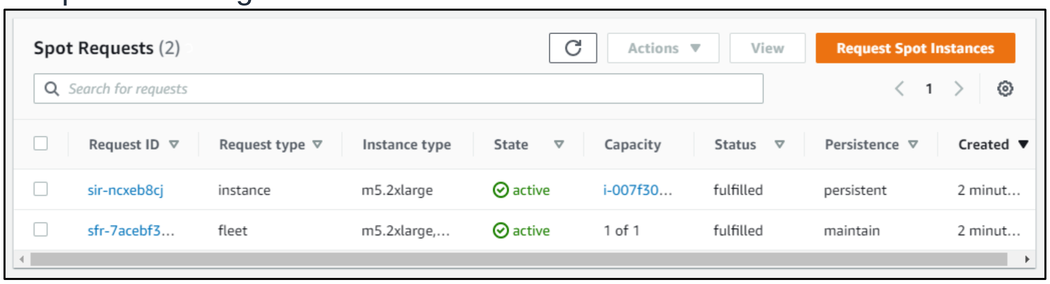
1. Log into your **Workstation** instance and load the scene you tested with earlier in **Blender.**

2. Submit a render to Blender again. Make sure to set the **Frame List** to **1-10** and the **group** to **linux_worker**. This will ensure the Spot Event Plugin recognizes that it needs to create new worker instances if they don't already exist.

   When the render first submits, it shouldn't start rendering because no workers exist in the **linux_worker** group. Wait a few minutes and then the spot event should kick in starting a bunch of workers to launch.

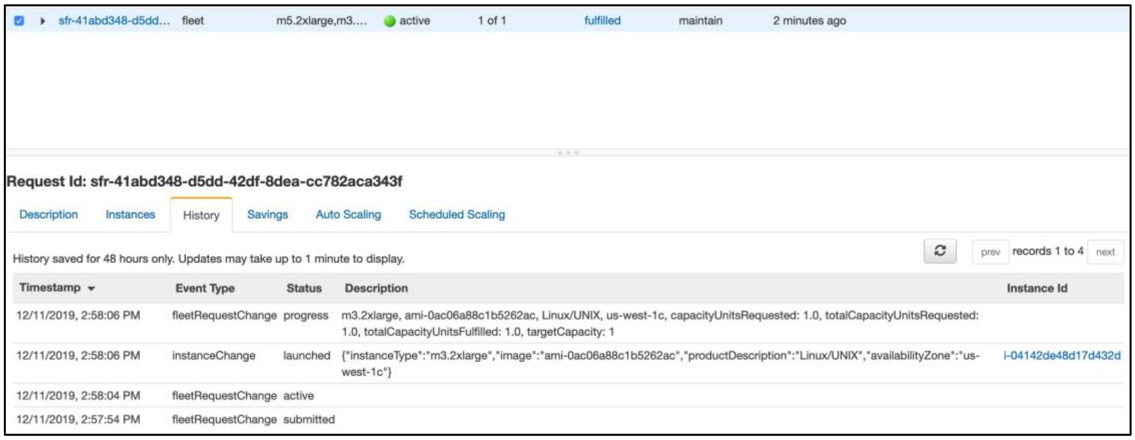
3. If you want to check on the Spot request, go to **Services→ EC2** and then click **Spot Requests.** A Fleet Request should appear, launching render workers.

   Note: It can take 5 minutes for the spot event plugin to start for the first time. If you don't see any requests and you are confident that it should be working, just be patient. It might take a while.



4. If the Spot request is not there double check that you set the correct region in the Spot event plugin. On your **Render Scheduler** instance, open the **Deadline Monitor** and select **Tools->Configure Events**. The click on **Spot** and check the **Region** setting.

5. If the Spot request is there, but no workers are spinning up, you can check for errors by clicking **pending fulfillment** on the Spot request status, and going to the **History** tab. It should give you a good idea of any issues you may encounter.

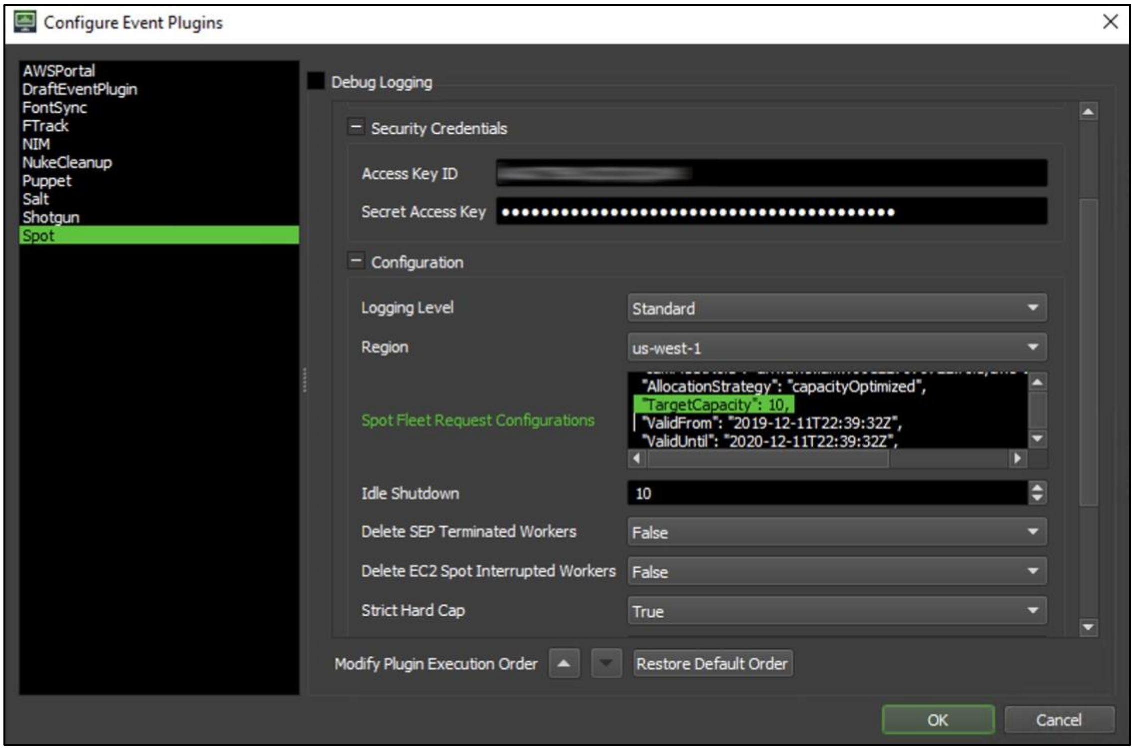


Once the status of the Spot request has changed from pending fulfillment to **fulfilled**, you should see a new worker pop up in the Worker list in the Deadline Monitor and your frames should start rendering.

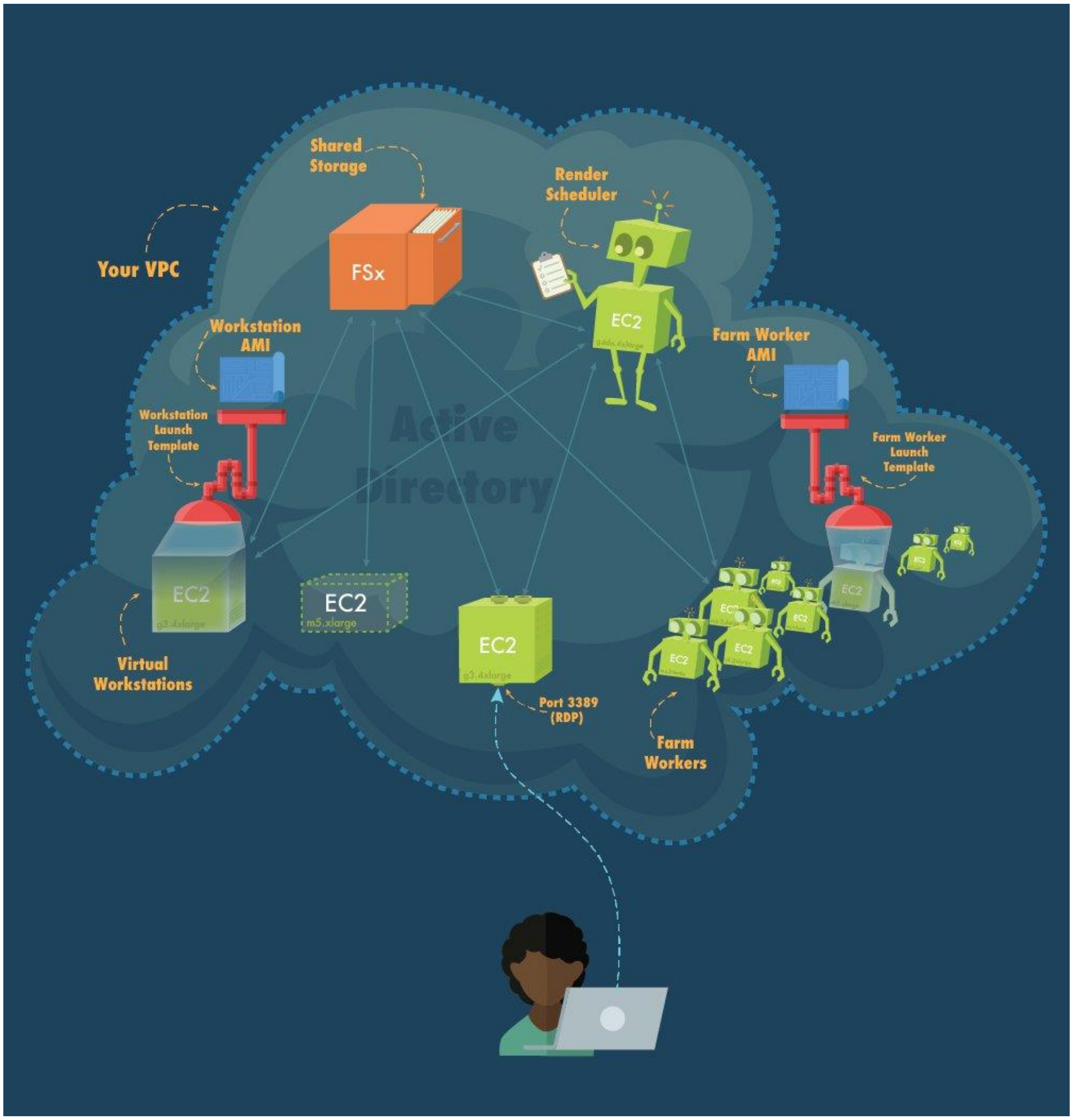
Increase the Number of Instances for the Fleet Request

Once you have verified the Spot event plugin is working, you can increase the number of instances that will launch for a render.

1. Go to the **Deadline Monitor** and choose **Tools** → **Configure Events...**
2. Select the **Spot** event.
3. In the **Spot Fleet Request Configurations** text area, scroll down until you see “TargetCapacity”.
4. Change the value to 10 (or whatever else you might want it to be).
5. Click **OK**.


Configure Event Plugins
AWSPortal
DraftEventPlugin
FontSync
FTrack
NIM
NukeCleanup
Puppet
Salt
Shotgun
Spot
Debug Logging
Security Credentials
Access Key ID
Secret Access Key
Configuration
Logging Level
Standard
Region
us-west-1
Spot Fleet Request Configurations
"AllocationStrategy": "capacityOptimized",
"TargetCapacity": 10,
"ValidFrom": "2019-12-11T22:39:32Z",
"ValidUntil": "2020-12-11T22:39:32Z",
Idle Shutdown
10
Delete SEP Terminated Workers
False
Delete EC2 Spot Interrupted Workers
False
Strict Hard Cap
True
Modify Plugin Execution Order
Restore Default Order
OK
Cancel

Your VPC Now



There have been a couple more additions to your VPC during this tutorial.

In our next tutorial, we'll walk through how you and your artists will actually use your Studio in the Cloud on a day-to-day basis. If you want to keep going, you can leave your User Management instance running and go straight to: [Tutorial 7: Onboarding New Artists and Sample Workflow]

If you'll be taking a little break before continuing, follow the notes below to shut down your User Management instance.

Shut Down Notes

Stop Any Workstations or Running Workers

- Feel free to terminate any running workers or workstations that you don't want to use.
- If you decide to start one up, you can always use your launch templates to quickly spin up instances for any specific tasks.
- If you aren't going to be rendering anytime soon, you can stop your Render Scheduler to save costs.

Appendix

Links to AWS Documentation

- Deadline Spot Event Plugin Documentation
- Where's My Secret Access key?

Downloads

- Spot Event Policy

Tutorial 7. Onboarding New Artists and Sample Workflow

Estimated Time to Complete: 25 minutes

In this tutorial, we'll walk through how you and your artists will actually use your Studio in the Cloud on a day-to-day basis. We'll start with a quick review of your basic infrastructure and then cover the process for onboarding a new artist. Finally, we'll step through a sample workflow that an artist would follow to animate and render a shot with Blender.

Startup Notes

If you're coming straight from the last tutorial and already have your Render Scheduler instance running, then you can continue straight to the next section, Review Your Infrastructure. Otherwise, go to the **EC2 Dashboard** to start it up again. Remember to start up the **Deadline Monitor**, **Deadline RCS (Remote Connection Server)**, and **Deadline Pulse** as well.

Note: We also had you stop your User Management instance at the end of the last tutorial. If you have already added accounts for all of your artists, you can leave that "stopped" for now.

Review Your Infrastructure

Let's quickly review all that you have created over the course of these tutorials.

- **Your VPC** - a private network in the cloud that contains all of the pieces for your studio.
- **User Management instance** - you used this to help set up your studio and add accounts for your artists. At the moment, that instance is stopped, but can be restarted at any time to add more artists.

- **FSx -** fast cloud storage for your user profile, studio tools, content creation applications and production data
- **Render Scheduler instance** - uses Deadline to manage the renders that your artists submit to the farm. Once production begins, this instance should remain running constantly so that it is always available for artist submissions.
- **Windows Workstation instances -** the GPU-enabled instances your artists will be using to create content, connected to FSx to retrieve and store production data. At the moment, you may not have any of these running, but you will add them one-by-one as you add artists.
- **Linux Farm Worker instances -** the instances that will be launched automatically whenever a render is submitted, connected to FSx to retrieve production data. These instances will be started and stopped on demand, so the number of running instances will vary and when there are no renders, there will be no running instances.

When combined, all the above services and different types of virtual workstations allow you to move your content creation to the cloud. Now that you have all the pieces in place, it's time to start thinking about your next steps before entering into production. Below we'll cover additional steps to complete before onboarding your artists.

> ⚠ *Note*: As we mentioned at the beginning of Tutorial 1, the security measures we have implemented so far are sufficient for initial setup and testing of a cloud-based production pipeline. However, before using your setup for production content that may require a higher level of security, we recommend adding extra measures, according to your individual security compliance goals.

Set Up Your Project's Directory Structure

Before beginning work on your cloud-based project, you should create a basic directory structure on your FSx file system to share your production data separate from your tools, profiles, installers and applications. Every studio and project is different, so create a structure that works for you. We suggest creating a top-level directory for your projects and then individual project directories and sub-directories for production shots, a library of assets, etc.

1. Connect to your Render Scheduler instance or another instance that is already connected to your FSx file system.
2. Open **File Explorer** and click **This PC**, then double-click your studio drive (e.g., Z:).

3. Right-click and select **New→Folder**, then create a folder named **projects**. This will keep all the data for your projects separate from your user profiles and studio tools.
4. Double-click the **project** folder.
5. Right-click and select **New→Folder**, then create a folder named for your project and repeat to create any sub-folders that you need (e.g. library, production, etc.).

Onboarding a New Artist

Once they are set up, your storage, Render Scheduler, and Farm Workers should run inside your VPC with little to no supervision. So the only thing you should need to worry about when adding a new artist is launching a new Windows Workstation for them to use.

To keep things simple, we're going to assume that you, as the administrator of your studio, will be responsible for starting and stopping Windows Workstations as they are needed. It is possible to allow individual artists to start and stop their own instances by granting them access to the AWS Console. However, doing so requires that you add them as users on your AWS account (separate from their user accounts on your studio). It also means tightly controlling their access to AWS services so they don't accidentally delete your FSx file system, etc. Ultimately the choice depends on your comfort level as well as how technical your artists are.

Note: Before launching additional Windows Workstations you may need to request a quota increase to your On-Demand G instances limit. Each of the g4dn.4xlarge or g3.4xlarge instances we use in these tutorials requires 16 vCPUs of quota. See the Appendix in Tutorial 1 for instructions on how to view your current quota and request an increase.

Launch a New Windows Workstation

1. Go to **Services → EC2** and click **Launch Templates** in the left panel.
2. Select the Windows Workstation launch template that you made in Tutorial 5 (e.g., My-Studio-Workstation-LT). *See the cheat sheet if needed for the name of your workstation launch template.*
3. Choose **Actions → Launch instance from template.**
4. Select the version.

5. Scroll down to **Instance tags** and add the name of the user to the end of the **Name** tag (e.g., **Workstation - jason**).
6. Choose **Launch instance from template.**
7. Click **Close.**

Send Workstation Login Information to Your Artist

In order for your artist to connect to their new virtual workstation, you will need to send them some login information.

1. Select **Instances** in the left panel
2. When your new workstation instance is done initializing, select it and make note of the **Public IPv4 address** on the **Details** tab at the bottom of the page.

   You will need to send the following information to your new artist:

   - IP address of their workstation
   - username
   - password
   - your Active Directory's NetBios name (e.g., mystudio), they will need this to login. *Refer to the Tutorial 2 section of the cheat sheet for your active directory's NetBios name.*
   - A reminder to change their password after they have logged in for the first time
   - A brief explanation of the directory structure that you set up for your project so they will know how and where to store their data

Workstation Setup - Artist Instructions

How to Connect to Your Windows Virtual Workstation

1. Download and install Remote Desktop for Windows, macOS or Linux.
2. Launch Remote Desktop.
3. In the **Computer** field, enter the **IP address** provided by your administrator.
4. Expand **Show options.**

5. For **User name** enter the **Active Directory NetBios name** and **username** provided by your administrator, with a "\" in between, like this **<Active Directory NetBios name>\<username>** (e.g., mystudio\jason).
6. Click **Connect.**
7. In the popup window enter the **password** provided by your administrator then click **OK.**
8. When a window pops up that says **The identity of the remote computer cannot be verified** click **Yes** to continue connecting.

A new window will open that shows the desktop of your cloud workstation.

Change Your Password

1. In your Remote Desktop session, hit **<ctrl>**+**<alt>**+**<end>.**
2. Next click **Change a password.**
3. Type in your old password and then your new one and press **<enter>.**

Sample Workflow

Creating Content with Blender

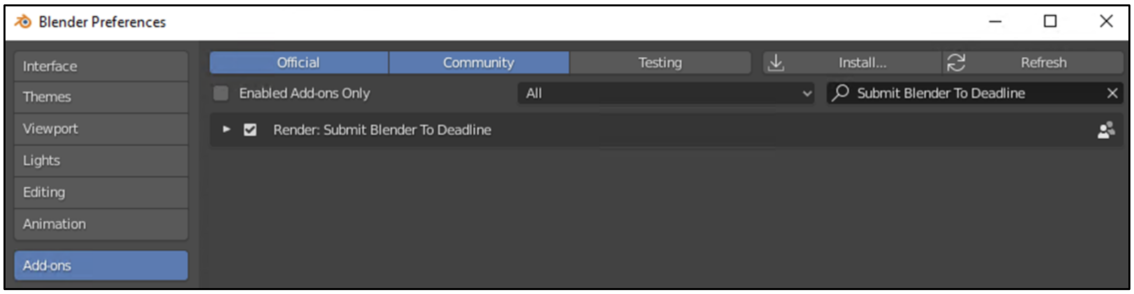
Blender is already installed on your workstation instance. You can launch it using a shortcut located here: **Z:\applications\blender .**

Create your content in Blender, being sure to save your work to the appropriate folder on the **Z: drive** and <u>not</u> to the C: drive of your workstation.

You will need to install the Deadline Blender submitter before submitting:

1. Run **Blender.**
2. Go **Edit → Preferences...** (or **File→Preferences** - for Blender 2.79 & earlier).
3. Click **Add-Ons** in the left panel.
4. Click **Install.**
5. Navigate to **Z:\installers\thinkbox\submission\Blender\Client .**
6. Choose **DeadlineBlenderClient.py .**
7. Click **Install Add-on.**

8. Click the check box next to **Render: Submit Blender to Deadline** add-on.



   - Close your preferences window.

9. When you are ready, submit a render to the farm by selecting **Render→Submit To Deadline.**
10. Set the **Group** to **linux_worker.**
11. Set the **Frame List** to your frame range.
12. Make sure the **Blender File** and **Output File** is set properly.
13. Click **Submit.**

Maintaining Your Studio

Once you've completed these tutorials, there is a minimum amount of AWS services that you will need to leave running in order to continue to use your Studio in the Cloud. There are other services that can be stopped and started as needed. Let's go through each one:

Render Scheduler instance

- In order to submit renders to your Linux worker farm, your Render Scheduler instance will need to be running.
- If you anticipate needing to render at any time of day you can leave it running constantly, but be aware that you will be charged for every hour it is running, regardless of whether there are any active render jobs.
- You can conserve money by starting and stopping the Render Scheduler as needed, but be aware that every time you restart it, you will need to login and make sure that the **Deadline Monitor**, **Deadline RCS,** and **Deadline Pulse** are running.

Active Directory

- This allows your users to login to your instances.
- There is nothing you need to do to maintain this service. But it also cannot be stopped and started like an instance.
- You will need to keep your Active Directory for as long as your studio is active. It will keep running until you actively delete it.

Amazon FSx File System

- This is your shared storage for your studio and it must be available for as long as your studio is running.
- Like Active Directory, you cannot stop and start FSx. As long as you leave it active, you should be good to go.

User Management instance

- You should have already stopped this instance during the course of the tutorials.
- You can leave this instance in a stopped state until you need to add new artists. At that point, you can start the instance, add your new artists and then stop it when you're done to conserve resources.

Windows Workstation instances

- If you don't currently have any artists working, you can stop all your Windows Workstation instances.
- You can launch new Window Workstations at any time by using your workstation launch template (e.g., My-Studio-Workstation-LT).
- Make sure to stop your Windows Workstation instances when you artists are not using them to reduce costs. Since they only need to be running when an artist is actively using them, there is not reason to leave them running all the time.

Add-on Tutorials

For additional tutorials that walk through the steps of adding Teradici as a desktop streaming solution to your existing Studio in the Cloud setup, see Studio in the Cloud Add-on Tutorials.

Shut Down Notes

If at any point you are finished with your Studio in the Cloud and would like to shut down all the services associated with it, see the instructions in Shutting It All Down.

Thank you!

We appreciate your time spent reading and following the steps in these tutorials. We hope that this has given you a taste of how to use the scale, power and convenience of AWS to move your studio to the cloud. Happy creating!

Shutting it all Down

Once you have seen how to set up your studio in the cloud, it's important to know how to shut it down. The following are the steps you can take to shut everything down.

Terminate All Running Instances

1. Go to the **EC2 Console** and click the **Instances.**
2. Select all **running** and **stopped** instances and choose **Instance state** → **Terminate instance.**
3. Click **Terminate.**

Delete Launch Templates

1. Click **Launch Templates**
2. One at a time, **select** each Launch Template and choose **Actions** → **Delete template**
3. Confirm deleting by typing **Delete** in the field.
4. Click **Delete**

Check that All Spot Requests are Canceled

1. Click **Spot Requests.**
2. Make sure all spot requests have been cancelled.

3. If they haven't, **select** the spot requests and choose **Actions** → **Cancel request.**

Deregister All AMIs

Make sure to wait until all instances have been terminated**.**

1. Click **AMIs.**
2. **Select** all AMIs associated with your studio. Choose **Actions** → **Deregister.**
3. Choose **Continue.**

Delete FSx File System

1. Choose **Services** → **FSx.**
2. **Select** the **studio** drive.
3. Choose **Actions** → **Delete file system.**

   In the popup window, you can choose to create a final backup, or not if you don't want to keep any of the data.
4. Then **type** the name of your drive in the field (e.g., fs-04af6ed309bc0d5a1).
5. Click **Delete file system.**

Note: It will take a few minutes for the file system to delete.

Delete Active Directory

Make sure to wait until the FSx file system has been deleted.

1. Choose **Services** → **Directory Service.**
2. **Select** your Active Directory.
3. Choose **Actions** → **Delete directory.**
4. In the popup window, enter the **directory name** (e.g., mystudio.com).
5. Click **Delete.**

Note: It will take a few minutes for the directory to delete.

Delete the VPC

It's time to delete the VPC. This will automatically take down all other associated objects (Subnets, Security groups, etc) as well.

1. Choose **Services** → **VPC.**
2. **Select** your Studio VPC.
3. Choose **Actions** → **Delete VPC.**
4. Click **Delete VPC.**

Delete the DHCP Options Set

1. Select **DHCP Options Sets.**
2. **Select** the studio DHCP options set.
3. Choose **Actions** → **Delete DHCP options set.**
4. Click **Delete DHCP options set.**

Delete the Secret

1. Choose **Services** → **Secrets Manager.**
2. Click **Secrets.**
3. **Select** your secret (e.g., Admin/DomainJoin).
4. Choose **Actions** → **Delete secret.**
5. Change the waiting period to 7 days.
6. Click **Schedule deletion.**