

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

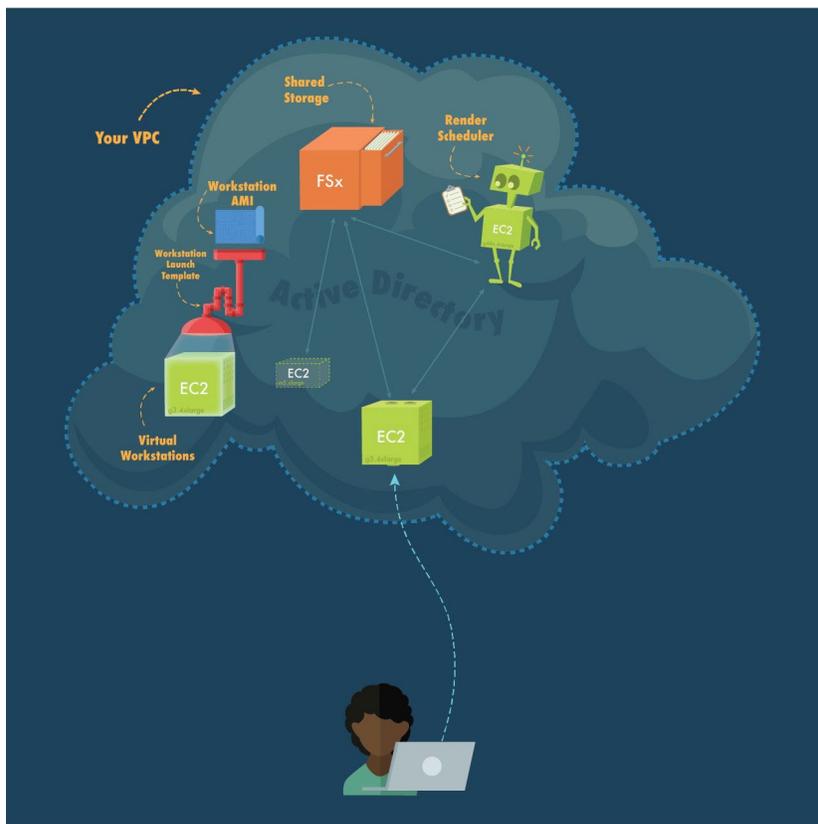
*Estimated Time to Complete: 3 hours*

## Review of: [Tutorial 5: Installing Applications and Creating a Workstation AMI](#)

In the last tutorial we created a virtual workstation that artists can use to do their work. We created a custom AMI with all of the applications we need installed on it, and made a launch template so that we can easily spin up new workstations when we onboard new artists into our studio.

## Overview of this Tutorial

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

- From the **AWS Console** go to **Services**→**EC2**.
- Click on the **Running Instances** link near the top of the page.
- Select your **Render Scheduler** and **Workstation\_Win** instances.
- Click the **Actions** button, 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

- Go to **Services** → **VPC**.
- Click on **DHCP Option Sets** in the left panel.
- Click **Create DHCP Options set**.
  - Enter a name for the options set (e.g., My-Studio-DHCP).
  - 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.*
  - 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.*
  - Leave the rest of the fields blank.

DHCP Options Sets > Create DHCP options set

## Create DHCP options set

Dynamic Host Configuration Protocol (DHCP) provides a standard for passing configuration information to hosts on a TCP/IP network. The options field of a DHCP message contains configuration parameters.

Name  ⓘ

**DHCP options (configuration parameters)**

Specify at least one of the following configuration parameters:

Domain name  ⓘ

Domain name servers  ⓘ

NTP servers  ⓘ

NetBIOS name servers  ⓘ

NetBIOS node type  ⓘ

▶ AWS Command Line Interface command

\* Required Cancel

- Click **Create DHCP Options Set**.
- Click **Close** after the creation confirmation appears.
- *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

- Select **Your VPCs** in the left panel.
- Select your studio's VPC (e.g. My-Studio-VPC).
- Choose **Actions** → **Edit DHCP options set**.
- From the drop down menu, select the id of the DHCP Option set you just created (e.g., `dopt-0857j234nf593hs3u`).

VPCs > Edit DHCP options set

## Edit DHCP options set

VPC ID `vpc-04ecf18bca5e572d6`

DHCP options set\*  ↕ ⓘ

\* Required

- Click **Save**.

## Launch a Linux Instance to Setup a Worker

### Create a 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.



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

#### Create Security Group

**Security group name** ⓘ My-Studio-SSH-SG

**Description** ⓘ Security Group for SSH

**VPC** ⓘ vpc-04ecf18bca5e572d6 | My-Studio-VPC

- Choose your **VPC** (e.g. My-Studio-VPC).
- Add this rule to the **Inbound** tab:

	Protocol	Port	Source	Description
SSH	TCP	22	0.0.0.0/0, ::/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.

- Click the **Create** button.
- Find the security group ID (e.g., sg-0jnb39bfu94hj30da) to the right of the group's name in the list 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

- Go to **Instances** in the left panel.
- Click **Launch Instance**.

- Select **Amazon Linux 2 AMI (HVM), SSD Volume Type** (it should be the first one on the list).
- Choose **m5.2xlarge** as the instance type.

<input type="checkbox"/>	General purpose	m5.xlarge	4	16	EBS only
<input checked="" type="checkbox"/>	General purpose	m5.2xlarge	8	32	EBS only
<input type="checkbox"/>	General purpose	m5.4xlarge	16	64	EBS only

- Click **Next: Configure Instance Details**.
  - **Network:** Your VPC (e.g., My-Studio-VPC).
  - Set subnet, **Public Subnet B**.
  - Set Auto Assign Public IPs to **Enable**.
  - Set IAM role to **EC2DomainJoin**.

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of

**Number of instances** ⓘ  [Launch into Auto Scaling Group](#) ⓘ

---

**Purchasing option** ⓘ  Request Spot instances

---

**Network** ⓘ  ⓘ [Create new VPC](#)

**Subnet** ⓘ  ⓘ [Create new subnet](#)  
250 IP Addresses available

**Auto-assign Public IP** ⓘ  ⓘ

---

**Placement group** ⓘ  Add instance to placement group

**Capacity Reservation** ⓘ  ⓘ [Create new Capacity Reservation](#)

---

**IAM role** ⓘ  ⓘ [Create new IAM role](#)

- Click **Next: Add Storage**.
  - Set the storage size to **300 GB**.

### Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance. You can also edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2.](#)

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IO
Root	/dev/xvda	snap-0b6c8d9cf53ef45a8	<input type="text" value="300"/>	General Purpose SSD (gp2)	10

[Add New Volume](#)

- Click **Next: Add Tags**.

- Add these tags as key - value pairs.
  - Key = **Name**, Value = **Worker-Linux**.
  - Key = **Studio**, Value = **My-Studio** (or the name of your studio).
- Click **Next: Configure Security Group**.
  - Choose **Select existing security group**.
  - 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.*

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#)

**Assign a security group:**  Create a new security group  
 Select an existing security group

Security Group ID	Name	Description
<input type="checkbox"/> sg-0cf273941d1d327ac	d-916735bbc5_controllers	AWS created security group for d-9
<input type="checkbox"/> sg-0469b547302b78a18	default	default VPC security group
<input checked="" type="checkbox"/> sg-099b11d866a4bd76e	My-Studio-Deadline-SG	Security Group for Render Schedul
<input type="checkbox"/> sg-0c0efc81a52047acb	My-Studio-Remote-Desktop-SG	Allows for Remote Desktop Connec
<input checked="" type="checkbox"/> sg-06323d5df2f0719fb	My-Studio-SSH-SG	Security Group for SSH
<input type="checkbox"/> sg-03ec498e33121a00f	My-Studio-Storage-SG	Security group for FSx

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

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

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

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

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

### Connect To Your Instance ×

I would like to connect with

A standalone SSH client ⓘ

EC2 Instance Connect (browser-based SSH connection) ⓘ

A Java SSH Client directly from my browser (Java required) ⓘ

---

Connect using a custom user name, or default to the user name for the AMI used to launch the instance. [Learn more](#)

**User name**  ⓘ

---

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

## Join Active Directory

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

```
sudo yum -y update
```

- Next, run this command to install some specific tools needed to connect to Active Directory:

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

- 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**.
- 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.
- Enter this command:

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

- To find the correct line type: **/PasswordAuthentication no** and hit **<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.
- Move the cursor over using the arrow keys to the word **no**.
- Type **cw**.
  - This puts you in **Change Word** mode.
- Type the word **yes**.
  - Replaces the word **no** with **yes**.

- Hit **esc**.
  - Takes you out of **Edit** mode.
- Type **:wq** and hit **<enter>**.
  - “:wq” **Writes** the file and then **Quits** vi.
  - [Here’s](#) a cheat sheet for vi commands.
- Next, restart the sshd service with this command:

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

- 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. Please close that 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

- In the AWS Console for EC2, select your worker instance and click the **Connect** button.
- Choose **EC2 Instance Connect**.
- Leave **User name** as **ec2-user**.
- Click **Connect**.
- A window will open with access to your instance.
- 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

- First run this command to install some utilities:

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

- Next run the command below and enter the Active Directory Admin password when prompted. Make sure to replace MYSTUDIO.COM 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
```

- 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,cuid=$(id -u),uid=$(id -u),sec=krb5 //fs-0c0b4fab1db1b9a0e.mystudio.com/share /mnt/studio -o vers=3.0
```

*Please note which text is capitalized. These commands ARE case sensitive.*

- To test and see if it mounted correctly, do an `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

- Run this command:

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

- If asked any questions type `y` and hit enter.
- Next, navigate to the Thinkbox installers directory

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

- Run this command to navigate to un-tar the tar file. Note: You may need to replace the version number (e.g., 10.1.1.3) with number of the version that you are using.

```
tar -xvf Deadline-10.1.1.3-linux-installers.tar
```

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

```
sudo ./DeadlineClient-10.1.1.3-linux-x64-installer.run
```

- 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).

- Remote Connection Server: **N**.
  - Is the selection above correct: **Y**.
  - Remote Connection Server: **2**.
  - 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.*
  - Client Certificate: (Blank).
  - Certificate Password: (Blank).
  - Launch Worker: **y**.
  - Install Launcher as Daemon: **y**.
  - User Name: (Blank).
  - Do you want to continue: **Y**. (you will be asked this twice)
- Wait for the installer to finish. You will see a message that the launcher service has been stopped and then started again.
  - Hit <enter> in the SSH session to get the commandline back and then navigate to the Deadline10 directory:

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

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

```
sudo vi deadline.ini
```

- Put your cursor over the word **False** next to ProxyUseSSL.
- Hit **cw** on your keyboard.
- Type **True**.
- Edit the line "ProxySSLCertificate=" so it says (all on one line):

```
ProxySSLCertificate=/mnt/studio/app_env/thinkbox/DeadlineCertificates/Deadline10RemoteClient.pfx
```

- Hit **esc**.
- Type **:wq** and hit **Enter**.

## Install Blender

- Navigate to **/mnt/studio/installers/blender**:

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

- Then extract Blender by running this command:

```
tar -xvf blender-2.81a-linux-glibc217-x86_64.tar.bz2
```

- Note #1: You may need to change the Blender version (e.g., 2.81a) to match the version of Blender that you are installing.
- Note #2: You may get some symlink errors when running the Blender installer from /mnt/studio, but don't worry you can ignore them.
- Rename the folder called **blender-2.81a-linux-glibc217-x86\_64** to **blender**. Note: Again, you will need to change the version number to match your version of Blender

```
mv blender-2.81a-linux-glibc217-x86_64 blender
```

- 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**.

The screenshot shows the Deadline Monitor application window. The 'Jobs' panel displays one job: 'Test Dead...' with a progress of 100% (10/10) and status 'Complete'. The 'Workers' panel shows two workers: 'IP-0A00000B' with Machine User 'Administrator' and 'ip-10-0-1-29' with Machine User 'root'. The 'root' worker is highlighted in green. The interface also shows a search bar, a menu bar, and a status bar at the bottom indicating 'Panels Locked' and 'User: administrator'.

## Render Something to Test the Setup

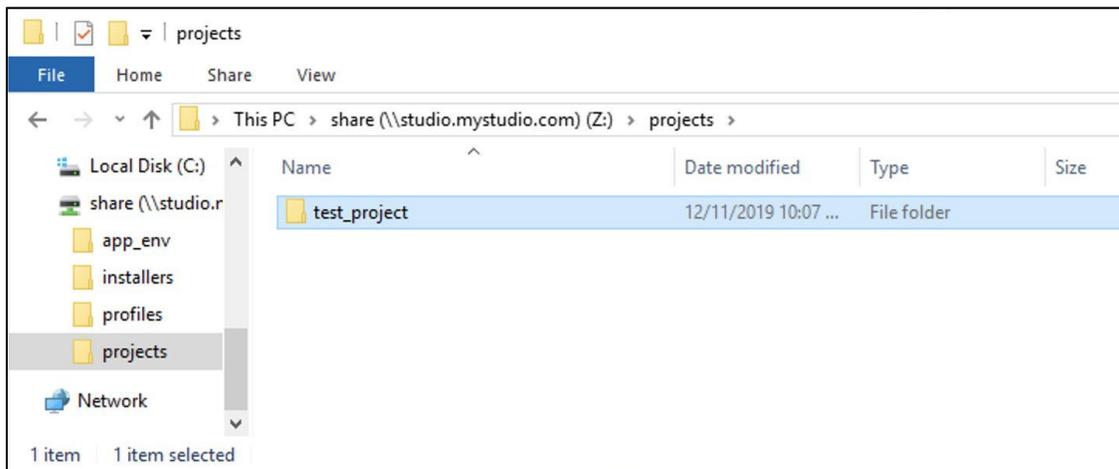
Now that you've got most of the pieces together, it's a good idea to test the setup to make sure you can submit a render from your workstation, have the Render Scheduler pick it up, and run the render on your Worker. To do this, make sure you have three instances running: your

Workstation, the Render Scheduler, and your Linux Worker. If any of those aren't running, please get them started.

## Set Up a Shot to Render

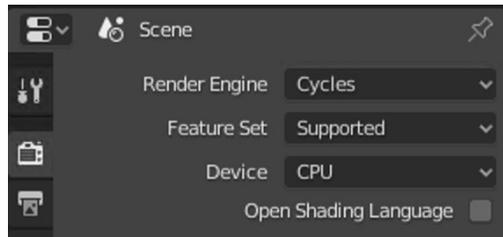
This will involve using the workstation to set up a shot to do a test render.

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



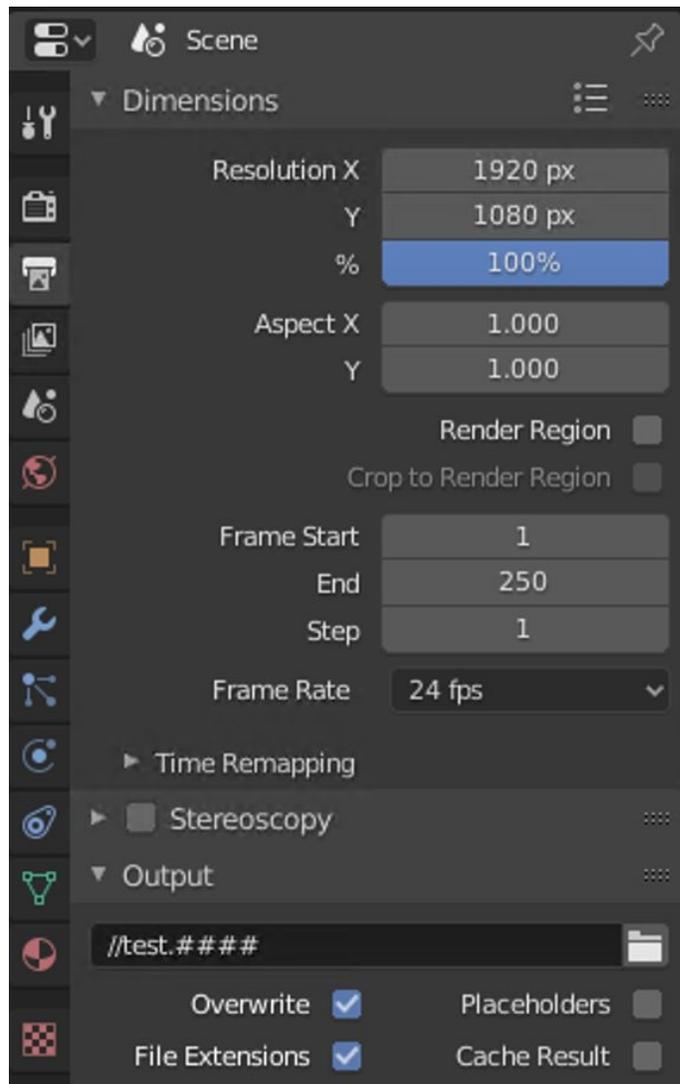
- Launch **Blender** by running the Blender shortcut located in **Z:\applications**.
- The Deadline Blender submitter needs to be re-installed for each user, so you should do that now:
  - Go **Edit** → **Preferences...** (or **File**→**Preferences** - for Blender 2.79 & earlier)
  - Click on **Add-Ons** in the left panel.
  - Click **Install...**
  - Navigate to **Z:\installers\thinkbox\submission\Blender\Client .**
  - Choose **DeadlineBlenderClient.py** .
  - Click **Install Add-on**.
  - Click the checkbox next to **Render: Submit Blender to Deadline** add-on and close the Preferences window.
- 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).

- Set your output settings
  - In the **Properties** Panel on the right, click on **Render Properties** (the icon looks like the back of an SLR camera).
  - For **Render Engine** choose **Cycles**.

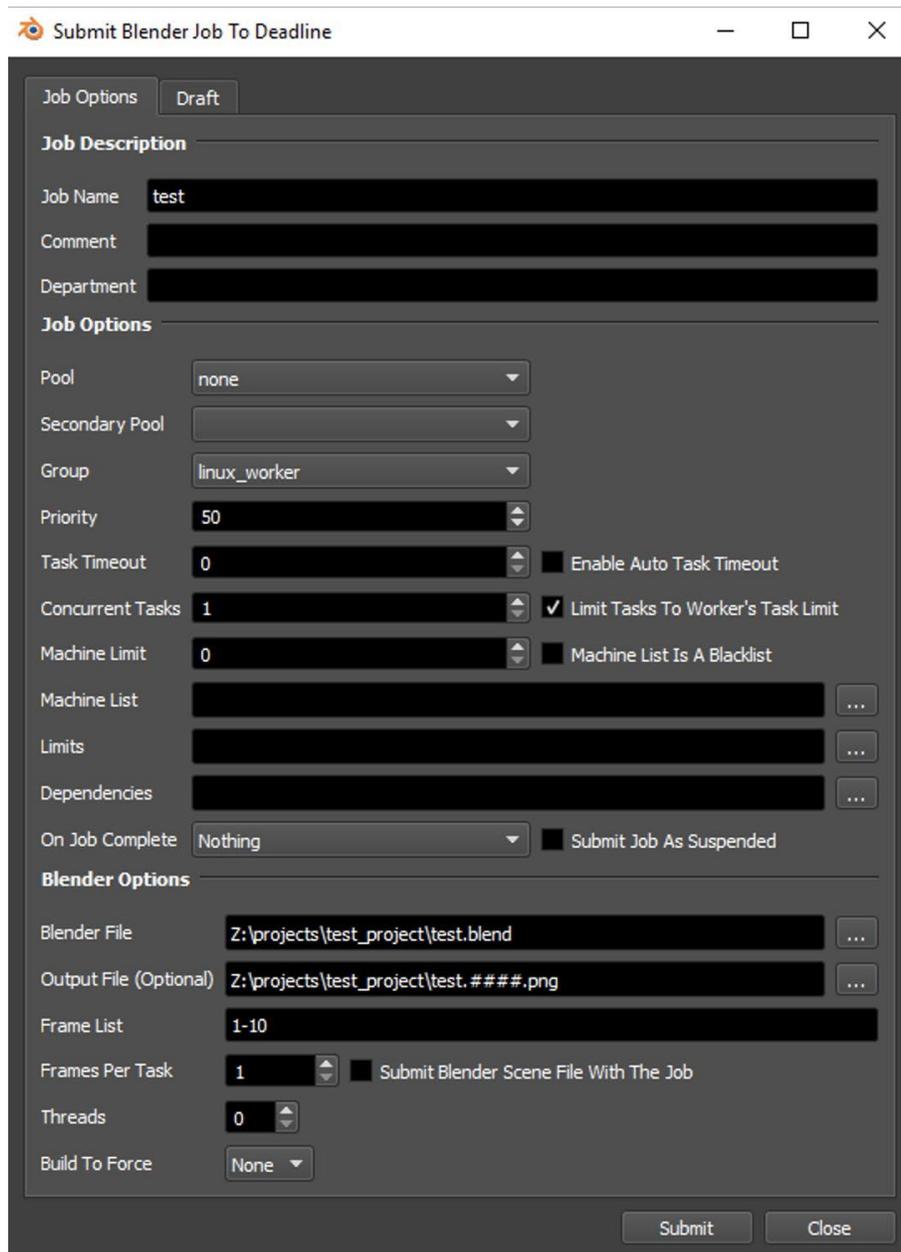


- Click on **Output Properties** (the icon looks like an inkjet printer).
- Under **Output** change the value to **//test.####** (this will ensure the images are written to the **test\_project** folder).

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

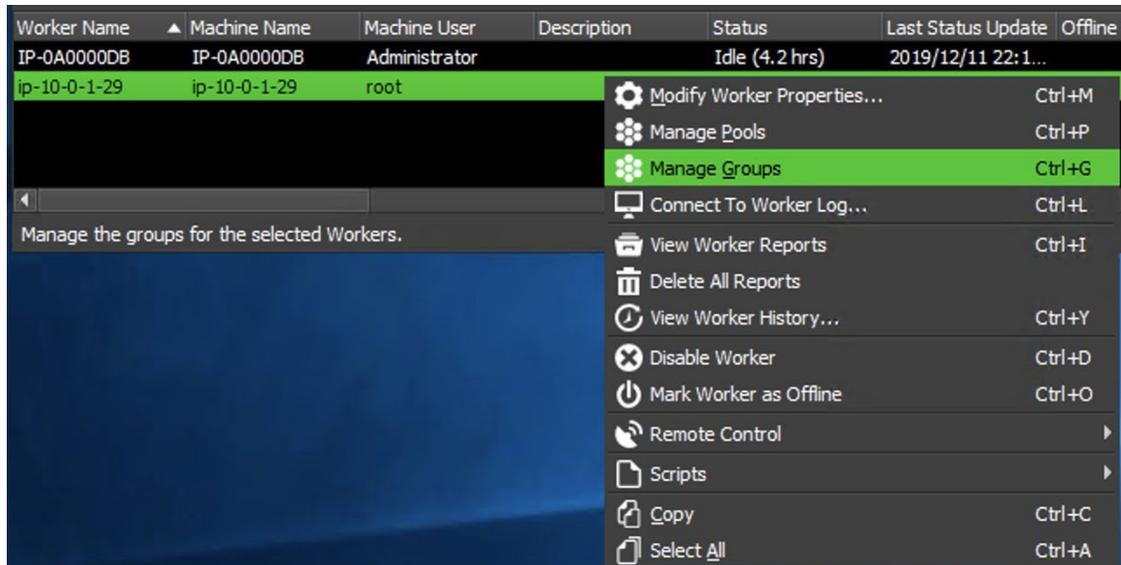


- Submit your render
  - Choose **Render**→**Submit to Deadline**.
  - Set the **Group** to `linux_worker`.
  - Set the **Frame List** to 1-10.
  - Make sure the **Blender File** and **Output File** is set properly.
  - Click **Submit**.

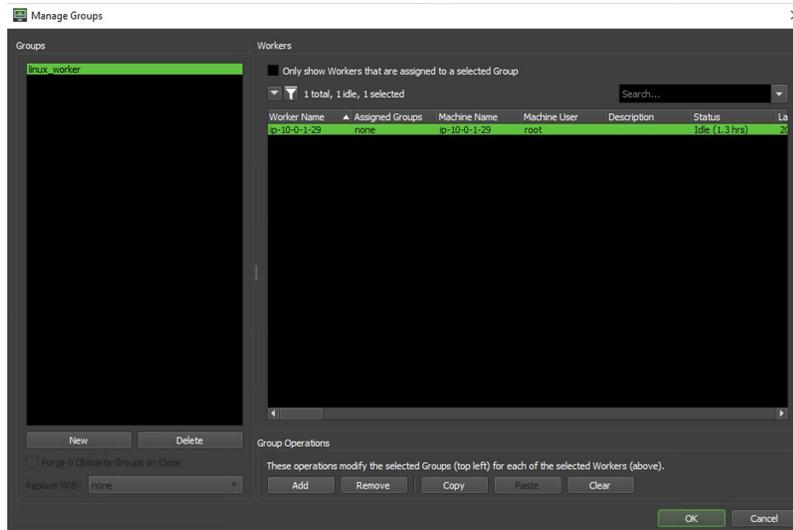


## Check the Render Scheduler

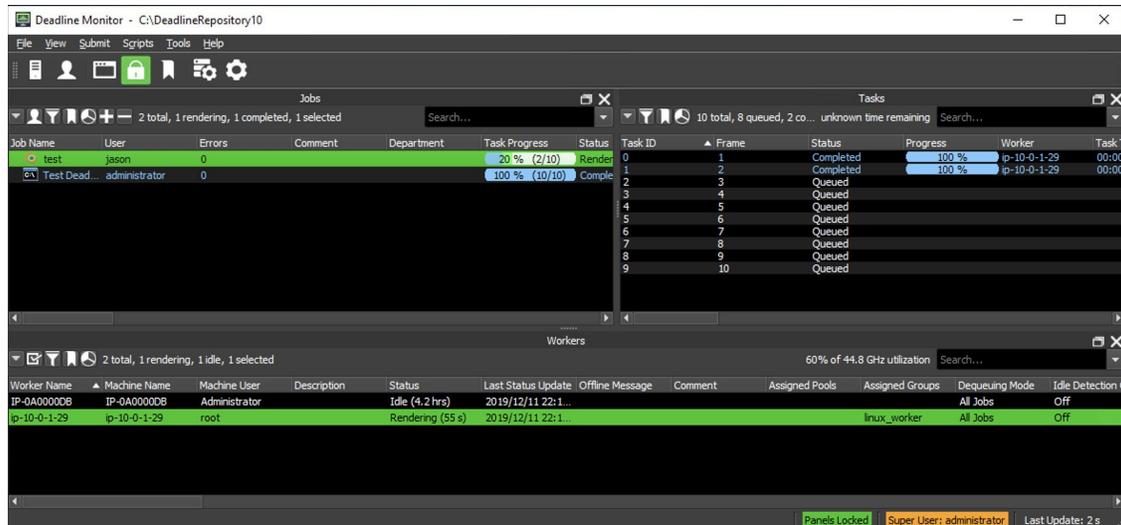
- Log into your **Render Scheduler** as **Administrator**.
- To get the **worker** to pick up the job, you need to add the **linux\_worker** group to it.
  - Make sure you have super user mode enabled by choosing **Tools**→**Super User Mode**.
  - Select the **worker** in the **list of workers** at the bottom of the monitor window.
  - Click with the **Right Mouse Button** and choose **Manage Groups**.



- Select **linux\_worker** on the left, and the machine on the right and click **Add** to add the **linux\_worker** group to the machine.
- Click **Ok**.
- Once your worker has gotten the **linux\_worker** group added to it, it should shortly 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 on 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.
    - Click on **Blender**.
    - You will see a list of potential locations for the Blender executable. As you can see, one of the locations it's looking is: `/usr/local/Blender/blender`.
    - 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.
    - 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 mouse click on the one who's name 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, please 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.
  - Go to **Services**→**EC2**.
  - Click on **Instances**.
  - Right click on your **Worker-Linux** instance and choose **Instance State**→**Stop**.
- Once stopped, right click on it again and choose **Image**→**Create Image**.
- Enter a name for your image (e.g., My-Studio-Worker-AMI) and a description for your image.
- Check that the **size** of your volume is set to **300 GiB**.
- Click **Create Image**.

### Create Image ✕

**Instance ID** ⓘ i-06134adcb09242c7c

**Image name** ⓘ

**Image description** ⓘ

**No reboot** ⓘ

**Instance Volumes**

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/xvda	snap-0b6c8d9cf53ef45a8	<input type="text" value="300"/>	General Purpose SSD (gp2) ⌵	900 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Total size of EBS Volumes: 300 GiB  
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

- In the navigation pane on the left, under Images, select **AMIs**.
- 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.*
- Add appropriate **Name** and **Studio** tags.
- Wait a few minutes while the AMI is pending.

### Create a Launch Template

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

- Go to **Services** → **EC2** and click on **Instances**.
- Right mouse click on your Linux Worker and choose **Create Template From Instance**.
- Name your launch template (e.g., My-Studio-Worker-LT).
- Give your launch template a description if you want.

## Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

### Launch template name and description

Source instance

i-06134adcb09242c7c

Launch template name - *required*

My-Studio-Worker-LT

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '\*', '@'.

Template version description

Launch Template for workers with Deadline and Blender

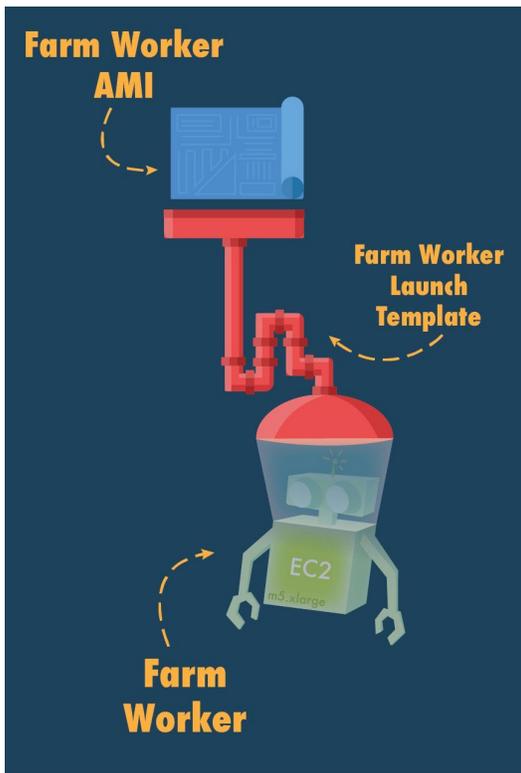
Max 255 chars

Auto scaling guidance [Info](#)

Select this if you intend to use this template with auto scaling

Provide guidance to help me set up a template that I can use with auto scaling

▶ Template tags



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

### Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

#### Amazon machine image (AMI) [Info](#)

AMI

My-Studio-Worker-AMI  
ami-0ac06a88c1b5262ac  
Catalog: My AMIs architecture: 64-bit (x86) virtualization: hvm

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

### Storage (volumes) [Info](#)

▼ Volume 1 (AMI Root) (Custom)

Volume type <a href="#">Info</a>	Device name - <i>required</i> <a href="#">Info</a>	Snapshot <a href="#">Info</a>
EBS	/dev/xvda	snap-0b58b78e81808927e
Size (GiB) <a href="#">Info</a>	Volume type <a href="#">Info</a>	IOPS <a href="#">Info</a>
300	Don't include in launch templ... ▼	2000
Delete on termination <a href="#">Info</a>	Encrypted <a href="#">Info</a>	Key <a href="#">Info</a>
Yes ▼	No ▼	MyKey

[Add new volume](#)

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

**Network interfaces** [Info](#)

---

**Network interface 1** Remove

Device index <a href="#">Info</a> <input type="text" value="0"/>	Network interface <a href="#">Info</a> <input type="text" value="eni-12345678"/>	Description <a href="#">Info</a> <input type="text" value="Primary network interface"/>
Subnet <a href="#">Info</a> <input type="text" value="subnet-042d148fed239b8ff"/>	Auto-assign public IP <a href="#">Info</a> <input type="text" value="Enable"/>	Primary IP <a href="#">Info</a> <input type="text" value="123.123.123.1"/>
Secondary IP <a href="#">Info</a> <input type="text" value="123.123.123.1"/>	IPv6 IPs <a href="#">Info</a> <input type="text" value="2001:0db8:85a3:0000:0000:ff00:0"/>	Security group ID <a href="#">Info</a> <input type="text" value="sg-06323d5df2f0719fb,sg-099b1"/>
Delete on termination <a href="#">Info</a> <input type="text" value="Yes"/>	Elastic Fabric Adapter <a href="#">Info</a> <input type="checkbox"/> Enable	

- Open **Advanced Details**
  - Set **Shutdown behavior**: Don't include in launch template.

**Advanced details** [Info](#)

---

**Purchasing option** [Info](#)  
 Request Spot Instances  
Request Spot Instances at the Spot price, capped at the On-Demand price

**IAM instance profile** [Info](#)  
  
arn:aws:iam::661227676722:instance-profile/EC2DomainJoin ↻

**Shutdown behavior** [Info](#)

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

**EBS-optimized instance** [Info](#)

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

```

https://studio-in-the-cloud-tutorial: x +
studio-in-the-cloud-tutorials.s3-us-west-1.amazonaws.com/tutorial-6/snippets/launch-template_user-data_01.txt

#!/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@${studio^^}.COM,cuid=$(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 on the image above to open the text file in a new tab

- Cut and paste the text from the browser tab into the **User data** entry field
- 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@${studio^^}.COM,cuid=$(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@${studio^^}.COM,cuid=${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.

- Finally, 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.

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

EC2 > Spot Requests > Request Spot Instances

## Request Spot Instances

Tell us your application or task need

To help us identify the most appropriate compute capacity for your job, select the closest match for your application or task need.

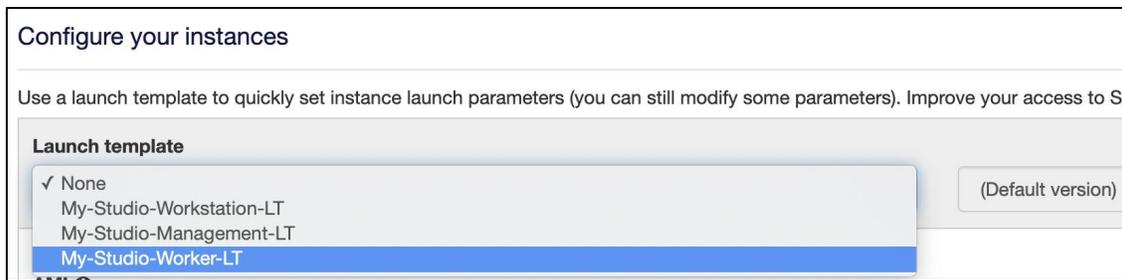
**Load balancing workloads**

Launch instances of the same size, in any Availability Zone. Good for running web services.

**Flexible workloads**

Launch instances of any size, in any Availability Zone. Good for running batch and CI/CD jobs.

- 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.
- Scroll down and select your **Total Target Capacity**. For now leave it at 1, you can change it later.
- Select the checkbox next to **Maintain target capacity**.
- Once you are ready to go **DO NOT CLICK Launch**.
- Scroll down to the bottom and look in the left corner.
- Click on **JSON config** and 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

- Go to **Services** → **EC2** and connect to your **Render Scheduler**
- Log in as Administrator.
- Make sure **Deadline Monitor** is open.
- Go **Tools** → **Configure Events**.
  - If you don't see Configure Events, make sure you **Enable Super User** mode in the **Tools** menu.
- Click on **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** you'll paste the contents of your **config.json**, but be sure to add it with the following line in front of it:

```
{"linux_worker":
```

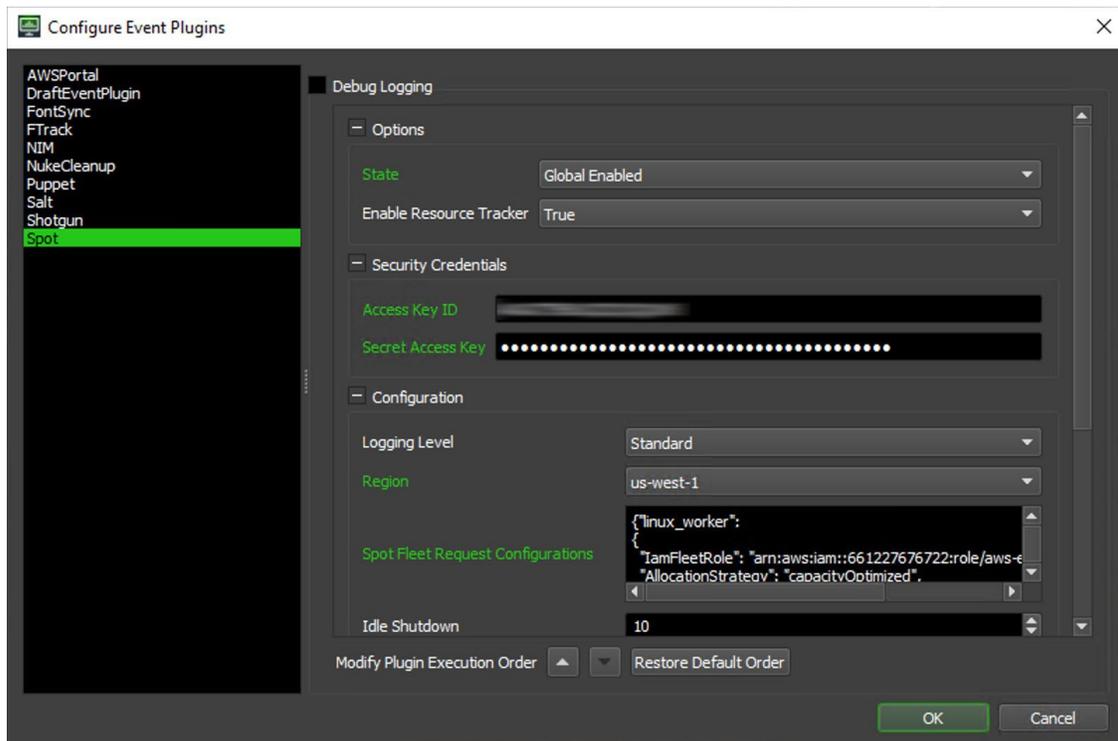
- And this line after:

```
}
```

- 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::468487574104:role/aws-ec2-spot-fleet-tagging-role",
    "AllocationStrategy": "capacityOptimized",
    "TargetCapacity": 1,
    "ValidFrom": "2019-08-21T00:26:20Z",
    "ValidUntil": "2020-08-21T00:26:20Z",
    "TerminateInstancesWithExpiration": true,
    "LaunchSpecifications": [],
    "Type": "maintain",
    "LaunchTemplateConfigs": [
      {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-07250a4807df997d7",
          "Version": "7"
        },
        "Overrides": [
          {
            "InstanceType": "m5.2xlarge",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-08cc6e4b19e8748eb"
          },
          {
            "InstanceType": "t3a.2xlarge",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-08cc6e4b19e8748eb"
          },
          {
            "InstanceType": "t3.2xlarge",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-08cc6e4b19e8748eb"
          },
          {
            "InstanceType": "t2.2xlarge",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-08cc6e4b19e8748eb"
          },
          {
            "InstanceType": "m3.2xlarge",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-08cc6e4b19e8748eb"
          },
          {
            "InstanceType": "r5d.2xlarge",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-08cc6e4b19e8748eb"
          },
          {
            "InstanceType": "r5a.2xlarge",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-08cc6e4b19e8748eb"
          },
          {
            "InstanceType": "r4.2xlarge",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-08cc6e4b19e8748eb"
          }
        ]
      }
    ]
  }
}
```

- 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 a **managed policy** to your IAM user profile.

- *Note: You may try to add this as an **inline policy**, but you won't be able to due to limits. Instead, create a **Managed Policy** and **attach** it to the user.*
- Go to the **Console** and open your **IAM** settings by going **Services** → **IAM**.
- In the lefthand navigation pane click on **Policies**.
- Click on **Create Policy**.
- Choose the **JSON** tab.
- <shift>+click on the image below to open a new browser tab with the text that needs to be entered into the JSON entry field:

```
https://studio-in-the-cloud-tutor x +
studio-in-the-cloud-tutorials.s3-us-west-1.amazonaws.com/tutorial-6/snippets/deadline_spot-event-policy_01.json

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "HouseCleaningStatement",
      "Effect": "Allow",
      "Action": [
        "ec2:ModifySpotFleetRequest",
        "ec2:CancelSpotFleetRequests",
        "ec2:RequestSpotFleet",
        "ec2:DescribeSpotFleetRequests",
        "ec2:DescribeSpotFleetInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles",
        "iam:PassRole",
        "iam:ListInstanceProfiles",
        "iam:GetRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy",
        "iam:DetachRolePolicy",
        "iam>DeleteRole",
        "iam>DeleteRolePolicy",
        "iam:GetRolePolicy",
        "iam:GetUser"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/*"
    }
  ]
}
```

**deadline\_spot-event-policy\_01.json** - <shift>+click on the image above to open the JSON file in a new tab

- Replace the existing text with the text from the above file.
- When you're all done, the JSON entry field should look like this:

```
Visual editor  JSON  Import managed policy

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "HouseCleaningStatement",
6       "Effect": "Allow",
7       "Action": [
8         "ec2:ModifySpotFleetRequest",
9         "ec2:CancelSpotFleetRequests",
10        "ec2:RequestSpotFleet",
11        "ec2:DescribeSpotFleetRequests",
12        "ec2:DescribeSpotFleetInstances"
13      ],
14      "Resource": "*"
15    },
16    {
17      "Effect": "Allow",
18      "Action": [
19        "iam:ListRoles",
20        "iam:PassRole",
21        "iam:ListInstanceProfiles",
22        "iam:GetRole",
23        "iam:CreateRole",
24        "iam:AttachRolePolicy",
25        "iam:PutRolePolicy",
26        "iam:DetachRolePolicy",
27        "iam>DeleteRole",
28        "iam>DeleteRolePolicy",
29        "iam:GetRolePolicy",
30        "iam:GetUser"
31      ],
32      "Resource": "*"
33    },
34    {
35      "Effect": "Allow",
36      "Action": "iam:CreateServiceLinkedRole",
37      "Resource": "arn:aws:iam:*:role/aws-service-role/*"
38    },
39    {
40      "Sid": "CloudFormationPermissions",
41      "Effect": "Allow",
42      "Action": [
43        "cloudformation:CreateStack",
44        "cloudformation>DeleteStack",
45        "cloudformation:DescribeStacks",
46        "cloudformation:ListStacks",
47        "cloudformation:UpdateTerminationProtection"
48      ],
49      "Resource": [
50        "*"

```

```
51     ]
52   },
53   {
54     "Sid": "DynamoDBPermissions",
55     "Effect": "Allow",
56     "Action": [
57       "dynamodb:CreateTable",
58       "dynamodb:DescribeTable",
59       "dynamodb>DeleteTable",
60       "dynamodb:TagResource",
61       "dynamodb:UntagResource",
62       "dynamodb:BatchWriteItem",
63       "dynamodb:ListTagsOfResource",
64       "dynamodb:Scan"
65     ],
66     "Resource": "*"
67   },
68   {
69     "Sid": "SQSPermissions",
70     "Effect": "Allow",
71     "Action": [
72       "sqs:CreateQueue",
73       "sqs:GetQueueAttributes",
74       "sqs>DeleteQueue",
75       "sqs:ListQueueTags",
76       "sqs:UntagQueue",
77       "sqs:TagQueue"
78     ],
79     "Resource": "*"
80   },
81   {
82     "Sid": "LambdaPermissions",
83     "Effect": "Allow",
84     "Action": [
85       "lambda:GetFunction",
86       "lambda:CreateFunction",
87       "lambda>DeleteFunction",
88       "lambda:GetFunctionConfiguration",
89       "lambda:CreateEventSourceMapping",
90       "lambda:GetEventSourceMapping",
91       "lambda>DeleteEventSourceMapping",
92       "lambda:AddPermission"
93     ],
94     "Resource": "*"
95   },
96   {
97     "Sid": "EventPermissions",
98     "Effect": "Allow",
99     "Action": [
100       "events:PutRule",
```

```
101       "events:DescribeRule",
102       "events:RemoveTargets",
103       "events>DeleteRule",
104       "events:PutTargets"
105     ],
106     "Resource": "*"
107   },
108   {
109     "Sid": "S3Permissions",
110     "Effect": "Allow",
111     "Action": [
112       "s3:GetObject"
113     ],
114     "Resource": [
115       "*"
116     ]
117   },
118   {
119     "Sid": "AutoScalingPermissions",
120     "Effect": "Allow",
121     "Action": [
122       "application-autoscaling:DescribeScalableTargets",
123       "application-autoscaling:RegisterScalableTarget",
124       "application-autoscaling:DeregisterScalableTarget",
125       "application-autoscaling:DescribeScalingPolicies",
126       "application-autoscaling:PutScalingPolicy",
127       "application-autoscaling>DeleteScalingPolicy"
128     ],
129     "Resource": "*"
130   }
131 ]
132 }
```

- Click **Review Policy**.
- On the new page, name the policy **DeadlineSpotEventCredentials**.

Review policy

Name\*

Use alphanumeric and '+,=,@,\_' characters. Maximum 128 characters.

Description

- Click **Create Policy**.
- Now **Attach** the policy to the user by going to **Users**.
- Choose the User for your account.
- Click **Add Permissions**.
- Click **Attach existing policies directly**.

## Add permissions to jason

### Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

 Add user to group

 Copy permissions from existing user

 Attach existing policies directly

- Search for **DeadlineSpotEventCredentials**.
- Click the checkbox next to **DeadlineSpotEventCredentials**.

Filter policies

	Policy name
<input checked="" type="checkbox"/>	DeadlineSpotEventCredentials

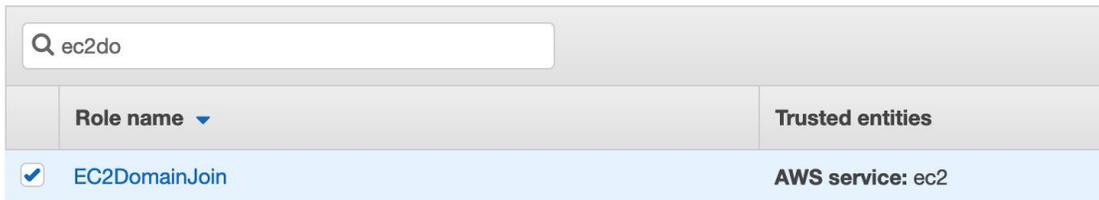
- Click **Next:Review**.
- Click **Add permissions**.

### Update EC2DomainJoin Instance Profile

Remember back in Tutorial 2 where we created an IAM Profile for your instances called EC2DomainJoin? Now it's time to modify that profile to give it a few more permissions.

- In the lefthand navigation pane, click on **Roles**.

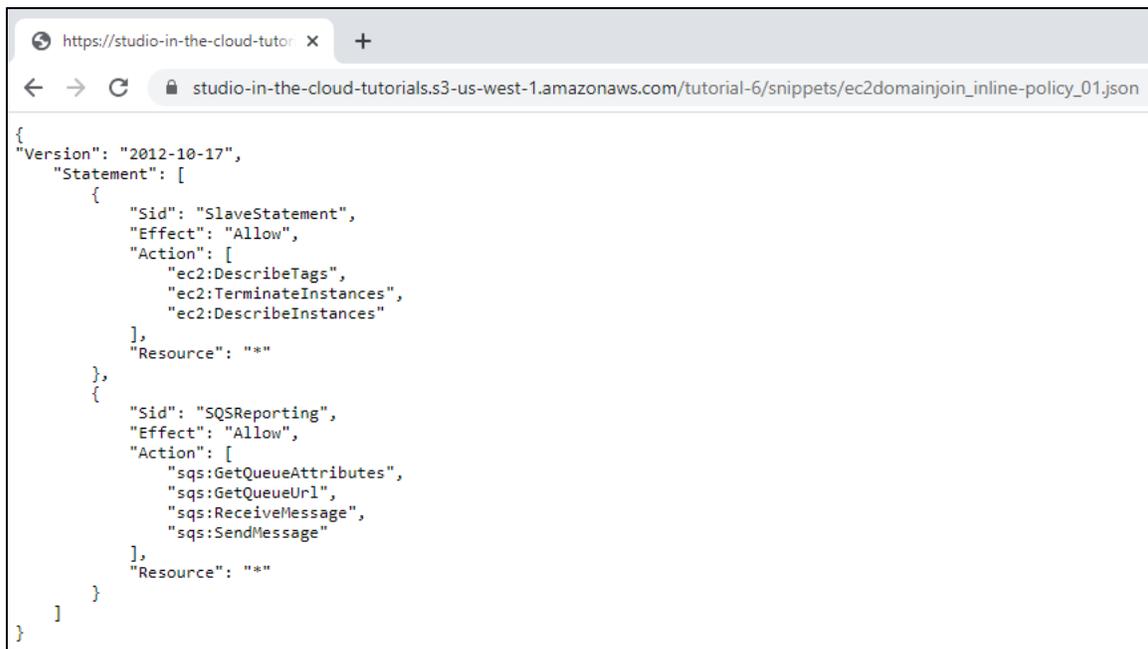
- In the search field, start to type in **EC2DomainJoin**.
- Click on **EC2DomainJoin** in the list.



- Click on **Add inline policy**.



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



**ec2domainjoin\_inline-policy\_01.json** - <shift>+click on the image above to open the JSON file in a new tab

- Replace the existing text with the text from the above file.
- When you're all done, the JSON entry field should look like this:

```

Visual editor  JSON  Import managed policy
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "SlaveStatement",
6       "Effect": "Allow",
7       "Action": [
8         "ec2:DescribeTags",
9         "ec2:TerminateInstances",
10        "ec2:DescribeInstances"
11      ],
12      "Resource": "*"
13    },
14    {
15      "Sid": "SQSReporting",
16      "Effect": "Allow",
17      "Action": [
18        "sqs:GetQueueAttributes",
19        "sqs:GetQueueUrl",
20        "sqs:ReceiveMessage",
21        "sqs:SendMessage"
22      ],
23      "Resource": "*"
24    }
25  ]
26 }

```

- Click **Review Policy**.
- Name it **WorkerSpotPolicy**.
- Click **Create Policy**.

### 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.

- In the lefthand navigation pane, click on **Roles** again.
- Click the **Create role** button.
- Under **Select type of trusted entity**, choose **AWS service**.
- Under **Choose a use case**, choose **Lambda** and then choose **Next: Permissions**.

Select type of trusted entity

 <b>AWS service</b> <small>EC2, Lambda and others</small>	 <b>Another AWS account</b> <small>Belonging to you or 3rd party</small>	 <b>Web identity</b> <small>Cognito or any OpenID provider</small>	 <b>SAML 2.0 federation</b> <small>Your corporate directory</small>
---	--	--	---

Allows AWS services to perform actions on your behalf. [Learn more](#)

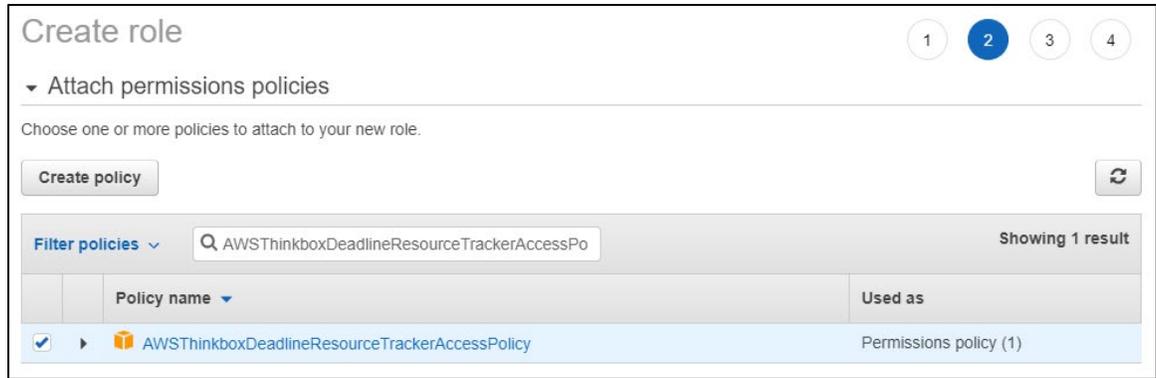
Choose a use case

**Common use cases**

**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

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



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

## Test the Fleet

Now that you've set up the event plugin, it's a great opportunity to test and make sure scaling works as expected.

- Log into your **Workstation** instance and load the scene you tested with earlier in **Blender**.
- 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.
- If you want to check on the Spot request, go to **Services** → **EC2** and then click on **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.

Request Id	Request type	Instance type	State	Capacity	Status	Persistence	Created	Max price
sir-73zif6jk	instance	m3.2xlarge	active	i-04142de48d17...	fulfilled	persistent	a minute ago	\$0.616
sfr-41abd348-d5dd...	fleet	m5.2xlarge.m3...	active	0 of 1	pending_fulfill...	maintain	a minute ago	

- 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**. Then click on **Spot** and check the **Region** setting.
- 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.

Request Id: sfr-41abd348-d5dd-42df-8dea-cc782aca343f

Description Instances History Savings Auto Scaling Scheduled Scaling

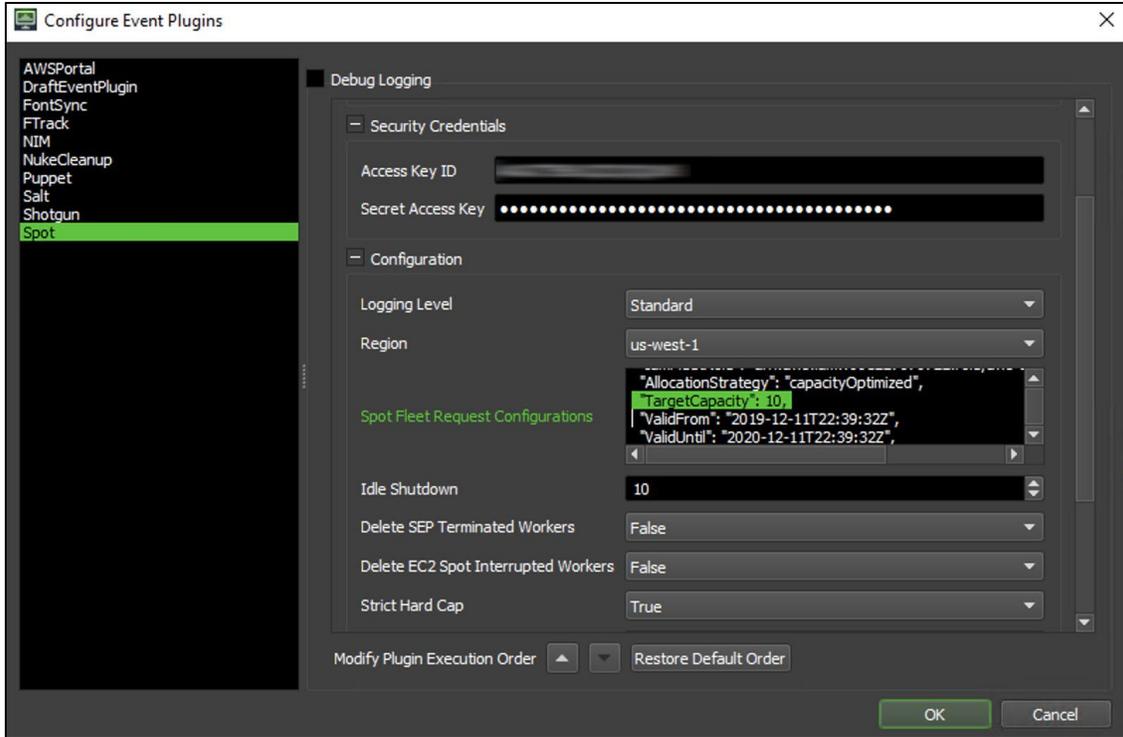
History saved for 48 hours only. Updates may take up to 1 minute to display.

Timestamp	Event Type	Status	Description	Instance Id
12/11/2019, 2:58:06 PM	fleetRequestChange	progress	m3.2xlarge, ami-0ac06a88c1b5262ac, Linux/UNIX, us-west-1-c, capacityUnitsRequested: 1.0, totalCapacityUnitsRequested: 1.0, totalCapacityUnitsFulfilled: 1.0, targetCapacity: 1	
12/11/2019, 2:58:06 PM	instanceChange	launched	{"instanceType": "m3.2xlarge", "image": "ami-0ac06a88c1b5262ac", "productDescription": "Linux/UNIX", "availabilityZone": "us-west-1-c"}	i-04142de48d17d432d
12/11/2019, 2:58:04 PM	fleetRequestChange	active		
12/11/2019, 2:57:54 PM	fleetRequestChange	submitted		

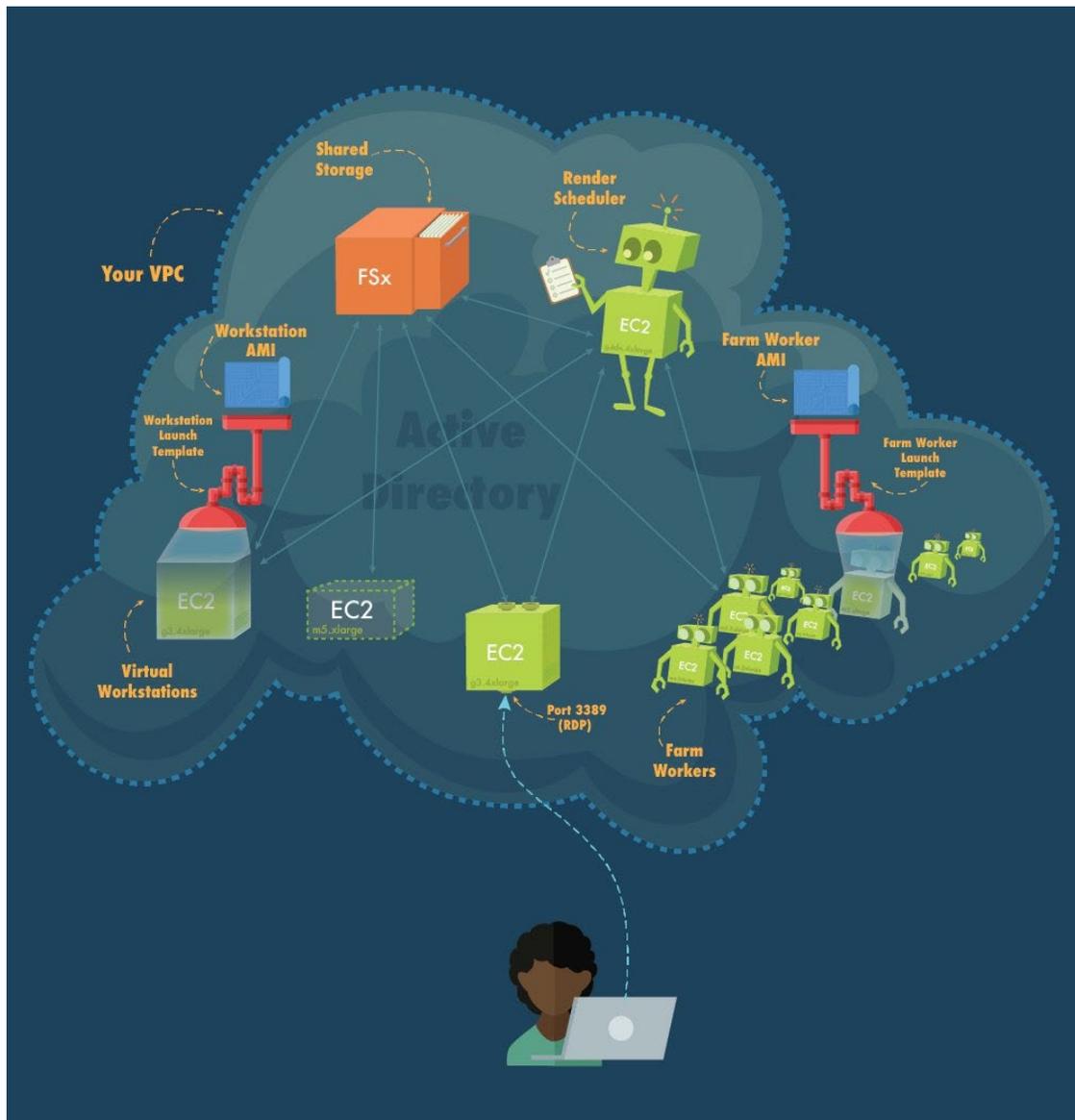
- Once the status of the Spot request has changed from pending fulfillment to **fulfilled**, you should see a new worker pop up in the Worker list in the Deadline Monitor and your frames should start rendering.

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



## 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, please 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

### Tutorial Table of Contents

[Tutorial 1: Getting Started with AWS Virtual Workstations](#)

[Tutorial 2: Creating a VPC and Active Directory for Your Studio](#)

[Tutorial 3: Setting Up an FSx File System and User Accounts](#)

[Tutorial 4: Building a Render Scheduler with AWS Thinkbox Deadline](#)

[Tutorial 5: Installing Applications and Creating a Workstation AMI](#)

[Tutorial 6: Setting Up a Linux Farm Worker and Spot Fleet Request – \*this page\*](#)

[Tutorial 7: Onboarding New Artists and Sample Workflow](#)

### Links to AWS Documentation

- [Deadline Spot Event Plugin Documentation](#)
- [Where's My Secret Access key?](#)

### Downloads

- [Important Information Cheat Sheet](#)
- [Spot Event Policy](#)