

Amazon Nimble Studio Classic

<-->

--- [-] -- {~w~} [...]

[~w~] -- <--> [...] -- {...} -- [~w~] -- <-->

{...} [<-->] ~w <--> ... [->] -- {...} [<-->] ~w <-->

-- [~w~] [... <--> -- -- {...} -- [~w~] [... <-->]

-- [-] -- {~w~} [...] ... <--> -- -- [-] -- {~w~} [...]

<--> -- -- {~w~} [...] -- [-] ... <--> -- -- {~w~} [...]

--- [-] -- {~w~} [...] ... <--> -- -- [-] -- {~w~} [...]

<--> [...] -- {...} -- [~w~] -- <--> [...]

^ -- {...} [<-->] ~w <-->

Amazon Nimble Studio Classic: Classic Guide

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

.....	xvi
What is Amazon Nimble Studio?	1
Supported software and operating systems	1
Remote team management	2
Security with Nimble Studio	2
Pricing for Nimble Studio	2
Setup and deployment	2
Storage for Nimble Studio	3
How to get started with Nimble Studio	3
Concepts and terminology	4
Key features	4
Key concepts and terminology	5
Availability Zones	12
Local Zones	13
Setting up	14
Set up IAM	14
Sign up for an AWS account	14
Create an administrative user	15
Related resources	16
Account security	16
Delete your account's access keys	16
Enable multi-factor authentication	17
Enable CloudTrail in all AWS Regions	17
Set up Amazon GuardDuty and notifications	18
Check AWS service quotas	20
Check your Spot Instance quota	20
Request a quota increase	21
Availability Zone ID	22
Request a quota increase for Amazon Nimble Studio streaming sessions	22
(Optional) Request a quota increase for G5 streaming sessions	23
Request a quota increase for VPC security groups per network interface	25
Request a quota increase for On-Demand Instances (G and VT)	25
(Optional) Opt in to the LA Local Zone	26
Getting started	27

How StudioBuilder works	28
How does StudioBuilder work?	28
What resources does StudioBuilder create?	29
Troubleshooting	38
Related resources	39
Deploy a new studio	39
Prerequisites	40
Step 1: Enable IAM Identity Center	42
Step 2: Access the StudioBuilder AMIs	45
Step 3: Launch the StudioBuilder EC2 instance	46
Step 4: Configure studio with StudioBuilder	48
Step 5: Deploy studio with StudioBuilder	59
Step 6: Link AWS Managed Microsoft AD as an IAM Identity Center identity source	61
Step 7: Confirm subscription to burst alert emails	64
Troubleshooting	65
Related resources	68
Delete a studio	69
Prerequisites	69
Step 1: Delete your Nimble Studio cloud studio	71
Step 2: Remove IAM Identity Center	72
Step 3: Remove storage	73
Step 4: Delete Amazon EC2 resources	73
Step 5: Remove Active Directory	74
Step 6: Remove Deadline Database	75
Step 7: Delete S3 buckets	75
Step 8: Delete CloudWatch Logs	76
Step 9: Remove CloudFormation stacks	76
Administration	77
Create launch profiles	77
Prerequisites	78
Step 1: Create a launch profile by copying an existing one	78
Step 2: (Optional) Perform a test launch	81
Step 3: Share launch profiles with studio users	83
Troubleshooting	87
Related resources	88
Modify launch profiles	88

Prerequisites	88
Update launch profile	88
Validate launch profile	89
Remove launch profile	90
Create custom configurations	90
Prerequisites	91
Step 1: Create the custom configuration	91
Step 2: Attach custom configuration to a launch profile	93
Custom configuration examples	93
Back up studio data	102
Back up option 1: Copy data to S3 bucket from a virtual workstation	102
Back up option 2: Copy data to S3 bucket using AWS DataSync	105
Back up option 3: Create Amazon FSx backups	112
Update StudioBuilder	112
Prerequisites	113
Step 1: Launch new instance with the latest StudioBuilder version	113
Step 2: Check updates to your launch profiles	114
Step 3: Update your studio	115
Step 4: Compare launch profiles after update is complete	117
Troubleshooting	119
Related resources	119
Change administrator password or policies	120
Prerequisites	120
Reset an administrator password for AWS Directory Service	120
Using password policies for AWS Directory Service	122
Supported policy settings for AWS Managed Microsoft AD	123
Rotate certificates	124
Prerequisites	125
Rotate the root CA	125
Rotate the render queue certificate	133
User management	134
Add studio users	134
Prerequisites	135
Step 1: Sign in to Nimble Studio portal as Admin	135
Step 2: Accept the EULA	137
Step 3: Launch a virtual workstation	137

Step 4: Add users to AWS Managed Microsoft AD	141
Step 5: Sync Active Directory and users in IAM Identity Center	146
Step 6: Add users to Nimble Studio	146
Troubleshooting	147
Related resources	147
Remove studio users	147
Step 1: Sign in to the portal as an administrator	148
Step 2: Run PowerShell commands to remove users	154
Workstations	155
Start and stop workstations	155
Workstation states	156
Start, stop, or terminate a workstation	159
Set streaming limits	159
Turn off persistent workstations	160
Enable uploads	161
Enable uploads by configuring a launch profile	161
Session auto backup	162
Prerequisites	162
Turn on session auto backup	163
Restore from backup	164
Delete EBS volumes	166
Create streaming session using API	167
Prerequisites	168
Using Python commands with API	168
Step 1: Prepare CloudShell	169
Step 2: Start Python 3 and import modules	172
Step 3: Create variables	173
Step 4: Create a Boto3 session	174
Step 5: Create Directory Service client	175
Step 6: Create IdentityStore client and get user identity	176
Step 7: Create Nimble client and spin up streaming session	177
Step 8: Get sessionId for a streaming session	183
Step 9: Get the current state of streaming session	183
Step 10: Terminate a streaming session	184
Troubleshooting	185
Related Resources	186

Provision workstations in multiple AZs or LZs	186
Prerequisites	187
Step 1: Create a workstation subnet in the new Availability Zone or Local Zone	189
Step 2: Create a new file systems subnet in the new Availability Zone	191
Step 3: Update NACLs	192
Step 4: Create a new FSx drive for Userprofiles for Windows	195
Step 5: Add the Amazon FSx drive as a storage component	196
Step 6: Create a new launch profile	197
Step 7: Update launch profile	197
Step 8: (Optional) Verify launch profile configuration	202
Administrator access for Windows users	203
Prerequisites	204
Step 1: Add an administrator access component to your studio	204
Step 2: Add the administrator Access component to a launch profile	205
Step 3: Test administrator Access component	206
Superuser access for Linux users	210
Prerequisites	91
Step 1: Add a superuser access component to your studio	211
Step 2: Add the Sudo Access component to a launch profile	213
Step 3: Test Sudo Access component	213
Storage	220
Set up Amazon FSx Windows	220
Prerequisites	221
Step 1: Create a new file system	221
Step 2: Attach the file system to the studio	223
Step 3: Update launch profiles	224
Set up FSx for Lustre	225
Prerequisites	225
Step 1: Create a new file system	225
Step 2: Attach the file system to the studio	228
Step 3: Attach a new component to each launch profile	229
Step 4: Change permissions on the Lustre mount point	231
Step 5: Sign in to your operating system	235
Set up Linux home directories	235
Prerequisites	237
Step 1: Create security groups for your shared file system and Nimble Studio	237

Step 2: Create a file system with Amazon Elastic File System (Amazon EFS)	239
Step 3: Update file system subnet network ACLs	241
Step 4: Create a custom studio resource	242
Step 5: Add the custom resource to a launch profile	244
Set up Weka	244
Prerequisites	245
Step 1: Gather information	245
Step 2: Prepare the network	248
Step 3: Deploy Weka using CloudFormation	250
Step 4: Update security groups	254
Step 5: Linux setup	255
Step 6: (Optional) Windows setup	261
Troubleshooting	268
Software setup and updates	269
Update AMIs	269
Prerequisites	270
General setup	272
Update AMIs for your operating system	277
Software-specific instructions	277
Troubleshooting	277
Related resources	278
Update Windows workstation AMI	278
Prerequisites	279
Step 1: Create a customer managed key	279
Step 2: Launch an instance with Windows AMI	280
Step 3: Restart the NICE DCV session as an administrator	284
Step 4: Connect with NICE DCV	284
Step 5: Download and run installers	286
Step 6: Prepare your instance for AMI creation	288
Step 7: Create a new AMI	291
Step 8: Update launch profiles	292
Step 9: Update the DCV server	293
Troubleshooting	293
Related resources	294
Update Linux workstation AMI	294
Prerequisites	295

Step 1: Launch an instance with Linux workstation AMI	295
Step 2: Connect with Session Manager	298
Step 3: Update software from Session Manager	299
Step 4: (Optional) Set up and connect with NICE DCV	301
Step 5: Prepare the virtual workstation and create an AMI	305
Step 6: Create a customer managed key	308
Step 7: Encrypt the AMI	309
Step 8: Update launch profiles	310
Troubleshooting	310
Related resources	310
Update Windows worker AMI	311
Prerequisites	312
Step 1: Prepare your test environment	312
Step 2: Launch an instance for AMI creation	323
Step 3: Connect to the TEST worker with Remote Desktop	325
Step 4: Download and run installers	326
Step 5: Validate the update	328
Step 6: Update the AMIBUILD worker instance	328
Step 7: Prepare your instance for AMI creation	329
Step 8: Create a new AMI	332
Step 9: Use StudioBuilder to update your render farm fleet	333
Step 10: Test your deploy	335
Step 11: Terminate the Worker_AMIBUILD and Worker_TEST instances	336
Troubleshooting	336
Update Linux worker AMI	336
Prerequisites	337
Step 1: Prepare your test environment	338
Step 2: Launch an instance for AMI creation	349
Step 3: Update the TEST worker	351
Step 4: Validate the update	352
Step 5: Update the AMIBUILD instance	353
Step 6: Create the new AMI	353
Step 7: Use StudioBuilder to update your render farm fleet	355
Step 8: Test your deploy	357
Step 9: Terminate the Worker_AMIBUILD and Worker_TEST instances	357
Troubleshooting	358

Software specific installation	358
Adobe Creative Cloud	359
Autodesk	359
Black Magic Design	359
Chaos group	360
Foundry	360
NVIDIA	360
SideFX	360
Substance	360
Adobe Creative Cloud	361
Blackmagic Design DaVinci Resolve 17	366
Perforce Helix Core	371
Incredibuild	374
License servers	381
Create license server	381
Set up Nuke license server	395
Set up RLM license server	402
Rendering with Nimble Studio	409
Related resources	409
Configure Deadline	409
Prerequisites	410
Step 1: Sign in to Nimble Studio portal as Admin	410
Step 2: Accept the EULA	413
Step 3: Launch a virtual workstation	415
Step 4: Open Deadline and create groups	419
Step 5: Adjust worker settings	421
Step 6: (Windows only) Set mapped paths	422
Related resources	425
Work with render farms	425
Set up Deadline UBL	426
Mount Deadline Repository on Linux	436
Create your first render	442
Prerequisites	443
Step 1: Launch Blender	443
Step 2: Create a Blender scene	444
Step 3: Save your scene to shared storage	445

Step 4: Configure the scene to render on the farm	449
Step 5: Enable AWSThinkboxDeadline submitter add-on	455
Step 6: Submit render to Deadline	456
Step 7: Check progress in Deadline Monitor	459
Related resources	469
Delete a render farm	469
Step 1: Remove compute farm studio component	470
Step 2: Update to latest StudioBuilder	470
Step 3: Delete your render farm	470
Artist tutorials	472
Sign in to the portal	472
Prerequisites	473
Sign in to Nimble Studio portal for the first time	473
Log out of Nimble Studio portal	478
Launch virtual workstation	479
Prerequisites	480
Step 1: Launch a virtual workstation	480
Step 2: Sign in to the virtual workstation	482
Troubleshooting	484
Related resources	484
Start and stop workstations	484
Stopping a workstation	485
Upload files	486
Upload files to your virtual workstation	487
Session auto backup	490
Prerequisites	491
Restore from backup	491
Testing using Maya	493
Prerequisites	494
Step 1: Downloading assets	494
Step 2: Setting up the scene	496
Step 3: Rendering the asset	501
Providing Feedback	507
Related Resources	17
Testing using Blender	507
Prerequisites	507

Step 1: Downloading assets	508
Step 2: Setting up the scene	509
Step 3: Rendering an image	512
Providing Feedback	514
Related Resources	514
Testing using Blender - Shockingly Fuzzy	514
Prerequisites	515
Step 1: Downloading assets	515
Step 2: Setting up the scene	516
Step 3: Simulating the cloud and hair	520
Step 4: Rendering an image sequence	522
Providing feedback	523
Related resources	523
Migrate studio	524
Migrate EBS volume data	524
Getting started	525
Option 1: Manually copy data to an Amazon WorkDocs Drive	525
Option 2: Copy files from EBS volume to Amazon FSx or Amazon EFS	526
Option 3: (EC2 only) Create AMIs from a streaming session	527
Option 4: (EC2 only) Turn on session auto backup and export snapshots	527
Export AMIs and backups	528
Export streaming session AMI	529
Export streaming session backup	531
Set up instances like workstations	534
Step 1: Prepare the subnets	535
Step 2: Prepare the security groups	535
Step 3: (Linux Only) Prepare AWS Managed Microsoft AD for a seamless domain join	536
Step 4: Prepare the AMI	536
Step 5: Create an EC2 instance	537
Step 6: Validate that your instance is joined to the Active Directory	538
Troubleshooting	539
Security	540
More Information	540
Data protection	541
Encryption at rest	542
Encryption in transit	543

Key management for Amazon Nimble Studio	544
Data security measures	545
Diagnostic data and metrics	546
Identity and Access Management	546
Audience	547
Authenticating with identities	547
Managing access using policies	550
How Amazon Nimble Studio works with IAM	552
ID-based policy examples	558
AWS managed policies	561
Cross-service confused deputy prevention	570
Troubleshooting	571
Logging and monitoring	574
Logging Nimble Studio calls using AWS CloudTrail	574
Compliance validation	580
Resilience	581
Infrastructure security	582
Network connectivity security model	582
Security best practices	583
Monitoring	583
Data protection	583
Permissions	584
Internet network traffic privacy	584
Traffic between AWS resources in the same AWS Region	584
Network ACLs	585
Prerequisites	585
What is a network ACL?	585
Network ACL rules	585
Default network ACL	586
Configuration and vulnerability analysis	591
Patching software	592
Monitoring	594
Monitoring with CloudWatch	595
Monitoring examples	596
Related resources	598
Troubleshooting	599

Troubleshooting session unavailability	599
Troubleshooting render farm issues	600
Streaming instances can't submit new render jobs	600
Troubleshooting home directory issues	601
Multiple Linux home directories sharing the same mount points	601
Deadline render <code>tls_cert.crt</code> isn't found after updating studio	601
Troubleshooting system initialization scripts	602
Linux system initialization script failure	602
Service quotas	605
To view service quotas	605
To request a quota increase	605
Support	607
AWS Support Center	607
Nimble Studio forum	607
Nimble Studio help page	607
AWSThinkboxDeadline Documentation	608
AWS Premium Support plans	608
Release notes	609
AMI release notes	609
Nimble Studio Windows 2022 workstation AMI	610
Nimble Studio Windows 2019 workstation AMI	612
Nimble Studio Linux workstation AMI	619
Nimble Studio Windows 2022 worker AMI	626
Nimble Studio Windows 2019 worker AMI	629
Nimble Studio Linux worker AMI	633
StudioBuilder v1.1.13 2023-05-12	637
Important notes	638
Updates	638
StudioBuilder v1.1.12 2023-01-25	638
Important notes	639
Updates	639
StudioBuilder v1.1.11 2022-12-08	639
Important notes	640
Updates	640
StudioBuilder v1.1.10 2022-10-18	641
Important notes	641

Updates	642
StudioBuilder v1.1.9 2022-07-27	642
Important notes	642
Updates	643
StudioBuilder v1.1.8 2022-07-08	643
Important notes	643
Updates	644
StudioBuilder v1.1.7 2022-04-22	645
Important notes	645
Updates	646
StudioBuilder v1.1.6 2022-03-28	646
Important notes	647
Updates	647
StudioBuilder v1.1.5 2021-12-10	648
Important notes	648
Updates	649
StudioBuilder v1.1.4 2021-10-06	650
Important notes	650
Updates	651
Bug fixes	651
StudioBuilder v1.1.3 2021-07-22	652
Important notes	650
Updates	653
Bug fixes	653
Document History	654
AWS Glossary	661

Nimble Studio workstations are no longer onboarding new customers. Existing customer workstations won't be available after **June 19th, 2024**. We recommend that you migrate your existing resources by following the instructions in [Migrating your studio](#).

What is Amazon Nimble Studio?

Welcome to the classic guide for Amazon Nimble Studio.

Amazon Nimble Studio is an AWS service that builds the infrastructure that you need to operate a cloud-based studio for producing visual effects (VFX), animation, and interactive content. Nimble Studio provides virtual workstations, cloud rendering, and shared storage to content creators so they can quickly scale resources to meet an increased demand for content.

With Nimble Studio, production studios can leverage a global workforce securely in the cloud, thus reducing the costs of added physical infrastructure and technical staff. Using state-of-the-art AWS security, your admins can keep your valuable studio resources secure, while allowing your teams to access the tools they need, such as scalable high-speed storage, licenses, and near-limitless rendering.

Contents

- [Supported software and operating systems](#)
- [Remote team management](#)
- [Security with Nimble Studio](#)
- [Pricing for Nimble Studio](#)
- [Setup and deployment](#)
- [Storage for Nimble Studio](#)
- [How to get started with Nimble Studio](#)

Supported software and operating systems

You can use Linux or Windows operating systems (OS) with Nimble Studio. Most user-based credential logins from Foundry, Autodesk, and Adobe can be used.

Nimble Studio supports bringing your own floating licenses for industry software. This includes industry standard digital content creation applications such as Houdini, Nuke, Maya, Vray, and many more. Nimble Studio doesn't support node-locked licenses.

Blender is included on Nimble Studio workstation Amazon Machine Images (AMIs). For Farm worker AMIs, you can find Nuke, Houdini, and Blender in the [AWS Marketplace](#).

Remote team management

Rapidly onboard and collaborate with artists remotely and securely with Nimble Studio. Our scalable service will help you meet the demands of animation and VFX production teams who might not be cloud technology experts. An added benefit of a remote studio and workforce is that it can reduce production costs and upfront capital investment, while you continue to grow.

Security with Nimble Studio

Amazon Nimble Studio's state-of-the-art AWS security gives your account admins and project owners the ability to add or remove artists, assign resources, and share projects.

AWS IAM Identity Center (IAM Identity Center) provides secure artist access to web identities in the Nimble Studio portal. Nimble Studio portal includes workstation and file system access control via AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD), enabling directory-aware workloads for your production security needs.

Pricing for Nimble Studio

Pricing for Nimble Studio is primarily based on usage. To calculate your estimated costs for running a cloud studio, see [Nimble Studio Pricing](#).

Important

Even if you don't use your studio or any of the infrastructure that your studio creates, you will be charged for storage and other studio resources. If you aren't using your studio, we recommend deleting it so that you don't accrue unnecessary charges. For information about your AWS bill, see the [AWS Cost Explorer Service](#) and [AWS Budgets](#).

Setup and deployment

The beauty of Nimble Studio is that setup and deployment takes a few hours, not weeks.

StudioBuilder is Nimble Studio's deployment tool, and is accessible through the [AWS Marketplace](#).

StudioBuilder builds the infrastructure and pipeline that you need to start working on day one. Create your studio environment using the Nimble Studio portal, name new projects, and invite artists to onboard.

Here are some of the features you get with Nimble Studio:

- AWS IAM Identity Center (IAM Identity Center) for seamlessly connecting to remote teams, assets, and resources
- On-demand access to powerful virtual workstations using Amazon Elastic Compute Cloud (Amazon EC2)
- Integrated render farm through the AWS Thinkbox Render Farm Deployment Kit (RFDK)
- Amazon Machine Images (AMI) functionality and templates
- AWS Identity and Access Management (IAM) policies and roles
- Amazon FSx storage for Windows and Linux
- A virtual network through [Amazon Virtual Private Cloud](#)

Storage for Nimble Studio

Render workers on Nimble Studio are configured to access the shared storage for your studio, known as the Z: drive on your Windows virtual workstation or /mnt/fsxshare on a Linux virtual workstation.

Users also have a profile in the AWS Directory Service for Microsoft Active Directory. This is called a profile directory, and is usually C:\Users\your-user-name\ on Windows, or /home/your-user-name on Linux. A profile directory isn't accessible to other artists or render workers.

How to get started with Nimble Studio

After you've familiarized yourself with the [Concepts and terminology for Amazon Nimble Studio](#) page, visit the [Getting started with Amazon Nimble Studio](#) page. In it, you'll find links to helpful information and step-by-step instructions for deploying a new cloud studio and configuring it for your team.

If you're an artist in film, VFX, or interactive content, the [Artist tutorials for Amazon Nimble Studio](#) will show you how to launch a virtual workstation, and use Nimble Studio to collaborate and create remotely.

Concepts and terminology for Amazon Nimble Studio

To help you get started with Amazon Nimble Studio, and understand how it works, you can refer to the key concepts and terminology in this guide.

Key features

Amazon Nimble Studio

Amazon Nimble Studio is an AWS service that enables creative studios to produce visual effects, animation, and interactive content entirely in the cloud, from storyboard sketch to final deliverable.

Nimble Studio supports Linux and Windows operating systems (OS) and creation applications such as Autodesk Maya, Blender, Houdini, or Foundry's Nuke. It also integrates with many other AWS services.

Amazon Nimble Studio console

The **Nimble Studio console** is a portion of the **AWS Management Console** that is devoted to our administrator IT customers. This console is where admins create their cloud studio and manage many settings. For instance, the Studio manager page allows you to add or remove resources, add launch profiles, and grant permissions to users and groups.

Amazon Nimble Studio portal

The Nimble Studio portal is the user interface that's dedicated to both types of Nimble Studio customers: artists and admins. The **Nimble Studio portal** is where admins can assign launch profiles to artists, and artists can launch streaming sessions. The portal's user-friendly interface makes it easy to review your launch profiles, check your workstation's status, see who else is in their cloud studio, and access support.

StudioBuilder

StudioBuilder is a Cloud Development Kit (CDK) application that deploys a fully functional, secure cloud studio with Nimble Studio through a command line interface (CLI). After you follow a few prompts and configure some settings, StudioBuilder builds the infrastructure that your cloud studio needs to operate. The process takes about 90 minutes.

StudioBuilder is available through the [AWS Marketplace](#).

AWS Thinkbox Deadline

Deadline is rendering management software that provides a wide range of compute management options to easily and securely access cloud-based resources for rendering, render management, and processing. AWS Thinkbox Deadline is compatible with Windows, Linux, and macOS based render farms.

Key concepts and terminology

Amazon EC2 instance

An instance is a virtual server in the cloud. Its configuration is a copy of the Amazon Machine Image (AMI) that you specified when you launched the instance. To connect your virtual workstation to a streaming session, first launch an instance. You can do this from the Nimble Studio portal.

Amazon Machine Image (AMI)

An Amazon Machine Image (AMI) provides the information required to launch an instance. To run StudioBuilder and deploy your cloud studio, launch an Amazon EC2 instance using the AMI. An AMI has all of the packages that you need for the deploy.

AWS Managed Microsoft AD

AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) is a fully managed Active Directory in the AWS Cloud.

When StudioBuilder builds your cloud studio, it creates an Active Directory environment using the AWS Managed Microsoft AD service. After deployment, you can connect your new AWS Managed Microsoft AD to AWS IAM Identity Center (IAM Identity Center). To learn how, see [Step 6: Link AWS Managed Microsoft AD as an IAM Identity Center identity source](#) in the [Deploying a new studio with StudioBuilder](#) tutorial.

By connecting your AWS Managed Microsoft AD to IAM Identity Center, administrators can grant users or groups seamless access to the resources that you want them to use.

Customers who want to bring their own Active Directory should follow the **Bring Existing Resources (BER)** steps in the [Getting started](#) page of the [Nimble Studio console](#).

AWS managed policies

An AWS managed policy is a standalone policy that is created and administered by AWS. Standalone policy means that the policy has its own Amazon Resource Name (ARN) that includes the policy name. For example, arn:aws:iam::aws:policy/IAMReadOnlyAccess is an AWS managed policy. For more information about ARNs, see [IAM ARNs](#).

AWS managed policies are used for granting permissions to common job functions. Job function policies are maintained and updated by AWS when new services and API operations are introduced. For example, the **AdministratorAccess** job function provides full access and permissions delegation to every service and resource in AWS. Whereas, partial-access AWS managed policies such as AmazonMobileAnalyticsWriteOnlyAccess and AmazonEC2ReadOnlyAccess can provide specific levels of access to AWS services without allowing full access. For learn more about access policies, see [Understanding access level summaries within policy summaries](#).

AWS Management Console

The [AWS Management Console](#) is a web application that provides access to a broad collection of service consoles for managing AWS services.

Each service also includes its own console. These consoles offer a wide range of tools for cloud computing. For instance, within the EC2 console, you can create a license server, plus update or add new software to your Linux worker Amazon Machine Image (AMI). There's even a service that helps with [billing and cost management](#).

AWS IAM Identity Center (IAM Identity Center)

IAM Identity Center is an AWS service that makes it easy to centrally manage access to multiple AWS accounts and business applications. With IAM Identity Center, you can provide users with single sign-on access to all their assigned accounts and applications from one place. You can also centrally manage multi-account access and user permissions to all of your accounts in AWS Organizations. For more information, visit [AWS IAM Identity Center FAQs](#).

AWS Systems Manager Session Manager

Session Manager is a fully managed AWS Systems Manager capability that lets you manage your Amazon Elastic Compute Cloud (Amazon EC2) instances, on-premises instances, and virtual machines (VMs) through an interactive one-click browser-based shell or through the AWS Command Line Interface (AWS CLI). For more information, visit [AWS Systems Manager Session Manager](#).

Amazon Virtual Private Cloud (Amazon VPC)

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you have defined. This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure of AWS. Amazon Nimble Studio provides you with a [default VPC](#) during deployment.

Availability Zone (AZ)

Availability Zones are multiple, isolated locations within each AWS Region. An Availability Zone is represented by an AWS Region code followed by a letter identifier (example: us-east-1a).

With Amazon VPC, you can define a virtual network topology closely resembling a traditional network that you might operate on your own premises. Multi-AZ deployment provides high availability and fault tolerance. You can use Amazon VPC to span multiple Availability Zones. This enables you to place independent infrastructure in physically separate locations.

AWS PrivateLink

AWS PrivateLink provides private connectivity between VPCs, AWS services, and your on-premises networks, without exposing your traffic to the public internet. AWS PrivateLink makes it easy to connect services across different accounts and VPCs. [AWS PrivateLink](#) is available for a monthly fee that is billed to your AWS account.

AWS Virtual Private Network (AWS VPN)

AWS Virtual Private Network solutions establish secure connections between your on-premises networks, remote offices, client devices, and the AWS global network. AWS VPN is comprised of two services: AWS Site-to-Site VPN and AWS Client VPN. Each service provides a highly available, managed, and elastic cloud VPN solution to protect your network traffic.

AWS Site-to-Site VPN creates encrypted tunnels between your network and your VPCs or transit gateways. For managing remote access, AWS Client VPN connects your users to AWS or on-premises resources using a VPN software client.

Digital Content Creation (DCC)

Digital Content Creation (DCC) refers to the category of applications that are used to produce creative content, including Blender, Nuke, Maya, and Houdini.

Desktop Cloud Visualization (DCV) NICE DCV is a high-performance remote display protocol. It lets you securely deliver remote desktops and application streaming from any cloud or data center

to any device, over varying network conditions. By using NICE DCV with Amazon EC2, you can run graphics-intensive applications remotely on Amazon EC2 instances. For more information about the DCV client, see [NICE DCV clients](#).

End user license agreement (EULA)

A EULA is a contract between the manufacturer of computer software and the person who installs and uses the software.

Launch profile

A launch profile controls your artist workforce's access to studio components, like compute farms, shared file systems, managed file systems, and license server configurations, as well as instance types and Amazon Machine Images (AMIs).

Studio administrators create launch profiles in the Nimble Studio console. Artists can use their launch profiles to launch an instance from the Nimble Studio portal. Each user's launch profile defines how they can launch a streaming session. By default, studio admins can use all launch profiles.

License server

A license server manages licenses and issues out entitlements for all application software used in digital content creation (DCC). Examples of software licenses used by Nimble Studio customers include Maya, Nuke, Arnold, and Houdini. A license server requires a cryptographically signed license key file. The software verifies the file before serving licenses to other computers through a network protocol.

License service

A license service is a centralized computer software system for Nimble Studio that provides access tokens or keys to enable licensed software to run. With Nimble Studio, your license service can be used as a proxy or as the direct license server—but install your own license management software.

On-Demand Instances

With On-Demand Instances, you pay for compute capacity by the second, with no long-term commitments. You have full control over the lifecycle of the instance—you decide when to launch, stop, hibernate, start, reboot, or shut it down. You pay only for the seconds that your On-Demand Instances are in the running state. The price per second for On-Demand Instance is fixed and is listed on the Amazon EC2 Pricing, [On-Demand Pricing](#) page.

Regions

Nimble Studio offers six AWS Regions from which to choose your home Region. Users close to the home Region will experience faster speed and improved performance. For more information, see [Availability Zones for Amazon Nimble Studio](#).

To see the mapping of IDs to Availability Zones in your account, see [AZ IDs for Your Resources](#) in the AWS RAM User Guide.

Remote Connection Server (RCS) The Remote Connection Server (RCS) is encapsulated by the Render Queue construct. It's the service that sits behind the Application Load Balancer (ALB) that is set up by the Render Queue. During instantiation, the Render Queue generates a self-signed certificate that the RCS is configured to use for communication between itself and the ALB. For more information, see the [Render Farm Deployment Kit on AWS](#) developer guide.

Render Farm Deployment Kit on AWS (RFDK)

The Render Farm Deployment Kit (RFDK) on AWS is an open-source software development kit that can be used to deploy, configure, and manage your render farm infrastructure in the cloud. The RFDK is built to operate with the AWS Cloud Development Kit (AWS CDK) (AWS CDK) and provides a library of classes, called constructs, that each deploy and configure a component of your cloud-based render farm. The current version of the RFDK supports render farms that are built using AWS Thinkbox Deadline render management software.

Render queue

A render queue is the main, central service component of a render farm, where clients and workers connect and access any information that they require to set up a render. Render queues allow teams to control the order in which objects will be rendered.

Streaming image

A streaming image is a resource within Nimble Studio that represents an Amazon Machine Image (AMI), and is specifically configured to work with virtual workstations. A streaming image allows users to connect to their workstations via a [NICE DCV client](#).

Streaming session

A streaming session represents a virtual workstation that a user can connect to so that they can access the files, settings, and applications they need to work on an asset. Users can see the

streaming session listed in their Nimble Studio portal, where they can connect to the session and shut it down.

Studio

A studio is the top-level container for other Nimble Studio-related resources. Your cloud studio manages the Nimble Studio web portal and the connections to essential resources in your AWS account such as your VPC, user directory, and storage encryption keys.

Studio component

Studio components are configurations within a customer's Nimble Studio that tell the service how to access resources like file systems, license servers, and render farms in your AWS account.

Nimble Studio contains a number of subtypes of studio components including a shared file system, compute farm, Active Directory, and license component. These subtypes describe resources that you would like your studio to use.

Studio home Region (home Region)

A studio home Region is the AWS Region where essential studio infrastructure exists, such as your main Amazon S3 data and render farm.

Your home Region is where your core production data lives, so it's typically closest to where the core production is happening. A geographically distributed studio might select the home Region to be close to the majority of its creative workforce.

Studio resources

Studio resources is an industry term that encapsulates the things a studio needs in their daily operations. Studio artists often refer to their render farm, AWS Managed Microsoft AD, and file storage as resources. When describing how resources fit into the infrastructure of a cloud studio, they might be also referred to as studio components.

Subnet

A subnet is a range of IP addresses in your VPC. When a subnet's traffic is routed to an internet gateway, the subnet is known as a public subnet.

VPN-only subnet: If a subnet doesn't have a route to the internet gateway, but has its traffic routed to a virtual private gateway for a site-to-site VPN connection, the subnet is known as a VPN-only subnet.

Private subnet: A subnet that doesn't have a route to the internet gateway is known as a private subnet. For more information, see [Examples for VPC](#), [Internet gateways](#), and [What is AWS Site-to-Site VPN?](#) in the AWS Site-to-Site VPN User Guide.

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value that you define.

Tags enable you to categorize your AWS resources in different ways. For example, you could define a set of tags for your account's Amazon EC2 instances that help you track each instance's owner and stack level. Tags also enable you to integrate your organization's shared file systems and render farms with Nimble Studio, to keep your workflows uninterrupted while you move your workforce to the cloud.

With tags, you can categorize your AWS resources by purpose, owner, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags that you've assigned to it.

User-managed VPC

A user-managed VPC is a virtual private cloud (VPC) in your AWS account that you control. StudioBuider provides you with a default VPC in Amazon Virtual Private Cloud (Amazon VPC) during deployment.

Nimble Studio accesses and manages studio components in your AWS account. These components must be connected to the user-managed VPC to provide network connectivity to Nimble Studio and to each other.

Virtual workstation (workstation)

A virtual workstation is configured with all of the applications, tools, and data that an artist needs to do their work. To access their virtual workstation, the artist must use launch profiles that the administrator assigned to them, and launch a streaming session from the portal. After the streaming session starts, the artist can use the software applications, storage, and render farm that was configured for them in their works.

Availability Zones for Amazon Nimble Studio

Availability Zones are distinct locations that are engineered to be isolated from failures in other Availability Zones.

To coordinate Availability Zones across accounts, use the Availability Zone ID, which is a unique and consistent identifier for an Availability Zone. For example, `use1-az2` is an ID for the `us-east-1` Region and it has the same location in every AWS account.

Viewing Availability Zone IDs enables you to determine the location of resources in one account relative to the resources in another account. For example, if you share a subnet in the Availability Zone with the ID `use1-az2` with another account, this subnet is available to that account in the Availability Zone whose ID is also `use1-az2`. The ID for each VPC and subnet is displayed in the Amazon VPC console.

Amazon Nimble Studio is available in a subset of the Availability Zones for each supported Region. The following table lists the IDs that you can use for each Region. To see the mapping of IDs to Availability Zones in your account, see [AZ IDs for Your Resources](#) in the AWS RAM User Guide.

Region name	Region code	Region ID
US East (N. Virginia)	<code>us-east-1</code>	<code>use1-az2, use1-az4, use1-az6</code>
US West (Oregon)	<code>us-west-2</code>	<code>usw2-az1, usw2-az2, usw2-az3, usw2-lax1-az1</code>
Asia Pacific (Sydney)	<code>ap-southeast-2</code>	<code>apse2-az1, apse2-az3, apse2-az2</code>
Asia Pacific (Tokyo)	<code>ap-northeast-1</code>	<code>apne1-az1, apne1-az2, apne1-az4</code>
Canada (Central)	<code>ca-central-1</code>	<code>cac1-az1, cac1-az2</code>
Europe (London)	<code>eu-west-2</code>	<code>euw2-az1, euw2-az2, euw2-az3</code>

For more information about Availability Zones and Availability Zone IDs, see [Regions, Availability Zones, and Local Zones](#) in the Amazon EC2 User Guide for Linux Instances.

Local Zones

If your studio is being created, or already exists, in the us-west-2 Region, you can create Workstations and file systems in the LA Local Zone. We recommend this for lower latency interactions, if you and your artists are located closer to California than Oregon. For instructions about how to opt in to the LA Local Zone, see the [\(Optional\) Opt in to the LA Local Zone](#) tutorial. For more information about Local Zones, see the [AWS Local Zones FAQs](#).

Setting up to use Nimble Studio

This tutorial is for administrator users who want to set up an Amazon Nimble Studio.

The following sections will guide you through the steps that you need to complete before deploying a studio in Nimble Studio.

Important

The studio that you create to use with [File Transfer](#) isn't compatible with Nimble Studio workstations. If you create a studio to use with File Transfer and later decide you want to set up Nimble Studio managed workstations, you need to delete your studio and create a new studio through the Nimble Studio setup process.

Contents

- [Set up IAM](#)
- [Related resources](#)
- [Set up AWS account security](#)
- [Check AWS service quotas](#)
- [\(Optional\) Opt in to the LA Local Zone](#)

Set up IAM

Review the following AWS Identity and Access Management (IAM) documentation before you start.

- [Security best practices in IAM](#)
- Sign in to your AWS account as an administrator user to complete the remaining setup.

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open <https://portal.aws.amazon.com/billing/signup>.

2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, [assign administrative access to an administrative user](#), and use only the root user to perform [tasks that require root user access](#).

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to <https://aws.amazon.com/> and choosing **My Account**.

Create an administrative user

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the [AWS Management Console](#) as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

For help signing in by using root user, see [Signing in as the root user](#) in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

For instructions, see [Enable a virtual MFA device for your AWS account root user \(console\)](#) in the *IAM User Guide*.

Create an administrative user

1. Enable IAM Identity Center.

For instructions, see [Enabling AWS IAM Identity Center](#) in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to an administrative user.

For a tutorial about using the IAM Identity Center directory as your identity source, see [Configure user access with the default IAM Identity Center directory](#) in the *AWS IAM Identity Center User Guide*.

Sign in as the administrative user

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

For help signing in using an IAM Identity Center user, see [Signing in to the AWS access portal](#) in the *AWS Sign-In User Guide*.

Related resources

- [Security Best Practices in IAM](#)
- [AWS service quotas - AWS General Reference](#)

Set up AWS account security

This guide shows how to set up your AWS account to receive notifications when your resources are compromised, and to allow specific AWS account users to access it. To secure your AWS account and track your resources, complete the following steps.

Contents

- [Delete your account's access keys](#)
- [Enable multi-factor authentication](#)
- [Enable CloudTrail in all AWS Regions](#)
- [Set up Amazon GuardDuty and notifications](#)

Delete your account's access keys

You can allow programmatic access to your AWS resources from the AWS Command Line Interface (AWS CLI) or with AWS APIs. However, AWS recommends that you don't create or use the access keys associated with your root account for programmatic access.

If you still have access keys, we recommend that you delete those and create a user. Then, grant that user only the permissions needed for the APIs that you're planning to call. You can use that user to issue access keys.

For more information, see [Managing Access Keys for Your AWS account](#) in the *AWS General Reference guide*.

Enable multi-factor authentication

[Multi-factor authentication](#) (MFA) is a security capability that provides a layer of authentication in addition to your user name and password.

MFA works like this: After you sign in with your user name and password, you must also provide an additional piece of information that only you have physical access to. This information can come from a dedicated MFA hardware device, or from an app on a phone.

You must select the type of MFA device that you want to use from the [list of supported MFA devices](#). For a hardware device, keep the MFA device in a secure location.

If you use a virtual MFA device (like a phone app), think about what might happen if your phone is lost or damaged. One approach is to keep the virtual MFA device that you use in a safe place. Another option is to activate more than one device at the same time, or use a virtual MFA option for device key recovery.

To learn more about MFA, see [Enabling a Virtual Multi-Factor Authentication \(MFA\) Device](#).

Related resources

- [Getting Started with Multi-Factor Authentication](#)
- [Securing Access to AWS Using MFA](#)

Enable CloudTrail in all AWS Regions

You can track all activity in your AWS resources by using [AWS CloudTrail](#). We recommend that you turn on CloudTrail now. This can help AWS Support and your AWS solutions architect troubleshoot a security or configuration issue, later.

To enable CloudTrail logging in all AWS Regions, see [AWS CloudTrail Update – Turn On in All Regions and Use Multiple Trails](#).

To learn more about CloudTrail, see [Turn On CloudTrail: Log API Activity in Your AWS account](#).

To learn how CloudTrail monitors Nimble Studio, see [Logging Nimble Studio calls using AWS CloudTrail](#).

Set up Amazon GuardDuty and notifications

Amazon GuardDuty is a continuous security monitoring service that analyzes and processes the following:

- [Data sources](#)
- Amazon VPC Flow Logs
- AWS CloudTrail management event logs
- CloudTrail S3 data event logs
- DNS logs

Amazon GuardDuty identifies unexpected and potentially unauthorized and malicious activity within your AWS environment. Malicious activity can include issues like escalations of privileges, uses of exposed credentials, or communication with malicious IP addresses or domains. To identify these activities, GuardDuty uses threat intelligence feeds, such as lists of malicious IP addresses and domains, and machine learning. For example, GuardDuty can detect compromised Amazon EC2 instances serving malware or mining bitcoin.

GuardDuty also monitors AWS account access behavior for signs of compromise. This includes unauthorized infrastructure deployments, like instances deployed in an AWS Region that has never been used. It also includes unusual API calls, like a password policy change to reduce password strength.

GuardDuty informs you of the status of your AWS environment by producing [security findings](#). You can view these findings in the GuardDuty console or through [Amazon CloudWatch events](#).

Set up an Amazon SNS topic and endpoint

Follow the instructions in the [Setup an Amazon SNS topic and endpoint](#) tutorial.

Set up an EventBridge event for GuardDuty findings

Create a rule for EventBridge to send events for all findings that GuardDuty generates.

To create an EventBridge event for GuardDuty findings

1. Sign in to the Amazon EventBridge console: <https://console.aws.amazon.com/events/>
2. In the navigation pane, choose **Rules**. Then choose **Create rule**.
3. Enter a **Name** and **Description** for the new rule. Then choose **Next**.
4. Leave **AWS events or EventBridge partner events** selected for **Event source**.
5. In **Event pattern**, choose **AWS services** for the **Event source**. Then **GuardDuty** for the **AWS services**, and **GuardDuty Finding** for the **Event type**. This is the topic that you created in [Set up an Amazon SNS topic and endpoint](#).
6. Choose **Next**.
7. For **Target 1**, select **AWS service**. Choose **SNS topic** in the **Select a target** dropdown. Then choose your **GuardDuty_to_Email** topic.
8. In the **Additional settings** section: Use the **Configure target input** dropdown to choose **Input transformer**. Select **Configure input transformer**.
9. Enter the following code into the **Input path** field in the **Target input transformer** section.

```
{  
    "severity": "$.detail.severity",  
    "Account_ID": "$.detail.accountId",  
    "Finding_ID": "$.detail.id",  
    "Finding_Type": "$.detail.type",  
    "region": "$.region",  
    "Finding_description": "$.detail.description"  
}
```

10. To format the email, enter the following code into the **Template** field.

```
"AWS <Account_ID> has a severity <severity> GuardDuty finding type <Finding_Type>  
in the <region> region."  
"Finding Description:  
" <Finding_description>.  
"For more details open the GuardDuty console at https://console.aws.amazon.com/  
guardduty/home?region=<region>#/findings?search=id=<Finding_ID>"
```

11. Choose **Create**. Then choose **Next**.
12. (Optional) Add tags if you're using tags to track your AWS resources.
13. Choose **Next**.
14. Review your rule. Then choose **Create rule**.

Now that you've set up your AWS account security, you can grant access to specific users and receive notifications when your resources are compromised.

Check AWS service quotas

When you use an existing AWS account for deployment, it's important that you aren't already reaching your quota limit. This guide shows how to check AWS service quotas and request a quota increase for instances and streaming sessions.

Note

Most service quotas are specific to each AWS Region. Therefore, when you request a quota increase, select the AWS Region where the increase is required.

Contents

- [Check your Spot Instance quota](#)
- [Request a quota increase](#)
- [Find the Availability Zone ID of a launch profile](#)
- [Request a quota increase for Amazon Nimble Studio streaming sessions](#)
- [\(Optional\) Request a quota increase for G5 streaming sessions](#)
- [Request a quota increase for VPC security groups per network interface](#)
- [Request a quota increase for On-Demand Instances \(G and VT\)](#)

Check your Spot Instance quota

Increasing your Spot Instance quota lets you scale your farm to the size that your team needs to handle project workloads.

The following instructions show how to check your Spot Instance quota in the [AWS Service Quotas console](#).

To check your Spot Instance quota

1. Sign in to the AWS Management Console and open the [Service Quotas](#) console.

2. Check that your **Region** is set to where you want to deploy your studio.
3. In the left navigation pane, choose **AWS services**.
4. Search for Amazon Elastic Compute Cloud (Amazon EC2).
5. Choose **All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests** from the list.
6. Choose **Request quota increase**.
7. In the window that appears, enter a new quota value in the field for **Change quota value**.
 - a. To choose your new quota value: Multiply the number of render workers that you want to run concurrently with the number of vCPUs that you want each render worker to have.
 - b. For example, if you have 20 workers and want each worker to have 16 vCPUs, you would request to change your quota value to 320.
8. Choose **Request** to submit your request.

For information about the specs of different instances, see [Amazon EC2 Instance Types](#).

 **Note**

The default quota for Amazon Simple Storage Service (Amazon S3) buckets is 100 per AWS account. StudioBuilder will fail whenever you reach 4 buckets under your limit. To request a quota increase up to 1,000 buckets, open the [AWS Service Quotas console](#), choose AWS services from the left navigation pane, and search Amazon S3.

Request a quota increase

Your AWS account is subject to quotas. Quotas might impact how many Nimble Studio virtual workstations, render workers, and Amazon EC2 G4 instances that you can launch.

The following instructions show how to request a quota increase. After your request for a service quota increase is approved, your team gains access to the studio resources that they need. For more information, see [AWS service quotas](#).

 **Important**

Provide the following information when you request a quota increase for these instances.

- G3 and G5 instances: Instance type (count), OS, AWS Region

- Example: G5.2xlarge (2), Windows, us-east-1
- Example: G5.8xlarge (1), Linux, us-east-1
- G4 instances: Instance type (count), OS, AWS Region, [Availability Zone ID](#)
- Example: g4dn.2xlarge (3), Windows, usw2-az1
- Example: g4dn.2xlarge (2), Linux, usw2-az2

Find the Availability Zone ID of a launch profile

When you request a quota increase for a G4 instance, supply the Availability Zone ID of the launch profile that requires the quota increase.

To find the Availability Zone ID of a launch profile

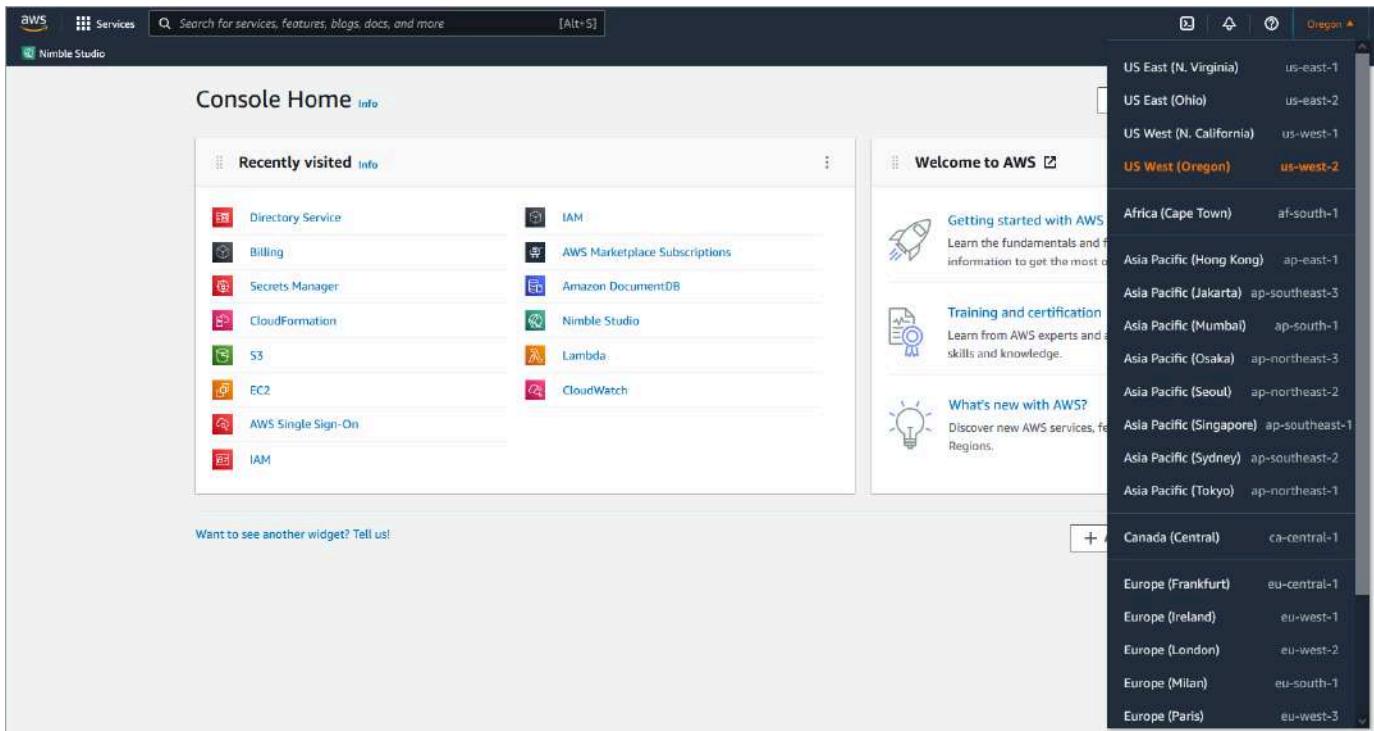
1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profile for the quota increase.
4. Select the **SubnetID** in the **Launch profile details** details section. The **Availability Zone ID** is in the **Details** section.

Request a quota increase for Amazon Nimble Studio streaming sessions

A streaming session is how artists connect to their virtual workstations so that they can access the files, settings, and applications that they need to work on an asset. By default, your AWS account is limited to two streaming sessions per studio. If you're planning to have more than two artists working at a time, request a quota increase for Nimble Studio streaming sessions.

To request a quota increase for Nimble Studio streaming sessions

1. Sign in to the AWS Management Console and open the [Service Quotas](#) console.
2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



3. In the left navigation pane, choose **AWS services**.
4. Search for **Amazon Nimble Studio**.
5. Choose **Amazon Nimble Studio** from the list.
6. Select the dot next to **Streaming sessions per studio**.
7. Choose **Request quota increase**.
8. Enter the desired number of concurrent streaming sessions into the **Change quota value** field.
 - Each artist requires their own unique streaming session. Therefore, choose the number of artists that will likely work in your studio at any given time.
9. Choose **Request** to submit your request.

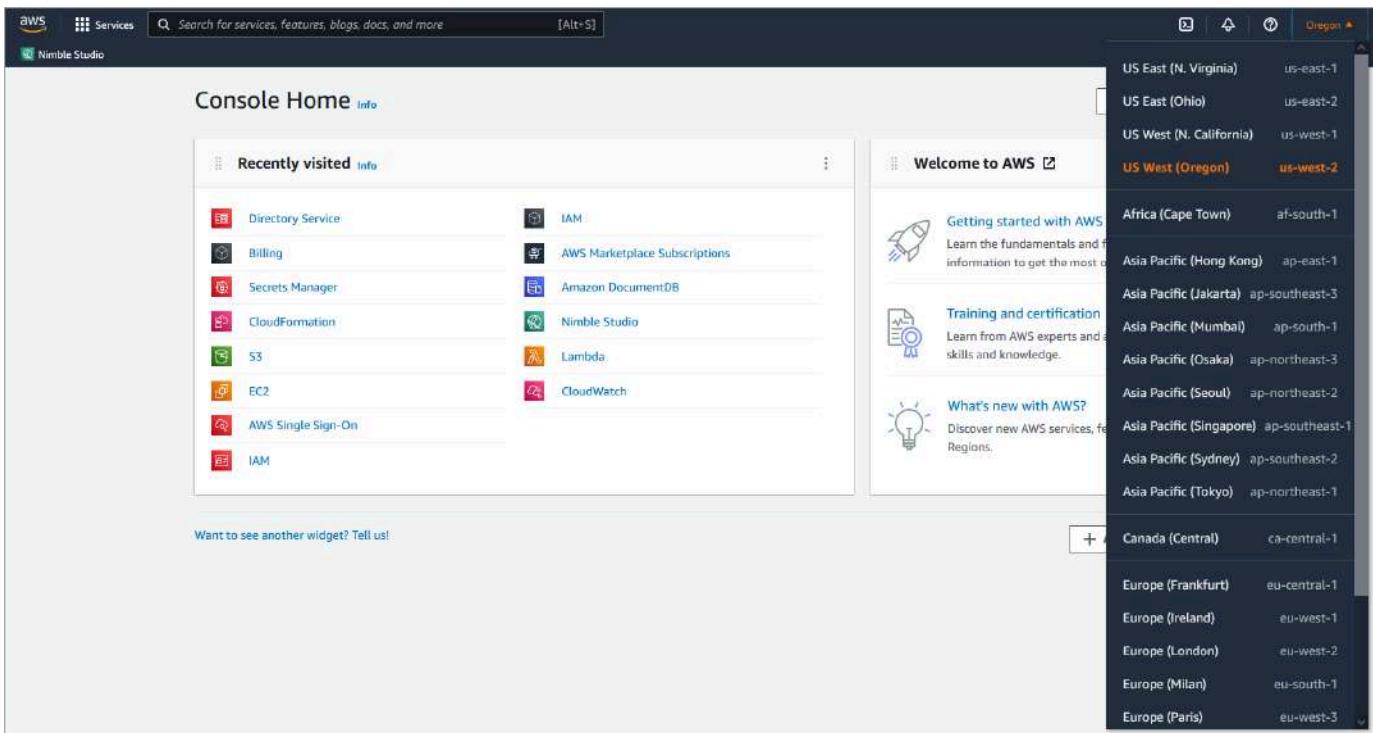
(Optional) Request a quota increase for G5 streaming sessions

By default, your studio has a quota of two streaming sessions. You can choose any supported instance types as part of this quota, except for G5 instances. For using G5 instances, request an increase in G5 streaming session quota. By default you have a quota of zero G5 streaming sessions.

To request a quota increase for G5 streaming sessions

1. Sign in to the AWS Management Console and open the [Service Quotas](#) console.

2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



3. In the left navigation pane, choose **AWS services**.
4. Search for **Amazon Nimble Studio**.
5. Choose **Amazon Nimble Studio** from the list.
6. Select the dot next to **G5 streaming sessions per studio**.
7. Choose **Request quota increase**.
8. Enter the desired number of G5 streaming sessions into the **Change quota value** field.
- Nimble Studio will increase the streaming sessions quota to match the number of G5 streaming sessions. Example: For a default quota of two and a request for five G5 instances, we raise the streaming session quota to five. This is to match the G5 quota.
9. Choose **Request** to submit your request.
10. You will be required to provide the following additional information: Instance type (count), OS, AWS Region.
- Example: G5.2xlarge (2), Windows, us-east-1
 - Example: G5.8xlarge (1), Linux, us-east-1

Request a quota increase for VPC security groups per network interface

1. In the left navigation pane, choose **AWS services**.
2. Search for **VPC**.
3. Choose **Amazon Virtual Private Cloud (Amazon VPC)** in the list.
4. Select the dot next to **Security groups per network interface**.
5. Choose **Request quota increase**.
6. In the **Change quota value** field, enter the desired number of security groups per network interface.
 - This number will depend on two things: How many studio components that you anticipate adding to your launch profiles and how many security groups are being used by those components. If you aren't sure, 10 is a good starting point. You can always increase this quota later.
7. Choose **Request** to submit your request.

Request a quota increase for On-Demand Instances (G and VT)

When you update the AMIs that your team uses to launch virtual workstations, you must launch GPU instances. This is so that you can install and test the software that you're adding to the AMI.

1. In the left navigation pane, choose **Dashboard**.
2. Choose **Amazon Elastic Compute Cloud (Amazon EC2)** from the list of dashboard cards.
3. Select **Running On-Demand G and VT instances** from the list and choose **Request quota increase..**
 - a. G stands for graphic-intensive instances.
 - b. VT stands for video transcoding instances.
4. In the window that appears, enter a new quota value in the **Change quota value** field.
 - a. Enter a minimum value of 4. We recommend entering at least **16** to be safe.
 - b. The minimum value of 4 allows you to run one g4dn.xlarge instance at a time for AMI creation because it has 4 vCPUs. When you run more than one instance at a time to create AMIs or a g4 instance with more vCPUs, request a higher quota value. A value of 16 allows

you to run 4 g4dn.xlarge instances at a time, or one g4dn.4xlarge instance that has 16 vCPUs.

5. Choose **Request** to submit your request.
6. You can check on the status of a quota request by choosing **Quota request history** in the navigation pane of the **Service Quotas** console. It can take 12–48 hours for a request to be resolved.

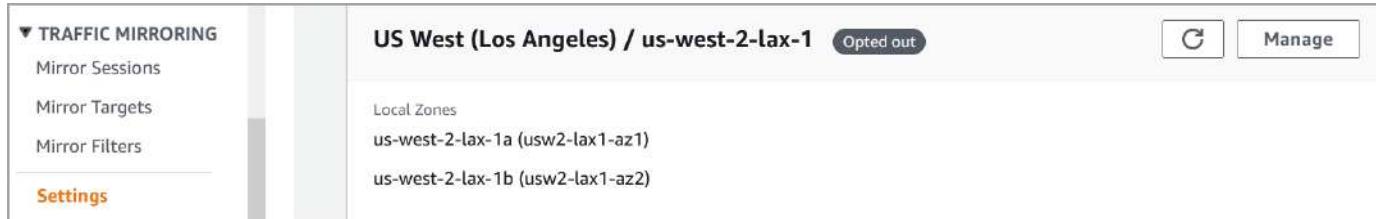
(Optional) Opt in to the LA Local Zone

To create a studio in the us-west-2 Region, you can opt in to the LA Local Zone. After you opt in, you can create workstations and file systems in the LA Local Zone. We recommend this for lower latency interactions. For example, if you and your artists are located closer to California than Oregon.

For more information about Local Zones, see the [AWS Local Zones FAQs](#). For more information about Availability Zones, see [Availability Zones for Amazon Nimble Studio](#).

To use the LA Local Zone, you must manually opt in from the Amazon VPC console. If you don't, the Local Zone won't appear in the Availability Zones list in the StudioBuilder deploy tool.

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. In the **Settings** section, select **Zones**.
3. Navigate down to **US West (Los Angeles)/us-west-2-lax-1** and choose **Manage**.



4. Select **Opted in**.
5. Select **Update zones**.

Now that you've opted in to the LA Local Zone, you can create resources there.

Getting started with Amazon Nimble Studio

When you have an AWS account set up as an administrator, you can deploy and configure Nimble Studio. This tutorial series shows admins how to get started with Amazon Nimble Studio. You'll learn how to use StudioBuilder to customize deployment, add studio users, and update your launch profiles. You'll also learn how to delete a studio.

Important

Complete all of the tutorials in the [Setting up to use Nimble Studio](#) section. These are prerequisites for deploying a Nimble Studio cloud studio.

Important

The studio that you create to use with [File Transfer](#) isn't compatible with Nimble Studio workstations. If you create a studio to use with File Transfer and later decide you want to set up Nimble Studio managed workstations, you need to delete your studio and create a new studio through the Nimble Studio setup process.

After you follow the steps in the [Setting up to use Nimble Studio](#) tutorial, you can begin this *Getting started* series. If you encounter any issues along the way, refer to the [Getting help and support](#) documentation.

Important

Even if you don't use your studio or any of the infrastructure that your studio creates, you will be charged for storage and other studio resources. If you aren't using your studio, we recommend deleting it so that you don't accrue unnecessary charges. For information about your AWS bill, see the [AWS Cost Explorer Service](#) and [AWS Budgets](#).

Topics

- [How StudioBuilder works with Amazon Nimble Studio](#)
- [Deploying a new studio with StudioBuilder](#)
- [How to delete a studio](#)

How StudioBuilder works with Amazon Nimble Studio

StudioBuilder is a command line interface (CLI) tool for IT admins to configure Nimble Studio and set up its infrastructure. Within the terminal, StudioBuilder converts the admin's input to a configuration file and deploys an [AWS Cloud Development Kit \(AWS CDK\)](#) application to create account resources using [AWS CloudFormation](#).

Contents

- [How does StudioBuilder work?](#)
- [What resources does StudioBuilder create?](#)
- [Troubleshooting](#)
- [Related resources](#)

How does StudioBuilder work?

StudioBuilder's deploy tool creates a configuration file to build your studio, based on answers that you provide within its CLI. After all of the questions are answered and validated in the terminal, you can deploy the StudioBuilder CDK application. StudioBuilder creates four CloudFormation stacks when it builds your studio, and these stacks create your studio resources.

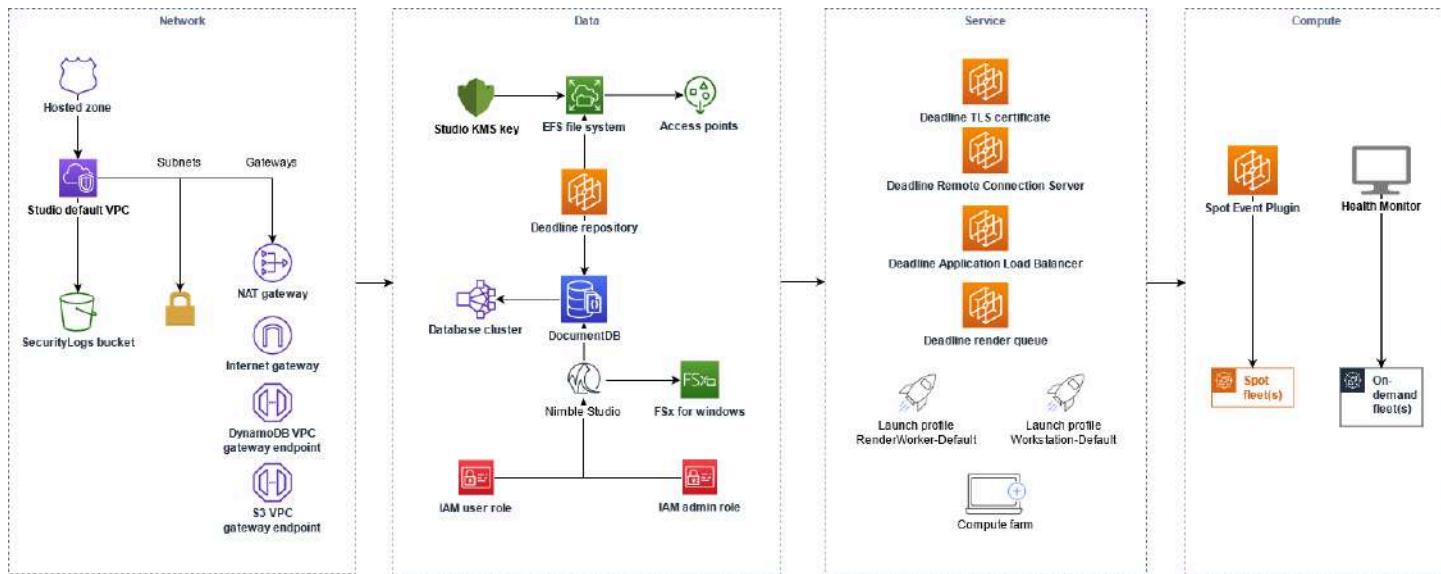
StudioBuilder's CDK application performs the following functions:

- Uses the configuration file to determine which resources get deployed by CloudFormation in your account.
- Creates an [Amazon Virtual Private Cloud \(Amazon VPC\)](#) and a range of secure IP addresses in your VPC (called [subnets](#)).
- Configures the [network access control lists \(NACLs\)](#) and security groups to help protect your resources from malicious users.
- Creates Nimble Studio components, including a render farm, storage, and [AWS Directory Service for Microsoft Active Directory \(AWS Managed Microsoft AD\)](#). Automatically configures these resources so they're ready for use.
- Launches and automatically scales at least one [fleet](#) of [Spot Instances](#), as needed, within a single [Amazon EC2 Auto Scaling group](#). This allows you to have at least one render worker always running.
- Creates two launch profiles—one for the IT administrator to configure the studio, and the other for render workers.

What resources does StudioBuilder create?

The StudioBuilder CDK application is broken into four tiers: the network tier, data tier, service tier, and compute tier. Each tier creates an individual CloudFormation stack which the next tier depends on. The compute tier stack depends on the service tier stack, the service tier stack depends on the data tier stack, and the data tier stack depends on the network tier stack.

Here is a high-level diagram of the CDK application showing these tier dependencies.



Network tier

The network tier is the base tier. Without the network tier, no other tier will work. The network architecture of the studio creates the following resources in your account.

Amazon VPC

- The default VPC that StudioBuilder creates for your studio is an Amazon VPC. It has the same name as your studio ID. This is the private network that launches AWS resources created by StudioBuilder for your Nimble Studio cloud studio.

Subnets

StudioBuilder creates 12 subnets in the studio default VPC. The subnets divide the Amazon VPC up into smaller blocks and the resources that require IP addresses are assigned to each subnet. This is done to isolate resources from each other for security, which is supplemented by NACLs and security groups.

Note

Some subnets need to exist in at least two availability zones due to the requirements of the services hosted within them.

- The public subnet named Public routes all traffic to the internet gateway.
- Two private subnets named Backend are created to put the [Amazon DocumentDB](#) file system and the [Amazon Elastic File System \(Amazon EFS\)](#) file system into after they're created in the data tier.
- Two private subnets named ActiveDirectory create the AWS Managed Microsoft AD in the data tier.
- The private subnet named FileSystems creates the [Amazon FSx](#) file system in the data tier.
- Two private subnets named ServiceEndpoints are created for the Application Load Balancer (ALB) in the service tier.
- The private subnet named Workstations is created to auto configure the AWS Managed Microsoft AD in the service tier.
- Two private subnets named WorkerSupport create the health monitor in the compute tier.
- The private subnet named RenderWorkers creates the Deadline render fleet in the compute tier.

Gateways and endpoints

- The network address translation (NAT) gateway inside of the Public subnet routes traffic between private subnets and internet gateway.
- The internet gateway in the Public subnet routes all traffic to the internet.
- The NACLs help limit traffic between each subnet for added security.
- Two endpoints are created to control access to your resources.
 - The [Amazon DynamoDB](#) VPC gateway endpoint is associated with the Backend subnets and gives DynamoDB access to the [Deadline Remote Connection Server \(RCS\)](#) when the [Deadline Resource Tracker](#) is being used. The RCS is a server application which controls access to the Deadline Database and Deadline Repository from the Deadline Client.
 - The [Amazon S3 VPC gateway endpoint](#) is associated with Backend, RenderWorker, WorkerSupport, and Workstation subnets and gives them access to all S3 buckets. Traffic

goes through S3 buckets from the gateway instead of through the public internet. You don't need to configure permissions or filtering rules.

Route 53

[Amazon Route 53](#) is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking.

The network tier creates the nimble.<aws_region>.aws domain. This domain connects with the RCS, created in the service tier, to run the render queue. The following three records are created with the RCS.

- The record nimble.<aws_region>.aws with type name server (NS) indicates which Domain Name System (DNS) server is authoritative for the domain.
- The record nimble.<aws_region>.aws with type start of authority (SOA) contains administrative information.
- The record renderqueue.nimble.<aws_region>.aws with type A (IPv4 address record of the compute farm) connects with the render farm in Deadline.

Security groups

- The VPC Interface endpoints security group limits access to VPC interface endpoints so that studio resources can securely communicate with AWS services.
- The License servers security group controls access to customer-created license servers.

Amazon S3

[Amazon Simple Storage Service \(Amazon S3\)](#) is an object storage service that offers scalability, data availability, security, and performance.

- **SecurityLogs:** The bucket named <studio_id> + network-SecurityLogs + ... stores flow logs for the studio Amazon VPC to create the render queue in the service tier.
- **SecurityLogsAccess:** The bucket named <studio_id> + network-SecurityLogsAccess + ... is the access logging bucket for the studio Amamzon VPC.
- **StudioConfig:** The studioconfig-<studio_id>-<account_id>-<region> bucket stores the backup config file.

IAM

[AWS Identity and Access Management \(IAM\)](#) is a web service that helps you securely control access to AWS resources. Admins use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

- The role named <studio_id> + BucketRole grants permissions to back up the StudioBuilder config file to the studioconfig-<studio_id>-<account_id>-<region> S3 bucket.

Data tier

The data tier stack only depends on the network stack. This is the stack where your studio is created. The data architecture of the studio creates the following resources in your account.

Security groups

- The Deadline EFS File System security group controls access to the Amazon EFS file system used by Deadline.
- The DocumentDB Security Group security group controls access to the Amazon DocumentDB cluster used by Deadline.
- The Deadline Repository Installer security group controls access to the Deadline Repository installer.
- The FSx File Systems security group controls access to the Amazon FSx file systems.
- The Active Directory auto-configuration security group allows the AWS Managed Microsoft AD configuration instance to work with the least amount of permissions.
- The Workstation access to file systems security group controls outbound access to the Nimble Studio workstation.
- The Workstation access to Deadline security group controls access to Deadline through the Nimble Studio workstation .

AWS KMS

[AWS Key Management Service \(AWS KMS\)](#) is a managed service that makes it easy for you to create and control the cryptographic keys that are used to protect your data.

- The <studio_id>-Key encrypts your studio and encrypts your SNS messages.

Amazon EFS

[Amazon Elastic File System \(Amazon EFS\)](#) provides a simple, serverless, set-and-forget elastic file system for use with AWS Cloud services and on-premises resources.

- The file system named RepositoryFS is where the Deadline repository is installed. You can install custom Deadline scripts in this file system.
- The access point named RepositoryFS is the access point for Deadline repository file system, which stores plugins, scripts, logs, and any auxiliary files.
- The access point named PaddingAccessPoint is the access point for the Lambda function PadEfsStoragePadFilesystem. The PadEfsStoragePadFilesystem Lambda function uses the PaddingAccessPoint access point to create random files in Amazon EFS to maintain burst credit for Deadline. For more information about burst credits, see [Demonstrate how to send an email alarm when EFS burst credits below a threshold](#) on GitHub.

Amazon DocumentDB

[Amazon DocumentDB \(with MongoDB compatibility\)](#) is a fast, reliable, and fully managed database service.

- The cluster with the prefix databasecluster is used to work with the Deadline Database to store the jobs, settings, and worker configurations of the Deadline render farm management.
- The cluster parameter group dbauditlogging + ... enables Amazon DocumentDB audit logging.

Amazon FSx

[Amazon FSx for Windows File Server](#) provides fully managed Microsoft Windows file servers, backed by a fully native Windows file system.

- The Amazon FSx file system named FSxWindows is created using the FileSystems subnet created in the network tier based on your configurations. By default, the file system has a 200GB SSD storage capacity and a 16MB/s throughput capacity.

Auto Scaling groups

An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management.

- The RepositoryInstaller Auto Scaling group is created for the Deadline Repository installer to help install the Deadline Repository.

Secrets Manager

[AWS Secrets Manager](#) helps you protect secrets needed to access your applications, services, and IT resources. The service allows you to rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle.

- The root certificate authority certificate <studio_id>RootCA will be used to sign the Transport Layer Security (TLS) certificate in the service stack. Both the public and the private parts of the certificate are stored in Secrets Manager.
- The RenderWorkerADCredentials secret stores the AWS Managed Microsoft AD's credentials for render workers.
- The StudioBuilderAdminADCredentials secret stores the AWS Managed Microsoft AD's credentials for the studio admin.

AWS Lambda

[AWS Lambda](#) is a serverless compute service that lets you run code without provisioning or managing servers, creating workload-aware cluster scaling logic, maintaining event integrations, or managing runtimes.

- The Lambda function RotateADCredentials rotates the AWS Managed Microsoft AD's passwords and updates the secrets, RenderWorkerADCredentials and StudioBuilderAdminADCredential with the new passwords.

CloudWatch Alarms

[CloudWatch Alarms](#) watch a single metric over a specified time period.

- Perform one or more specified actions, based on the value of the metric, relative to a threshold over time.
- Deploy four CloudWatch Alarms, all named Burst Credits. They fire at different thresholds if the file system becomes too busy.

AWS Step Functions

[AWS Step Functions](#) is a serverless orchestration service that lets you combine Lambda functions and other AWS services to build business-critical applications.

Step Functions is based on state machines and tasks. A state machine is a workflow. A task is a state in a workflow that represents a single unit of work that another AWS service performs. Each step in a workflow is a state.

- A state machine named PadEfsStorageStateMachine is created so that when the Burst Credits CloudWatch Alarms fire, you can manually expand the data stored on the Deadline Repository EFS share.
 - The PadEfsStorageStateMachine is part of the Render Farm Deployment Kit (RDFK).
 - For more information about the RDFK, see the [Render Farm Deployment Kit on AWS Developer Guide](#).

IAM

- The IAM role studioadminrole grants permissions for studio admin. The studio administrator can add, modify, or delete launch profiles and studio components. The studio administrator can also accept any end-user license agreements (EULAs).
- The IAM role studiouserrole grants permissions for studio user. Non-administrator studio users can sign into the Nimble Studio portal and use their launch profile(s) to start a streaming workstation.

Nimble Studio resources

- The AWS Managed Microsoft AD named ActiveDirectory is used by studio administrators to manage users, teams, and groups. The AWS Managed Microsoft AD password is defined by you during the studio configuration and is stored in Secrets Manager. The AWS Managed Microsoft AD is used to authenticate users when they sign in to streaming workstations, controlling file access on Amazon FSx for Windows volumes, and is used by resources like [AWS IAM Identity Center \(IAM Identity Center\)](#) to sign into the Nimble Studio portal. You can find this AWS Managed Microsoft AD in the [Directory Service console](#).
- The custom configuration InstanceConfiguration is a custom Amazon Elastic Compute Cloud (Amazon EC2) instance that controls file permissions for new files.
- The NimbleStudioRegistration key, which is what creates your Nimble Studio cloud studio.

⚠️ Important

Don't delete your data stack; this deletes your NimbleStudioRegistration key which will automatically delete your Nimble Studio cloud studio.

Service tier

The service tier stack depends on the data stack and the network stack. AWS Managed Microsoft AD auto configuration happens in the service tier. The service architecture of the studio creates the following resources in your account.

Security groups

- The Render Worker Access security group controls Nimble Studio worker fleet access.

AWS Lambda

- The Lambda function GetStreamingImageIdsFunction is created to obtain a streaming image ID.
- The Lambda function PadEfsStoragePadFilesystem is created to maintain the burst credit for Deadline.

Deadline

[Deadline](#) is an administration and compute management toolkit for Windows, Linux, and macOS based render farms.

The service tier creates six Deadline resources to help you manage your render farms.

- Deadline's TLS certificate, which gets signed with the root certificate authority certificate that was created in the data tier.
- Deadline RCS, which controls access to the Deadline Database and Repository from the Deadline client.
- Deadline ALB load balances HTTP/HTTPS request within AWS. The ALB is only created if you have more than one RCS instance in your Nimble Studio cloud studio.

- Deadline's Spot Event Plugin policies for the compute tier is used to create a Spot Event Plugin configuration in the compute tier.

Nimble Studio resources

- The studio component ComputeFarm runs the render farm in Deadline.
- The compute farm RenderFarm is the render farm that runs in Deadline. The EC2 instance associated with this compute farm is named <studio_id>Service/RenderQueue/Cluster/RCS Capacity.
- The RenderQueueClusterRCSCapacity Auto-Scaling Group is created for the <studio_id>Service/RenderQueue/Cluster/RCS Capacity instance. It's used by Deadline and is usually a single EC2 instance, but can be scaled up if there is a large number of jobs or projects in the queue.
- The ADAutoConfigurationInstance EC2 instance is created for <studio_id>Service/ADAutoConfiguration/Instance. It automatically configures the AWS Managed Microsoft AD before shutting itself down. It creates some initial AWS Managed Microsoft AD users and configures the group policies so that roaming profiles, which workstations rely on, are turned on by default
- The launch profile **RenderWorker-Default** is associated with the **Workstations** subnet created in the network tier, and the [Amazon Machine Images \(AMIs\)](#) that you configured when running StudioBuilder. If you add a new storage component to your studio and want to render a file using that component, it needs to be added to **RenderWorker-Default**. The **RenderWorker-Default** launch profile only exists if you deploy a render farm.
- The launch profile **Workstation-Default** is associated with the Workstations subnet created in the network tier, and the AMIs that you configured when running StudioBuilder. The **Workstation-Default** launch profile is always created, regardless of whether you deploy a compute farm.

Compute tier

The compute tier stack depends on the service stack, the data stack, and the network stack. The compute architecture of the studio creates the following resources in your account.

Security groups

- The security group **ConfigureSpotEventPluginConfiguratorSecurityGroup** is created to limit access to the SEP.

Spot Event Plugin

- The [Deadline Spot Event Plugin \(SEP\)](#) is a Deadline plugin that makes Spot Fleet requests spin up worker EC2 instances when farm jobs are in the queue. One SEP configuration is created for each Spot render fleet. The SEP configuration contains the EC2 instance types and launch templates it needs to use when spinning up workers for the group.
- Deadline also deploys a [Deadline Resource Tracker](#) when it deploys a Spot Fleet as part of the Spot Event Plugin's operation.

Nimble Studio resources

- Render fleets are created for the render farm. Render fleets are collections of EC2 instances. The render fleets also include On-Demand fleets. On-Demand fleet sizes are managed by an Auto Scaling group and Spot Fleets. Multiple fleets can be defined for a farm. A Deadline group of the same name is automatically created for each fleet. Fleets can be either Linux or Windows, based on your configuration.
- The render fleet health monitor is created to monitor the health of On-Demand fleets. If an EC2 instance is deemed unhealthy, it's terminated and replaced. If enough EC2 instances are unhealthy, the fleet is scaled down to zero.

Troubleshooting

My Spot Fleets are being terminated.

The Deadline Resource Tracker is likely the cause. To resolve this issue, follow the instructions in the Instances Shutting Down section of the [Troubleshooting](#) guide in the Deadline Resource Tracker documentation.

I got a CloudWatch Alarm about burst credits

If you're getting a CloudWatch Alarm email about burst credits, your file system is too busy. To fix the issue, increase the burst credits availability that you have each month.

To increase the available burst credits

1. Sign in to the AWS Management Console and open the [Step Functions](#) console.
2. Select the step function with the name beginning with **PadEfsStorageStateMachine**.
3. Choose **Start execution**.
4. (Optional) Give the execution a name.
5. Enter the following code in the **Input** section. For desired padding, enter the GiB of data that you want to add to the EFS.

```
{  
  "desiredPadding": 100,  
}
```

6. Choose **Start execution**.

Related resources

- [Step 4: Configure studio with StudioBuilder](#)
- [Step 5: Deploy studio with StudioBuilder](#)

Deploying a new studio with StudioBuilder

This tutorial is for administrator users. It guides you through the process of deploying Amazon Nimble Studio in your AWS account using the StudioBuilder app. In this step-by-step tutorial, you'll learn how to configure your studio, including choosing your studio name, setting up AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD), creating shared storage, and selecting options for your render farm.

Important

Even if you don't use your studio or any of the infrastructure that your studio creates, you will be charged for storage and other studio resources. If you aren't using your studio, we recommend deleting it so that you don't accrue unnecessary charges. For information about your AWS bill, see the [AWS Cost Explorer Service](#) and [AWS Budgets](#).

⚠️ Important

The studio that you create to use with [File Transfer](#) isn't compatible with Nimble Studio workstations. If you create a studio to use with File Transfer and later decide you want to set up Nimble Studio managed workstations, you need to delete your studio and create a new studio through the Nimble Studio setup process.

Contents

- [Prerequisites](#)
- [Step 1: Enable IAM Identity Center](#)
- [Step 2: Access the StudioBuilder AMIs](#)
- [Step 3: Launch the StudioBuilder EC2 instance](#)
- [Step 4: Configure studio with StudioBuilder](#)
- [Step 5: Deploy studio with StudioBuilder](#)
- [Step 6: Link AWS Managed Microsoft AD as an IAM Identity Center identity source](#)
- [Step 7: Confirm subscription to burst alert emails](#)
- [Troubleshooting](#)
- [Related resources](#)

Estimated time: 2 hours

Prerequisites

⚠️ Important

Complete all of the tutorials in the [Setting up to use Nimble Studio](#) section. These are prerequisites for deploying a Nimble Studio cloud studio.

- We recommend that you check your quotas as described in the [Setting up to use Nimble Studio](#) tutorial before you begin deployment. Quota increase requests can take up to 48 hours to be fulfilled.

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open <https://portal.aws.amazon.com/billing/signup>.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, [assign administrative access to an administrative user](#), and use only the root user to perform [tasks that require root user access](#).

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to <https://aws.amazon.com/> and choosing **My Account**.

Create an administrative user

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the [AWS Management Console](#) as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

For help signing in by using root user, see [Signing in as the root user](#) in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

For instructions, see [Enable a virtual MFA device for your AWS account root user \(console\)](#) in the *IAM User Guide*.

Create an administrative user

1. Enable IAM Identity Center.

For instructions, see [Enabling AWS IAM Identity Center](#) in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to an administrative user.

For a tutorial about using the IAM Identity Center directory as your identity source, see [Configure user access with the default IAM Identity Center directory](#) in the *AWS IAM Identity Center User Guide*.

Sign in as the administrative user

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

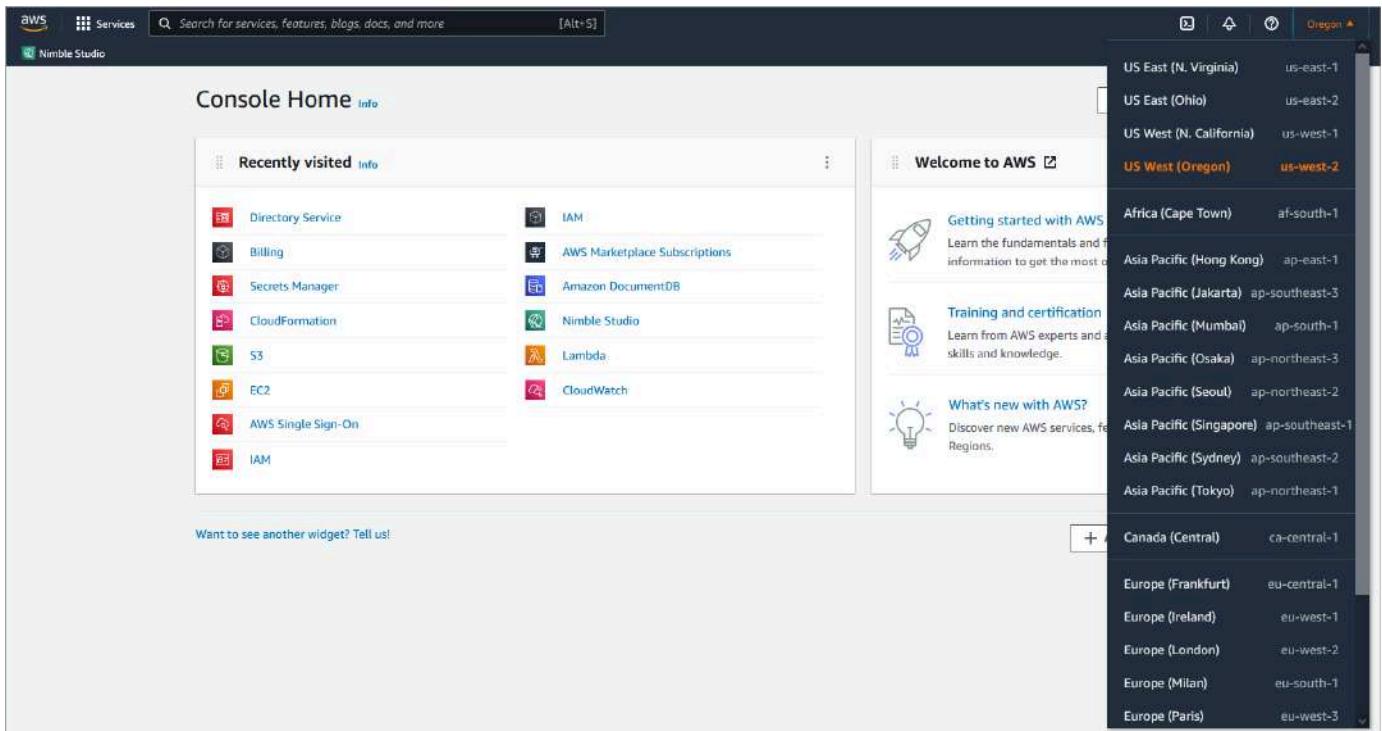
For help signing in using an IAM Identity Center user, see [Signing in to the AWS access portal](#) in the *AWS Sign-In User Guide*.

Step 1: Enable IAM Identity Center

This first step will enable AWS IAM Identity Center (IAM Identity Center) for your account. With IAM Identity Center, you can manage user permissions and access to your accounts and applications in one place. To enable IAM Identity Center, go to the Nimble Studio service page and follow the first step to set up your studio. After you enable IAM Identity Center, you're ready to deploy your studio with StudioBuilder.

To deploy your studio and enable IAM Identity Center

- Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
- In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



3. Choose Set up Nimble Studio.

Amazon Nimble Studio

Your creative studio in the cloud

Amazon Nimble Studio is a suite of services that empower creative studios to produce visual effects, animation, and games content entirely in the cloud. Onboard creative talent in a matter of minutes, and with availability in major creation markets, you can look for and hire the best talent for the project.

How Nimble Studio works

In just a few hours, you can create a new studio environment in which creative talent can immediately access virtual workstations that are powered by Amazon Elastic Compute Cloud (EC2) instances, and high-speed storage from Amazon FSx. Artists can work with their preferred third-party creative applications and custom software applications by using Amazon Machine Images (AMIs) with support for both Windows and Linux operating systems.

Get started with Nimble Studio

Set up your Nimble Studio services using the AWS Management console.

Nimble Studio

Nimble Studio provides the infrastructure that you need to operate a fully cloud-based creative studio using streaming workstations to produce visual effects (VFX), animation, and game content.

[Get started](#)

Related applications

You can download and install applications onto your local machine, EC2 instance, or Nimble Studio workstation.

AWS Thinkbox software

AWS Thinkbox software, which includes AWS Thinkbox Deadline and a series of 3D plugins like AWS Thinkbox Krakatoa, helps creative studios scale your creativity on premises, hybrid, or on the cloud. All AWS Thinkbox products are available free of charge.

[Download](#)

4. Choose **Enable IAM Identity Center**.

- If you see a green confirmation that IAM Identity Center has already been enabled, skip to [Step 2: Access the StudioBuilder AMIs](#).

5. In the **Enable IAM Identity Center** pop-up, choose **Create AWS organization**.

- A success message confirms that IAM Identity Center has been enabled.

Step 2: Access the StudioBuilder AMIs

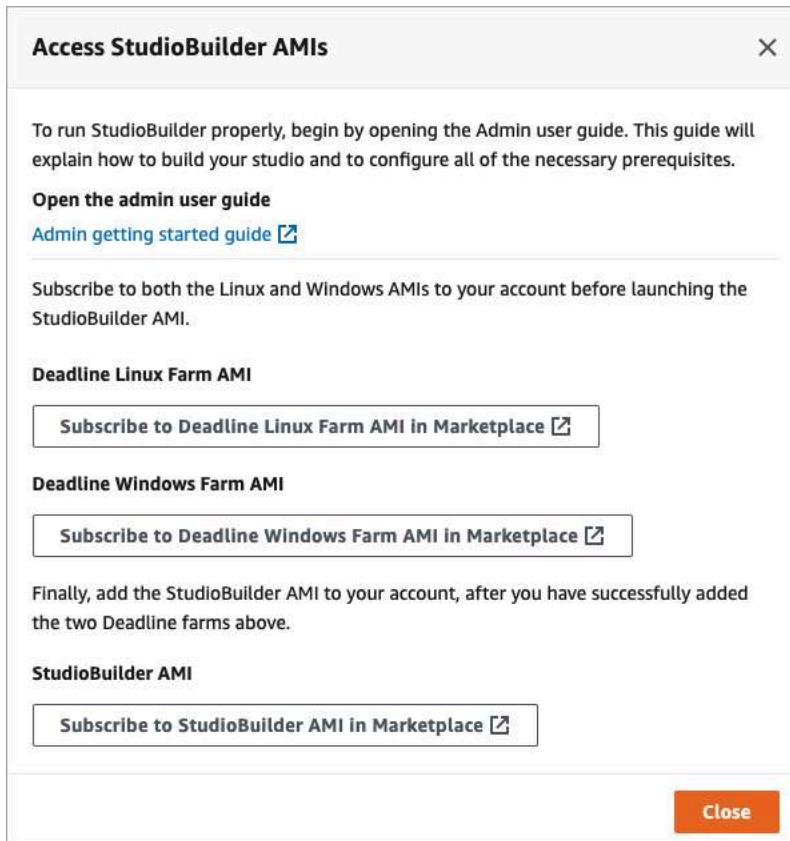
To run StudioBuilder and deploy your studio, launch an Amazon Elastic Compute Cloud (Amazon EC2) instance using the StudioBuilder Amazon Machine Image (AMI). This has all of the packages that you need for to deploy your studio.

The following steps show how to add the Linux, Windows, and StudioBuilder AMIs to your account. These steps are necessary before you can launch an instance with the StudioBuilder AMI.

To navigate to the StudioBuilder AMIs in the AWS Marketplace

1. Choose Access StudioBuilder AMIs.

- The **Access StudioBuilder AMIs** window will open.



2. Choose **Subscribe to Deadline Linux Farm AMI in Marketplace**.

- The Nimble Studio Deadline Linux farm worker page will open in the AWS Marketplace.

3. Choose **Continue to Subscribe**.

4. Read the terms and conditions. Then choose **Accept Terms**.

5. Return to the **Access StudioBuilder AMIs** window and choose **Subscribe to Deadline Windows Farm AMI in Marketplace**.
6. Repeat the previous steps to accept the terms and conditions.
7. Return to the **Access StudioBuilder AMIs** window again and choose **Subscribe to StudioBuilder AMI in Marketplace**.
8. Repeat the same steps to accept the terms and conditions.
9. After the subscribe request has finished processing, choose **Launch new instance**. Stay on the **Subscribe to StudioBuilder AMI in Marketplace** page until the subscription process is completed.

Nimble Studio StudioBuilder Amazon Machine Image

Launch new instance

Nimble Studio's StudioBuilder helps you create a virtual studio from scratch. Walk through and set up your studio by creating networking, render farm, and storage resources. The StudioBuilder process creates and deploys new resources into your region of choice, helping you build a studio in just a few hours. [Read more on AWS Marketplace](#)

10. In the **Region** dropdown menu, select the AWS Region that you want to deploy your studio to and choose **Continue to launch through EC2**.

The Amazon EC2 console will open to guide you through the rest of the launch process.

Since you have already chosen an AMI, the Amazon EC2 console will automatically fill in the **Application and OS Images (Amazon Machine Image)** section.

Step 3: Launch the StudioBuilder EC2 instance

Follow the instructions in the [Launch an instance using defined parameters](#) tutorial in the *Amazon EC2 User Guide for Windows Instances* while using the following information.

1. For **Name and tags**, give the instance a name so that you can easily find it later, such as NimbleStudioBuilder.
2. For **Instance Type**, select **t3.medium** from the list.
3. For **Key pair (login)** choose **Proceed without a key pair** from the first dropdown.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Proceed without a key pair (Not recommended) Default value ▾ [Create new key pair](#)

- A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance. You will use EC2 Instance Connect so you don't need a key pair.

4. On **Network settings**, choose **Edit**.
5. Set **Auto-assign Public IP** to **Enable** so that your instance receives a public IP address that you will use when connecting to it later.
6. In **Advanced details**, specify an AWS Identity and Access Management (IAM) role.
 - a. The IAM role enables administrator access from your StudioBuilder instance.
 - b. If you have connected to StudioBuilder before, choose the **StudioBuilder_Instance_Admin_Role** role that you created previously and continue to [Step 4: Configure studio with StudioBuilder](#).
 - c. If you have not connected to StudioBuilder before, choose **Create new IAM profile** to open a new tab with the **Create role** page in **IAM**.

▼ 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](#)

StudioBuilder_Instance_Admin_Role
arn:aws:iam:: [Create new IAM profile](#)

- i. Choose **Create role**.
- ii. Select **AWS service**.
- iii. Choose **EC2**.

- iv. Choose **Next: Permissions**.
 - v. Select **AdministratorAccess**.
 - vi. Choose **Next: Tags**.
 - vii. Choose **Next: Review**.
 - viii. Enter the **Role name**. Example: **StudioBuilder_Instance_Admin_Role**.
 - ix. Enter a **Role description**. Example: **Gives administrative access to the StudioBuilder instance**.
 - x. Choose **Create role**.
 - xi. Close the tab and switch back to the Launch instance wizard tab.
- d. To update the list of available IAM roles, choose the Refresh button next to **Create new IAM profile**.
- e. Select the new role you created from the list.
7. Choose **Launch**.
8. On the **Launch Status** page, choose **View Instances**.
9. After the **instance state** of your instance changes from **Initializing** to **Running**, select it and choose **Connect**.

Step 4: Configure studio with StudioBuilder

Next, you'll connect to your instance to run StudioBuilder. StudioBuilder helps you deploy a studio by asking a series of questions about how you want to configure your studio. Throughout the process, StudioBuilder builds your studio based on your answers.

Connect to the instance with EC2 Instance Connect

1. On the **Connect to instance** page, select **EC2 Instance Connect**.
2. Change the user name to **ec2-user**. If you leave the user name as **root**, StudioBuilder might not run correctly.
3. Choose **Connect**.
 - a. A connection to your instance is established.
 - b. A new window opens to the StudioBuilder Command Line Interface (CLI) so you can configure your studio.

- If at any time during the configuration process you need to reconnect to your instance, follow the previous steps to reconnect with EC2 Instance Connect to continue where you left off.

Configure your studio

Note

During deployment or deletion, Amazon Nimble Studio might collect generic user metrics, called telemetry, to inform us about StudioBuilder errors and configuration preferences. These metrics are for diagnosis purposes only, and aren't shared or transferred.

If StudioBuilder doesn't automatically run, see the [Troubleshooting](#) section of this tutorial for instructions about how to run StudioBuilder manually.

StudioBuilder will run and ask you the following questions about how to configure your studio.

1. Welcome to StudioBuilder! What would you like to do?

a. Using the arrow keys, choose Create a studio with StudioBuilder and press the enter (or return) key.

```
StudioBuilder [version 1.0.8]
StudioBuilder will build and deploy your Amazon Nimble Studio.

Use arrow keys to select your choice, then press Enter. To select "Other" from a list, press Enter, then type in your response. Press Enter again to submit.

Unable to find existing studio(s).

? Welcome to StudioBuilder! What would you like to do? (Use arrow keys)
❯ Create a studio with StudioBuilder.
  Exit to command line.
```

2. Please confirm that you wish to deploy your studio to Region <current-region>.
 - a. In the line that precedes the question, check that the Region you want is listed, then choose Deploy my studio in this Region and press the enter (or return) key.

Set up studio

1. Enter a name for your studio.
 - a. Enter a name for your studio and press the enter (or return) key.
 - b. Your studio name must be between 3-64 characters and only include lowercase letters from a-z and numbers from 0-9.
2. Enter your studio name as you would like it displayed.
 - a. Enter the display name for your studio and press the enter (or return) key. You can enter a more descriptive name here, using whatever characters you like.

Configure VPC settings

1. Specify a CIDR block for the virtual private cloud (VPC) resource, or leave at 10.0.0.0. Then, press your enter (or return) key.
 - a. You can specify a particular CIDR block, or just leave this at the default of 10.0.0.0 and press the enter (or return) key.
2. Select an Availability Zone for your studio, then press Enter.
 - a. We recommend leaving this at the default of Please choose for me (Recommended), then pressing the enter (or return) key.
3. Select a Local Zone for your studio, then press Enter.
 - a. You can only choose a Local Zone if your studio is hosted in the us-west-2 Region and if you have opted in to using Local Zones from the console. For instructions about how to opt in, see the [Availability Zones for Amazon Nimble Studio](#) tutorial.
4. What domain name would you like to associate your studio and studio resources with?
 - a. By default, the domain name contains your studio ID.
5. Would you like to deploy VPC endpoints to privately connect to supported AWS services?

- a. Your studio uses VPC endpoints to privately connect to supported AWS services without using public IP addresses. This is safer than using a public internet connection because traffic between your VPC and other services doesn't leave the Amazon network.
- b. The default selection here is Yes, I'd like to use VPC Interface Endpoints.

Choose your AWS Directory Service name

The command line interface (CLI) will ask you to:

1. Create your Admin password for AD using 3-64 lower and uppercase letters, numbers, and symbols.
 - a. Specify a password and press the enter (or return) key.
 - b. Confirm the password and press the enter (or return) key.

 **Note**

If you ever forget this password, you can retrieve it from the **StudioBuilder-SecretForAD** secret in [AWS Secrets Manager](#).

AWS Directory Service - POSIX file system support

1. Would you like to specify a POSIX GID and UID?
 - a. Nimble Studio will configure AWS Managed Microsoft AD to operate on POSIX file systems, such as Linux. That allows your users to work on both Windows and Linux workstations.
2. If you plan to bring an existing file system, we recommend that you specify a uid and gid to prevent collisions with permissions on pre-existing files or folders.
3. However, if you don't plan to bring an existing file system, we recommend choosing Please choose for me (Recommended) and press the enter (or return) key.

Deletion protection

1. Deletion protection prevents you from accidentally deleting your studio and farm, which could cause you to lose data. Within the CLI interface, do one of the following:
 - a. (Recommended) Choose **Enable deletion protection**.
 - b. Press the enter (or return) key to continue without enabling deletion protection.

Render fleets

Follow these steps when the CLI prompt says: If you don't have a render farm in your studio, you can create one now. Render farm costs can vary. For pricing estimates, see <https://aws.amazon.com/nimble-studio/pricing>.

1. Use the arrow keys to choose if you would like to create a render farm or not, and then press the enter (or return) key.
 - a. If you choose Yes, I want to create a render farm for my studio:
 - i. Render fleet questions that are shown in the following section of this tutorial will be displayed.
 - ii. Follow the prompts in the CLI for configuring your render farm.
2. If you choose No, I don't want to create a render farm at this time
 - a. Skip all render farm questions and prompts. Later, you can create a render farm by following the steps in the [Update to latest StudioBuilder version](#) tutorial.

UBL Licensing

1. Do you want to enable UBL support for your studio?
 - a. Use the arrow keys to choose if you would like to enable UBL support for your studio or not, and then press the enter (or return) key.
 - b. If you choose Yes, I want to enable UBL support for my studio, see the [Setting up Deadline Usage Based Licensing with Nimble Studio](#) tutorial for more information about integrating UBL with your Nimble Studio cloud studio.

Storage

Use the arrow keys to choose the option that you want, then press the enter (or return) key.

Default storage (FSx for Windows)

1. Are you planning to use Window workstations? If so, we will deploy a default Amazon FSx for Windows for Roaming Profiles.
 - a. If you choose Yes, I want a file system for Roaming Profiles, the command line interface (CLI) will ask you to:

- i. Select the Availability Zone that you want to create this file system in.

 **Note**

If you chose to deploy to the LA Local Zone, we recommend that you also deploy your storage to the LA Local Zone.

- ii. Enter your desired SSD storage capacity in GB. The suggested minimum is 1G/user. We recommend that you start with 40.
 - iii. Enter your desired throughput capacity in MB/s. We recommend that you start with 32.
- b. If you choose No, I don't want a file system for Roaming Profiles, you will receive the next prompt.

CloudWatch Alarms

Amazon CloudWatch alarms are notifications that are triggered based on metrics that are monitored for the AWS services that your studio uses.

1. Enter an email address for CloudWatch Alarms to send warnings to when the given file system's burst credits are under four different thresholds.
 - a. Enter the email address that you would like to use to receive these alarms, and press the enter (or return) key.
 - i. CloudWatch Alarms will email you when the Amazon EFS file system that's storing the Deadline Repository depletes burst credits.

 **Important**

Confirm your subscription to the CloudWatch Alarm emails to receive notification messages when your burst credits are low. Failure to take appropriate action based on these emails could result in your render farm losing the ability to function.

- b. When your deploy is complete, follow the instructions in [Step 7: Confirm subscription to burst alert emails](#).
 - i. For more information about Amazon CloudWatch alarms, see [Using Amazon CloudWatch Alarms - Amazon CloudWatch](#).

- ii. For more information about Amazon Elastic File System (Amazon EFS) performance, see [Amazon EFS performance](#).

Linux home directory

1. Are you planning to use Linux Workstations? We will deploy an EFS file system for persistent home directories.
 - a. If you choose Yes, I want persistent user home directories for Linux workstations, Nimble Studio will deploy an Amazon EFS file system for your studio. You will then be prompted to choose an Availability Zone for where you want to deploy the Amazon EFS file system.

 **Note**

If you chose to deploy to the LA Local Zone, we suggest that you also deploy your storage to the LA Local Zone.

- b. If you choose No, I don't want persistent user home directories for Linux workstations, Nimble Studio won't deploy an Amazon EFS file system.

Additional storage

1. Do you want to add additional storage?
 - a. If you choose No, you will receive the next prompt.
 - b. If you choose Yes, you will be asked Which storage type do you want to use? Use the arrow keys to choose a type of storage and then press the enter (or return) key.
 - i. If you choose FSx for Windows, the command line interface will ask you to:
 - A. Select the Availability Zone that you want to create this file system in.

 **Note**

If you chose to deploy to the LA Local Zone, we suggest that you also deploy your storage to the LA Local Zone.

- B. Enter an ID name for this storage. We recommend the name FSxWindows1.

- C. Add a description for your storage.
- D. Enter your desired SSD storage capacity in GB. We recommend 200.
- E. Enter your desired throughput capacity in MB/s. We recommend 32.
- F. Enter a Windows Drive letter. We recommend D.
- G. Enter the Linux mount path. The path can be any valid folder, except the system folder. For example: /usr/bin, /usr/lib, /usr/lib64 or /usr/sbin. We recommend /mnt/fsxwindows1

ii. If you choose FSx for Lustre, the command line interface will ask you to:

- A. Select the Availability Zone that you want to create this file system in.

 **Note**

If you chose to deploy to the LA Local Zone, we suggest that you also deploy your storage to the LA Local Zone.

- B. Enter an ID name for this storage. We recommend the name FSxLustre1.
- C. Add a description for your storage.
- D. Enter your desired Amazon FSx storage capacity in GiB. The smallest valid value is 1,200. Other valid values begin at 2,400 GiB and increase in 2,400 increments. We recommend 1,200.
- E. Choose the amount of read and write throughput for each 1 TiB of storage, in MB/s/TiB.
- F. Enter the Linux mount path. The path can be any valid folder, except the system folder. For example: /usr/bin, /usr/lib, /usr/lib64 or /usr/sbin. We recommend /mnt/fsxlustre1

iii. If you choose EFS, the command line interface will ask you to:

- A. Select the Availability Zone that you want to create this file system in.

 **Note**

If you chose to deploy to the LA Local Zone, we suggest that you also deploy your storage to the LA Local Zone.

- C. Add a description for your storage.
 - D. Enter the Linux mount path. The path can be any valid folder, except the system folder.
For example: /usr/bin, /usr/lib, /usr/lib64 or /usr/sbin. We recommend /mnt/efs1
2. Do you want to add additional storage?
 - a. If you choose Yes, you run through the previous prompts again.
 - b. If you choose No, you will receive the next prompt.

Render fleet questions

1. Which type of fleet would you like?
 - a. Use the arrow keys to choose the type of fleet that you would like to use for your render farm workers and press the enter (or return) key.
 - b. Choosing Spot Instance will use Spot instances for your render fleet, while On-Demand Instance will use On-Demand instances. With On-Demand instances, you pay for the time that you use them, and while in-use they won't be interrupted. Spot instances are unused On-Demand instances that are available for up to a 90% discount off the On-Demand price, but they might be interrupted if demand for On-Demand instances becomes too high.
 - c. We recommend choosing Spot Instance because of their potential cost savings from using Spot instances. While Spot instances can be subject to occasional interruption, the individual frames of a render job make them well suited to being automatically required with little impact. In addition, Spot instances are automatically launched when needed and automatically terminated when they aren't, resulting in the maximum cost savings for you.
 - d. For more information about Spot and On-Demand instances, see [Instance purchasing options](#).
2. Enter a unique identifier for this fleet. This ID will also be used for your AWS Thinkbox Deadline Group.
 - a. This is the name of the group that your artists will choose when rendering with this fleet.
Example: farm-default.
 - b. You can only use alphanumeric, upper and lowercase letters and the hyphen(-) in your fleet name.

 **Important**

Remember the identifier because you will need it later when configuring Deadline.

3. Which Operating System will this fleet use?

- a. Use the arrow keys to choose either Linux or Windows and press the enter (or return) key. This will be the operating system that is used for your render farm workers.
- b. We recommend choosing Linux because of the cost savings over Windows.

 **Note**

Some digital content creation applications, such as Adobe Creative Cloud and Autodesk 3ds Max, won't run on Linux. Consider this when choosing an OS for your render farm workers.

4. Enter the AMI ID for this Render Worker fleet.

- a. Select the Amazon Machine Image (AMI) to be used for your render workers. The AMI contains the software configuration that will be used for your workers.
- b. For new studios, we recommend leaving this at the default value specified by StudioBuilder, but you can enter your own custom worker AMI, if you like.

5. Enter the EC2 instance type to use for the render workers.

- a. Use the arrow keys to choose the instance type that you would like to use for your render worker fleet and press the enter (or return) key. To start, we recommend choosing the default value of m5.4xlarge, which should be sufficient for testing purposes. You can change the instance type later, depending on your particular render needs.
- b. If you choose **Other**, you will be prompted to **Enter the exact instance type (or types separated by spaces) that you would like to use**. This way, you can create a render fleet consisting of multiple instance types.

 **Note**

If you chose to create a fleet with On-Demand instances, your fleet will only be created with the first instance type that you list.

- c. For more information about the differences in Spot Instance pricing and specifications, see [Amazon EC2 Spot Instances Pricing](#).

6. Enter the Min number of render workers to have running.

- a. Choose the minimum number of render workers that are initially deployed on your farm and press the enter (or return) key. You can change this value later to a different number.

7. Enter the Max number of render workers to have running.

- a. Choose the maximum number of render workers that you want to have running on your farm and press the enter (or return) key. You can change this value later to a different number.

Note

Make sure that the maximum number of render workers that you specify doesn't exceed your quota value. For details, see [Check your Spot Instance quota](#).

8. Would you like to add another fleet?

- a. If you would like to add another fleet at this time, you can. However, you can always add another fleet later, so we recommend pressing N here.

Review

1. Would you like to generate a studio configuration with your selections?

- a. If you're happy with all of the selections that you have made in the preceding steps, enter Y and press the enter (or return) key to proceed. If not, press N to go back and make changes.

Ready to deploy your studio build

1. Enter **BUILD MY STUDIO** and then press the enter (or return) key continue, or enter **QUIT** and then press the enter (or return) key.

- a. Enter BUILD MY STUDIO and press the enter (or return) key to continue.

```
frastrucure.  
A render/worker fleet is a group of instances. You have two choices for instances: Spot Instances or On-Demand Instances.  
Let's configure your Render/Worker Fleet(s)  
* With On-Demand Instances, you pay for the compute capacity that you use, with no long-term commitments. We recommend that you use On-Demand Instances  
    for applications with short-term, irregular workloads that cannot be interrupted.  
* A Spot Instance runs when you provide a Spot Instance request, and utilizes unused EC2 instances. When the Spot price exceeds the maximum price for  
    your request or capacity is no longer available, Amazon EC2 provides a Spot Instance interruption notice, before it stops the instance.  
? Which type of fleet would you like? Spot Instance (Recommended)  
? Enter a unique identifier for this fleet. This ID will also be used for your AWS Thinkbox Deadline Group. farm-default  
? Which Operating System will this fleet use? Linux  
? Enter the AMI ID for this Render Worker fleet. ami-00000000  
? Enter the EC2 Instance type to use for the render workers. m5.4xlarge  
? Enter the Max number of render workers to have running 5  
? Would you like to add another fleet? No  
  
Please review your choices above.  
  
Would you like to generate a studio configuration with your selections (Y)? ('N' to Exit) (Y/N)? Y  
Copying application files to cdkapp  
Writing config file cdkapp/appconfig.yaml  
SUCCESS: Generated CDK application in cdkapp  
  
== Ready to deploy your studio build ==  
  
NOTE: Please ensure that you have subscribed to the following AMIs in the AWS Marketplace in order for your deploy to succeed:  
Nimble Studio Deadline Linux Farm Worker: https://aws.amazon.com/marketplace/pp/B01LZCJH8Q  
Nimble Studio Deadline Windows Farm Worker: https://aws.amazon.com/marketplace/pp/B01LZCJH8Q  
  
? Please type BUILD MY STUDIO (and then press enter) to continue, or type QUIT (and then press enter) to exit. BUILD MY STUDIO
```

- b. After you press the enter (or return) key, StudioBuilder will run the deploy to build all of the components for your studio.

Step 5: Deploy studio with StudioBuilder

StudioBuilder will continue to run in the terminal window until all of the components for your studio are built. This process will take approximately one hour.

Note

If you get disconnected from your StudioBuilder instance, or if the deploy fails, consult the [Troubleshooting](#) section at the end of this tutorial.

As the StudioBuilder app runs, various outputs will display. At times, the output screen might go completely blank. This is normal. You can navigate up in the window to see the current progress of your deploy.

Monitoring using CloudFormation

In addition to watching the output in the EC2InstanceConnect window, you can also monitor the deployment in the AWS Management Console.

1. Sign in to the AWS Management Console and open the [AWS CloudFormation](#) console.
 2. As the deploy runs, it will create new AWS CloudFormation stacks and show you the status of each one. Over the course of the one-hour deploy time, it will create four stacks:
 - a. <your-studio-name>Network
 - b. <your-studio-name>Data
 - c. <your-studio-name>Service
 - d. <your-studio-name>Compute

Your deploy has finished after the status of the Compute stack changes to CREATE_COMPLETE.

After your deploy is complete

After StudioBuilder is finished running, you will be asked what you want to do next. After that has happened, you can close the StudioBuilder browser tab and then terminate your StudioBuilder instance by using the following instructions.

1. Close the EC2 Instance Connect tab that was connected to StudioBuilder.
2. Sign in to the **AWS Management Console** and open the [EC2](#) console.
3. Choose **Instances** in the left navigation pane.
4. Select your **NimbleStudioBuilder** instance from the list of instances.
5. Open the context menu (right-click) for the instance and choose **Terminate Instance**.

 **Note**

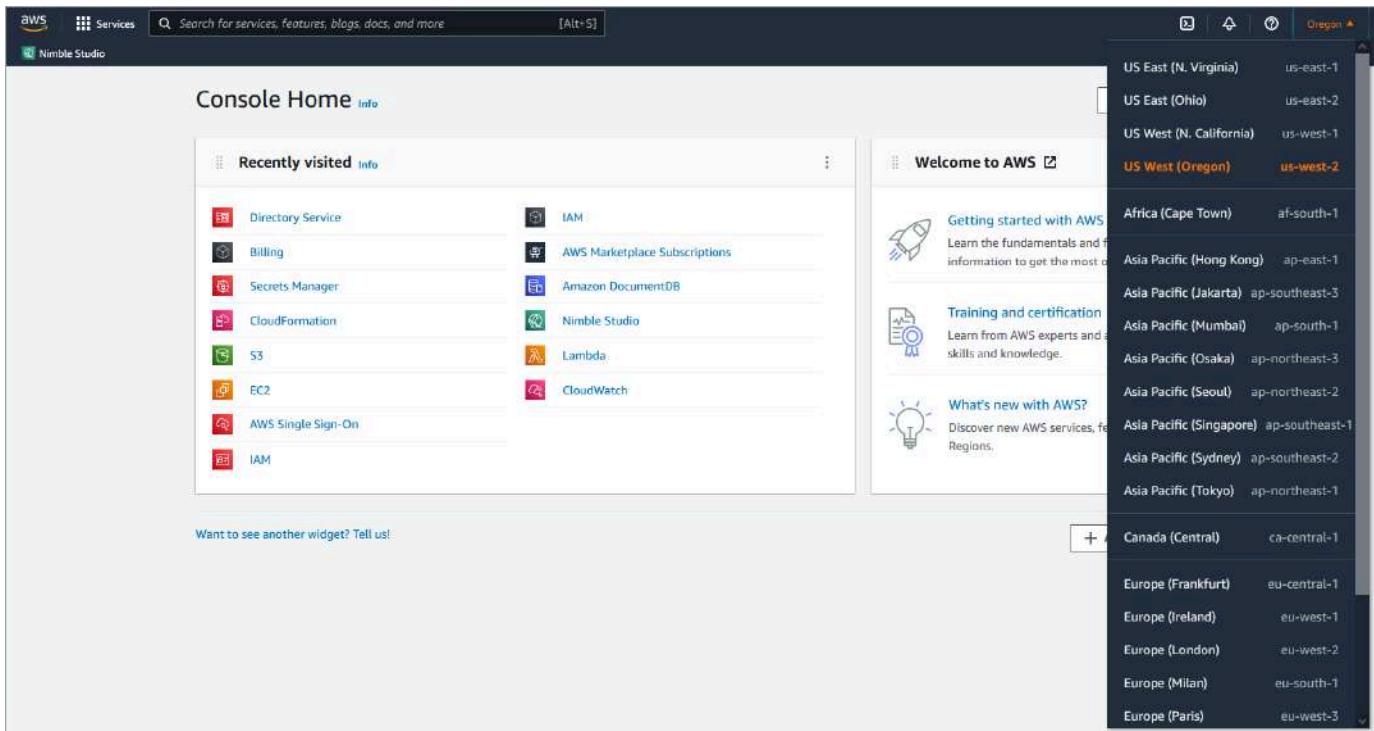
When you're actively deploying a studio, modifying it, or terminating it, we recommend that you run a StudioBuilder instance only. This reduces costs and prevents others from accessing and impacting your studio.

Step 6: Link AWS Managed Microsoft AD as an IAM Identity Center identity source

During the StudioBuilder process, an AWS Managed Microsoft AD is automatically created. This will be the directory that stores your studio users. In this step, you will link that directory to IAM Identity Center so that you can assign roles, such as administrator, to each of your users in Nimble Studio.

To update IAM Identity Center to look at AWS Managed Microsoft AD

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



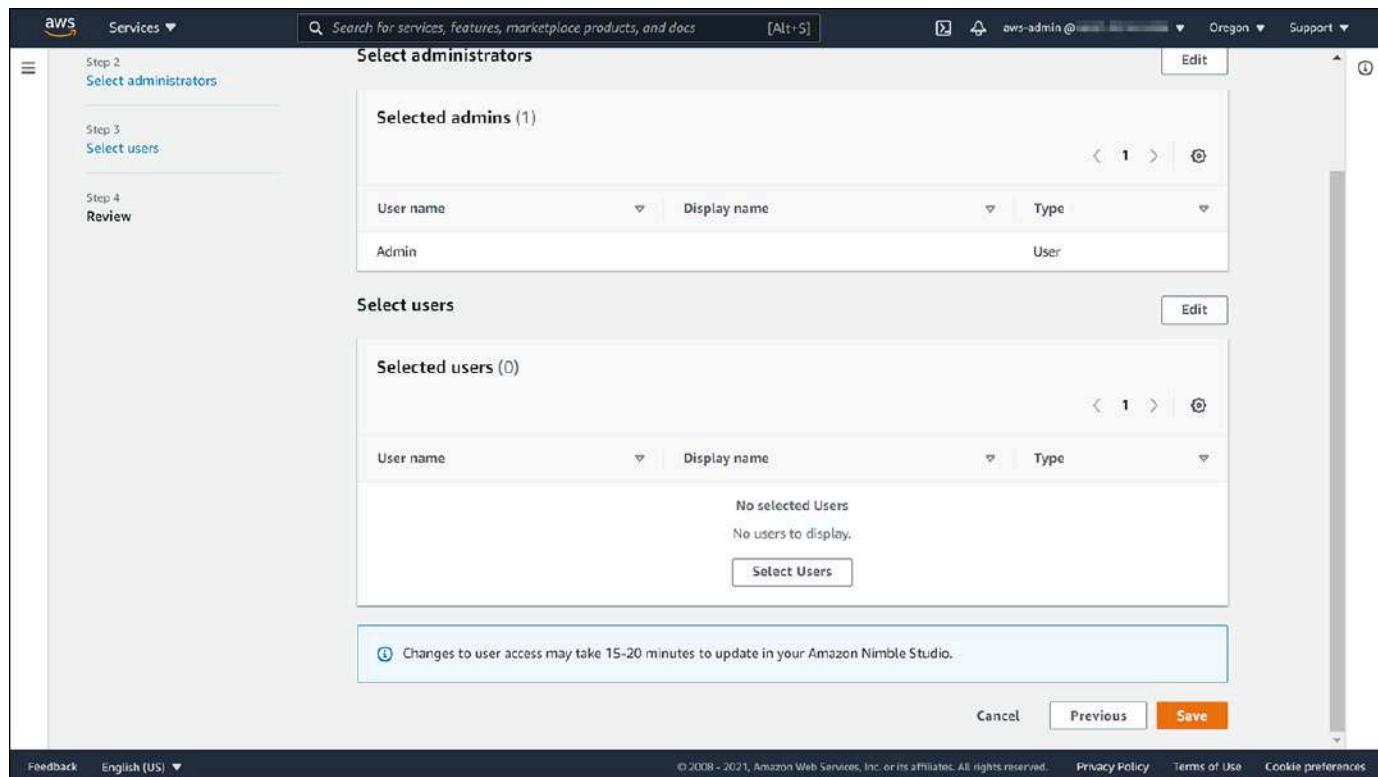
3. Choose **Studio manager** in the left navigation pane.
4. In **Studio setup**, go to **Step 5: Allow studio access** and choose **Allow**.
5. Open the [**IAM Identity Center console**](#) in a new tab.
6. Select **Choose your identity source**.
7. Select **Change identity source**.
8. Select **Active Directory** and choose **Next**.
9. Choose the desired Active Directory and choose **Next**.
10. Enter **ACCEPT** and select **Change identity source**.
11. Wait until the identity source has been changed and the **Settings** page opens.
12. Select **Enable sync**.
 - Read the terms and conditions and if you agree, select the check box next to **I have read the information above and confirm that I want to enable configurable AD sync**.
13. Choose **Users** from the left navigation pane.
 - If you successfully enabled sync, a green banner will display with **Start guided setup**.
14. Select **Manage sync**.
15. Select **Add users and groups**.
16. Choose the **Users** tab. In **User**, enter the exact user name and choose **Add**.

- If this is a new studio, enter **admin** and choose **Add** to get started.
17. When all of the users have been added, select **Submit**.
18. Wait for the users to appear in the IAM Identity Center user pool.
- It can take 10–20 minutes for users to appear.
19. Return to the Nimble Studio tab. On the **Select administrators** page, you will choose which of your users have administrator access to Nimble Studio. Administrators are able to share launch profiles to your studio users later in the [Creating launch profiles](#) tutorial.
- In the **Search text** field, enter **Admin** and choose **Search**.
20. Select **Admin**.
- Make sure to select **Admin** (not administrator). **Admin** is the administrator from your AWS Managed Microsoft AD, so that is what you want to choose here.

The screenshot shows the 'Select administrators' step of the Nimble Studio setup. The search bar at the top contains 'Admin'. Below it, a table lists users under 'Active Directory domain'. The 'Admin' user is selected, indicated by a checked checkbox. The 'Administrator' user is also listed but not selected. At the bottom, a summary shows 'Selected users and groups (1)' with a 'Remove' button.

Name	Display name	Type	Domain
<input checked="" type="checkbox"/> Admin	-	User	[redacted]
<input type="checkbox"/> Administrator	-	User	[redacted]

21. Choose **Next**.
22. On the **Select users** page, since you don't have any users yet, choose **Next**.
23. On the **Review** page, verify that your **Selected admins** is set to **Admin**. Then navigate and choose **Save**.



Step 7: Confirm subscription to burst alert emails

During the deploy, a message will be sent to the email address that you provide. The message asks you to confirm a subscription to a **Burst Alert** email topic. Confirm this subscription to receive important notification messages when the burst credits are low for the Amazon EFS file system that stores your render farm's Deadline Repository. Failure to receive and take action based on these emails could result in your render farm being unable to function.

1. Check the inbox for the email address that you provided when you configured your studio with StudioBuilder.
2. Open the email message with the subject **AWS Notification - Subscription Confirmation**.
 - This email gets sent during the deploy process.
3. Choose the **Confirm subscription** link.
 - a. A new browser tab will open with a message confirming that you have successfully subscribed.
 - b. For more information about Amazon Elastic File System (Amazon EFS) performance, see [Amazon EFS performance](#).

- c. StudioBuilder deploys an Amazon EFS in bursting throughput mode for Deadline's Repository file storage.
 - i. The baseline throughput of the Amazon EFS is 50 KiB/s per 1 GiB stored in the file system. StudioBuilder checks that there is at least 40 GiB of data in this file system during deployment, which provides 2 MiB/s of baseline throughput.
 - ii. In bursting throughput mode, this file system can burst up to 100 MiB/s, with reads counting as 1/3 of their actual throughput for this purpose. Bursting consumes bursting credits from the file system, and those burst credits replenish when not bursting.
 - iii. The alarms configured by StudioBuilder will alert you when your burst credits are on a downward trajectory so that you can take action to increase the baseline throughput of the Amazon EFS by adding additional data to it.

You have completed deploying Amazon Nimble Studio using StudioBuilder in your AWS account. Your new studio is now ready for you to customize. We recommend that you continue with the [Adding studio users](#) tutorial, next.

Troubleshooting

Restart StudioBuilder

StudioBuilder failed to start or I exited StudioBuilder before completing my configuration.

1. Run the following commands to start StudioBuilder.

```
cd /home/ec2-user  
studio_builder/bin/studio_builder
```

2. Return to the [Configure your studio](#) section of this tutorial to continue your deploy.

Continue deployment after disconnection

I accidentally got disconnected from my StudioBuilder's EC2 Instance Connect session before completion.

1. If you were in the process of choosing options for your studio and have not yet entered **BUILD MY STUDIO:**

- a. Follow the instructions in the [Step 4: Configure studio with StudioBuilder](#) tutorial to reconnect to your StudioBuilder instance.
 - b. After reconnecting, find the last configuration question that you answered to continue.
2. If you already entered **BUILD MY STUDIO** and your deploy is running, you don't need to reconnect to your StudioBuilder instance in order for the deploy to complete. Instead, you can monitor your deploy progress from CloudFormation in the AWS Management Console. For more information, see [Monitoring using CloudFormation](#).
 - If you prefer to monitor progress using EC2 Instance Connect, you can reconnect to your StudioBuilder instance using the instructions in [Connect to the instance with EC2 Instance Connect](#).

Re-run a deployment after error and failure

My deploy failed with an error on creating the AWS Managed Microsoft AD.

If your deploy fails on a step related to AWS Managed Microsoft AD, follow the next steps to remove the affected services from your account, and rerun the deploy:

1. Wait for the deploy process to stop.
 - a. If you're still connected to your StudioBuilder instance, wait for the deploy to finish failing.
 - After it finishes, return either to the command line prompt or the StudioBuilder prompt that states Welcome to StudioBuilder! What would you like to do?
 - b. If you're monitoring the deploy in CloudFormation, wait until the status of the <name>Data stack has changed to ROLLBACK_FAILED. The <name> portion of the status refers to the name that you chose for your studio during the StudioBuilder configuration questions.
2. Go to **Services** and open **Amazon DocumentDB**
3. Select **Clusters** in the left navigation pane.
4. If there are no clusters listed, skip to the next step. If you see clusters listed, delete each one individually by following these steps:
 - a. In **Clusters**, select the cluster's name, which will take you to its details page.

- b. On the cluster's details page, choose **Events & tags**.
 - c. Navigate to the **Tags** section.
 - d. If you see a tag key that contains **StudioBuilder**, return to the list of clusters by choosing **Clusters** in the left navigation pane. Follow the next steps to delete the cluster.
 - e. Select the cluster that you need to delete by choosing the check box next to its name.
 - You only need to select the entry with the role **Cluster** and it will automatically delete the entry with the role **Primary**.
 - f. Choose **Action**. Then choose **Delete**.
 - g. Select **No** in the section where you're asked to create a final cluster snapshot and select the check box to acknowledge your choice.
 - h. Enter **delete entire cluster** and then choose **Delete**.
- i. Repeat this step (*step 4, a – h*) for any other clusters that have **StudioBuilder** tags.
5. After the clusters have been deleted, select **Subnet groups** in the left navigation pane.
 6. If there are no subnet groups listed, skip to the next step. If you see subnets listed, delete each one individually by following these steps:
 - a. For **Subnet groups**, select the subnet's name, which will take you to its details page.
 - b. On the subnet's details page, navigate down to the **Tags** section.
 - c. If you see a tag key that contains **StudioBuilder**, return to the list of subnet groups by choosing **Subnet groups** in the left navigation pane. Follow the next steps to delete the subnet group.
 - d. Select the subnet group that you need to delete from the list.
 - e. Choose **Action**. Then choose **Delete**.
 - f. Repeat for each subnet group that has a **StudioBuilder** tag.
 7. Sign in to the AWS Management Console and open the [AWS CloudFormation](#) console.
 8. Select the <name>Data stack from the list of stacks, and then choose **Delete**.
 - a. If you're asked to edit the termination protection, select **Disabled**, then choose **Save** and try the delete process again.
 - b. The status of the stack will change to **DELETE_IN_PROGRESS**. When the <name>Data stack disappears from the list of stacks, or its status changes to **DELETE_COMPLETE**, it has been deleted and you can rerun your deploy.

9. If you're no longer connected to your StudioBuilder instance, follow the instructions in Step 4: [Connect to the instance with EC2 Instance Connect](#).
10. Return to [Step 4: Configure studio with StudioBuilder](#) and repeat the instructions to configure your studio.

Update deployment with a new version of StudioBuilder

How do I update my studio deployment using a new version of StudioBuilder?

1. Go to **Services** and open [AWS Marketplace Subscriptions](#) .
2. Choose **Manage for Nimble Studio StudioBuilder**.
3. On the **Nimble Studio StudioBuilder** page, choose **Launch new instance**.
4. Select the latest version from the **Software version** dropdown menu.
5. Select the AWS Region that your studio is deployed in.
6. Choose **Continue to launch through EC2**.
7. Return to [Step 3: Launch the StudioBuilder EC2 instance](#) to finish launching and connect to your new StudioBuilder instance.

 **Important**

After connecting to your StudioBuilder instance, choose **Update and/or edit your studio** and follow the prompts to update your studio.

Related resources

- [How to delete a studio](#)
- [AWS IAM Identity Center](#)
- [Set up EC2 Instance Connect - Amazon Elastic Compute Cloud](#)
- [Using Amazon CloudWatch Alarms - Amazon CloudWatch](#)
- [Amazon EFS performance - Amazon Elastic File System](#)
- [Instance purchasing options - Amazon Elastic Compute Cloud](#)
- [Amazon EC2 Spot Instances Pricing](#)

How to delete a studio

This tutorial shows how to delete your Amazon Nimble Studio cloud studio from your AWS account so that you no longer pay for its resources. This process involves manually deleting resources that were created by studio admins, and triggering a delete process for anything that was created by StudioBuilder.

Contents

- [Prerequisites](#)
- [Step 1: Delete your Nimble Studio cloud studio](#)
- [Step 2: Remove IAM Identity Center](#)
- [Step 3: Remove storage](#)
- [Step 4: Delete Amazon EC2 resources](#)
- [Step 5: Remove Active Directory](#)
- [Step 6: Remove Deadline Database](#)
- [Step 7: Delete S3 buckets](#)
- [Step 8: Delete CloudWatch Logs](#)
- [Step 9: Remove CloudFormation stacks](#)

Prerequisites

Before you can delete your cloud studio, first complete all of the prerequisites in this section. These prerequisites include the following:

- Back up and safely store all of your data by following the [How to back up your studio data](#) tutorial.
- Check that there isn't anything in your studio that you need. Make sure that everything that remains can be deleted or removed.
- Cancel all Spot Fleet requests, end all render jobs, and shut down render job instances.

Cancel all Spot Fleet requests manually

Deadline creates or modifies a Spot Fleet request for every Deadline Group that has a Spot Fleet configuration. To cancel a Spot Fleet request using the AWS Management Console, follow these steps:

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Spot Requests** in the left navigation pane.
3. Select your Spot Fleet request.
4. Choose **Actions**. Then choose **Cancel Spot request**.
5. In **Cancel Spot request**, confirm that you want to cancel the Spot Fleet. To keep the fleet at its current size, clear **Terminate instances**. When you're ready, choose **Confirm**.

For more information, see [Cancel a Spot Fleet request](#) in the *Amazon EC2 User Guide for Linux Instances*.

End all render jobs

To suspend all render jobs in the Deadline Monitor application, follow these steps:

1. Launch a virtual workstation.
2. Open **Deadline Monitor**.
3. Choose **Tools**. Then choose **Super User Mode**.
4. Select (right-click) all active jobs. This includes jobs that are either rendering or queued.
5. Suspend each job.

Shut down render job instances (worker instances)

Next, you need to shut down all of your render job instances. These are also called worker instances.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Open the **EC2 Dashboard**. In the top **Resources** section, select **Instances (running)**.
3. Select the check box of each **Deadline** instance that is running on your Nimble Studio account.
4. Navigate to **Instance State**.
5. Choose **Terminate Instance** from the dropdown.

6. Repeat this process for other instances that you spun up in your account.
 - a. For example, instances being used as license servers, or to update Amazon Machine Images AMIs.
 - b. You don't need to terminate the **<studioname>Service/RenderQueue/Cluster/RCS Capacity** instance.

 **Note**

Before you delete a studio, it's a good practice to back up your data. To learn how, see [How to back up your studio data](#).

Step 1: Delete your Nimble Studio cloud studio

First, remove all sessions, launch profiles, and studio resources from your studio. Then, remove your studio.

To end all active streaming sessions

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane. Then choose **Streaming sessions**.
3. If any sessions have a status of **Running**, contact the user who owns the session and ask them to end their session.
4. You can also end a session by selecting it and choosing **Actions**. Then choose **Terminate**.

 **Important**

Any data in a running session that hasn't been saved to shared storage will be permanently lost when you terminate a session.

To delete your launch profiles

1. Choose **Launch profiles** in the left navigation pane.
2. Select each **LaunchProfile**.

3. Choose **Action**. Then choose **Delete**.
4. When the **Delete launch profile?** pop-up appears, enter **Delete** in the input field, and choose **Delete**.
5. Repeat this for each **LaunchProfile**.

To delete your studio resources

1. Choose **Studio resources** in the left navigation pane.
2. Select **File Storage**.
3. Select one item at a time.
4. Choose **Action**. Then choose **Remove**.
5. When the **Remove FSxWindows?** pop-up appears, enter **Remove** in the input field, and then choose **Remove**.
6. Repeat this process for the **Compute Farm**, **Amazon Machine Images**, **Active Directory**, and **Custom Configuration** sections.
 - You can't remove the default AMIs, **NimbleStudioWindowsStreamImage**, and **NimbleStudioLinuxStreamImage**.

To delete your studio

1. Choose **Studio manager** in the left navigation pane.
2. In the **Studio name** section, choose **Delete studio**.
3. Select **Check for resources**. After the request is processed, choose **Delete studio**.
4. Enter **Delete** and choose **Delete**.

It takes a few seconds for a studio to delete.

Step 2: Remove IAM Identity Center

1. Sign in to the AWS Management Console and open the [IAM Identity Center](#) console.
2. On the **Dashboard**, select **Choose your identity source**.

1

[Choose your identity source](#)

The identity source is where you administer users and groups, and is the service that authenticates your users.

3. In the **Identity source** section, select the **Actions** dropdown and choose **Change identity source**.
4. Select the **IAM Identity Center** box and choose **Next**.
5. In the input field for the **Review and confirm** section, enter **ACCEPT**.
6. Choose **Change Identity source**. This will take a few seconds.
7. On the **Settings** page, go to the section with three tabs. Choose the **Management** tab.
In the **Delete IAM Identity Center configuration** section, choose **Delete**.
8. When a pop-up appears, select all of the check boxes on it.
In the confirmation input field, enter the IAM Identity Center ID that's provided on the pop-up, and then choose **Confirm**.

Step 3: Remove storage

Some storage services are tied to an Active Directory environment. Delete this storage before you can remove the Active Directory environment. If you don't remove the storage first, you might get errors because of this dependency.

Follow the [Clean up resources](#) tutorial in the *Amazon FSx for Windows File Server User Guide* for all of your Nimble Studio storage and backup resources.

It can take around 20 minutes to delete your storage and backups. You can continue to the next step while your storage is being deleted.

Step 4: Delete Amazon EC2 resources

Next, remove some Amazon Elastic Compute Cloud (Amazon EC2) resources, including your Auto Scaling groups, launch configurations, load balances, instances, Amazon Machine Images (AMIs), volumes and snapshots, and key pairs.

 **Note**

You can identify your Nimble Studio resources by finding the resources with <your-studio-name> or NimbleStudio in their name.

To delete your Auto Scaling groups

Follow the instructions in the [Delete your Auto Scaling group](#) section of the *Elastic Load Balancing User Guide* to delete all of your Nimble Studio Auto Scaling groups.

To delete your launch configurations

Follow the instructions in the [Delete the launch configuration](#) section of the *Amazon EC2 Auto Scaling User Guide* to delete all of your Nimble Studio launch configurations.

To delete your Load Balancers

Follow the instructions in the [Delete an Application Load Balancer](#) section of the *Elastic Load Balancing User Guide* to delete all of your Nimble Studio load balancers.

To delete your EC2 instances

Follow the instructions in the [Terminate an instance](#) section of the *Amazon EC2 User Guide for Linux Instances* to delete all of your Nimble Studio instances.

To deregister your AMIs and delete snapshots

Follow the instructions in [Clean up your Amazon EBS-backed AMI](#) to delete all of your Nimble Studio AMIs and snapshots.

To delete your volumes

Follow the instructions in the [Delete an Amazon EBS volume](#) section of the *Amazon EC2 User Guide for Linux Instances* to delete all of your Nimble Studio volumes.

To delete your key pairs

Follow the instructions in [Delete your public key on Amazon EC2](#) to delete all of the key pairs associated with Nimble Studio instances.

Step 5: Remove Active Directory

Next, remove the Active Directory environment. It can be difficult to remove it because other services might be connected to it. If that's the case, you will be unable to delete the Active Directory environment.

Follow the instructions in the [Delete your directory](#) tutorial of the AWS Directory Service Administration Guide to delete your Nimble Studio Active Directory.

1. Select the Nimble Studio Active Directory.
 - This is named ad.<your-studio-name>..nimble.<studio-region>.aws
2. Disable any service that has **Status: Enabled**.
 - That means that service is still connected and has to be deleted first before you can proceed.

Step 6: Remove Deadline Database

You only need to remove the Deadline database if you deployed a render farm.

Follow the instructions in the [Deleting an Instance Using the AWS Management Console](#) tutorial to delete your Nimble Studio Amazon DocumentDB instance.

1. When choosing the instance that you want to delete, choose the root item in the list. Choose the root item because it will delete all other items contained within that cluster.
2. To determine which cluster is affiliated with Nimble Studio, select the cluster. Then, in the **Security Groups** section, see if the security group has <your-studio-name> in its name ID.

Now, follow the instructions in the [Delete a snapshot](#) tutorial to delete all of your snapshots.

Step 7: Delete S3 buckets

There are a few S3 buckets that Nimble Studio deploys to store security logs and other things. You can now safely remove these.

Follow the instructions in the [Emptying a bucket](#) and the [Deleting a bucket](#) tutorials in the *Amazon Simple Storage Service User Guide* to delete your Nimble Studio buckets.

1. Empty and delete all buckets that have a name beginning with #your-studio-name>
2. Empty and delete the bucket with a name beginning with cdktoolkit-stagingbucket-

Important

There's a chance that you have other S3 buckets with your data. Don't empty or delete these buckets.

Step 8: Delete CloudWatch Logs

Your AWS account tracks what happens in your account and makes that data available in CloudWatch logs. Because you're no longer using your studio, you can delete your CloudWatch logs. The following steps show you how.

1. Sign in to the AWS Management Console and open the [CloudWatch](#) console.
2. Select **Log groups** in the **Logs** section.
3. Select all log groups in the table.
4. Choose **Action**. Then choose **Delete**.

Step 9: Remove CloudFormation stacks

Finish deleting your studio by automatically deleting the rest of your resources by using CloudFormation. Follow the instructions in the [Deleting a stack on the AWS CloudFormation console](#) tutorial in the *AWS CloudFormation User Guide*.

1. Delete the stacks in the following order: **Compute**, **Service**, **Data**, **Network**, and **CDKToolkit**.
 - If a dialog box warns that the stack is protected by deletion protection, choose **Edit termination protection**. Choose **Disable**. Then choose **Save**.
2. The deletion can fail for a variety of reasons. If it's caused by a **Timeout**, try to delete the stack again.
 - Other deletion fails are listed in an error message in the **Events** tab, when you select the stack name.

Your studio is now fully deleted. If you would like to deploy a new studio, see the tutorial for [Deploying a new studio with StudioBuilder](#).

Administration

These tutorials will guide you through the process of deploying and configuring a new studio, adding users, and customizing infrastructure to fit your team's needs.

Now that you have your basic Nimble Studio deployment, it's time to start customizing your setup to fit your team's needs. These tutorials explain how to change the administrator password, update your StudioBuilder version, back up your studio, modify your launch profiles, and more.

Topics

- [Creating launch profiles](#)
- [Modifying launch profiles](#)
- [Creating custom configurations](#)
- [How to back up your studio data](#)
- [Update to latest StudioBuilder version](#)
- [Change administrator password in AWS Directory Service](#)
- [Rotating certificates created in Nimble Studio](#)

Creating launch profiles

This administrator tutorial shows you how to create launch profiles. To learn how to update launch profiles, see the [Modifying launch profiles](#) tutorial.

Launch profiles control access to resources in your studio such as compute farms, shared file systems, instance types, and Amazon Machine Images (AMIs). StudioBuilder creates two default launch profiles for you, the first time that it runs: **Workstation-Default** and **RenderWorker-Default**. The **Workstation-Default** launch profile is what your artists will use to launch virtual workstations. You can use the **RenderWorker-Default** launch profile to manage rendering resources and their access to storage.

To customize your team's access to studio resources, this tutorial shows you how to create your own launch profiles.

Contents

- [Prerequisites](#)

- [Step 1: Create a launch profile by copying an existing one](#)
- [Step 2: \(Optional\) Perform a test launch](#)
- [Step 3: Share launch profiles with studio users](#)
- [Troubleshooting](#)
- [Related resources](#)

Estimated time: 20 minutes

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- Optionally, you can complete the steps in the [Adding studio users](#) tutorial before starting this one. By adding users first, you can assign your new launch profile to specific users at the end of this tutorial. However, if you prefer to create your launch profiles first, you can return to the end of this tutorial after adding users.

Step 1: Create a launch profile by copying an existing one

The easiest way to create a new launch profile is to copy an existing one. From there, you can make edits, so that the new profile is configured in the way that you want. The added benefit of copying an existing launch profile is that you can start with default settings for each element of the launch profile.

To copy an existing launch profile

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profile that you want to copy.
 - If this is your first new launch profile, you will be copying the **Workstation-Default** launch profile that was created by StudioBuilder.
4. Choose **Actions**. Then choose **Copy to new**.
5. Give your launch profile a **name**.

- To keep track of launch profiles, you can include details such as the version number, a date, or a relevant studio department.
 - Example: animation-2021-03-08, studio-v003
6. Give your launch profile a **description**.
- a. Describe what makes this launch profile unique in your description.
 - b. Example: "Added Adobe Photoshop and Yeti; added X: drive; increased instance size to XLarge (g4dn.16xlarge)."
7. In the **Amazon Machine Images (AMI)** section, choose the **Amazon Machine Images** that you want to be available to artists when they launch workstations with your new launch profile.
- a. If you specify a non-gp3 EBS volume type (gp2, io1, io2, etc.) for your AMI, it converts to a gp3 EBS volume for your Nimble Studio workstations. You only pay for gp3 EBS volumes.
 - b. If you haven't created any new AMIs yet, choose either the Windows or Linux default AMI.
 - c. If you have created a new workstation AMI, choose the new one that you made.
 - d. You can select multiple AMIs for your launch profile. Multiple AMIs give the artist a choice when launching. This is useful if you allow artists to choose their operating system, such as Linux or Windows.
8. In the **Launch profile components** section: Select the components that you want artists to access when they launch workstations with your new launch profile.
- If you haven't created any new storage or farm components, keep these as default.
9. In the **Subnet IDs** section, don't modify the subnets, unless you want to launch workstations in a different Availability Zone than the one you chose during StudioBuilder deployment.
10. In the **Virtual workstation type** section, choose which instance type that you want your artists to have access to when using this launch profile. Enter the maximum consecutive hours that instances can be in the Running and Stopped states.
- a. To learn about various instance sizes, such as how many vCPUs, GPUs, and how much memory they have, see the **Product Details** section of the [Amazon EC2 G4 Instances page](#).
 - b. To use G5 instances, increase your quota limit for G5 streaming sessions per studio. For instructions about how to increase your quota limit, see [\(Optional\) Request a quota increase for G5 streaming sessions](#).
 - c. For information about pricing, see [Amazon Nimble Studio Pricing](#).

11. Choose to keep persistent storage enabled in the **Streaming session storage volume** section.

With persistent storage, you can [Start and stop workstations](#). To keep persistent storage, configure the gp3 EBS volume by entering the following information:

- a. For **Size**, enter the size of the volume, in GiB. For more information, see [Constraints on the size and configuration of an EBS volume](#).
- b. For **IOPS**, enter the maximum number of input/output operations per second (IOPS) that the volume should provide.
- c. For **Throughput**, enter the throughput that the volume should provide, in MiB/s.
- d. Enter the maximum consecutive hours that instances can be in the **Running** and **Stopped** states. To learn more about **Running**, **Stopped**, and **Terminated** states for workstations, see [Starting and stopping workstations](#).
- e. To enable backups, select **Turn on auto backup** in the **Auto backup** section. Then, choose the **Maximum backups per streaming session** and select the box next to the agreement. For more information about automatic backups, see [Session auto backup](#).

12. In the **Streaming session upload** section, select the box next to **Enable uploads**.

- a. In the **Linux upload location**, enter a valid directory path. This is the location where files will be uploaded for Linux workstations. Make sure that the artists who will use this launch profile have access to the folder that you choose. If you don't specify a location, the directory path defaults to \$HOME/Downloads.
- b. In the **Windows upload location**, enter a valid directory path. This is the location where files will be uploaded for Windows workstations. Make sure that the artists who will use this launch profile have access to the folder that you choose. If you don't specify a location, the directory path defaults to %HOMEPATH%\Downloads.

13. (Optional) Add tags if you're using tags to track your AWS resources.

14. Select **Create launch profile**. This creates a launch profile and checks that it can be launched.

- a. Validating the launch profile doesn't mean that the launch profile will work, but it can detect known failures that would prevent it from launching.
- b. If the validation succeeds, the launch profile goes into the Ready state. However, if the validation fails, the launch profile goes into the Impaired state. Regardless of the validation state, the launch profile is created.

- c. If your validation failed, an error banner will display at the top of the menu bar. If the failure is because a dependent service has an outage, the validation is cancelled, and your launch profile isn't marked as impaired.
- d. You can view the details of your launch profile validation in the **Launch profile details** section of your launch profile.

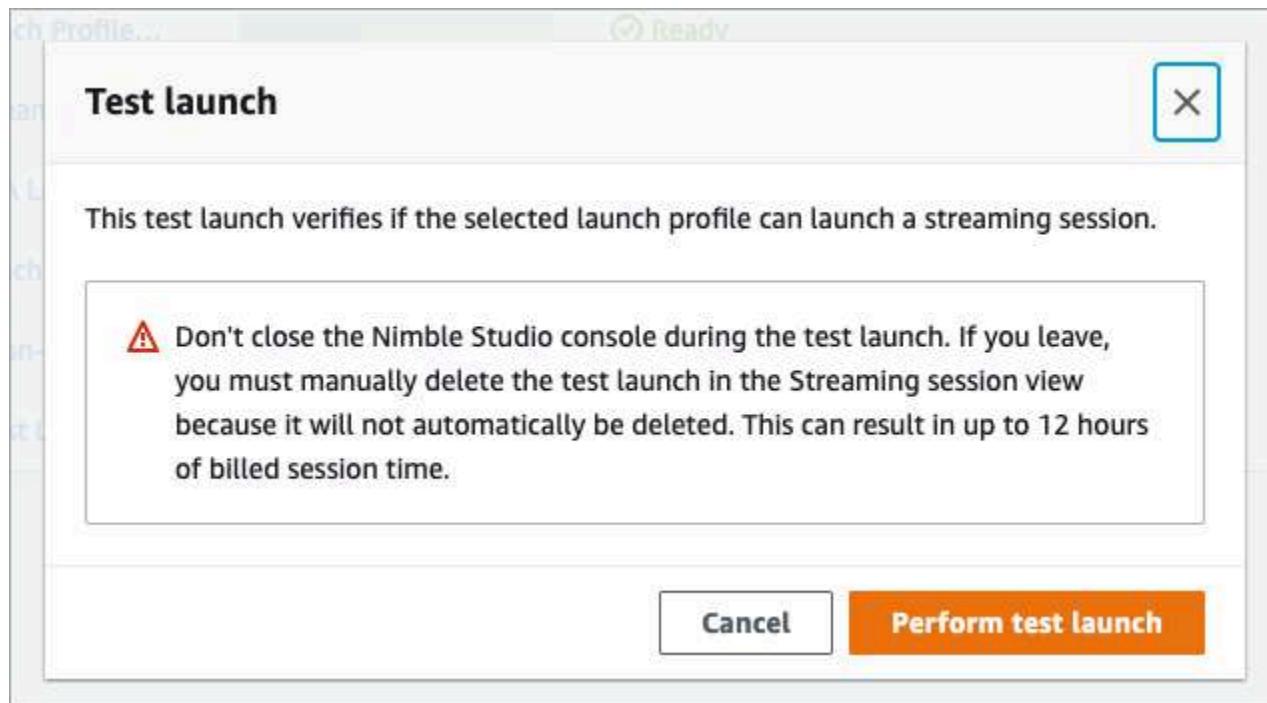
Step 2: (Optional) Perform a test launch

Before you launch a virtual workstation with the new launch profile, you can test if the launch profile can create a streaming session.

 **Note**

A test launch is the same as a regular launch, so the charges are the same for both.

1. Choose the launch profile that you just created.
2. For the **Actions** dropdown, choose **Perform test launch**.
 - You can only run a test launch on a launch profile that has the status Ready or Impaired.
3. Choose **Perform test launch**.



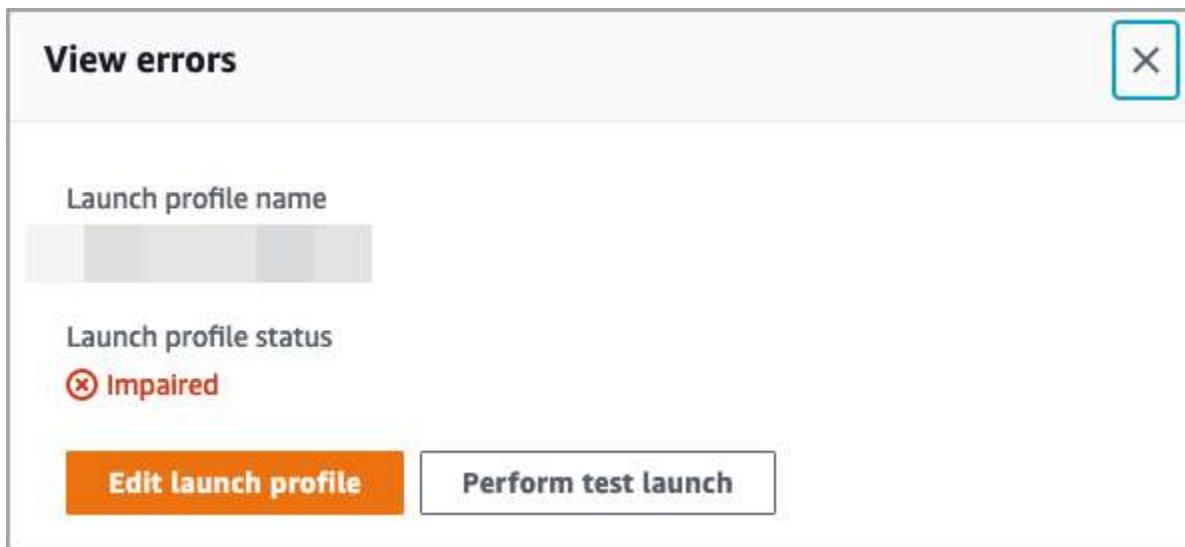
4. The status of your launch profile will change to Performing test launch.

Launch profiles						Actions ▾	Create launch profile
Name	ID	Status	Active Sessions	Region	Description		
		Performing test launch	Performing test launch X	us-west-2	This Workstation...		

- The amount of time it takes to finish the launch test depends on the AMI. After one hour, the request times out (expires) and the test launch fails.
5. If the test launch is successful, the launch profile goes into the Ready state. However, if the test launch fails, the launch profile goes into the Impaired state.
- a. To troubleshoot a failed test launch, select the status of the failed launch. When the **Impaired: Streaming session launch failed** bubble appears, select the **View errors** link.

Ready	us-west-2
Impaired	est-2
Ready	us-west-2

- b. Alternatively, choose the launch profile that you just created and select **Actions**. Then choose **View errors**.

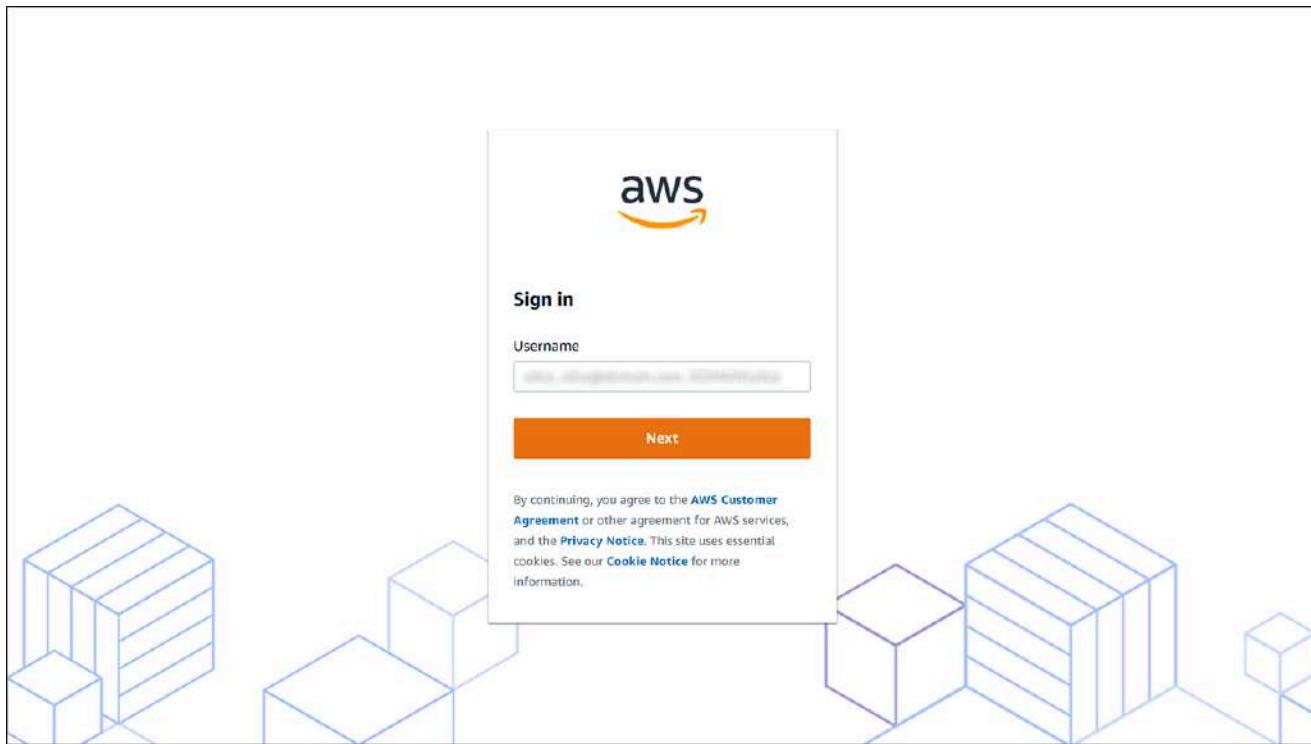


6. In the **View errors** section, you can either choose **Edit launch profile** to adjust the settings, or **Perform test launch** to retry the test launch with the unmodified launch profile.
7. You can view the details of your test launch in the launch profile details section of your launch profile.

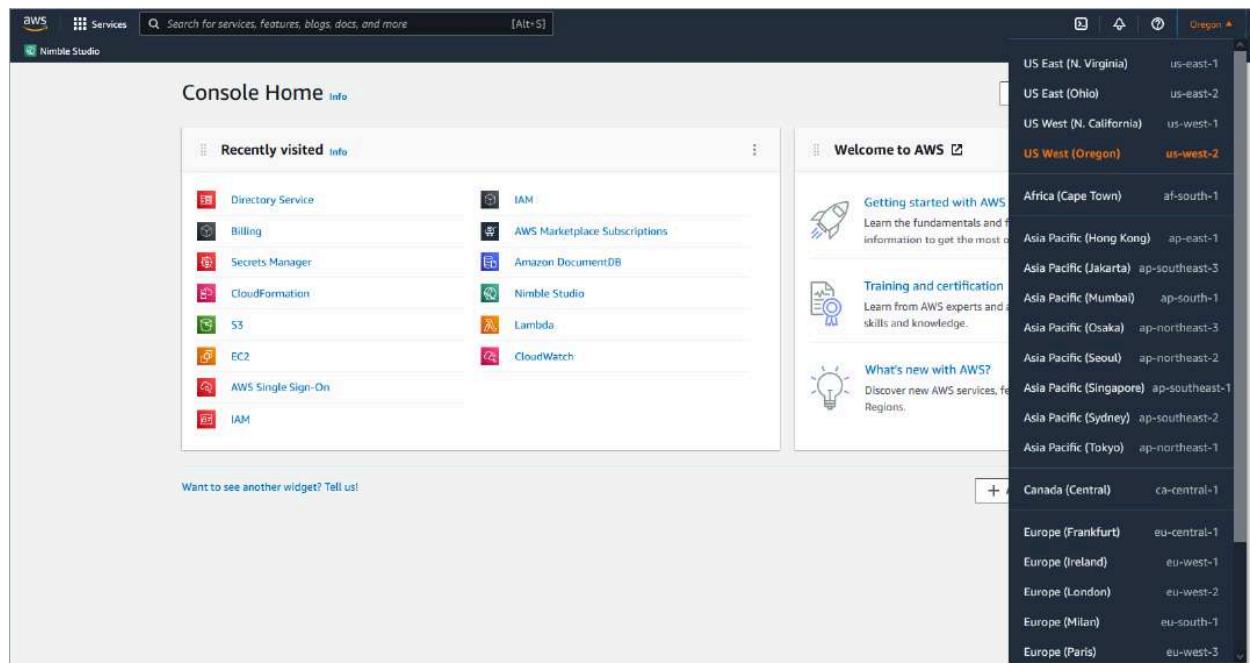
Step 3: Share launch profiles with studio users

By default, studio admins can use all launch profiles. However, for other users to access launch profiles, admins must share these in the Nimble Studio portal. The following instructions show you how.

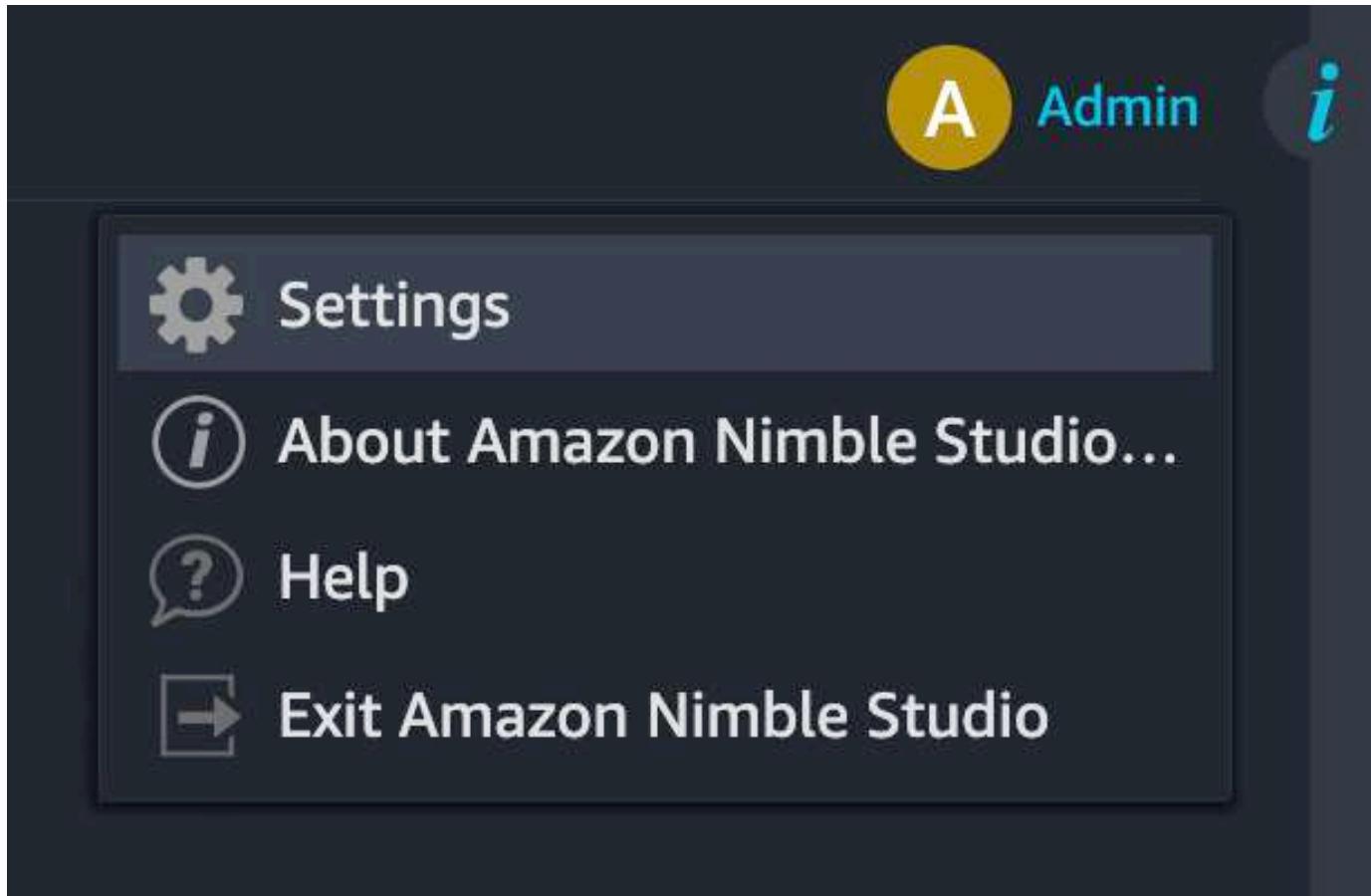
1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. On the **Studio manager** page, choose **Go to Nimble Studio portal**.
3. Sign in to the Nimble Studio portal using your AWS Managed Microsoft AD administrator credentials.
 - a. Enter **Admin** as the user name. Use the password that you set up during StudioBuilder deployment.



- b. If you forgot your password, do the following:
- Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
 - In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



- iii. Select the **Directory ID** for your studio's Active Directory.
 - iv. Choose **Reset user password**.
4. In the upper-right corner of the Nimble Studio portal, choose your **user name**.
5. Choose **Settings** from the dropdown menu.



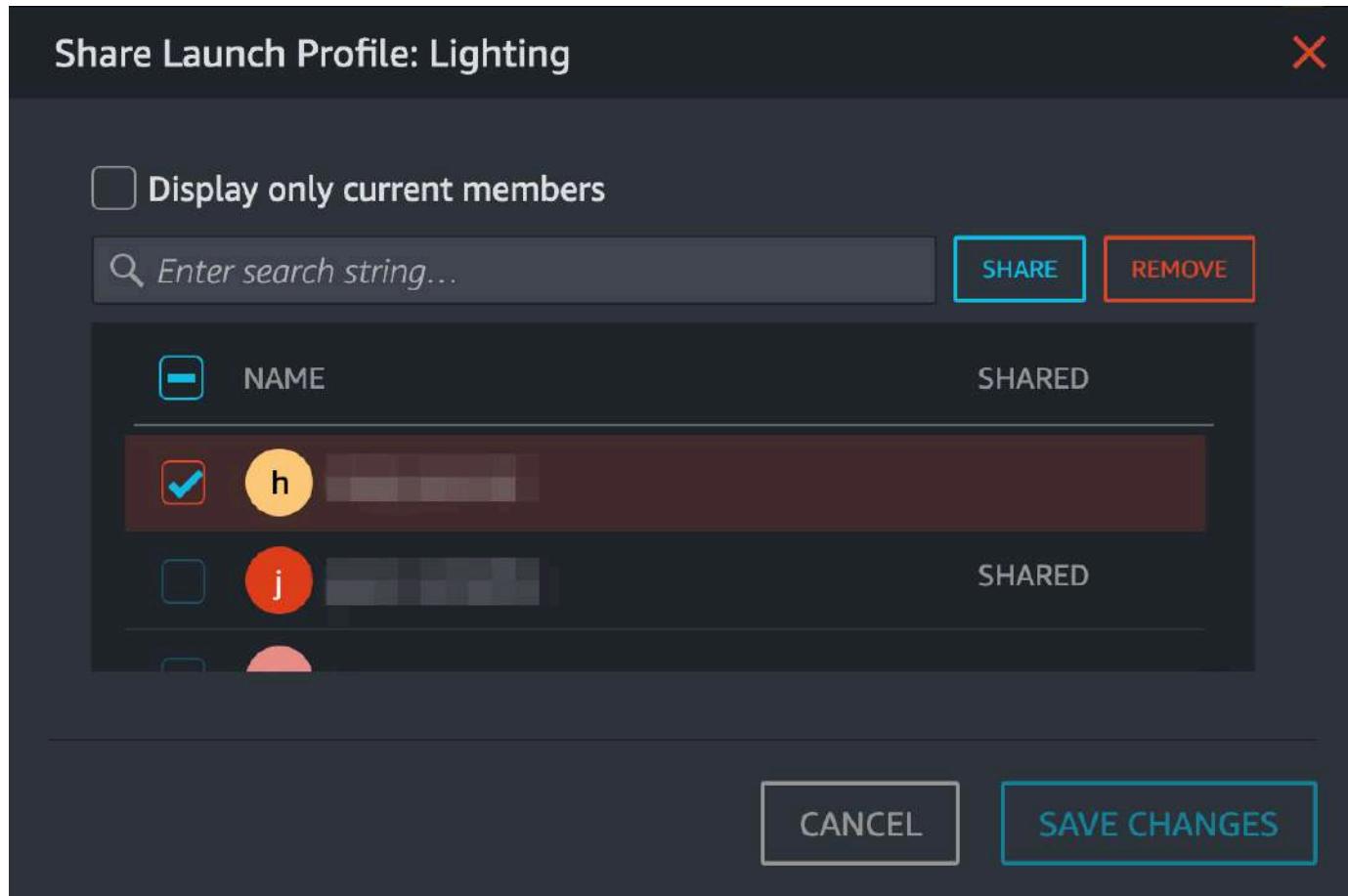
6. Choose **Launch profiles** in the left navigation pane.

LAUNCH PROFILES	SHARED WITH	STATUS	ACTIONS
<u>3D Animation</u>	k c k n a j d		
<u>Rigging</u>	A		
<u>FX Simulation</u>	j a j		
<u>2D Animation</u>	j		
<u>Comp Artist</u>	c c h a		
<u>Lighting</u>	c k h a d		
<u>Surfacing</u>	j A		

7. Locate the launch profile that you want to share.
8. Choose the **share icon** next to it.



9. Select the user or users who should have access to the launch profile.
 - You can search for user names in the search field to help you find users.
10. Choose **Share** to grant the user access to the launch profile.
11. Choose **Save changes**.



12. When the user reloads their page, they will see the launch profile that you shared with them.

Now that you've created launch profiles, you can also add new studio components to them, make updates, and customize your studio setup.

Troubleshooting

I get an error when trying to create a new launch profile.

The default quota value for the number of launch profiles is 50. Therefore, if you got an error when choosing **Create launch profile**, you might have exceeded your studio's limit.

To correct this error, you can either remove launch profiles that you don't need by following [Remove launch profile](#) instructions. Or you can request a quota increase for more launch profiles. To request a quota increase, see [Request a quota increase](#) in the **Setting up to use Nimble Studio** tutorial.

I need to upgrade my NVIDIA drivers

Upgrade your NVIDIA drivers so that the NVIDIA GRID version is 13.1 or later. For instructions about how to upgrade your NVIDIA drivers, see [NVIDIA](#).

Related resources

- [Amazon EC2 G4 Instances](#)

Modifying launch profiles

This administrator tutorial shows you how to update and remove existing launch profiles. To learn how to create launch profiles, see the [Creating launch profiles](#) tutorial.

Contents

- [Prerequisites](#)
- [Update launch profile](#)
- [Validate launch profile](#)
- [Remove launch profile](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- Optionally, you can complete the steps in the Adding studio users [Adding studio users](#) tutorial and the [Creating launch profiles](#) tutorial first so that you understand how to create and assign launch profiles to specific users.

Update launch profile

If you change an underlying studio component of a launch profile, you can test their impacts on your launch profile and verify that the launch profile can still launch.

To update a launch profile

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profile that you want to update.
4. Choose **Action**. Then choose **Edit**.
5. Update the launch profile to your specifications.
6. Select **Update launch profile**. This updates the launch profile and checks that it can be launched.
 - a. Validating the launch profile doesn't mean that the launch profile will work, but it can detect known failures that would prevent it from launching.
 - b. If the validation is successful, the launch profile goes into the Ready state. However, if the validation fails, the launch profile goes into the Impaired state. Regardless of the validation state, the launch profile is updated.
 - c. If your validation failed, an error banner will display at the top of the menu bar. If the failure is due to a dependent service with an outage, the validation is aborted and your launch profile isn't marked as impaired.
 - d. You can view the details of your launch profile validation in the **Launch profile details** section of your launch profile.

You've now successfully verified and updated your launch profile.

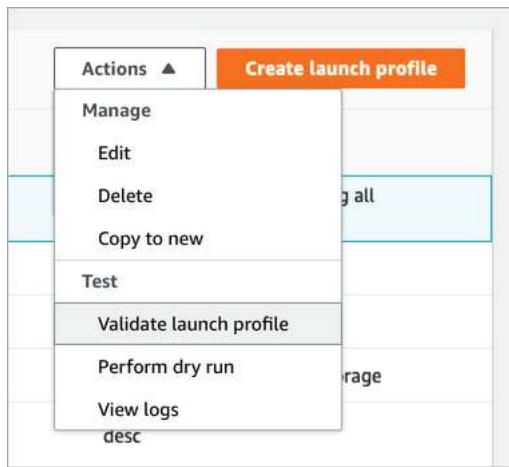
Validate launch profile

If you change a configuration or a dependency outside of Nimble Studio and you want to check the change didn't impact your existing launch profiles, you can validate your launch profiles. Verifying your launch checks that your launch profile has a valid configuration without launching anything.

Alternatively, you can perform a [Step 2: \(Optional\) Perform a test launch](#) to have the highest confidence that your launch profile can successfully launch. However, a test launch takes longer because it actually launches a session.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profile that you want to validate.

4. Choose **Actions**. Then choose **Validate launch profile**.



You've now successfully validated your launch profile.

Remove launch profile

You can reduce the need for quota increase requests by removing unwanted launch profiles.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profile that you want to remove.
4. Choose **Action**. Then choose **Delete**.
5. Wait for the **Status** of the launch profile to change to Deleted.

You've now successfully deleted your launch profile.

Creating custom configurations

Administrators can use custom components to configure instances. This allows admins to set up and control additional properties of their streaming workstations. Custom components use PowerShell scripts for Windows, and shell scripts for Linux instances. These configurations can be added to launch profiles for retrieval. After you create custom configurations, you can add resources to your workstations, and run custom scripts on your instance, system, and user initialization.

This tutorial explains how to create and attach custom configurations. Example custom configurations and their uses are also provided.

Contents

- [Prerequisites](#)
- [Step 1: Create the custom configuration](#)
- [Step 2: Attach custom configuration to a launch profile](#)
- [Custom configuration examples](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.

Step 1: Create the custom configuration

First, create the custom configuration. This section provides several examples of custom configurations.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. Choose **Add** in the **Custom configuration** studio resource type.
4. For each section, enter the following information.
 - a. **Region:** Select the AWS Region that your studio is deployed in.. This is prefilled with the correct value.
 - b. **Custom configuration name:** Enter the name associated with this configuration. You can reference the custom configuration name in launch profiles later.
 - c. **Custom configuration description:** Enter an optional description for this custom configuration.
 - d. **Parameter name:** Enter the name for the parameter. You can use the parameter name as a key variable in the initialization scripts later.
 - e. **Parameter value:** Enter the value of the parameter. You can use the parameter value in the scripts to replace the name (key) during runtime.

- i. Parameter values give you a way to add variables into scripting. Parameter values simplify replacing values.
 - ii. You can add multiple parameters to each custom configuration.
 - iii. For some examples of parameters, see the [Custom configuration examples](#).
- f. (Optional) IAM Roles: Choose the IAM role that you want to associate with this custom component. Users with this IAM profile can access different AWS services through their launch profiles.
- i. During IAM role creation, select the **Choose a service to view use case** dropdown. Enter **Nimble Studio** and choose **Nimble Studio**.
 - ii. Select the radio button next to **Nimble Studio - Allows Nimble Studio resources access to AWS resources**. Then choose **Next**.
 - iii. Add permissions to the newly created role to grant it access to the AWS resources you need.
 - iv. Choose an **Initialization role**. This role provides temporary access to AWS resources from system initialization scripts. For initialization role examples, see the [Custom configuration examples](#) section.
 - v. Choose a **Runtime role**. This role provides runtime access to AWS resources anytime that the instance is running.
- g. **Initialization scripts:** Define the Windows PowerShell or Linux shell scripts that run during system initialization time, during user initialization time, or during both.
- For some examples of initialization scripts, see the [Custom configuration examples](#).
- h. **Security groups:** Choose the security group that you want to be associated with this custom configuration.
- Security groups allow administrators to open new ports on instances so they can perform certain operations on the instance. This could include opening ports for custom file storage support, or ports to communicate with license servers.
5. (Optional) Add tags if you're using tags to track your AWS resources.
 6. Choose **Save custom configuration**.

Step 2: Attach custom configuration to a launch profile

Next, attach your custom configuration to a launch profile. This custom configuration is used to program launch profiles to run the attached scripts.

1. Choose **Launch profiles** in the left navigation pane.
2. Select the launch profile that you want to add the custom configuration to.
3. Choose **Action**. Then choose **Edit**.
4. Navigate to **Launch profile components**.
5. Choose the check box next to the custom component that you created in [Creating custom configurations](#).
6. Choose **Update launch profile**.

You have now created and attached a custom configuration to a launch profile. To see an example of when you can use a custom configuration, see the [Provide Superuser access for Linux users](#) tutorial.

Custom configuration examples

The following examples provide parameter values and initialization scripts for custom configurations. They also provide a summary of when you can use this custom configuration.

Custom configuration to hide the Windows Server network wizard after user login

This example creates a custom configuration component that sets up a Group Policy Object (GPO) for the AWS Managed Microsoft AD. This GPO hides the network wizard, which would otherwise show when a user logs into a Windows Server instance.

Script parameters

- **Parameter name:** GPOName **Parameter value:** "HideNetworkWizard"
- **Parameter name:** GPOComment **Parameter value:** "Hides the Network Wizard at Login"
- **Parameter name:** RegKey **Parameter value:** "HideNetworkWizard" -Context User -Key "HKCU \Software\Microsoft\Windows NT\CurrentVersion\Network\NwCategoryWizard" -ValueName "Show" -Type DWORD -Value 0 -Action Update

Windows system initialization scripts

The following Windows system initialization script uses the script parameters from the previous section to set up a GOP for the AWS Managed Microsoft AD.

```
New-GPLink -Name "HideNetworkWizard" -Target $target
$domain = (Get-WmiObject Win32_ComputerSystem).Domain
$split = $domain.Split(".")
$target = 'ou=' + $split[0] + ',dc=' + $split[0] + ',dc=' + $split[1] + ',dc=' +
$split[2] + ',dc=' + $split[3]

New-GPO -Name "HideNetworkWizard" -Comment "Hides the Network Wizard at Login"

Set-GPPrefRegistryValue -Name "HideNetworkWizard" -Context User -Key "HKCU\Software
\Microsoft\Windows NT\CurrentVersion\Network\NwCategoryWizard" -ValueName "Show" -Type
DWORD -Value 0 -Action Update
```

Set the NICE DCV server priority

This example creates a custom configuration component that can increase or decrease the process priority for the NICE DCV streaming server. This is helpful when users run CPU-heavy applications because it prioritizes the streaming server.

Script parameters

- Parameter name:** WinExecutable **Parameter value:** "dcvserver.exe"
- Parameter name:** WinPriority **Parameter value:** "256"
- Parameter name:** LinuxExecutable **Parameter value:** "/user/bin/dcvserver"
- Parameter name:** LinuxPriority **Parameter value:** "19"
- Parameter name:** LinuxConfigFile **Parameter value:** "/etc/dcv/dcv.conf"

Windows system initialization scripts

```
Get-WmiObject Win32_process -filter 'name = "dcvserver.exe"' | foreach-object
{ $_.SetPriority(256) }
```

Linux system initialization scripts

```
PSID=ps aux | grep -i /usr/bin/dcvserver | tr -s ' ' | cut -d ' ' -f 2 | head -n 1
```

```
renice -n 19 -p $PSID
```

Use a studio component initialization role

This example shows how to retrieve a secret from AWS Secrets Manager by using the *initialization role*. Credentials for the initialization role aren't ever made accessible to the workstation user. This role gives administrators the required AWS access to provision a machine before the user gains access to the workstation.

To use this initialization script, first create a Secrets Manager secret by following the [Create a secret](#) tutorial in the *AWS Secrets Manager User Guide*.

- For **Key**, enter example-secret-key and enter example-secret-value as the **Value**.
 - In production, example-secret-key could be an API key required to perform some one time setup to provision the workstation.
- For **Name**, enter a name. For example, example-secret.
- Notice the secret's Amazon Resource Name (ARN). You will reference this ARN in when creating the initialization role.
- Notice the secret's AWS Region. You will need the AWS Region so that you can retrieve the secret from the AWS CLI in the system initialization script.

Next, create an initialization IAM that can access the secret by following the [Creating a role for an AWS service \(console\)](#) tutorial in the *IAM User Guide*.

- Choose Nimble Studio as a **trusted entity**.
- During role creation, select **Custom trust policy** on the **Select trusted entity** page. Then enter the following policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "identity.nimble.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:DecodeJWT",
                "sts:GetSessionToken"
            ]
        }
    ]
}
```

```
        "sts:TagSession"
    ]
}
]
```

- Choose **Create policy** and enter the following JSON text into the JSON editor. Replace **<SECRET_ARN>** with the ARN of the secret that you created earlier.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GetExampleSecretValue",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "<SECRET_ARN>"
        }
    ]
}
```

- Name this policy **GetExampleSecretValue** and add this policy to the role that you're creating.
- For **Role name**, enter **ExampleStudioComponentInitRole**.

Choose this role when you create a studio component. When you add Nimble Studio as a trusted entity, this role will display in the dropdown in the IAM roles section of the **Create Studio Component** page.

Linux system initialization scripts

Now that you've attached the **ExampleStudioComponentInitRole** to the studio component, the credentials corresponding to that role will automatically be available from your system initialization scripts. You can access these credentials through the **AWS_ACCESS_KEY_ID**, **AWS_SECRET_ACCESS_KEY**, and **AWS_SESSION_TOKEN** environment variables. This means that you can use the AWS CLI or SDK without any manual credential management. If your studio component requires an API key to provision a machine with the functionality that it provides, use the following script to attain that key.

```
# The AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY, and AWS_SESSION_TOKEN
# environment variables now contain credentials corresponding to
# ExampleStudioComponentInitRole, so we can go ahead and
```

```
# fetch our secret.

# Install jq to parse output of secrets manager (if not included in ami)
sudo yum install -y jq

# Retrieve the example secret from Secrets Manager.
EXAMPLE_SECRET_KEY=$(
    aws secretsmanager get-secret-value \
        --region <REGION_OF_EXAMPLE_SECRET> \
        --secret-id example-secret \
        | jq '.SecretString | fromjson | .["example-secret-key"]' \
)

# Do stuff with $EXAMPLE_SECRET_KEY to provision the instance before the
# user has ever logged in. The user will only gain access to the secret
# if this script persists it to disk.
```

Windows system initialization scripts

Now that you've attached the `ExampleStudioComponentInitRole` to the studio component, the credentials corresponding to that role will automatically be available from your system initialization scripts. You can access these credentials through the `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, and `AWS_SESSION_TOKEN` environment variables. This means that you can use the AWS CLI or SDK without any manual credential management. If your studio component requires an API key to provision a machine with the functionality that it provides, use the following script to attain that key.

```
# The $Env:AWS_ACCESS_KEY_ID, $Env:AWS_SECRET_ACCESS_KEY, and $Env:AWS_SESSION_TOKEN
# environment variables now contain credentials corresponding to
# ExampleStudioComponentInitRole, so we can go ahead and
# fetch our secret.

# Retrieve the example secret from Secrets Manager.
$exampleSecretKey = (Get-SECSecretValue -Region us-east-1 -SecretId example-
secret).SecretString `

| ConvertFrom-Json `

| select -exp "example-secret-key" `

# Do stuff with $exampleSecretKey to provision the instance before the
# user has ever logged in. The user will only gain access to the secret
# if this script persists it to disk.
```

Use a studio component runtime role

This example shows how to interact with Amazon S3 from the user initialization script. This occurs while you're logged into the instance using the *studio component runtime role*. The runtime role is intended to give studio users access to tools that are configured by studio components. Credentials for the runtime role are made available on workstations by a studio component that's specific to the AWS profile. The environment variable provides the profile to the system and to the user initialization scripts.

To use this initialization script, first create a runtime IAM by following the [Creating a role for an AWS service \(console\)](#) tutorial in the *IAM User Guide*. This role will be able to access Amazon S3.

- Choose Nimble Studio as a **trusted entity**.
- On the **Select trusted entity** page, select **Custom trust policy**. Then enter the following policy.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "identity.nimble.amazonaws.com"  
            },  
            "Action": [  
                "sts:AssumeRole",  
                "sts:TagSession"  
            ]  
        }  
    ]  
}
```

- Choose the **AmazonS3ReadOnlyAccess AWS Managed policy**.
- For **Role name**, enter **ExampleStudioComponentRuntimeRole**.

Choose this role when creating a studio component. When you add Nimble Studio as a trusted entity, this role will display in the dropdown in the IAM roles section of the **Create Studio Component** page.

Linux user initialization scripts

The main intention of the runtime role is to enable tools configured by studio components to get the AWS access they need. However, it's possible to use the credentials directly. This is useful while developing a studio component or for AWS access needs without manual credential management.

In the script, the **AWS_PROFILE** environment variable contains the name of an AWS profile that has been configured in the shared credentials file. You can either configure the tools which the profile name they should use, or you can directly interact with the AWS CLI. The **AWS_PROFILE** environment variable tells the AWS CLI which profile to use.

```
# The AWS_PROFILE environment variable now contains
# the name of an AWS profile that has been configured in the shared
# credentials file. From here, we can either configure our tools
# which the profile name they should use, or we can directly interact
# with the AWS CLI (since AWS_PROFILE environment variable tells
# the AWS CLI which profile to use).

# Optionally configure the default region for the profile. If you
# do this, then you don't need to specify the region on a per-command
# basis using the `--region` option as is done in the rest of this
# example.
#
#     aws configure --profile $AWS_PROFILE set region us-west-2
#

# Example of configuring a tool.
echo $AWS_PROFILE > $HOME/my-tools-config.txt

# Example of using the AWS CLI directly. Write a list of s3 buckets
# to a text file in the user's home directory.
aws s3 ls --region us-west-2 > $HOME/buckets.txt

# Example of configuring your shell (bash in this case) to set the
# AWS_PROFILE environment variable on start.
# Take caution when doing this. If you do this in multiple studio
# components that are added to the same launch profile, then only
# one will end up taking affect.
if ! grep "export AWS_PROFILE" $HOME/.bashrc >> /dev/null 2>&1 ; then
    echo "export AWS_PROFILE=$AWS_PROFILE" >> $HOME/.bashrc
fi
```

To use these runtime role credentials on a Linux machine, follow these instructions.

To use runtime role credentials on a Linux machine

1. Open a terminal.
2. Run the following command to list the available Amazon S3 buckets in us-west-2: `aws s3 ls --region us-west-2`
 - This command works without explicitly specifying the `--profile` option because the user init script configured the shell to export `$AWS_PROFILE`s environment variable with the value of the studio component role name.

Windows user initialization scripts

The main intention of the runtime role is to enable tools configured by studio components to get the AWS access they need. However, it's possible to use the credentials directly. This is useful while developing a studio component or for AWS access needs without manual credential management.

In the script, the `$Env:AWS_PROFILE` environment variable contains the name of an AWS profile that has been configured in the shared credentials file. You can either configure the tools which the profile name they should use, or you can directly interact with the AWS powershell commandlets. AWS Tools for PowerShell is already configured to use this profile for the duration of this script.

```
# The $Env:AWS_PROFILE environment variable now contains
# the name of an AWS profile that has been configured in the shared
# credentials file. From here, we can either configure our tools
# which the profile name they should use, or we can directly interact
# with the AWS Powershell commandlets as the AWS Tools for Powershell
# has already been configured to use this profile for the duration of
# this script.

# Optionally configure the default region for the profile. If you
# do this, then you don't need to specify the region on a per-command
# basis using the `'-Region` option as is done in the rest of this
# example. There is no AWS Tools for Powershell equivalent
# of this command, so you need to ensure that you have installed
# the AWS CLI on your AMI for this to work.
#
# aws configure --profile $Env:AWS_PROFILE set region us-west-2
#
```

```
# Example of configuring a tool.  
$Env:AWS_PROFILE | Out-File "$Env:USERPROFILE\my-tools-config.txt"  
  
# Example of using the AWS CLI directly. Write a list of s3 buckets  
# to a text file in the user's home directory.  
Get-S3Bucket -Region us-west-2 | Out-File "$Env:USERPROFILE\buckets.txt"  
  
# Example of configuring your powershell use the AWS profile on start.  
# Take caution when doing this. If you do this in  
# multiple studio components that are added to the same launch profile,  
# then only one will end up taking affect. This only  
# configures powershell (and not command prompt).  
if (!(Test-Path $profile)) {  
    New-Item -Path $profile -ItemType file -Force  
}  
if (!(Select-String -Path $profile -Pattern "^\s*Set-AWSCredential")) {  
    echo "Set-AWSCredential -ProfileName $Env:AWS_PROFILE" `  
        | Add-Content -Path "$profile"  
}
```

The main intention of the runtime role is to enable tools that are configured by studio components to get the AWS access they need. However, it's possible to use the credentials directly. This is useful while developing a studio component or for ad-hoc AWS access needs without manual credential management. To use these runtime role credentials on a Windows machine, follow these instructions.

To use runtime role credentials on a Windows machine

1. Open PowerShell.

 **Note**

Windows takes some time to initialize after launch. If at first you are not in your expected user directory when you open a Powershell, close the Powershell, wait a few minutes, and try again.

2. Run the following command to list the available Amazon S3 buckets in us-west-2: Get-S3Bucket -Region us-west-2

- This command works without explicitly specifying the `-ProfileName` option because the user init script configured the shell to export `$AWS_PROFILE`s environment variable with the value of the studio component role name.

You now know how to create a custom configuration and attach it to a launch profile. You've also seen several examples of what custom configurations can do. To see another example of when you can use a custom configuration, see the [Provide Superuser access for Linux users](#) tutorial.

How to back up your studio data

In this administrator tutorial, you'll learn how to back up your studio's data by using one of the following options.

- The section [Back up option 1: Copy data to S3 bucket from a virtual workstation](#) requires you to [Launching a virtual workstation](#). This method might take several hours if you're backing up large volumes of data.
- [Back up option 2: Copy data to S3 bucket using AWS DataSync](#) is a great option if you have a lot of data and don't want to risk this process timing out during backup.
- [Back up option 3: Create Amazon FSx backups](#) is a quick method, but it requires that you create a new Amazon FSx file system and mount it to a virtual workstation.

Backups only cover local data and won't handle backing up shared storage.

To learn how to back up session data, see the [Session auto backup](#) tutorial.

Contents

- [Back up option 1: Copy data to S3 bucket from a virtual workstation](#)
- [Back up option 2: Copy data to S3 bucket using AWS DataSync](#)
- [Back up option 3: Create Amazon FSx backups](#)

Back up option 1: Copy data to S3 bucket from a virtual workstation

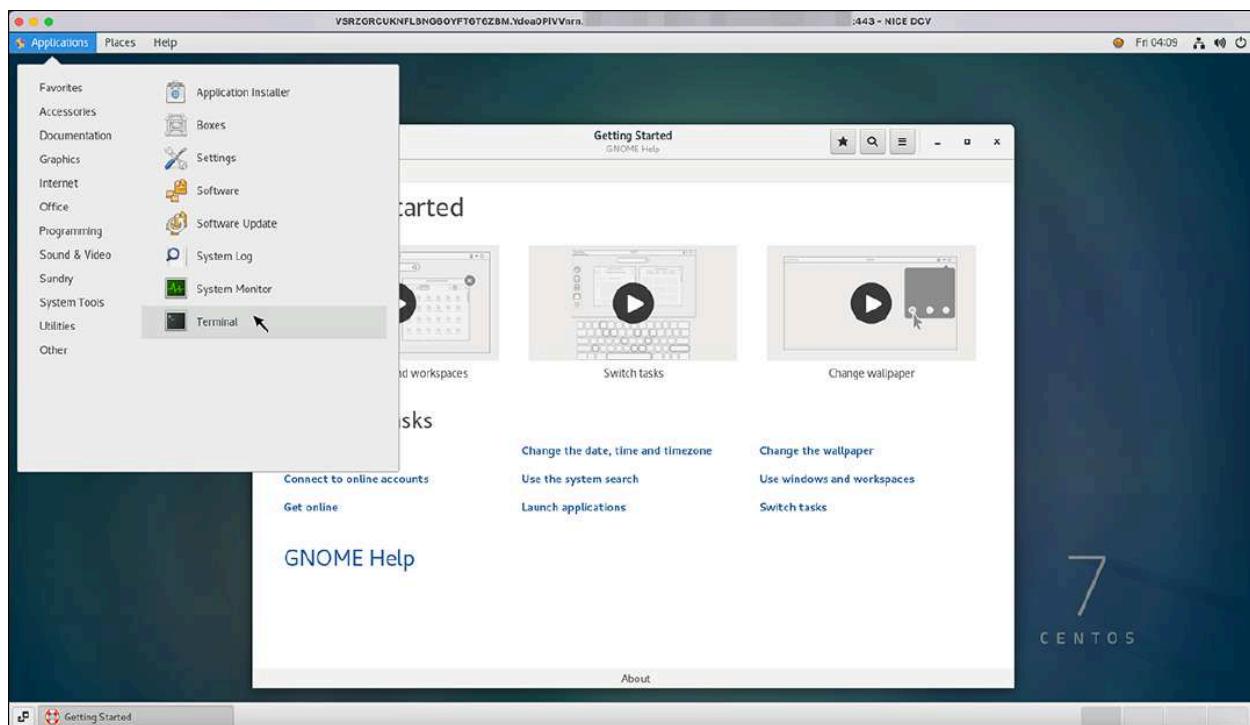
In this option, you will copy your studio data to an Amazon S3 bucket from a virtual workstation.

Create an Amazon S3 bucket

1. Sign in to the AWS Management Console and open the [Amazon S3](#) console.
2. Choose **Create bucket** and complete the following fields:
 - a. Enter a **Bucket name**.
 - We recommend naming it <your-studio-name>-<filesystem>-backup.
 - b. Set the **AWS Region** to the Region in which your cloud studio from Nimble Studio is deployed.
 - c. (Optional) Navigate to the **Tags** section and choose **Add tag**.
 - Enter **Studio** as the key and <your-studio-name> as the value.
 - d. Choose **Create bucket**.

Launch a virtual workstation

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio manager** in the left navigation pane.
3. On the **Studio manager** page, choose [Go to Nimble Studio portal](#).
4. Launch a virtual workstation by following the instructions in the [Launching a virtual workstation](#) tutorial.
5. After you're connected and logged in to your virtual workstation, follow the steps in [Getting Started with the AWS Tools for Windows PowerShell](#) to specify which credentials are needed to run the AWS Tools for PowerShell. For Linux, see [Configuration Basics](#).
 - a. If you're on a Windows workstation, open **PowerShell** from the start menu.
 - b. If you're on a Linux workstation, open a **Terminal**.
 - Select **Applications** in the menu bar. Then choose **System Tools** and **Terminal**.



6. Use the command `aws s3 sync <LocalPath> <S3 URI>` and press the enter (or return) key to sync data from your virtual workstation to Amazon S3.
 - a. In this case, the Amazon S3 URL is `s3://<bucket_name>`.
 - b. To see what files will get synced (before actually syncing them), you can add the flag `--dryrun` at the end of the command. This will show you a list of the files.
 - c. To sync your Z drive to your S3 bucket, refer to the following code examples for Linux and Windows.

Windows

```
aws s3 sync Z:\s3://<your-studio-name>-zdrive-backup
```

Linux

```
aws s3 sync /mnt/fsxshare/s3://<your-studio-name>-zdrive-backup
```

- d. Depending on how much data that you have in your studio, this process might take a while.
7. Wait until all of your files have been synced to Amazon S3 before closing the shell window and shutting down your virtual workstation.

Back up option 2: Copy data to S3 bucket using AWS DataSync

This option doesn't require you to launch a virtual workstation. Instead, you can run this option from the [AWS Management Console](#) to avoid the risk of timing out. While this option requires more steps to set up, it's the recommended option for backing up a lot of data when you can't use an Amazon FSx backup.

For backing up your data using Amazon FSx, see [Back up option 3: Create Amazon FSx backups](#).

Prerequisites

- Create an [Create an Amazon S3 bucket](#) by following the instructions in [Back up option 1: Copy data to S3 bucket from a virtual workstation](#).

How to back up data to an S3 bucket using DataSync

1. Sign in to the AWS Management Console and open the [DataSync](#) console.
2. By default, you'll land in the **Tasks** section of the DataSync console.
 - a. A list of your tasks will display. These tasks are configurations for data transfer and synchronization.
 - b. To back up data, choose **Create task** (upper-right button).
3. A page will open called **Configure source location**.

Configure your source location in DataSync

1. Specify source and destination location using the method that applies to you.
 - a. If you have never backed up data before using DataSync, choose **Create a new location**.
 - b. If you have created locations before with DataSync, opt for **Choose an existing location**.
2. In the **Location type** dropdown, choose the location type for the file system that contains the data that you're backing up.
 - In this example, we chose **Amazon FSx for Windows File Server**, because that is where the **Z** drive is. If you have additional Amazon FSx file systems in your studio, your drive letter might be different, such as **Y** or **X**, for example.
3. Choose the Region of the studio where you'll be backing up data.

4. Next, go to the **FSx file system** dropdown. The list of file systems that you have in that selected AWS Region will display.
 - a. The default Amazon FSx file system for the Z drive, which was created by StudioBuilder, will be named after your studio name.
 - b. You will also see file systems from the selected Region that you named yourself.
 - In this tutorial, we will choose the default Amazon FSx file system.
5. The following section is where you enter your **Share name**.
 - a. If you kept the default share name from our **File storage configuration** section of the [Nimble Studio console](#), then use `/share/`.
 - b. If you changed the default share name, use the name that you created.
6. Next, open the **Additional settings** using the triangle icon to open the **Security groups** section.
7. Use the dropdown in the security groups field to find the security groups associated with your account in that Region.

Find the security group for your file system

1. To find the correct security group associated with this file system, open a new tab so that you can return to this page easily, later.
2. In your new tab or window, open the [AWS Management Console](#), and then open the [Amazon FSx](#) console.
3. After the Amazon FSx page opens, choose **File systems** from the left navigation pane.
4. In the **File systems** section of that page, choose the **File system name** of the file system that you want to back up. You'll want to keep this page open to return to later.
5. A new page will open for that file system. Check that the **Network & security** tab is selected, then navigate to the **Subnet** section.
6. Choose the blue link in **Network interface**, which is the ID of your network interface.
7. Your network interface link will open a new tab in the **EC2 Dashboard** called **Network interfaces**.
8. Select the **Network interface** that matches the ID from the previous step.
9. Choose the **Details** tab.

10. Find the **Security group** IDs in the top right of the **Details** section.
 - Look at the last four digits of an ID, so you can compare that with the ID on the DataSync page.
11. Go back to the DataSync page that you kept open.
 - a. In the **Configuration** section, open the **Security groups** dropdown, and navigate through the IDs until you find the last four digits that match the ID that you found in **Network interfaces**.
 - Select the matching ID.
 - b. Remove the **default** security group by choosing the **X** to the right of its name.

 **Note**

For your security, no external services can access or copy your data. This can sometimes result in your security group not having access, and thus failing. To mitigate this, see [Add a temporary rule to the security group](#) in the following section.

Add a temporary rule to the security group

This security rule permits access to the Amazon FSx file system for the purpose of completing these steps only. **For security reasons, after you add this temporary rule, complete the steps in [Remove temporary rule from security group](#)**

1. Go back to the **Networks interfaces** tab and choose the security group that you selected previously to go to its **Details** page.
2. Choose **Edit inbound rules**.
3. On the **Edit inbound rules** page, navigate to the bottom and choose **Add rule**.
 - a. A new row will appear on the page.
 - b. Open the dropdown list called **Custom TCP**.
4. For DataSync to work, it needs access to **SMB**, so select that.
 - a. The row will automatically fill with Transmission Control Protocol (TCP), and port 445.
 - b. You now have two more columns to fill out: **Source** and **Description**.

5. Fill out the **Source** section next, so DataSync can use your security group to access the Amazon FSx file system.
 - a. Select the **Search** field for the **Source** column.
 - b. The **Search** field has a list of security groups.
 - c. Navigate through the list until you find your Amazon FSx security group ID, and select it.
6. Write something in **Description** that tells you what it's so that you can remove it after you complete these steps. For example, Remove security rule after completing option 2 backup!.
 - This security rule permits access to the Amazon FSx file system for the purpose of completing these steps only. Remove this rule after you complete the steps in this section. To remove this security group see [Remove temporary rule from security group](#)
7. Choose **Save rules**.
8. The details page will appear showing the new rule you made, confirming that it was saved.
 - To find the new rule that you made, you can search for the description that you just wrote for it.

Finish configuring DataSync source location

1. Go back to the DataSync page to **User settings**.
2. In the **User** field, enter **Admin**.
 - Amazon FSx is connected to your AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD), and the administrator is the role that has access to both.
3. For **Password**, use the AWS Managed Microsoft AD password that you used when you set up your studio.
4. In the **Domain** field, use the domain for your AWS Managed Microsoft AD.
 - a. To find the AWS Managed Microsoft AD domain, open the [AWS Management Console](#), and then open the [Amazon FSx](#) console.
 - b. Select **File systems**.
 - c. If you kept that tab open, find the page showing your file system ID.
5. Go to the **Network & security** tab and find the section called **Network & security**.
6. Look for your **AWS Managed Microsoft AD ID** link and choose it.

7. When the **Directories** page opens, notice the **Directory DNS name**.
8. Go back to DataSync and paste the Directory DNS name into the **Domain** field in **User settings**.
9. Choose **Next**.
10. The page titled **Configure destination location** will open.

Configure destination location in DataSync

1. After the **Configure destination** location opens, go the **Configuration** section, and choose **Amazon S3** for the **Location type**.
2. Confirm that the Region is the same one that you chose for the **Configure location** step.
3. Using the dropdown of the **S3 bucket** field, select the name of the S3 bucket that you created at the beginning of the Option 2 instructions.
4. For **S3 storage class**, select **Standard**.
5. For the **Folder** field, you can leave it empty, because you won't have any folders in this bucket, yet.
6. To create the **IAM role**, choose **Autogenerate**.
7. Within seconds, an IAM role will appear in the field for you to use.
 - Now that you have an IAM role, if you were to run the backup again for this S3 bucket, you can select this role from the dropdown menu in the **IAM role** field.
8. Choose **Next** to continue to the next step.

Configure Settings in DataSync

1. For **Task Name**, create a specific name that helps you find and identify the source and destinations for your backup. For example: Z Drive FSx to S3 <your-studio-name>-zdrive-backup.
2. For the task execution configuration, you can select **Verify only the data transferred**.
3. For **Set Bandwidth limit**, choose one of the following options based on your use case:
 - a. Choose **Use available** to maximize the bandwidth available for creating your backup.
 - i. We recommend this option if you're backing up data before deleting your studio, or you know that no users will be accessing the file system during the backup.

- ii. If you have users who might still be accessing the file system, choosing this option might negatively affect the throughput they get when accessing it.
- b. Choose **Set bandwidth limit** to limit the amount of bandwidth used for your backup.
- i. Use this setting if you have users who will be accessing the file system during the backup.
 - ii. You can choose a number (in MiB/s), up to the throughput capacity of the file system that you're backing up.
 - iii. Find the throughput capacity.
 - A. Choose **Services**. Then choose **FSx** and **File systems**.
 - B. Look in the **Summary** section.
4. In the **Data transfer configuration** setting, choose **Transfer only data that has changed**.
- Choose **Keep deleted files** and **Overwrite files**.
5. (Optional) In the **Filtering configuration** section, you can enter a pattern to use as a filter that will exclude specific task transfers. For example, you can exclude backup folders or a trash folder.
- a. To find those folders, go to **Amazon S3**. Choose **Buckets** and choose **Objects**. Look through your stored entities there.
 - b. Create an **Exclude pattern** in the **Filtering configuration** section. Using the trash and backup folders as an example, you would enter `/.Trash*` and `/backup*`.

Filtering configuration

When no filters are specified, the entire contents of the source location are transferred.

Exclude patterns
Files, folders, and objects with the specified patterns are excluded from the transfer. The pattern path is relative to the source location path. For example, `/my-folder` is a folder directly under the task's source location. Specifying `/my-folder` excludes the folder itself and all of its contents).

[Learn more and see syntax examples](#)

<code>/.Trash*</code>	Remove pattern
<code>/backup*</code>	Remove pattern

[Add pattern](#)

- c. For detailed information about filtering and syntax for creating patterns, see [Filtering the data transferred by AWS DataSync](#).
6. In the **Task logging** section, if the CloudWatch log group dropdown is blank, choose **Autogenerate** to generate a log group.
7. Choose **Next**.

Create the task and start the backup

1. Review the information that you entered for the source, destination, and task settings.
2. Choose **Create task**.
3. Choose **Start**. Then choose **Start with defaults** to start the backup.
 - The **Task status** will change from **Available** to **Running**.
4. Choose **Running** to go to the execution overview page to monitor the progress of your backup.
 - a. The **Execution status** will change from **Launching** to **Running**.
 - b. After the status has changed to **Success**, your backup is complete.
5. If your execution fails and the status changes to **Error**, choose **View logs in CloudWatch** at the top right of the execution overview page to review the error logs.
 - For more information about using CloudWatch logs, see [Working with Log Groups and Log Streams](#) in the [Amazon CloudWatch Logs User Guide](#).

Remove temporary rule from security group

After your backup is complete, remove the temporary rule that you added to your security group earlier.

1. Return to the security group for your Amazon FSx file system.
2. Choose the **Inbound rule** tab and then choose **Edit inbound rules**.
3. Find the temporary inbound rule that you added by using the description that you gave it earlier.
 - a. The rule will be of type **SMB** and the port range will be **445**.
 - b. Choose **Delete** to the right of the rule.

- c. Choose **Save rules** to save your changes.

Back up option 3: Create Amazon FSx backups

You won't be able to access the backed up data until you use the backup to create a new Amazon FSx file system, launch a workstation, and connect to it.

1. Sign in to the AWS Management Console and open the [Amazon FSx](#) console.
2. Select the file system that you want to create the backup for.
3. Enter a **Backup name** and choose **Create backup**.
4. Wait until you get a confirmation that your backup has successfully been created. This can take a few minutes.
5. Choose **View backups** in the banner at the top of the menu bar to see your backup.

Your backup will display at the top of the list. Use your backup to create a new Amazon FSx file system and access your data.

For more information about how to use your backup, see [Working with user-initiated backups](#) in the [Amazon FSx for Windows File Server User Guide](#).

Update to latest StudioBuilder version

This administrator tutorial will show you how to update your studio to the latest version of StudioBuilder.

Contents

- [Prerequisites](#)
- [Step 1: Launch new instance with the latest StudioBuilder version](#)
- [Step 2: Check updates to your launch profiles](#)
- [Step 3: Update your studio](#)
- [Step 4: Compare launch profiles after update is complete](#)
- [Troubleshooting](#)
- [Related resources](#)

Prerequisites

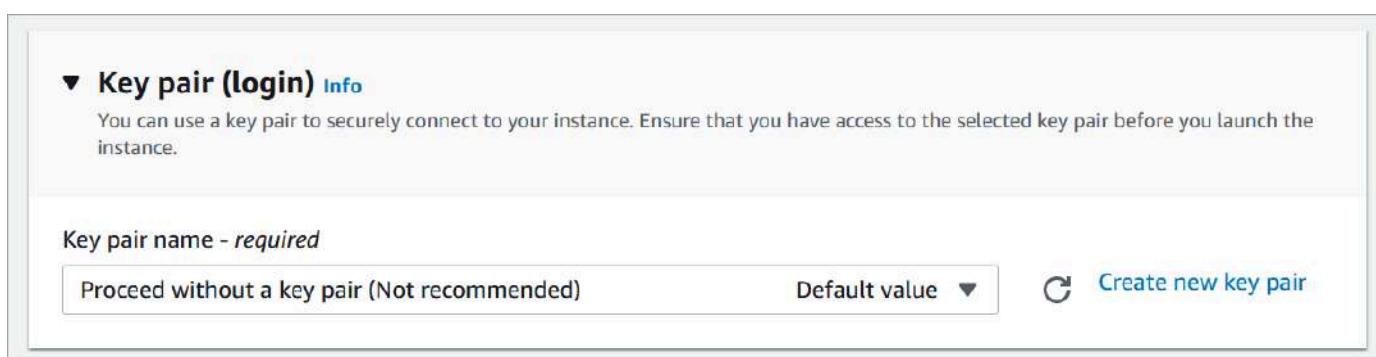
- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.

Step 1: Launch new instance with the latest StudioBuilder version

The first step to updating your studio to the latest release is to launch a new instance of StudioBuilder from AWS Marketplace.

Follow the instructions in the [Launch an instance using defined parameters](#) tutorial in the *Amazon EC2 User Guide for Windows Instances* while using the following information.

- Sign in to the AWS Management Console and open the [AWS Marketplace](#) console.
- Select Manage subscriptions.
- Find **Nimble Studio StudioBuilder** and choose **Launch new instance**.
 - When the **Launch new instance** page opens, the latest version of StudioBuilder will display.
- In the **Region** section, select the Region where you want to deploy this version.
- Choose **Continue to launch through EC2**.
- For **Instance Type**, select **t3.medium** from the list.
- For **Key pair (login)** choose **Proceed without a key pair** from the first dropdown.



- A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance. You will use EC2 Instance Connect so you don't need a key pair.
- On **Network settings**, choose **Edit**.

9. Set the **VPC** to the default VPC.
10. Set **Auto-assign Public IP** to **Enable** so that your instance receives a public IP address that you will use when connecting to it later.
11. In **Advanced details**, specify an AWS Identity and Access Management (IAM) role by selecting the role that you created during deployment. For example, in the [Step 3: Launch the StudioBuilder EC2 instance](#) tutorial, you used **StudioBuilder_Instance_Admin_Role**.
12. Choose **Launch**.
13. On the **Launch Status** page, choose **View Instances**.
14. When the **Instances** page opens, the **Status check** column will show **Initializing**. After the process is complete, the status will change to **2/2 checks passed**.
15. While you're waiting, change the name of the instance by selecting the edit icon (which looks like a small square).
 - Enter the name and include the new version number, then choose **Save**.
16. After the **Status check** column changes to **2/2 checks passed**, you can choose **Connect**.

Step 2: Check updates to your launch profiles

Before you complete the update, it's important to keep track of the modifications that you've made to the **RenderWorker-Default** and **Workstation-Default** launch profiles.

The new version will automatically replace the **RenderWorker-Default** and **Workstation-Default**, so those names will remain the same. Therefore, in this section, we'll create a copy of the old version and rename that, so we can compare the changes.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select **RenderWorker-Default**.
4. Choose **Actions**. Then choose **Copy to new**.
5. After the **Create launch profile** opens, change the name in the **Launch profile name** input field to **TEMPORARY_RENDER_WORKER**.
6. Now, you'll do the same for the other launch profile:
 - a. Go back to the **Launch profiles** page and select **Workstation-Default**.
 - b. Choose **Actions**. Then choose **Copy to new**.

- c. After the **Create launch profile** opens, change the **Launch profile name** to TEMPORARY WORKSTATION DEFAULT.
7. If you created your studio with additional storage, added it to your custom launch profiles, and now want to delete that storage, you need to delete the custom launch profiles. This is because StudioBuilder can't delete storage if your launch profiles reference it.
 - a. Select the custom launch profile that is connected to the storage that you want to delete.
 - b. Choose **Action**. Then choose **Edit**.
 - c. Uncheck the box next to the storage you want to delete in the **Launch profile components** section.
 - d. Select **Update launch profile**.
8. Go back to the **Instances** page and select the refresh icon in the upper-right corner.
9. After the page refreshes, the StudioBuilder instance should show **2/2 checks passed** in the **Status check**. If not, wait a few minutes and refresh again.
10. Choose **Connect** at the top of the menu bar next to the refresh icon.
11. When **Connect to instance** opens, select the **EC2 Instance Connect** tab.
12. In the **User name** field, enter **ec2-user**, then choose **Connect**.
13. Your StudioBuilder Command Line Interface (CLI) will open.
14. Use the arrow keys on your keyboard to select **Update** and/or edit your studio. Then press the enter (or return) key.
15. A question at the bottom of the interface will ask if you would like to edit the configuration. Type in no and press the enter (or return) key.
16. You will be prompted to type **UPDATE MY STUDIO**. After typing that, press the enter (or return) key, and then continue to the following section, [Step 3: Update your studio](#).

Step 3: Update your studio

After entering **UPDATE MY STUDIO**, you will receive a series of prompts depending on the status of your render farm. You will also receive questions about your storage. Follow the steps that apply to you.

If you don't have a render farm for your studio

Follow these steps when the CLI prompt says: If you don't have a render farm in your studio, you can create one now. Render farm costs can vary. For pricing estimates, see <https://aws.amazon.com/nimble-studio/pricing>.

- Use the arrow keys to choose if you would like to create a render farm or not, and then press the enter (or return) key.
 - a. If you choose Yes, I want to create a render farm for my studio.
 - Follow the prompts for configuring your render farm. For more information about the CLI questions, see [Deploying a new studio with StudioBuilder](#).
 - b. If you choose No, I don't want to create a render farm at this time.
 - You will skip all render farm questions and prompts. You can come back to create a render farm later.

If you already have a render farm for your studio

Important

Before deleting a render farm, follow *steps one through four* of the [How to delete a render farm built by StudioBuilder](#) tutorial.

Follow these steps if the CLI prompt asks: Would you like to delete the farm that StudioBuilder created for you?

- Use the arrow keys to choose if you would like to delete your render farm or not, and then press the enter (or return) key.
 - a. If you choose No I want to keep my existing farm:
 - The CLI will continue updating your studio without deleting your render farm.
 - b. If you choose Yes I want to delete my render farm:
 - i. Warning: Before deleting your farm, complete [Step 1: Remove compute farm studio component](#) and [Step 2: Update to latest StudioBuilder](#).

- ii. Follow the prompts to delete your render farm.

Storage questions

Follow these steps when the CLI prompt asks: Do you want to keep, modify or delete storage: FSxLustre1?

- Use the arrow keys to choose if you would like to keep, modify, or delete your storage, and then press the enter (or return) key.
 - a. If you choose Keep, StudioBuilder will finish updating with no further prompts.
 - If you choose Modify, follow the prompts for modifying your storage. For more information about the CLI questions, see [Deploying a new studio with StudioBuilder](#).
 - b. If you choose Delete, you will receive the following prompt: Deleting storage will PERMANENTLY DELETE all data stored on it.
 - i. To delete your storage, type DELETE <name of your storage> and press the enter (or return) key.
 - ii. If you don't want to delete your storage, type QUIT and press the enter (or return) key. This will exit StudioBuilder and your studio won't be updated.

As the CLI finishes the update, you won't need to do anything. It might take approximately 20 minutes.

Step 4: Compare launch profiles after update is complete

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Right-click **RenderWorker-Default** and select **Open Link in New tab**.
4. Next, right-click **TEMPORARY RENDER WORKER** and select **Open Link in New tab**.
5. Now, compare the temporary launch profiles with new updated versions so that you can make manual adjustments.
 - a. Select **RenderWorker-Default**,
 - b. Choose **Action**. Then choose **Edit**.

- c. In the **Edit launch profile** page, confirm that your **Amazon Machine Images (AMI)** and **Launch profile components** appear to be accurate.
 - d. Next, choose **TEMPORARY RENDER WORKER**.
 - e. Choose **Action**. Then choose **Edit**.
 - f. Compare the temporary (older) **Launch profile details** page with your new version. See if any components or AMIs are missing from the new version. Example: **prod** shows in the temporary launch profile, but not in the updated StudioBuilder launch profile. Therefore, in the next step, we will add **prod** to the updated version.
6. Go to the **RenderWorker-Default** edit page that you left open. This is where you can add the missing components.
 7. Next, choose **Update launch profile**.
 8. In the **Launch profiles** list, select **TEMPORARY RENDER WORKER**.
 9. Choose **Action**. Then choose **Delete**.
 10. A window will open asking you to confirm deletion. Type **Delete** in the field, and then choose **Delete**.
 - Tip: Instead of typing, try selecting the **Delete** word and dragging it into the input field. After that, choose **Delete** to finish.

Now, you'll repeat these steps for the **Workstation-Default** and **TEMPORARY WORKSTATION DEFAULT** launch profiles.

1. Right-click **Workstation-Default** and select **Open Link in New tab**.
2. Right-click **TEMPORARY WORKSTATION DEFAULT** and select **Open Link in New tab**.
3. Select **Workstation-Default**.
4. Choose **Action**. Then choose **Edit**.
5. In the **Edit launch profile** page, confirm that your **Amazon Machine Image** and **Launch profile components** are accurate.
6. Next, choose the **TEMPORARY WORKSTATION DEFAULT**.
7. Choose **Action**. Then choose **Edit**.
8. Confirm that the components and AMIs are the same for both versions of the profile. Add anything from the older temporary version into the new version.
 - If the new version upgrades your instance types, keep the upgrade.

9. If you need to make any edits, you can save the changes by choosing **Update launch profile**.
10. If no edits were made, just close the tabs to these pages.
11. Return to the **Launch profiles** page and select **TEMPORARY WORKSTATION DEFAULT**.
12. Choose **Action**. Then choose **Delete**.
13. A window will open asking you to confirm deletion. Drag or type Delete in the field, and then choose **Delete**.

You've now updated your studio to a new version of StudioBuilder. When newer versions are released, you can return to this tutorial to install those as well.

Troubleshooting

The ConfigureSpotEventPlugin failed to update in the Compute stack.

If the ConfigureSpotEventPlugin fails to update in the Compute stack, it might be because you have active Spot Fleet Requests (SFR) in your account. To see if you have active SFRs, and to remove them, follow these steps.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Spot Requests** in the left navigation pane.
3. See if you have active SFRs.
4. If you have SFRs, go to **Services** and then choose CloudFormation.
5. Manually delete the DeadlineResourceTracker that was created by the Spot Event Plugin when your studio was created.
6. Retry updating your studio.

Unable to satisfy !00% MinSuccessfulInstancesPercent requirement

`UPDATE_FAILED | AWS::AutoScaling::AutoScalingGroup`

If the Deadline Resource Tracker stack is in CloudFormation, delete it and run the StudioBuilder update again.

Related resources

[Deploying a new studio with StudioBuilder](#)

Change administrator password in AWS Directory Service

This tutorial is for admins who need to change their password or a policy for their [AWS Directory Service for Microsoft Active Directory](#).

- To change the administrator password for your AWS Directory Service account, see [Reset an administrator password for AWS Directory Service](#) in this tutorial.
- To modify an AWS Directory Service account policy, see [Using password policies for AWS Directory Service](#) in this tutorial.

To reset a user password for your AWS Directory Service, see [Reset a user password](#) in the AWS Directory Service Administration Guide.

Contents

- [Prerequisites](#)
- [Reset an administrator password for AWS Directory Service](#)
- [Using password policies for AWS Directory Service](#)
- [Supported policy settings for AWS Managed Microsoft AD](#)

Prerequisites

- You need an AWS account with administrator permissions to use the [AWS Management Console](#).
- You need to have an account for AWS Directory Service for Microsoft Active Directory. StudioBuilder automatically creates that account for you. You can find that account by following [Step 1: Update administrator password in AWS Directory Service](#).

Reset an administrator password for AWS Directory Service

Resetting your administrator password involves two steps. Follow both steps in this section to reset your password.

Step 1: Update administrator password in AWS Directory Service

1. Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.

2. When the **Directories** page opens, select the directory associated with the administrator account that you want to modify.
3. Choose **Actions**. Then choose **Reset user password**.
 - a. When a new page opens, go to the **Username** field and enter **Admin**.
 - b. In the **New password** field, enter a new password. Follow these requirements:
 - i. The password must be eight or more characters long.
 - ii. The password must contain characters from three of the following categories: Uppercase letters A-Z, lowercase letters a-z, numbers 0-9, and special characters, such as \$ # %
 - iii. Remember this password, because you will need it next, in [Step 2: Update your Secret in Secrets Managers](#).
 - c. After you've confirmed the new password, choose **Reset password** and the **Directories** page will open.

 **Note**

If you're using StudioBuilder version 1.1.3 or later, you can skip [Step 2: Update your Secret in Secrets Managers](#), and go directly to [Using password policies for AWS Directory Service](#).

Step 2: Update your Secret in Secrets Managers

 **Important**

If you're using a StudioBuilder version prior to version 1.1.3, complete the following **step 2**, or your password reset will fail.

1. Sign in to the AWS Management Console and open the [Secrets Manager](#) console.
2. Search for **StudioBuilder-SecretForAD** and select it when it appears in the **Secret name** list.
3. On the **Secrets** page, navigate to the **Secret value** section and choose **Retrieve secret value**.
4. Choose **Edit** to open the **Edit secret value** module.
5. In the **Edit secret value** module, remove your old password.

6. Type in the new password that you created earlier, in [Step 1: Update administrator password in AWS Directory Service](#).
7. Choose **Save**.
8. The **StudioBuilder-SecretForAD** page will open, and your newly saved password will display in the **Secret value** section of that page.

This completes the process of resetting your AWS Directory Service password. To modify policies for AWS Directory Service, continue to the following section of this page.

Using password policies for AWS Directory Service

If you don't configure any password policies in your AWS Directory Service for Microsoft Active Directory directory, AWS Managed Microsoft AD uses the default domain group policy. This policy includes the following settings:

Policy	Setting
Enforce password history	24 passwords remembered
Maximum password age	42 days
Minimum password age	1 day
Minimum password length	7 characters
Password must meet complexity requirements	Activated
Store passwords using reversible encryption	Deactivated

 **Note**

The 42-day maximum password age includes the administrator password.

Sometimes admins want to modify a policy setting to allow for different levels of user access.

For example, senior managers who regularly access confidential information might need strict settings. On the other hand, users that need access to low sensitivity information might need less

strict policies. One such policy change might include increasing the maximum password age, so that it doesn't need resetting at 42 days.

Step 3: Modify password policies in AWS Managed Microsoft AD

User accounts that are a member of the **AWS Delegated Fine Grained Password Policy**

Administrators security group can use the following procedure to assign policies to users and security groups.

To assign password policies to your users:

1. Launch [Active Directory administrative center \(ADAC\)](#) from any managed EC2 instance that you joined to your AWS Managed Microsoft AD domain.
2. Switch to the **Tree View** and navigate to **System\Password Settings Container**.
3. Double-click on the fine-grained policy that you want to edit.
4. Select **Add** to edit the policy properties, and add users or security groups to the policy.
 - For more information about the default fine-grained policies provided with AWS Managed Microsoft AD, see [AWS pre-defined password policies](#).
5. To verify that the password policy has been applied, run the following PowerShell command:
`Get-ADUserResultantPasswordPolicy -Identity 'username'`

Supported policy settings for AWS Managed Microsoft AD

AWS Managed Microsoft AD includes five fine-grained policies with a non-editable precedence value. The policies have a number of properties that you can configure to enforce the strength of passwords, and account lockout actions in the event of login failures.

You can assign the policies to zero or more AWS Managed Microsoft AD groups. If an end user is a member of multiple groups and receives more than one password policy, AWS Managed Microsoft AD enforces the policy with the lowest precedence value.

Important

We don't recommend that you set the maximum password age to -1. This presents a security risk.

Password policy properties

You can edit the following properties in your password policies to conform to the compliance standards that meet your business needs.

- [Enforce password history](#)
- [Minimum password length](#)
- [Minimum password age](#)
- [Maximum password age](#)
- [Store passwords using reversible encryption](#)
- [Password must meet complexity requirements](#)

You can't modify the precedence values for these policies.

For more details about how these settings affect password enforcement, see [AD DS: Fine-grained password policies](#) on the **Microsoft TechNet** website. For general information about these policies, see [Password policy](#) on the **Microsoft TechNet** website.

Account lockout policies

You can also modify the following properties of your password policies to specify if, and how, the AWS Managed Microsoft AD should lock out users from an account after login failures:

- Number of failed logon attempts allowed
- Account lockout duration
- Reset failed logon attempts after some duration

For general information about these policies, see [Account lockout policy](#) on the **Microsoft TechNet** website.

For more information about modifying password age, length, or other requirements, see [Supported policy settings](#) in the **AWS Directory Service** Administration Guide.

Rotating certificates created in Nimble Studio

Amazon Nimble Studio uses X.509 certificates to make and validate identity claims between the endpoints of a Nimble Studio communication channel. The X.509 certificate defines a specific

format of digital certificate that can be used for identification and message encryption. For more information about certificates, see the [RFDK documentation](#).

These self-signed certificates are required by Deadline. By default, they have a three year validity period and will expire after that time. To mitigate the availability risk posed by these certificates expiring, you need to manually rotate the certificates.

Important

You only need to rotate your certificates three years after your studio was created. At the earliest, you need to rotate your certificates by April 28, 2024.

Contents

- [Prerequisites](#)
- [Rotate the root CA](#)
- [Rotate the render queue certificate](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- Remove any custom configurations for your studio. If you don't, you might run into errors when you try to rotate your certificates.

Rotate the root CA

The root certificate authority (CA) is a self-signed certificate that is used to sign all other all X.509 certificates used in Render Farm Deployment Kit (RFDK), including the Deadline Client certificate, render queue certificate, and MongoDB. Rotating this certificate causes the Application Load Balancers (ALB's) certificate to no longer validate with respect to the old root CA. This means that any connection to the render queue is disrupted. To rotate it, you need to tear down the worker fleets and service tier.

Step 1: Delete the dependent CloudFormation stacks

You need to delete the Compute and Service stacks. Both stacks have termination protection turned on by default. Turn off termination protection so that you can delete them.

Disable termination protection for your RenderQueue Application Load Balancer (ALB) for the service tier rotation. If you don't, deleting the CloudFormation stack might fail. Once you've deleted the stack, you can add deletion protection back to the ALB.

1. Sign in to the AWS Management Console and open the [AWS CloudFormation](#) console.
2. Select the <your-studio-name>Compute stack. The stack must be currently deployed.
3. In the stack details pane, select **Stack actions** and then select **Edit termination protection**. CloudFormation displays the **Edit termination protection** dialog box.
4. Choose **Disable**, and then select **Save**.
5. Select **Save**. In the stack details pane, choose **Delete**.
6. Select **Delete stack** when prompted.
7. After the status of the service stack changes to Delete, go to **Services** and then select [EC2](#).
8. On the navigation pane, choose **Load Balancers**.
9. Select the load balancer that has Render in its name.
10. On the **Description** tab, choose **Edit attributes**.
11. On the **Edit load balancer attributes** page, clear **Enable for Delete Protection**, and then choose **Save**.
12. Choose **Save**.
13. Go to **Services**. Then choose **CloudFormation**.
14. Select the <your-studio-name> Service stack. The stack must be currently running.
15. In the stack details pane, select **Stack actions**, then choose **Edit termination protection**. CloudFormation displays the **Edit termination protection** dialog box.
16. Choose **Disable**, and then select **Save**.
17. Select **Save**. In the stack details pane, choose **Delete**.
18. Select **Delete stack** when prompted.
 - If there are new launch profiles in the studio, the deletion will fail.

Step 2: Give Lambda permissions to create a secret

Follow the instructions in the [To embed an inline policy for a user or role \(console\)](#) section of the Adding IAM identity permissions (console) tutorial in the IAM User Guide. Use the following information while you follow the tutorial.

1. Choose the IAM role with the name that starts with <your-studio-name>Data-RootCAGeneratorServiceRole.
2. Also for permissions policies, search for Secret and select the check box for **SecretManagerReadWrite**.

 **Important**

Remove this policy from the role before deleting the studio. If you don't, the studio deletion will fail on the Data stack.

Step 3: Change the RootCAPassphrase secret value

To protect the certificate, change the value of the RootCAPassphrase secret.

1. Sign in to the AWS Management Console and open the [Secrets Manager](#) console.
2. Select the secret with the name beginning with RootCAPassphrase.
3. On the secret details page, choose **Retrieve secret value** in the **Secret value section** to update the secret value.
4. Choose **Edit** and enter a new secret value.
5. Select **Save**.
6. Notice the ARN of the secret in the **Secret details** section. This will be used to create the Lambda function in [Step 4: Invoke the Lambda function](#).
 - Example: `arn:aws:secretsmanager:region:user-id:secret:RootCAPassphraseuniqueId`

Step 4: Invoke the Lambda function

After the policy is attached to Lambda, you can invoke the Lambda function to create a new root CA. You can invoke the Lambda function one of three ways: through the console, by using [AWS CloudShell](#), or by using the AWS CLI.

The following JSON is the Lambda function that you will run. Replace *ARN of the secret containing the password* with the ARN of the secret that you found in step 6 of [Step 3: Change the RootCAPassphrase secret value](#).

```
{  
    "RequestType": "Create",  
    "ResourceProperties": {  
        "DistinguishedName": {  
            "CN": "your-studio-nameRootCA",  
            "O": "AWS",  
            "OU": "Thinkbox"  
        },  
        "Passphrase": "arn:aws:secretsmanager:us-west-2:675872700355:secret:RootCAPassphraseC6002E3F-0R3shg9dBi38-srKIKQ",  
        "Secret": {  
            "NamePrefix": "your-studio-nameData/RootCA",  
            "Description": "your-studio-nameData/RootCA",  
            "Tags": []  
        },  
        "CertificateValidFor": "1095"  
    }  
}
```

Invoke the Lambda function using the console

1. Go to **Services** and then select **Lambda**.
2. Select the Lambda function named *your-studio-name*Data-RootCAGenerator.
3. Select **Actions** and then select **Test**.
4. In the **Test event** section, choose **New event**.
5. In **Template**, enter the JSON template that is provided previously.
6. Enter a **Name** for this test.
7. Choose **Save changes**, and then choose **Test**. Each user can create up to 10 test events per function. Those test events are not available to other users.

- a. Lambda runs your function on your behalf. The function handler receives and then processes the sample event.
- b. If, in the **Log output**, you see "Status": "SUCCESS", the Lambda function was successfully invoked.

There might be Failures in the console when running this script. The new secret and certifications will still be created and will be available in Secrets Manager. You can find the new certifications by modifying the available columns to include create date and finding the newly created certs based on that.

Invoke the Lambda function using CloudShell

Note

CloudShell isn't supported in the eu-west-2 or ca-central-1 Regions. If you're in either of those Regions, use the AWS CLI instead.

1. Sign in to the AWS Management Console and open the [Lambda](#) console.
2. Select the Lambda function named <your-studio-name>Data-RootCAGenerator.
 - Notice the name of this Lambda function. You will need it for step 6.
3. Sign in to the AWS Management Console.
4. Go to **Services** and then select **CloudShell**.



5. Wait for the CloudShell session to load.
6. Run the following command: `aws lambda invoke --cli-binary-format raw-in-base64-out --function-name Lambda function name --payload `Template from previous example` response.json`

- a. Replace *Lambda function name* with the full name of the Lambda function that you found in step 6a.
- b. You can either set the template as blob in json format, or you can save the template as a local file and then run the command using the format --payload file://template.json.
- c. You will get an Unhandled error when you run this command, but it will still create the root CA.

Invoke the Lambda function using the AWS CLI

Note

If you're in the eu-west-2 or ca-central-1 regions, use the AWS CLI to invoke the Lambda function. The following steps instruct you about how to install and configure the AWS CLI.

1. To install or upgrade the AWS CLI on your local machine, follow the instructions in [Installing the AWS Command Line Interface version 2](#) in the *AWS Command Line Interface User Guide*.
2. Configure the AWS CLI by following the instructions in [Setting up new configuration and credentials](#).
3. Verify the installation or upgrade by running `aws nimble help`. This command displays a list of available Nimble Studio commands.
4. Sign in to the AWS Management Console and open the [Lambda](#) console.
5. Select the Lambda function named <your-studio-name>Data-RootCAGenerator.
 - Notice the name of this Lambda function. You will need it for step 7.
6. Run the following command:
`aws lambda invoke --cli-binary-format raw-in-base64-out --function-name Lambda function name --payload `Template from previous example` response.json`
 - a. Replace *Lambda function name* with the full name of the Lambda function that you found in step 3a.

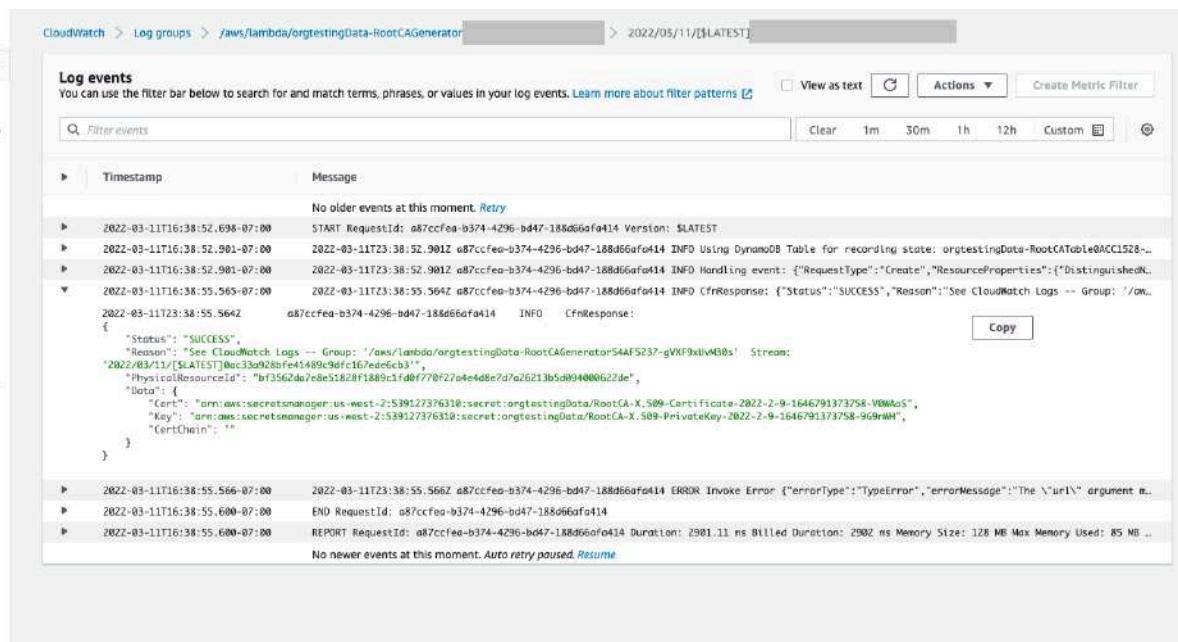
- b. You can either set the template as blob in json format, or you can save the template as a local file and then run the command using the format `--payload file://template.json`.
- c. You will get an Unhandled error when you run this command, but it will still create the root CA.

Verify that you invoked the Lambda function

Note

To verify that this step was successful, check the CloudWatch logs.

1. Open the [CloudWatch console](#).
2. In the navigation pane, choose **Log groups**.
3. For **Log Groups**, choose latest logs in the Data-RootCAGenerator log group.
4. Verify that the CloudFormation response was successful.



Timestamp	Message
2022-03-11T16:38:52.698-07:00	START RequestId: a87ccfca-0374-4296-bd47-188d66fa414 Version: \$LATEST
2022-03-11T16:38:52.901-07:00	a87ccfca-0374-4296-bd47-188d66fa414 INFO Using DynamoDB Table for recording state: orgtestingData-RootCAtable@ACCL1528~
2022-03-11T16:38:52.901-07:00	2022-03-11T23:38:52.901Z a87ccfca-0374-4296-bd47-188d66fa414 INFO Handling event: {"RequestType": "Create", "ResourceProperties": {"DistinguishedName": "CN=AWS Root CA", "Subject": "CN=AWS Root CA", "SubjectAltNames": "DNS:awsrootca.com", "KeyUsage": "DigitalSignature", "NotBefore": "2022-03-11T16:38:55.565Z", "NotAfter": "2022-03-11T23:38:55.565Z", "SignatureAlgorithm": "SHA256WithRSAEncryption", "Version": "3"}, "PhysicalResourceId": "bf3562a47e8e51828f1889c1fd8f770f220e4d8e7d7a26213b5d094000622de", "OldResource": null}
2022-03-11T16:38:55.565-07:00	2022-03-11T23:38:55.564Z a87ccfca-0374-4296-bd47-188d66fa414 INFO CfnResponse: {"Status": "SUCCESS", "Reason": "See CloudWatch Logs -- Group: /aws/lambda/orgtestingData-RootCAGenerator\$4AF5237-gVXF9xlv4B0s", "PhysicalResourceId": "bf3562a47e8e51828f1889c1fd8f770f220e4d8e7d7a26213b5d094000622de", "OldResource": null}
2022-03-11T16:38:55.564Z	a87ccfca-0374-4296-bd47-188d66fa414 INFO CfnResponse:
	{ "Status": "SUCCESS", "Reason": "See CloudWatch Logs -- Group: /aws/lambda/orgtestingData-RootCAGenerator\$4AF5237-gVXF9xlv4B0s", "PhysicalResourceId": "bf3562a47e8e51828f1889c1fd8f770f220e4d8e7d7a26213b5d094000622de", "OldResource": null}
2022-03-11T16:38:55.566-07:00	2022-03-11T23:38:55.566Z a87ccfca-0374-4296-bd47-188d66fa414 ERROR Invoke Error {"errorType": "TypeError", "errorMessage": "The \\"url\\\\" argument is required."}
2022-03-11T16:38:55.600-07:00	END RequestId: a87ccfca-0374-4296-bd47-188d66fa414
2022-03-11T16:38:55.600-07:00	REPORT RequestId: a87ccfca-0374-4296-bd47-188d66fa414 Duration: 2981.11 ms Billed Duration: 2982 ms Memory Size: 128 MB Max Memory Used: 85 MB ..

Step 5: Place the certificate and private key into a secret

After the certificate and private key are generated, they need to be manually placed in the original Secrets Manager to keep them secure.

1. Sign in to the AWS Management Console and open the [Secrets Manager](#) console.
2. Select the secret with the name <your-studio-name>/RootCA-X.509-Certificate. This is the secret generated by the Lambda function. The **Created on (UTC)** should be the current date.

Secret name	Description	Last retrieved (UTC)	Created on (UTC) ▾

3. On the secret details page, choose **Retrieve secret value** in the **Secret value section**.
 - Notice the value of the secret. You will need it for *step 6*.
4. Select the secret with the name beginning with <your-studio-name>Data/RootCA-X.509-Certificate.
 - This is the new secret created by Lambda.
5. On the secret details page, choose **Retrieve secret value** in the **Secret value section** to update the secret value.
6. Choose **Edit** and enter the value of the root CA that you found in *step 3a*.
7. Select **Save**.
8. Select the secret with the name <your-studio-name>/RootCA-X.509-PrivateKey.
 - This is the secret generated by the Lambda function.
9. On the secret details page, choose **Retrieve secret value** in the **Secret value section**.
 - Notice the value of the secret. You will need it for *step 12*.
10. Select the secret with the name beginning with <your-studio-name>Data/RootCA-X.509-PrivateKey.
11. On the secret details page, choose **Retrieve secret value** in the **Secret values section** to update the secret value.
12. Choose **Edit** and enter the value of the root CA that you found in *step 9a*.
13. Select **Save**.

Now you can redeploy Nimble Studio again by following the [Update to latest StudioBuilder version](#) tutorial. All other certificates that are signed by the root CA will be automatically generated.

Rotate the render queue certificate

The render queue certificate is used for encryption traffic to the render queue. It is signed by the root CA and is generated in the service tier. The render queue can be rotated more often than the root ca. You need to rotate the render queue certificate if you don't need to rotate the root certificate. You don't need to rotate the render queue certificate after root ca is rotated.

 **Note**

The rotation procedure requires an outage of the render farm so that a maintenance window should be scheduled.

The best way to rotate this certificate is to remove and redeploy the service tier.

To remove the service tier

1. Sign in to the AWS Management Console and open the [AWS CloudFormation](#) console.
2. Select the <your-studio-name>Service stack. The stack must be currently running.
3. In the stack details pane, select **Stack actions** and then select **Edit termination protection**. CloudFormation displays the **Edit termination protection** dialog box.
4. Choose **Disable**, and then select **Save**.
5. Select **Save**. In the stack details pane, choose **Delete**.
6. Select **Delete stack** when prompted.

Now you can redeploy Nimble Studio again by following the [Update to latest StudioBuilder version](#).

Managing studio users

This administrator content shows how to manage studio users in Nimble Studio. As an administrator, you can add and remove users, and allow access to resources.

Topics

- [Adding studio users](#)
- [Removing studio users](#)

Adding studio users

In this administrator tutorial, you'll learn how to add studio users to your Amazon Nimble Studio cloud studio. Add studio users before others can sign in and use your studio. Adding studio users involves launching a virtual workstation (or instance) and then using tools on that instance to add new users to your AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD). The following steps show you how.

Contents

- [Prerequisites](#)
- [Step 1: Sign in to Nimble Studio portal as Admin](#)
- [Step 2: Accept the EULA](#)
- [Step 3: Launch a virtual workstation](#)
- [Step 4: Add users to AWS Managed Microsoft AD](#)
- [Step 5: Sync Active Directory and users in IAM Identity Center](#)
- [Step 6: Add users to Nimble Studio](#)
- [Troubleshooting](#)
- [Related resources](#)

Estimated time: 30 minutes

Prerequisites

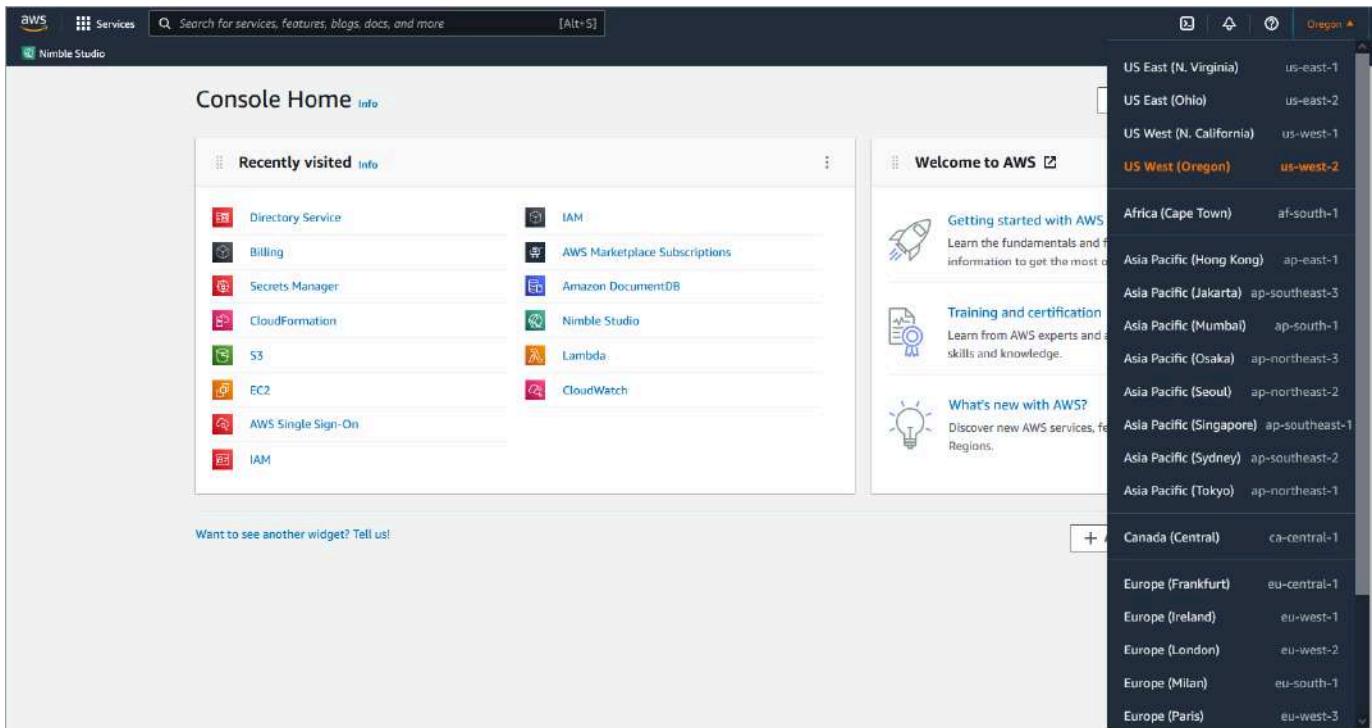
- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- You need the administrator password for your studio's AWS Managed Microsoft AD.
- You also need administrator access to the AWS Management Console for your account.

Step 1: Sign in to Nimble Studio portal as Admin

Next, sign in to the Nimble Studio portal as **Admin** and launch a Windows virtual workstation (an instance).

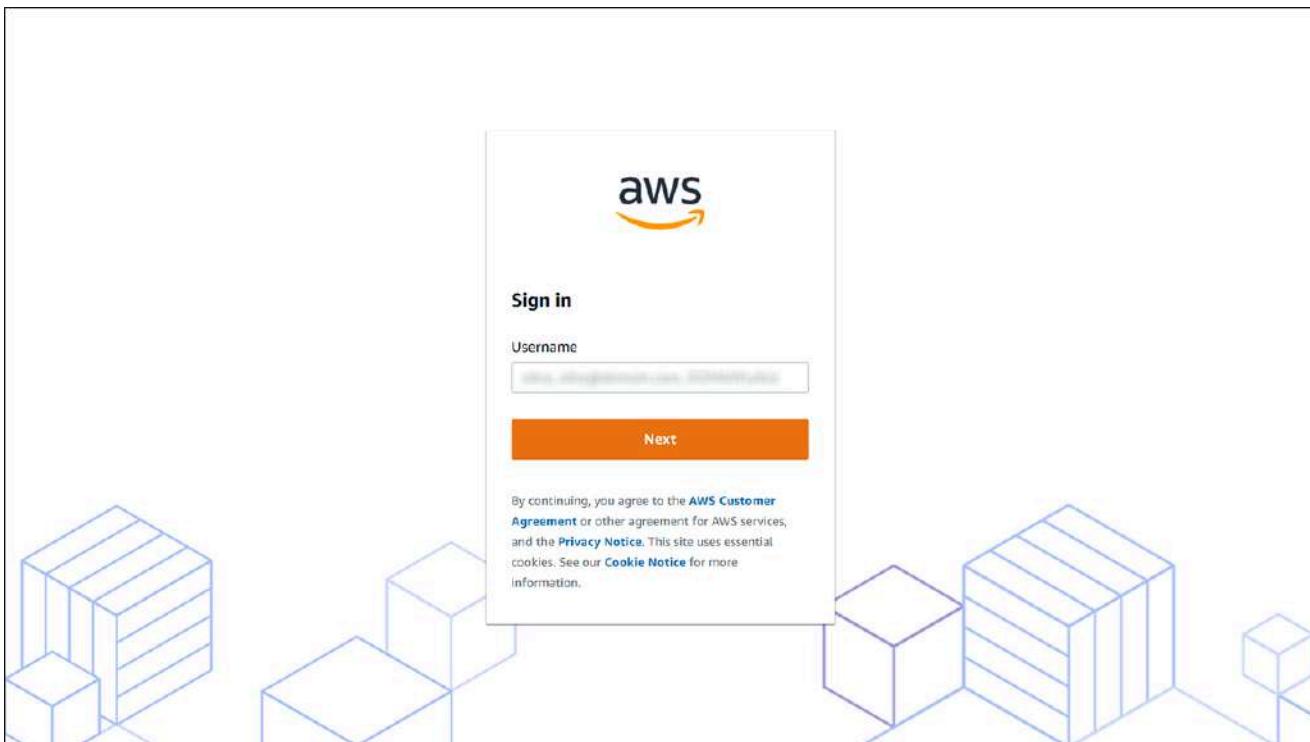
To connect to the Nimble Studio portal

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.

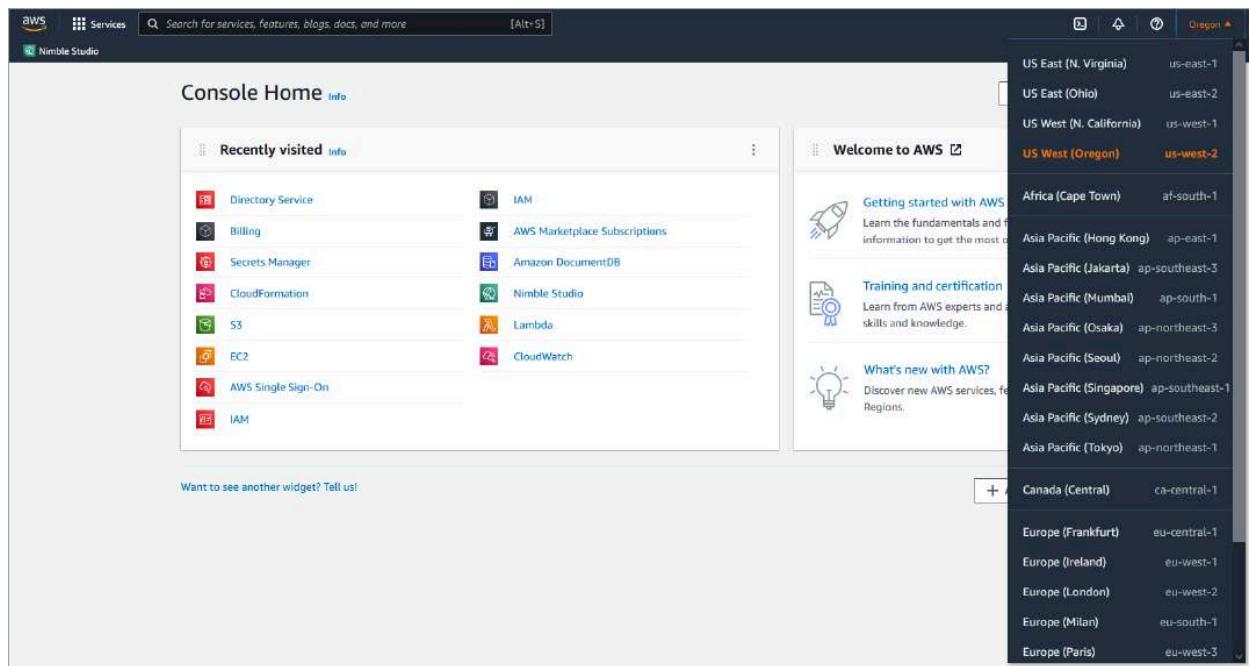


3. Choose **Studio manager** in the left navigation pane.
4. On the **Studio manager** page, choose **Go to Nimble Studio portal**.

5. Sign in to the Nimble Studio portal using your AWS Managed Microsoft AD administrator credentials.
 - a. Enter **Admin** as the user name. Use the password that you set up during StudioBuilder deployment.



5. Sign in to the Nimble Studio portal using your AWS Managed Microsoft AD administrator credentials.
 - a. Enter **Admin** as the user name. Use the password that you set up during StudioBuilder deployment.
 - b. If you forgot your password, do the following:
 - i. Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
 - ii. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



- iii. Select the **Directory ID** for your studio's Active Directory.
 - iv. Choose **Reset user password**.
6. Bookmark your portal's URL so that you can get to your studio directly, later.

Step 2: Accept the EULA

Before using Nimble Studio, accept the End User License Agreements. You can access these agreements on the settings page in the Nimble Studio portal.

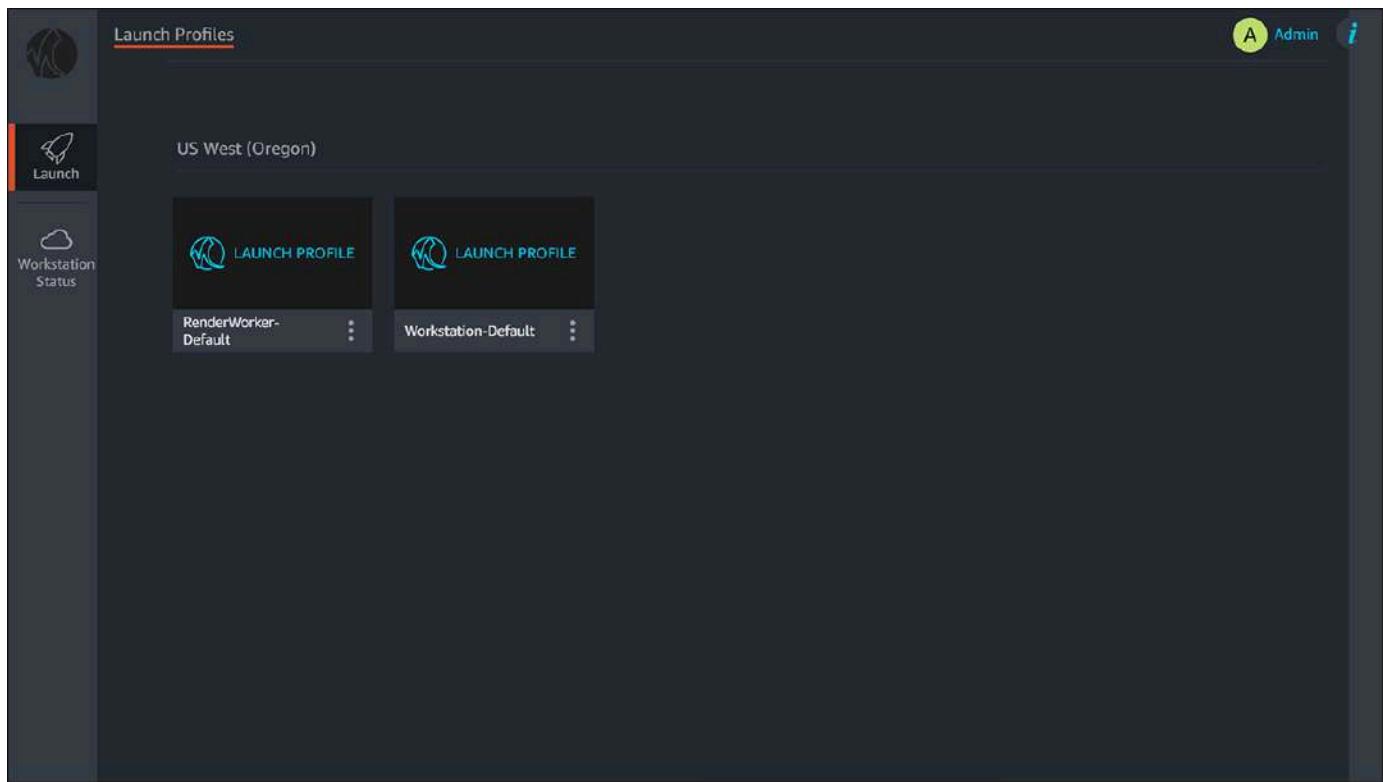
- Sign in to the Nimble Studio portal.

Step 3: Launch a virtual workstation

Now that you've accepted the EULA, you can continue on to launching a virtual workstation. Before you can launch a virtual workstation, first install the latest [DCV client](#).

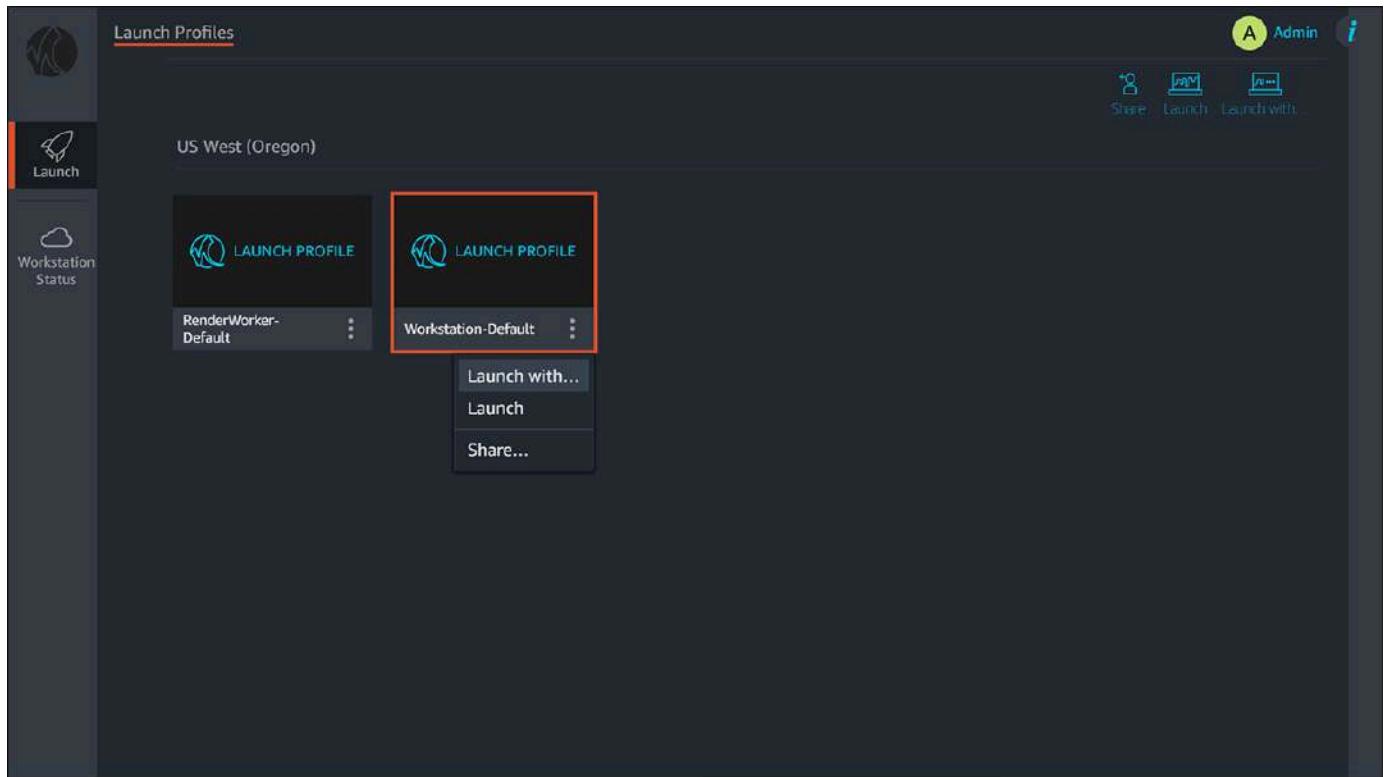
To launch a virtual workstation

1. Choose the **Launch** tab from the left navigation pane.



2. Select the vertical ellipsis

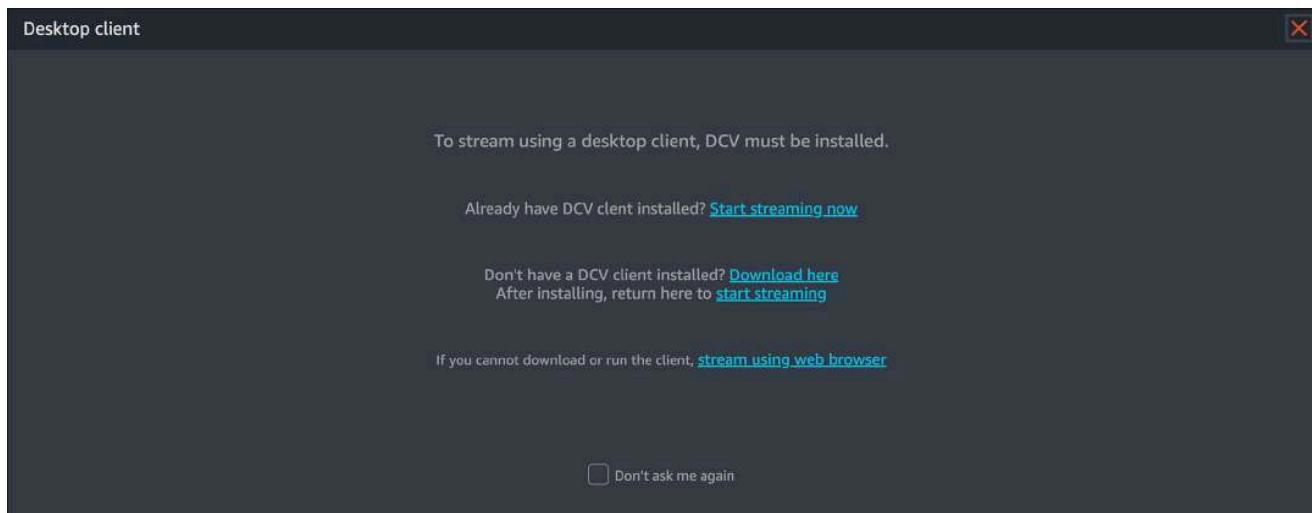
(⋮) on the card to open a dropdown menu.



3. Choose **Launch with...**
4. For **Instance Type**, keep it at the default setting.
5. For **Amazon Machine Image**, verify that **NimbleStudioWindowsStreamImage** is selected.
6. For **Streaming Preference**, choose your streaming preference.
 - a. For the best performance, we recommend choosing **Launch native client**.
 - b. You must download the NICE DCV client before connecting to your workstation. For more information about the DCV client, as well as links to download, see NICE DCV clients [NICE DCV clients](#).
7. Choose **Launch**.
8. A status bar will appear that shows you the progress of launching your virtual workstation. This might take up to 10 minutes.

To connect to the virtual workstation

1. When your virtual workstation is ready, a new window appears reminding you that the client must be installed.
2. Choose **Start streaming now**.
 - If you haven't installed the NICE DCV desktop client, choose **Download here** and install the client first.



3. When your browser displays a window that prompts you to open NICE DCV, choose **Open** to continue. The exact wording of this might vary depending on what browser you're using.

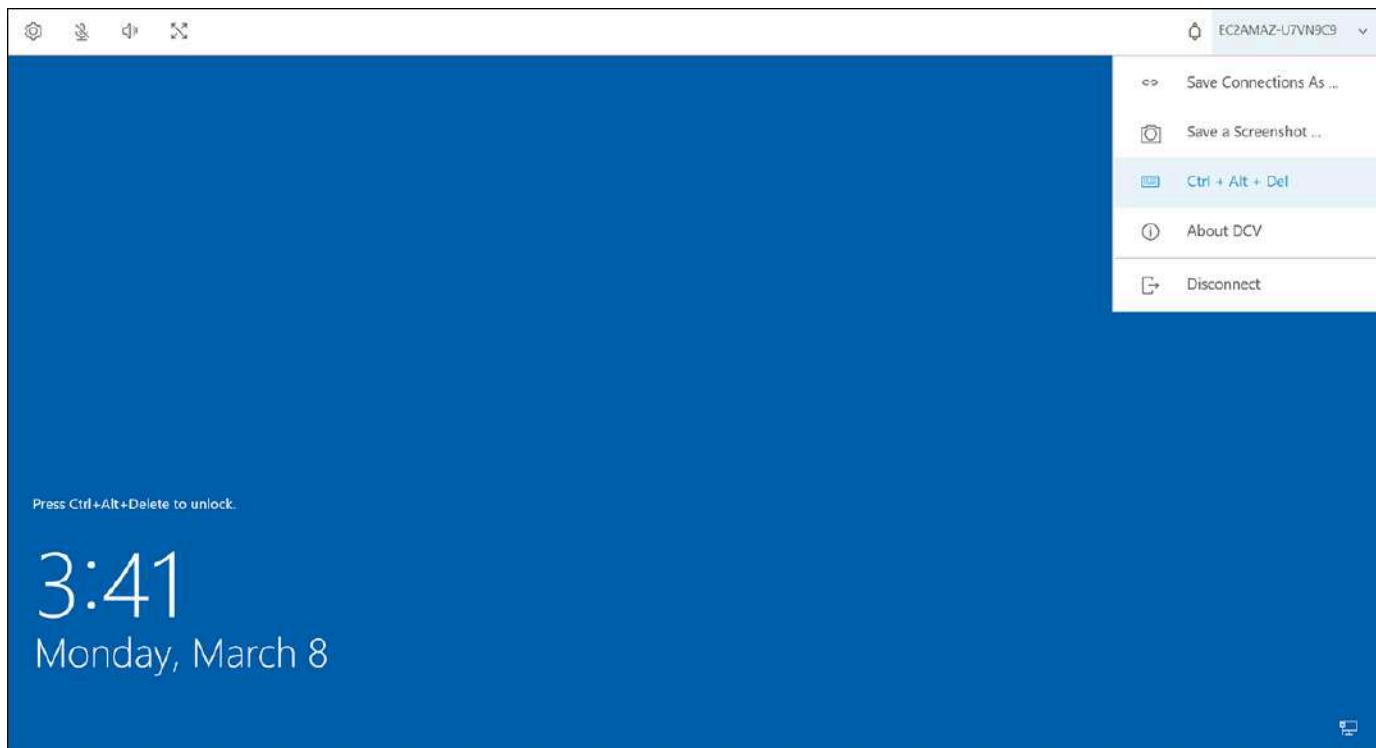
Note

The NICE DCV web browser client runs inside a web browser. You don't need to install the web client. We recommend using the Google Chrome browser to avoid latency. For more information, go to the [Web browser client](#) page in the *NICE DCV User Guide*.

4. After the NICE DCV client application opens in a new window, the Windows login screen will display.
5. Open the instance menu near the top right of the screen and choose **Ctrl + Alt + Del**. For an OS X DCV client, open the **Connection** dropdown menu and select **Send Ctrl + Alt + Del**.

Important

Don't enter **Ctrl+Alt+Delete** on your keyboard. Doing so sends the command to your local computer, not to your workstation.



6. For **User name**, enter **Admin**. For **Password**, enter the password that you created during your studio deploy. Then press the enter (or return) key.

You're now connected to your virtual workstation.

Step 4: Add users to AWS Managed Microsoft AD

After you're logged in to your virtual workstation, the next step is to add users to your studio's AWS Managed Microsoft AD. You will use custom Windows PowerShell commands provided by Nimble Studio to add and manage your users. These custom commands configure AWS Managed Microsoft AD so that your users are able to work on both Windows and Linux virtual workstations.

Add new users

Important

Use the **New-NimbleUser** command described in the following instructions to add your users. If you attempt to use the built-in AWS Managed Microsoft AD commands or UI, your users might not be set up correctly to work with both Windows and Linux virtual workstations.

To add new users

1. Choose the **Start** menu in the lower left-hand corner of your desktop.
2. Enter **PowerShell** to search for **Windows PowerShell** and then choose it from the top of the search results.

Note

There might be multiple search results with similar names. Make sure to choose the result that is named just **Windows PowerShell**.

3. Enter **New-NimbleUser** into PowerShell and press the enter (or return) key.
4. Enter a **user name** for the user and press the enter (or return) key.
 - The user name must use lower case letters (a-z) and numbers. The user name must start with a letter and must be 2-20 characters long.
5. Enter a temporary **password** for the user and press the enter (or return) key. The user will change this password on their first login.
 - a. The password must be eight or more characters long.

- b. The password must contain characters from three of the following categories: Uppercase letters A-Z, lowercase letters a-z, numbers 0-9, and special characters, such as \$ # %
6. Enter the user's **email address** and press the enter (or return) key.

 **Note**

If you don't input an email for the user, they won't be able to change or reset their password. Each user must have a unique email address or they won't be able to sign in to Nimble Studio portal.

- After the user has been created, a confirmation message will appear with information about the user's groups, UID and GID.
7. (Optional) If you would like to specify a first name or last name when creating your users, add the `-DisplayFirstName` and `-DisplayLastName` flags to your `New-NimbleUser` command. For example:

```
New-NimbleUser -DisplayFirstName "Martha" -DisplayLastName "Rivera"
```

- a. The command will prompt you to enter their user name, password and email address.
- b. You can also specify the user name and email address in your `New-NimbleUser` command, as well:

```
New-NimbleUser -DisplayFirstName "Martha" -DisplayLastName "Rivera"  
-UserName "martha" -EmailAddress <your email address>
```

- c. For more information about how to use the `New-NimbleUser` command, enter `get-help New-NimbleUser -detailed` and press the enter (or return) key.

8. Repeat this process for each artist or user on your team.

Other user commands

In addition to `New-NimbleUser`, there are other custom commands available that you can use to manage your users.

Security group inbound rules

Command	Description
New-NimbleUser	Creates a new user in the AWS Managed Microsoft AD that is also usable in Linux.
Remove-NimbleUser	Deletes a user that was created by New-NimbleUser.
Set-NimbleUser	Updates an existing AWS Managed Microsoft AD user so it can be usable in Linux.
Repair-NimbleStudioAD	Iterates through all users and groups in the AWS Managed Microsoft AD so that they have proper IDs, GIDs, and other requirements, making them usable in Linux.
Test-NimbleStudioAD	Tests the state of the AWS Managed Microsoft AD configuration.
New-NimbleGroup	Creates a new group in the AWS Managed Microsoft AD that can be used in Linux
Set-NimbleGroup	Updates a group with a group ID.

9. The commands you will use most frequently are **New-NimbleUser** and **Remove-NimbleUser**.
10. If you accidentally add a user with the AWS Managed Microsoft AD commands or UI, you can use **Set-NimbleUser** to update that user so that it will be usable on Linux virtual workstation or use **Repair-NimbleStudioAD** to update all users and groups to be usable on Linux.
11. **Test-NimbleStudioAD** will test the state of your AWS Managed Microsoft AD and list any problems that are detected.
12. **New-NimbleGroup** and **Set-NimbleGroup** are used to manage groups and group IDs to be used for Linux. The management of those groups is beyond the scope of this tutorial.

View current users

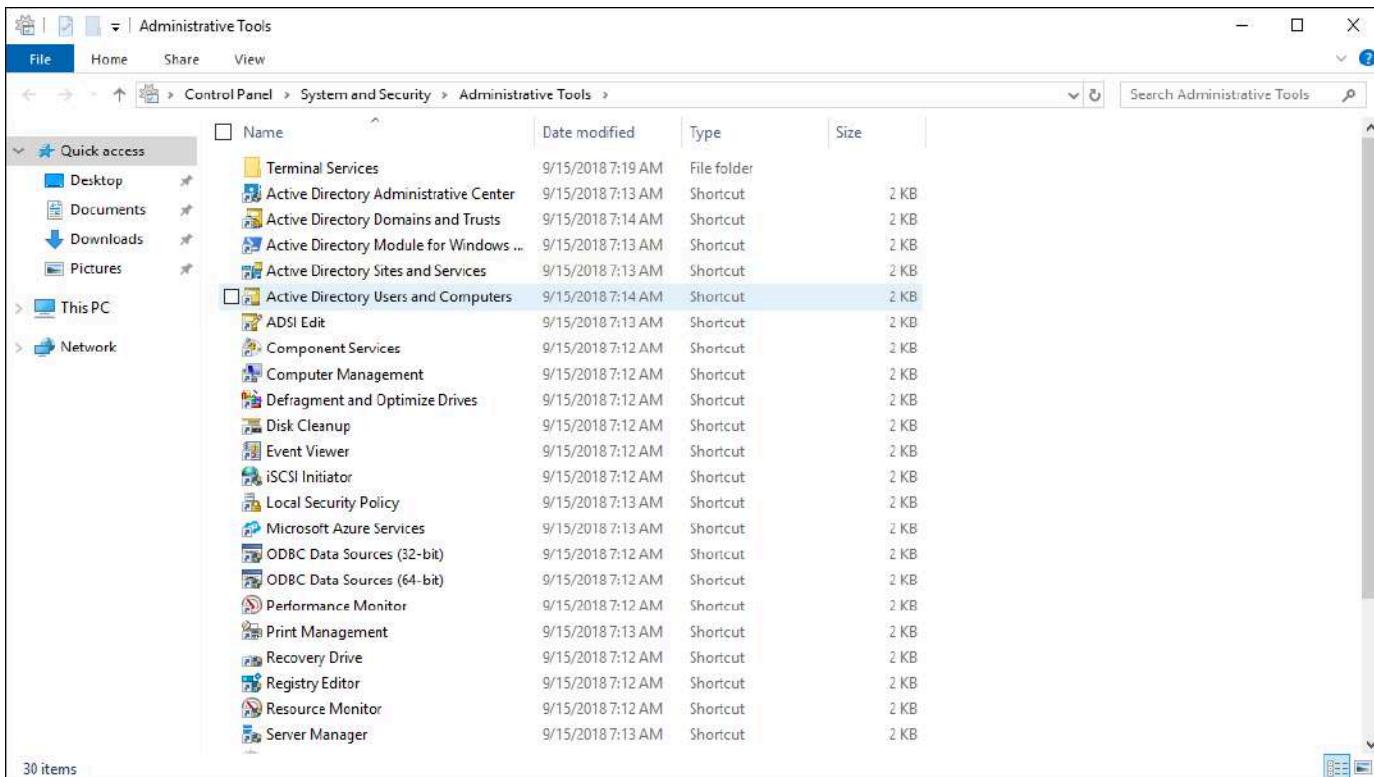
To view a list of your users, you will need to use the AWS Managed Microsoft AD tools that are built in to Windows.

To view a list of your current users

1. Choose the **Start Menu**.
2. Enter **Admin** to search for Administrative Tools and then choose **Administrative Tools** from the top of the search results.
3. In the window that appears, open (double-click) **Active Directory Users and Computers**.

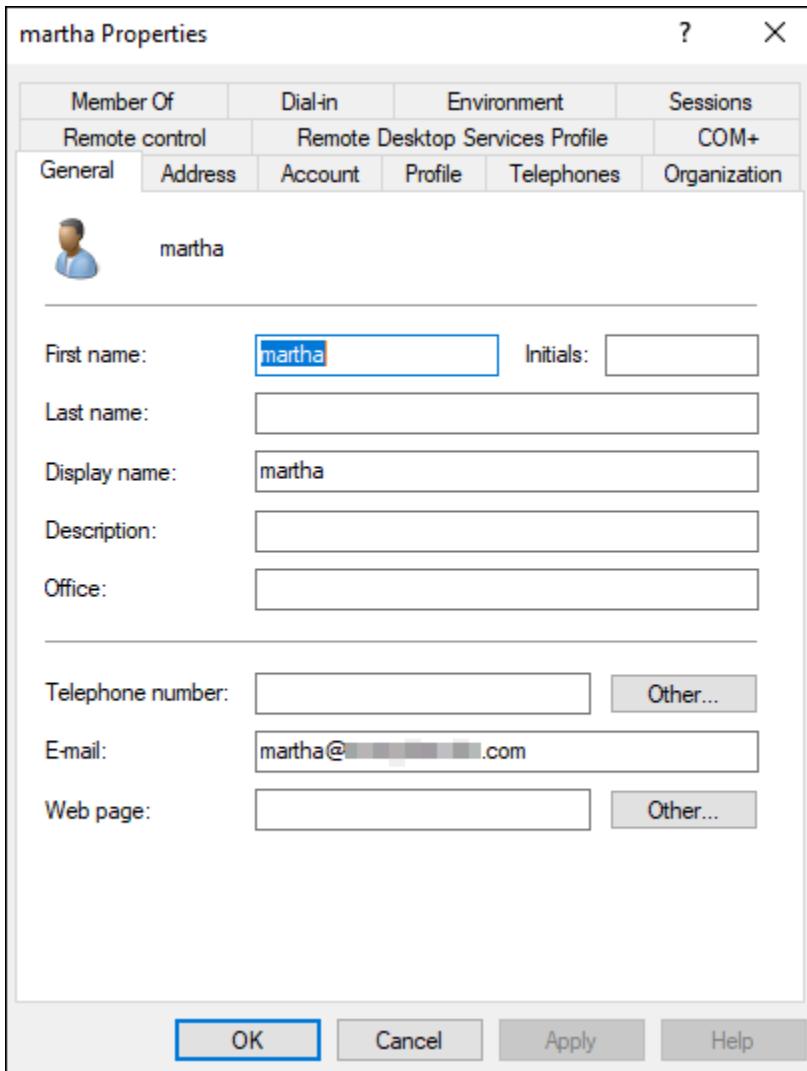
Note

It might take several minutes for your virtual workstation to connect to the AWS Managed Microsoft AD and for the **Active Directory Users and Computers** window to appear.



4. Open your domain (example: **ad.<your-studio-name>.nimble.<region>.aws**) by choosing the arrow to the left of your domain name.
 - You might notice a delay when interacting with some parts of the Active Directory Users and Computers UI. The UI might appear to freeze, but after a minute or two, it will become responsive again.

5. Open your NetBIOS name (**ad**) by choosing the arrow to the left of it. Then choose **Users**.
6. For your NetBIOS name, choose **Users**.
 - A list of your created users and groups will display.
7. Open (double-click) a user to open the **Properties** window for the user.



- You can make simple changes to the user, such as changing their name or email address, or user group assignment within this window. However, don't attempt to use the built-in Windows Active tools or this UI to add new users or groups. Use **New-NimbleUser** and the other custom commands described previously or your users and groups might not have the proper IDs to work with Linux.

Step 5: Sync Active Directory and users in IAM Identity Center

Users from the AWS Managed Microsoft AD must be manually synced to the IAM Identity Center user pool before they can be used in other AWS applications that use IAM Identity Center and AWS Managed Microsoft AD. Follow these steps to add members to Nimble Studio when using the IAM Identity Center and Active Directory Sync feature.

1. Sign in to the AWS Management Console and open the [IAM Identity Center](#) console.
2. Choose **Users** from the left navigation pane.
 - If you successfully enabled sync, a green banner with **Start guided setup** will display.
3. Select **Manage sync**.
4. Select **Add users and groups**.
5. Choose the **Users** tab. For **User**, enter the exact user name and choose **Add**.
6. When all of the users have been added, select **Submit**.
7. Wait for the users to appear in the IAM Identity Center user pool.
 - It can take 10–20 minutes for users to appear.

Step 6: Add users to Nimble Studio

Next, you will add the AWS Managed Microsoft AD users to Nimble Studio as studio users.

To add users

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio manager** in the left navigation pane.
3. Choose **User access** from the left navigation menu.
4. You aren't adding any administrators at this time, so skip this page and choose **Next**.
5. In the **Search text** field, enter the name of one of the users that you just added and choose **Search**.
6. Select the user from the search results. This will add them to a **Selected users and groups** list.
7. Repeat the searching and selecting steps for all the users that you created.
8. When all of your users are listed in Selected users and groups, choose **Next**.
9. Confirm that your users are listed in the **Selected users** list, and then choose **Save**.

10. Send users their user name and the URL to your studio, and direct them to follow the instructions in the [Logging in to the Nimble Studio portal](#) tutorial. This tutorial includes instructions about how to set an initial password.
 - a. If users don't follow the [Logging in to the Nimble Studio portal](#) tutorial and reset their password, they won't be able to sign in.
 - b. To give users access to resources after login, complete *step 12*.
11. After AWS Managed Microsoft AD has synced with IAM Identity Center, [Step 3: Share launch profiles with studio users](#) so they can access resources.
 - Generally, syncing IAM Identity Center with AWS Managed Microsoft AD can take up to four hours. To share launch profiles immediately, you can tell users to sign in to the Nimble Studio portal. That will force AWS Managed Microsoft AD to sync with IAM Identity Center.

Troubleshooting

I got disconnected from my remote machine and can't reconnect.

If your streaming session is left idle for more than one hour, it will automatically disconnect. It's still available and all of your open applications are running. You just need to reconnect.

1. Go back to the browser window where the **Nimble Studio** portal is open.
2. Hover over the streaming icon and choose the **vertical ellipsis** menu.
3. Choose **Reconnect stream**.
4. Choose **Proceed to native client**.
5. Choose **Open DCV client**.

Related resources

- [Creating launch profiles](#)

Removing studio users

In this administrator tutorial, you'll learn how to remove studio users from your Amazon Nimble Studio cloud studio. After a studio user completes a project, it's a security best practice to remove

that user. That way, only artists who are working on a project can see the information for that project.

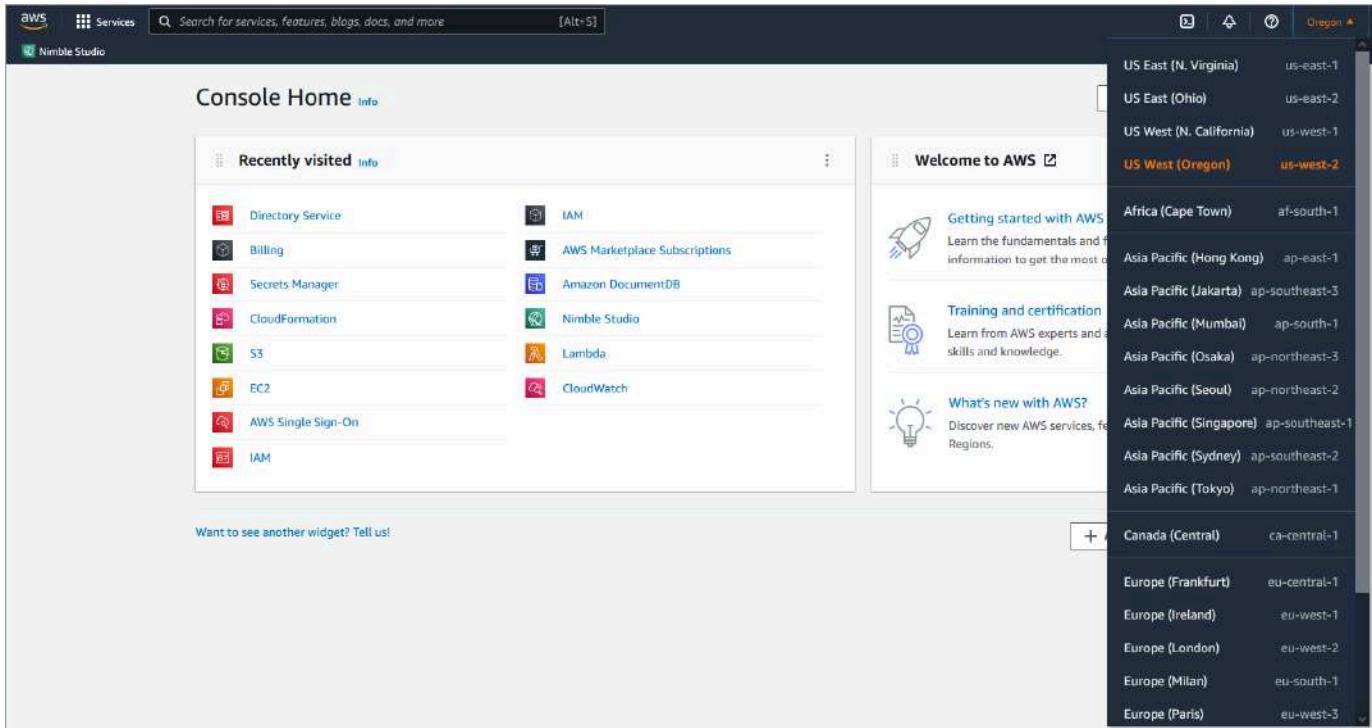
Contents

- [Step 1: Sign in to the portal as an administrator](#)
- [Step 2: Run PowerShell commands to remove users](#)

Step 1: Sign in to the portal as an administrator

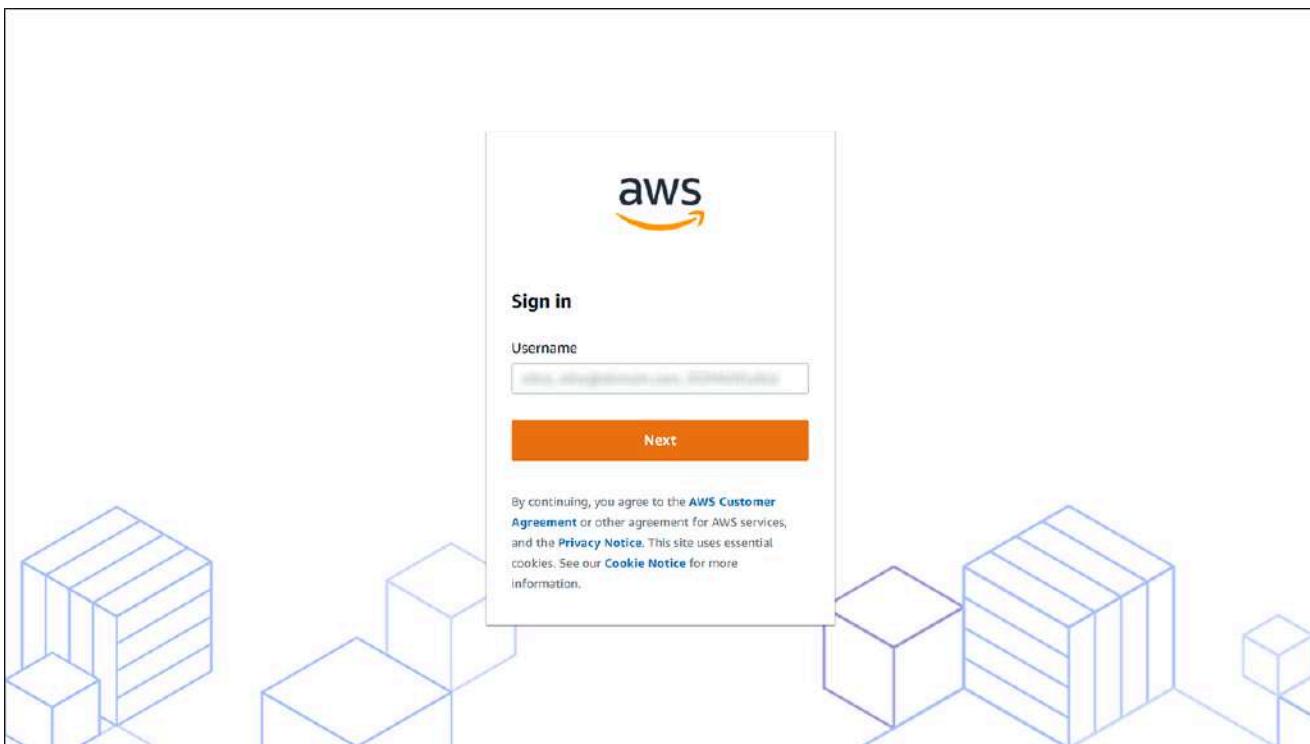
To connect to the Nimble Studio portal

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.

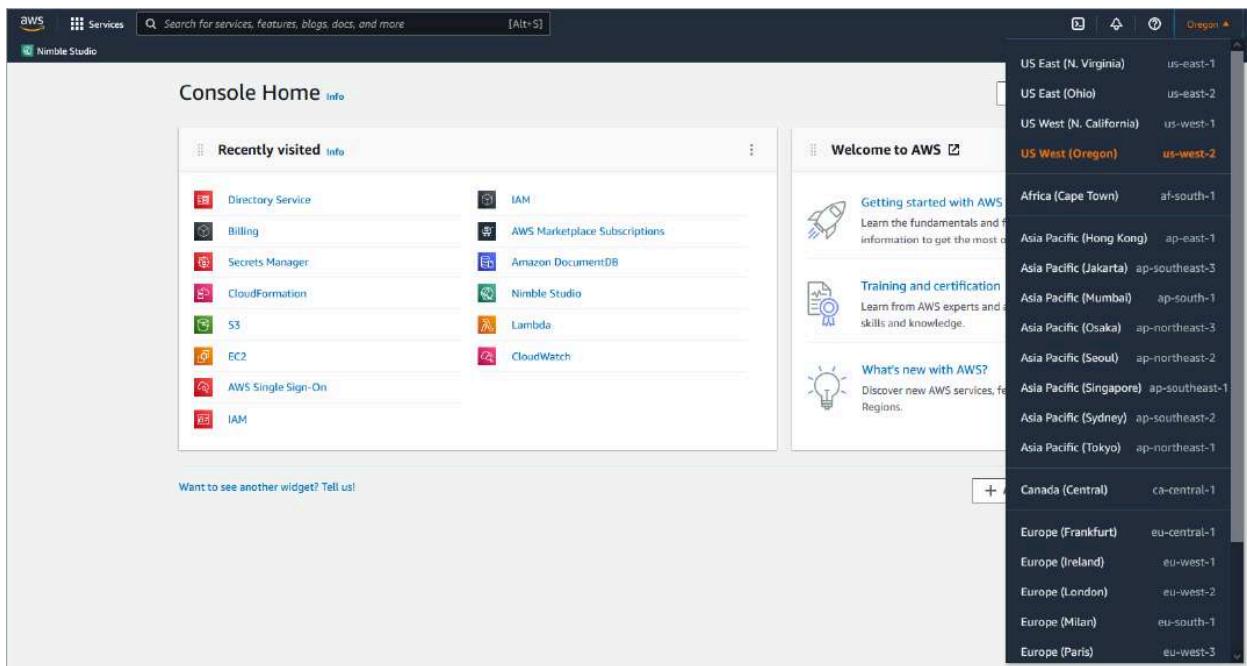


3. Choose **Studio manager** in the left navigation pane.
4. On the **Studio manager** page, choose **Go to Nimble Studio portal**.
5. Sign in to the Nimble Studio portal using your AWS Managed Microsoft AD administrator credentials.

- a. Enter **Admin** as the user name. Use the password that you set up during StudioBuilder deployment.



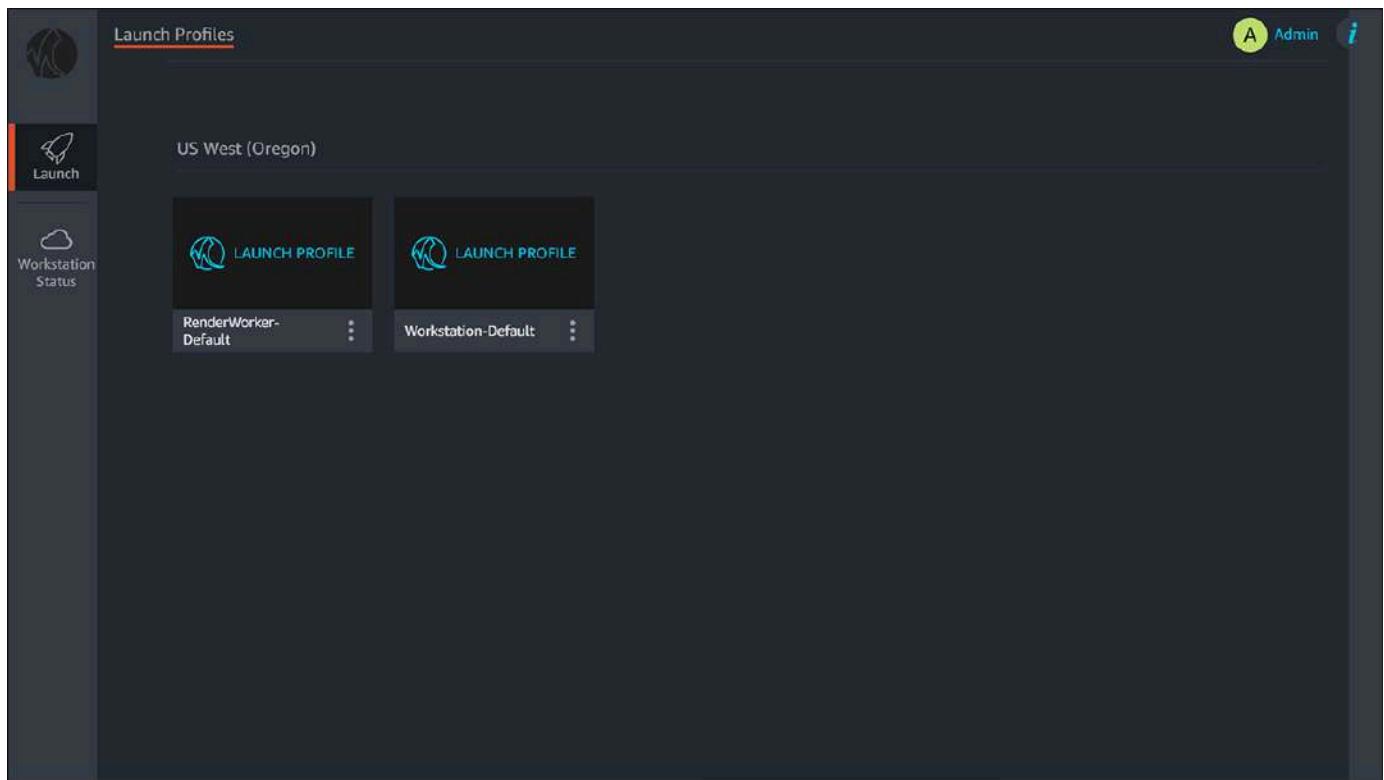
- b. If you forgot your password, do the following:
 - i. Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
 - ii. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



- iii. Select the **Directory ID** for your studio's Active Directory.
- iv. Choose **Reset user password**.
6. Bookmark your portal's URL so that you can get to your studio directly, later.

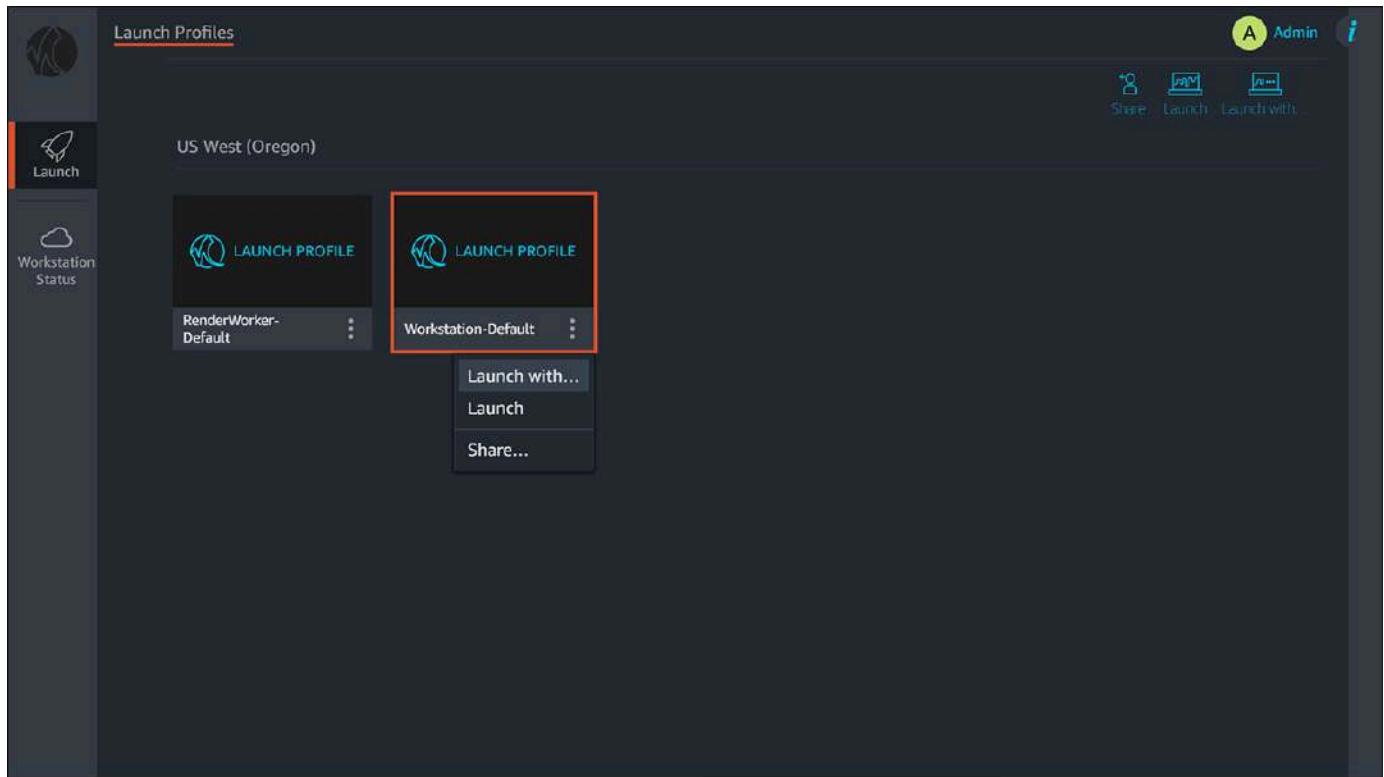
To launch a virtual workstation

1. Choose the **Launch** tab from the left navigation pane.



2. Select the vertical ellipsis

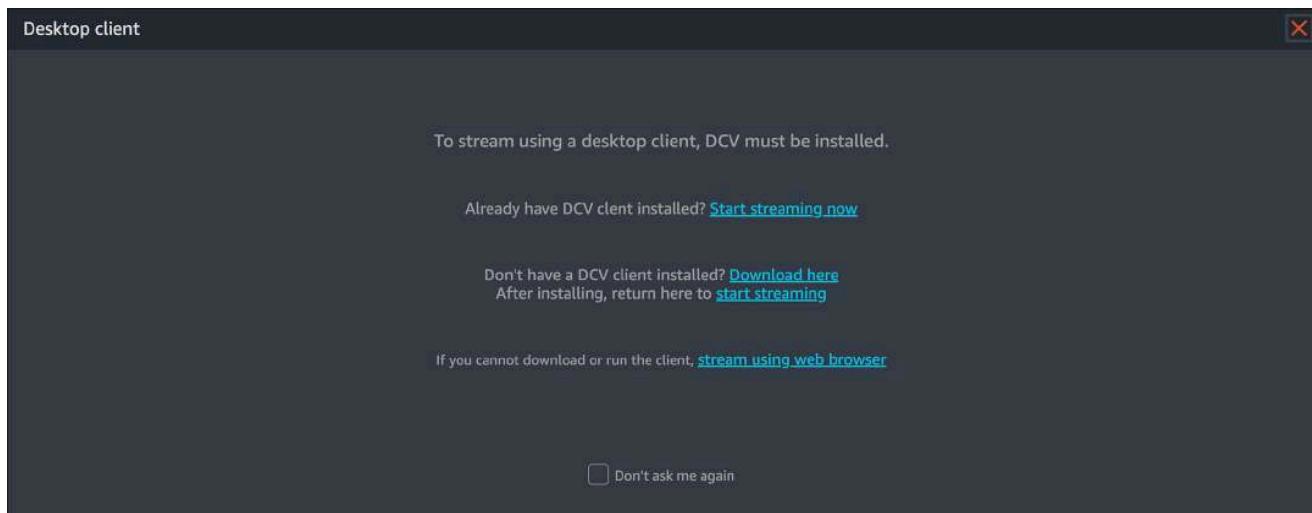
(⋮) on the card to open a dropdown menu.



3. Choose **Launch with...**
4. For **Instance Type**, keep it at the default setting.
5. For **Amazon Machine Image**, verify that **NimbleStudioWindowsStreamImage** is selected.
6. For **Streaming Preference**, choose your streaming preference.
 - a. For the best performance, we recommend choosing **Launch native client**.
 - b. You must download the NICE DCV client before connecting to your workstation. For more information about the DCV client, as well as links to download, see NICE DCV clients [NICE DCV clients](#).
7. Choose **Launch**.
8. A status bar will appear that shows you the progress of launching your virtual workstation. This might take up to 10 minutes.

To connect to the virtual workstation

1. When your virtual workstation is ready, a new window appears reminding you that the client must be installed.
2. Choose **Start streaming now**.
 - If you haven't installed the NICE DCV desktop client, choose **Download here** and install the client first.



3. When your browser displays a window that prompts you to open NICE DCV, choose **Open** to continue. The exact wording of this might vary depending on what browser you're using.

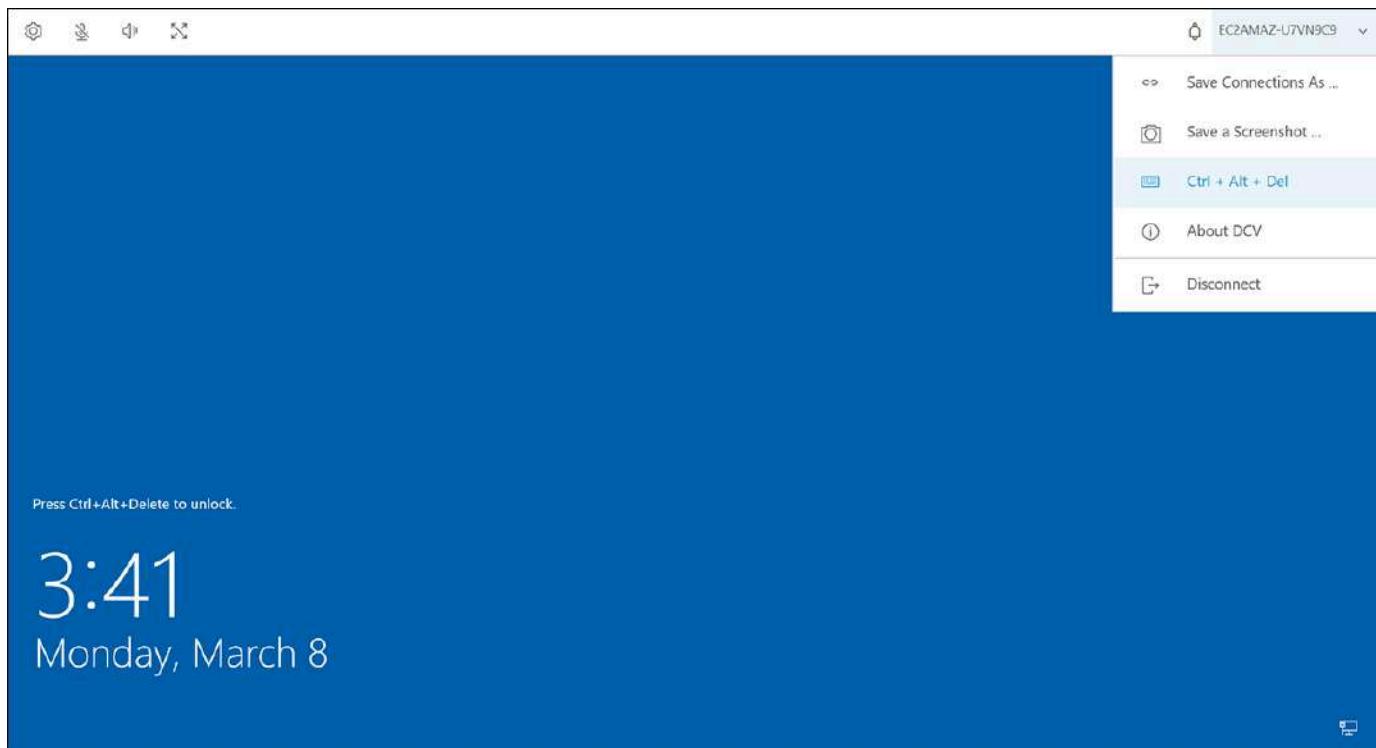
Note

The NICE DCV web browser client runs inside a web browser. You don't need to install the web client. We recommend using the Google Chrome browser to avoid latency. For more information, go to the [Web browser client](#) page in the *NICE DCV User Guide*.

4. After the NICE DCV client application opens in a new window, the Windows login screen will display.
5. Open the instance menu near the top right of the screen and choose **Ctrl + Alt + Del**. For an OS X DCV client, open the **Connection** dropdown menu and select **Send Ctrl + Alt + Del**.

Important

Don't enter **Ctrl+Alt+Delete** on your keyboard. Doing so sends the command to your local computer, not to your workstation.



6. For **User name**, enter **Admin**. For **Password**, enter the password that you created during your studio deploy. Then press the enter (or return) key.

You're now connected to your virtual workstation.

Step 2: Run PowerShell commands to remove users

1. Choose the **Start** menu in the lower-left corner of your desktop.
2. Enter PowerShell to search for **Windows PowerShell**. Then choose it from the top of the search results.
3. Run the following command for each user that you want to remove: Remove-NimbleUser <username>
 - a. For example: Remove-NimbleUser martha
 - b. For efficiency, create a list of users that you want to remove, then iterate through that list.

```
$users = "user1", "user2", "user3"
foreach ($user in $users) {
    # test and see if the user exists
    if (Test-NimbleUser $user)
    {
        Remove-NimbleUser $user
    }
}
```

You've now removed a user or users from your cloud studio.

Managing your workstations

After creating your Amazon Nimble Studio, you can further configure and manage it by making custom workstations. This includes allowing your workstations to enter the stopped state, allowing them to upload files, and granting them superuser access, among other things. The tutorials in this section will walk you through how to manage your workstations.

Topics

- [Starting and stopping workstations](#)
- [Enabling uploads to Nimble Studio workstations](#)
- [Session auto backup](#)
- [Use API to create a streaming session](#)
- [Provisioning workstations in multiple Availability Zones or Local Zones](#)
- [Provide administrator access for Windows users](#)
- [Provide Superuser access for Linux users](#)

Starting and stopping workstations

This tutorial is for admins who use or manage workstations and want to learn how to start and stop them. For information about how artists can stop and start workstations, see [Starting and stopping workstations](#).

With *persistent storage*, you can start and stop your workstations while preserving your root device volume between artist working sessions. This means that whenever you sign in, your previous workstation settings persist (are unchanged). This helps virtual workstations operate more like on-premises computers.

By default, you can start and stop your workstations. To turn off this feature, follow the instructions in the [Turn off persistent workstations](#) section of this guide.

Important

When persistence is disabled, Amazon Nimble Studio supports free root device volume storage for AMIs that are smaller than 500 GB. By default, persistence is enabled, and you

are charged for all Amazon EBS related costs. To learn more about the cost of persistent workstations, see [Amazon Nimble Studio Pricing](#).

Contents

- [Workstation states](#)
- [Start, stop, or terminate a workstation](#)
- [Set streaming limits](#)
- [Turn off persistent workstations](#)

Workstation states

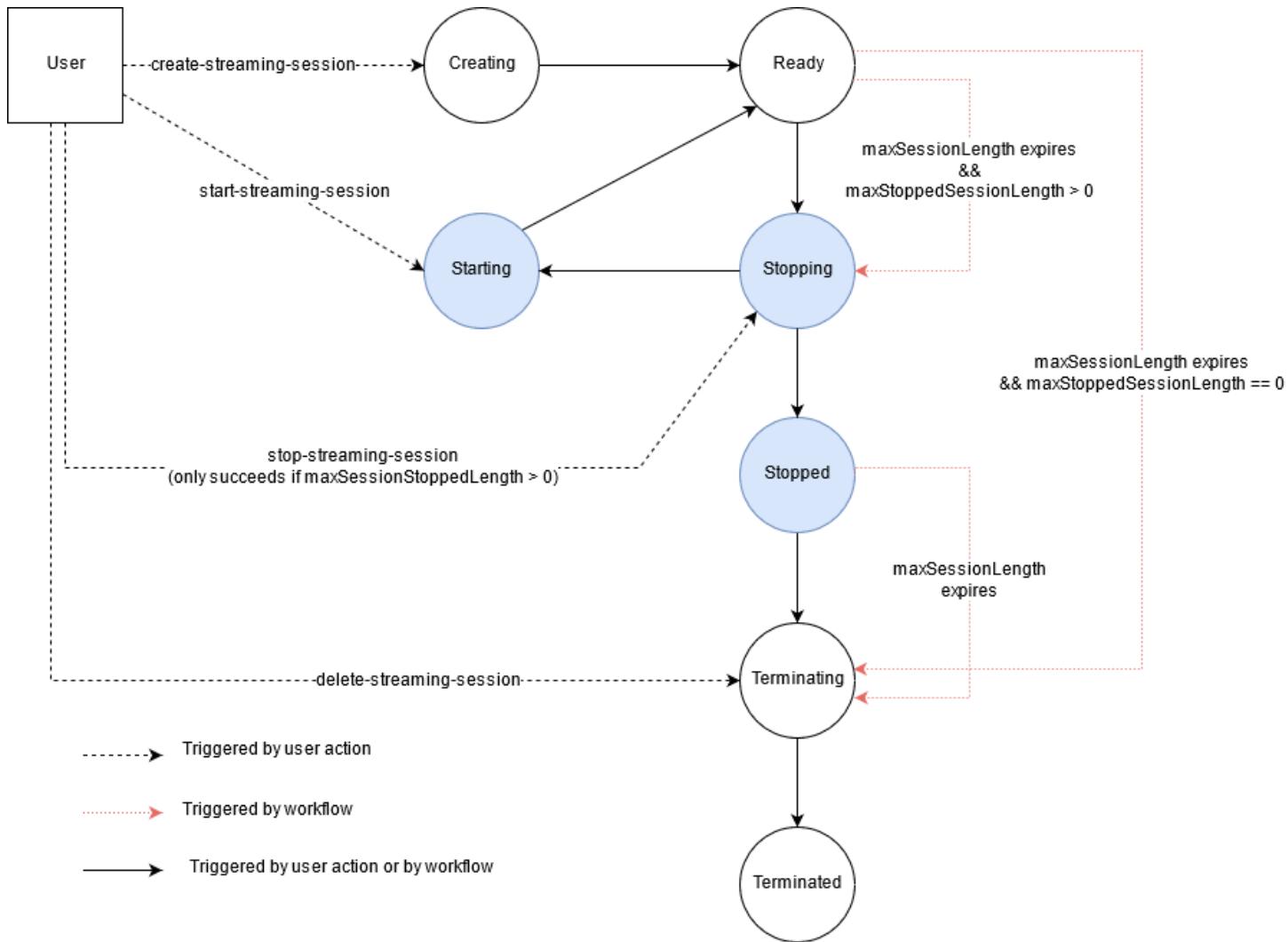
Your workstation transitions through different states, from the moment it's launched, through to its termination.

Stopped workstations have two main benefits compared to terminated workstations. Startup times are reduced, giving artists fast access to their workstation when they resume their work. Additionally, Stopped workstations provide a persistent environment for artists to work in because the EC2 instance and the root EBS volume are both preserved.

You aren't charged for Stopped instances. Stopped instances are automatically terminated after a maximum of four days from the moment the workstation was last stopped. This means that your artists will still need to periodically start new sessions.

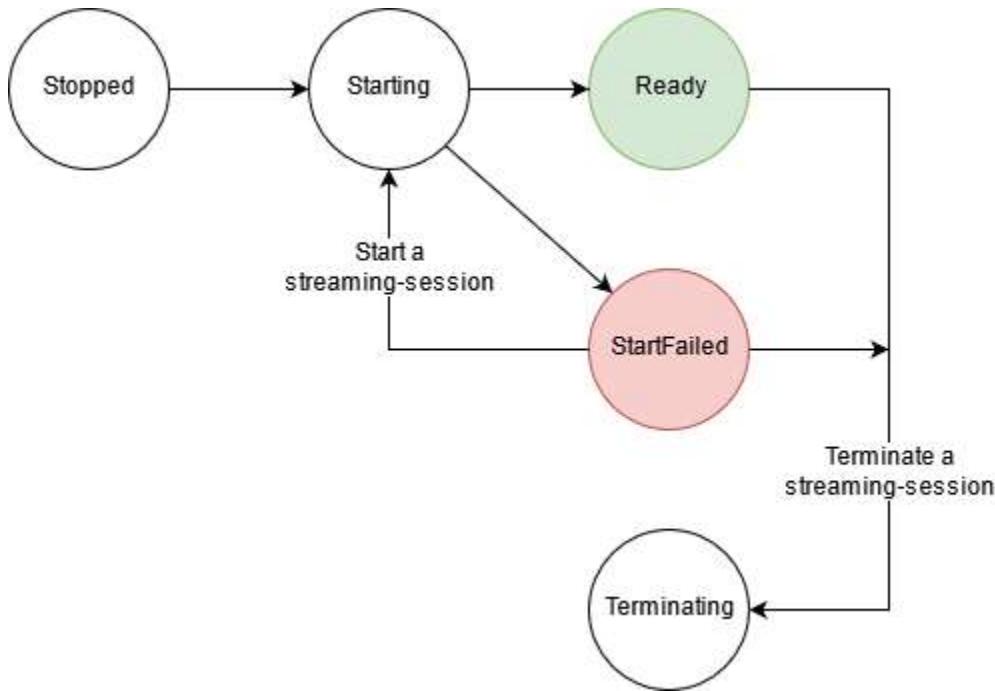
Workstation state: normal workflow

The following diagram shows the transition between workstation states when no failures are encountered. When you launch an instance for the first time, the status transitions from Creating to Ready. A Running workstation transitions to Stopped either because of a user request, an API request, or because the maximum runtime was reached. A Stopped workstation transitions to Terminated either because of a user request, an API request, or because the maximum stopped time was reached. A Stopped workstation transitions to a Running state when either by user or by API request.



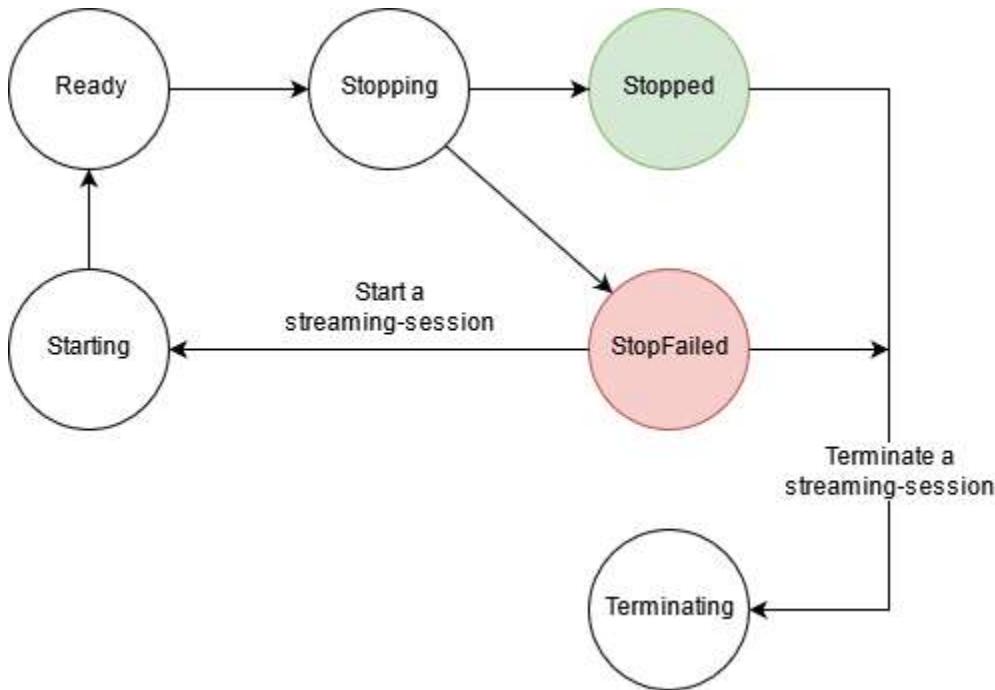
Workstation state: failure to start streaming

The following diagram shows the transitions between workstation states when it encounters a failure to start streaming a session. Whenever you start a stopped session, the session either progresses to a Ready state or a StartFailed state. From the StartFailed state, you can try to start the session again; if restarting fails, you can choose to terminate the session.



Workstation state: failure to stop streaming

The following diagram shows the transitions between workstation states when it encounters a failure to stop a streaming session. Whenever you start a stopped session, the session transitions from the StopFailed state to the Stopping and Ready states. If the session fails to stop, it enters the StopFailed state, where you can start the session, or choose to terminate the session.

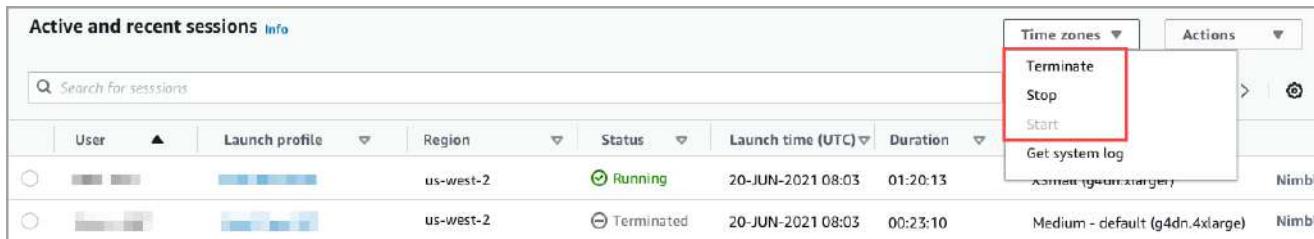


Start, stop, or terminate a workstation

You can start, stop, or terminate a workstation from the AWS Management Console or from [the portal](#).

To start, stop, or terminate a workstation from the AWS Management Console

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane. Then choose **Streaming sessions**.
3. Select the streaming session that you want to modify the state of.
4. Select **Actions**. Depending on what state that your workstation is in, you might not be able to start or stop the workstation.
5. Choose one of the following three options.
 - a. Choosing **Start** will start up a stopped workstation that is associated with the streaming session that you selected.
 - b. Choosing **Stop** will change the workstation status to Stopped. You won't need to reinstall any software or reconfigure the workstation.
 - c. Choosing **Terminate** will change the workstation status to Terminated. Your artist will need to set up a new workstation the next time they launch a workstation.



6. Follow the prompts to **Start**, **Stop**, or **Terminate** the workstation.

Set streaming limits

To automatically stop or terminate launch profiles, you can set streaming limits for their running and stopped states.

⚠️ Important

If you allow instances to remain in a stopped state indefinitely, AWS charges you the full cost of persisting Amazon EBS storage.

To set streaming limits

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Choose a launch profile to modify its streaming limits.
4. Choose **Action**. Then choose **Edit**.
5. In the **Streaming session storage volume** section, enter the maximum times for the **Running** state and the **Stopped** state.
6. Select **Update launch profile**.

Your running sessions will now stop automatically and your stopped sessions will now terminate automatically.

Turn off persistent workstations

By default, all new launch profiles have persistence turned on. One way to avoid paying for a persistent workstation session's EBS volume is to turn off persistence in all of your launch profiles. The workstation session's EBS volume will then be non-persistent, and you won't pay for EBS volume costs while workstations are running.

To turn off persistent workstations

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Choose a launch profile to modify its streaming limits.
4. Choose **Action**. Then choose **Edit**.
5. In the **Streaming session storage volume** section, clear the check box next to **Enable persistent storage**.
6. Select **Update launch profile**.

You've now turned off persistent storage. Workstation sessions that are created after this update can be terminated, but can't be stopped. However, existing workstations do have the ability to stop.

With persistence turned off, back up local workstation session data to group storage, and terminate all workstation sessions. The workstations will preserve their EBS volumes until they are terminated.

Enabling uploads to Nimble Studio workstations

This tutorial is for admins who want to enable uploads to their artists' workstations. For information about how artists can upload files to their workstations, see [Uploading files to your virtual workstation](#).

You can allow your artists to upload data from their local machine to their workstation by enabling Amazon Nimble Studio uploads. By enabling uploads, your artists can upload images and other files to their workstations for content creation. When you enable uploads, you can specify the location on the workstation where artists can upload their data.

Contents

- [Enable uploads by configuring a launch profile](#)

Enable uploads by configuring a launch profile

In these section, you'll configure a launch profile to allow users to upload files from their local workstations to their Nimble Studio workstations.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select a launch profile or [Creating launch profiles](#) for your artists to use when uploading files.
4. Choose **Action**. Then choose **Edit**.
5. In the **Streaming session upload** section, select the box next to **Enable uploads**.
6. In the **Linux upload location**, enter a valid directory path. This is the location where files will be uploaded for Linux workstations. Make sure that the artists who will use this launch profile have access to the folder that you choose.
 - If you don't specify a location, the directory path defaults to \$HOME/Downloads.

7. In the **Windows upload location**, enter a valid directory path. This is the location where files will be uploaded for Windows workstations. Make sure that the artists who will use this launch profile have access to the folder that you choose.
 - If you don't specify a location, the directory path defaults to %HOMEPATH%\Downloads.
8. Choose **Update launch profile**.

Your artists can now upload files from their local workstations onto their Nimble Studio workstation by using this launch profile.

 **Note**

If your artists are using a Linux workstation, their uploaded files won't persist after a session is over, unless the `LinuxHome` studio component attached to the launch Profile. For instructions about how to create the `LinuxHome` studio component, follow the [Setting up Linux home directories](#) tutorial.

Session auto backup

This tutorial is for admins who use or manage workstations and want to learn how to back up and restore sessions. For information about how artists restore sessions, see [Session auto backup](#).

Contents

- [Prerequisites](#)
- [Turn on session auto backup](#)
- [Restore from backup](#)
- [Delete EBS volumes](#)

Prerequisites

- Turn on persistence and configure session storage by following the instructions in [Starting and stopping workstations](#).

Turn on session auto backup

When you have session auto backup enabled, Nimble Studio automatically backs up your streaming session storage when your streaming sessions are Stopped. Nimble Studio will also back up your streaming session storage every four hours that your streaming sessions are Running.

Session auto backup only covers data that is stored locally on an instance. It doesn't cover shared storage, such as Amazon FSx. For backup of shared storage, use one of the methods listed in the [How to back up your studio data](#) page.

Data stored in your roaming profile persists through backup restores. This is important to understand because different programs store user preferences on either the root volume, or within a user's roaming profile. Some applications support local (root) settings, some support roaming profiles. For example, if you change your desktop background and then restore from a backup, your desktop background will remain the same.

Note

Your first backup will take longer to create than all subsequent backups. This is because Nimble Studio is backing up your entire Amazon EBS volume. Every subsequent backup only backs up your most recent changes.

To turn on session auto backup

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Choose a launch profile to modify its streaming limits.
4. Choose **Action**. Then choose **Edit**.
5. In the **Streaming session storage volume** section, select the check box next to **Enable persistent storage**.
 - To allow session auto backup, you must allow persistent workstations. For information about starting and stopping your streaming sessions, see [Starting and stopping workstations](#).
6. To enable backups, select **Turn on auto backup** in the **Auto backup** section. This sets up your sessions for back up every 4 hours that they're Running.

7. Choose the **Maximum backups per streaming session**.
8. Select the check box next to **I understand that streaming session storage backups incurs a cost**.
9. Select **Update launch profile**.

Automatic backups are now allowed. After the launch profiles are updated, the backup settings will take effect on new streaming sessions that are launched after you've updated the launch profile.

 **Important**

If you deactivate session auto backup in a launch profile, Nimble Studio continues to back up sessions that were created when session backup was activated.

Restore from backup

When restoring to a backup, you replace all existing EBS volume data with the backup's data. If there is any specific data you want to keep before restoring from a backup, save that data to a shared storage device. You can copy that data back to the workstation volume after the backup is restored.

 **Important**

Auto backup creates a backup every time a session is Stopped, and every 4 hours that the session is Running. If you restore from a backup and stop the session, Nimble Studio will create a backup of that session volume's current state.

To restore from backup

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane. Then choose **Streaming sessions**.
3. Choose the streaming session that you want to restore from backup.
4. Select **Actions**. Then choose **Restore backup**.
5. Select a backup to restore to.
6. Enter restore into the field and choose **Restore backup**.

You've now restored your streaming session to a previous state.

 **Note**

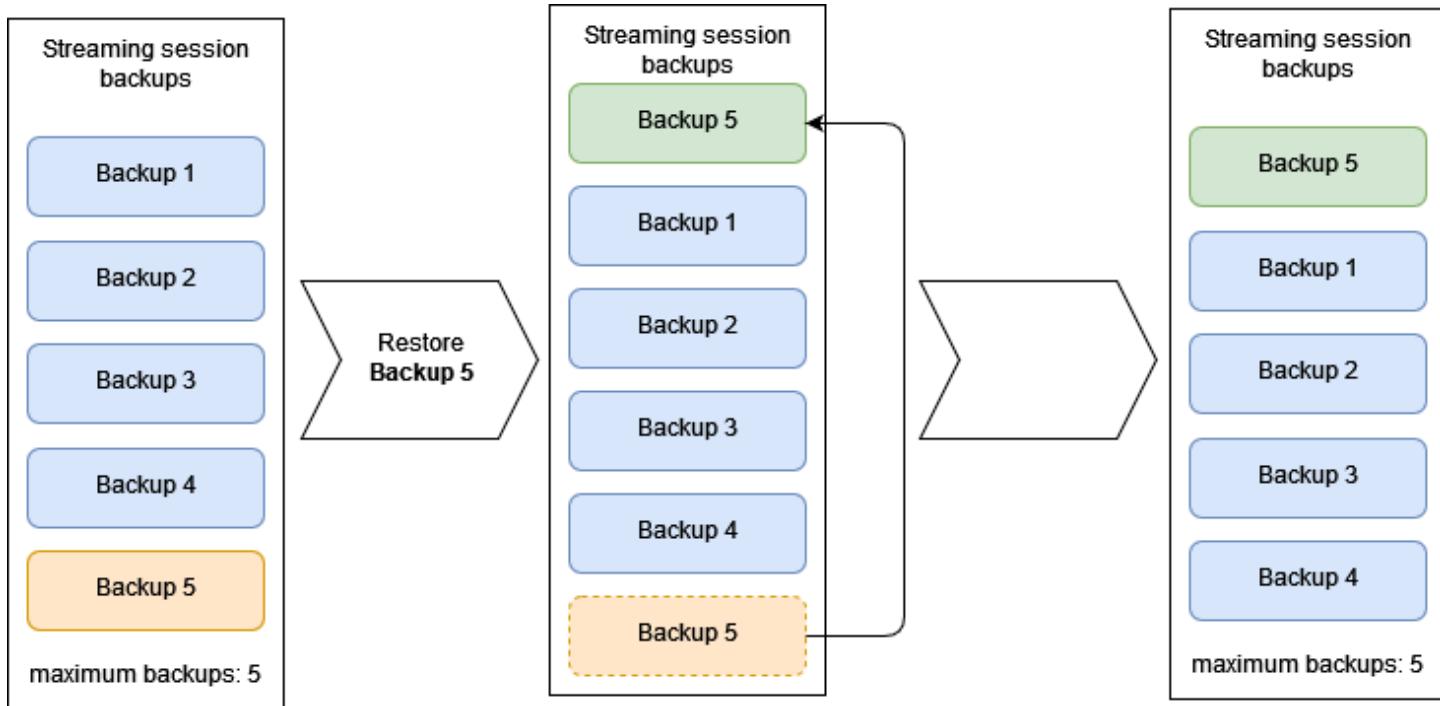
If you restore from a backup that was created while your session was Running, a prompt will display after the user logs into the workstation. The prompt states: Why did the computer shut down unexpectedly? This is expected behavior for Amazon Elastic Compute Cloud (Amazon EC2), Amazon EBS Snapshot, and Windows Server interactions.

For more information about creating and restoring backups of your Amazon EBS volumes that are attached to Windows on Amazon EC2 instances, see [Create a VSS application-consistent snapshot](#).

Every time you stop a session, a new backup of that session is created with the current timestamp, and it's marked as the most recent backup. New backups are created until you reach the maximum number of backups.

After you reach the maximum number of backups, every time a new backup is created, the oldest backup is deleted. When you restore from a backup, that backup becomes the most recent backup and the timestamp is updated. The timestamp only reflects the date that the backup was created or restored, not when the backup was originally created.

The following diagram depicts restoring from a backup at the maximum number of five backups. When the oldest backup (Backup 5) is restored, a new backup is created by copying it. Then, that oldest backup is deleted, and the new copy becomes the most recent backup.



Delete EBS volumes

You can delete the Amazon EBS volumes attached to a workstation whenever the streaming session is Stopped.

To delete the EBS volume

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane. Then choose **Streaming sessions**.
3. Select the streaming session that has an EBS volume that you want to delete.
4. Select **Actions**. Then choose **Stop**.
5. A pop-up titled **Stop session?** appears. Select the check box next to **Delete EBS volume**.
6. Enter delete in the confirmation field. Then choose **Stop**.

You've deleted the EBS volume associated with this streaming session. When you start this streaming session again, you must select a backup that the streaming session will be restored from.

Use API to create a streaming session

This tutorial will teach you how to use Python and the Amazon Nimble Studio API to create a streaming session for your artists without requiring them to start a session manually from the Nimble Studio portal.

In this tutorial, you will be using [AWS CloudShell](#), a browser-based shell that allows you to interact with AWS resources from the AWS Management Console. You will be making API calls using the Python interpreter in interactive mode. This allows you to run commands one at a time to learn how they work.

Note

CloudShell is currently unavailable in Canada (ca-central-1) and London (eu-west-2). If you're deployed in those regions, or if you would prefer to run these commands on your own local machine and build a repeatable Python script, see [Installation](#) and [Configuration](#) in the [Boto3 Documentation](#).

Contents

- [Prerequisites](#)
- [Using Python commands with API](#)
- [Step 1: Prepare CloudShell](#)
- [Step 2: Start Python 3 and import modules](#)
- [Step 3: Create variables](#)
- [Step 4: Create a Boto3 session](#)
- [Step 5: Create Directory Service client](#)
- [Step 6: Create IdentityStore client and get user identity](#)
- [Step 7: Create Nimble client and spin up streaming session](#)
- [Step 8: Get sessionId for a streaming session](#)
- [Step 9: Get the current state of streaming session](#)
- [Step 10: Terminate a streaming session](#)
- [Troubleshooting](#)
- [Related Resources](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- You also need an understanding of the Python programming language.

Using Python commands with API

To create a streaming session for one of your artists using the Nimble Studio API, you will require **five** important pieces shown in the following list. In the API, an ID is represented as **Id**, therefore this documentation might use both terms.

- **The user ID** - The IAM Identity Center UserId of the artist for which the streaming session will be launched.
- **The launch profile ID** - The identity number of the launch profile that you want to use for the streaming session. Use this as the value for launchProfileId.
- **The Amazon Machine Image (AMI) ID** - The AMI for the streaming session that you will use. Use this as the value for amazonImageId.
- **The instance type** - The type of instance the streaming session will use. Use this as the value for ec2InstanceType.
- **Your studio ID** - The identity number of the studio. Use this as the value for the studioId.

With this data, you can use the Nimble Studio API to start a session with the [create_streaming_session](#) command. An example of this is shown in the following basic Python script.

```
import boto3
from botocore.config import Config

# Create a boto3 session
session = boto3.Session()

# Create a Nimble session client in region us-west-2
nimble_client = session.client('nimble', region_name = 'us-west-2')

# Create the streaming session
```

```
response = nimble_client.create_streaming_session(  
    ownedBy = 'xxxxxxxx-xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxx',  
    launchProfileId = 'xxxxxxxxxxxxxx-x',  
    amazonImageId = 'xxxxxxxxxxxxxx',  
    ec2InstanceType = 'g4dn.4xlarge',  
    studioId = 'us-west-2:xxxx-xxxxxxxxxx',  
)
```

Important note about commands

Throughout the documentation, you can find commands that you're expected to run. Some of these commands need to be run within the Python interactive session. The commands are presented without the Python interactive prompt: >>>.

Here are example commands, as documented in this tutorial.

```
# Variables  
USERNAME = 'Admin'  
  
print(f'User: {USERNAME}')
```

These are example commands as they look in the Python interactive session.

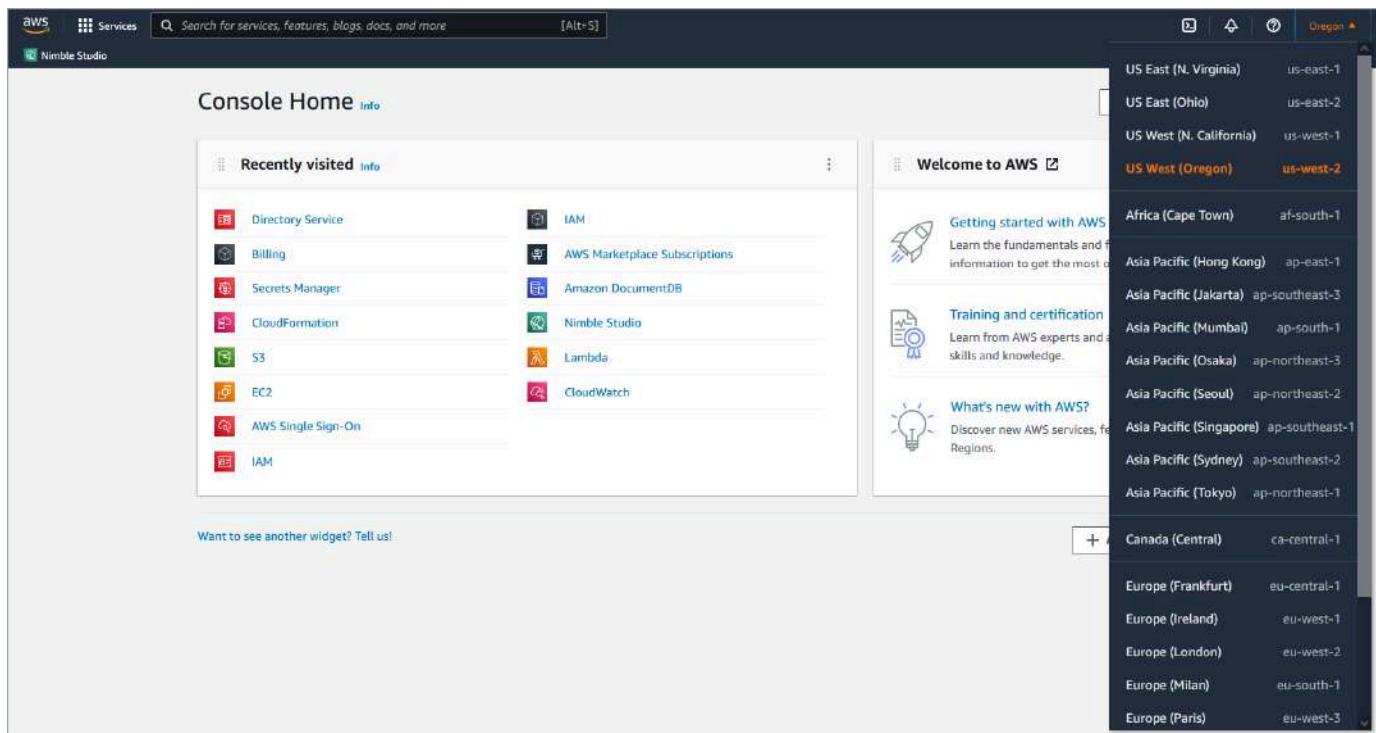
```
>>> # Variables  
>>> USERNAME = 'Admin'  
>>>  
>>> print(f'User: {USERNAME}')
```

Step 1: Prepare CloudShell

Before you can start working with Nimble Studio's API, you'll need to update one of your Python modules. The first step to doing that is launching a CloudShell session. Here's how.

To start up CloudShell

1. Sign in to the AWS Management Console.
2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



3. Sign in to the AWS Management Console.
4. Go to **Services** and then select **CloudShell**.



5. Wait for the CloudShell session to load.
6. The CloudShell icon will launch CloudShell. It might take a few minutes for the environment to run.
 - a. If you have never used CloudShell in the past, a prompt will display that welcomes you to AWS CloudShell.
 - b. Close the prompt by choosing **Close**.
7. When CloudShell is ready, a prompt where you can enter shell commands will display.

Upgrade modules and other requirements

Python 3.6+ is the minimum required version of Python to run for the Boto3 SDK. It's available automatically as part of CloudShell. For more information about CloudShell environments, see [CloudShell compute environment: specifications and software](#) in the AWS CloudShell User Guide.

If you intend to run the SDK on your local machine, see [Installation](#) in the **Boto3 documentation**.

- The minimum required version of **Boto3** is **1.18.19**. To upgrade, run the following command in CloudShell.

```
pip3 install boto3 --upgrade
```

Test Python 3

- To enter a Python 3 interactive shell, run the following command in CloudShell.

```
python3
```

- You will be presented with a Python prompt that looks similar to this.

```
Python 3.7.10 (default, Jun 3 2021, 00:02:01)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-13)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- After receiving the prompt, you're ready to enter Python commands.
- Import **boto3** and enter the `import boto3` command within the Python interactive shell to make sure that it's working.
- You will know you're in the interactive shell if the prompt looks like `>>>`.

```
import boto3
```

- If the `>>>` prompt returns with no errors, your environment is ready, and you can continue to the third step, **To exit out of Python 3....**
 - If you receive an error, you might not have updated `boto3` correctly or you might have entered the `import boto3` command incorrectly. Try again until you receive no errors.

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'boto3'
```

- d. For more information about common errors and how to resolve those, see the [Troubleshooting](#) section.
3. To exit out of Python 3, use the following command.

```
exit()
```

- After exiting, your CloudShell prompt should look similar to the following string.

```
[cloudshell-user@ip-10-0-123-45 ~]$
```

Step 2: Start Python 3 and import modules

In this step, you will import the two modules required by Python to use the Nimble Studio API: **boto3** and **botocore.config**.

Start Python 3

- If you aren't already in a Python 3 interactive shell, run the following command.

```
python3
```

- If you entered Python correctly, a Python 3 prompt will display.

```
[cloudshell-user@ip-10-0-123-45 ~]$ python3
Python 3.7.10 (default, Jun  3 2021, 00:02:01)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-13)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Import modules

1. Import **boto3** and **botocore.config** from within the interactive shell.

```
import boto3
from botocore.config import Config
```

2. If there are no errors, your shell should look similar to the following.

```
[cloudshell-user@ip-10-0-123-45 ~]$ python3
Python 3.7.10 (default, Jun 3 2021, 00:02:01)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-13)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import boto3
>>> from botocore.config import Config
>>>
```

Step 3: Create variables

Next, you will run a series of commands from within the Python interactive shell. The shell will present a prompt that begins with >>>.

You will access the following list of **variables**. These variables will either be used directly in the `create_streaming_session` command (example: **INSTANCE_TYPE**), or used with another API call to grab a specific ID (`Id/id`) (example: **USER_NAME**).

- The **USER_NAME** - The name of the artist for whom you want to launch an instance.
- The **LAUNCH_PROFILE_NAME** - The name of the launch profile you will use to launch.
- The **IMAGE_NAME** - The name of the Amazon Machine Image.
- The **INSTANCE_TYPE** - The type of instance the streaming session will use.
- The **REGION_NAME** - The region in which you will launch the streaming session.

1. Define the **USER_NAME** variable: Enter the user name of the artist for whom you want to launch an instance. The following example uses Admin.

```
USER_NAME = 'Admin'
```

2. The `create_streaming_session` command takes a launch profile ID as a parameter. In an API, the Id is a long string of characters that you can retrieve through the API by providing a human readable name.

- For this example, you can use the **Workstation-Default** launch profile name. If you prefer to use a different launch profile, you can use that instead.

```
LAUNCH_PROFILE_NAME = 'Workstation-Default'
```

3. The **Amazon Machine Image** parameter also requires an Id. As before, you will enter the name of the AMI, and then use the API to get the Id.

```
IMAGE_NAME = 'NimbleStudioWindowsStreamImage'
```

4. The **Instance Type** parameter will be used to define the type of instance that you will launch.

```
INSTANCE_TYPE = 'g4dn.4xlarge'
```

5. Specify the **Region** where you want to launch the instance.

```
REGION_NAME = 'us-west-2'
```

6. When applied together, the code should look similar to the following code.

```
>>> # Define Variables  
>>> USER_NAME = 'Admin'  
>>> LAUNCH_PROFILE_NAME = 'Workstation-Default'  
>>> IMAGE_NAME = 'NimbleStudioWindowsStreamImage'  
>>> INSTANCE_TYPE = 'g4dn.4xlarge'  
>>> REGION_NAME = 'us-west-2'
```

Step 4: Create a Boto3 session

Using Boto3, you will create a session that will store your configuration state and allow you to create service clients and resources. This session will enable you to connect to various service clients, get your artist's user ID, and control Nimble Studio. In this case, you will create three clients:

- [DirectoryService](#) - To manage the Active Directory for your studio.
- [IdentityStore](#) - To retrieve IAM Identity Center user information.
- [NimbleStudio](#) - To work with Nimble Studio.

Before you can create the clients, create the Boto3 session. To do so, enter these commands within the Python interactive shell.

- Create the session by running the following command.

```
session = boto3.Session()
```

For more information about sessions in Boto3, see [Session reference](#) in the [Boto3 documentation](#).

Step 5: Create Directory Service client

Next, get the user ID of an artist from DirectoryService.

1. To create a **Directory Service client**, run the following code in CloudShell.

```
ds_client = session.client('ds', region_name = REGION_NAME )
```

2. Next, use the **client** to get a list of the available directories.

```
response = ds_client.describe_directories()
```

3. Find a directory that has the word 'nimble' in it. This will be the directory used for NimbleStudio.

```
# Get the directory that has the word 'nimble' in it
try:
    directory = [d for d in response['DirectoryDescriptions'] if '.nimble.' in
    d['Name']][0]
except IndexError: # if there is an error
    print('Could not find Nimble Studio Directory')
```

4. Press **return** until you're presented with the Python prompt >>>.
5. Now, find the directory ID.

```
directory_id = directory['DirectoryId']
```

- To see what the code returned, you can print the **DirectoryId**.

```
print(directory_id)
```

- The response will look like d-123456abcde.

6. Get the domain name.

```
domain = directory['Name']
```

- a. Print the domain to see if it returns the correct domain.

```
print(domain)
```

- b. The result should be similar to: <your-studio-name>.nimble.us-west-2.aws.

Step 6: Create IdentityStore client and get user identity

Now that you have the domain, you can use the IdentityStore client to get the UserId of your artist. Continue running these commands within the Python interactive shell.

1. Create the IdentityStore client.

```
# Create an Identity Store client.  
identity_client = session.client('identitystore', region_name=REGION_NAME)
```

2. Get the UserId that matches the name of the user.

```
# List the users in the Identity Store and filter for the USER_NAME  
response = identity_client.list_users(  
    IdentityStoreId=directory_id,  
    Filters=[  
        {  
            'AttributePath': 'UserName',  
            'AttributeValue': f'{USER_NAME}@{domain}'  
        },  
    ]  
)  
  
user_id = response['Users'][0]['UserId']
```

- a. Review the UserId.

```
print(user_id)
```

- b. The result should be similar to: xxxxxxxx-xxxxxx-xxxx-xxxx-xxxx-xxxxxxxx.

Step 7: Create Nimble client and spin up streaming session

In this section, you will create a Nimble Studio client session that will allow you to connect to the Nimble Studio service. Use these commands within the Python interactive shell.

Create a Nimble Studio client session

- Create a **Nimble Studio client** with this command.

```
nimble_client = session.client('nimble', region_name = REGION_NAME )
```

Get the studio ID

1. Use the Nimble Studio API to list all of the **available studios** for this account.

```
response = nimble_client.list_studios()
```

2. Get the `studioId` that will be used with the `create_streaming_session` command.

```
studio_id = response['studios'][0]['studioId']
```

- a. To review the `studioId`, print it out with the following code.

```
print(studio_id)
```

- b. The result should be similar to: us-west-2:xxxx-xxxxxxxxxxxx.

Get the launch profile ID

To get the ID, you will use the `LAUNCH_PROFILE_NAME` that you specified previously in [Step 3: Create variables](#). Continue running these commands within the Python interactive shell.

1. Get a list of launch profiles in the studio

```
response = nimble_client.list_launch_profiles(  
    studioId=studio_id,  
)
```

2. Iterate through the list of launch profiles and return the one that matches the launch profile name.

```
lp_id = None  
  
for lp in response['launchProfiles']:  
    lp_name = lp['name']  
    lp_state = lp['state']  
    if lp_state != 'READY': continue  
    if lp_name != LAUNCH_PROFILE_NAME: continue  
    lp_id = lp['launchProfileId']
```

- a. To review the launchProfileId, print it with the following code.

```
print(lp_id)
```

- b. The result should be similar to: xxxxxxxxxxxx-x.

Get the streaming image ID

Using the IMAGE_ID variable, get the streamingImageId. Continue entering these commands within the Python interactive shell.

1. Get the streaming image IDs that are available for the launch profile by using these commands.

```
image_id = None  
  
response = nimble_client.get_launch_profile_details(  
    launchProfileId=lp_id,  
    studioId=studio_id  
)  
  
for image in response['streamingImages']:  
    if image['name'] != IMAGE_NAME: continue
```

```
image_id = image['streamingImageId']
```

2. Display the streamingImageId that matches the name.

```
print(f'{IMAGE_NAME}: {image_id}')
```

- This should return something similar to: NimbleStudioWindowsStreamImage: XX_XXXXXXXXXX.

Review all the parameters

- The `create_streaming_session` command requires five parameters. To review them, you can copy and paste the following code into your Python interactive shell.

```
print("""  
ownedBy: \t\t {user_id}  
launchProfileId: \t {lp_id}  
amazonImageId: \t\t {image_id}  
ec2InstanceType: \t {INSTANCE_TYPE}  
studioId: \t\t {studio_id}  
""")
```

Spin up a streaming session

To spin up a streaming session for your artist, run these commands into your Python interactive shell.

1. Use the `create_streaming_session` command to create the streaming session.

```
response = nimble_client.create_streaming_session(  
    ec2InstanceType=INSTANCE_TYPE,  
    launchProfileId=lp_id,  
    ownedBy=user_id,  
    streamingImageId=image_id,  
    studioId=studio_id,  
)
```

2. Review the status of the streaming session.

```
print(response['session'])
```

- This will return interesting fields, such as the **arn** for the streaming session and the **sessionId**.

3. Get the sessionId.

```
session_id = response['session']['sessionId']
```

4. Get the current state of the session.

```
response = nimble_client.get.streaming_session(  
    sessionId = session_id,  
    studioId=studio_id  
)  
  
state= response['session']['state']  
  
print(state)
```

- a. If the session is spinning up, the state will return CREATE_IN_PROGRESS.
 - b. After the session is ready for your artist to use, the state will return READY.
5. For more information about the data you can get about the client and various states, see [get_streaming_session](#) in the **Boto3 documentation**.
 6. View the streaming session in the Nimble Studio console.
 - a. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
 - b. Choose **Launch profiles** in the left navigation pane.
 - c. Choose **Streaming sessions** to review the streaming session that you have started up. You will also see the status of all other active and recent sessions.

Summary of code to create a streaming session

Here is a summary of the Python code to create a streaming session and return the Id/ID. These commands are supplied as reference. You can take them and incorporate them into a Python script to make these actions repeatable.

```
# Import modules
import boto3
from botocore.config import Config

# Set up variables
USER_NAME = 'Admin'
LAUNCH_PROFILE_NAME = 'Workstation-Default'
IMAGE_NAME = 'NimbleStudioWindowsStreamImage'
INSTANCE_TYPE = 'g4dn.4xlarge'
REGION_NAME = 'us-west-2'

# Create a Session
session = boto3.Session()

# Create a Directory Services client to return the domain name.
ds_client = session.client('ds', region_name= REGION_NAME)
response = ds_client.describe_directories()

# Get the directory that has the word 'nimble' in it
try:
    directory = [d for d in response['DirectoryDescriptions'] if '.nimble.' in
    d['Name']][0]
except IndexError: # if there is an error
    print('Could not find Nimble Studio Directory')

directory_id = directory['DirectoryId']
domain = directory['Name']

# Create a Identity Store client to get the id of the user
identity_client = session.client('identitystore', region_name= REGION_NAME)
response = identity_client.list_users(
    IdentityStoreId=directory_id,
    Filters=[
        {
            'AttributePath': 'UserName',
            'AttributeValue': f'{USER_NAME}@{domain}'
        },
    ],
)
user_id = response['Users'][0]['UserId']

# Create a Nimble Studio client to interact with Nimble Studio
nimble_client = session.client('nimble', region_name= REGION_NAME)
```

```
# List available studios for this account
response = nimble_client.list_studios()

# Get the studioId
studio_id = response['studios'][0]['studioId']

# Get the launch profiles available for the studio
response = nimble_client.list_launch_profiles(
    studioId=studio_id,
)

# Get the launch profile ID that matches the LAUNCH_PROFILE_NAME variable
lp_id = None

for lp in response['launchProfiles']:
    lp_name = lp['name']
    lp_state = lp['state']
    if lp_state != 'READY': continue
    if lp_name != LAUNCH_PROFILE_NAME: continue
    lp_id = lp['launchProfileId']

# Get the streaming image ids that are available for that launch profile
response = nimble_client.get_launch_profile_details(
    launchProfileId=lp_id,
    studioId=studio_id
)

image_id = None

for image in response['streamingImages']:
    if image['name'] != IMAGE_NAME: continue
    image_id = image['streamingImageId']


# Create the streaming session
response = nimble_client.create_streaming_session(
    ec2InstanceType=INSTANCE_TYPE,
    launchProfileId=lp_id,
    ownedBy=user_id,
    streamingImageId=image_id,
    studioId=studio_id,
)
```

```
# Get the streaming session id.  
session_id = response['session']['sessionId']  
  
print(f'Streaming Session Id: {session_id}')
```

Step 8: Get sessionId for a streaming session

To get the status of a session, or delete a session for a particular user, you will need to know the studioId and the userId. For more information see [list_streaming_session](#) in the **Boto3 documentation**.

```
# This code assumes you have:  
#   nimble_client: An existing boto3 session client  
#   studio_id:      The id of the studio  
#   user_id:        The id of the session you want to terminate.  
  
# Get the existing streaming sessions for the user  
response = nimble_client.list_streaming_sessions(  
    studioId=studio_id,  
    ownedBy=user_id,  
)  
  
# Get the sessionId  
session_id = response['sessions'][0]['sessionId']  
  
# Print the results  
print(f"""  
Studio id: \t{studio_id}  
Session id: \t{session_id}  
""")
```

Step 9: Get the current state of streaming session

To get the status of a session, you will need to know the studioId and the sessionId. For more information, see [get_streaming_session](#) in the **Boto3 documentation**.

```
# This code assumes you have:  
#   nimble_client: An existing boto3 session client  
#   studio_id:      The id of the studio  
#   session_id:     The id of the session you want to get info about.
```

```
response = nimble_client.get_streaming_session(  
    sessionId = session_id,  
    studioId = studio_id,  
)  
  
state = response['session']['state']  
createdAt = response['session']['createdAt']  
  
# Print the results  
print(f"""\nState: \t\t{state}\nDate Created: \t{createdAt}\n""")
```

Step 10: Terminate a streaming session

To terminate a streaming session, you will need the `studioId` and the `sessionId`. For more information, see [delete_streaming_session](#) in the **Boto3 documentation**.

```
# This code assumes you have:  
#   nimble_client: An existing boto3 session client  
#   studio_id:      The id of the studio  
#   session_id:     The id of the session you want to terminate.  
  
# Delete the streaming session  
response = nimble_client.delete_streaming_session(  
    sessionId=session_id,  
    studioId=studio_id  
)  
  
state= response['session']['state']  
  
# Print the results  
print(f"""\nSession id: \t{session_id}\nState: \t\t{state}\n""")
```

You have now created, retrieved the status of, and terminated a streaming session using the Nimble Studio API.

Troubleshooting

"Command not found" import error with boto3

Error: bash: import: command not found

This error occurs when you aren't importing boto3 from within the Python interactive shell.

To enter the interactive shell, use the following python3 command.

```
[cloudshell-user@ip-10-1-23-456 ~]$ *python3*
Python 3.7.10 (default, Jun 3 2021, 00:02:01)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-13)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

ImportError with boto3 in Python

Error:

```
`Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named boto3`
```

This error occurs when you try to import boto3 from the Python interactive shell.

You might get the ImportError if you started Python 2.7+ instead of Python 3.7+. To determine which Python version you're using, review the Python version number that is displayed after the command you used to launch Python. In the following example, the Python version **2.7.18** is being used.

```
*Python 2.7.18* (default, Jun 10 2021, 00:11:02)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-13)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

To start the Python interactive shell with the correct version of Python, use the command python3 instead of python. The following example shows where the version can be found.

```
[cloudshell-user@ip-10-1-23-456 ~]$ *python3*
Python 3.7.10 (default, Jun 3 2021, 00:02:01)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-13)] on linux
```

Type "help", "copyright", "credits" or "license" for more information.

>>>

Related Resources

- [Nimble Studio API reference guide](#)
- [Boto3 documentation](#)
- [DirectoryService API](#)
- [IdentityStore API documentation](#)
- [Nimble Studio API documentation](#)

Provisioning workstations in multiple Availability Zones or Local Zones

By default, when you create a new studio, your workstations are created in one Availability Zone. With Nimble Studio launch profiles, you can provision workstations in multiple Availability Zones or Local Zones in the Region that your studio is deployed in. One reason for creating a different Local Zone or AZ is because your default AZ is low on a certain type of instance. Another benefit is to decrease latency by provisioning a workstation that's located closer to your artists.

This tutorial explains how to provision a workstation in a different Availability Zone or Local Zone by creating new launch profiles for them.

Note

If you're using this tutorial to create launch profiles in different [Local Zones](#): When you're asked to choose a different Availability Zone, choose the Local Zone that you want to work in.

Contents

- [Prerequisites](#)
- [Step 1: Create a workstation subnet in the new Availability Zone or Local Zone](#)
- [Step 2: Create a new file systems subnet in the new Availability Zone](#)
- [Step 3: Update NACLs](#)

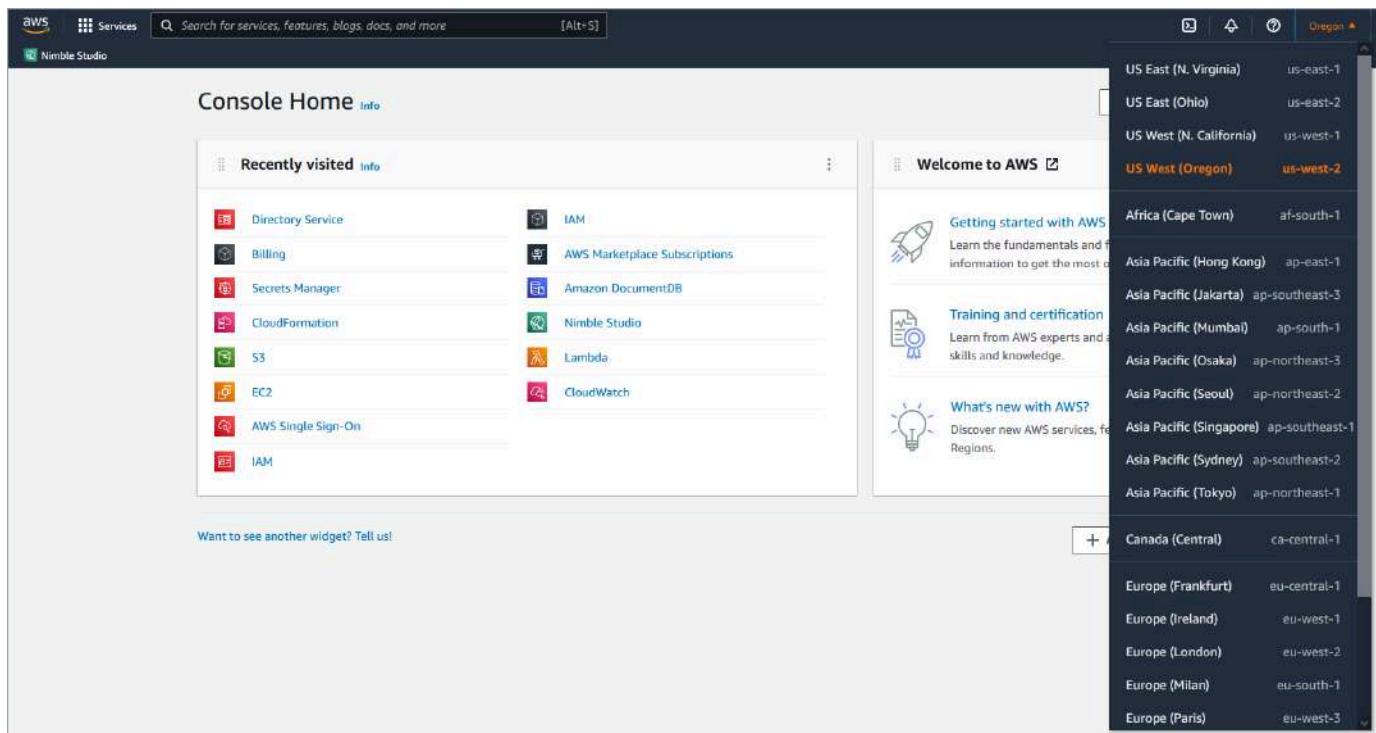
- [Step 4: Create a new FSx drive for Userprofiles for Windows](#)
- [Step 5: Add the Amazon FSx drive as a storage component](#)
- [Step 6: Create a new launch profile](#)
- [Step 7: Update launch profile](#)
- [Step 8: \(Optional\) Verify launch profile configuration](#)

Prerequisites

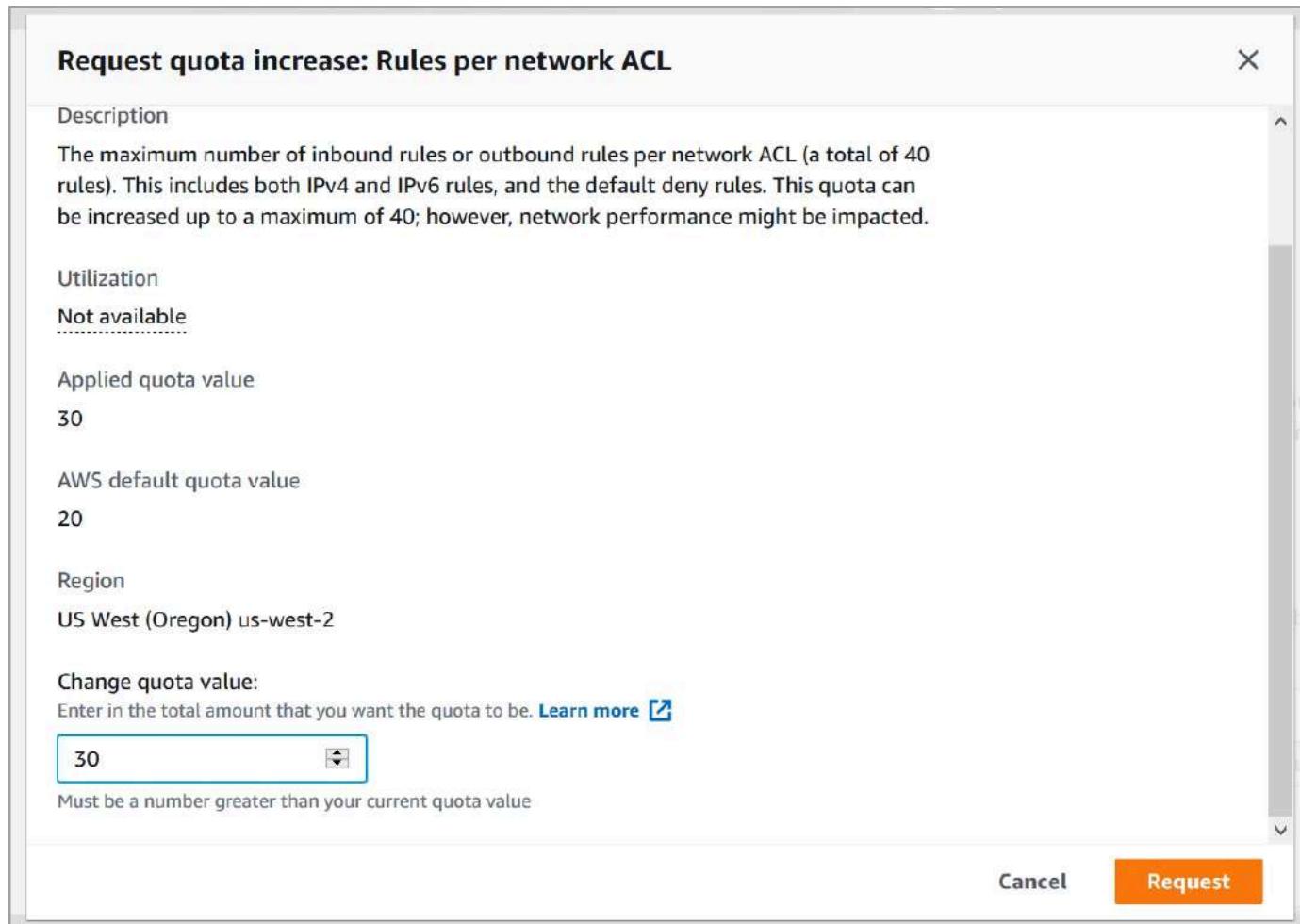
- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- You also need to complete the [Adding studio users](#) tutorial.
- To launch a workstation in a different Local Zone, enable that Local Zone by following the [\(Optional\) Opt in to the LA Local Zone](#) tutorial.
- The quota for **Rules per network ACL** of Amazon Virtual Private Cloud (Amazon VPC) must be greater than 30. If it isn't, request a quota increase by following these instructions:

Request a quota increase for Amazon Nimble Studio streaming sessions

1. Sign in to the AWS Management Console and open the [Service Quotas](#) console.
2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



3. In the left navigation pane choose **AWS services**.
4. In the search bar, enter **VPC**.
5. In Services, select **Amazon Virtual Private Cloud (Amazon VPC)** from the list.
6. Choose **Rules per network ACL** from the list.
7. Choose **Request quota increase**.
8. The **Request quota increase** window appears. In the **Change quota value** field, enter a new quota value.
 - The default quota value is 20. Your new quota value must be at least 30.



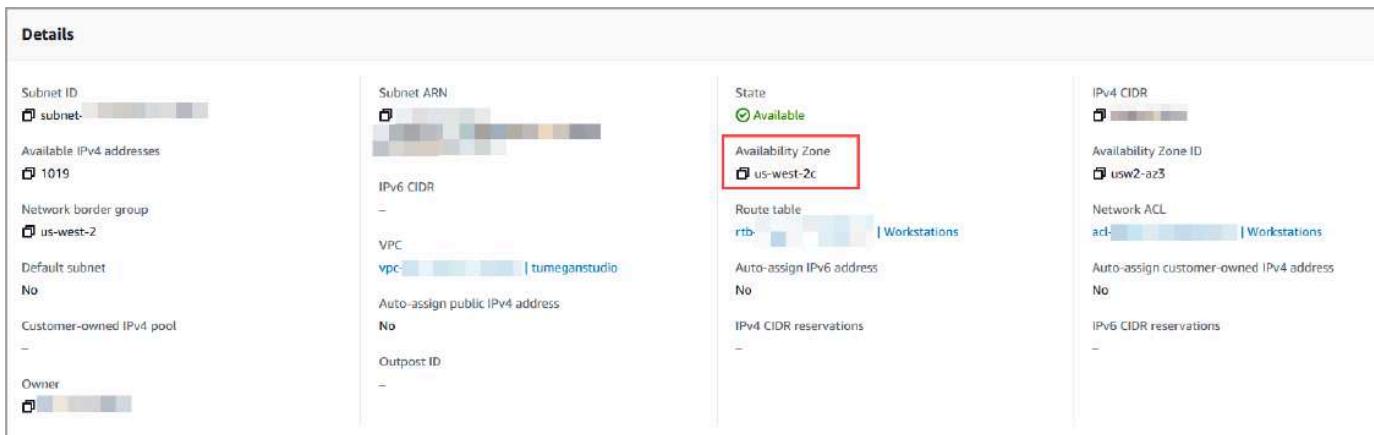
9. Choose **Request**.
10. You can check on the status of a quota request by choosing **Quota request history** in the navigation pane of the **Service Quotas** console. It can take 12–48 hours for a request to be resolved.

Step 1: Create a workstation subnet in the new Availability Zone or Local Zone

First, create a new workstation subnet for the new Availability Zone or Local Zone. This subnet must be in an Availability Zone that is different from the one that your studio is in.

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Subnets** from the left navigation pane.

3. Select the Workstations subnet.
4. Look at the **Availability Zone** in the **Details** section. When you create another subnet in *step 6c* of this section, choose a different Availability Zone.



5. Select **Create subnet**.
6. Provide the following information in the specified fields.
 - a. **VPC:** Choose the VPC for your studio. For example: <your-studio-name>
 - b. **Subnet name:** Enter **Workstations2**
 - c. **Availability Zone:** Choose something other than the subnet that you found in *step 4*.
 - Notice which Availability Zone or Local Zone that you choose. You will need this to create a file systems subnet in [Step 3: Update NACLs](#).
 - d. **IPv4 CIDR block:** Enter the /22 CIDR block. For example, 10.0.48.0/22
7. Select **Create subnet**.
8. Select the new subnet.
9. Choose the **Route table** tab.
10. Choose **Edit route table association**.
11. For **Route table ID**, select the Workstations subnet. The new subnet uses the same route table as the **Workstations** subnet.
12. Choose **Save**.
13. Choose the **Network ACL** tab.
14. Choose **Edit network ACL association**.
15. For **Network ACL ID**, select the Workstations subnet.
 - This means it has the same route table as the subnet in the original Availability Zone.

16. Choose **Save**.
17. Attach the workstations route table to the new subnet.
18. Attach the workstations NACL to the new subnet.

Step 2: Create a new file systems subnet in the new Availability Zone

Next, you'll create a new subnet for file systems. This sets you up for [Step 5: Add the Amazon FSx drive as a storage component](#), where you'll create a new Amazon FSx file system to store user profiles for this new subnet's workstations.

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Create subnet**.
3. Provide the following information in the specified fields.
 - a. **VPC:** Choose the VPC for your studio. For example: <your-studio-name>
 - b. **Subnet name:** Enter **Filesystems2**
 - c. **Availability Zone:** Choose something other than the subnet that you found in step 4.
 - Notice which Availability Zone or Local Zone that you choose. You will need this to create a file systems subnet in [Step 3: Update NACLs](#).
 - d. **IPv4 CIDR block:** Enter the /25 CIDR block. For example, **10.0.48.0/25**
4. Select **Create subnet**.
5. Select the new subnet.
6. Choose the **Route table** tab.
7. Choose **Edit route table association**.
8. For **Route table ID**, select the Filesystems subnet. The new subnet will use the same route table as workstations subnet.
9. Choose **Save**.
10. Choose the **Network ACL** tab.
11. Choose **Edit network ACL association**.
12. For **Network ACL ID**, select the **Workstations** subnet.
 - This means it has the same route table as the subnet in the original Availability Zone.
13. Choose **Save**.

Step 3: Update NACLs

Next, update all of the NACLs that are affiliated with the old Availability Zone by adding duplicate rules for the new Availability Zone or Local Zone. It's easiest to do this in two steps. First, look through your NACLs to see which have **Rules** with **Source** set to the CIDR block of the workstations subnet (10.0.40.0/22). Duplicate every rule with that **Source**. Then, look through your NACLs to see which have **Rules** with **Source** set to the CIDR block of the file systems subnet (10.0.2.128/25). Duplicate every rule with that **Source**.

To create duplicate rules for the workstations subnet

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Network ACLs** from the left navigation pane.
3. Sort by **Default**.
 - Ignore all subnets that are marked **Yes** for **Default**.
4. Select a network ACL.
5. Select the **Inbound rules** tab.
6. Check to see if it has a **Rule** with **Source** set to the CIDR block of your workstations subnet (10.0.40.0/22).
 - a. If it has a **Rule** with **Source** 10.0.40.0/22, proceed with the following steps.
 - b. If it doesn't have a **Rule** with **Source** 10.0.40.0/22, select a different network ACL and check if it has a rule with source 10.0.40.0/22.
7. Choose **Edit inbound rules**.
8. Select **Add new rule**. Provide the following information in the specified fields.
 - a. **Rule number**: To make it easier to keep track of the duplicate rules, give all of the new rules the same first two digits, and match the last three digits of the rule it was based on.
 - Example: If the original rule number was 201, give the new rule the number 21201.
 - b. **Type**: Give this new rule the same type as the rule you found in step 6a.
 - c. **Protocol**: Give this new rule the same type as the rule you found in step 6a.
 - d. **Port range**: Give this new rule the same type as the rule you found in step 6a.
 - e. **Source**: Enter **10.0.48.0/22**
 - f. **Allow/Deny**: Give this new rule the same type as the rule you found in step 6a.

- g. Create new rules for every rule with Source 10.0.40.0/22.

For every rule in the 200s, the corresponding rule in the 21000s has the same Type, Protocol, and Port range. Every rule in the 200s has a source of 10.0.40.0/22 while every rule in the 21000s has a source of 10.0.48.0/22.

200	SMB (445)	TCP (6)	445	10.0.40.0/22	Allow	Remove
201	WinRM-HTTP (5985)	TCP (6)	5985	10.0.40.0/22	Allow	Remove
202	Custom TCP	TCP (6)	988	10.0.40.0/22	Allow	Remove
203	Custom TCP	TCP (6)	1021 - 1023	10.0.40.0/22	Allow	Remove
21200	SMB (445)	TCP (6)	445	10.0.48.0/22	Allow	Remove
21201	WinRM-HTTP (5985)	TCP (6)	5985	10.0.48.0/22	Allow	Remove
21202	Custom TCP	TCP (6)	988	10.0.48.0/22	Allow	Remove
21203	Custom TCP	TCP (6)	1021 - 1023	10.0.48.0/22	Allow	Remove

9. Select **Save changes**.

10. Select the **Outbound rules** tab.

11. Check to see if it has a **Rule with Source** set to the CIDR block of your workstations subnet (10.0.40.0/22).

- a. If it has a **Rule with Source** 10.0.40.0/22, proceed to the following steps.
- b. If it doesn't have a **Rule with Source** 10.0.40.0/22, select a different network ACL and check if it has a rule with source 10.0.40.0/22.

12. Choose **Edit Outbound rules**.

13. Repeat steps 8–10.

Do this for all of your NACLs.

To create duplicate rules for the file systems subnet

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Network ACLs** from the left navigation pane.
3. Sort by **Default**.
 - Ignore all subnets that are marked **Yes** for **Default**.
4. Select a network ACL.

5. Select the **Inbound rules** tab.
6. Check to see if it has a **Rule** with **Source** set to the CIDR block of your workstations subnet (10.0.40.0/25).
 - a. If it has a **Rule** with **Source** 10.0.40.0/25, proceed with the following steps.
 - b. If it doesn't have a **Rule** with **Source** 10.0.40.0/25, select a different network ACL and check if it has a rule with source 10.0.40.0/25.
7. Choose **Edit inbound rules**.
8. Select **Add new rule**. Provide the following information in the specified fields.
 - a. **Rule number:** To make it easier to keep track of the duplicate rules, give all of the new rules the same first two digits, and match the last three digits of the rule it was based on.
 - Example: If the original rule number was 201, give the new rule the number 21201.
 - b. **Type:** Give this new rule the same type as the rule you found in *step 6a*.
 - c. **Protocol:** Give this new rule the same type as the rule you found in *step 6a*.
 - d. **Port range:** Give this new rule the same type as the rule you found in *step 6a*.
 - e. **Source:** Enter **10.0.48.0/25**
 - f. **Allow/Deny:** Give this new rule the same type as the rule you found in *step 6a*.
 - g. Create new rules for every rule with Source 10.0.40.0/25.

200	SMB (445)	TCP (6)	445	10.0.40.0/22	Allow	<input type="button" value="Remove"/>
201	WinRM-HTTP (5985)	TCP (6)	5985	10.0.40.0/22	Allow	<input type="button" value="Remove"/>
202	Custom TCP	TCP (6)	988	10.0.40.0/22	Allow	<input type="button" value="Remove"/>
203	Custom TCP	TCP (6)	1021 - 1023	10.0.40.0/22	Allow	<input type="button" value="Remove"/>
21200	SMB (445)	TCP (6)	445	10.0.48.0/22	Allow	<input type="button" value="Remove"/>
21201	WinRM-HTTP (5985)	TCP (6)	5985	10.0.48.0/22	Allow	<input type="button" value="Remove"/>
21202	Custom TCP	TCP (6)	988	10.0.48.0/22	Allow	<input type="button" value="Remove"/>
21203	Custom TCP	TCP (6)	1021 - 1023	10.0.48.0/22	Allow	<input type="button" value="Remove"/>

9. Select **Save changes**.
10. Select the **Outbound rules** tab.
11. Check to see if it has a **Rule** with **Source** set to the CIDR block of your workstations subnet (10.0.40.0/25).
 - a. If it has a **Rule** with **Source** 10.0.40.0/25, proceed to the following steps.

- b. If it doesn't have a **Rule with Source** 10.0.40.0/25, select a different network ACL and check if it has a rule with source 10.0.40.0/25.
12. Choose **Edit Outbound rules**.
13. Repeat steps 8–10.

Do this for all of your NACLs.

Step 4: Create a new FSx drive for Userprofiles for Windows

1. Sign in to the AWS Management Console and open the [Amazon FSx](#) console.
2. Choose **Create file system**.
3. Choose **Amazon FSx for Windows File Server**.
4. Choose **Next**.
5. Provide the following information in the specified fields.
 - a. **Name:** Enter **UserProfiles**
 - b. **Deployment type:** Leave **Multi-AZ** selected.
 - c. **Storage type:** Select **SSD**
 - d. **Storage capacity:** Enter the value you chose for the Amazon FSx file system when you ran StudioBuilder. By default, this value is 200 GiB.
 - e. ***VPC:** Choose your studio's VPC. Example: <your-studio-name>
 - f. **VPC Security Group:** Select the security group with **FSxFileSystems** in the name.
 - Delete the default security group.
 - g. **Preferred subnet:** Choose **Filesystems**.
 - h. **Standby subnet:** Choose **Filesystems2**.
 - i. **Windows authentication:** Choose **AWS Directory Service for Microsoft Active Directory**.
 - Choose the **Nimble Studio Active Directory**. This is the AWS Directory Service for Microsoft Active Directory that contains your studio name.
 - j. **Encryption key:** Choose the Nimble Studio encryption key. Example: <your-studio-name>-Key.
6. Choose **Next**.

7. Review your values and choose **Create file system**.

Wait until the status of the file system changes from **Creating** to **Available** before proceeding to the next step.

Step 5: Add the Amazon FSx drive as a storage component

Next, add the new Amazon FSx drive as a Nimble Studio storage component.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. Choose **Add in File Storage**.
4. Provide the following information in the specified fields.
 - a. **Region:** Select the AWS Region that your studio is deployed in.
 - b. **Name:** Enter **WindowsUserProfiles**
 - c. (Optional) **File storage description:** Enter **FSx file system for new AZ workstations**.
 - d. **Storage type:** Select **Amazon FSx for Windows File Server**
 - e. **Available AWS file systems:** Choose the new drive that you created in [Step 4: Create a new FSx drive for Userprofiles for Windows](#).
 - f. **Linux mount point:** Enter **/mnt/fsxshare**
 - Use the same mount point for your other Amazon FSx file system for the workstations in the other Availability Zone.
 - g. **Windows mount drive:** Enter **Z**
 - Use the same mount point for your other Amazon FSx file system for the workstations in the other Availability Zone.
5. Read the terms and conditions and if you agree:
 - Select the check box next to **I understand that Nimble Studio will access my existing file storage**.
6. Choose **Save connection parameters**.

Step 6: Create a new launch profile

Next, create a new launch profile with the new Amazon FSx storage that you created in [Step 5: Add the Amazon FSx drive as a storage component](#). Base this new launch profile on the default launch profile.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the **Workstation-Default** launch profile.
4. Choose **Actions**. Then choose **Copy to new**.
5. Change the **Name**. Example: Workstation-Default-newAZ.
6. (Optional) Add a description. Example: Copy of Workstation-Default, but connected to storage in the new Availability Zone.
7. In **Studio file storage components**, clear **FSxWindows** and check **WindowsUserProfiles** in the **Launch profile components** section.
 - The order that you select these subnets matters: Select **Workstations** first, and then **Workstations2** second.
8. In the **Subnets** section, select **Workstations** and then select **Workstations2**.
9. Choose **Create launch profile**.

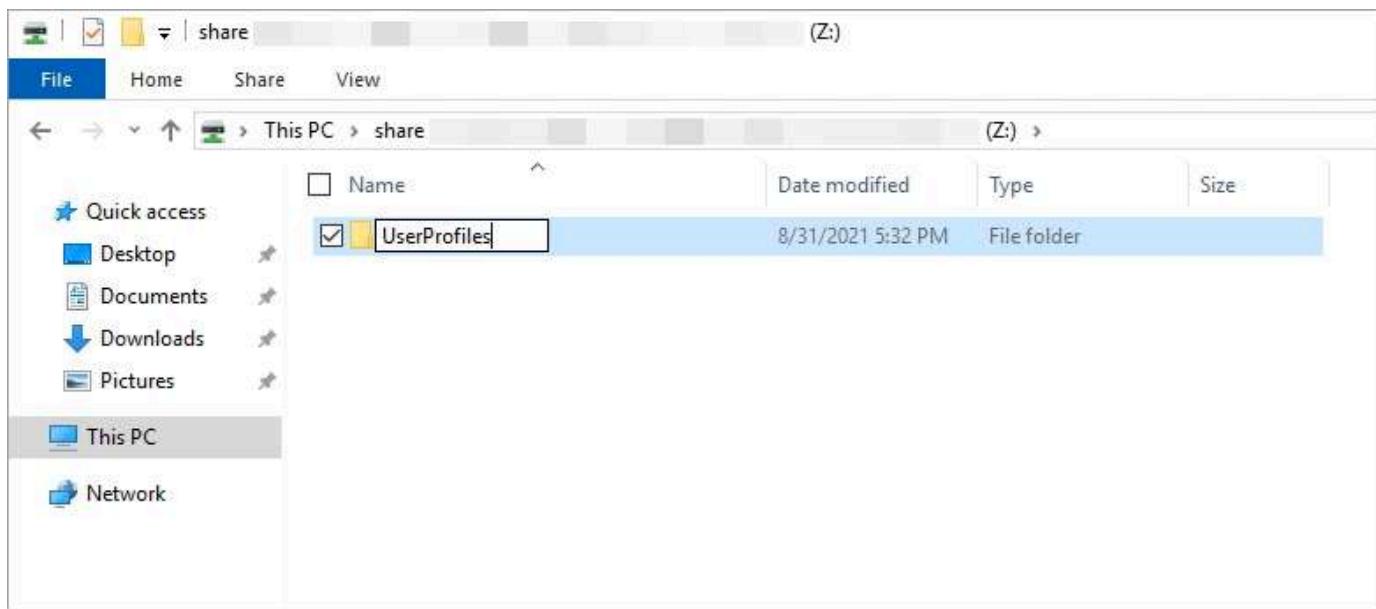
Wait until the **Status** of the launch profile changes from **Creating** to **Ready** before proceeding to the next step.

Step 7: Update launch profile

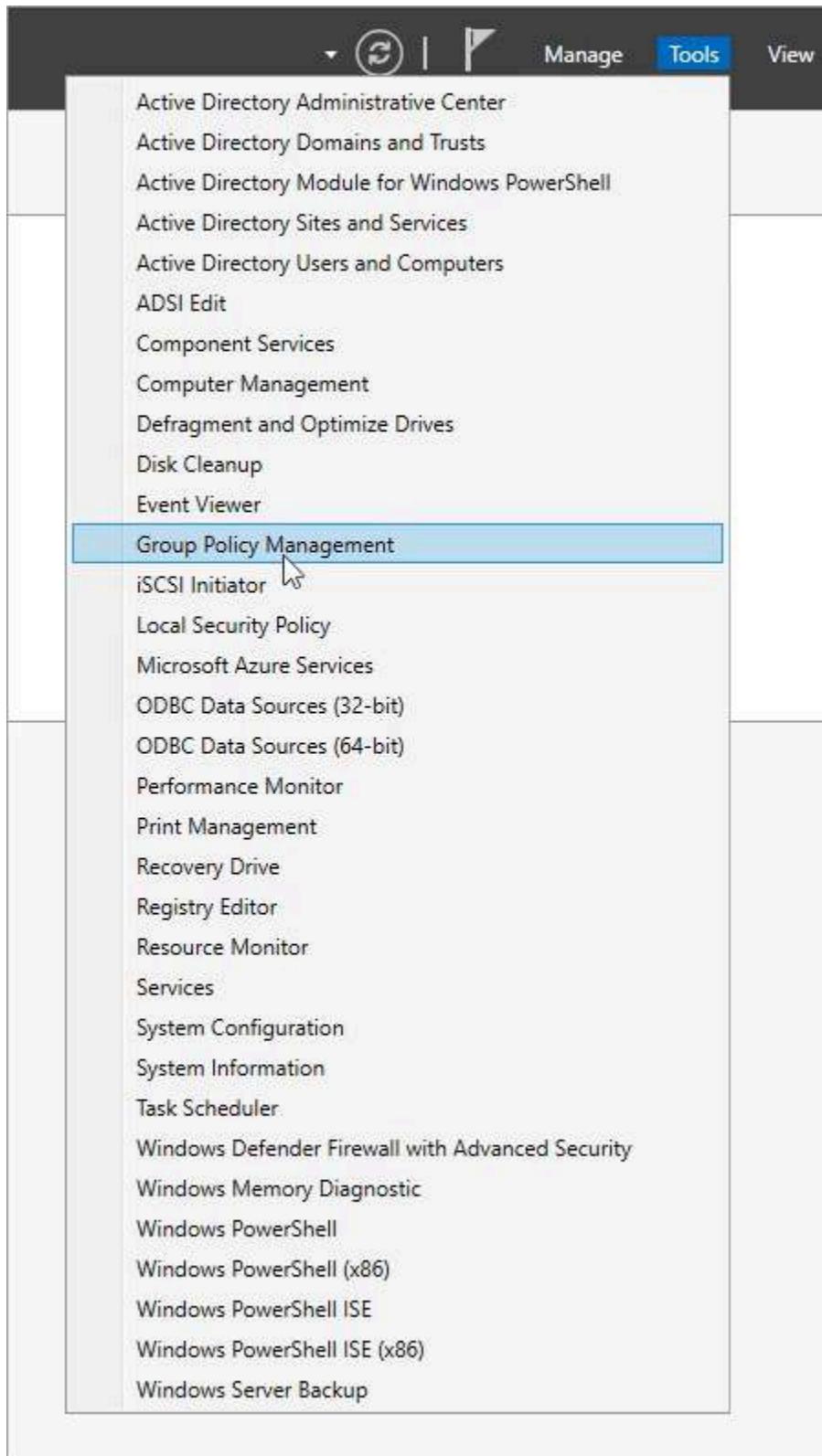
Launch a new Windows instance with the new launch profile as an administrator user. Then you can make some adjustments before granting other users access to this launch profile.

1. Sign in to the AWS Management Console and open the [Amazon FSx](#) console.
2. Select the new file system.
3. In the **Network & security** section, notice the first section of the DNS name. You will need this information for *step 19* of this section.

- If the DNS name is <random-string-of-characters>.your-studio-name.nimble.<studio-availability-zone>.aws, notice the <random-string-of-characters>.
4. Sign in to the Nimble Studio portal by following the instructions in [Logging in to the Nimble Studio portal](#).
 5. Launch the virtual workstation that you created in [Step 6: Create a new launch profile](#) by following the instructions in [Launching a virtual workstation](#).
 6. Open **File explorer**.
 7. Go to **This PC**.
 8. Double-click the Amazon FSx file system named **share**.
 9. Create a new folder and name it **UserProfiles**.



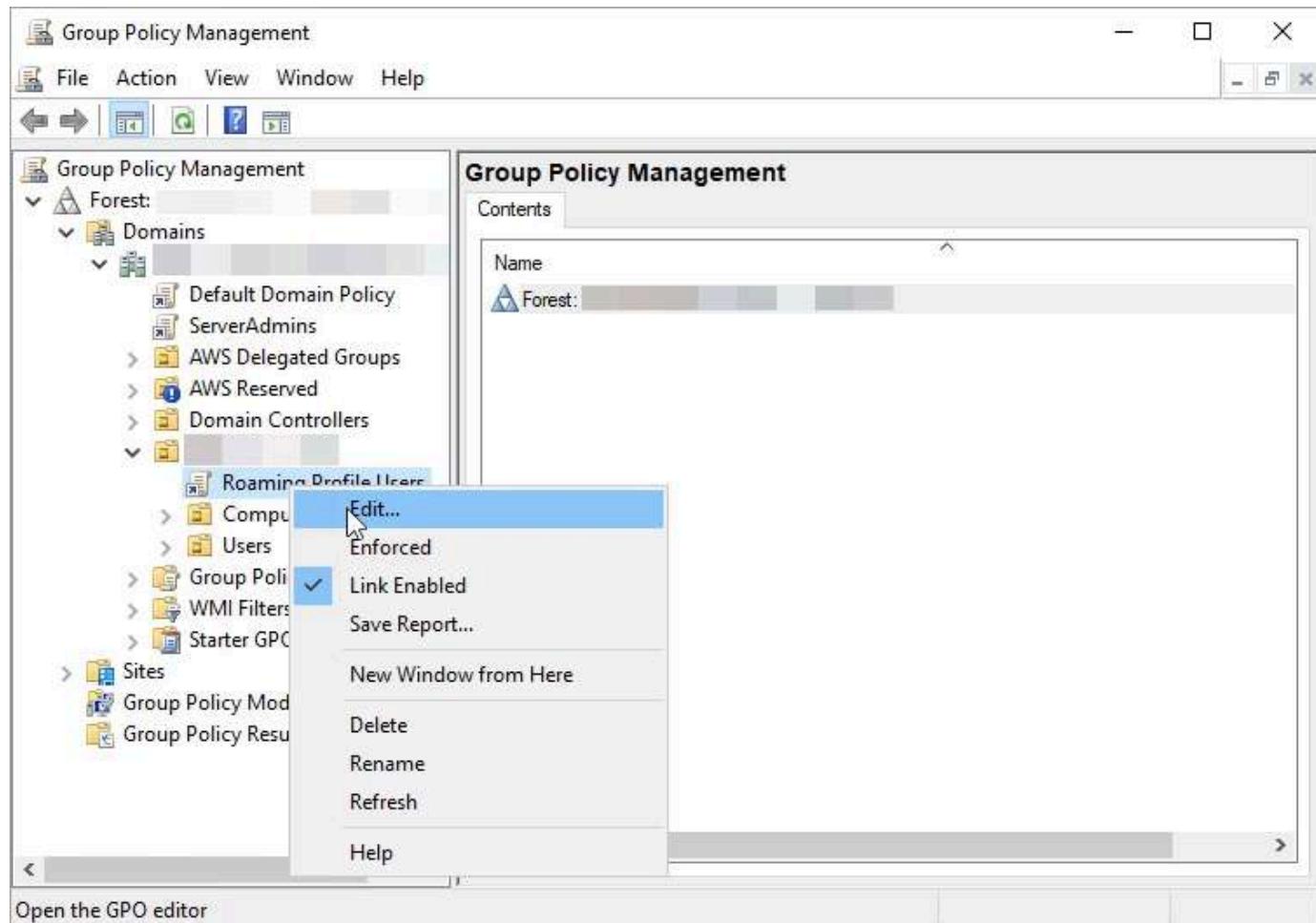
10. Choose the **Start Menu**.
11. Enter **Server Manager** and select the **Server Manager** desktop application from the top of the search results.
12. Select **Tools** from the top-right corner of the window. Then select **Group Policy Management**.



13. Open the dropdown menu named **Forest:<your-studio-name>**.
14. Select **Domain**. Then select **<your-studio-domain>** and **<your-studio-name>**.

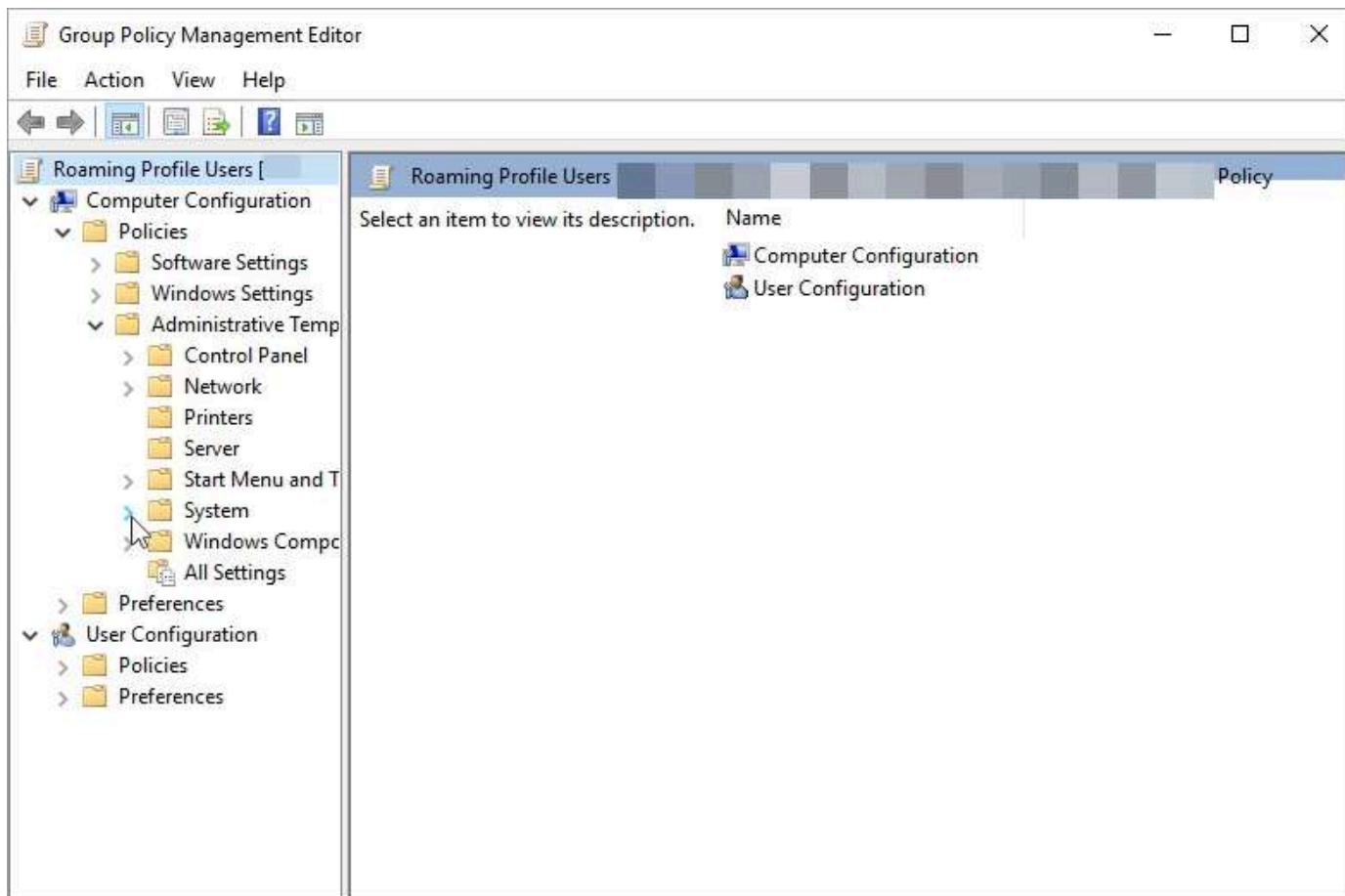
15. Right-click on **Roaming Profile Users** and choose **Edit**.

The following image shows the **Group Policy Management** window. The menu is extended for the **Roaming Profile Users** file, and shows the **Edit** option.

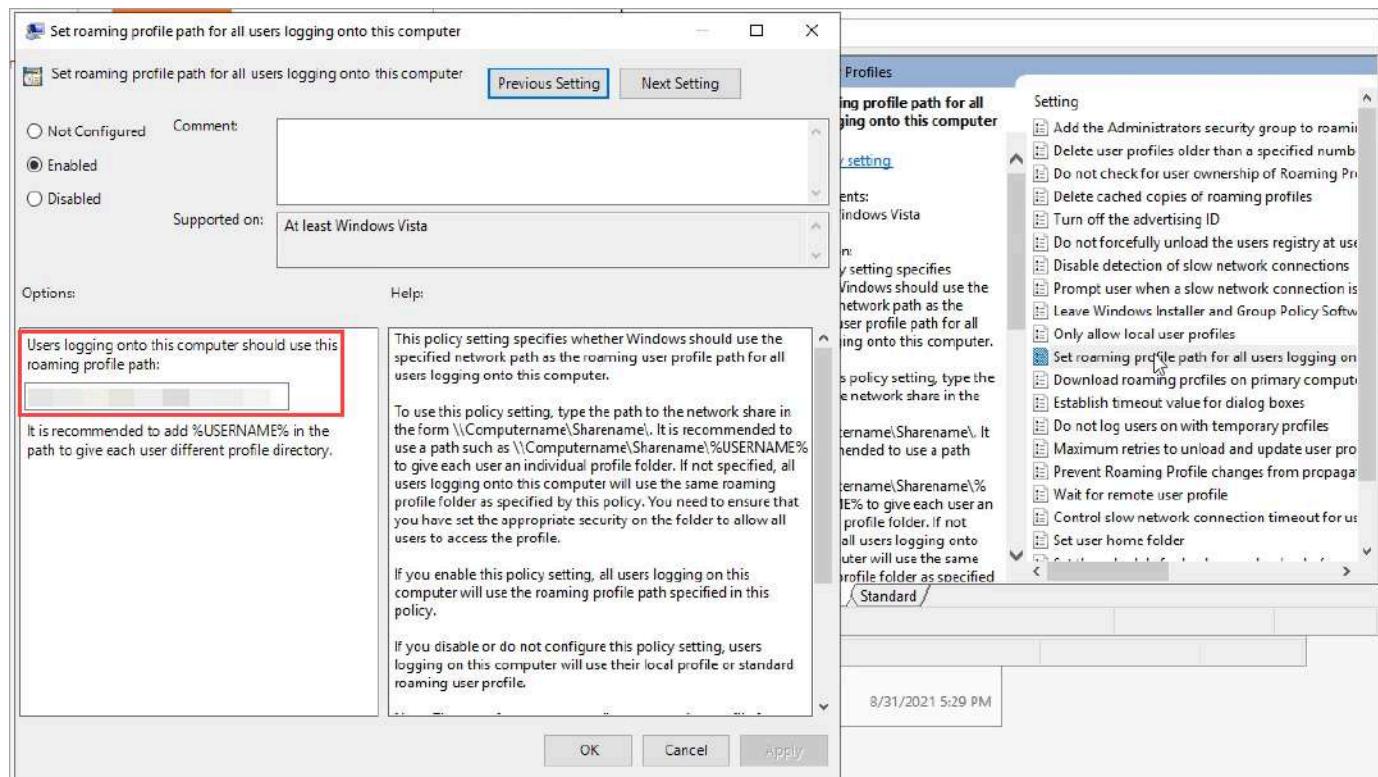


16. In **Computer Configuration**, open **Policies**, **Administrative Templates**, and **System**.

The following image shows the **Group Policy Management** editor window with the **System** file. The **Computer Configuration**, **Policies**, and **Administrative Templates** files are open.



17. Choose **User Profiles**.
18. Double-click the entry named **Set roaming profile path for all users logging on to this computer**.
19. Change the location of the roaming file path to the ID of the new Amazon FSx file system that you found in *step 3* of this section.
 - Replace the portion of the current roaming profile path. It looks something like `fs-<string-of-letters-and-numbers>` with the string you found in *step 3* of this section.



20. Select **OK**.

21. Open a **Command Prompt**.

22. Run the following command to update the policy: `gpupdate /force`.

- If the command succeeds, it will output Computer Policy update has completed successfully.

23. Shut down your virtual workstation and proceed to the following section, Step 8.

Step 8: (Optional) Verify launch profile configuration

We recommend that you verify the correct configuration of this launch profile. This saves your artists time from troubleshooting setup issues.

1. Sign in to the Nimble Studio portal by following the instructions in [Logging in to the Nimble Studio portal](#).
2. Launch the virtual workstation that you created in [Step 6: Create a new launch profile](#) by following the instructions in [Launching a virtual workstation](#).
3. Open **File explorer**.

4. Go to **This PC**.
5. Double-click the Amazon FSx file system named **share**.
6. Double-click the **UserProfiles** file.
7. There should be a folder named after your artist.

You've successfully created new launch profiles to provision workstations in multiple Availability Zones (AZ) and Local Zones.

Provide administrator access for Windows users

Providing secure user access is a key aspect of Amazon Nimble Studio. This tutorial shows how to provide administrator access to Windows users, so that they can install software. One way to do this is to use AWS Directory Service for Microsoft Active Directory's Add-LocalGroupMember command to provide local administrator access for a particular user.

 **Important**

Any user with administrator access will have full read and write access to all files and folders, including the files in the mounted shared file systems, that are attached to the system. Example: FSx for Windows File Server

This guide shows how to provide **administrator** capability in three steps. To begin, you'll create a specific component that you can attach to a launch profile. This component provides administrator capability only to users that you deem appropriate. By granting those users administrator access, you give them permission to install software and perform root level operations.

However, be aware of security risks when providing administrator access. Users with administrator permissions can access all files and folders in the studio. For information about security recommendations, see [Security best practices in AWS Identity and Access Management](#) in the IAM User Guide.

Contents

- [Prerequisites](#)
- [Step 1: Add an administrator access component to your studio](#)
- [Step 2: Add the administrator Access component to a launch profile](#)

- [Step 3: Test administrator Access component](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.

Step 1: Add an administrator access component to your studio

In this step, you will use the custom configuration component to create a system initialization script that enables administrator use.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. Choose **Add** in the **Custom configuration** studio resource type.
4. In the **Custom configuration info** section, complete the fields as follows:
 - a. For **Region**, Select the AWS Region that your studio is deployed in.
 - b. Choose the name for your component. Example: **Administrator Access**.
 - c. (Optional) Give your component a description. Example: **Adds local administrator access to a Windows workstation using Active Directory**.

The screenshot shows the 'Custom configuration info' form. It includes fields for Region Info (set to US West (Oregon) us-west-1 (HOME)), Custom configuration name (set to 'my custom configuration'), and Custom configuration description - optional (set to 'description for my custom configuration').

Custom configuration info	
Region Info Select the Region that this custom configuration will be associated with. Defaults to the home Region. US West (Oregon) us-west-1 (HOME)	
Custom configuration name Set the name for the custom configuration that you are registering. my custom configuration	
Custom configuration description - optional Enter an optional description of this custom configuration. description for my custom configuration	

5. Create a new **Script parameter** by choosing **Add new parameter**.
6. In the **Parameter name** section, enter **USERLIST**.

7. In the **Parameter value** section, enter user names to grant administrator access to specific users. User names should be separated by only a space. Example: **Administrator maria richard**

Parameter name	Parameter value - optional
USERLIST	Admin maria richard

Add new parameter

8. In the **Initialization script** section, go to the **Windows** section. To give specific users administrator access, add the following code to the system initialization section.

```
foreach ($userName in $USERLIST.split()) {  
    Add-LocalGroupMember -Group "Administrators" -Member "$userName"  
}
```

9. In the **Security groups** section, choose the **LicenseServer** security group.
10. (Optional) Add tags if you're using tags to track your AWS resources.
11. Choose **Save custom configuration**.

Step 2: Add the administrator Access component to a launch profile

These steps explain how to attach the **Administrator Access** component to an existing launch profile. Before you begin, we recommend that you create a specific launch profile that's dedicated to Windows administrative work; this makes it clear who has access to this component. To create a specific launch profile, follow the instructions in [Creating launch profiles](#) before returning to this step.

1. Choose **Launch profiles** in the left navigation pane.
2. Select a launch profile by selecting the dot to the left of its name.
3. Choose **Action**. Then choose **Edit**.
4. Navigate to the **Launch profile components** section.
5. Select the check box next to the **Administrator Access** component that you just created.
6. Choose **Update launch profile**.

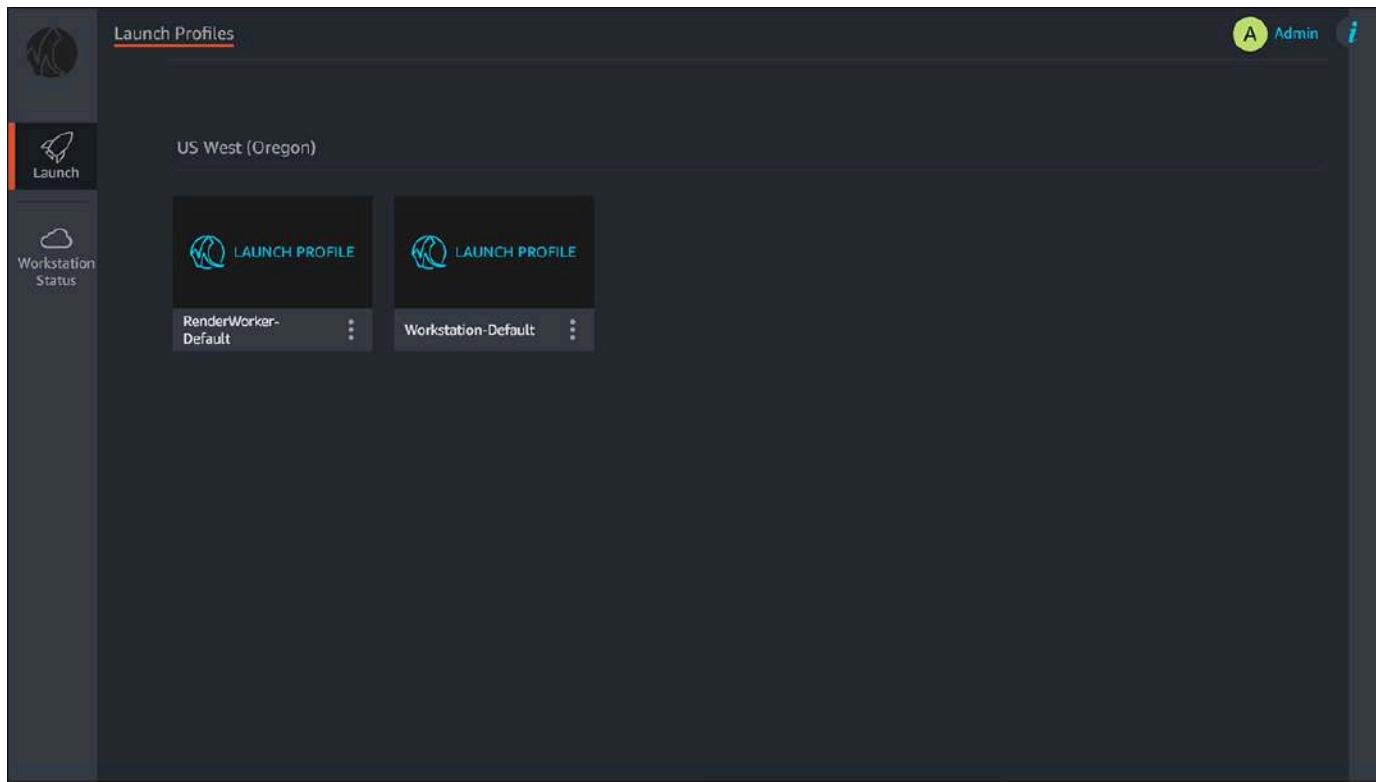
7. Repeat these steps for all launch profiles that you want to have access to the **Administrator Access** component.

Step 3: Test administrator Access component

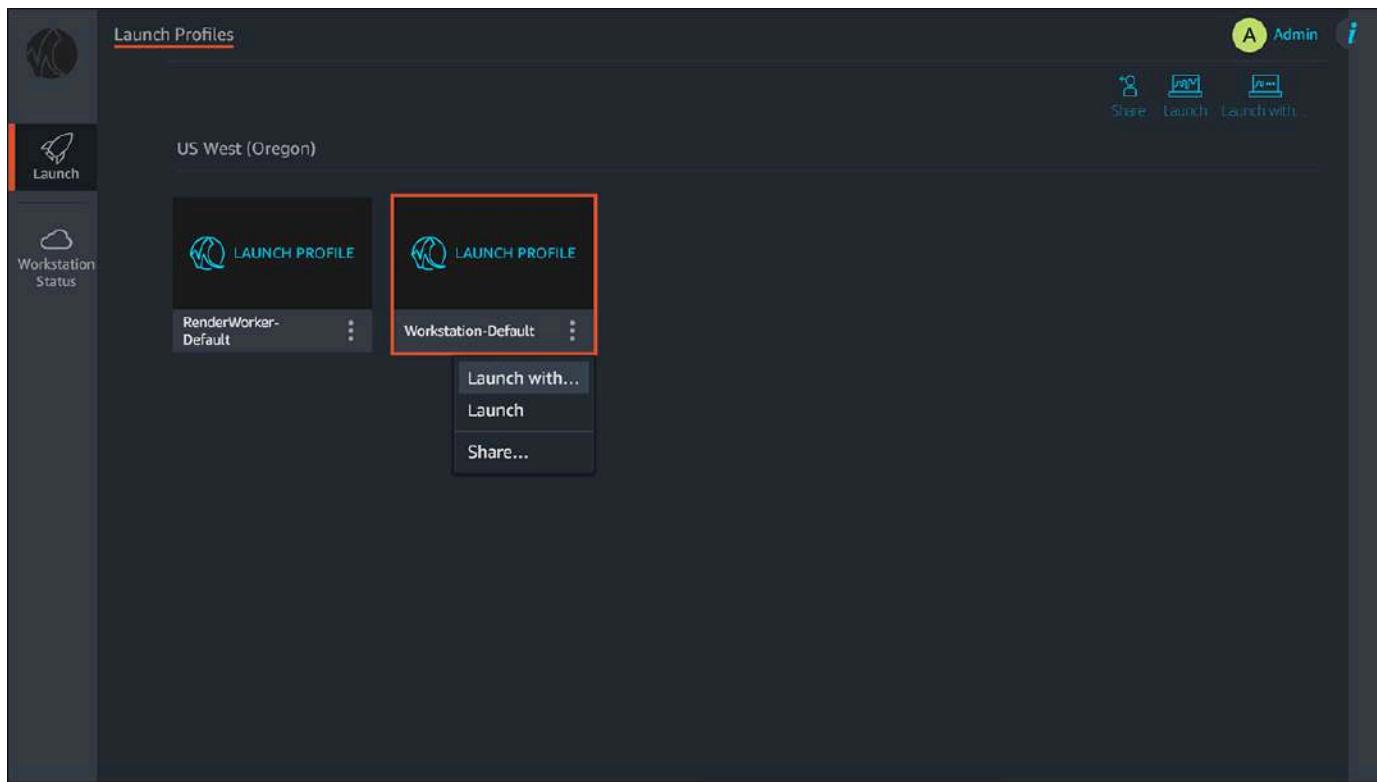
To check that the **administrator Access** component is working as expected, sign in to a virtual workstation and test the process. Choose the launch profile that you added the **Administrator Access** component to.

To launch a virtual workstation

1. Choose the **Launch** tab from the left navigation pane.



2. Select the vertical ellipsis (⋮) on the card to open a dropdown menu.

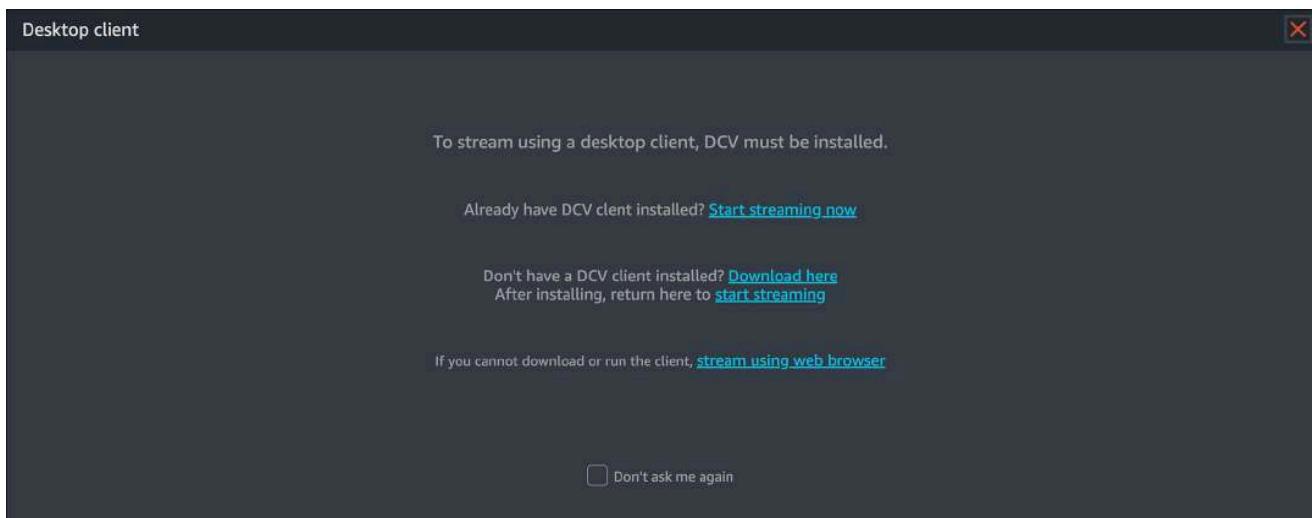


3. Choose **Launch with...**
4. For **Instance Type**, keep it at the default setting.
5. For **Amazon Machine Image**, verify that **NimbleStudioWindowsStreamImage** is selected.
6. For **Streaming Preference**, choose your streaming preference.
 - a. For the best performance, we recommend choosing **Launch native client**.
 - b. You must download the NICE DCV client before connecting to your workstation. For more information about the DCV client, as well as links to download, see **NICE DCV clients**.
7. Choose **Launch**.
8. A status bar will appear that shows you the progress of launching your virtual workstation. This might take up to 10 minutes.

To connect to the virtual workstation

1. When your virtual workstation is ready, a new window appears reminding you that the client must be installed.
2. Choose **Start streaming now**.

- If you haven't installed the NICE DCV desktop client, choose **Download here** and install the client first.



3. When your browser displays a window that prompts you to open NICE DCV, choose **Open** to continue. The exact wording of this might vary depending on what browser you're using.

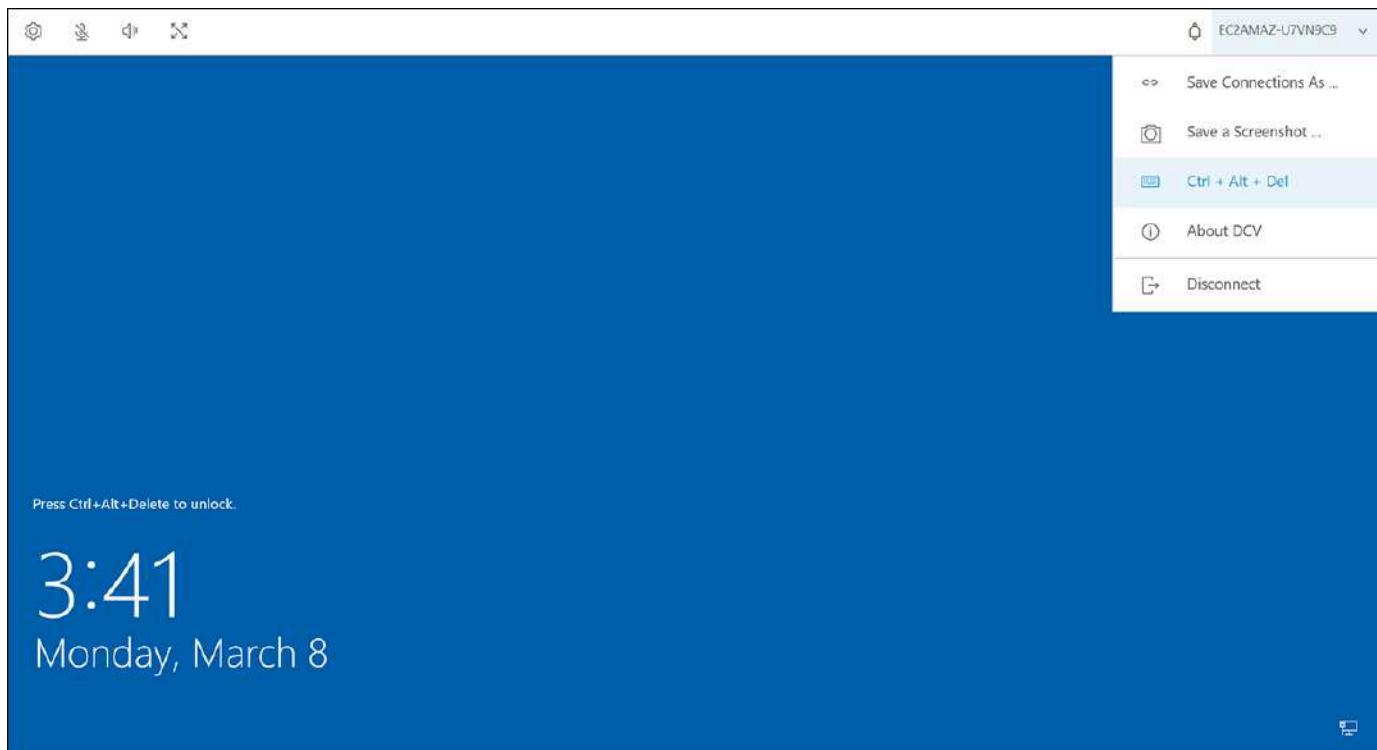
 **Note**

The NICE DCV web browser client runs inside a web browser. You don't need to install the web client. We recommend using the Google Chrome browser to avoid latency. For more information, go to the [Web browser client](#) page in the *NICE DCV User Guide*.

4. After the NICE DCV client application opens in a new window, the Windows login screen will display.
5. Open the instance menu near the top right of the screen and choose **Ctrl + Alt + Del**. For an OS X DCV client, open the **Connection** dropdown menu and select **Send Ctrl + Alt + Del**.

 **Important**

Don't enter **Ctrl+Alt+Delete** on your keyboard. Doing so sends the command to your local computer, not to your workstation.



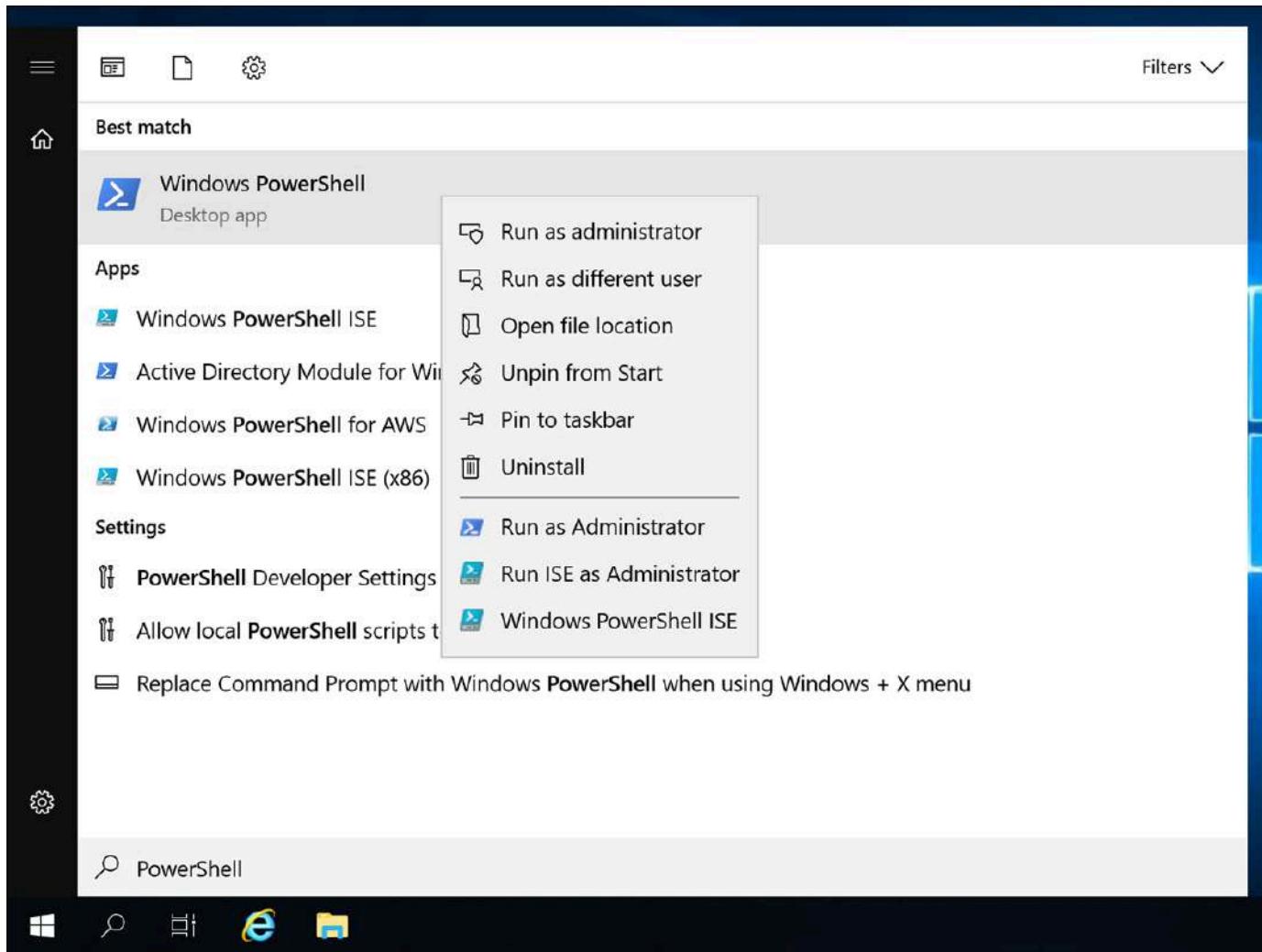
6. For **User name**, enter **Admin**. For **Password**, enter the password that you created during your studio deploy. Then press the enter (or return) key.

You're now connected to your virtual workstation.

Now that you logged in to your virtual workstation, you can test if your **Administrator Access** component worked as expected. To do this, open **PowerShell** as an administrator.

Test administrator access

1. Choose the **Start Menu** and search for **PowerShell**.
2. Open the context menu (right-click menu) on **Windows PowerShell**, then choose **Run as administrator**.



3. After you're prompted to allow the app to make changes to your device, enter your user name and password.
 - a. If your account has administrator permissions, PowerShell will open with the title: **Administrator: Windows PowerShell**.
 - b. If you don't have administrator access, you will be prompted to enter a user name and password again. Connect the **Administrator Access** component to your launch profile.

You have now provided administrator access to your Windows users.

Provide Superuser access for Linux users

Providing secure user access is a key aspect of Amazon Nimble Studio. However, there might be cases where you want to provide superuser access to particular users so they can install software.

One way to do this is to use the [sudoers policy module](#) to indicate a user's sudo privileges and give that user temporary superuser access.

Important

Any user with sudo access will have full read and write access to all files and folders, including the files in the mounted shared file systems, that are attached to the system.

Examples of POSIX file systems: FSx for Lustre and EFS.

This tutorial explains how to create a specific component that you can attach to a launch profile so that you can provide **sudo** capability. This component will provide sudo capability only to users that you deem appropriate. By granting those users superuser access, you're giving them permission to install software and perform root level operations. However, it's important to be aware that there are risks when providing superuser access. Any user with sudo access can access any file or folder in the studio. For information about security recommendations, see [Security best practices in AWS Identity and Access Management](#) in the IAM User Guide.

Contents

- [Prerequisites](#)
- [Step 1: Add a superuser access component to your studio](#)
- [Step 2: Add the Sudo Access component to a launch profile](#)
- [Step 3: Test Sudo Access component](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.

Step 1: Add a superuser access component to your studio

In this step, you will use the custom configuration component to create a system initialization script that enables sudo use.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.

2. Choose **Studio resources** in the left navigation pane.
3. Choose **Add** in the **Custom configuration** studio resource type.
4. In the **Custom configuration info** section, complete the fields as follows:
 - a. **Region:** Select the AWS Region that your studio is deployed in.
 - b. Choose the name for your component. Example: **Sudo Access**.
 - c. (Optional) Give your component a description.

Custom configuration info

Region [Info](#)
Select the Region that this custom configuration will be associated with. Defaults to the home Region.

US West (Oregon) us-west-1 (HOME)

Custom configuration name
Set the name for the custom configuration that you are registering.

my custom configuration

The length is between 3-128 characters.

Custom configuration description - *optional*
Enter an optional description of this custom configuration.

description for my custom configuration

The description can be 3-256 characters.

5. Create a new **Script parameter** by choosing **Add new parameter**
6. In the **Parameter name** section, enter **USERLIST**.
7. In the **Parameter value** section, enter user names to grant superuser access to specific users. User names should be separated by only a space. Example: Admin maria richard

Parameter name	Parameter value - <i>optional</i>
USERLIST	Admin maria richard

[Add new parameter](#) [Delete](#)

8. In the **Initialization script** section, navigate to the **Linux** section. Add the following code to the system initialization section to give specific users sudo access.

```
SUDOERS_FILE=/etc/sudoers.d/50-ad-admin-user
echo -e "# Created by studio component $studioComponentId $studioComponentName" >
$SUDOERS_FILE
```

```
# save IFS so we can revert it
OLDIFS=$IFS
IFS=" "

for SUDO_USER in $USERLIST; do
    echo -e "$SUDO_USER\tALL=(ALL)\tALL" >> $SUDOERS_FILE
done

# revert IFS
IFS=$OLDIFS

chmod 440 $SUDOERS_FILE
```

9. In the **Security groups** section, choose the LicenseServer security group.
10. (Optional) Add tags if you're using tags to track your AWS resources.
11. Choose **Save custom configuration**.

Step 2: Add the Sudo Access component to a launch profile

These steps explain how to attach the **Sudo Access** component to an existing launch profile. We recommend creating a specific launch profile that's dedicated to Linux administrative work, so it's clear who has access to this component. Navigate to [Creating launch profiles](#) and follow the steps in that tutorial before returning to this step.

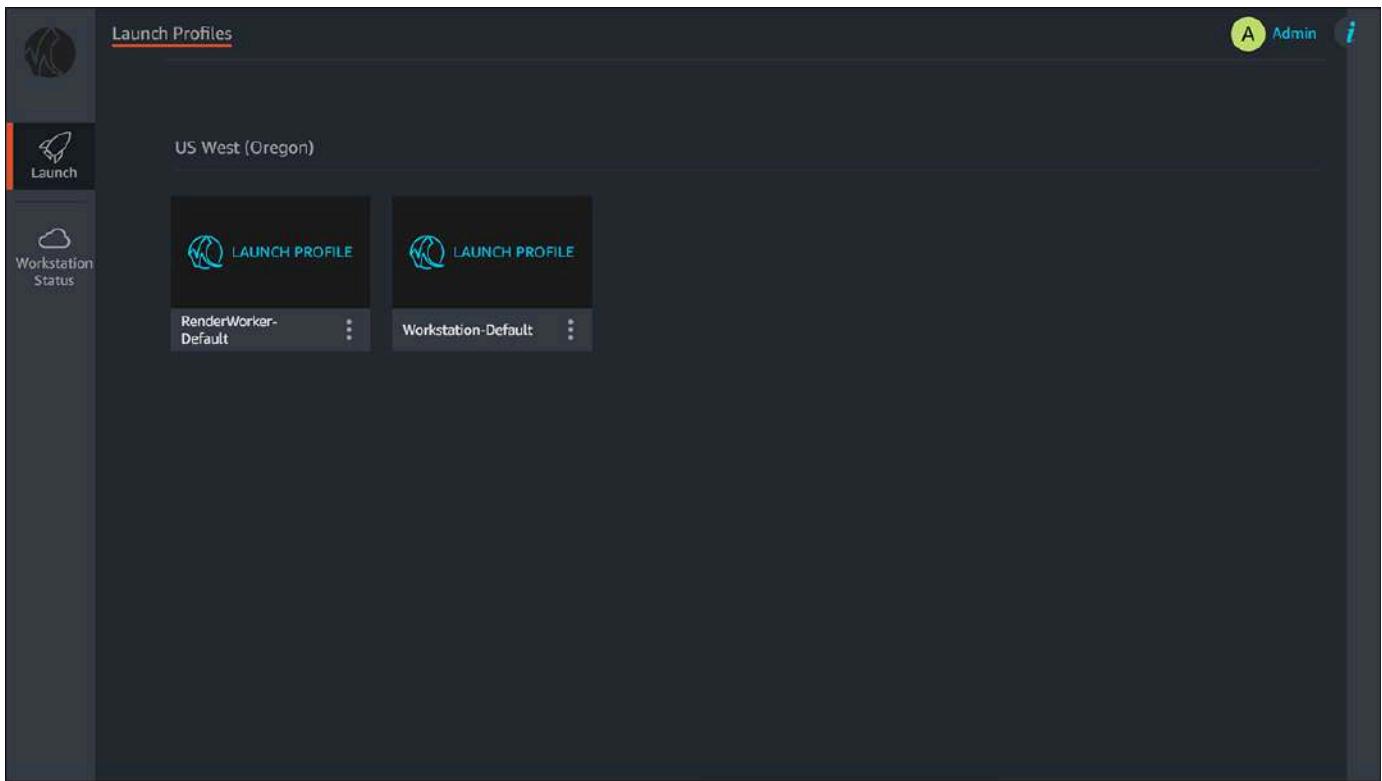
1. Choose **Launch profiles** in the left navigation pane.
2. Select a launch profile by selecting the dot to the left of its name.
3. Choose **Action**. Then choose **Edit**.
4. Navigate to the **Launch profile components** section.
5. Select the check box next to the **Sudo Access** component that you just created.
6. Choose **Update launch profile**.
7. Repeat these steps for all launch profiles that you want to have access to the **Sudo Access** component

Step 3: Test Sudo Access component

Sign in to a virtual workstation and test the process to check that the **Sudo Access** component is working as expected. Choose the launch profile that you added the **Sudo Access** component to.

To launch a virtual workstation

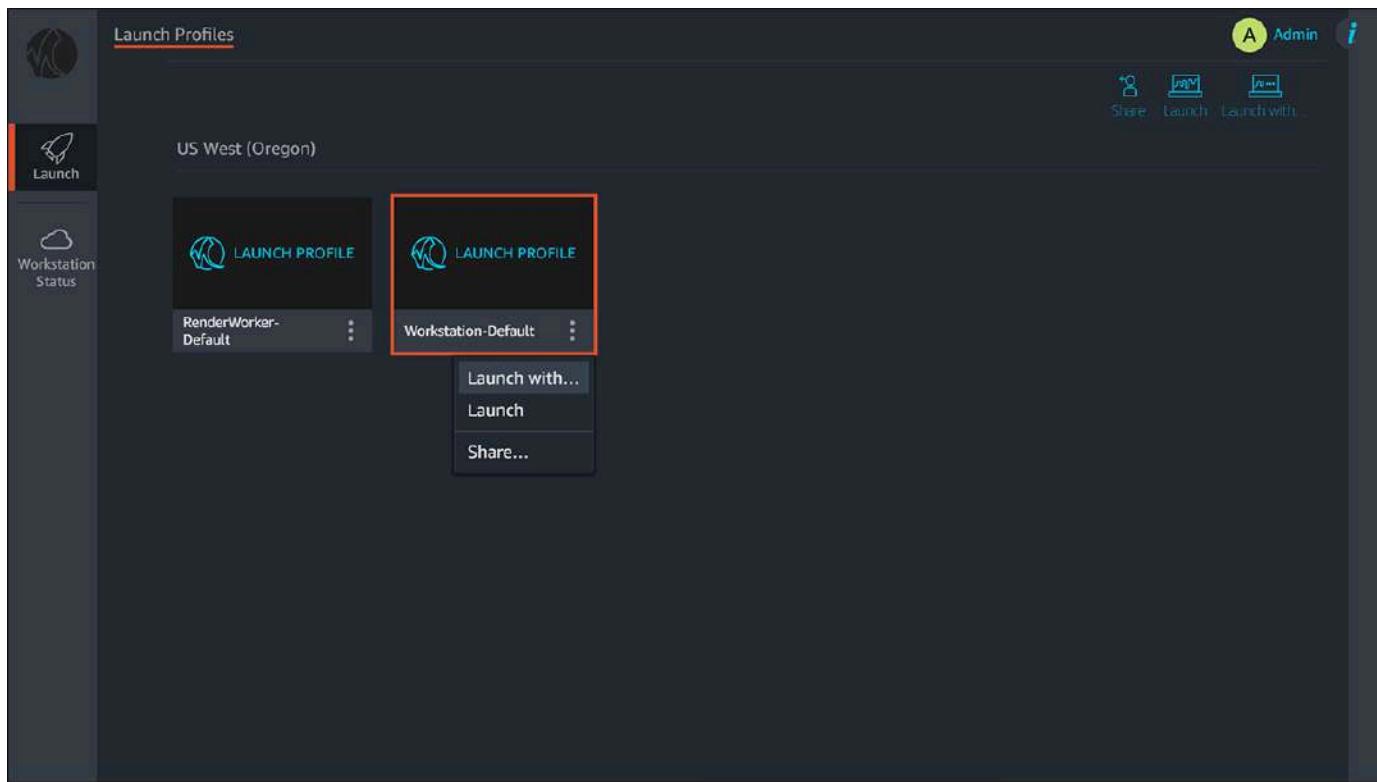
1. Choose the **Launch** tab from the left navigation pane.



2. Select the vertical ellipsis

()

on the card to open a dropdown menu.

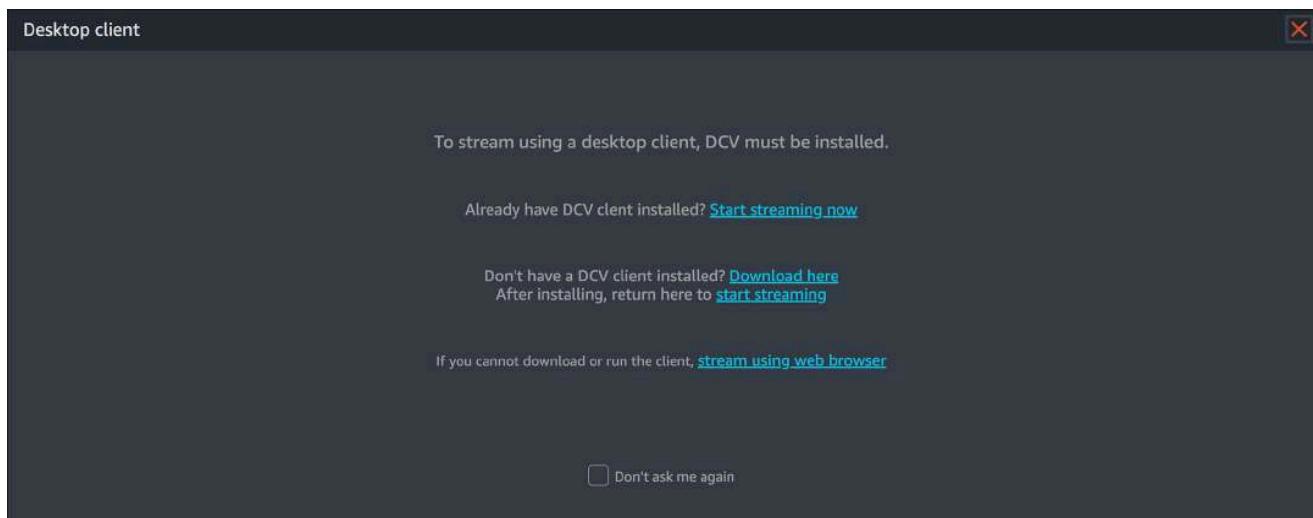


3. Choose **Launch with...**
4. For **Instance Type**, keep it at the default setting.
5. For **Amazon Machine Image**, verify that **NimbleStudioWindowsStreamImage** is selected.
6. For **Streaming Preference**, choose your streaming preference.
 - a. For the best performance, we recommend choosing **Launch native client**.
 - b. You must download the NICE DCV client before connecting to your workstation. For more information about the DCV client, as well as links to download, see **NICE DCV clients**.
7. Choose **Launch**.
8. A status bar will appear that shows you the progress of launching your virtual workstation. This might take up to 10 minutes.

To connect to the virtual workstation

1. When your virtual workstation is ready, a new window appears reminding you that the client must be installed.
2. Choose **Start streaming now**.

- If you haven't installed the NICE DCV desktop client, choose **Download here** and install the client first.



3. When your browser displays a window that prompts you to open NICE DCV, choose **Open** to continue. The exact wording of this might vary depending on what browser you're using.

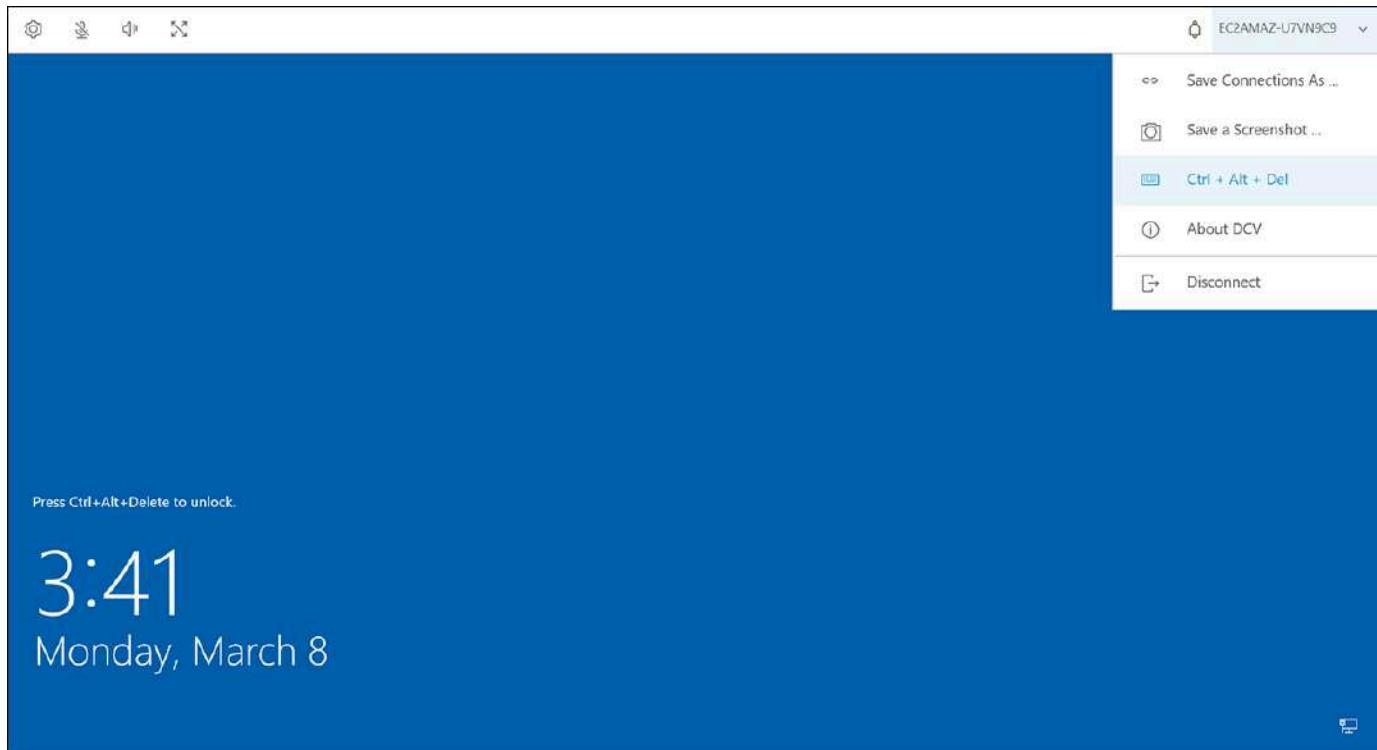
 **Note**

The NICE DCV web browser client runs inside a web browser. You don't need to install the web client. We recommend using the Google Chrome browser to avoid latency. For more information, go to the [Web browser client](#) page in the *NICE DCV User Guide*.

4. After the NICE DCV client application opens in a new window, the Windows login screen will display.
5. Open the instance menu near the top right of the screen and choose **Ctrl + Alt + Del**. For an OS X DCV client, open the **Connection** dropdown menu and select **Send Ctrl + Alt + Del**.

 **Important**

Don't enter **Ctrl+Alt+Delete** on your keyboard. Doing so sends the command to your local computer, not to your workstation.



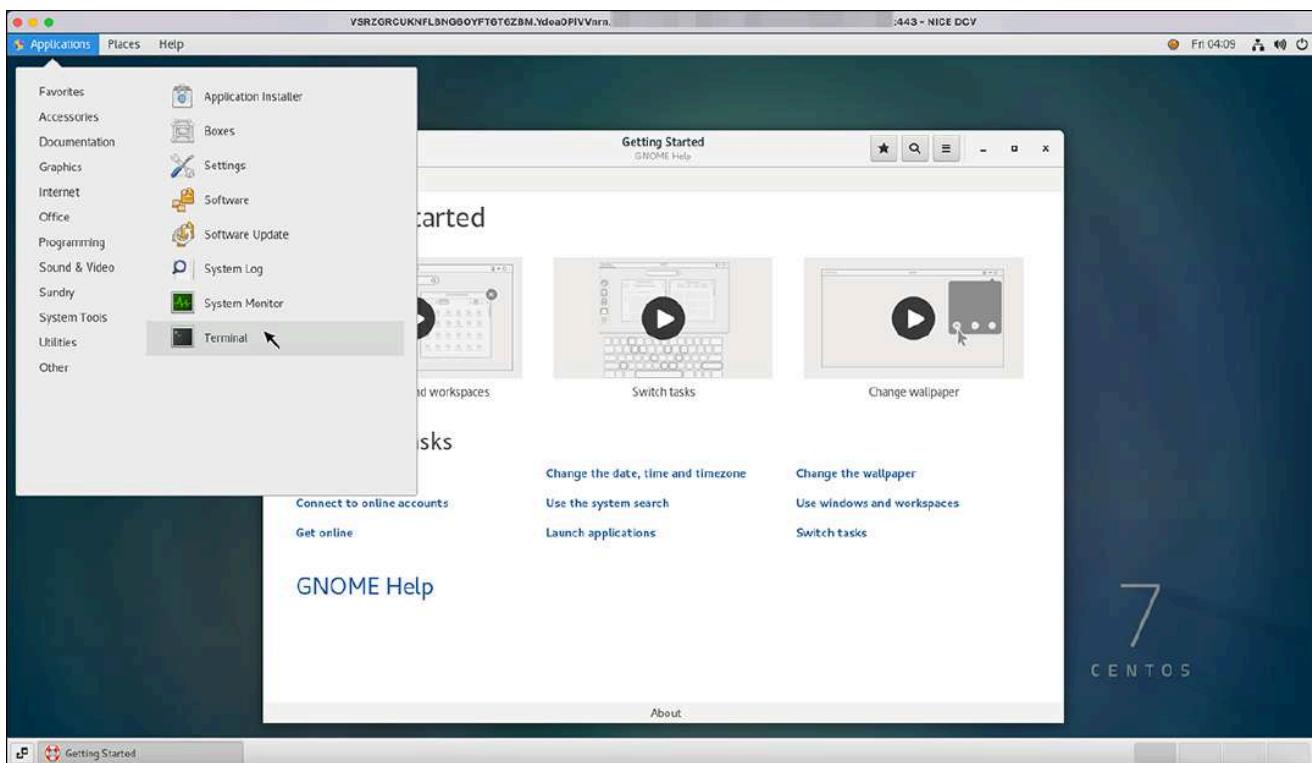
6. For **User name**, enter **Admin**. For **Password**, enter the password that you created during your studio deploy. Then press the enter (or return) key.

You're now connected to your virtual workstation.

Now that you have logged in to your virtual workstation, you can test whether your Sudo Access component worked as expected. To do this, use the **Terminal** to run a sudo command.

Test sudo access

1. Open the **Terminal**.
 - Select **Applications** in the menu bar. Then choose **System Tools** and **Terminal**.



2. In the terminal, enter the command sudo -v to test if your user has sudo access.
3. Enter the password for your account.
 - a. If your account has sudo permissions, a command prompt will appear.

```
admin@[REDACTED]:~/.local/share/nautilus/applications:/opt/Thinkbox/Deadline10/bin - □ ×
File Edit View Search Terminal Help
[admin@[REDACTED] bin]$ sudo -v
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for admin:
[admin@[REDACTED] bin]$
```

A screenshot of a terminal window. The title bar shows 'admin@[REDACTED]:~/.local/share/nautilus/applications:/opt/Thinkbox/Deadline10/bin'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The command '[admin@[REDACTED] bin]\$ sudo -v' is entered, followed by a message: 'We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things: #1) Respect the privacy of others. #2) Think before you type. #3) With great power comes great responsibility.' A password prompt '[sudo] password for admin:' appears, followed by a blank line '[admin@[REDACTED] bin]\$'.

- b. If you don't have access, you will receive a result like, Sorry, user [username] may not run sudo on [hostname]. If your account doesn't have permission to run sudo

commands, double-check your launch profile to check that you have connected the **Sudo Access** component.

You have now provided superuser access to your Linux users.

Update storage in Nimble Studio

During the StudioBuilder process of creating your studio, you created an Amazon FSx file system. You can repeat the StudioBuilder process at any time to update the capacity or throughput of that storage.

You can also manually add Amazon FSx file systems to your studio without using StudioBuilder. For step-by-step instructions, see the following tutorials.

Topics

- [Setting up an Amazon FSx Windows file system](#)
- [Setting up an Amazon FSx Lustre file system](#)
- [Setting up Linux home directories](#)
- [Setting up Weka in Nimble Studio](#)

Setting up an Amazon FSx Windows file system

There are several reasons why you can create additional file systems for your studio. For example, you can separate data for different projects, or have a separate drive for studio tools and scripts. Or you might need to store large datasets, such as rendered images or simulation data, without impacting performance on your other working drives.

This administrator tutorial will walk you through the process of creating a new file system with FSx for Windows File Server and attaching the file system to your studio as a new component. This new file system can be mounted to Windows or Linux workstations, and to render workers.

Contents

- [Prerequisites](#)
- [Step 1: Create a new file system](#)
- [Step 2: Attach the file system to the studio](#)
- [Step 3: Update launch profiles](#)

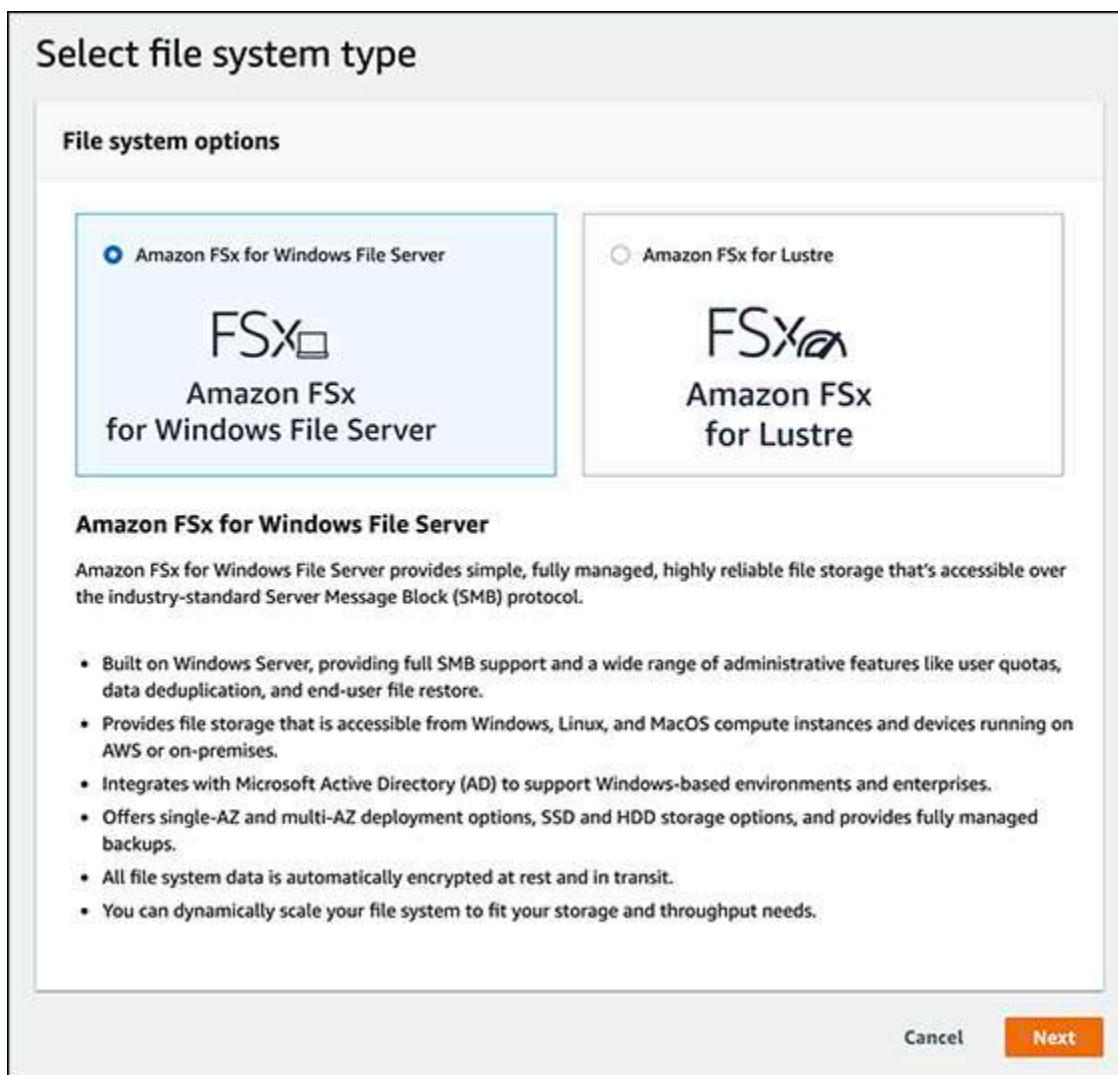
Estimated time: 45 minutes

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.

Step 1: Create a new file system

- Sign in to the AWS Management Console and open the [Amazon FSx](#) console.
- Make sure you're in the same Region that you deployed your studio in.
- Choose **Create File System** in the upper right-hand corner.
- Choose **FSx for Windows File Server** and choose **Next**.



- In the **File system details** section, complete the fields as follows:

- a. Give the file system a **name**. For example, you might call it prod or tools, depending on the purpose of the file system.
 - b. For **Deployment type**, choose **Single-AZ**.
 - c. For Storage type, choose **SSD**.
 - d. Decide how much **Storage capacity** you want. This will depend on how much data you think your studio will be storing on this particular file system. You can choose an amount between 32 GB and 65,536 GB.
 - i. If you don't know how much storage capacity your studio will need, you can start with 100 GB.
 - ii. This can be adjusted later if you need to increase your storage capacity.
 - iii. For more information, check out [Managing storage capacity](#).
6. In the **Network & Security** section, complete the fields as follows:
 - a. Choose your studio's **Virtual Private Cloud (VPC)**. The Name tag for your studio's VPC has been set to your studio name by default.
 - b. Search for **FSX** in the **VPC Security Groups** dropdown menu and select it. Example: <your-studio-name>Data-FSxFileSystems
 - i. **Important**

Be careful to choose the correct security group here, as there are many security groups for your studio, and choosing the incorrect one will prevent your new storage from being mounted.
 - ii. If you see a **default** security group selected already, remove it by choosing the **X** to the right of its name.
 - c. For **Subnet**, choose the one called **Filesystems**.
 - d. In the **Windows authentication section**, choose your studio's AWS Managed Microsoft AD name from the dropdown menu. This will be the same name that you chose during deployment. Example: <name>.nimble.<region>.aws
 - e. For the **Encryption key**, choose the key called <your-studio-name>-Key.
 7. Choose **Next**.
 8. Choose **Create File System**.

- This process can take some time. For example, it takes around 30 minutes to create a 500 GB file system.

Step 2: Attach the file system to the studio

This part of the process will allow you to use your new file system in your Nimble Studio cloud studio.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. Choose **Add in File Storage**.
4. Provide the following information in the specified fields.
 - a. **Name** the storage. This name can match the name that you gave to the Amazon FSx file system that you just created.
 - b. (Optional) **File storage description**: Give your file storage a description.
 - c. **Storage type**: Choose **FSx for Windows File Server**.
 - d. **Available AWS file systems**: Choose the file system that you just created.
 - e. **Linux mount point**: Enter `/mnt/<filesystem_name>`. For example, enter `/mnt/prod`.
 - f. **Windows mount point**: Enter a drive letter on your Windows system where your new Amazon FSx file system will be mounted. For example, enter `X`
 - g. **Share name**: Enter `share`.
 - h. **Security Groups**: Choose the security group that starts with `<your-studio-name>-WorkstationEgress`.
5. (Optional) Add tags if you're using tags to track your AWS resources.
6. Read the terms and conditions and if you agree:
 - Select the check box next to **I understand that Nimble Studio will access my existing file storage**.
7. Choose **Save connection parameters**.

Step 3: Update launch profiles

To mount your new Amazon FSx file system on your render workers, you will need to update the render worker launch profile (**RenderWorker-Default**). You will also need to update the workstation launch profile (**Workstation-Default**) or other custom launch profiles that you've created in your studio.

1. Go to the [Launch profiles](#) page for your studio from Nimble Studio.
2. Select the launch profile that you want to edit.
 - For example: **RenderWorker-Default** or **Workstation-Default**.
3. Choose **Action**. Then choose **Edit**.
4. Navigate down to **Launch profile studio components** and select the Amazon FSx storage component that you attached in the previous section.

The screenshot shows the 'Launch profile studio components' section of the Nimble Studio configuration interface. It includes three main sections: 'Studio file storage components', 'Studio Active Directory component', and 'Studio farm components'. Under 'Studio file storage components', there is a note 'Select all that apply.' followed by two checked checkboxes: 'FSxWindows' and 'Prod'. Under 'Studio Active Directory component', there is a note 'Select one.' followed by one checked checkbox: 'ActiveDirectory'. Under 'Studio farm components', there is a note 'Select all that apply.' followed by one checked checkbox: 'RenderQueue'.

5. Choose **Update launch profile**.
6. Repeat these steps to make the new Amazon FSx file system available to your artists on any other launch profiles that you set up for your artists to use to launch workstations.
 - Once the launch profiles are updated, artists can launch with them and will have access to the new file system.

⚠️ Important

If you will be using your new file system, a Windows workstation, and Linux farm workers, set up Path Mapping in the Deadline Monitor so that Linux workers can find the submitted file, even though it was sent from a Windows workstation. See [Configuring AWSThinkboxDeadline](#) to configure Deadline to handle paths properly.

Setting up an Amazon FSx Lustre file system

This document will show you how to add a FSx for Lustre file system to your studio in Amazon Nimble Studio.

Be aware that part of this tutorial involves granting sudo access to studio users. Sudo access will automatically give studio users root user privileges and full access to your FSx for Lustre file system. Exercise caution when assigning sudo access to any user.

Contents

- [Prerequisites](#)
- [Step 1: Create a new file system](#)
- [Step 2: Attach the file system to the studio](#)
- [Step 3: Attach a new component to each launch profile](#)
- [Step 4: Change permissions on the Lustre mount point](#)
- [Step 5: Sign in to your operating system](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.

Step 1: Create a new file system

1. Sign in to the AWS Management Console and open the [Amazon FSx](#) console.
2. Make sure that you're in the same Region that you deployed your studio in.

3. Choose **Create File System**.
4. Select **Amazon FSx for Lustre** for **File systems options** and choose **Next**.
5. In the **File system details** section, you can input a **File System name (Optional)** that will help you recognize the file system later.
 - You can use up to 256 Unicode letters, white space, and numbers plus the special characters + - = . _ : / to create the name. This example uses My-Test-Lustre-File-System.

Create file system

File system details

File system name - optional [Info](#)

Maximum of 256 Unicode letters, whitespace, and numbers, plus + - = . _ : /

Deployment and storage type [Info](#)
Select a deployment type and storage type to fit your workload requirements
 Persistent, SSD
 Persistent, HDD
 with SSD cache
 Scratch, SSD

Throughput per unit of storage [Info](#)
Throughput (MB/s) per unit of storage (TiB)
 50 MB/s/TiB (up to 1.3 GB/s/TiB burst)
 100 MB/s/TiB (up to 1.3 GB/s/TiB burst)
 200 MB/s/TiB (up to 1.3 GB/s/TiB burst)

Storage capacity [Info](#)

6. In the **Deployment and storage type** section choose one of these options:
 - a. Choose **Persistent, SSD** for latency-sensitive workloads or workloads requiring the highest levels of IOPS/throughput. For information about in-transit encryption of data for file systems in AWS Regions, see [Encrypting Data in Transit](#).
 - b. Choose **Persistent, HDD** for throughput-focused workloads that aren't latency-sensitive. For HDD-based file systems, the optional SSD cache improves performance by automatically placing your most frequently read data on SSD (the cache size is 20% of your file system size).

7. Next, choose the **Throughput per unit of storage** that you want for your file system. This option is only valid for **Persistent** deployment types.
8. For **Storage capacity**, provide a storage capacity for your file system, in tebibyte (TiB):
 - a. For a persistent SSD or **scratch 2 file system**, this value can be 1.2 TiB or increments of 2.4 TiB.

 **Tip**

If you're experimenting, try the lowest storage capacity (such as 1.2TiB) to save costs.

- b. For a persistent HDD file system, you can choose increments of 6.0 TiB for 12 MB/s/TiB file systems and increments of 1.8 TiB for 40 MB/s/TiB file systems.
 - c. A **scratch 1 file system** can be 1.2, 2.4, or increments of 3.6 TiB.
 - d. You can increase the amount of storage capacity as needed after you create the file system. For more information, see [Managing storage and throughput capacity](#).
 - e. For **Data compression type**, choose **NONE**. For more information, see [Lustre data compression](#).
9. In the **Network & security** section, select your studio's **Virtual Private Cloud (VPC)** that you want to associate with your file system.

 **Important**

Don't choose the default Amazon VPC that you see in the Amazon VPC field. Instead, use the VPC that you specifically set up for your studio.

10. In the **VPC security groups** input field, enter **FSx File Systems** to find the security group for your studio, and then select it.

 **Important**

For security reasons, select the **FSx File Systems** security group, so that only the network traffic required by your FSx file systems is permitted. Don't use a security group with **(default)** next to it. If **(default)** is selected, select X to clear it.

11. In the **Subnet** dropdown, select the subnet associated with your file system's network interface.
12. Review the other options on the page, such as **Encryption** and **Tags**, and adjust to your own specifications.
13. Choose **Next** to proceed.
14. Confirm your settings and choose **Create file system**.
15. When the **File systems** page opens, your new FSx for Lustre file system will be listed.

Step 2: Attach the file system to the studio

In this step, you will add a new studio component so that your streaming instances and render worker instances can access Lustre.

1. If you're logged in to the AWS Management Console, proceed to the next step.
 - If you aren't logged in, go to the [AWS Management Console](#) and sign in as you normally would.
2. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
3. Choose **Studio resources** in the left navigation pane.
4. Choose **Add in File Storage**.
5. Provide the following information in the specified fields.
 - a. **Region:** Select the AWS Region that your studio is deployed in.
 - b. **File storage name:** Enter the name that you created in [Step 1: Create a new file system](#).
Example: My-Test-Lustre-File-System.
 - c. **(Optional) File storage description:** Give your file storage a description.
 - d. **Storage type:** Select **Amazon FSx for Lustre**
 - e. **Available AWS file systems:** Choose your Lustre instance. Example: My-Test-Lustre-File-System.
 - f. **Linux mount point:** Enter the directory where this file system will be mounted on Linux systems. For example: /mnt/lustre.
 - g. **Security groups:** Select the existing **FSxFileSystems** security group.

- To search for a security group, you can select the settings icon to open the **Preferences** module. There, you can select to view up to fifty items on a page. Use the find tool in your browser to search for **FSxFileSystems**.



6. Read the terms and conditions and if you agree:

- Select the check box next to **I understand that Nimble Studio will access my existing file storage.**

7. Choose **Save connection parameters**.

8. Your **Studio resources** page will open. Refresh the page to see your new Lustre file system, located in the **File storage** section.

This completes the process to create the studio component.

Step 3: Attach a new component to each launch profile

In these steps, you will attach your Lustre studio component to your **Workstation-Default** launch profile, so that your new file system is accessible there.

To attach your new component

- Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
- Choose **Launch profiles** in the left navigation pane.
- Next, select the **Workstation-Default** launch profile.
- Choose **Action**. Then choose **Edit**.
- When the **Edit launch profile** page opens, go to the **Launch profile components** section and select your new file system. Example: My-Test-Lustre-File-System.

Launch profile components**Studio license server components** [Info](#)

Select all that apply.

 Sudo Access**Studio custom configuration** [Info](#)

Select all that apply.

 InstanceConfiguration**Studio file storage components** [Info](#)

Select all that apply.

 Lustre **My-Test-Lustre-File-System** **FSxWindows****Studio farm components** [Info](#)

Select all that apply.

 RenderFarm**Studio Active Directory component** [Info](#)

Select one.

 ActiveDirectory

6. Select your **Streaming** preferences next. If you're testing things out, you can try XSmall (g4dn.xlarge).
7. Select **Update launch profile** to save these changes.
8. To see that the **Workstation-Default** has been updated, you might need to refresh your page after a few minutes. The **Status** will change from **Updating** to **Ready** to show you when it's complete.
9. Auto-mount Lustre on streaming instances and workers that use your launch profile upon their next boot.
10. To attach the same Lustre studio component to a **RenderWorker-Default** launch profile, follow *steps 2-4* as described earlier in this procedure.

Note

To make the previous updates effective, you or a user must re-launch the streaming workstation and render workers that are associated with the launch profiles that you just updated.

Step 4: Change permissions on the Lustre mount point

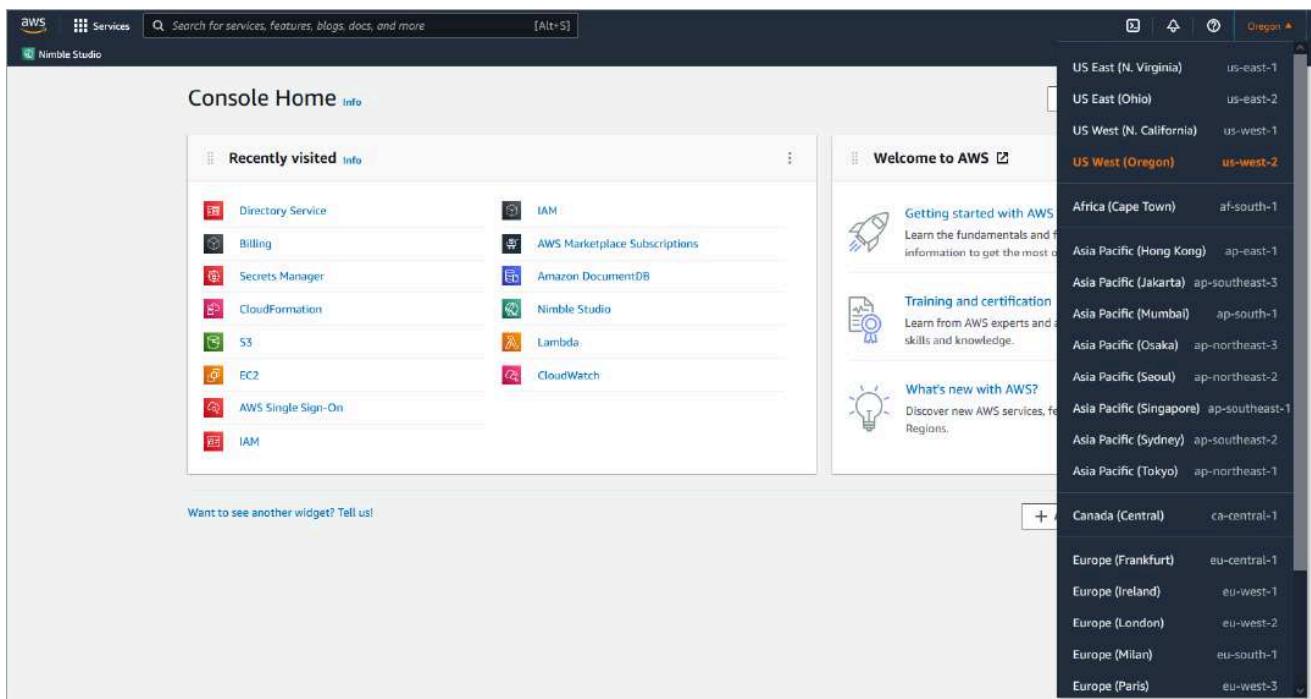
When you change permissions on the Lustre mount point, all Nimble Studio users of the **studiousers** group will attain read and write access to the file system. The instructions in this step will show you how.

⚠ Important

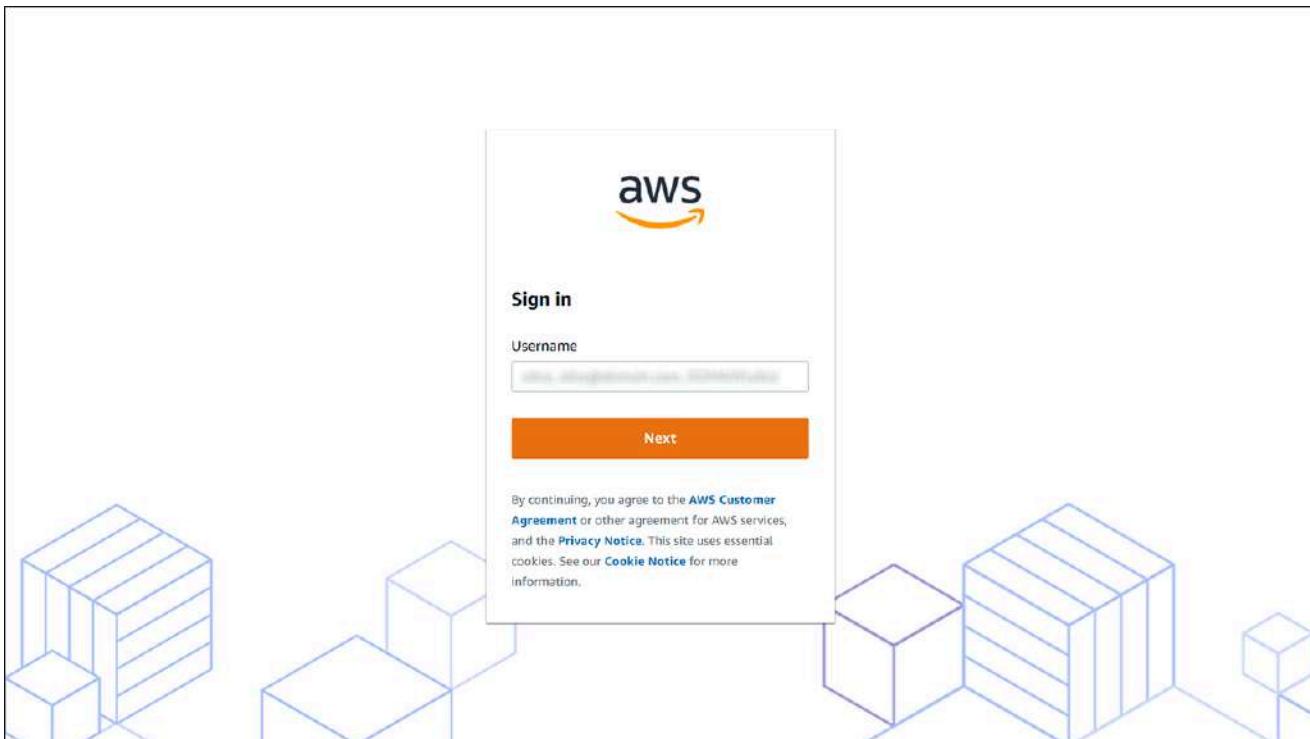
If you don't change permissions of the Lustre mount point, you will encounter a restrictive 0755. This will restrict write access to everyone except the root user. All other users will have read-only access.

To change permissions on the Lustre mount point

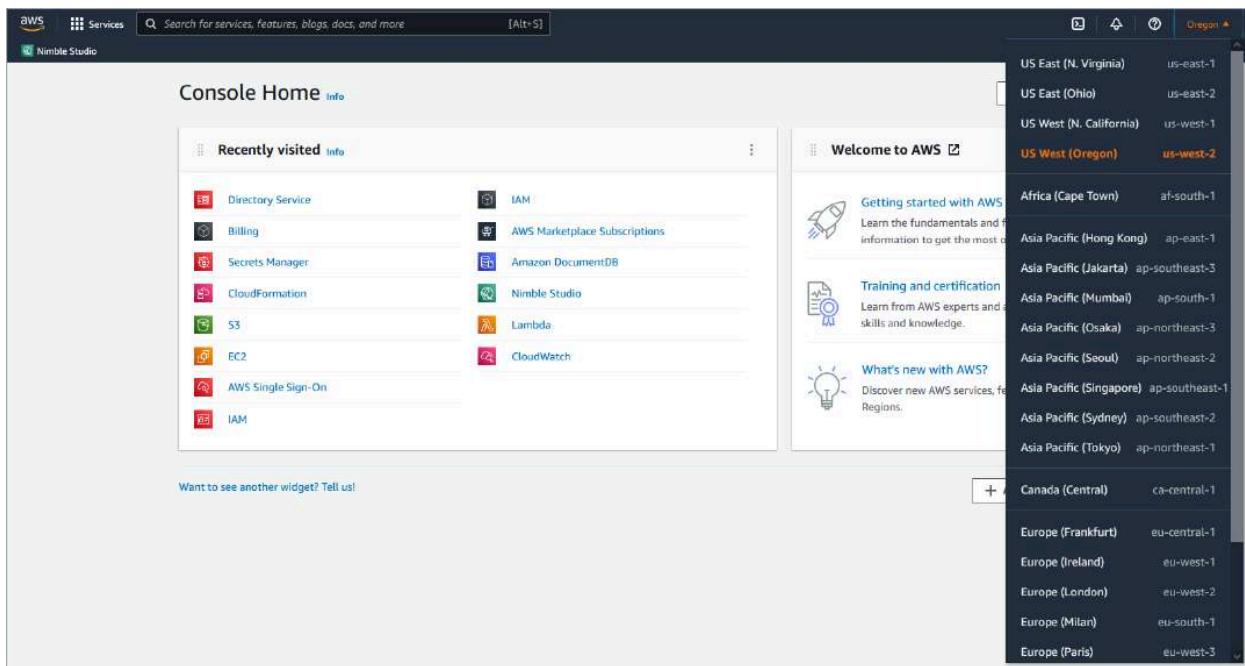
1. Follow the steps in the [Provide Superuser access for Linux users](#) tutorial to enable sudo access for the administrator in streaming instances.
2. Connect to the Nimble Studio portal:
 - a. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
 - b. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



- c. Choose **Studio manager** in the left navigation pane.
 - d. On the **Studio manager** page, choose **Go to Nimble Studio portal**.
3. Sign in to the Nimble Studio portal using your AWS Managed Microsoft AD administrator credentials.
- a. Enter **Admin** as the user name. Use the password that you set up during StudioBuilder deployment.



- b. If you forgot your password, do the following:
 - i. Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
 - ii. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



iii. Select the **Directory ID** for your studio's Active Directory.

iv. Choose **Reset user password**.

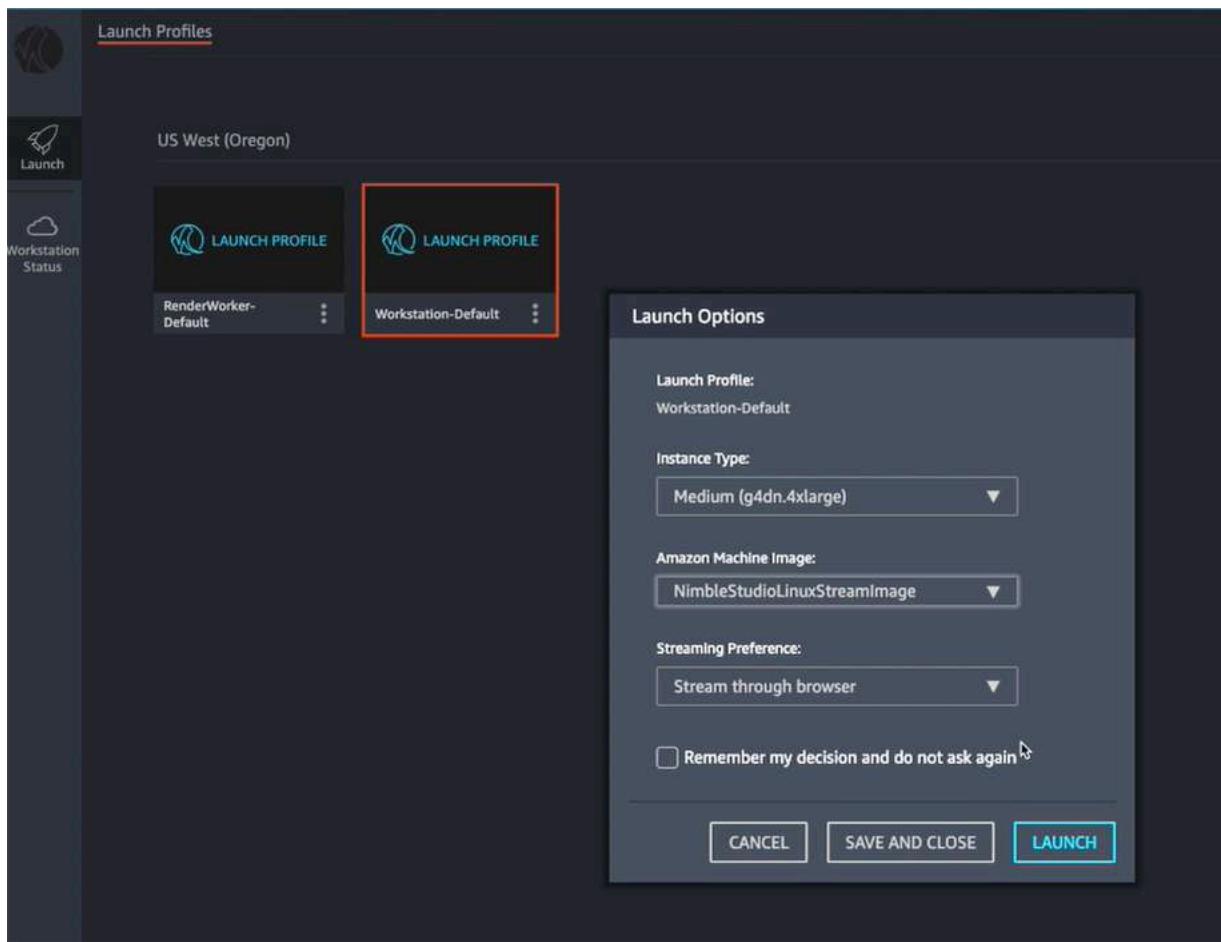
4. Choose **Launch profiles** in the left navigation pane.

The screenshot shows the 'Settings' page in Nimble Studio. The left sidebar has tabs for 'Personal', 'Launch Profiles' (which is selected and highlighted in orange), 'Studio Members', and 'EULA'. The main content area is titled 'LAUNCH PROFILES' and lists seven profiles: '3D Animation', 'Rigging', 'FX Simulation', '2D Animation', 'Comp Artist', 'Lighting', and 'Surfacing'. Each profile has a 'SHARED WITH' section showing user icons (e.g., k, c, n, a, j, d) and a 'STATUS' column with a green checkmark. A 'Dots' icon in the 'ACTIONS' column indicates a dropdown menu for each profile.

5. When you see your launch profiles, use the 3-dot tab in the **Workstation-Default** launch profile to open its dropdown.
6. Select **Launch with...** to open the **Launch Options** settings.
7. Choose **NimbleStudioLinuxStreamImage** from the dropdown.

- This will launch a new workstation streaming instance using a Linux AMI.

8. Select **Launch**.



9. If you get an error message saying that the ...current status is CREATING, you can confirm the status of the workstation by following these steps:
 - Sign in to the AWS Management Console and open the [Amazon FSx](#) console.
 - In the **Files systems** page, your **Files system name** will show the **Status**.
 - Wait for it to change from **Creating** to **Available** before you continue to the next step. This can take around 10 minutes.

Step 5: Sign in to your operating system

Important

Sudo access will automatically give studio users root user privileges and full access to your FSx for Lustre file system. Exercise caution when assigning sudo access to any user.

1. After you sign in to the operating system, open a terminal window.
2. Use `sudo -i` to get root privileges from your administrator terminal.
3. From this root terminal, enter `chown` and `chmod` commands for your Lustre mount point. For example, if you run these two commands, you will grant read and write permissions to all studio users of the Lustre file system.

```
chown admin:studiousers /mnt/lustre
```

```
chmod 3770 /mnt/lustre
```

Note

These permissions will be preserved in the Lustre File System immediately after you run the previous commands. The impact of the permission change is global. This means that it will affect all of the streaming workstations and render worker instances that have this Lustre File System mounted. All new streaming and worker instances will see these updated permissions.

Now that you have completed this tutorial, you have an FSx for Lustre file system! For more information about FSx for Lustre, see the [Amazon FSx for Lustre User Guide](#) or the [Amazon FSx Troubleshooting](#) chapter.

Setting up Linux home directories

This tutorial is for studios that will be adding the use of Linux virtual workstations for their artists. The following steps explain how administrators can manually create a shared file system within an Amazon Nimble Studio cloud studio that hosts Linux home directories. Placing home directories on shared storage allows user preferences to persist across virtual workstation streaming sessions.

At the end of this tutorial, you will have a component that you can attach to launch profiles to make sure that your users have Linux home directories mounted in `/home/<username>` when they use a Linux virtual workstation.

Important

We recommend that you include this component with any workstation launch profile that contains a Linux AMI, otherwise your users won't have access to their `/home/<username>` directory.

- Linux user home directory: `/home/<username>`
- Windows user home directory: `C:\Users\<username>`
 - Windows users utilize roaming profiles, which will save their storage on the Z: drive in `Z:\UserProfiles\<username>`
 - No custom setup is necessary for Windows users.

Note

A user's Windows and Linux home directories aren't shared. Any files a user has saved in their Linux home directory won't be available to them if they sign in to a Windows workstation. To transfer files between operating systems, it's recommended to save files in shared storage.

Contents

- [Prerequisites](#)
- [Step 1: Create security groups for your shared file system and Nimble Studio](#)
- [Step 2: Create a file system with Amazon Elastic File System \(Amazon EFS\)](#)
- [Step 3: Update file system subnet network ACLs](#)
- [Step 4: Create a custom studio resource](#)
- [Step 5: Add the custom resource to a launch profile](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.

Step 1: Create security groups for your shared file system and Nimble Studio

To create a secure connection between the shared file system containing your user's home directory and Nimble Studio, you will be creating two security groups. One security group will be connected to the custom storage component as part of Nimble Studio, and the second security group will be connected to the file system. We allow the second security group to connect to the first security group through a single port.

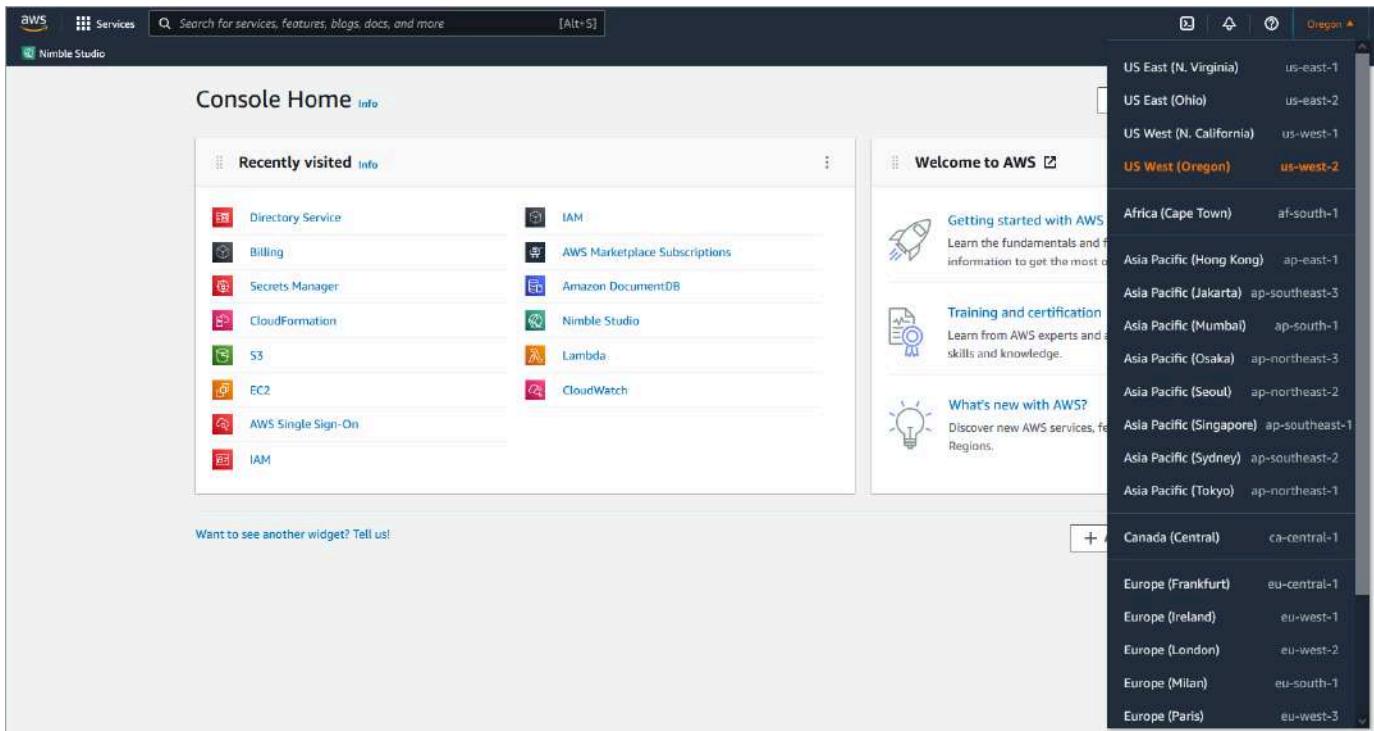
This security group configuration allows for flexibility. If you decide to create more shared file systems, you can add them to the second security group and they will be able to easily connect to the custom component.

Important

The configuration of security items, such as security groups and network ACLs, which control who can access your cloud resources, is your responsibility as part of the Shared Responsibility Model. These instructions and recommendations are resources for you, but consult with your studio administrator and IT team before making security decisions. See the following document for more information: [Shared Responsibility Model - Amazon Web Services \(AWS\)](#).

To create a custom studio component security group

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



3. Choose **Security Groups** in the left navigation pane.
4. Choose **Create security group**.
5. Provide the following information in the specified fields.
 - a. Security group name: **<your-studio-name>_LinuxHome_StudioComponent_SG**.
 - b. Description: **Connected to LinuxHome Studio Component**.
 - c. **VPC**: Choose your studio's VPC.
 - Your studio's VPC is called **<your-studio-name>**.
 - d. Keep all default inbound and outbound rules.
 - e. Optional: Choose **Add new tag** and enter **Name** as the key and **<your-studio-name>_LinuxHome_StudioComponent_SG** as the value.
 - This makes it easier to find the security group in the console.
 - f. Choose **Create security group**.

To create a security group for the elastic file system

1. Navigate back to the **Security Groups** page.
2. Choose **Create security group**.

3. Provide the following information in the specified fields.
 - a. Security group name: <your-studio-name>_LinuxHome_EFS_SG.
 - b. Description: **Controls access to the shared home file system.**
 - c. **VPC:** Choose your studio's VPC.
 - Your studio's VPC is called <your-studio-name>.
 - d. In the **Inbound rules** section:
 - i. Choose **Add rule**.
 - ii. Set the **Type** to **NFS**.
 - iii. For the **Source**, search for and select the <your-studio-name>_LinuxHome_StudioComponent_SG security group configured previously.
 - e. Optional: Choose **Add new tag** and enter **Name** as the key and <studio>_LinuxHome_EFS_SG as the value.
 - f. Choose **Create security group**.

Step 2: Create a file system with Amazon Elastic File System (Amazon EFS)

Amazon EFS provides file storage for your Amazon EC2 instances. With Amazon EFS, you can create a file system, mount the file system on your EC2 instances, and then read and write data from your EC2 instances to and from your file system. This step of the tutorial describes how to create an Amazon EFS file system that is suitable for storing user desktop preferences and other miscellaneous files. After this step, you will attach the file system to Nimble Studio as a Storage Component. To learn more, see the [Amazon EFS documentation](#).

To create a file system in Amazon EFS

1. Sign in to the AWS Management Console and open the [Amazon EFS](#) console.
2. Choose **File systems** in the left navigation pane.
3. Choose **Create file system** at the top of the window.
4. Provide the following information in the specified fields.
 - a. Name: <your-studio-name>_LinuxHome.
 - b. **VPC:** Choose your studio's VPC.

- Your studio's VPC is called <your-studio-name>.
- c. Availability and Durability: **One Zone**.
- d. Availability Zone: Select the Availability Zone from the dropdown list where your studio's streaming instances will be running.
- To check which AZ is the correct one:
 - A. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
 - B. Choose **Subnets** in the left navigation pane.
 - C. Search for **Workstations** in the filter field.
 - D. Select the check box next to the **Workstations** subnet.
 - E. In the **Details** tab, look at the **Availability Zone**.
5. Choose **Customize**.
6. Choose **Next**.
7. Select the **Filesystems** subnet for **Mount targets**.
8. Remove the default security group and choose the <your-studio-name>_LinuxHome_EFS_SG that you created earlier.
9. Choose **Next**.
10. Skip the files system policy section and choose **Next**.
11. Choose **Create**.
12. Wait for the file system to be created.

Now that the Amazon EFS file system is set up, notice the target IP address. This address will allow you to connect with the custom file storage component later in the tutorial.

Copy target IP address for the file system

1. Choose **Attach** at the top of the window on the file system detail page.
 2. Select **Mount via IP**.
 3. Select the Availability Zone from the dropdown list where your studio's streaming instances will be running.
- To check which AZ is the correct one:

- i. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
 - ii. Choose **Subnets** in the left navigation pane.
 - iii. Search for **Workstations** in the filter field.
 - iv. Select the check box next to the **Workstations** subnet.
 - v. In the **Details** tab, look at the **Availability Zone**.
4. The IP address will be displayed in red text within the **Using the NFS client** command.
 - Example: **192.0.2.0**
 5. Save this IP address in a text editor for later use.

Step 3: Update file system subnet network ACLs

Update the security settings on the network ACLs for the file system subnet to allow your workstation to communicate with the file system over NFS.

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Choose **Subnets** in the left navigation pane.
3. Search for the subnet called **Filesystems**.
4. Select the check box next to the **Filesystems** subnet.
5. Navigate down to the **Details** panel and choose the **Network ACL**.
6. Select the check box next to the network ACL
7. In the lower panel, choose **Inbound rules**.
8. Choose **Edit inbound rules**.
9. Choose **Add new rule**.
10. Fill out the rule as follows:
 - a. **Rule number:** choose any number. For example: 1.
 - b. **Type:** **NFS (2049)**

The screenshot shows a dialog box for adding a new inbound rule. The fields are as follows: Rule number (1), Type (NFS (2049)), Protocol (TCP (6)), Port (2049), Source (0.0.0.0/0), Action (Allow), and a Remove button.

1	NFS (2049)	TCP (6)	2049	0.0.0.0/0	Allow	Remove
---	------------	---------	------	-----------	-------	--------

11. Choose **Save changes**.

Step 4: Create a custom studio resource

A custom studio resource is needed to do three things: check that the virtual workstation that's used by artists on your team is allow-listed for access to the home directory, open up Linux security settings to allow write permissions to NFS storage, and enable the Amazon EFS file system to be remounted if the instance is rebooted.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. Choose **Add in File Storage**.
4. Provide the following information in the specified fields.
 - a. **Region:** Select the AWS Region that your studio is deployed in.
 - b. **Name:** Enter **LinuxHome**
 - c. **(Optional) File storage description:** Give your file storage a description. For example: **Set up Linux home directories.**
 - d. **Storage type:** Choose **Custom**
 - e. **Linux mount point:** Enter **/home**
 - f. **Windows mount drive:** Enter **Q**
 - The letter that you enter won't be used by your Windows workstation.
 - g. **Share:** Select **share**
 - The share value must be present to create the component but isn't used for anything else.
 - h. **Endpoint:** Select the IP address that you saved earlier in the tutorial
 - i. **Windows system initialization script:** Enter **# N/A**
 - You won't be using any Windows scripts, as this component is purely for Linux virtual workstations, but this means that the Windows portion of the initialization script gets ignored.
 - j. **Linux system initialization script:** Enter the following text.

```
#  
# Default SELinux configuration prevents `gdm-session-worker` from having write  
access to /home
```

```
# if it is NFS mounted. Enabling write access is required in order for `gdm` to
# create the user home
# directory if it doesn't exist
#
setsebool -P use_nfs_home_dirs=True

#
# Add mount entry to fstab if it doesn't exist to ensure it is
# remounted if the instance is rebooted
#
if ! grep "Linux-Home" /etc/fstab > /dev/null; then
    echo "Adding mount for /home"
    cat <<EOF >> /etc/fstab
# Linux-Home on Amazon EFS
$endpoint:/ /home nfs4
    nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 0
    0
EOF
fi

#
# Perform the mount
#
mount /home
```

- There is no need to modify the code. It will use the IP address that you entered in the **endpoint** field previously.
5. **Security groups:** Choose the **<your-studio-name>_LinuxHome_StudioComponent_SG** security group that you created earlier in the tutorial.
 6. (Optional) Add tags if you're using tags to track your AWS resources.
 7. Read the terms and conditions and if you agree:
 - Select the check box next to **I understand that Nimble Studio will access my existing file storage.**
 8. Choose **Save connection parameters.**

Step 5: Add the custom resource to a launch profile

This step explains how to attach the custom component to an existing launch profile. To create a new launch profile, follow the steps in the [Creating launch profiles](#) tutorial before returning to this step.

Important

We recommend that you include this component with any workstation launch profile that contains a Linux AMI, otherwise your users won't have access to their /home/<username> directory.

1. Choose **Launch profiles** in the left navigation pane.
2. Select a launch profile by selecting the dot to the left of its name.
3. Choose **Action**. Then choose **Edit**.
4. Navigate down to **Launch profile components**.
5. Choose the check box next to the **LinuxHome** that you just created.
6. Choose **Update launch profile**.
7. Repeat these steps for all launch profiles that you want to have access to the **LinuxHome** custom component that you created.

After you complete these steps, a studio user can launch a Linux workstation and make changes to their user preferences. These changes will persist across all machines that are launched by that user.

Setting up Weka in Nimble Studio

Amazon Nimble Studio supports the Amazon FSx server by default. Nimble Studio also supports third-party storage providers, such as Weka.

[Weka](#) is a software-based, scalable file system that is fully distributed and POSIT-compliant. For more information about Weka, see the [Weka documentation](#).

This tutorial shows you how to deploy a Weka cluster in Nimble Studio.

Contents

- [Prerequisites](#)
- [Step 1: Gather information](#)
- [Step 2: Prepare the network](#)
- [Step 3: Deploy Weka using CloudFormation](#)
- [Step 4: Update security groups](#)
- [Step 5: Linux setup](#)
- [Step 6: \(Optional\) Windows setup](#)
- [Troubleshooting](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- Have a [Weka account](#).

Step 1: Gather information

You need to gather some information about your Nimble Studio cloud studio resources before you can deploy the Weka cluster in [Step 3: Deploy Weka using CloudFormation](#). This information will also be used in [Step 6: \(Optional\) Windows setup](#).

Find the Nimble Studio VPC CIDR

In the following steps you'll find the IPv4 Classless Inter-Domain Routing (CIDR) of your studio's Amazon Virtual Private Cloud (Amazon VPC).

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Choose **Your VPCs** in the left navigation pane.
3. Choose your studio's VPC.
 - Your studio VPC is named <your-studio-name>.
4. In the **Details** section, notice the **IPv4 CIDR**. You will use this information in [Step 3: Deploy Weka using CloudFormation](#).

Details			
VPC ID vpc-[REDACTED]	State Available	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP options set dopt-[REDACTED]	Main route table rtb-[REDACTED]	Main network ACL acl-[REDACTED]
Default VPC No	IPv4 CIDR [REDACTED]	IPv6 pool -	IPv6 CIDR (Network border group) -
Route 53 Resolver DNS Firewall rule groups -	Owner ID [REDACTED]		

Find the Workstation subnet Availability Zone

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Subnets** in the left navigation pane.
3. Choose the subnet named **Workstations**.
4. In the **Details** section, notice the **Availability Zone**.

Details			
Subnet ID subnet-[REDACTED]	Subnet ARN [REDACTED]	State Available	IPv4 CIDR [REDACTED]
Available IPv4 addresses 1019	IPv6 CIDR -	Availability Zone us-west-2c	Availability Zone ID usw2-az3
Network border group us-west-2	VPC vpc-[REDACTED] tumeagostudio	Route table rtb-[REDACTED] Workstations	Network ACL acl-[REDACTED] Workstations
Default subnet No	Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No
Customer-owned IPv4 pool -	Outpost ID -	IPv4 CIDR reservations -	IPv6 CIDR reservations -
Owner [REDACTED]			

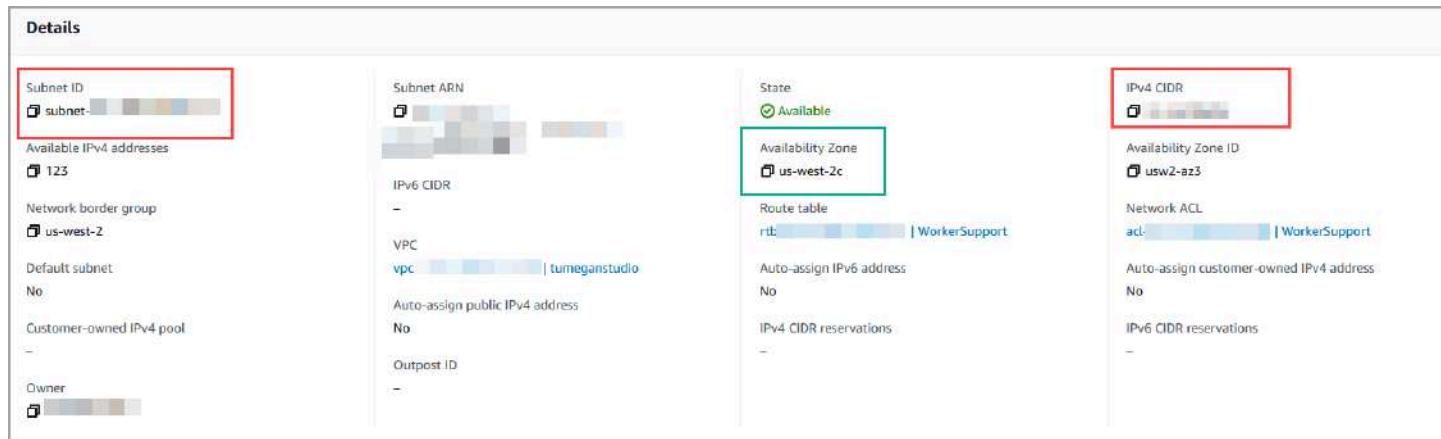
You will use this information in the following section, **Find the WorkerSupport subnet ID**.

Find the WorkerSupport subnet ID

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Subnets** in the left navigation pane.
3. There are two subnets named **WorkerSupport**. Do the following for both subnets.
4. Select the subnet.
5. In the **Detail** section, find the **Availability Zone**.

6. If the **Availability Zone** matches the Workstations subnet's **Availability Zone**, notice the following information. You will use this information in [Step 3: Deploy Weka using CloudFormation](#).

- Subnet ID**
- IPv4 CIDR range of that WorkerSupport subnet**



Find the Active Directory information

AWS Directory Service for Microsoft Active Directory is an AWS Managed Microsoft AD hosted on the AWS Cloud. To join Weka with AWS Managed Microsoft AD, first find the AWS Managed Microsoft AD information.

1. Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
2. Select **Directories** in **Active Directories** in the left navigation pane.
3. Select the AWS Managed Microsoft AD that Nimble Studio deployed. That **Directory name** is <your-studio-name>.nimble.<studio-region>.aws
4. Notice the following information. You will use this information in [Step 3: Deploy Weka using CloudFormation](#).
 - Directory DNS name**
 - Directory NetBIOS name**
 - Directory DNS addresses** (both IP addresses)

The screenshot shows the 'Directory details' section with 'Directory type' set to 'Microsoft AD'. It also shows 'Edition' as 'Standard'. The 'Networking & security' tab is active. In the 'Networking details' section, there are fields for 'VPC', 'Availability zones', and 'Subnets'. A red box highlights the 'DNS address' field. To the right, there's a summary of the status: 'Status' is 'Active', last updated on 'Wednesday, September 22, 2021', and it was launched on the same day.

Step 2: Prepare the network

Before deploying Weka, reconfigure your security groups to work with Weka.

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Network ACLs** in the **Security** section on the left navigation menu.
3. Update the **WorkerSupport**, **Workstations**, **Active Directory**, and **RenderWorkers** security groups by modifying their inbound and outbound rules.
4. Select the security group that you want to modify.
5. Select **Actions**, **Edit inbound rules** to add or modify an inbound rule, or **Actions**, **Edit outbound rules** to add or modify an outbound rule.

This screenshot shows the 'Actions' menu for managing network ACLs. The 'Edit inbound rules' and 'Edit outbound rules' options are highlighted with a red box. Other visible options include 'View details', 'Create network ACL', 'Edit subnet associations', 'Manage tags', and 'Delete network ACLs'.

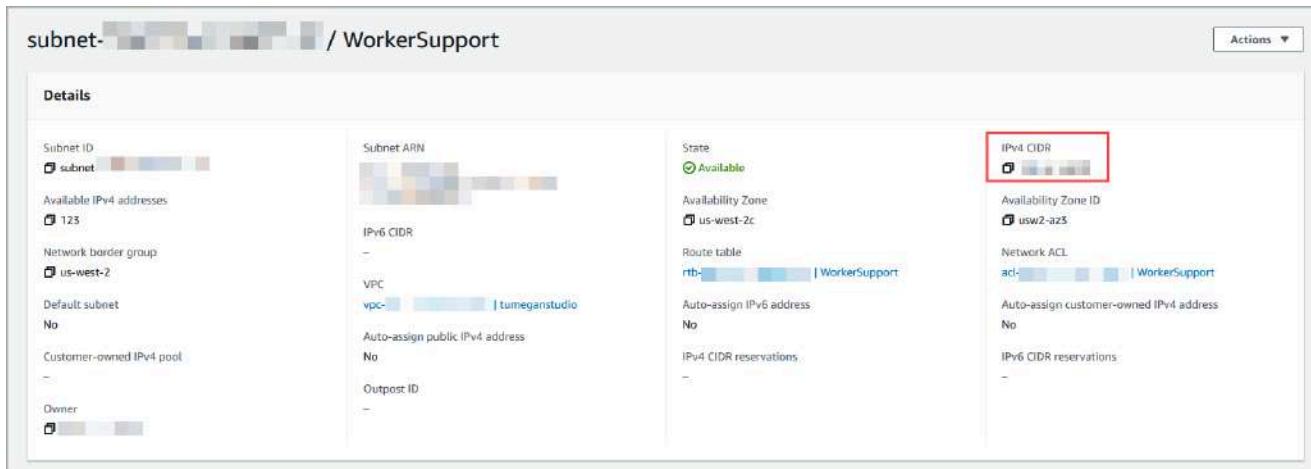
6. The following table describes how each of the security groups' rules should be modified.

NACL name	Inbound/Outbound	Operation	Rule number	Protocol	Ports	Source	Destination
WorkerSubnet	Inbound	MODIFY	10	TCP	0-65535	0.0.0.0	
	Inbound	ADD	30000	All Traffic		10.0.0.0/16	
	Outbound	ADD	30000	All Traffic			10.0.0.0/16
Workstations	Inbound	ADD	30000	All Traffic		10.0.0.0/16	
Active Director	Inbound	ADD	30000	All Traffic		IPv4 CIDR WorkerSubnet subnet	
	Outbound	ADD	30000	All TCP			IPv4 CIDR WorkerSubnet subnet
	Outbound	ADD	30100	All UDP			IPv4 CIDR WorkerSubnet subnet
RenderWorkers	Inbound	ADD	30000	All Traffic		10.0.0.0/16	

If you use VPC endpoints

If your studio was deployed with a VPC endpoint instead of a default AWS public endpoint, you'll need to complete one additional step to configure the network. This is so that your VPC endpoints can be reached from the WorkerSupport subnet.

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Subnets** in the **Virtual Private Cloud** section.
3. Choose the WorkerSupport subnet.
 - Notice the **IPv4 CIDR**, because you will use it in *step 4b*.



4. Select **Security Groups** in the **Security** section.
5. Select the **Vpc Interface Endpoints** security group.
6. Select **Actions**. Then select **Edit Inbound Rules**.
7. Select **Add Rule**.
 - a. Set the type to **HTTPS**.
 - b. Set the **Source** to the **IPv4 CIDR range** of the WorkerSupport subnet.
8. Select **Save rules**.

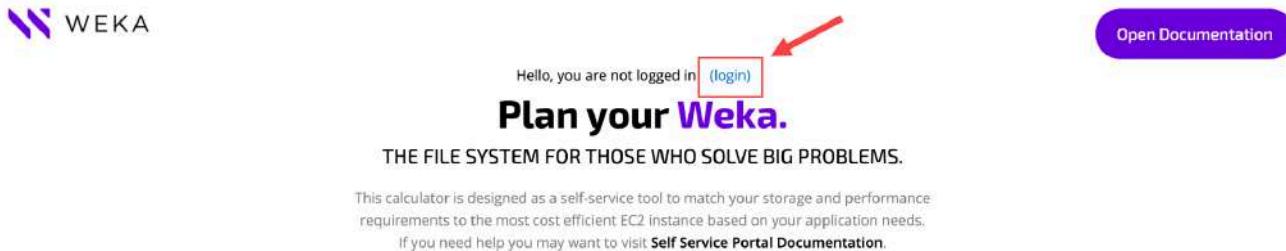
Step 3: Deploy Weka using CloudFormation

To deploy Weka, you will need to create a new key pair. After that, you will select the Weka storage that you need from the Weka website.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.

2. Choose **Key Pairs** in the left navigation pane.
3. Select **Create Key Pair**.
4. Enter a name for your key pair.
5. Make sure that your **Key pair type** is RSA and that the **Private key file format** is .pem.
6. Select **Create key pair**. You will need this key for *step 18*.

Go to start.weka.io and sign in to your Weka account.



7. Choose the specifications that you want for your file storage.
 - For example, 15TB of SSD only storage in us-west-2. Use the same region that your Nimble Studio is located in.

Start with **Your Requirements**

CAPACITY ?

Total Capacity 15 TB ▼ 1

Tiering SSD Only SSD+S3

PERFORMANCE ?

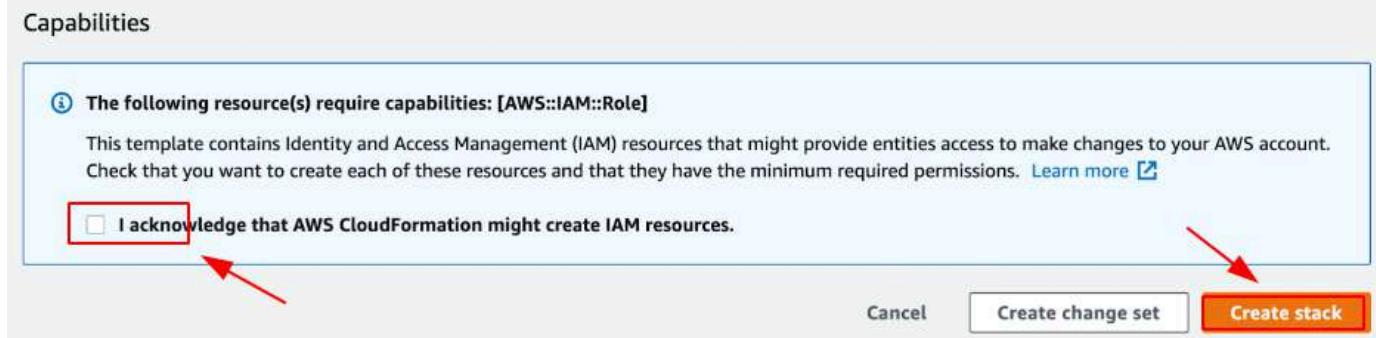
IOPs BW

R/W 80 %

IOPS 100 K

Region US West (Oregon) ^ 2

8. Select the most relevant storage based on your needs and usage.
9. . For example, the `i3en.2xlarge` instance type.
10. Select **Deploy to AWS** to the right of the specifications of your storage.
11. Choose the Weka version that you want to use from the **Version** dropdown.
12. Select the AWS Region that your studio is deployed in.
13. Clear the **Add clients to cluster** box.
14. Select **Deploy to AWS**.
15. You will be redirected to the AWS CloudFormation creation form in the AWS CloudFormation console.
16. Name your stack.
 - A security group will be created with the same name. You will access this group in [Step 3: Deploy Weka using CloudFormation](#).
17. In the **Network Configuration** section, provide the following information in the specified fields.
 - a. **VPC:** Choose the VPC of your Nimble Studio. For example: `<your-studio-name>`
 - b. **Subnet:** Choose the Workstations subnet.
 - c. **Network Topology:** Choose the network topology that corresponds to your Nimble Studio. For example: `NAT internet routing`
18. In the **Amazon EC2 Configuration** section, select the SSH key that you created in step 5 of this section.
19. Notice the value in the **API Token** field. This is automatically filled by Weka. You will need this token later to **Mount the Weka FS on Linux Workstations**.
20. In the **Admin Password** section, enter a secure password.
 - CloudFormation recommends using parameters for sensitive data. For more information about using parameters with CloudFormation, see [Security best practices for AWS CloudFormation](#) in the AWS CloudFormation User Guide.
21. Select the check box to acknowledge that CloudFormation might create IAM resources.
22. Select **Create stack**.



Your stack will be successfully created when its status changes to **CREATE_COMPLETE**. It should take about 40 minutes to deploy.

Step 4: Update security groups

Update the security group network rules to allow Nimble Studio workstations to connect to the Weka cluster.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Security Groups** in the left navigation pane in **Network & Security**.
3. Select the security group with the name that contains WorkstationsEgress.
4. Navigate to the **Details** section. Notice the **Security group ID** there because you will need it in *step 9* of this section.
5. Select the security group with the same prefix as the stack that you created in *step 16* of [Step 3: Deploy Weka using CloudFormation](#). For example: WEKA-STACK-
6. Go to the **Inbound rules** tab.
7. Select **Edit inbound rules**.
8. Select **Add rule**.
9. Find the rule with **Type: All traffic**. Enter the security group ID that you found in *step 4* in the **Source** field of that rule.
 - Select the name of the security group when it appears in the dropdown at the bottom of the field.
10. Add another rule with the following information:
 - a. **Type: Custom TCP**

b. **Port Field:** 14000 - 14100

c. **Source:** Anywhere-IPv4

11. Select **Save rules**.

Your workstations can now connect to the Weka cluster.

Step 5: Linux setup

The following two sections explain how to add two initialization scripts. You will need these scripts in the **Add Sudo access** section, when you create the launch profile.

Mount the Weka FS on Linux workstations

Add a script to initialize your Weka cluster when you launch a workstation.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select the instance named Backend-0.
 - Notice its IP in **Private IPv4**. You will need this in *step 9b* of this section to write the initialization script.

Instance ID	Private IP	Status	Type	Check Status	Alarms	Actions
WEKA-STACK-Backend-0	172.31.10.10	Running	i3en.2xlarge	2/2 checks passed	No alarms	Details
WEKA-STACK-Backend-3	172.31.10.11	Running	i3en.2xlarge	2/2 checks passed	No alarms	Details
WEKA-STACK-Backend-2	172.31.10.12	Running	i3en.2xlarge	2/2 checks passed	No alarms	Details
WEKA-STACK-Backend-1	172.31.10.13	Running	i3en.2xlarge	2/2 checks passed	No alarms	Details
WEKA-STACK-Backend-4	172.31.10.14	Running	i3en.2xlarge	2/2 checks passed	No alarms	Details
WEKA-STACK-Backend-5	172.31.10.15	Stopped	g4dn.xlarge	-	No alarms	Details
WEKA-STACK-Backend-6	172.31.10.16	Stopped	g4dn.xlarge	-	No alarms	Details

Instance: i-1234567890 (WEKA-STACK-Backend-0)

[Details](#) [Security](#) [Networking](#) [Storage](#) [Status checks](#) [Monitoring](#) [Tags](#)

Instance summary [Info](#)

Instance ID: i-1234567890 | Public IPv4 address: -

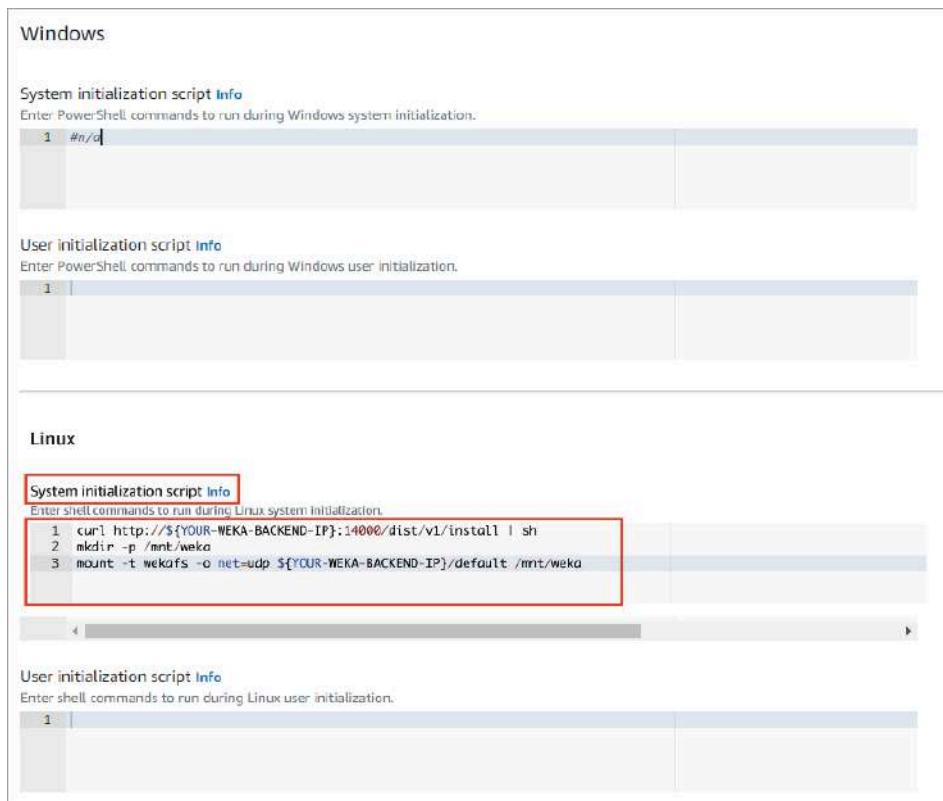
Private IPv4 address copied

4. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
5. Choose **Studio resources** in the left navigation pane.

6. Choose **Add in File Storage**.
7. Provide the following information in the specified fields.
 - a. **File storage name:** (recommended) **Weka**
 - b. (Optional) **File storage description:** Give your file storage a description.
 - c. **Storage type:** Select Custom
 - d. Enter **not applicable** in all of the **File storage configuration** fields.
8. Enter the following code in the **System Initialization script** field in the **Linux** section. This code will automatically allow every new instance to install the Weka client software, create the mounting point, and mount the Weka file system to the mount point.
 - a. Replace \${your-weka-token} with the token that you saved from *step 19 of [Step 3: Deploy Weka using CloudFormation](#)*.
 - b. Replace \${YOUR-WEKA-BACKEND-IP} with the backend private IPv4 that you found in *step 2*.

```
curl https://${your-weka-token}@get.weka.io/dist/v1/install/3.13.1/3.13.1 |  
sh  
mkdir -p /mnt/weka  
mount -t wekafs -o net=udp ${YOUR-WEKA-BACKEND-IP}/default /mnt/weka
```

9. Enter #n/a in the **Windows System initialization script** section.



10. Select the **WorkstationEgress** in the **Security groups** section.

11. Read the terms and conditions and if you agree:

- Select the check box next to **I understand that Nimble Studio will access my existing file storage.**

12. Choose **Save connection parameters**.

Add Sudo access

To get the required superuser privileges in a workstation, you need to create another Initialization script. This script gives Sudo access to the Admin user at startup.

Follow the [Provide Superuser access for Linux users](#) tutorial to create the Sudo access component.

Create a launch profile with superuser privileges

To benchmark and troubleshoot the Weka cluster, create a new launch profile with the Weka initialization script and superuser access.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.

2. Select **Launch profiles**.
3. Select the **Workstation-Default Launch Profile**.
4. Choose **Actions**. Then choose **Copy to new**.
5. Name the launch profile Weka setup.
6. Enter a description for your launch profile.
7. In the **Amazon Machine Image (AMI)** section, select **NimbleStudioLinuxStreamingImage**. Clear all other streaming profiles.
8. In the **Studio custom configuration** section, select **Sudo Access** and **Weka Init**.

 **Note**

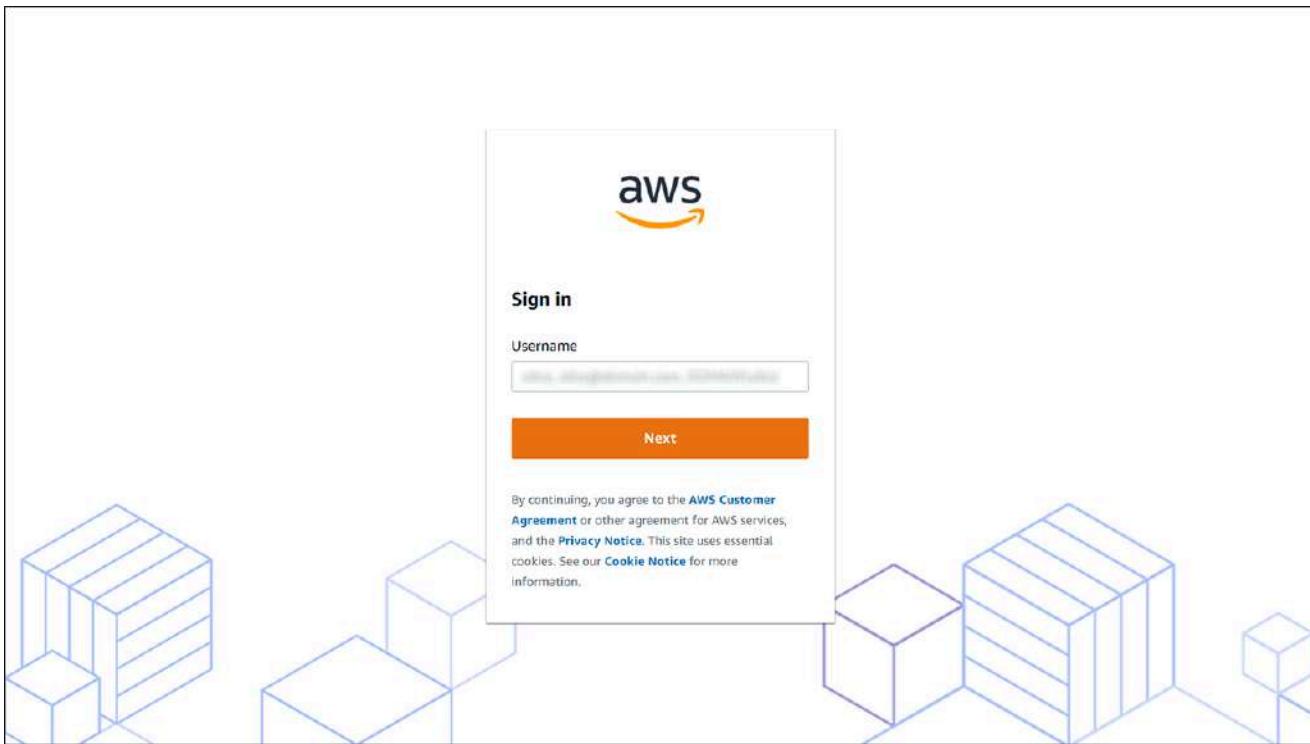
If you don't see one of those options, follow the instructions in [Mount the Weka FS on Linux workstations](#) and [Add Sudo access](#).

9. Select **Create launch profile**.

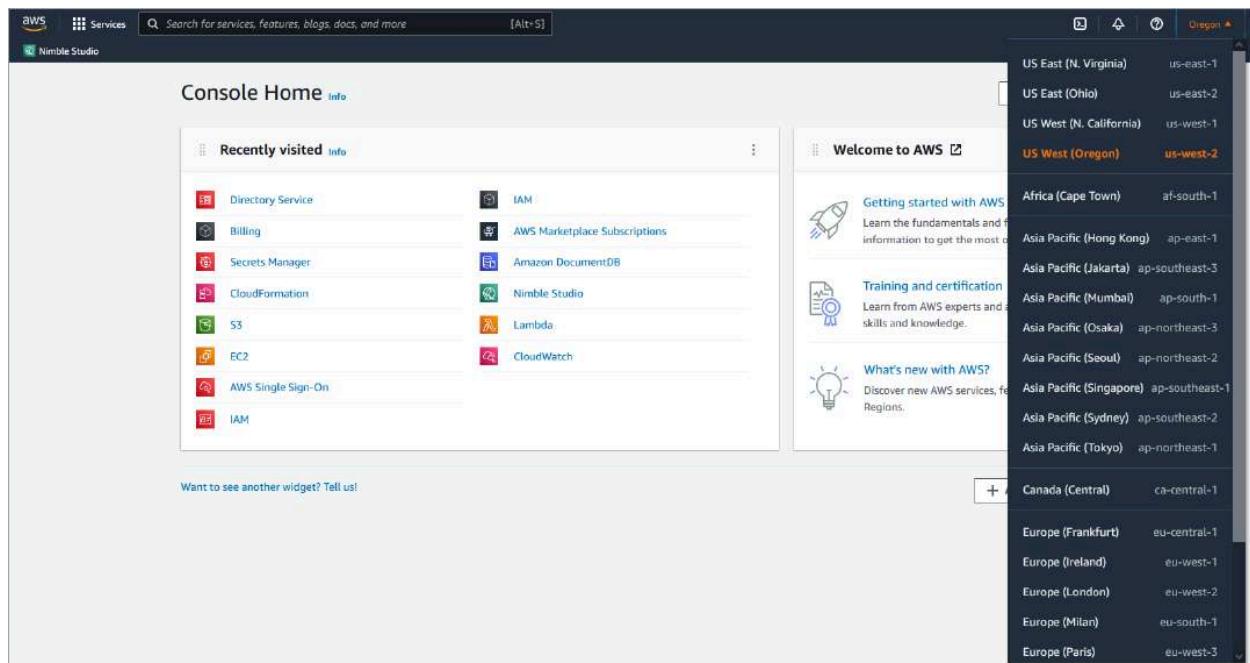
Benchmark the Weka file system from a Linux client

To benchmark the Weka cluster that you just configured, connect to a workstation. Use the launch profile with superuser privileges and the initialization script that mounts the Weka file system automatically.

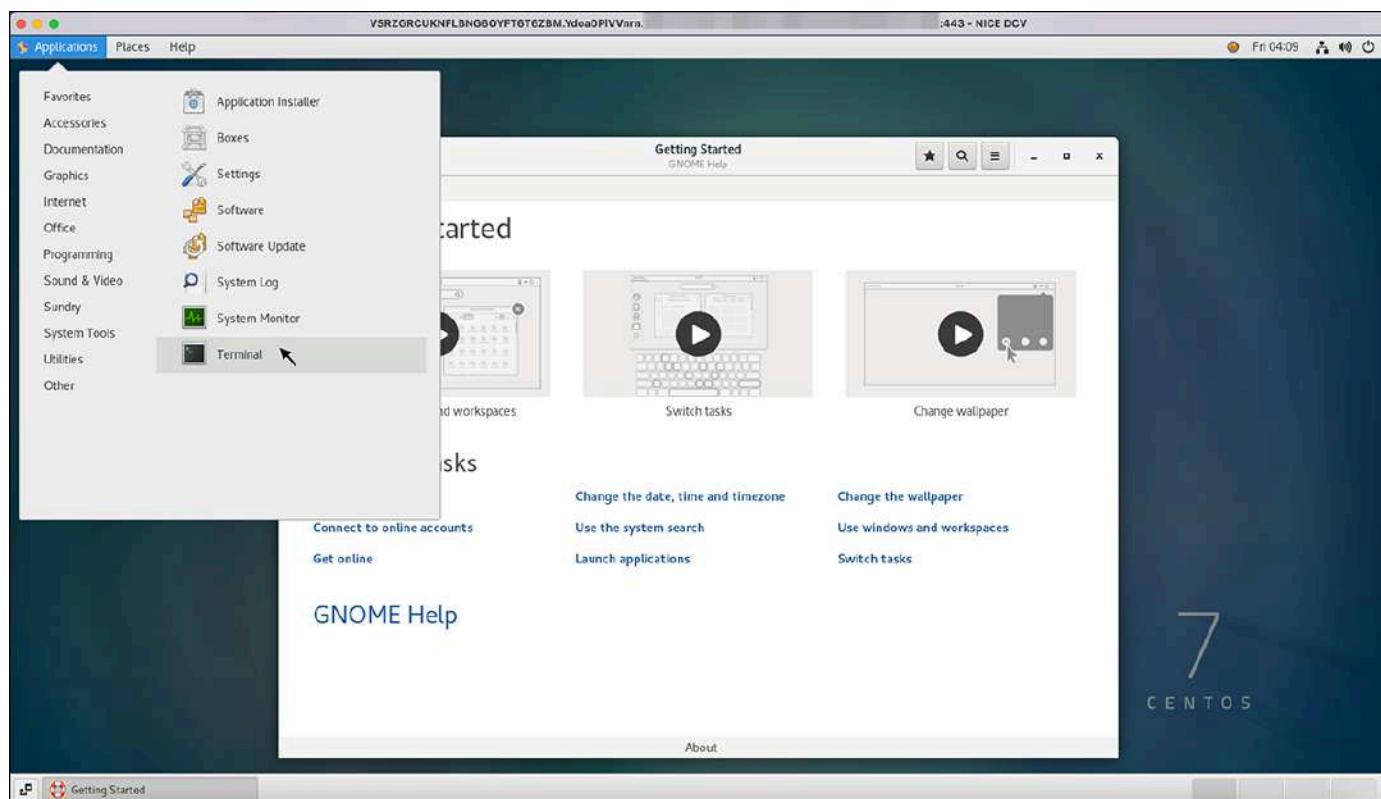
1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio manager** in the left navigation pane.
3. On the **Studio manager** page, choose **Go to Nimble Studio portal**.
4. Sign in to the Nimble Studio portal using your AWS Managed Microsoft AD administrator credentials.
 - a. Enter **Admin** as the user name. Use the password that you set up during StudioBuilder deployment.



- b. If you forgot your password, do the following:
- Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
 - In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



- iii. Select the **Directory ID** for your studio's Active Directory.
 - iv. Choose **Reset user password**.
5. Choose the launch profile that you created in the previous section, **Create a launch profile with superuser privileges**.
 6. Select **Launch with...** in the options dropdown.
 7. In the **Launch Options** pop-up, select **NimbleStudioLinuxStreamImage**.
 8. Choose **Launch native client in Streaming Preference**.
 9. Select **Launch**.
 10. Sign in to the workstation with your Nimble Studio credentials.
 11. Select **Applications** in the menu bar. Then choose **System Tools** and **Terminal**.



12. Run the following command to check if the Weka file system is mounted: `mount -t wekafs`
 - It will check for all the wekafs volumes. One line will display: default on /mnt/weka type wekafs (rw, realtime, write)
13. Alternatively, run the following command to check if the Weka file system is mounted: `df -h`
 - If you see a line containing /mnt/weka, it means that your cluster has been mounted successfully at startup.

14. Run the following command to install fio: `sudo yum install fio`
15. Run the following command to run fio to benchmark the file system:

```
fio --name=1m --rw=randread --bs=1m --ioengine=posixaio --direct=1 --directory=/mnt/weka --nrfiles=200 --size=1GB --iodepth=1 --runtime=10000 --numjobs=16 --group_reporting --time_based --create_serialize=0
```

If you see Starting 16 processes Jobs: 16, it means that your Weka cluster has been mounted successfully.

Step 6: (Optional) Windows setup

Complete the Linux setup before continuing the Windows setup.

Access Weka Cluster backend nodes using SSH

You can connect to one of the Weka backend nodes using SSH. This can be useful for troubleshooting and for setting up custom configurations on your Weka cluster.

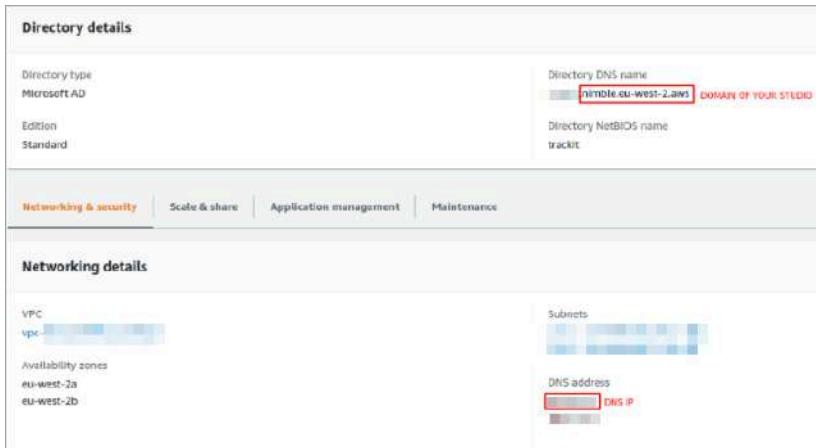
1. Open the key (. pem file) that you downloaded in [Step 3: Deploy Weka using CloudFormation](#).
2. Open the key with any text editor application and copy the text.
3. Go back to the terminal of your workstation and run the following command: `nano key.pem`
4. Exit by entering **CTRL + X**.
 - It will ask you to save. Enter **y**.
5. Run the following command: `chmod 400 key.pem`
6. Run the following command: `ssh -i ${key-text} ec2-user@${backend-0-IPv4}`
 - a. Replace \${key-text} with the text that you copied in *step 3*.
 - b. Replace \${backend-0-IPv4} with the IP that you found in *step 2 of Mount the Weka FS on Linux workstations*.

If your command prompt starts with `ec2-user@...` you have successfully connected to the backend using SSH.

Configure the backends

Before you build the Samba server, configure the Weka backends.

1. Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
2. Select **Directories**.
3. Select the ID of the AWS Managed Microsoft AD that you want to use.
4. Navigate to **Directory details** and notice your studio's **Directory DNS name**.
5. In **Networking details**, notice the IP of the **DNS address** of the AWS Managed Microsoft AD.



6. Connect to one of the backends using SSH by following the instructions in [Access Weka Cluster backend nodes using SSH](#).
7. Edit the `resolv.conf` file to make the backends reach the AWS Managed Microsoft AD server by running the following command: `sudo nano /etc/resolv.conf`
 - a. Replace the nameserver IP with the IP of the **DNS address** of the AWS Managed Microsoft AD that you found in *step 4*.
 - b. Add a search line with the domain of your Nimble Studio. This is the **Directory DNS name** that you found in *step 5*.
8. Exit by entering **CTRL + X**.
 - It will ask you to save. Enter **y**.
9. Apply the same configurations to the other backends by running the following command:
`sudo nano ~/.ssh/id_rsa`
 - Enter the content of the `.pem` file you downloaded in [Step 3: Deploy Weka using CloudFormation](#).

10. Exit by entering **CTRL + X**.

- It will ask you to save. Enter **y**.

11. Sign in to the Weka client by running the following command: `weka user login`

12. Enter **admin** for the user name and enter the password that you defined in [Step 3: Deploy Weka using CloudFormation](#).

13. Copy the `resolv.conf` file to all the backends by running the following command:

```
for i in `weka cluster host -b -o ips`; do scp /etc/resolv.conf $i:./; ssh $i "sudo cp ./resolv.conf /etc/resolv.conf"; done
```

14. Enter **yes** and press the enter (or return) key.

Keep your SSH connection active for the next section.

Build the Samba server

1. Run the following command to create the Samba shared directory: `sudo mkdir /mnt/weka/smb`

2. Run the following command to give every user permission to read, edit, create, delete files in this directory: `sudo chmod 777 /mnt/weka/smb`

3. Create the Weka server by running the following command:

- Replace `<DirectoryDNS-name>` with the **Directory DNS name** that you found in step 4 of [Configure the backends](#).

```
weka smb cluster create wekaSamba <DirectoryDNS-name> --samba-hosts 0,1,2,3,4  
--idmap-backend rid
```

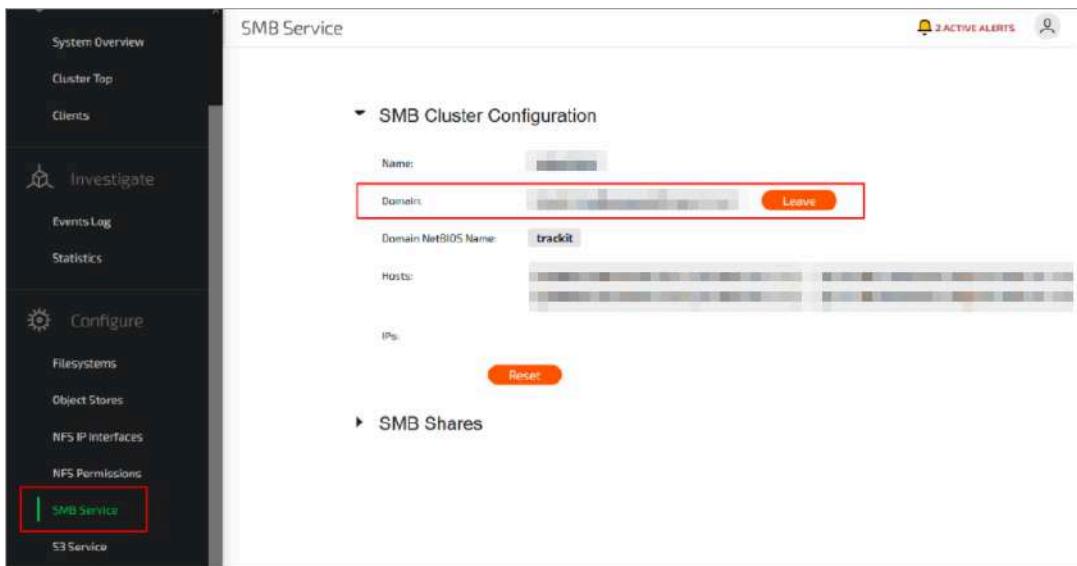
4. Wait for the Samba server to be ready. You can see its status by running the following command: `weka smb cluster status`

5. Join the server that you just built by running the following command: `weka smb domain join Admin`

6. Enter your Nimble Studio password.

- If it doesn't work with the user name Admin, try with the user name and password that you used to sign in to the Nimble Studio portal.

7. If you text is returned saying "Joined <your-studio-name> to dns domain", it means that you have successfully joined the Samba server.
8. Open an internet browser to create an SMB share.
9. Go to [http://\\${IP-OF-BACKEND-0}:14000](http://${IP-OF-BACKEND-0}:14000)
 - Replace \${IP-OF-BACKEND-0} with the IP address of the Weka backend.
10. To sign in, use **admin** as a user and use the password that you defined in *step 20 of Step 3: Deploy Weka using CloudFormation*.
11. Open the menu and select **SMB Service**. If everything worked in the previous steps, the domain will already be Joined.



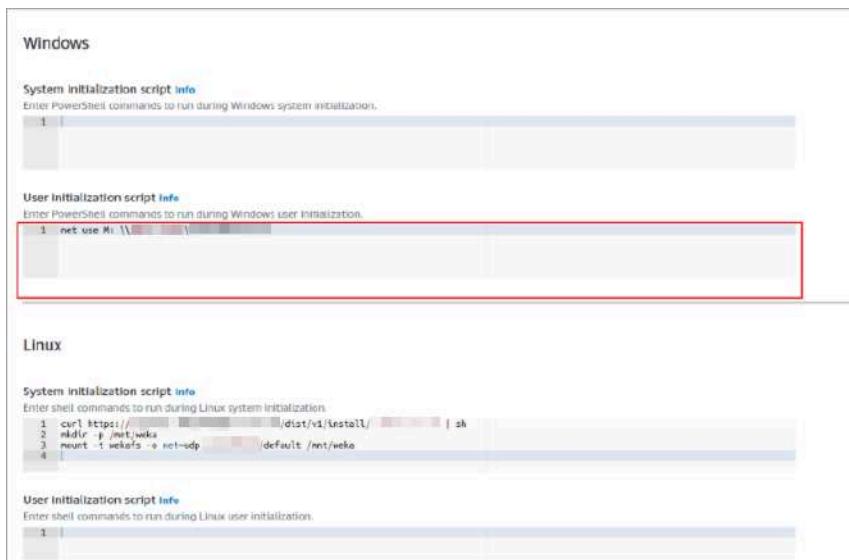
12. Select **SMB Shares**. Then select **Create Share**.
13. In the **Create Share** pop-up, provide the following information in the specified fields.
 - a. **Name:** Enter a name for this Samba share. Notice the name, because it will be used in the next section.
 - b. **Description:** Enter a description for this Samba share.
 - c. **Filesystem:** **default**
 - d. **Path:** **/smb**
14. **Enable ACLs.**
15. Select **Create**.
16. Close the SSH connection by entering **exit** and press the enter (or return) key.
17. Remove the .pem file by running the following command in your terminal: **rm -rf key.pem**

- You need to remove the . pem file so that your credentials aren't contained in the instance. This keeps your instances secure.

Initialization script to connect to the Samba server on Windows

You can connect the Samba server that you just built to the Windows workstations at startup. To do this, modify the initialization script that you created to mount the storage at Linux startup.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select the Backend-0 instance and notice its IP in **Private IPv4**.
4. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
5. Choose **Studio resources** in the left navigation pane.
6. Go to the **File Storage configuration** section. Find and select the initialization that you created in [Mount the Weka FS on Linux workstations](#).
7. Choose **Edit**.
8. Enter the following code in the **User initialization script** field for **Windows** in the **Initialization scripts** section: `net use M: \\${YOUR-WEKA-BACKEND-IP}\${Name-of-samba-share}`
 - a. Replace `\${Name-of-samba-share}` with the name of the Samba share that you created in the previous section.
 - b. Replace `\${YOUR-WEKA-BACKEND-IP}` with the backend private IPv4 that you found in *step 3*.



9. Read the terms and conditions and if you agree:

- Select the check box next to **I understand that Nimble Studio will access my existing file storage.**

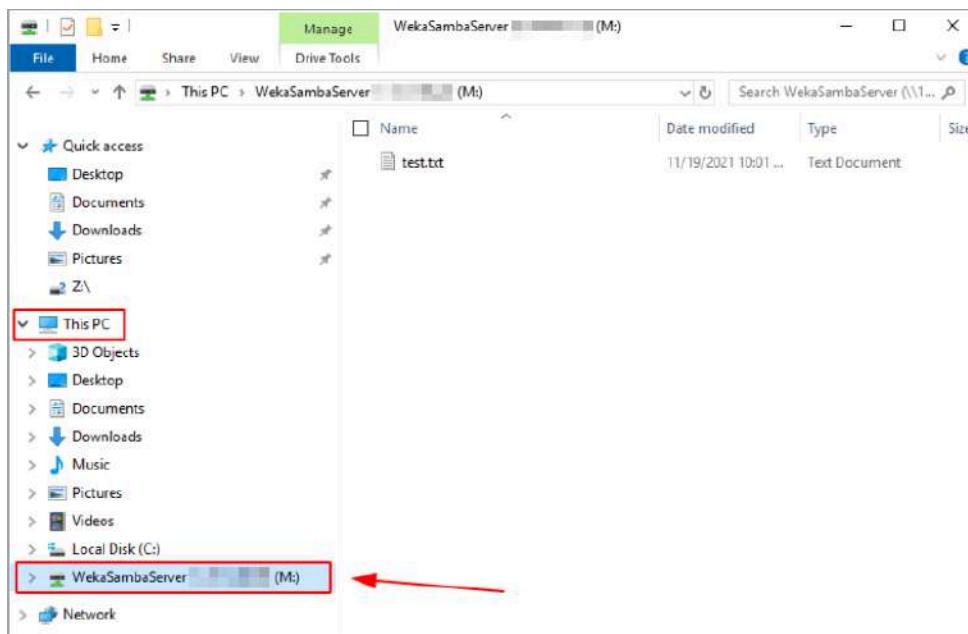
10. Choose **Save connection parameters**.

Benchmark the Weka file system from a Windows client server

Test your Samba server by following these instructions.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Select **Launch profiles**.
3. Select Weka setup.
4. In the **Amazon Machine Image (AMI)** section, select NimbleStudioWindowsStreamingImage.
5. Select **Update launch profile**.
6. Select **Studio manager**.
7. Choose **Go to Nimble Studio portal** and sign in.
8. Choose the Weka setup launch profile.
9. Select **Launch with...** in the options dropdown.
10. In the **Launch Options** pop-up, select NimbleStudioWindowsStreamImage.
11. Choose **Launch native client** in the **Streaming preference** dropdown.

12. Select **Launch**.
13. Sign in with your Nimble credentials.
14. Open the **File explorer**.
15. Select **This PC**. Your Samba share should be connected.
16. Select it and try to create a text file. This verifies that you have the correct permissions.



(Optional) Test storage performance

You can test the storage performance by following these steps.

1. Open PowerShell.
2. Run the following commands:
 - a. `$client = new-object System.Net.WebClient`
 - b. `$client.DownloadFile("https://github.com/microsoft/diskspd/releases/download/v2.0.21a/DiskSpd.zip",".\DiskSpd-2.0.21a.zip")`
 - c. `Expand-Archive -LiteralPath .\DiskSpd-2.0.21a.zip -DestinationPath C:\DISKSPD`
 - d. `cd C:\DISKSPD\amd64`
3. Create a large text file to test the storage performance by running the following command:
 - `fsutil file createnew M:\1.txt 2000000000`

4. Download this file to test the storage performance:

- .\diskspd -d15 -F1 -w0 -r -b4k -o10 M:\1.txt

After it's done, you can see the throughput speed of the storage in the fourth column named MiB/S.

Total IO	thread	bytes	I/Os	MiB/s	I/O per s	file
	0	2068725760	505060	131.52	33669.82	M:\1.txt (1907MiB)
total:		2068725760	505060	131.52	33669.82	
Read IO	thread	bytes	I/Os	MiB/s	I/O per s	file
	0	2068725760	505060	131.52	33669.82	M:\1.txt (1907MiB)
total:		2068725760	505060	131.52	33669.82	
Write IO	thread	bytes	I/Os	MiB/s	I/O per s	file
	0	0	0	0.00	0.00	M:\1.txt (1907MiB)
total:		0	0	0.00	0.00	

ps C:\DISKSPD\amd64>

Troubleshooting

Building the Samba server.

Error: Trying to set an invalid samba configuration: Cluster is already configured, please reset first.

If you receive this error after running the command, weka smb cluster create wekaSamba <DirectoryDNS-name> --samba-hosts 0,1,2,3,4 --idmap-backend rid, perform the following steps.

1. Reset the Samba configuration with the following command: weka smb cluster destroy
2. Enter **yes** and press the enter (or return) key.
3. Run the following command again: weka smb cluster create wekaSamba <DirectoryDNS-name> --samba-hosts 0,1,2,3,4 --idmap-backend rid

Software setup and updates for Amazon Nimble Studio

In this administrator tutorial series, you'll learn about software setup and updates for a Nimble Studio cloud studio. Setting up specific software versions and tools for your team is important for maintaining a working cloud studio. You'll accomplish this by updating Amazon Machine Images (AMIs) with new software. You'll also learn to set up license servers and environment variables so that your artists' virtual workstations can find license files.

Topics

- [Update AMIs: Setting up](#)
- [Update Windows workstation AMI](#)
- [Update a Linux workstation AMI](#)
- [Update a Windows worker AMI](#)
- [Update a Linux worker AMI](#)
- [Software specific installation tips](#)
- [Working with license servers](#)

Update AMIs: Setting up

Setting up the right software for your team is an important part of maintaining a working studio. You can accomplish this by updating your Amazon Machine Images (AMIs) with specific software versions or new tools.

In the [Creating launch profiles](#) tutorial, you learned how to select AMIs for your launch profiles. The AMIs assigned to launch profiles determine which software and operating systems are available to your artists when they start a streaming session.

In this tutorial, the [General setup](#) section will show you what you need to do before you can update your AMIs. You will learn how to create an Amazon Simple Storage Service (Amazon S3) bucket to store your installers, a security group, IAM policies, and roles. After you complete this setup, you can follow the instructions to [Update AMIs for your operating system](#).

Contents

- [Prerequisites](#)

- [General setup](#)
- [Update AMIs for your operating system](#)
- [Software-specific instructions](#)
- [Troubleshooting](#)
- [Related resources](#)

Prerequisites

Complete the following before you begin this tutorial.

- Deploy a studio from Nimble Studio, as described in [Deploying a new studio with StudioBuilder](#).
- Confirm that you have your default virtual private cloud (VPC) from Amazon Virtual Private Cloud (Amazon VPC) in the Region where your studio is deployed. If you suspect that you might have deleted it, see [View your default VPC and default subnets](#). If you need to create a new default VPC, see [Creating a default VPC](#).

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open <https://portal.aws.amazon.com/billing/signup>.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, [assign administrative access to an administrative user](#), and use only the root user to perform [tasks that require root user access](#).

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to <https://aws.amazon.com/> and choosing **My Account**.

Create an administrative user

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the [AWS Management Console](#) as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

For help signing in by using root user, see [Signing in as the root user](#) in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

For instructions, see [Enable a virtual MFA device for your AWS account root user \(console\)](#) in the *IAM User Guide*.

Create an administrative user

1. Enable IAM Identity Center.

For instructions, see [Enabling AWS IAM Identity Center](#) in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to an administrative user.

For a tutorial about using the IAM Identity Center directory as your identity source, see [Configure user access with the default IAM Identity Center directory](#) in the *AWS IAM Identity Center User Guide*.

Sign in as the administrative user

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

For help signing in using an IAM Identity Center user, see [Signing in to the AWS access portal](#) in the *AWS Sign-In User Guide*.

General setup

Before you update an AMI, do a general setup in your account, including setting up cloud storage to store your installers. You only need to do these steps one time.

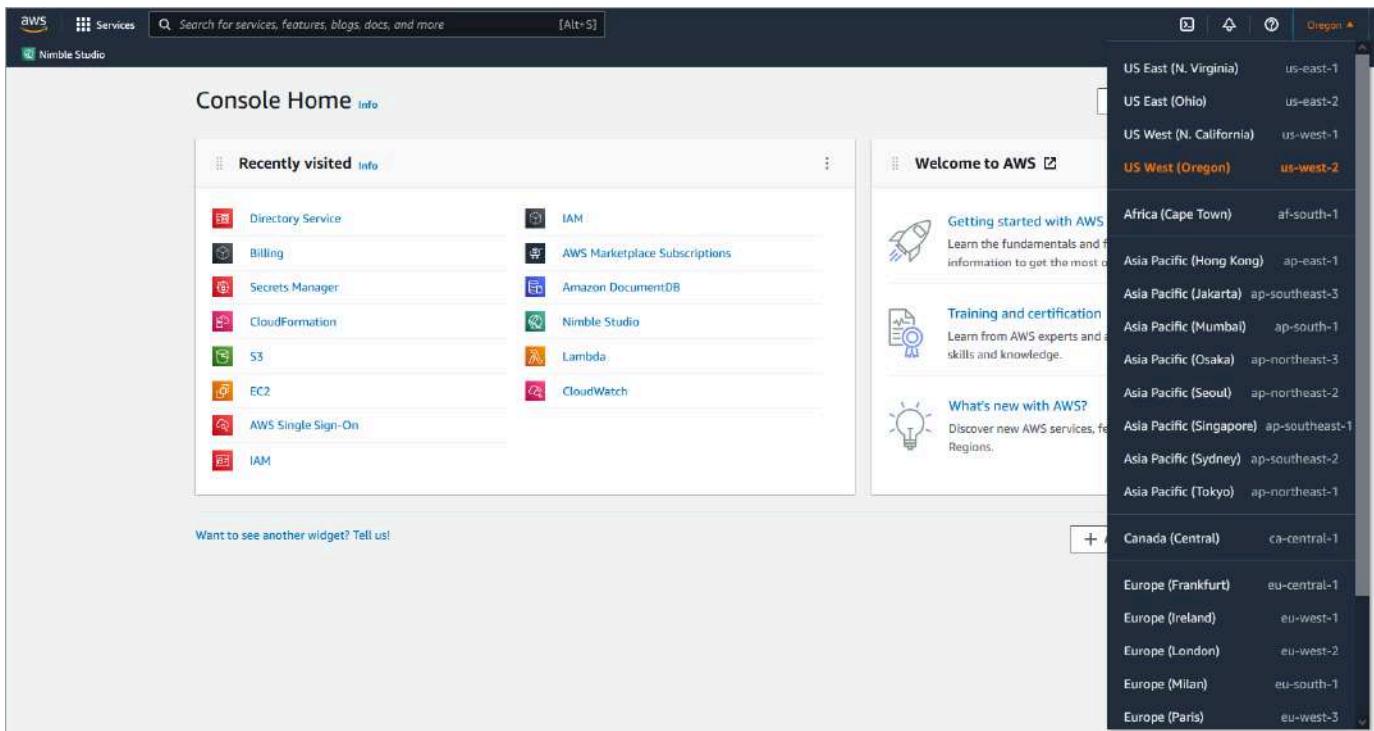
If you have already completed the following steps, you can skip to [Update AMIs for your operating system](#).

Step 1: Create an Amazon Simple Storage Service (Amazon S3) bucket to store your installers

Create an Amazon S3 bucket to store your installers. The storage containers used by Amazon S3 are called **buckets**, which are similar to folders or directories. For more information, see [Buckets overview](#) in the [Amazon Simple Storage Service User Guide](#).

Use an Amazon S3 bucket as the central storage for installers to help you track which versions of software that you have installed on your AMIs. You might not always want to use the latest version of a given application. Storing installers in one location for the versions that you do want to use will make it easier to check that you're installing the same version of software on all your AMIs.

1. Sign in to the AWS Management Console and open the [Amazon S3](#) console.
2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



3. On the buckets page, choose **Create bucket**.
4. For **Bucket name**, enter a name, such as <your-studio-name>-installers.
5. Select the AWS Region that your studio is deployed in.
6. We recommend using the default settings for everything else on this page. However, you can change the settings based on your needs. For more information about S3 bucket settings, see [Creating a bucket](#) in the Amazon Simple Storage Service User Guide.
7. Choose **Create bucket**.
8. After your bucket is created, find it in the list, then choose its name.
9. To create folders for your installers, choose **Create folder**.
10. To upload installers from your local machine to S3, choose **Upload**.
 - The next image shows the page for the bucket that you created. Notice that it has a subfolder **blender** with the Blender 2.92.0 Windows installer uploaded to it.

Step 2: Create IAM policies

To update an AMI, launch an instance and update its software. For that instance to have access to the AWS services that it needs, create an AWS Identity and Access Management (IAM) role with the correct permissions policies attached. This will allow you to connect to the instance, access the Amazon S3 bucket that you created in the last step, and access the license for NICE DCV (for instances with a GUI).

Amazon provides a policy to connect to your instances with Session Manager, but create policies to provide access to your S3 bucket and to the NICE DCV license file for your Region.

Create a policy for the installers bucket

First you will create a policy that allows read-only access to the Amazon S3 installers bucket you created in the last step. Follow the instructions in [Creating policies on the JSON tab](#) in the IAM User Guide while using the following information.

To create a policy for the installers bucket

1. Replace the JSON data in the text field with the following JSON text. Make sure to change the two example lines in the following code **DOC-EXAMPLE-BUCKET** to the name of the bucket that you just created in **Step 2**.

{

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "s3>ListBucket"
        ],
        "Resource": "arn:aws:s3:::<BUCKET-NAME>"
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetObject"
        ],
        "Resource": "arn:aws:s3:::<BUCKET-NAME>/*"
    }
]
}
```

2. Name your new policy **NimbleStudioInstallersS3ReadOnly**.

Create a policy for the NICE DCV license

Next, you will create a policy to allow instances that use NICE DCV to access the NICE DCV license file for your region.

To create a policy for the NICE DCV license

1. Choose **Create policy**.
2. Select the **JSON** tab.
3. Replace the JSON data in the text field with the JSON provided here: Licensing the [NICE DCV Server - NICE DCV](#).
 - Make sure that you replace the region placeholder with your AWS Region (for example, us-west-2).
4. Choose **Next: Tags**.
5. Choose **Next: Review**.
6. Name your new policy **NimbleStudioDCVLicenseS3Access-*region*** . Replace "region" with the AWS Region that your studio is located in.
7. Add a description.

8. Choose **Create policy**.

Step 3: Create an IAM role

After you create the IAM policies from the previous step, create the IAM role that you will attach to the instances that you use to create AMIs.

Follow the instructions in [Creating a role for an AWS service \(console\)](#) in the IAM User Guide while using the following information.

To create an IAM role

1. For the service that you want to allow to assume this role, select **EC2**.
2. For permissions policies, search for S3 and select the two policies that you created in the previous step from the list. Make sure to select the check box for each policy. Don't select the name.
3. Also for permissions policies, search for SSM and select the check box for **AmazonSSMManagedInstanceCore**
4. (Optional) Enter **Studio** as the key and enter <your-studio-name> as the value.
5. For **Role name**, enter **Nimble_Studio_Build_AMI**. (Optional) Add a description.

Step 4: Create a security group

Next, create a security group that allows you to connect to your instance from your local machine.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Security Groups** in the left navigation pane in **Network & Security**.
3. Choose **Create security group**.
4. For **Basic details**, fill out the following:
 - a. For **Security group name**, enter **Nimble_Studio_Build_AMI**.
 - b. For **Description**, enter something like **Security group for building Nimble Studio AMIs**.
 - c. For **VPC**, the default VPC should already be selected. If not, select it from the dropdown list.

5. Create an inbound rule for this security group by choosing **Add rule**, and then filling out the information listed in this table.

Security group inbound rules

Type	Protocol	Port range	Source	Description
Custom TCP	TCP	8443	Consult with IT	NICE DCV connection

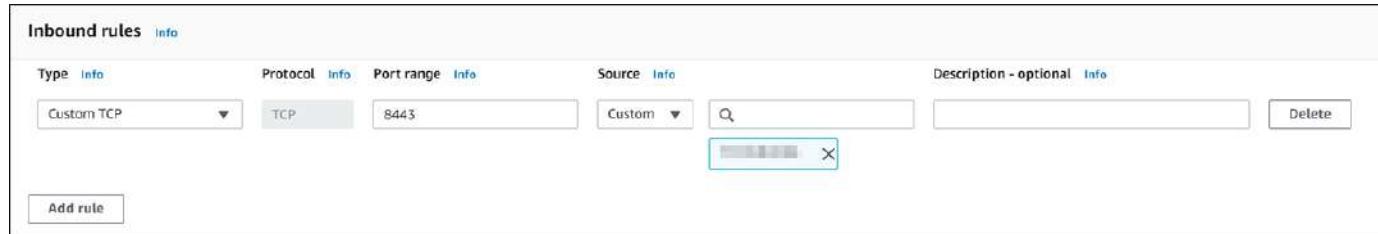
Important

We recommend that you don't use a source value of **Anywhere** because that allows any IP address to connect to your rule's port(s).

To determine the ideal source value to use for your security group, based on your studio's security compliance goals, consult with your studio administrator or IT team.

You can also specify a range of IP addresses by selecting **Custom** and using CIDR block notation (xxx.xxx.xxx.xxx/n). This will limit traffic by only allowing a range of IP addresses to connect. Your studio administrator or IT team can help you determine the correct CIDR block to use.

Your **Source** value will vary depending on the recommendation of your studio administrator or IT team.



The screenshot shows the 'Inbound rules' configuration screen. It displays a single rule entry with the following details:

- Type: Custom TCP
- Protocol: TCP
- Port range: 8443
- Source: Custom (dropdown menu open)
- Description - optional: NICE DCV connection

At the bottom left is a 'Delete' button and at the bottom center is an 'Add rule' button.

6. Navigate to the bottom of the screen and choose **Create security group**. After this process has been successfully completed, a green banner will confirm this at the top of the menu bar.

Update AMIs for your operating system

For your artists to access a new version of software through a launch profile, first update your software on an AMI that you create. After you create and update that AMI, you can add it as an option in a launch profile. Then, you can create different AMIs for different versions of software or for specific tasks, such as texture painting or lighting.

The following tutorials provide instructions for updating AMIs for specific operating systems:

- [Update Windows workstation AMI](#)
- [Update a Linux workstation AMI](#)
- [Update a Windows worker AMI](#)
- [Update a Linux worker AMI](#)

Software-specific instructions

Some software that you install on your AMI might require special steps for it to work correctly. For more information, see [Software specific installation tips](#).

Troubleshooting

Reduce the AMI size.

Error: I got this error when trying to add an AMI to my studio: You cannot add an AMI that exceeds 500 GBs.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select the instance that you created the AMI with, then connect to it with NICE DCV.
4. There are two main areas that you can reduce the AMI size:
 - a. Downloaded files.
 - b. Installed applications
5. After you remove the files that you don't need, create a new version of your AMI, encrypt it, then add it to your studio.

Remove AMIs or increase your quota.

Error: I got this error when trying to add an AMI to my studio: Your studio exceeds the custom streaming image quota.

To remove AMIs from your studio

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. Navigate to **Amazon Machine Image**.
4. Select the AMI that you want to remove.
5. Choose **Action**. Then choose **Remove**.
6. Enter **Remove** and choose **Remove**.

To increase your custom streaming image quota

1. Sign in to the AWS Management Console and open the [Service Quotas](#) console.
2. Choose **AWS services** in the left navigation pane.
3. Search for Nimble Studio.
4. Choose **Amazon Nimble Studio** from the list.
5. Select **Custom streaming images per studio**.
6. Choose **Request quota increase**.
7. For **Change quota value**, enter the number of custom streaming images you would like to have in your studio.
8. Choose **Request**.

Related resources

- [Security group rules](#)

Update Windows workstation AMI

This tutorial covers the steps to update a Windows workstation Amazon Machine Image (AMI) with new software.

Note

This tutorial applies to both WindowsServer2019 AMIs and WindowsServer2022 AMIs.

Contents

- [Prerequisites](#)
- [Step 1: Create a customer managed key](#)
- [Step 2: Launch an instance with Windows AMI](#)
- [Step 3: Restart the NICE DCV session as an administrator](#)
- [Step 4: Connect with NICE DCV](#)
- [Step 5: Download and run installers](#)
- [Step 6: Prepare your instance for AMI creation](#)
- [Step 7: Create a new AMI](#)
- [Step 8: Update launch profiles](#)
- [Step 9: Update the DCV server](#)
- [Troubleshooting](#)
- [Related resources](#)

Prerequisites

- Complete the prerequisites and follow the steps in the [Update AMIs: Setting up](#) tutorial.
- Check your service quota for running On-Demand Instances (limit name "G"). For instructions about how to do that, see [Check AWS service quotas](#).
- This tutorial uses [NICE DCV](#) to connect to an instance. Download the latest [DCV client](#) and install it on your local machine before you begin. For more information, see the [NICE DCV User Guide](#).

Step 1: Create a customer managed key

Create a customer managed key that will be used to encrypt your AMI. You only need to create a customer managed key one time. If you already created one, you can reuse it and skip to [Step 2: Launch an instance with Windows AMI](#).

Follow the instructions in the [Creating symmetric encryption KMS keys \(console\)](#) tutorial in the AWS Key Management Service Developer Guide. Follow these additional steps:

1. For the **Alias** in the **Add labels** section, include the name of your studio, and mention custom AMI. For example: <your_studio_name>-customAMI-key
2. For **Key administrators** in the **Define key administrative permissions** section, enter and select the name of the administrator user that you used to sign in to the AWS Management Console. For example: aws-admin
3. For **This account** in the **Define key usage permissions** section, enter and select the same administrator user that you selected in the previous step.

Step 2: Launch an instance with Windows AMI

To launch an instance using a Windows AMI that you want to update, you have two choices:

- If you've never created a Windows AMI for your studio, follow the steps in the [Start with the default Windows AMI from StudioBuilder](#) section.
- If you, or a trusted team member, have already made another Windows AMI for your studio that you would like to start from, see [Start with another Windows AMI already created in your account](#) in this tutorial.

 **Note**

If you created a custom AMI, you don't need to subscribe to it in the AWS Marketplace.

Start with the default Windows AMI from StudioBuilder

1. Go to the [AWS Marketplace](#).
2. Search for **Nimble Studio Windows Workstation**.
3. Choose **Nimble Studio Windows Workstation** from the search results.
 - You can select a WindowsServer2019 AMI or a WindowsServer2022 AMI.
4. On the AWS Marketplace page for the Nimble Studio Windows Workstation, select **Continue to Subscribe**.
5. Read the terms and conditions. Then choose **Accept Terms**.

6. After the subscribe request has finished processing, select **Continue to Configuration**.
7. In the **Region** dropdown menu, select the AWS Region that your studio is located in and select **Continue to Launch**.
8. In the **Choose Action** dropdown menu, select **Launch through EC2** and select **Launch**.
 - The Amazon EC2 console will open and will guide you through the rest of the launch process.

After the launch process is complete, go directly to the [To launch an instance in the default Amazon VPC](#) section to continue.

Start with another Windows AMI already created in your account

If you already know the AMI ID for the AMI that you would like to update in your account, skip to following section of this tutorial: [To find the AMI ID of the Windows AMI to update](#).

Otherwise, to find the AMI ID by looking at the launch profile that uses it, continue to the next step.

To find the AMI ID of the Windows AMI to update

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Choose the **name** or **ID** of a launch profile that contains the Windows AMI that you would like to update.
4. On the **Launch profile details** page, choose the Windows AMI that you would like to update.
5. On the **Amazon Machine Image details** page, notice the **AMI ID**.

To search for the AMI ID

1. Go to the [EC2 Dashboard](#).
2. Choose **AMIs** in the left navigation pane in **Images**.
3. Paste the AMI ID that you copied into the filter field.
 - If you see the message, No AMIs found matching your filter criteria, open the dropdown to the left of the search field and select one of these filter options: **Owned by me, Public images, or Private images**.

4. In the EC2 console, select the AMI from the search results, and then choose **Launch instance from AMI**.

To launch an instance in the default Amazon VPC

1. Enter **WinWorkstationAMIBuilder** as the **Name**.
2. Select **g4dn.xlarge** for the **Instance type**.
 - a. If your service quota for On-demand G4 Instances is high enough, you can select an instance with more than four vCPUs. If you're unsure of your quota value, refer to [Amazon Nimble Studio service quotas](#) in the **Setting up** tutorial for instructions about how to check it.
 - b. To use the **EnableFastLaunch** feature to launch your Windows AMIs faster, check that you have enough vCPU limits to launch temporary instances that create snapshots. At minimum, you need 5 times vCPU for the required instance types to use **EnableFastLaunches**.
3. If you already have an existing [key pair](#) that you created and downloaded previously, select it from the **Key pair (login)** dropdown. If you don't have a key pair, follow these instructions to create one:
 - a. To create a new key pair, select **Create a new key pair**.
 - b. Enter a name for your key pair.
 - c. Choose **Download Key Pair** and save the downloaded file in a location on your local computer that you can retrieve later. AWS stores the public key file for you, but you need to store the private key file.
4. In the **Network settings** section, choose **Edit** and check that the following parameters are selected.
 - a. For **Network**, select the **default virtual private cloud (VPC)** in the dropdown. Its name will end in **(default)**.
 - b. Set **Auto-assign Public IP** to **Enable** so that your instance receives a public IP address that you will use when connecting to it later.
 - c. Choose **Select an existing security group**. Then choose the **Nimble_Studio_Build_AMI** security group that you created in the [Step 2: Create IAM policies](#) section of the **Update AMIs: Setting up** tutorial.

5. In the **Configure storage** section, check that the size of your instance's root volume is large enough to accommodate all the new software that you're installing.
 - You might need to increase the size of the root volume to be sure that you have enough space available.
6. Choose **Advanced** to the right of **Configure storage**
 - a. Select the dropdown next to the storage volume.
 - b. Select **Yes**.
 - c. Choose your AMI encryption key from the **KMS key** dropdown dialogue box.
7. Select the dropdown in the **Advanced details** section. For **IAM instance profile**, choose the **Nimble_Studio_Build_AMI** role that you created in [Step 2: Create IAM policies](#) of the **Update AMIs: Setting up** tutorial.
8. Choose **Launch**.

 **Note**

If there is a warning near the top of the window stating that your security group is open to the world, this is because you used a source value of **Anywhere** when creating your security group. We recommend that you don't use a source value of **Anywhere** because that allows any IP to connect to your rule's port(s). For instructions about how to update your security group, see the [Update AMIs: Setting up](#) tutorial.

- If warning message pops up that says you can't connect to your instance because port 3389 isn't open, choose **Continue**. Ignore this message. You will be connecting to the instance using Session Manager, which doesn't require port 3389 to be open.
9. On the **Launch Status** page, navigate to the bottom and choose **View instances**.

Since you're using this instance for setup only, you're launching it in the default VPC that you got with your AWS account. While this isn't the same as your studio's VPC, check with your studio administrator and IT team before making security decisions. You can also choose to limit which IP addresses can access this instance. Learn to change the inbound rules for your security group in the [Step 4: Create a security group](#) section of the **Updating Amazon Machine Images** tutorial.

(Optional) Configure faster launching for Windows AMIs

Follow the instructions in the [Start faster launching for Windows AMIs](#) to enable faster launches.

Step 3: Restart the NICE DCV session as an administrator

Before you can connect to your instance with NICE DCV, close the existing NICE DCV session on your instance and restart it as an administrator. Logging in as **Administrator** later gives you the privileges that you need to install software.

After your instance has initialized, you will connect to it so that you can install the new software.

1. Wait for the **Instance state** of your **WinWorkstationAMIBuilder** instance to change from **Initializing** to **Running**, and for the **Status check** to change from **Initializing** to **2/2 checks passed**. You might need to refresh the page to see the status change.
2. Select the instance and choose **Connect**.
3. Make sure the **Session Manager** tab is selected and then choose **Connect**.
 - If the **Connect** button is grayed out, or if you see a warning message, wait a few minutes for Session Manager to finish initializing on your instance, and then try again.
4. In the **Session Manager** browser tab that opens, run the following commands:

```
cd 'C:\Program Files\NICE\DCV\Server\bin\'  
.\\dcv close-session console  
.\\dcv create-session --owner Administrator console  
}
```

5. After all three commands have been run, close the **Session Manager** tab.
6. In the **Connect to instance** browser tab, choose **Cancel** to close the connection options and return to the list of instances.

Step 4: Connect with NICE DCV

Now that you restarted NICE DCV, you can connect to your instance as an administrator and install new software. To connect with NICE DCV, you will need to find the public IP address of your instance, and then decrypt the password.

To find the public IP address

1. In the list of instances in the EC2 console, select the **WinWorkstationAMIBuilder** instance. Then look in the **Details** tab.
2. Find the entry for **Public IPv4 address** and copy it to the clipboard. As a shortcut, you can choose the **copy icon** located to the left of the address.
3. Open the **NICE DCV client** on your local computer.
4. Paste the **Public IPv4 address** that you copied into the **Hostname/IP Address** field in the NICE DCV client window and choose **Connect**.
5. In the window that says Your connection isn't secure choose **Trust** (or **Proceed** if on MacOS) to trust the connection and continue.
 - By default, NICE DCV generates a self-signed certificate that is used to secure traffic between the NICE DCV client on your local computer and the server on your workstation. If you would like to use your own certificate, choose **Go Back** instead and see [Changing the TLS Certificate - NICE DCV](#) in the **NICE DCV Administrator Guide**.
6. For **Username**, enter **Administrator**.

To decrypt the administrator password and connect

Next, you will need to get the unique administrator password assigned to your instance, and then decrypt it using the key pair that you chose when launching the instance.

1. Back in the **EC2 console**, select the **WinWorkstationAMIBuilder** instance.
2. Choose **Actions**. Then choose **Security** and **Get Windows password**.
3. Choose **Browse** and then open the **key pair file** that you chose when launching your instance.
4. Choose **Decrypt Password**.
5. Copy the **decrypted password** to the clipboard by choosing the **copy icon** to the left of the password.
6. Back in the **NICE DCV client** window, paste the **Administrator password** into the password field and choose **Login**.
 - After a moment, NICE DCV will connect and your Windows virtual workstation desktop will appear.

Step 5: Download and run installers

After you connect to, and sign in to your virtual workstation, you can download and install software. One way is from the public internet, and the other is by copying installers that you stored in an Amazon Simple Storage Service (Amazon S3) bucket.

We recommend that you use an S3 bucket to store the installers for the software that your studio will use. This conveniently eliminates any need to search online for the installer. This will also help verify that consistent versions of software are installed on the different AMIs in your studio.

If you don't already have an S3 bucket with your installers, follow the [Update AMIs: Setting up](#) tutorial to create one.

Open PowerShell and connect to Amazon S3

1. Go to the **Start Menu** and search for **PowerShell**.
2. Select **Windows PowerShell** from the list.
3. Verify that your virtual workstation can access your installers S3 bucket. Replace <BUCKET-NAME> with the name of your installer's S3 bucket.

```
aws s3 ls s3://<BUCKET-NAME>
```

4. The PowerShell Administrator aws s3 command returns this output: PRE blender/PRE davinciResolve/.
5. Change to the Administrator\Downloads folder by running the following command:

```
cd C:\Users\Administrator\Downloads
```

Find the installers in your S3 bucket

If you're comfortable using the command-line tools to locate the file path to the installers in your S3 bucket, you can skip this section and go to [Download installers from your S3 bucket](#). Otherwise, here we'll show you how to find the file paths using the Amazon S3 console.

1. Sign in to the AWS Management Console and open the [Amazon S3](#) console.
2. Find the bucket for installers that you created in the [Update AMIs: Setting up](#) tutorial.
3. Navigate to the first installer that you want to install on your virtual workstation.
4. Choose the Name of the installer file.

5. In the **Object overview** section, copy your **S3 URI** so that you can paste it in the next step.

Download installers from your S3 bucket

On your virtual workstation, you will run commands to download the installer that you located in Amazon S3.

1. Run the following command in PowerShell to download the installer that you just located in Amazon S3: `aws s3 cp S3-URI`.
 - Replace S3-URI with the URI that you just copied from the Amazon S3 console.
2. Check that the command prompt displays a confirmation that the download was successful. If not, check that you ran the `aws s3 cp` command using the correct **S3 URI** and try again.
3. Open **File Explorer** and navigate to **Downloads** to verify that the downloaded file is there. If not, check that you ran the `aws s3 cp` command from the `C:\Users\Administrators\Downloads` folder and try again.
4. Repeat this process for any other installers that you want to download.
5. After you download the installers, run them to install or update software on the virtual workstation.

Note

If any software that you install requires a restart of the virtual workstation, the restart will disconnect the NICE DCV session that you set up to run as administrator. To reconnect to the virtual workstation with NICE DCV, first repeat the steps in [Step 3: Restart the NICE DCV session as an administrator](#).

Software specific installation tips

Some software requires that you complete extra steps for it to work correctly as part of an AMI. For more information, including currently-known software that requires specific install instructions, see the [Software specific installation tips](#) tutorial.

Step 6: Prepare your instance for AMI creation

The next steps will check that your virtual workstation is prepared for AMI creation after installing software. These steps include removing any installers from the C : drive of the virtual workstation, and cleaning up any information that you don't want duplicated when the AMI is created.

Disconnect network drives

Unless you manually mapped network drives to this virtual workstation on your own, there shouldn't be any network drives connected to it. However, leaving network drives connected can cause problems.

1. Open **File Explorer**.
2. Choose **This PC** from the navigation pane.
3. Navigate to the **Devices and drives section** to see if you have any network drives to disconnect.
 - a. If you only see the C : drive listed, you can skip this step and proceed to [Remove installers and unneeded files](#).
 - b. If you have drives other than the C : drive, disconnect those, as follows:
 - i. Select each drive and open the context menu (right-click).
 - ii. Choose **Disconnect** for each of the drives, so that only the C : drive remains.

Remove installers and unneeded files

The files that you created or downloaded on the C : drive of your virtual workstation will be copied during the AMI creation process. These files will appear on any other virtual workstations that are launched using that AMI. For that reason, remove any installers or other files that you don't want copied.

1. In **File Explorer**, check the C:\Users\Administrator\Downloads folder for any installers that you downloaded in previous steps.
2. Check the following folders for files that you can delete.
 - a. C:\Users\Administrator\Documents
 - b. The **Desktop**

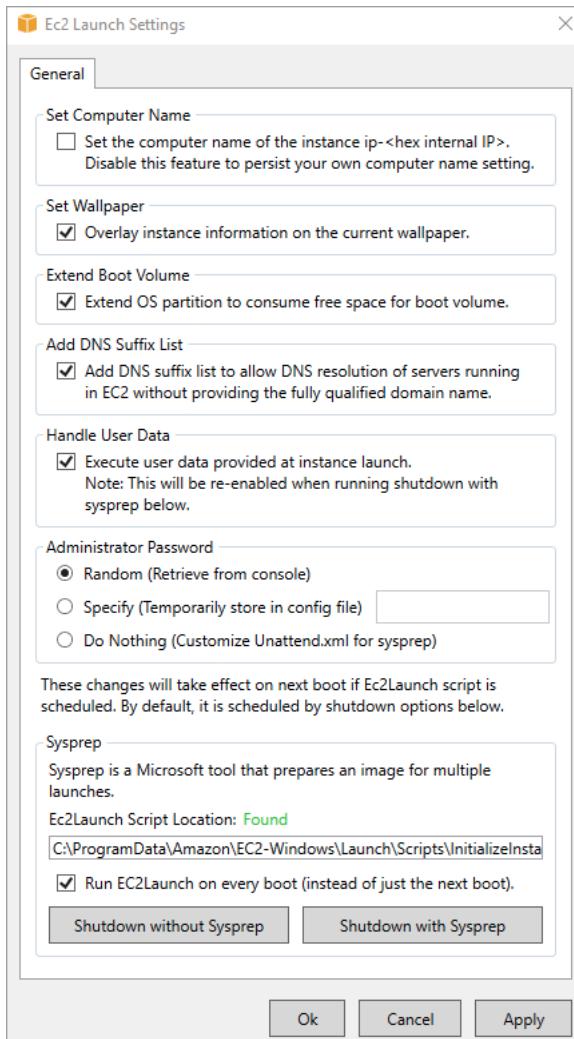
c. The Recycle Bin

Run Sysprep using EC2LaunchSettings

After you remove the extra files from your virtual workstation, you're ready to run a special application to complete the preparation process.

Windows 2019

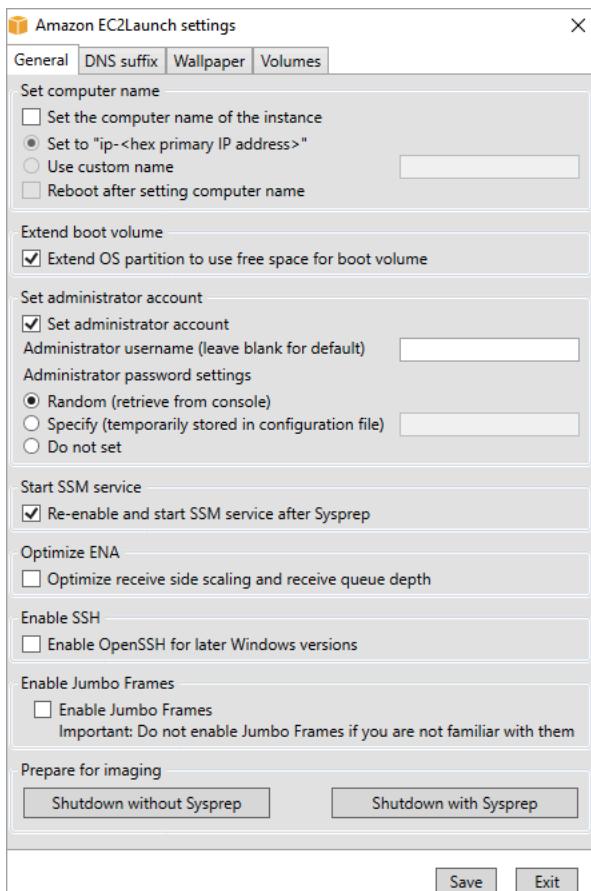
1. Open the **Start Menu**, search for **EC2LaunchSettings**, and then choose it from the list.
2. In the **EC2 Launch Settings** window, check that **Administrator password settings** is set to **Random (retrieve from console)**.
3. Next, go to the bottom of the list and select **Run EC2Launch on every boot**. Then choose **Shutdown with Sysprep**.



4. In the **Sysprep Confirmation** window, choose **Yes**.
5. After a few minutes, your virtual workstation will shut down, and your Remote Desktop session will disconnect.

Windows 2022

1. Open the **Start Menu**, search for **EC2LaunchSettings**, and then choose it from the list.
2. In the **EC2 Launch Settings** window, check that **Administrator password settings** is set to **Random (retrieve from console)**.
3. In the **Prepare for imaging** section, choose **Shutdown with Sysprep**.



4. In the **Sysprep Confirmation** window, choose **Yes**.
5. After a few minutes, your virtual workstation will shut down, and your Remote Desktop session will disconnect.

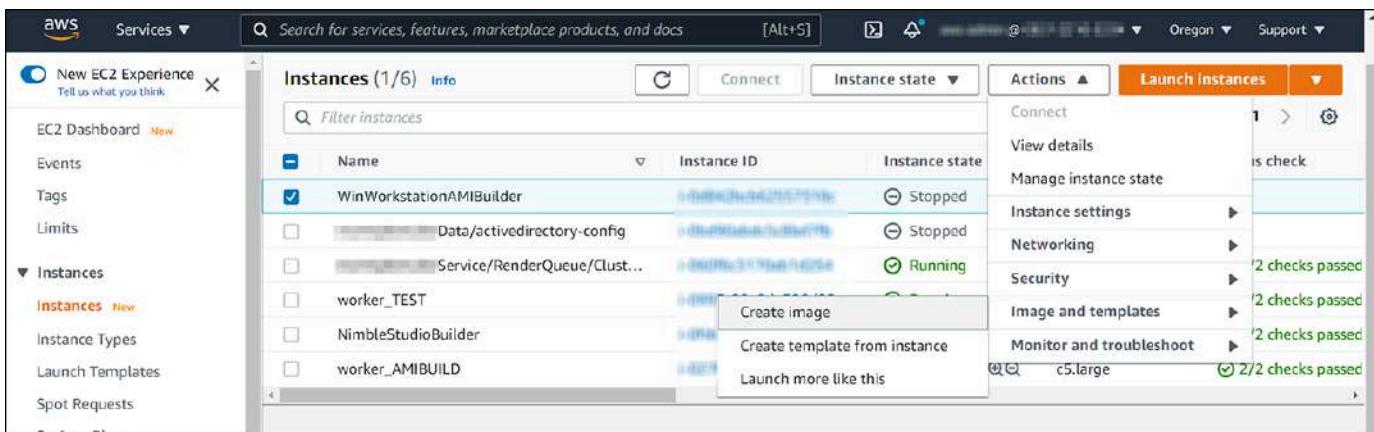
Step 7: Create a new AMI

Note

Before adding AMIs to your studio, check that these don't exceed 500 GB (size) and 10 (quantity). For detailed instructions, see [Reduce the AMI size](#), and [Remove AMIs or increase your quota](#), in the **Updating Amazon Machine Images (AMIs)** tutorial.

Now that your virtual workstation has shut down, you can create an AMI from it.

1. Go to the [EC2 console](#).
2. Choose **Instances** in the left navigation pane.
3. Wait for the **Instance state** of your instance to change to **Stopped**.
4. Choose the instance. Then choose **Actions, Images and templates**, and **Create image**.



5. Enter an **Image name**:
 - To help you keep track of the different AMIs that you create, it's a good idea to give them descriptive names. Descriptive names should include the operating system, intended use (workstation), the department that will use the AMI, and a date or version number.
Example: <your-studio-name>-win-workstation-animation-2021-03-11
6. Enter an **Image description**:
 - To make your image description, you can include what you changed on this AMI, such as what makes it unique, or the new software that you installed. Example: Windows workstation for animation with Blender 2.92.0
7. Navigate to the bottom, then choose **Create image**.

8. Choose **AMIs** in the left navigation pane in **Images**.
9. Your new AMI will be in the list with a **Status of pending**. When the status changes to **available**, you can continue with the next step. This process takes 10-20 minutes, depending on the amount of software installed on your instance.
10. You might also want to add a name to your AMI by hovering over the **Name** field and choosing the **edit icon**.

Step 8: Update launch profiles

Now that your new AMI is encrypted, you will need to add it to Nimble Studio and update your launch profiles so that your artists can use it.

Add AMI to Nimble Studio

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. Choose **Add** on the Studio resources page in **Amazon Machine Image (AMI)**.
4. Enter an **AMI name**. You can use any name that you like, but we suggest copying the name that you originally chose when creating your AMI for the first time. For example: <your-studio-name> Win Workstation - Animation.
5. Enter the **AMI ID** for the encrypted AMI that you just created in the last step.
6. (Optional) Enter an **AMI description**.
7. Choose **Next**.
8. A message will display saying that **Your encrypted AMI can be added to Nimble Studio**.
9. Choose **Add AMI**.

Update launch profiles

Follow the steps in the [Creating launch profiles](#) tutorial to create and share a new launch profile, or update an existing launch profile to use the encrypted AMI that you just created. When you get to the step where you choose AMIs for your launch profile, select your new AMI.

Step 9: Update the DCV server

Important

Before you update the NICE DCV server, first log out of NICE DCV. As an alternative, you can use a different **Remote Desktop Service** during the update.

This section provides PowerShell commands that will help you update your NICE DCV server without accidentally resetting your registry keys.

If you tried updating your NICE DCV server but encountered issues with your AMI or stream afterward, it was due to those registry keys. To remedy this issue, run the following PowerShell commands.

```
$autoConsoleSessionRegKey = "HKEY_USERS\S-1-5-18\Software\GSettings\com\nicesoftware
link \dcv\session-management\automatic-console-session"
$connectivityKey = "HKEY_USERS\S-1-5-18\Software\GSettings\com\nicesoftware\dcv
\connectivity"
$displayRegKey = "HKEY_USERS\S-1-5-18\Software\GSettings\com\nicesoftware\dcv\display"
reg DELETE $autoConsoleSessionRegKey /v owner /f
reg DELETE $autoConsoleSessionRegKey /v storage-root /f
reg DELETE $displayRegKey /v target-fps /f
reg DELETE $displayRegKey /v quality /f
reg DELETE $displayRegKey /v frames-in-transit /f
reg DELETE $displayRegKey /v frame-queue-weights /f
reg DELETE $displayRegKey /v web-client-max-head-resolution /f
reg ADD $autoConsoleSessionRegKey /v owner /d "ec2-nimble" /f
reg ADD $autoConsoleSessionRegKey /v storage-root /d "C:\Users\ec2-nimble\Downloads" /f
reg ADD $displayRegKey /v target-fps /t REG_DWORD /d 0 /f
reg ADD $displayRegKey /v quality /d "(20, 100)" /f
reg ADD $displayRegKey /v frames-in-transit /d "(1, 10)" /f
reg ADD $displayRegKey /v frame-queue-weights /d "(5, 3, 1)" /f
reg ADD $displayRegKey /v web-client-max-head-resolution /d "(4096, 2160)" /f
```

Troubleshooting

Reconnecting to a virtual workstation.

I had to restart my virtual workstation as part of installing new software and now NICE DCV won't connect.

Restarting your virtual workstation will disconnect the NICE DCV session that you set up to run as an administrator. To reconnect to the virtual workstation with NICE DCV, you will first need to repeat the steps in [Step 3: Restart the NICE DCV session as an administrator](#). Then you will be able to reconnect using NICE DCV.

Related resources

- [NICE DCV clients](#)
- [Connect to your Windows instance - Amazon Elastic Compute Cloud](#)
- [Security group rules - Amazon Elastic Compute Cloud](#)
- [Amazon EC2 key pairs and Windows instances - Amazon Elastic Compute Cloud](#)

Update a Linux workstation AMI

This tutorial covers the steps necessary to manually update a Linux workstation Amazon Machine Image (AMI) with new software.

 **Note**

The Nimble Studio Linux workstation AMI is a CentOS 7 instance with some necessary components to work with Nimble Studio. To use CentOS 7, we recommend using the Nimble Studio Linux workstation AMI because starting with CentOS 7 can cause launch errors.

Contents

- [Prerequisites](#)
- [Step 1: Launch an instance with Linux workstation AMI](#)
- [Step 2: Connect with Session Manager](#)
- [Step 3: Update software from Session Manager](#)
- [Step 4: \(Optional\) Set up and connect with NICE DCV](#)
- [Step 5: Prepare the virtual workstation and create an AMI](#)
- [Step 6: Create a customer managed key](#)
- [Step 7: Encrypt the AMI](#)
- [Step 8: Update launch profiles](#)

- [Troubleshooting](#)
- [Related resources](#)

Prerequisites

- Complete the prerequisites and follow the steps in the [Update AMIs: Setting up](#) tutorial.
- Check your service quota for running On-Demand Instances (limit name "G"). For instructions about how to do that, see [Check AWS service quotas](#).
- (Optional) To use a GUI to install software on your Linux virtual workstation, instal the NICE DCV client on your local machine. See the [NICE DCV User Guide](#) to download one, if needed.

Step 1: Launch an instance with Linux workstation AMI

To launch an instance using a Linux AMI that you want to update, you have two choices:

- If you've never created a Linux AMI for your studio, follow the steps in the [Start with the default Linux workstation AMI from StudioBuilder](#) section.
- If you, or a trusted team member, have already made another Linux AMI for your studio that you would like to start from, see [Start with another Linux workstation AMI already created in your account](#) in this tutorial.

 **Note**

If you created a custom AMI, you don't need to subscribe to it in the AWS Marketplace.

Start with the default Linux workstation AMI from StudioBuilder

1. Go to the [AWS Marketplace](#).
2. Search for **Nimble Studio Linux Workstation**.
3. Choose **Nimble Studio Linux Workstation AMI** from the search results.
4. On the AWS Marketplace page for the Nimble Studio Linux Workstation, select **Continue to Subscribe**.
5. Read the terms and conditions. Then choose **Accept Terms**.
6. After the subscribe request has finished processing, select **Continue to Configuration**.

7. In the **Region** dropdown menu, select the AWS Region that your studio is located in and select **Continue to Launch**.
8. In the **Choose Action** dropdown menu, select **Launch through EC2** and select **Launch**.
 - The Amazon EC2 console will open and will guide you through the rest of the launch process.

After the launch process is complete, go directly to this tutorial's section **To launch an instance in the default Amazon VPC** to continue.

Start with another Linux workstation AMI already created in your account

If you already know the AMI ID of the AMI that you want to update, go directly to the **To search for the AMI ID** of this tutorial.

However, if you don't know the AMI ID, you can find it by looking at the launch profile that uses it. The following steps show you how.

To find the AMI ID of the Linux workstation AMI in launch profiles

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Choose the **name** or **ID** of a launch profile that contains the Linux workstation AMI that you want to update.
4. On the **Launch profile details** page, choose the Linux workstation AMI that you want to update.
5. On the **Amazon Machine Image details** page, notice **AMI ID**.

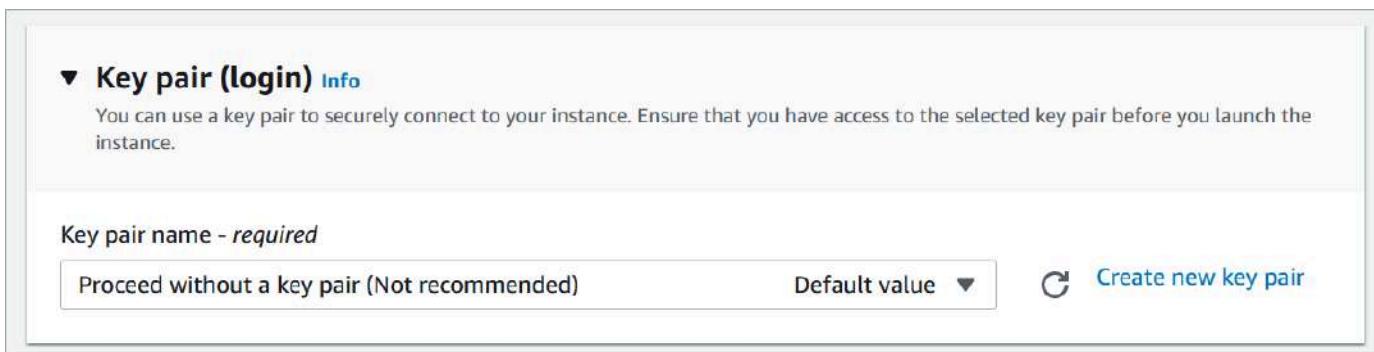
To search for the AMI ID

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **AMIs** in the left navigation pane in **Images**.
3. Paste the AMI ID that you copied into the filter field.
4. If you see the message, No AMIs found matching your filter criteria, open the dropdown to the left of the search field and select one of these filter options: **Owned by me**, **Public images**, or **Private images**.

5. In the Amazon EC2 console, select the AMI from the search results, and then choose **Launch instance from AMI**.

To launch an instance in the default Amazon VPC

1. Enter **LinuxWorkstationAMIBuilder** as the **Name**.
2. Select **g4dn.xlarge** for the **Instance type**.
 - a. If your service quota for On-demand G4 Instances is high enough, you can select an instance with more than four vCPUs. If you're unsure of your quota value, refer to [Amazon Nimble Studio service quotas](#) in the **Setting up** tutorial for instructions about how to check it.
 - b. On Linux, some applications are built from source code to be installed, such as Unreal Engine. This means that smaller instances can take longer to configure. To save time, consider choosing a larger instance type, such as a **g4dn.8xlarge** instance.
3. For **Key pair (login)** choose **Proceed without a key pair** from the first dropdown.



- A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance. You will use Session Manager to connect so you don't need a key pair.
4. In the **Network settings** section, choose **Edit** and select following parameters.
 - a. For **Network**, select the **default virtual private cloud (VPC)** in the dropdown. Its name will end in **(default)**.
 - b. Set **Auto-assign Public IP** to **Enable** so that your instance receives a public IP address that you will use when connecting to it later.
 - c. For **Firewall (security groups)**, choose **Select an existing security group**. Then choose the **Nimble_Studio_Build_AMI** security group that you created in the [Step 2: Create IAM policies](#) section of the **Update AMIs: Setting up** tutorial.

5. In the **Configure storage** section, check that the size of your instance's root volume is large enough to accommodate all the new software that you're installing.
 - You might need to increase the size of the root volume so that you have enough space available.
6. Select the dropdown in the **Advanced details** section. For **IAM instance profile**, choose the **Nimble_Studio_Build_AMI** role that you created in [Step 2: Create IAM policies](#) of the **Update AMIs: Setting up** tutorial.
7. Choose **Launch**.

 **Note**

If there is a warning near the top of the window stating that your security group is open to the world, this is because you used a source value of **Anywhere** when creating your security group. We recommend that you don't use a source value of **Anywhere** because that allows any IP to connect to your rule's port(s). For instructions about how to update your security group, see the [Update AMIs: Setting up](#) tutorial.

- A warning message might pop up that says you can't connect to your instance because port 22 isn't open. Ignore this message. You will be connecting to the instance using Session Manager and NICE DCV, neither of which require port 22 to be open.
8. On the **Launch Status** page, navigate to the bottom and choose **View instances**.

Since you're using this instance for setup only, you're launching it in the default VPC that you got with your AWS account. While this isn't the same as your studio's VPC, check with your studio administrator and IT team before making security decisions. You can also choose to limit which IP addresses can access this instance. Learn to change the inbound rules for your security group in the [Step 4: Create a security group](#) section of the **Updating Amazon Machine Images** tutorial.

Step 2: Connect with Session Manager

After your instance has initialized, you will connect to it with Session Manager so that you can install the new software.

1. Wait for the **Instance state** of your **LinuxWorkstationAMIBuilder** instance to change from **Initializing** to **Running**, and for the **Status check** to change from **Initializing** to **2/2 checks passed**. You might need to refresh the page to see the status change.
2. Select the instance and choose **Connect**.
3. Select **Session Manager** and then choose **Connect**.
 - Session Manager will connect to your instance and open a command prompt in a separate browser tab.

Step 3: Update software from Session Manager

After you connect to, and log in to your virtual workstation, you have many options about how to update the software of your virtual workstation. The simplest method is to use a package manager, such as yum, which is automatically installed on the default Linux AMI. We provide an example of using yum in the following steps.

If appropriate, you can also download updates from the internet using a browser on your virtual workstation, or by downloading files from an Amazon Simple Storage Service (S3) bucket. For detailed instructions about how to download files from S3, see the [Find the installers in your S3 bucket](#) section of the **Update Windows workstation AMI** tutorial. Those instructions make use of Windows PowerShell, but they can be applied to Session Manager or the Linux terminal, as well.

 **Note**

If you require a GUI to install your desired updates, you can go directly to [Step 4: \(Optional\) Set up and connect with NICE DCV](#).

To update the software using yum in Session Manager

1. In the Session Manager window, run the commands that are needed to update or install the desired software.
 - a. For example, you can run this command to apply security updates: `sudo yum -y update --security`

⚠ Important

Make sure to run any updates or installers using the sudo command, otherwise they might not install correctly.

- b. We recommend that you update the NVIDIA GRID drivers and the DCV server after you update your Linux CentOS AMI, or any session built from the Linux CentOS AMI. For information about how to update your NVIDIA drivers, see [NVIDIA](#). For information about how to reinstall the DCV server, see the [Installing the NICE DCV Server on Linux](#) page in the NICE DCV Administrator Guide.
2. Run any additional commands to install all the desired software.
3. If you need to install any software that requires a GUI, continue to the next section.
4. After you're finished with your updates, you can go to [Step 5: Prepare the virtual workstation and create an AMI](#).

If you're planning to stop your workstations when not in use, instead of terminating them, we recommend that you regularly patch, update, and secure the operating system and applications on your EC2 instances.

You can use automatic update services or recommended processes for installing updates that are provided by the application vendor. The following section, **Get Linux automatic updates using yum-cron**, shows you how to install updates using yum-cron provided by RedHat.

 ⓘ Note

yum-cron only works for RHEL and CentOS Linux AMIs.

Get Linux automatic updates using yum-cron

1. Follow the instructions in [Connect to your Linux instance using Session Manager](#) in the AWS Systems Manager User Guide.
2. Run the following command to install yum-cron: `$ sudo yum install -y yum-cron`
3. After you've installed yum-cron, open the configuration file, `/etc/yum/yum-cron.conf` to configure your update schedule.

- a. Change `apply_updates = no` to `apply_updates = yes`.
 - b. Change `random_sleep = 360` to `random_sleep = 0`.
 - Changing the `random_sleep` parameter causes yum-cron to run immediately. The `random_sleep` parameter exists so that you don't use up bandwidth for updates occurring simultaneously.
4. Run the following commands to allow yum-cron to run when you boot the system.

```
$ sudo systemctl start yum-cron  
$ sudo systemctl enable yum-cron
```

5. Check the `yum.log` file after a few minutes to see if any updates have been downloaded and applied to your AMI. The updates might longer to appear, depending on how many updates your AMI requires.

```
$ sudo tail -10 /var/log/yum.log
```



```
Aug 04 10:59:54 Installed: libmodman-2.0.1-8.el7.x86_64  
Aug 04 10:59:54 Installed: libproxy-0.4.11-11.el7.x86_64  
Aug 04 10:59:54 Installed: glib-networking-2.56.1-1.el7.x86_64  
Aug 04 10:59:54 Installed: cockpit-bridge-195.6-1.el7.centos.x86_64  
Aug 04 10:59:55 Installed: cockpit-system-195.6-1.el7.centos.noarch  
Aug 04 10:59:55 Installed: cockpit-ws-195.6-1.el7.centos.x86_64  
Aug 04 10:59:55 Installed: cockpit-195.6-1.el7.centos.x86_64  
Aug 04 16:47:55 Installed: python-chardet-2.2.1-3.el7.noarch  
Aug 04 16:47:55 Installed: python-kitchen-1.1.1-5.el7.noarch  
Aug 04 16:47:55 Installed: yum-utils-1.1.31-54.el7_8.noarch
```

6. After this initial run, reconfigure the `/etc/yum/yum-cron.conf` file so that `random_sleep = 0` is reset to `random_sleep = 360`.

Step 4: (Optional) Set up and connect with NICE DCV

To install some software, you'll first need to connect to your Linux instance with a GUI. To achieve this, modify the DCV setup on your instance to allow you to connect with the DCV client.

Note

If you don't require a GUI to install your updates, you can run those commands from the Session Manager command prompt that you used in the last step. After that, you can skip to [Step 5: Prepare the virtual workstation and create an AMI](#).

To modify the DCV setup

1. Run the following command to open the DCV configuration file in a text editor, and edit the file:

```
sudo vi /etc/dcv/dcv.conf
```

2. Use the **arrow keys** to move the cursor under the line that contains [session-management/automatic-console-session].
3. Press **a** to enter insert mode.
4. Copy the following line to the clipboard:

```
owner="root"
```
5. Paste it into the text editor. Then press the enter (or return) key to add an extra line under the line you just pasted.
 - After completing these changes, the file should look like this:

```
[license]

[log]

[session-management]
create-session = true
agent-launch-strategy="xdg-autostart"

[session-management/defaults]

[session-management/automatic-console-session]
owner="root"

[connectivity]
```

```
[security]

[display]
target-fps = 0
web-client-max-head-resolution = (4096, 2160)
quality = (20, 100)
frames-in-transit = (1, 10)
frame-queue-weights = (5, 3, 1)
```

6. Press **esc** to exit edit mode.
7. Enter **:wq** and press the enter (or return) key to save the file and quit the text editor.
8. Run the following command to restart the DCV server:

```
sudo systemctl restart dcvserver
```

9. Run the following command to set the password for the root user. Remember this password because you will be using it in the next section.

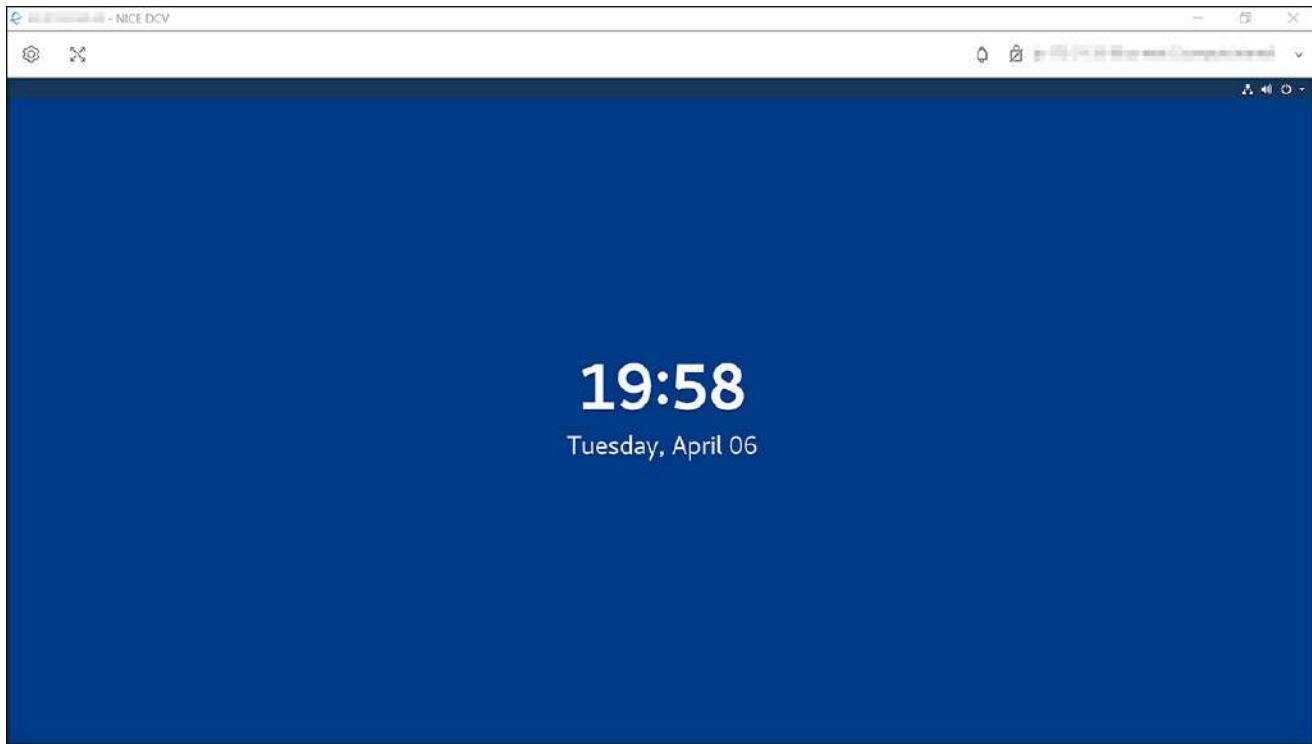
```
sudo passwd root
```

10. Close the Session Manager browser tab.

To connect with DCV

1. In the **Connect to instance** browser tab, choose **Cancel** to exit the connection window.
2. In the list of instances, select your Linux workstation.
3. In the **Description** tab, notice the **Public IPv4 address**.
4. Run the **DCV client** on your local computer.
5. Paste the **Public IPv4 address** that you copied into the **Hostname/IP Address** field in the DCV client window and choose **Connect**.
6. In the window that says Your connection isn't secure, choose **Trust** (or **Proceed** if on MacOS) to trust the connection and continue.
 - By default, DCV generates a self-signed certificate that it used to secure traffic between the DCV client on your local computer and the server on your workstation. If you prefer to use your own certificate, choose **Go Back**, and see these instructions: [Changing the TLS Certificate - NICE DCV](#).

7. For **Username**, enter **root** and then enter the password that you set in the previous steps.
8. Choose **Login**.
 - a. After a moment, DCV will connect to your Linux virtual workstation. The screen might be locked when it connects. If so, it might show a blue screen with the current time and date.

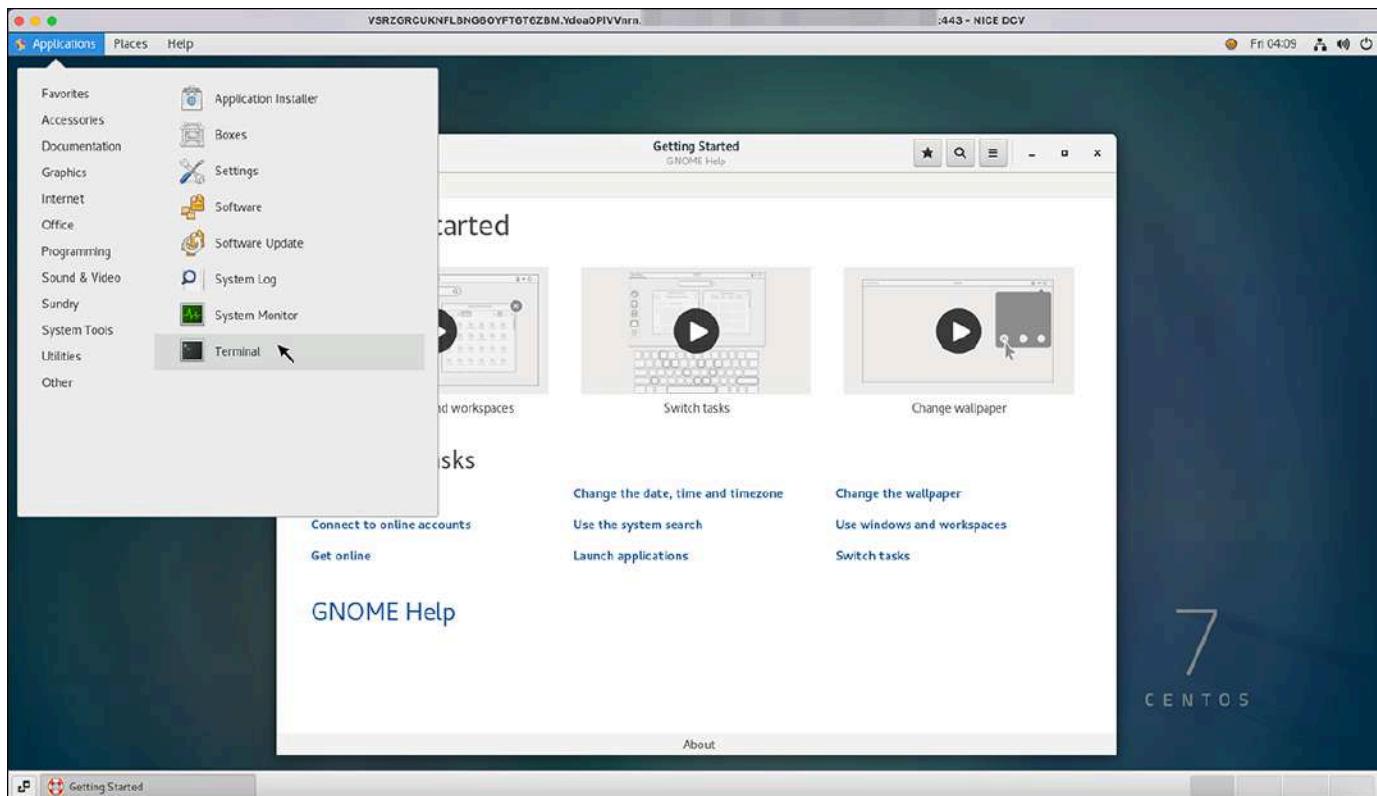


- b. To unlock the screen, first choose the lock screen, then press the enter (or return) key.
9. For **Username**, enter **root** and press the enter (or return) key.
10. Enter the **password** that you set for user **root** earlier, and press the enter (or return) key.

After you successfully logged in, the CentOS desktop opens.

To update the operating system using yum in DCV

1. Select **Applications** in the menu bar. Then choose **System Tools** and **Terminal**.



2. In the terminal window, run the commands that are needed to update or install the desired software.
 - For example, you can run this command to apply security updates:

```
sudo yum -y update --security
```
3. Run any additional commands to install all the desired software, then continue on to the next section.

Step 5: Prepare the virtual workstation and create an AMI

After installing updates, you will create a new AMI from this virtual workstation which will be used to launch other Linux virtual workstations. Before creating the AMI, remove any unneeded files from the virtual workstation first.

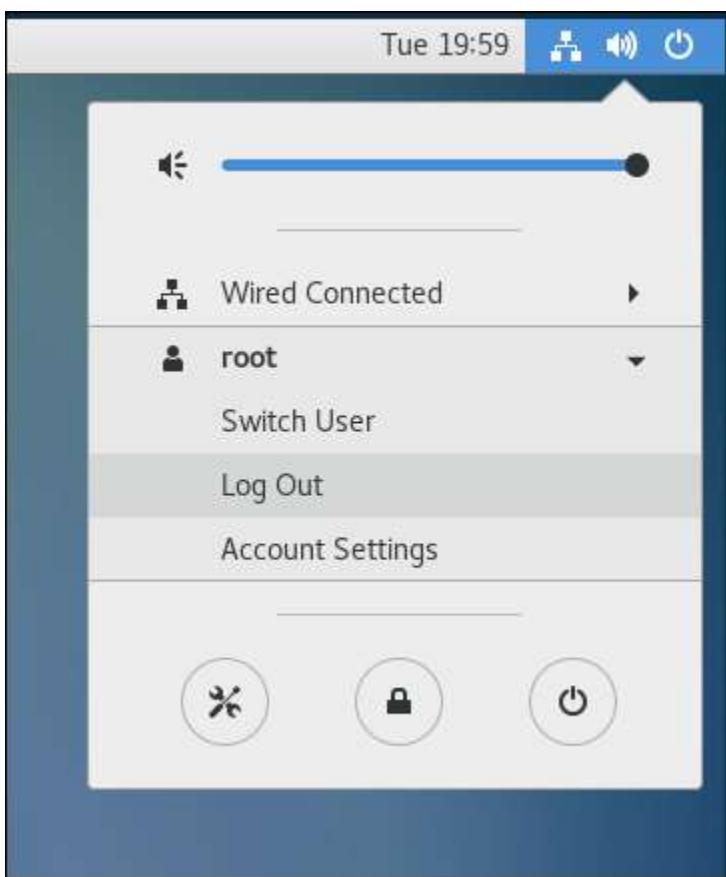
Any files that are present on the virtual workstation that you updated will be copied as part of the AMI creation process. These files will appear on any virtual workstations that are launched using that AMI. Because of that, remove any unneeded files that were created while you updated the operating system of the virtual workstation.

To remove unneeded files

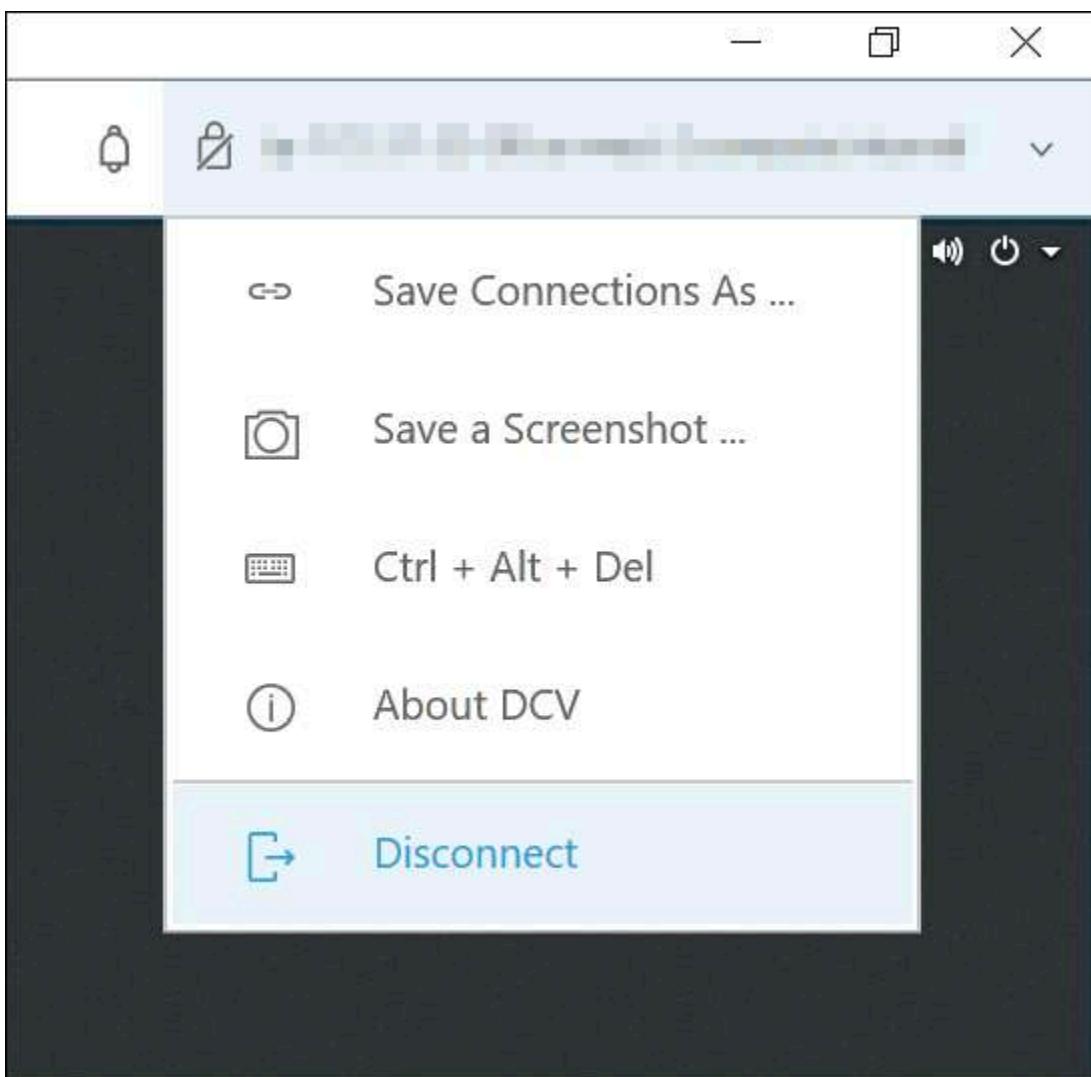
- If you're connected to your workstation with DCV, use a **Terminal** window or the **Files** application to check for files that you can delete. If you're connected with Session Manager, you can use that to check for unneeded files. Some suggested directories for you to look in are:
 - a. **/root/Downloads**
 - b. **/root/Documents**
 - c. **/root/Desktop**
 - d. The **Trash**

To log out and disconnect DCV

1. If you're connected with Session Manager, you can close the Session Manager browser tab and continue to **To create an AMI**.
2. If you're connected with DCV, log out of the virtual workstation by opening the **power menu** of the CentOS desktop (top-right corner, next to volume icon) and selecting **root**, and **Log Out**.



3. Disconnect your DCV session by opening the DCV menu (top right of the DCV window) and choosing **Disconnect**, or by closing the DCV window.



To create an AMI

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select your **LinuxWorkstationAMIBuilder** instance. and then choose **Actions**. Then choose **Images and templates** and **Create image**.
4. Enter an **Image name**:

- To help you keep track of the different AMIs that you create, it's a good idea to give them descriptive names. Descriptive names should include the operating system, intended use (workstation), the department that will use the AMI, and a date or version number.
 - For example: <your-studio-name>-linux-workstation-animation-2021-04-07

5. Enter an **Image description:**

- For the description, you can include what you changed on this AMI, what makes it unique, or the new software that you installed.
 - For example: Linux workstation for animation with OS security updates from 2021-04-07

6. Navigate to the bottom, then choose **Create image**.

7. Choose **AMIs** in the left navigation pane in **Images**.

8. Your new AMI will be in the list with a **Status of pending**. When the status changes to **available**, you can continue with to next step. This process will take 10-20 minutes. Timing for this process is approximate because it depends on the amount of software installed on your instance.

9. To add a name to your AMI, hover your pointer over the **Name** field and choose the **edit icon**.

10. After your AMI has been created, choose **Instances** in the navigation pane to go back to the list of your instances.

11. Select your **LinuxWorkstationAMIBuilder** instance.

- You no longer need this instance to be running, so stopping it will prevent you from incurring the hourly fees to keep it running.

12. Choose **Instance state**. Then choose **Stop instance**.

Step 6: Create a customer managed key

Next, you will create a customer managed key that will be used to encrypt your AMI. You only need to create a customer managed key ont time. If you already created one, you can reuse it and skip to [Step 7: Encrypt the AMI](#).

Follow the instructions in the [Creating symmetric encryption KMS keys \(console\)](#) tutorial in the AWS Key Management Service Developer Guide. Follow these additional steps:

1. For the **Alias** in the **Add labels** section, include the name of your studio and mention custom AMI. For example: <your_studio_name>-customAMI-key
2. For **Key administrators** in the **Define key administrative permissions** section, enter and select the name of the administrator user that you used to sign in to the AWS Management Console. For example: aws-admin

Step 7: Encrypt the AMI

Your AMI needs to be encrypted so that it can be used securely with Amazon Nimble Studio.

Now that you have a customer managed key, you will copy your AMI to your key to complete the encryption process.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **AMIs** in the left navigation pane in **Images**.
3. Make sure that the dropdown to the left of the filter field is set to **Owned by me**, and then select the AMI that you created earlier from the list.
4. Choose **Actions**. Then choose **Copy AMI**.
5. **Destination region:** Select the AWS Region that your studio is deployed in.
6. Next, append **cenc** to the beginning of the name in the **Name** field. That way, you will know that this is a customer encrypted AMI.
7. For **Encryption**, select **Encrypt target EBS snapshots**.
8. For **Master key**, select the name of the customer managed key that you created in the previous step.
9. Choose **Copy AMI**.
10. Choose **Done** to return to your list of AMIs, then refresh the page in your browser. Your copied AMI will be listed with a **Pending** status.
11. Change the name of the new AMI by hovering over the **Name** field and choosing the **edit** icon.
12. Enter a name for the AMI. You can use the name of the AMI that you copied it from, but with **Encrypted** appended to the beginning.
13. When the status changes to **available**, you can continue. This process takes about 10 minutes.
14. Find the **AMI ID** by selecting the encrypted AMI and then looking in the **Details** tab. You will need this information in the next step.

Step 8: Update launch profiles

Now that your new AMI is encrypted, add it to Nimble Studio and update your launch profiles so that your artists can use it.

Add AMI to Nimble Studio

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. Choose **Add** on the Studio resources page in **Amazon Machine Image (AMI)**.
4. Enter an **AMI name**. You can use any name that you like, but we suggest copying the name that you originally chose when creating your AMI for the first time. For example: <your-studio-name> Studio Linux Workstation - Animation
5. Enter the **AMI ID** for the encrypted AMI that you just created in the previous step.
6. (Optional) Enter an **AMI description**.
7. Choose **Next**.
8. A message will display saying that **Your encrypted AMI can be added to Nimble Studio**.
9. Choose **Add AMI**.

Update launch profiles

Follow the steps in the [Creating launch profiles](#) tutorial to create and share a new launch profile, or update an existing launch profile to use the encrypted AMI that you just created. When you get to the step where you choose AMIs for your launch profile, select your new AMI.

Troubleshooting

I get a blank screen when trying to launch an instance after updating an AMI.

You can resolve this issue by installing new NVIDIA grid drivers. For more information about how to do this, follow the [Install NVIDIA drivers on Linux instances tutorial](#) in the Amazon EC2 User Guide for Linux Instances.

Related resources

- [Nice DCV clients](#)

- [Connect to your Linux instance - Amazon Elastic Compute Cloud](#)
- [Security group rules - Amazon Elastic Compute Cloud](#)
- [Amazon EC2 key pairs and Linux instances - Amazon Elastic Compute Cloud](#)

Update a Windows worker AMI

The following programs are installed on the Windows worker Amazon Machine Image (AMI) by default.

- Deadline
- Blender
- Houdini
- Nuke

However, you can add new applications, or update one of the existing applications to a different version. This tutorial guides you through the process of updating or adding new software to your Windows worker AMI. After you update the AMI, you will learn how to connect the new AMI to your render farm, so you can render with the applications that you installed.

 **Note**

This tutorial applies to both WindowsServer2019 AMIs and WindowsServer2022 AMIs.

Contents

- [Prerequisites](#)
- [Step 1: Prepare your test environment](#)
- [Step 2: Launch an instance for AMI creation](#)
- [Step 3: Connect to the TEST worker with Remote Desktop](#)
- [Step 4: Download and run installers](#)
- [Step 5: Validate the update](#)
- [Step 6: Update the AMIBUILD worker instance](#)
- [Step 7: Prepare your instance for AMI creation](#)

- [Step 8: Create a new AMI](#)
- [Step 9: Use StudioBuilder to update your render farm fleet](#)
- [Step 10: Test your deploy](#)
- [Step 11: Terminate the Worker_AMIBUILD and Worker_TEST instances](#)
- [Troubleshooting](#)

Prerequisites

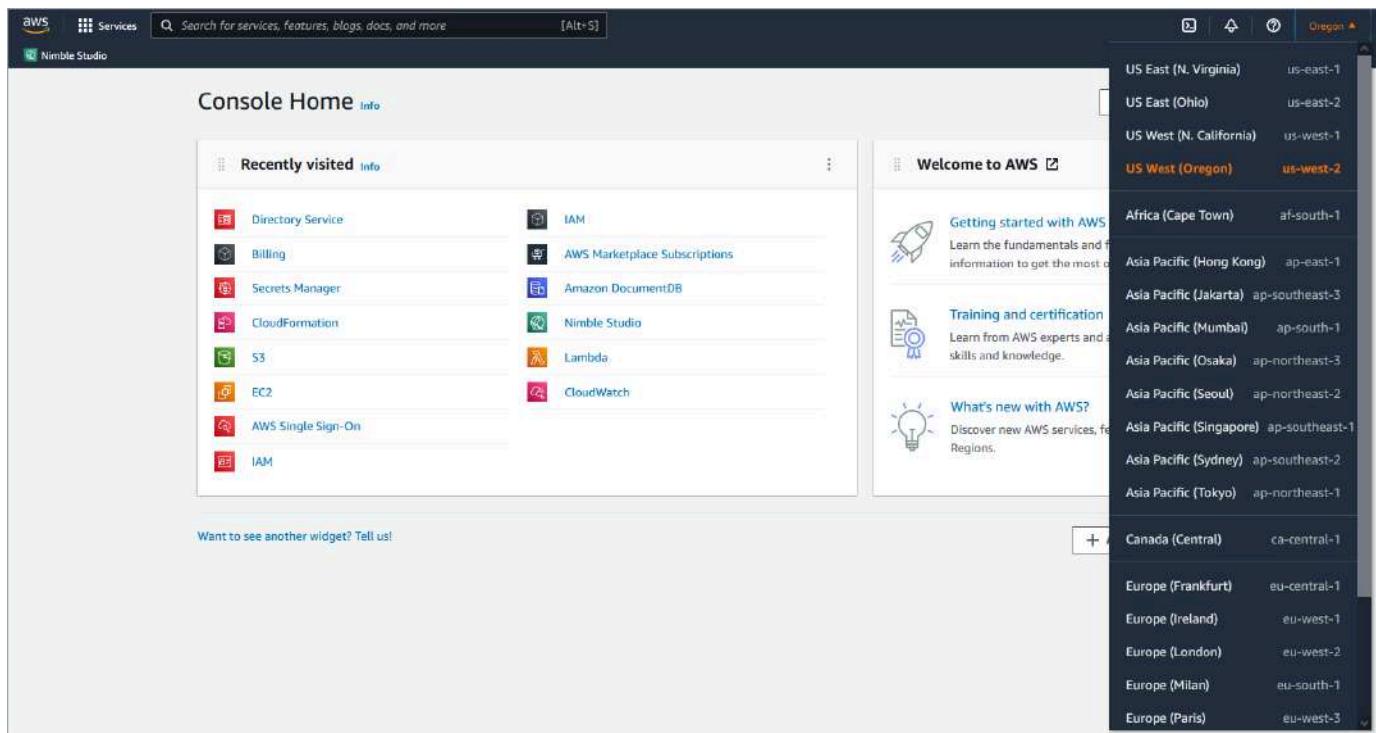
- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- The Windows worker AMI that you update must be part of a Spot Instance render fleet that you created with StudioBuilder, not an On-Demand Instance render fleet.
- The type of render fleet in your studio is determined in StudioBuilder, when you complete [Step 4: Configure studio with StudioBuilder](#).

Step 1: Prepare your test environment

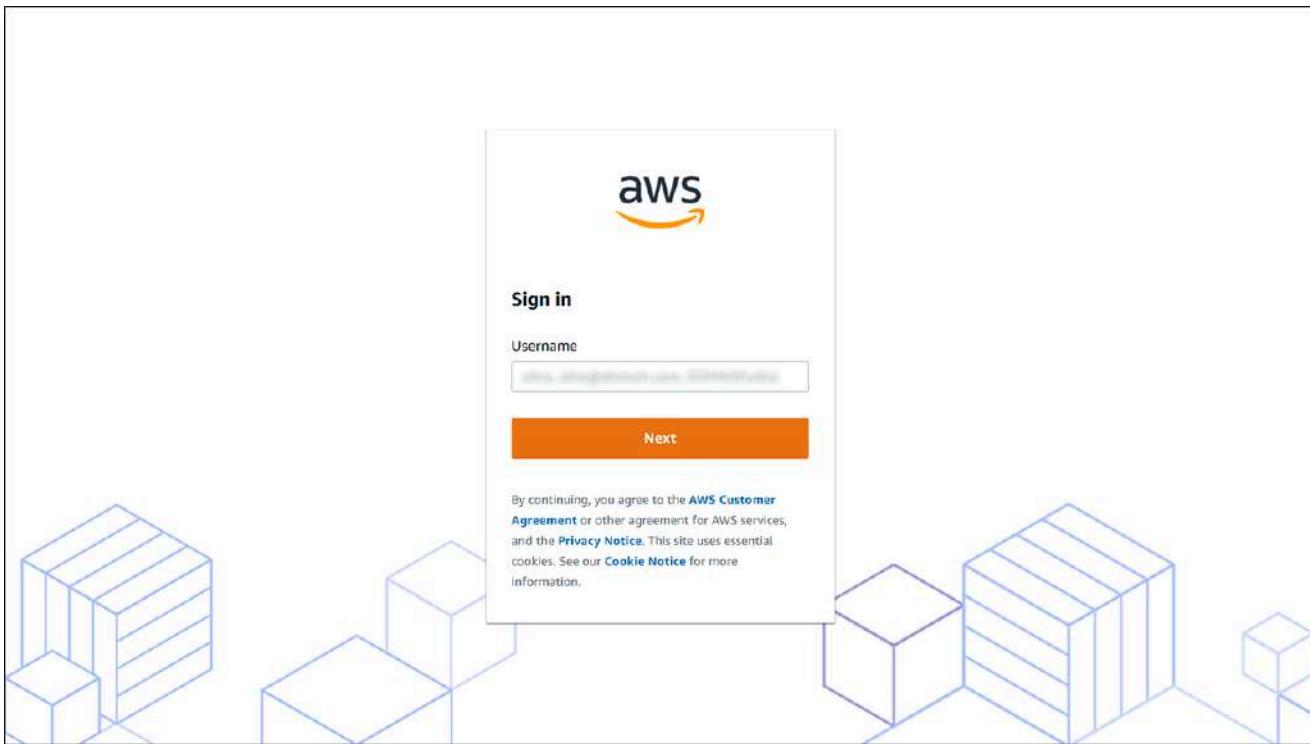
In this step, you will test that the software that you're installing is successfully able to render in your working environment in Nimble Studio. To do this, you will launch two workers: One worker will be connected to [Deadline](#) and will run test renders. The other worker will be outside of your Nimble Studio environment, and will be used to create a clean, updated AMI. This new AMI will be the one that you ultimately use on your render farm.

Launch a Nimble Studio workstation

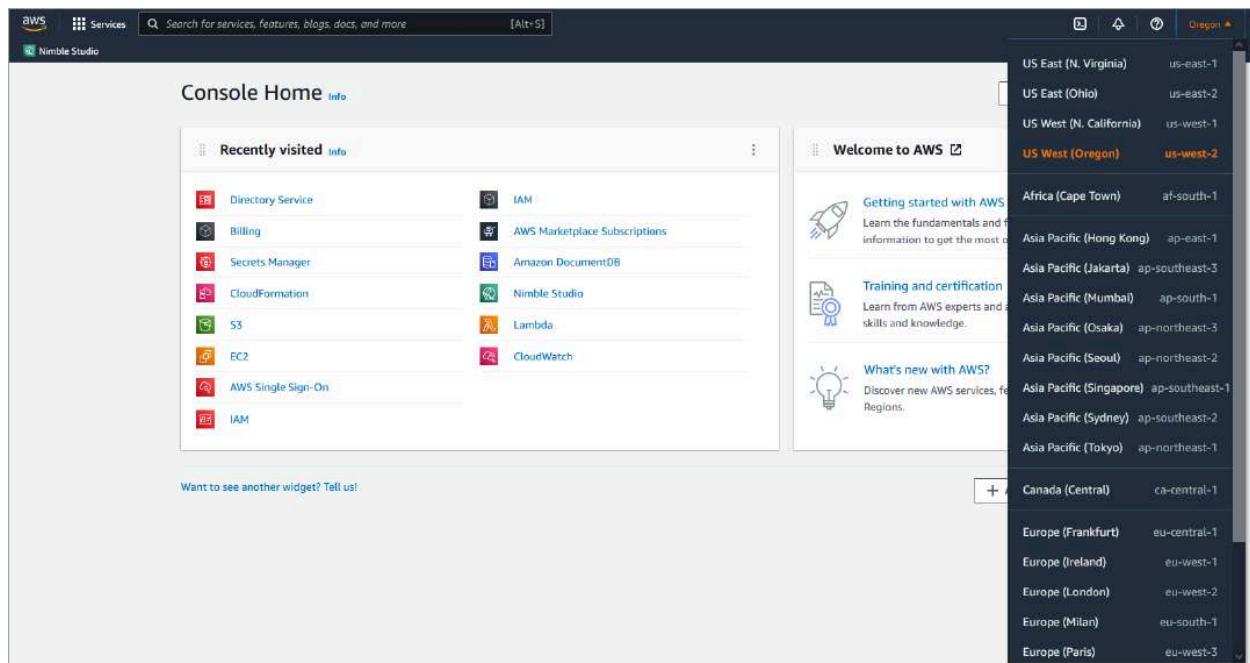
1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



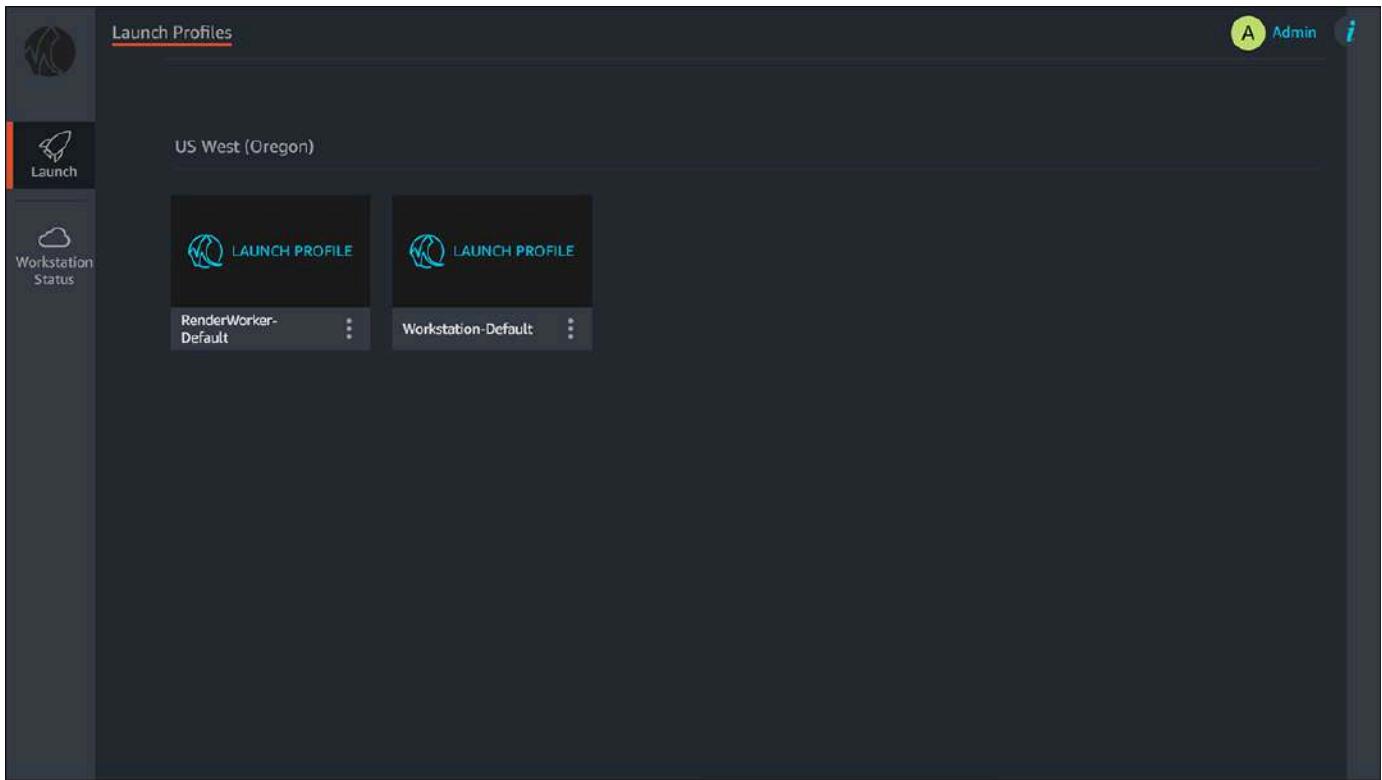
3. Choose **Studio manager** in the left navigation pane.
4. On the **Studio manager** page, choose **Go to Nimble Studio portal**.
5. Sign in to the Nimble Studio portal using your AWS Managed Microsoft AD administrator credentials.
 - a. Enter **Admin** as the user name. Use the password that you set up during StudioBuilder deployment.



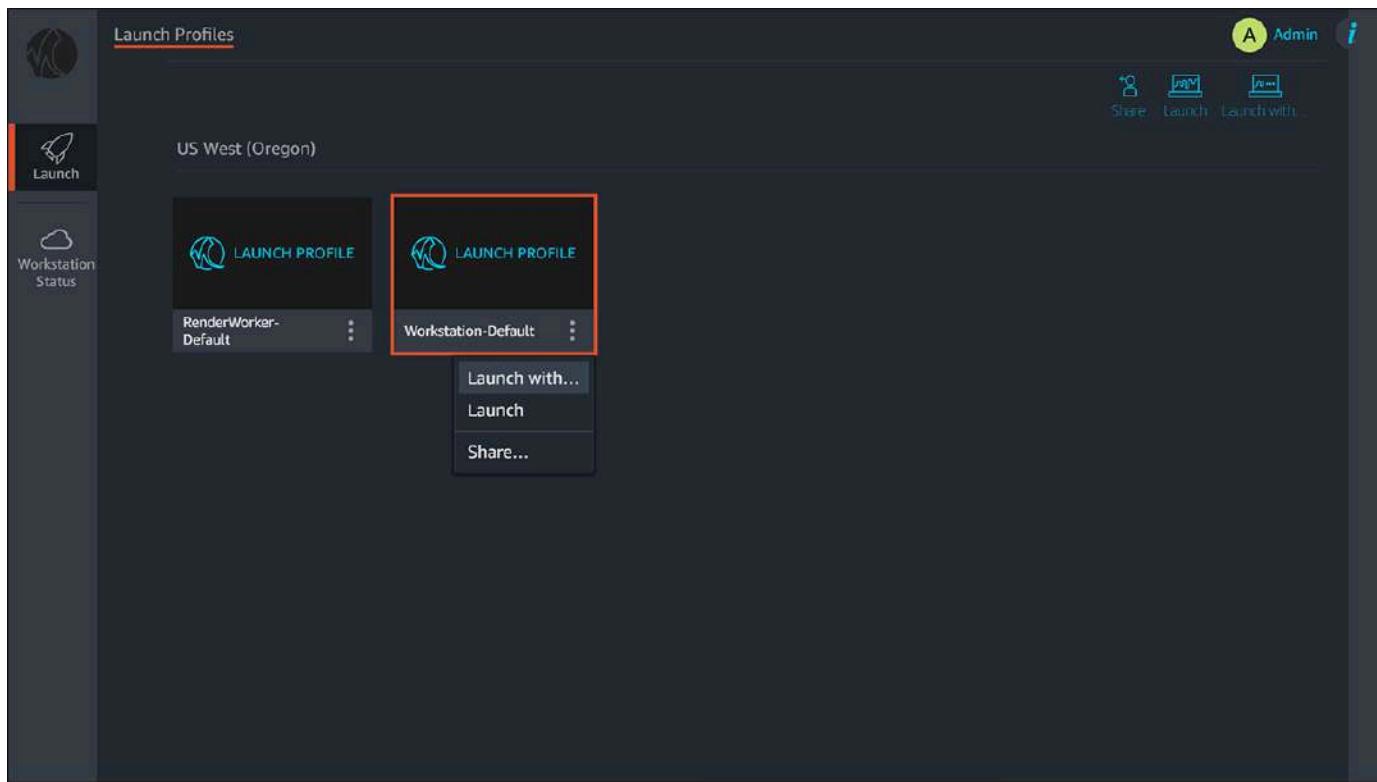
- b. If you forgot your password, do the following:
- Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
 - In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



- iii. Select the **Directory ID** for your studio's Active Directory.
 - iv. Choose **Reset user password**.
6. You will be taken automatically to the **Launch** tab. If not, choose the **Launch** tab from the left navigation pane.



7. Select the vertical ellipsis
(⋮) on the card to open a dropdown menu.



8. Choose **Launch with...**
9. For **Instance Type**, leave it at the default setting.
10. For **Amazon Machine Image**, select **NimbleStudioWindows2019StreamImage** or **NimbleStudioWindows2022StreamImage**.
11. For **Streaming Preference**, choose your streaming preference.
 - a. For the best performance, we recommend choosing **Launch native client**.
 - b. Before connecting to your workstation, download the NICE DCV client. For more information about the NICE DCV client and supported features, see [NICE DCV clients](#).
12. Choose **Launch**.

A status bar will appear that shows you the progress of launching your virtual workstation. This can take up to 10 minutes.

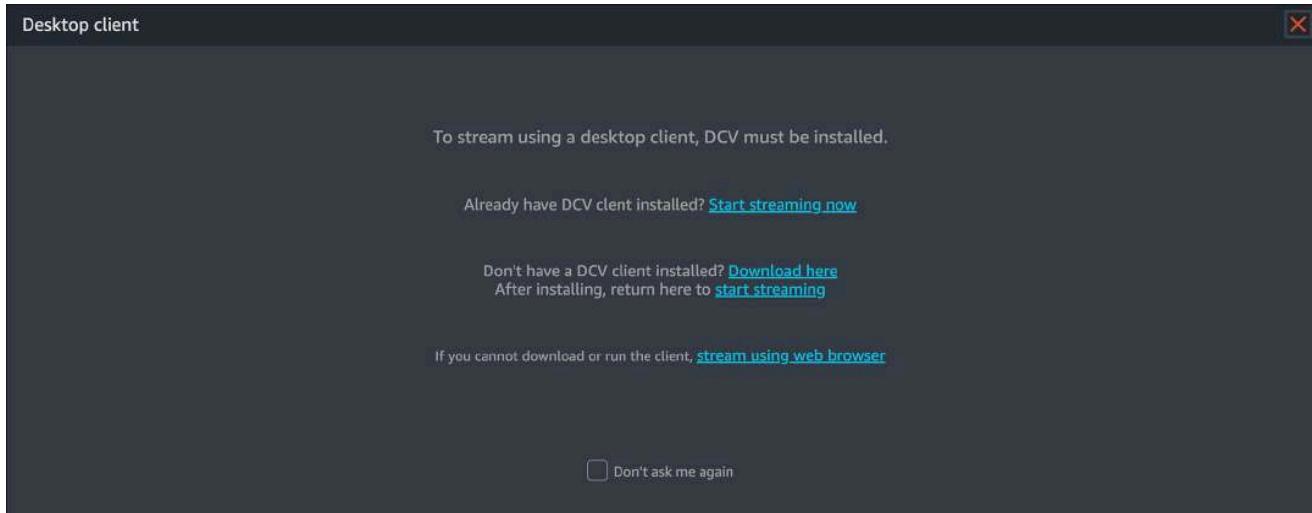
Connect to the virtual workstation

To connect to the virtual workstation

1. When your virtual workstation is ready, a new window appears reminding you that the client must be installed.

2. Choose **Start streaming now**.

- If you haven't installed the NICE DCV desktop client, choose **Download here** and install the client first.



3. When your browser displays a window that prompts you to open NICE DCV, choose **Open** to continue. The exact wording of this might vary depending on what browser you're using.

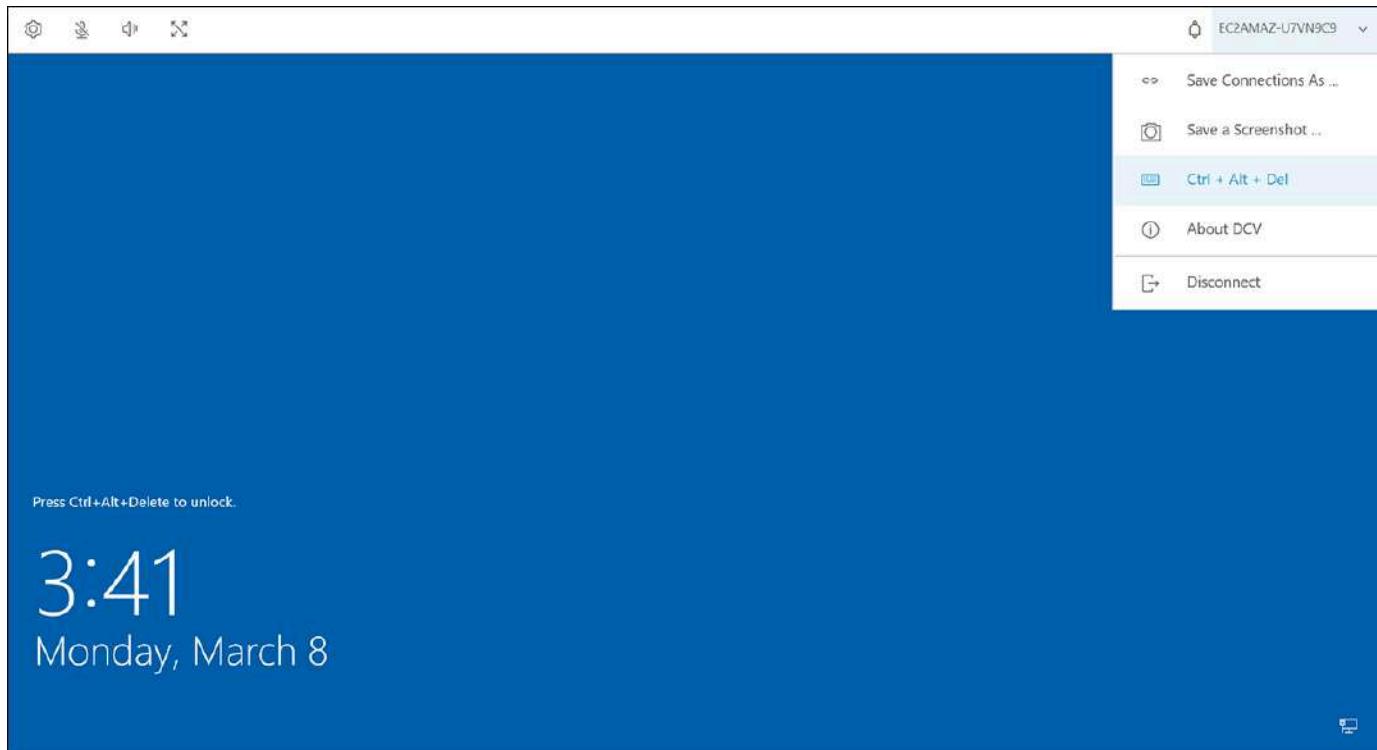
Note

The NICE DCV web browser client runs inside a web browser. You don't need to install the web client. We recommend using the Google Chrome browser to avoid latency. For more information, go to the [Web browser client](#) page in the *NICE DCV User Guide*.

4. After the NICE DCV client application opens in a new window, the Windows login screen will display.
5. Open the instance menu near the top right of the screen and choose **Ctrl + Alt + Del**. For an OS X DCV client, open the **Connection** dropdown menu and select **Send Ctrl + Alt + Del**.

Important

Don't enter **Ctrl+Alt+Delete** on your keyboard. Doing so sends the command to your local computer, not to your workstation.

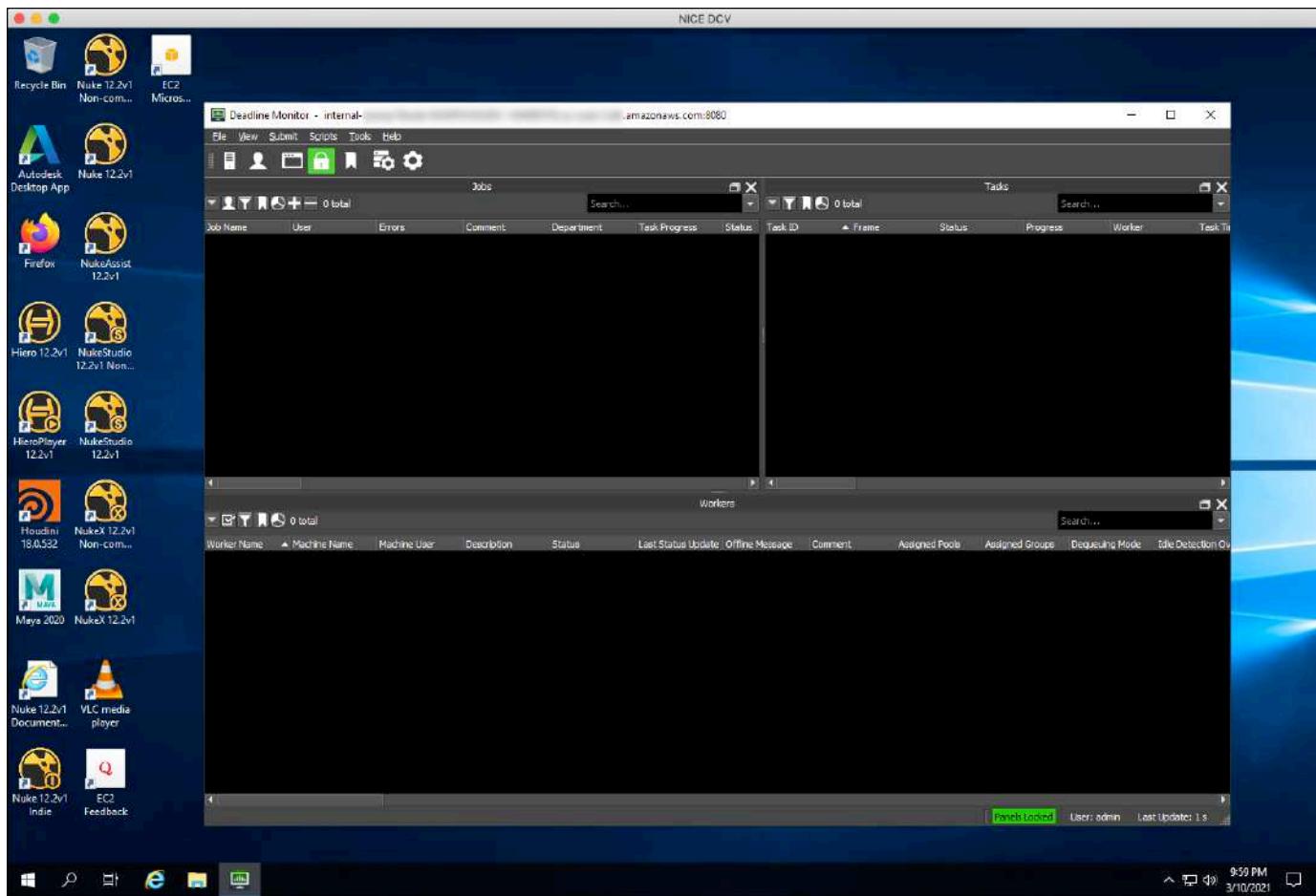


6. For **User name**, enter **Admin**. For **Password**, enter the password that you created during your studio deploy. Then press the enter (or return) key.

You're now connected to your virtual workstation.

To open Deadline

1. After the desktop has loaded, choose the start menu in the lower left corner.
2. Search for **Deadline Monitor** and select it from the list of items.
3. If a window pops up and asks if it's ok to create a new user, choose **OK**.



Update TEST worker IAM role and assign to fleet

When you deployed your studio, StudioBuilder created an IAM role. Update this role to connect to the TEST worker using Session Manager, and to access S3 bucket data from the instance.

To update the TEST worker role and assign to your fleet

1. Follow the instructions in [To use a managed policy as a permissions policy for an identity \(console\)](#) in the IAM User Guide.
2. Search for **S3** and select the box next to the **NimbleStudioInstallersS3ReadOnly** policy.
 - If you can't find this policy, return to [Step 2: Create IAM policies](#) and follow the steps to create it.
3. For identity, search for **DeadlineSpot** and select the name of the role called **DeadlineSpot-<region>-<fleet_name>**.

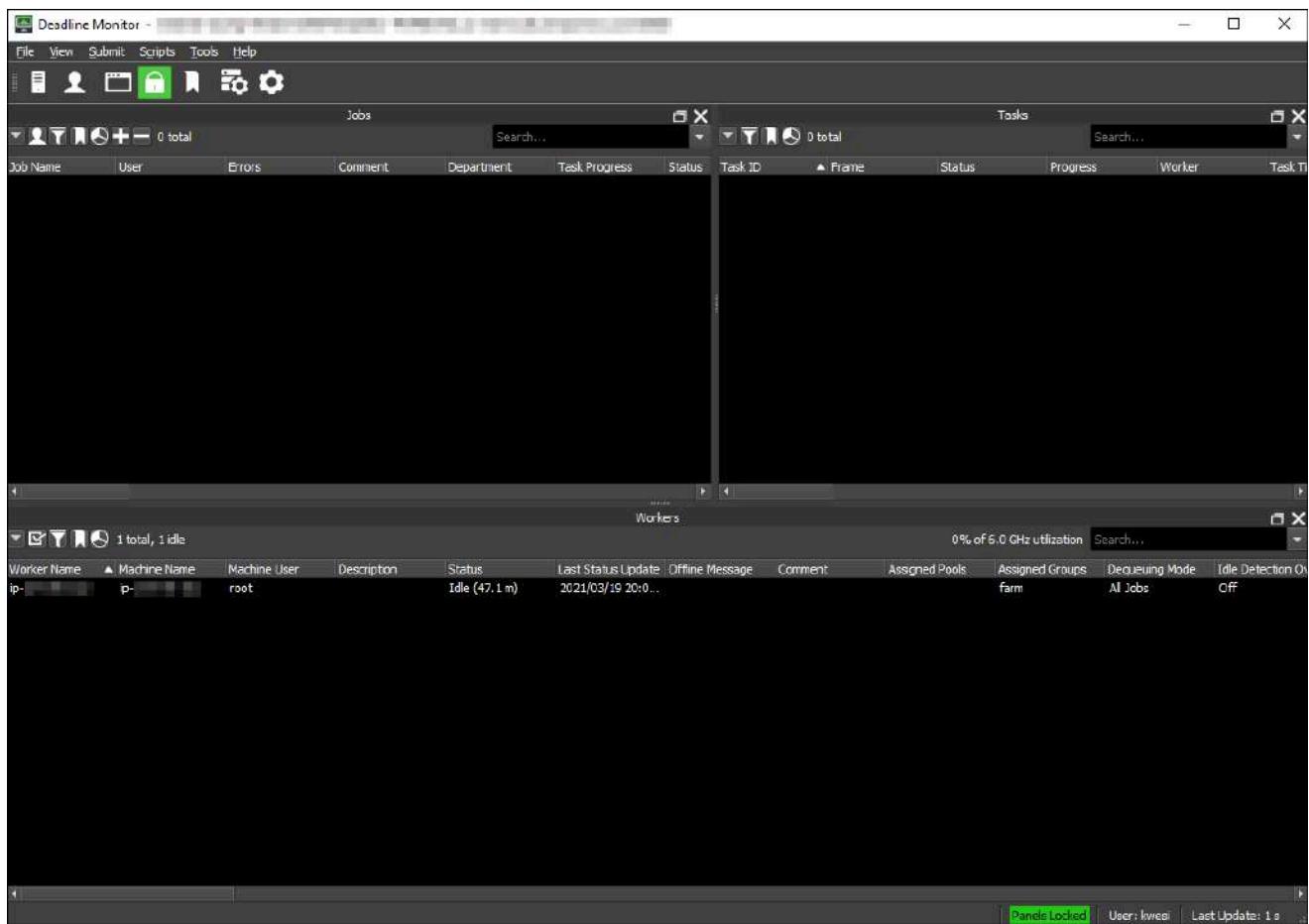
- a. There will be a role for each fleet in your studio.
- b. Select the role for the fleet that you want to test.

Launch a TEST worker

The worker launch template launches a Spot Instance by default. Because of this, it is possible that the instance will be terminated if it's interrupted. If you prefer to launch an On-Demand Instance, follow these steps:

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Select **Launch Templates** in the left navigation pane.
3. Select the worker **Launch template** for the Windows render worker AMI that you want to update.
4. Notice the **AMI ID** in the **Instance details** tab of the **Launch template version details** section. You will need this AMI ID in [Step 2: Launch an instance for AMI creation](#).
5. Choose **Actions**. Then choose **Launch instance from template**.
6. Navigate to **Key pair (login)**.
7. Choose **Create a new key pair**.
 - You will need this later to login to the test instance and the instance you launch to create the new AMI in *step 13* of [Step 2: Launch an instance for AMI creation](#).
8. Navigate to **Resource tags**.
9. Choose **Add tag**.
10. For the **Key**, enter **Name**.
11. For the **Value**, enter **worker_TEST**.
12. Choose **Actions**. Then choose **Launch instance from template**.
13. Return to the EC2 console.
 - Your newly launched worker instance will be **Running**.
14. Choose **Instances** in the left navigation pane.
15. On your virtual workstation, check that the Deadline Monitor shows your worker listed in the bottom left of the pane. It might take a few minutes for the worker to finish initializing and appear in the list.

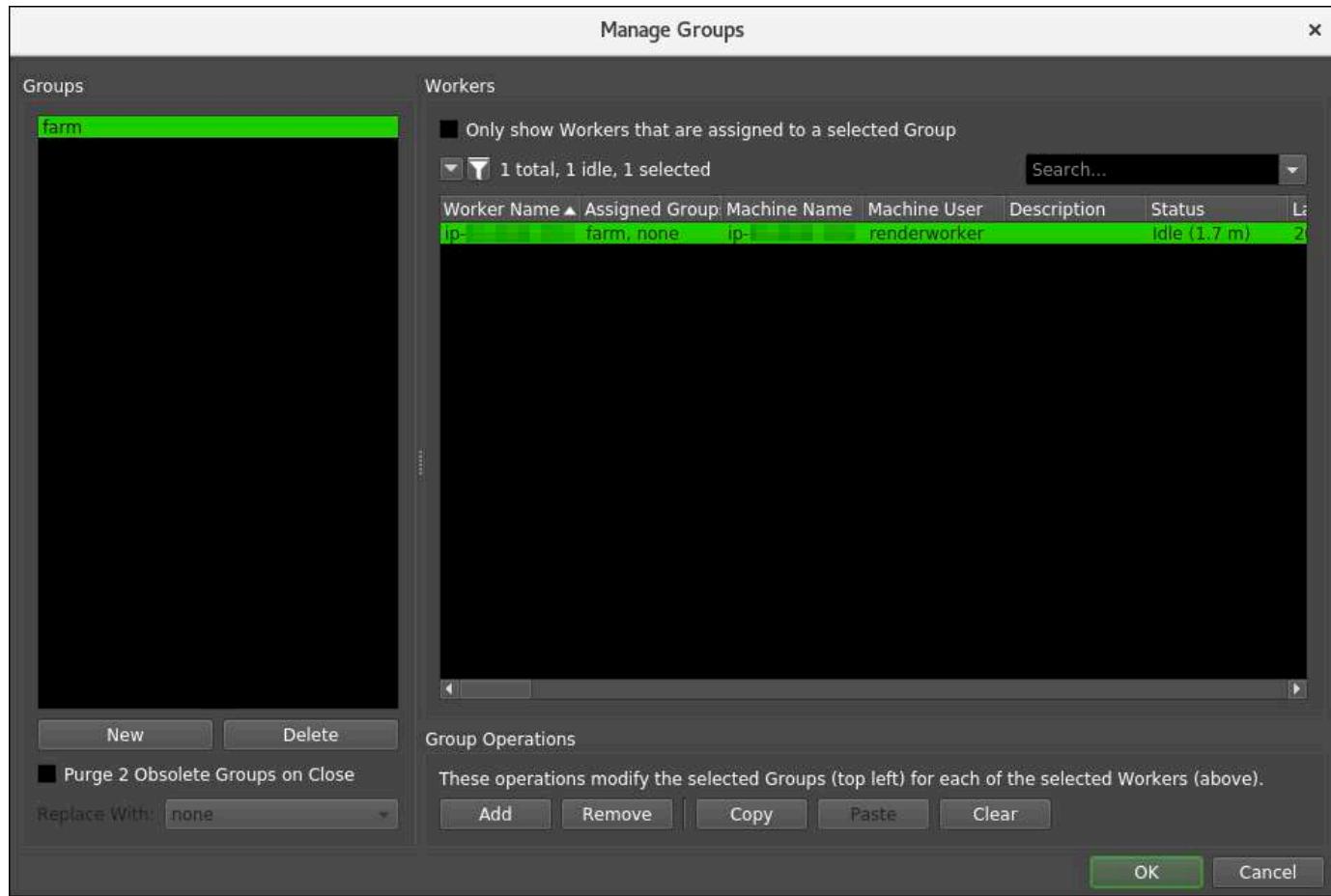
- If after several minutes, you don't see a worker show up, try launching your TEST worker again. If you're still having trouble, consult the [AWSThinkboxDeadline Documentation](#).



Remove the group from the worker

When the worker spins up, it will automatically be a part of the fleet group that you created during your Nimble Studio cloud studio deployment. This worker will terminate automatically, unless you remove the group assignment.

1. Choose **Tools**. Then choose **Super User Mode**.
2. Select the worker in the list of workers in Deadline Monitor.
3. Open the context menu (right-click) for the worker, and then choose **Manage groups**.
4. Select the **group** in the left pane, then select the **worker** in the right pane.



5. Choose **Remove** to remove the group from your worker.
 - The **Assigned Group** for the worker will change to **none**.

Create a new group for the worker

1. Choose **New** to create a new group.
2. Enter **UPDATING-AMI-admin-only** for the group name and choose **OK**.
3. Select the **new group** in the left pane and then select **worker** in the right pane.
4. Choose **Add** to assign the worker to the new group.
5. Choose **OK** to exit the Manage Groups window.
 - a. The **Assigned Group** will get reassigned after a few seconds.
 - b. The worker should now stay active until you terminate it manually.

Prevent the worker from rendering jobs from other users

Even though you've assigned the worker to a new group, there is still a chance that it could pick up render jobs if another user from your studio submits them to the **none** group.

To prevent the worker from rendering jobs from other users, follow these steps:

1. Check that your TEST worker is selected in the list of workers.
2. Open the context menu for the worker (right-click), and then choose **Manage Worker Properties...**
3. Choose **Job Dequeuing** in the left pane.
4. Choose **Only Jobs Submitted From These Users**.
5. In the **User List**, choose **admin**, then choose the **right arrow** to move it to **Job Users**.
6. Choose **OK**.

Now your TESTs worker will only render jobs that are submitted by the administrator user.

Step 2: Launch an instance for AMI creation

To launch the instance for AMI creation outside of your Nimble Studio environment, follow these steps:

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **AMIs** in the left navigation pane in **Images**.
3. In the search field, enter the **AMI ID** from the **Launch a TEST worker** section of [Step 1: Prepare your test environment](#).
 - If you see a message that says No AMIs found matching your filter criteria, open the dropdown menu to the left of the search field and choose one of these two options:
 - i. If this is your first time updating this AMI, select **Public images**
 - ii. If you've updated this AMI before, select **Owned by me**.
4. Make sure that the **NimbleStudioWindows2019WorkerImage** or the **NimbleStudioWindows2022WorkerImage** AMI is selected in the search results and choose **Launch**.
5. For **Instance Type**, select **c5.large**.

6. Choose **Next: Configure Instance Details** in the lower-right corner and complete the instance details as follows:
 - a. Leave **VPC** as `vpc-<id>` (default).
 - b. Leave **Subnet** as No preference (default subnet in any Availability Zone).
 - c. Set **Auto-assign Public IP** to **Enable** so that your instance receives a public IP address that you will use when connecting to it later.
 - d. From the IAM role dropdown, choose the **Nimble_Studio_Build_AMI** profile that you created in [Prerequisites](#).
 - e. Choose **Next: Add Storage** in the lower-right corner.
7. Leave the storage settings as default.
8. Choose **Next: Add Tags** in the lower-right corner and complete the fields as follows:
 - a. Choose **Add Tag**.
 - b. For the **Key**, enter **Name**.
 - c. For the **Value**, enter **worker_AMIBUILD**.
 - d. Choose **Next: Configure Security Group** in the lower-right corner.
9. Leave **Create a new security group** selected.
 - a. You can remove the security group rule by choosing the **x** on the far-right side of the menu bar.
 - b. You will be using port forwarding to connect to this instance and won't need any security group rules to connect.
10. Choose **Review and Launch** in the lower-right corner.
11. Choose **Launch**.
12. In the key pair pop-up, select Choose an existing key pair.
 - Select the key pair that you created in step 7 of the [Launch a TEST worker section of Step 1: Prepare your test environment](#).
13. Read the terms and conditions and if you agree:
 - Select the check box next to **I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to sign in to my instance.**
14. Choose **Launch instances**.
15. Return to **EC2**.

16. Choose **Instances** in the left navigation pane.

You will now see both of your worker instances running.

Step 3: Connect to the TEST worker with Remote Desktop

Now that you've launched a TEST worker and an AMIBUILD worker, it's time to connect to these instances so that you can install the software that you want to use to render your projects. To connect to these instances, you will need to download and install the Session Manager plugin on your local computer so that you can setup port forwarding to securely connect to the worker instances.

Install the Session Manager plugin on your local computer

1. For instructions about how to install Session Manager, see [Install the Session Manager plugin for the AWS CLI](#).
 - Install the correct version for your local operating system (OS).
2. After the plugin is installed, run the following command in Terminal or in PowerShell: `aws configure`
 - a. You will need to have the AWS CLI installed to run this command.
 - b. To install or upgrade the AWS CLI on your local machine, follow the instructions in [Installing the AWS Command Line Interface version 2](#) in the *AWS Command Line Interface User Guide*.
 - c. Configure the AWS CLI by following the instructions in [Setting up new configuration and credentials](#).
 - d. Verify the installation or upgrade by running `aws nimble help`. This command displays a list of available Nimble Studio commands.
3. Run the following command that matches your local machine's operating system to update the instance-id with your `worker_TEST` instance ID. You will connect to the `worker_AMIBUILD` instance in a later step.
 - Replace <instance-id> with your `worker_AMIBUILD` instance ID.

Linux & macOS

```
aws ssm start-session \
--target <instance-id> \
--document-name AWS-StartPortForwardingSession \
--parameters '{"portNumber":["3389"], "localPortNumber":["55678"]}'
```

Windows

```
aws ssm start-session ^
--target <instance-id> ^
--document-name AWS-StartPortForwardingSession ^
--parameters portNumber="3389",localPortNumber="55678"
```

Connect with RDP

Follow the instructions in the [Connect to your Windows instance using RDP](#) tutorial in the Amazon EC2 User Guide for Windows Instances .

You're now connected to your worker instance. Perform the same installation steps on the AMIBUILD worker instance that you performed on your TEST worker instance to [Step 3: Update the TEST worker.](#)

Step 4: Download and run installers

After you connected to, and logged in to the instance, you can download and install software. One way is from the public internet, and the other is by copying installers that you stored in an Amazon Simple Storage Service (Amazon S3) bucket.

We recommend that you use an S3 bucket to store the installers for the software that your studio will use. This conveniently eliminates any need to search online for the installer. This will also help verify that consistent versions of software are installed on the different AMIs in your studio.

If you don't already have an S3 bucket with your installers, follow [Step 1: Create an Amazon Simple Storage Service \(Amazon S3\) bucket to store your installers](#) to create one.

Open PowerShell and connect to Amazon S3

1. Go to the **Start Menu** and search for **PowerShell**.

2. Select **Windows PowerShell** from the list.
3. Run the following command to verify that the **TEST** worker can access your installers S3 bucket: `aws s3 ls s3://<BUCKET-NAME>`
 - Replace `<BUCKET-NAME>` with the name of your installer's S3 bucket.
4. The PowerShell Administrator aws s3 command returns this output: PRE blender/PRE davinciResolve/.
5. Navigate to the Administrator\Downloads folder by running the following command: `cd C:\Users\Administrator\Downloads`

Find the installers in your S3 bucket

If you're comfortable using the command-line tools to locate the file path to the installers in your S3 bucket, you can skip this section and go to [Download installers from your S3 bucket](#). Otherwise, here we'll show you how to find the file paths using the Amazon S3 console.

1. Sign in to the AWS Management Console and open the [Amazon S3](#) console.
2. Find the bucket for installers that you created in the [Update AMIs: Setting up](#) tutorial.
3. Navigate to the first installer that you want to install on the instance.
4. Choose the installer file.
5. In the **Object overview** section, notice **S3 URI**. You will use this in step 2a of the next section.

Download installers from your S3 bucket

On the TEST worker, you will run commands to download the installer that you located in Amazon S3.

1. Run the following command in PowerShell to download the installer that you just located in Amazon S3: `aws s3 cp S3-URI`.
 - Replace S3-URI with the URI that you just copied from the Amazon S3 console.
2. Verify that the command prompt displays a confirmation that the download was successful. If not, check that you ran the aws s3 cp command using the correct **S3 URI** and try again.
3. Open **File Explorer** and navigate to **Downloads** to verify that the downloaded file is there. If not, check that you ran the `aws s3 cp` command from the `C:\Users\Administrators\Downloads` folder and try again.

4. Repeat this process for any other installers that you want to download.
5. After you download the installers, run them to install or update software on the TEST worker.

 **Note**

If any software that you install requires a restart of the virtual workstation, the restart will disconnect the NICE DCV session that you set up to run as administrator. To reconnect to the TEST worker with NICE DCV, first repeat the steps in .

Step 5: Validate the update

Verify that the updates that you made to your render farm are working.

1. Switch back to the NICE DCV window that is connected to the workstation that you launched earlier.
2. Launch a render using the application that you just installed on the TEST worker instance.
3. On your workstation, submit a render.
 - When you launch your render, set your **Group** to **UPDATING-AMI-admin-only**. This render will get picked up by your TEST instance.
4. If your render successfully completes, repeat the installation steps from the TEST instance on this AMIBUILD worker instance.

After you validate that the installations and changes to your TEST worker instance are working, repeat the steps on the AMIBUILD worker instance before creating a new worker AMI in [Step 8: Create a new AMI](#).

Step 6: Update the AMIBUILD worker instance

- Run the following command that matches your local machine's operating system.
 - Replace <instance-id> with your worker_AMIBUILD instance ID.

Linux & macOS

```
aws ssm start-session \
```

```
--target <instance-id> \
--document-name AWS-StartPortForwardingSession \
--parameters '{"portNumber":["3389"], "localPortNumber":["55678"]}'
```

Windows

```
aws ssm start-session ^
--target <instance-id> ^
--document-name AWS-StartPortForwardingSession ^
--parameters portNumber="3389",localPortNumber="55678"
```

Connect with RDP

Follow the instructions in the [Connect to your Windows instance using RDP](#) tutorial in the Amazon EC2 User Guide for Windows Instances.

You're now connected to your worker instance. Perform the same installation steps on the AMIBUILD worker instance that you performed on your TEST worker instance to Update the TEST worker.

Step 7: Prepare your instance for AMI creation

The next steps will check that your virtual workstation is prepared for AMI creation after installing software. These steps include removing any installers from the C: drive of the virtual workstation, and cleaning up any information that you don't want duplicated when the AMI is created.

Disconnect network drives

Unless you manually mapped network drives to this instance on your own, there shouldn't be any network drives connected to it. However, leaving network drives connected can cause problems.

1. Open **File Explorer**.
2. Choose **This PC** from the navigation pane.
3. Navigate to the **Devices and drives section** to see if you have any network drives to disconnect.
 - a. If you only see the C: drive listed, you can skip this step and proceed to [Remove installers and unneeded files](#).

- b. If you have drives other than the C: drive, disconnect those.
 - i. Select each drive and open the context menu (right-click).
 - ii. Choose **Disconnect** for each of the drives, so that only the C: drive remains.

Remove installers and unneeded files

The files that you created or downloaded on the C: drive of your instance will be copied during the AMI creation process. These files will appear on any other workers that are launched using that AMI. For that reason, remove any installers or other files that you don't want copied.

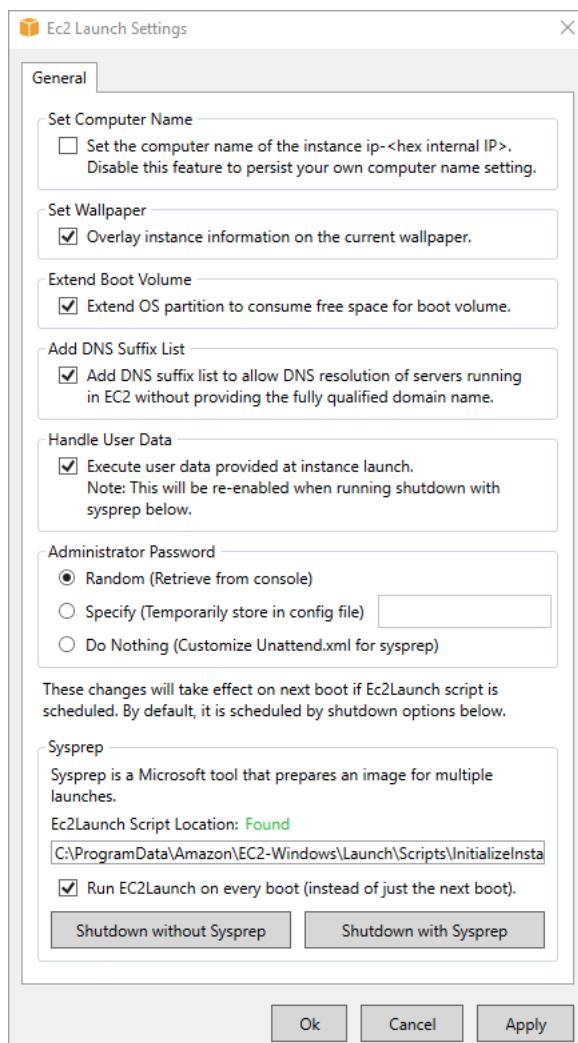
1. In **File Explorer**, check the C:\Users\Administrator\Downloads folder for any installers that you downloaded in previous steps.
2. Check the following folders for files that you can delete.
 - a. C:\Users\Administrator\Documents
 - b. The **Desktop**
 - c. The **Recycle Bin**

Run Sysprep using EC2LaunchSettings

After you remove the extra files from your instance, you're ready to run a special application to complete the preparation process.

Windows 2019

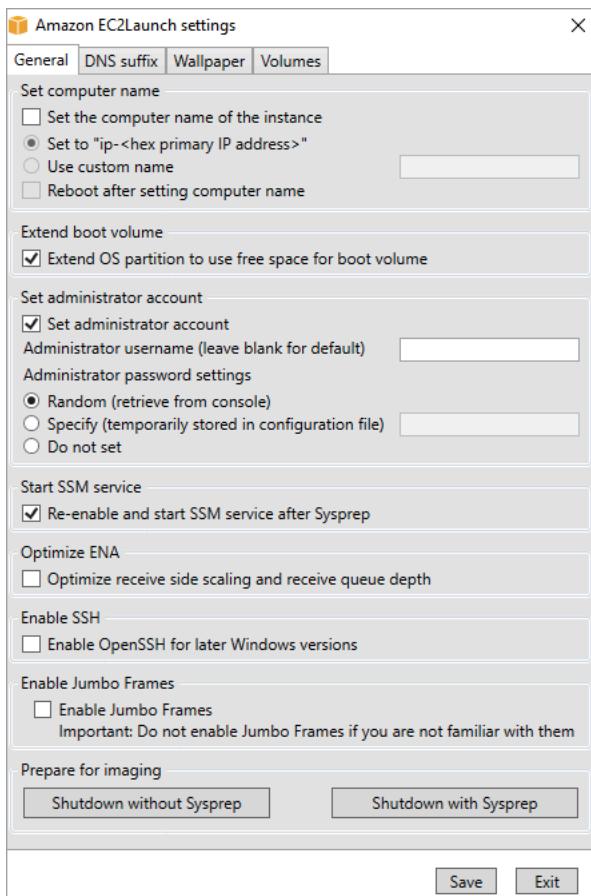
1. Open the **Start Menu**, search for **EC2LaunchSettings**, and then choose it from the list.
2. In the **EC2 Launch Settings** window, check that **Administrator password settings** is set to **Random (retrieve from console)**.
3. Next, go to the bottom of the list and select **Run EC2Launch on every boot**. Then choose **Shutdown with Sysprep**.



4. In the **Sysprep Confirmation** window, choose **Yes**.
5. After a few minutes, your virtual workstation will shut down, and your Remote Desktop session will disconnect.

Windows 2022

1. Open the **Start Menu**, search for **EC2LaunchSettings**, and then choose it from the list.
2. In the **EC2 Launch Settings** window, check that **Administrator password settings** is set to **Random (retrieve from console)**.
3. In the **Prepare for imaging** section, choose **Shutdown with Sysprep**.



4. In the **Sysprep Confirmation** window, choose **Yes**.
5. After a few minutes, your virtual workstation will shut down, and your Remote Desktop session will disconnect.

Step 8: Create a new AMI

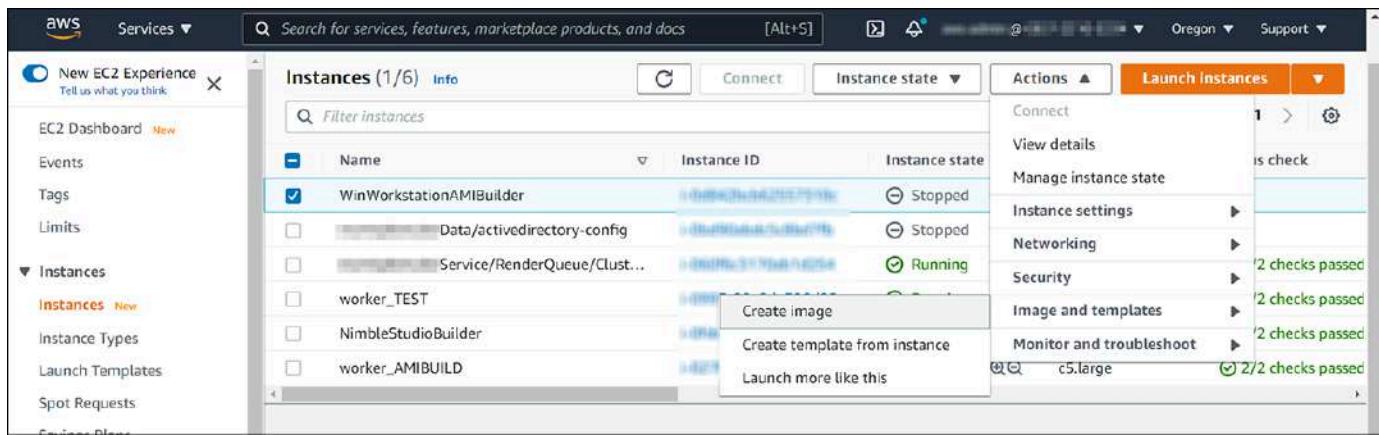
Note

Before adding AMIs to your studio, check that these don't exceed 500 GB (size) and 10 (quantity). For detailed instructions, see [Reduce the AMI size](#), and [Remove AMIs or increase your quota](#), in the [Updating Amazon Machine Images \(AMIs\)](#) tutorial.

Now that your AMIBUILD worker instance has shut down, you can create an AMI from it.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.

3. Wait for the **Instance state** of your worker_AMIBUILD instance to change to **Stopped**.
4. Choose the instance. Then choose **Actions, Images and templates**, and **Create image**.



5. Enter an **Image name**:
 - To help you keep track of the different AMIs that you create, it's a good idea to give them descriptive names. Descriptive names should include the operating system, intended use (worker), the department that will use the AMI, and a date or version number. Example: <your-studio-name>-win-worker-2021-03-11
6. Enter an **Image description**:
 - To make your image description, you can include what you changed on this AMI, such as what makes it unique, or the new software that you installed. Example: Windows worker with Blender 2.92.0
7. Navigate to the bottom, then choose **Create image**.
8. Choose **AMIs** in the left navigation pane in **Images**.
9. Your new AMI will be in the list with a **Status of pending**. When the status changes to **available**, you can continue with the next step. This process takes 10-20 minutes, depending on the amount of software installed on your instance.
10. You might also want to add a name to your AMI by hovering over the **Name** field and choosing the **edit icon**.

Step 9: Use StudioBuilder to update your render farm fleet

To update your render farm fleet to use the new AMI that you created, you will use StudioBuilder to modify the fleet properties.

Connect to the StudioBuilder instance

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select your StudioBuilder instance.
 - If you don't have an instance of StudioBuilder running, follow the tutorial [Deploying a new studio with StudioBuilder](#) tutorial to get an instance of StudioBuilder running in your account.
4. Connect to your instance using one of the techniques specified in the [Deploying a new studio with StudioBuilder](#) tutorial, either through SSH or EC2 Instance Connect.
5. Make sure to connect as ec2-user.

Start StudioBuilder to update your resources

1. When you connect to your instance, a **Welcome back** prompt will display. This gives you options to work with your studio.
 - If you don't see the prompt, enter **studio_builder** and press the enter (or return) key to launch the StudioBuilder tool.
2. Use the arrow keys to choose **Update an existing resource**.
3. StudioBuilder will display your current configuration.
4. Press **Y** to edit the configuration and press the enter (or return) key to continue.
5. StudioBuilder will remember most of your previous selections, so you can press the enter (or return) key to accept earlier choices. Continue until you get to the first question about the farm.
6. **Would you like to modify or delete fleet: <farmname>? Please select an option.**
 - Choose **Modify**
7. **Which type of fleet would you like?**
 - To accept your previous choice press the enter (or return) key.
8. **Which Operating System will this fleet use?**
 - To accept your previous choice press the enter (or return) key.
9. **Enter the AMI ID for this Render Worker fleet.**
 - a. Delete the **AMI-ID** that is listed.

- b. Enter the **AMI-ID** of your new worker AMI.
10. Continue through the rest of questions and press the enter (or return) key to accept the defaults unless you would like to change them.

Review

- **Would you like to generate a studio configuration with your selections?**
 - If you're happy with all the selections that you made in the preceding steps, enter **Y** and press the enter (or return) key to proceed. Enter **N** and press the enter (or return) key to go back and make changes.

Ready to deploy your studio build

- **Please type BUILD MY STUDIO (and then press enter) to continue, or type QUIT (and then press enter) to exit**
 - Enter **BUILD MY STUDIO** and press the enter (or return) key to continue.
 - i. StudioBuilder will run the deploy to update all the components of your studio.
 - ii. It will take approximately 10 minutes to complete updating your studio.

Once your update is complete

After StudioBuilder has finished running, you will be asked what you want to do next. After that, you can close the StudioBuilder browser tab and [terminate your StudioBuilder instance](#). We recommend terminating unused instances to prevent incurring costs.

Step 10: Test your deploy

Now, when you submit renders to your farm, the new AMI that you created will be used to spin up the render workers. To test whether the correct AMIs are being used, we will launch a test render. During this process, we will check the worker to verify that it's using the correct AMI.

1. Launch a new render following the workflow from the [Creating your first render on the farm](#) tutorial.
 - Wait a few minutes for workers to spin up.

2. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
3. Choose **Instances** in the left navigation pane.
4. A few workers will be spinning up in the list of instances. Select one of the new workers.
5. Choose the **Details** tab.
6. Open **Instance Details** to see more information about the selected instance.
7. Find the **AMI ID** for your instance. It should match the ID of the AMI that you used in StudioBuilder.

Step 11: Terminate the Worker_AMIBUILD and Worker_TEST instances

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select the worker_AMIBUILD and the worker_TEST instances.
4. Choose **Instance state**. Then choose **Terminate**.

Troubleshooting

The render worker that I created doesn't appear in the Deadline Monitor.

Consult the [AWSThinkboxDeadline Documentation](#). The launch logs might provide additional information about why you are receiving errors. You can find the launch logs in the C:\\ProgramData\\Amazon\\EC2-Windows\\Launch\\Log file.

You can also perform a test launch [Step 2: \(Optional\) Perform a test launch](#) to verify that there aren't any problems with the launch profile.

Update a Linux worker AMI

The following programs are installed on the Linux worker Amazon Machine Image (AMI) by default.

- Deadline
- Blender
- Houdini
- Nuke

However, you can add new applications, or update one of the existing applications to a different version. This tutorial guides you through the process of updating or adding new software to your Linux worker AMI. After you update the AMI, you will learn how to connect the new AMI to your render farm, so you can render with the applications that you installed.

Contents

- [Prerequisites](#)
- [Step 1: Prepare your test environment](#)
- [Step 2: Launch an instance for AMI creation](#)
- [Step 3: Update the TEST worker](#)
- [Step 4: Validate the update](#)
- [Step 5: Update the AMIBUILD instance](#)
- [Step 6: Create the new AMI](#)
- [Step 7: Use StudioBuilder to update your render farm fleet](#)
- [Step 8: Test your deploy](#)
- [Step 9: Terminate the Worker_AMIBUILD and Worker_TEST instances](#)
- [Troubleshooting](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- You also need to go through the steps in [Update AMIs: Setting up](#). That tutorial explains how to set up an Amazon Simple Storage Service (Amazon S3) bucket with installers, create an AWS Identity and Access Management (IAM) role that enables Amazon S3 access, and set up a security group.

 **Note**

The Linux worker AMI that you're updating must be part of a Spot Instance render fleet that you created with StudioBuilder, rather than an On-Demand Instance render fleet.

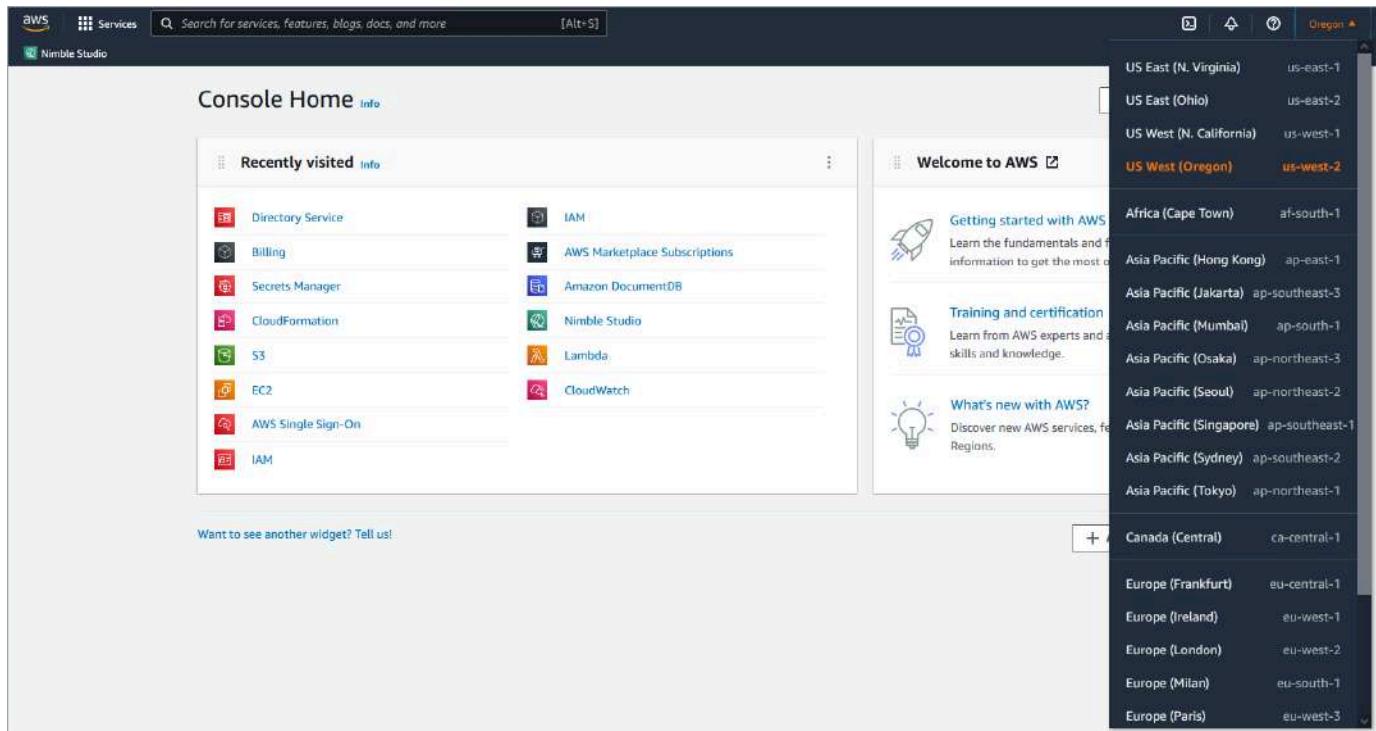
- The type of the render fleet in your studio is determined in StudioBuilder when you complete [Step 4: Configure studio with StudioBuilder](#).

Step 1: Prepare your test environment

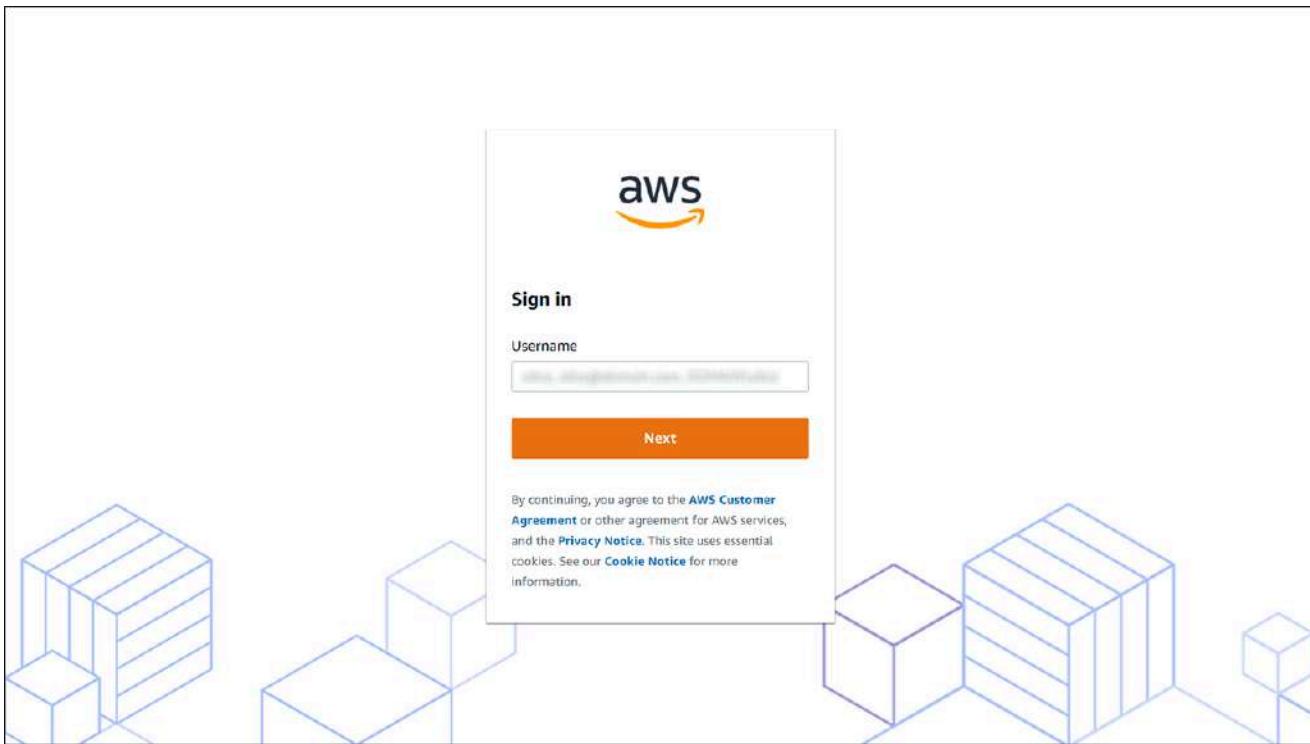
You can test that the software that you're installing is successfully able to render in your working environment in Nimble Studio. To do this, you will launch two workers: One worker will be connected to [Deadline](#) and will run test renders. The other worker will be outside of your Nimble Studio environment, and will be used to create a clean, updated AMI. This new AMI will be the one that you ultimately use on your render farm.

Launch a Nimble Studio workstation

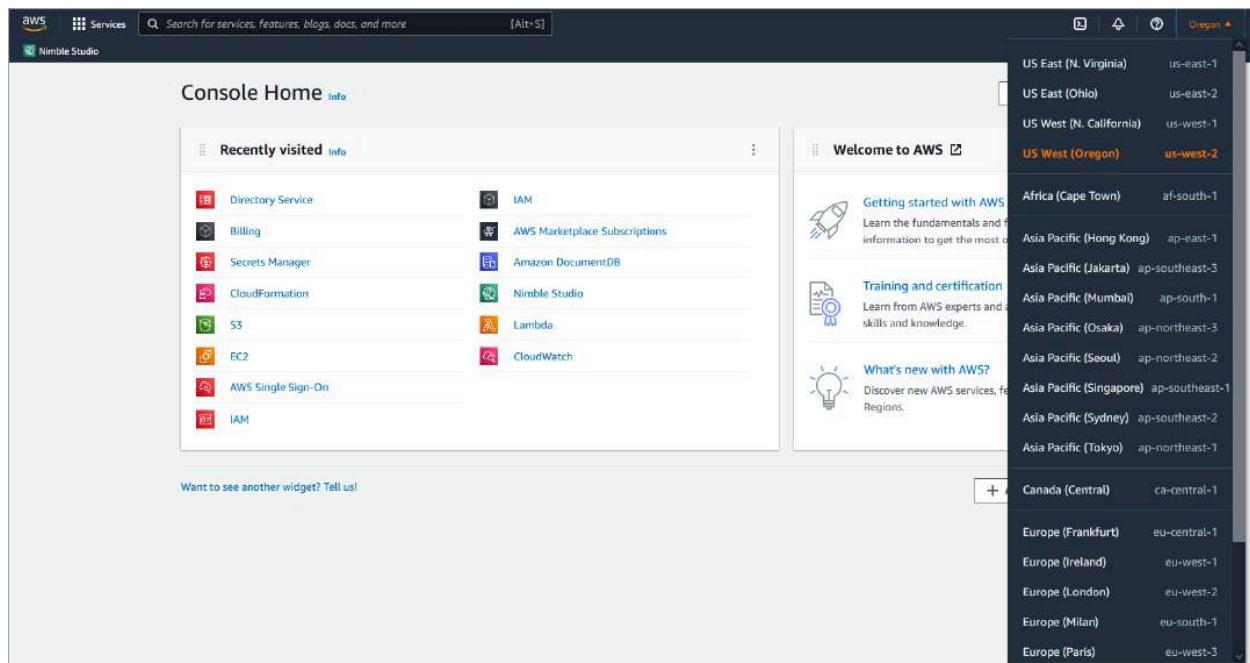
1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Go to Studio manager**.
3. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



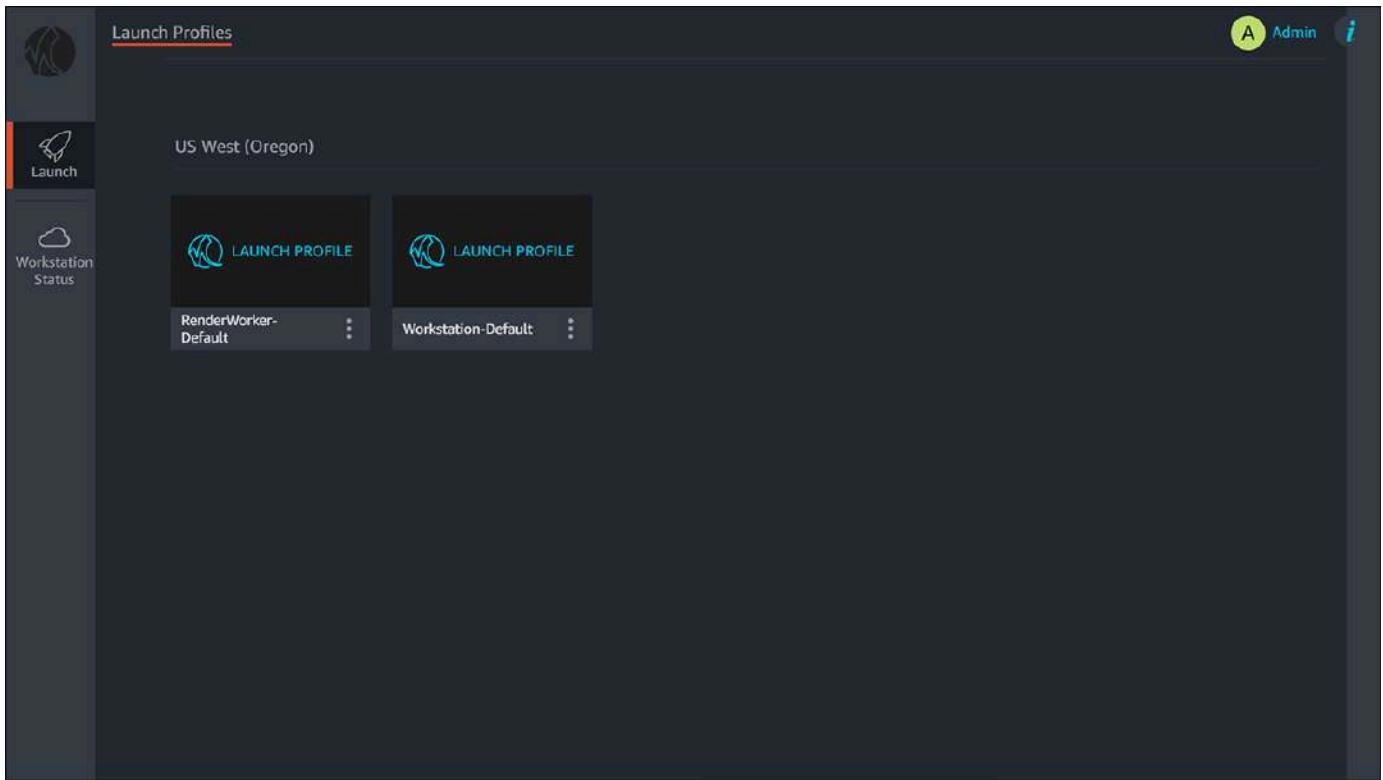
4. On the Studio manager page, choose **Go to Nimble Studio portal**.
5. Sign in to the Nimble Studio portal using your AWS Managed Microsoft AD administrator credentials.
 - a. Enter **Admin** as the user name. Use the password that you set up during StudioBuilder deployment.



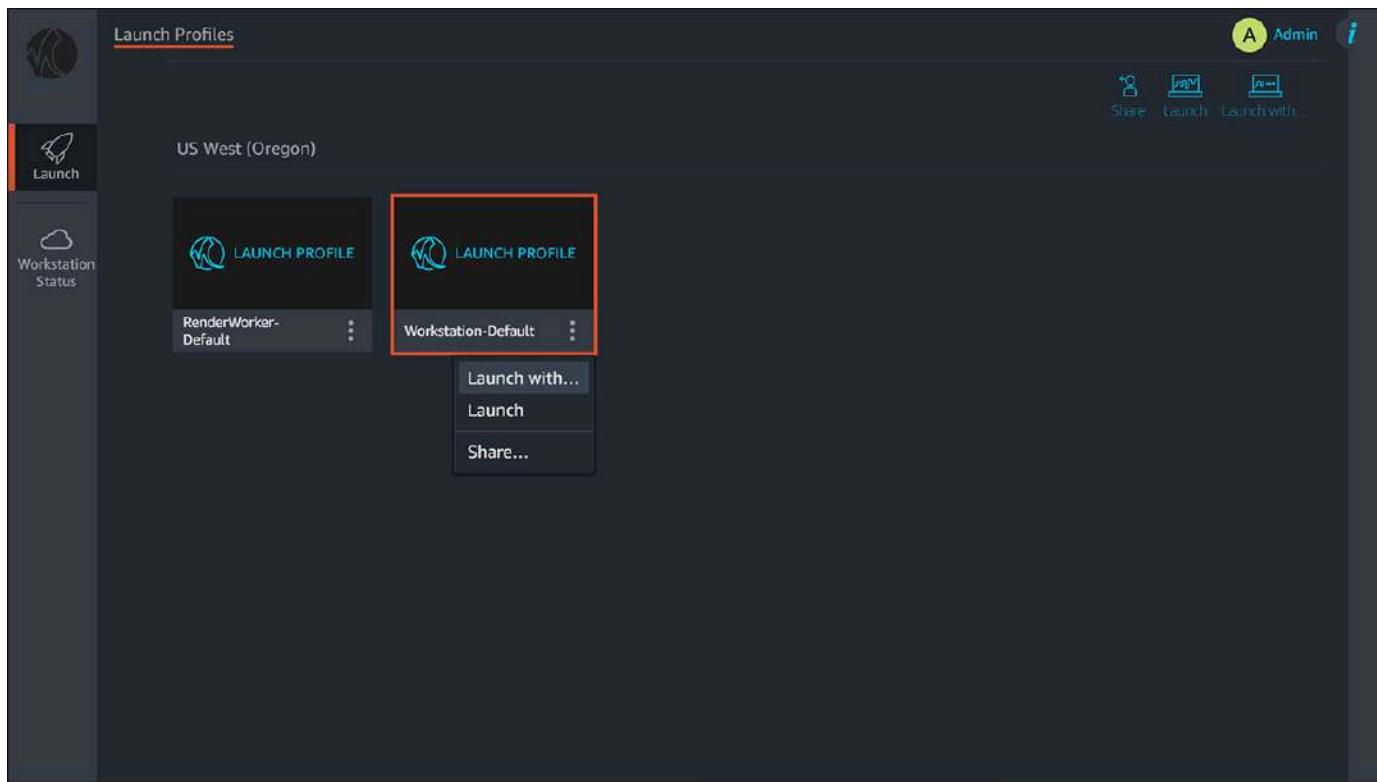
- b. If you forgot your password, do the following:
- Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
 - In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



- iii. Select the **Directory ID** for your studio's Active Directory.
 - iv. Choose **Reset user password**.
6. You will be taken automatically to the **Launch** tab. If not, choose the **Launch** tab from the left navigation pane.



7. Select the vertical ellipsis
(⋮) on the card to open a dropdown menu.



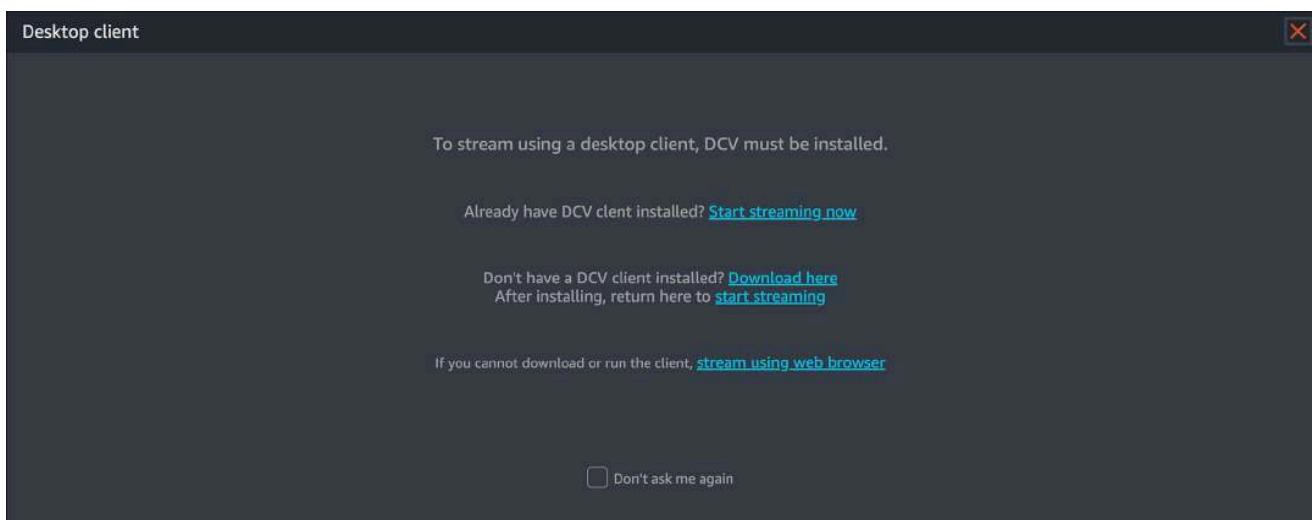
8. Choose **Launch with...**
9. For **Instance Type**, leave it at the default setting.
10. Select **NimbleStudioWindowsStreamImage** for **Amazon Machine Image**.
11. For **Streaming Preference**, choose your streaming preference.
 - a. For the best performance, we recommend choosing **Launch native client**.
 - b. You will have to download the NICE DCV client before connecting to your workstation. For more information about the NICE DCV client, and links to download, see [NICE DCV clients](#).
12. Choose **Launch**.

A status bar will appear that shows you the progress of launching your virtual workstation. This might take up to 10 minutes.

Connect to the virtual workstation

1. When your virtual workstation is ready, a new window will appear reminding you that the client must be installed.
2. Choose **Start streaming now**.

- If you haven't installed the DCV desktop client, choose **Download here** and install the client first.



3. When your browser displays a window that prompts you to open NICE DCV, choose **Open** to continue. The exact wording of this might vary depending on what browser you're using.

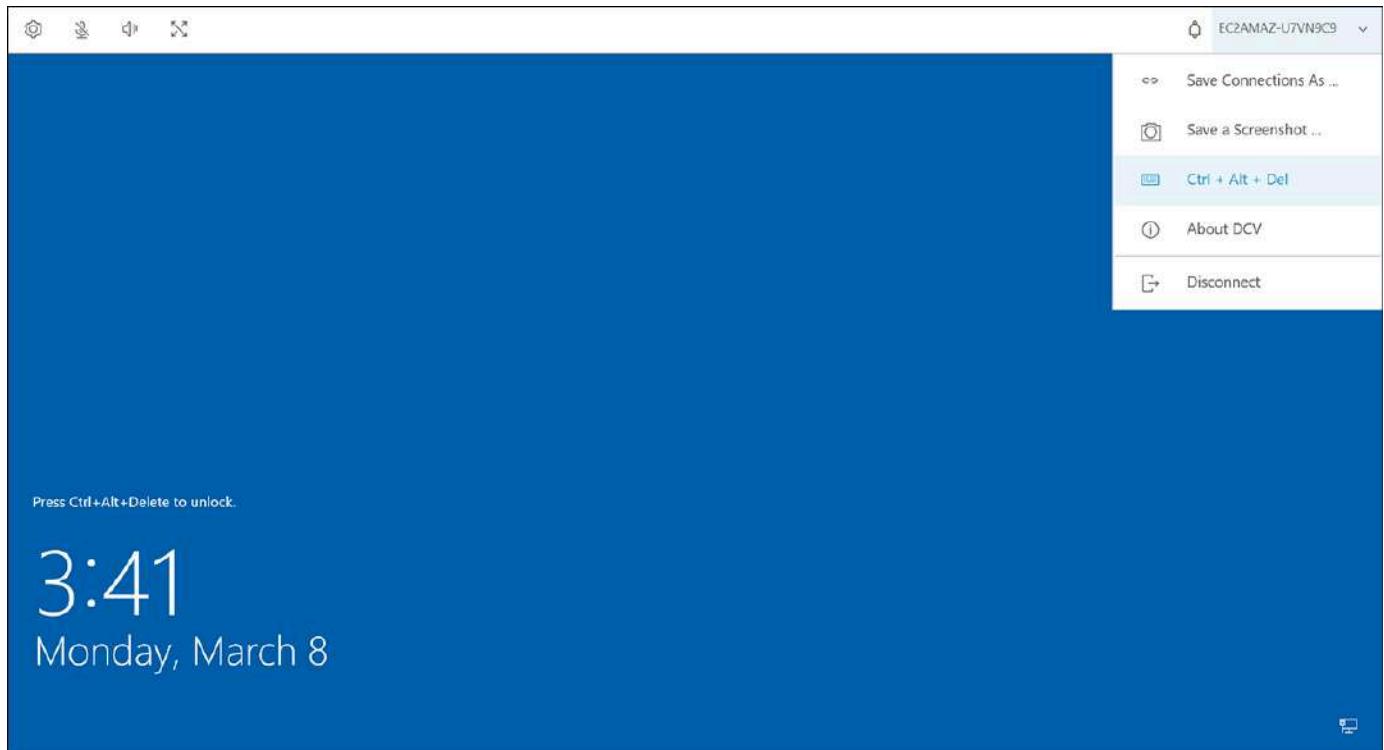
Note

The NICE DCV web browser client runs inside a web browser. You don't need to install the web client. We recommend using the Google Chrome browser to avoid latency. For more information, go to the [Web browser client](#) page in the *NICE DCV User Guide*.

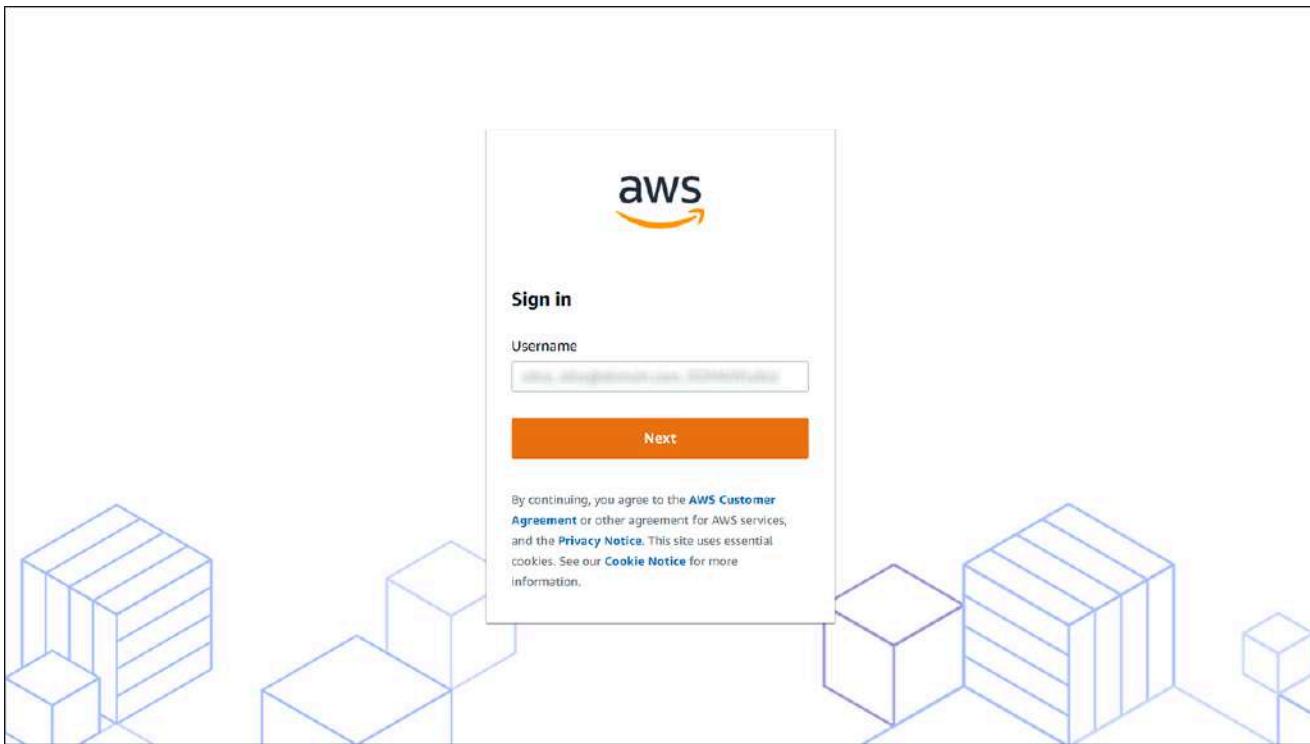
4. After the NICE DCV client application opens in a new window, the Windows login screen will display.
5. Open the instance menu near the top right of the screen and choose **Ctrl + Alt + Del**. For an OS X DCV client, open the **Connection** menu bar and select **Send Ctrl + Alt + Del**.

Important

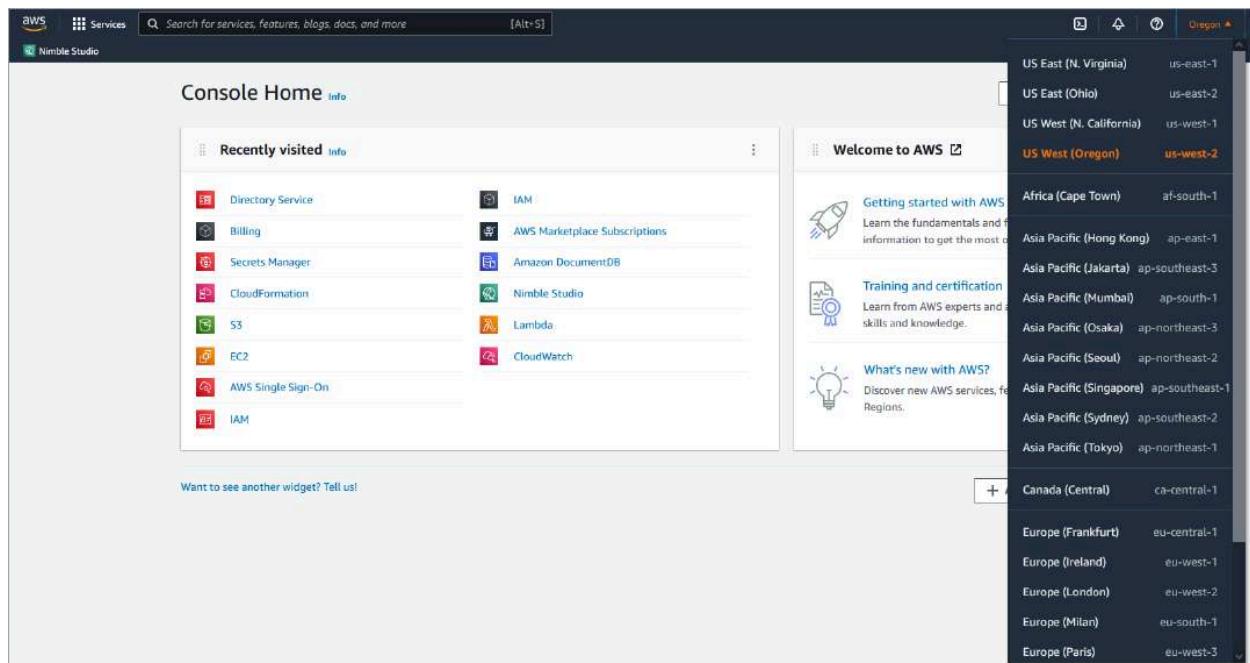
Don't enter **Ctrl+Alt+Delete** on your keyboard. Doing so sends the command to your local computer, not to your workstation.



6. Sign in to the Nimble Studio portal using your AWS Managed Microsoft AD administrator credentials.
 - a. Enter **Admin** as the user name. Use the password that you set up during StudioBuilder deployment.



- b. If you forgot your password, do the following:
- Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
 - In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



- iii. Select the **Directory ID** for your studio's Active Directory.
 - iv. Choose **Reset user password**.
7. After the desktop has loaded, choose the start menu in the lower left corner.
 8. Search for **Deadline Monitor** and select it from the list of items.
 9. If a window pops up and asks if it's ok to create a new user, choose **OK**.

Update test worker IAM role and assign to fleet

When you deployed your studio, StudioBuilder created an IAM role. Update this role to connect to the test worker using Session Manager, and to access S3 bucket data from the instance.

To update the test worker role and assign to your fleet

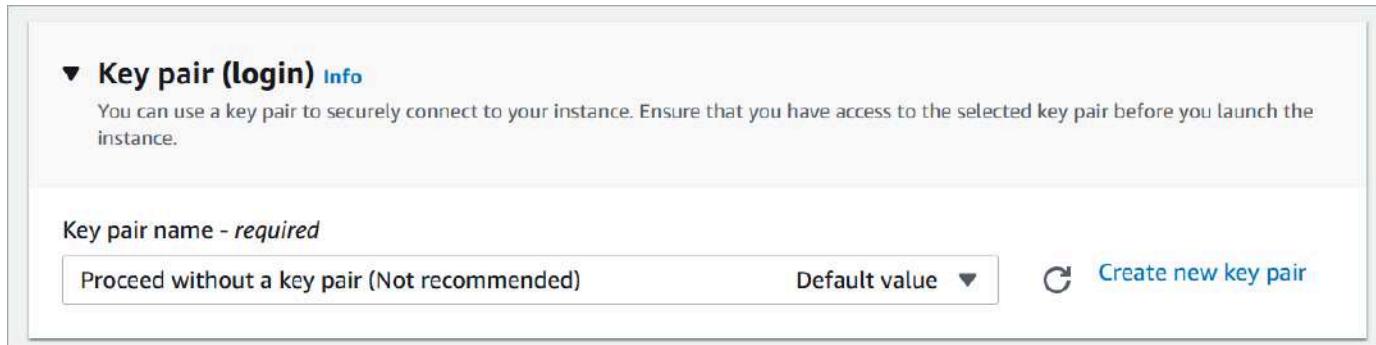
1. Follow the instructions in [To use a managed policy as a permissions policy for an identity \(console\)](#) in the IAM User Guide.
2. Search for **S3** and select the box next to the **NimbleStudioInstallersS3ReadOnly** policy.
 - If you can't find this policy, return to Step 3 of the [Update AMIs: Setting up](#) tutorial and follow the steps to create it.
3. For identity, search for **DeadlineSpot** and select the name of the role called **DeadlineSpot-<region>-<fleet_name>**.
 - a. There will be a role for each fleet in your studio.
 - b. Select the role for the fleet that you want to test.

Launch a test worker

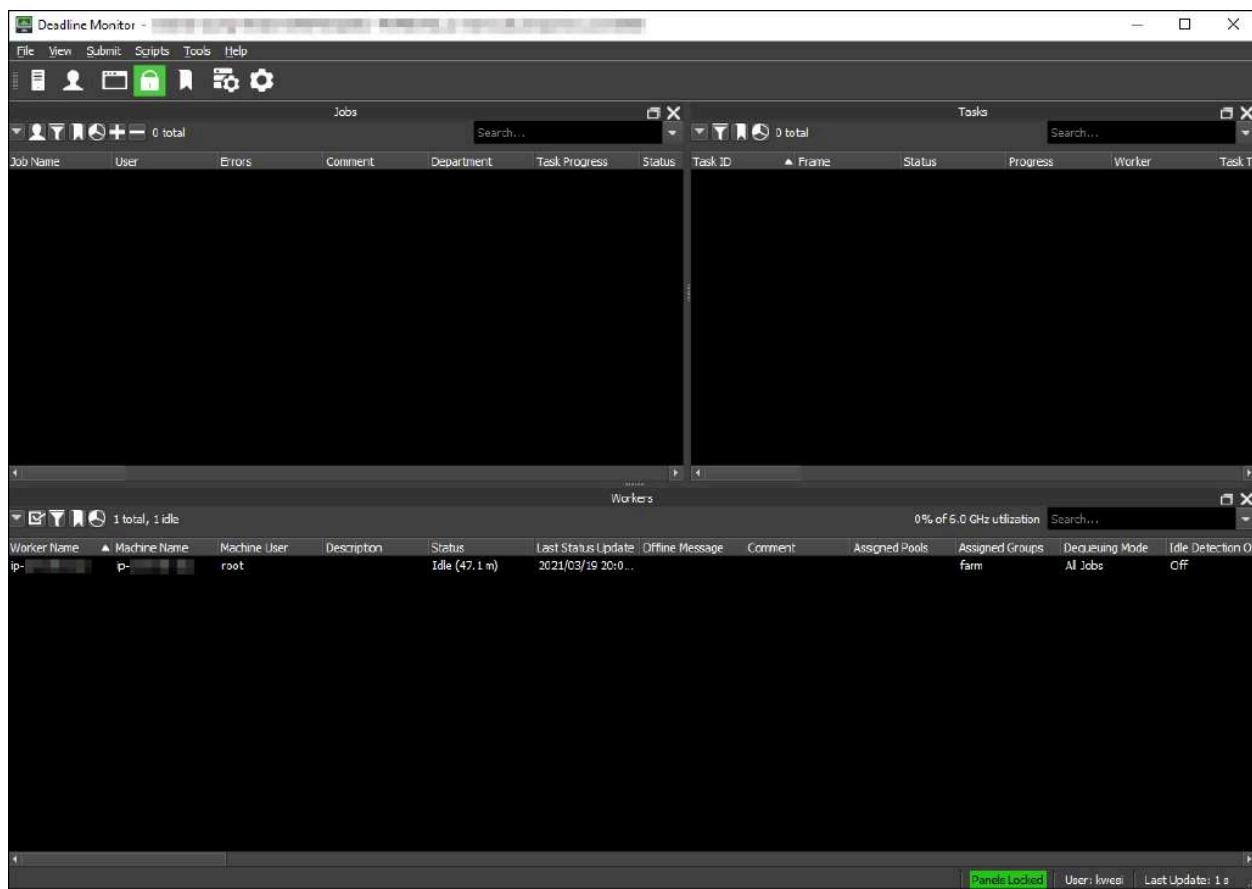
The worker launch template launches a Spot Instance by default. Because of this, it is possible that the instance will be terminated, if it's interrupted. If you prefer to launch an On-Demand Instance, follow these steps:

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Select **Launch Templates** in the left navigation pane.
3. Select the worker **Launch template** for the Linux render worker AMI that you want to update.
4. Notice the **AMI ID** in the **Instance details** tab of the **Launch template version details** section. You will need this AMI ID in [Step 2: Launch an instance for AMI creation](#).

5. Choose **Actions**. Then choose **Launch instance from template**.
6. For **Name and tags**, give the instance a name so that you can easily find it later, such as `worker_TEST`.
7. For **Key pair (login)** choose **Proceed without a key pair** from the first dropdown.



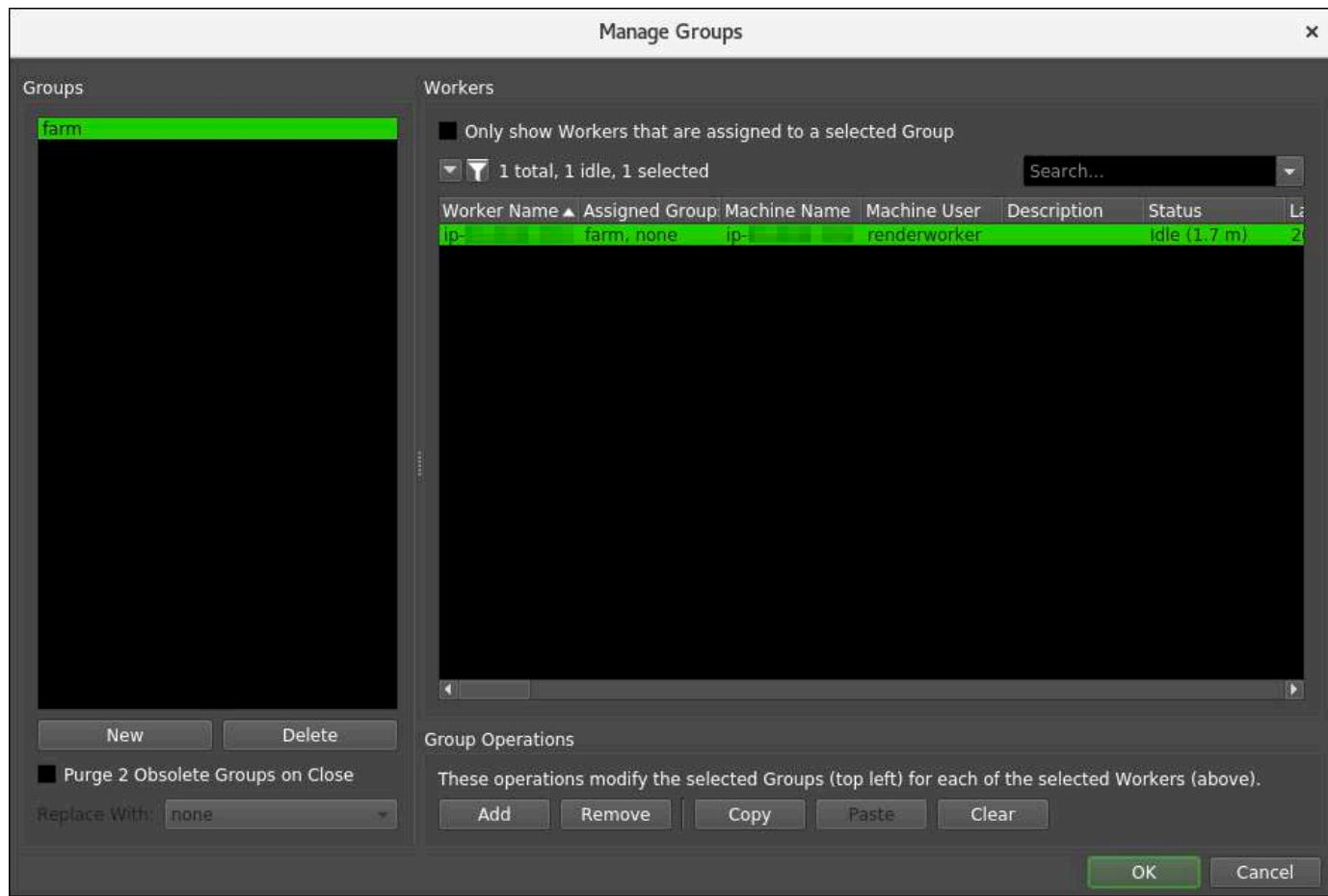
- A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance. You will use Session Manager to connect so you don't need a key pair.
8. Choose **Launch instances**.
 9. Return to the EC2 console.
 - Your newly launched worker instance will be **Running**.
 10. Choose **Instances** in the left navigation pane.
 11. On your virtual workstation, check that the Deadline Monitor shows your worker listed in the bottom left of the pane. It might take a few minutes for the worker to finish initializing and appear in the list.
 - a. If after several minutes, you don't see a worker show up, use the [Troubleshooting](#) guide to start the deadline worker application on the instance.
 - b. If that doesn't work, try launching your test worker again. If you're still having trouble, consult the [AWSThinkboxDeadline Documentation](#).
 - Make sure that the documentation version matches the version of Deadline that you're using in your studio.



Remove the group from the worker

When the worker spins up, it will automatically be a part of the fleet group that you created during your Nimble Studio cloud studio deployment. This worker will terminate automatically, unless you remove the group assignment.

1. Choose **Tools**. Then choose **Super User Mode**.
2. Select the worker in the list of workers in Deadline Monitor.
3. Open the context menu (right-click) for the worker, and then choose **Manage groups**.
4. Select the **group** in the left pane, and select the **worker** in the right pane.



5. Choose **Remove** to remove the group from your worker.
 - The **Assigned Group** for the worker will change to **none**.

Create a new group for the worker

1. Choose **New** to create a new group.
2. Enter **UPDATING-AMI-admin-only** for the group name and choose **OK**.
3. Select the **new group** in the left pane, and select **worker** in the right pane.
4. Choose **Add** to assign the worker to the new group.
5. Choose **OK** to exit the Manage Groups window.
 - a. The **Assigned Group** will be reassigned after a few seconds.
 - b. The worker should now stay active until you terminate it manually.

Prevent the worker from rendering jobs from other users

Even though you've assigned the worker to a new group, there is still a chance that it could pick up render jobs if another user from your studio submits them to the **none** group.

To prevent the worker from rendering jobs from other users, follow these steps:

1. check that your test worker is selected in the list of workers.
2. Open the context menu for the worker (right-click), and then choose **Manage Worker Properties...**
3. Choose **Job Dequeuing** in the left navigation pane.
4. Choose **Only Jobs Submitted From These Users**.
5. In the **User List**, choose **admin**, then choose the **right arrow** to move it to **Job Users**.
6. Choose **OK**.

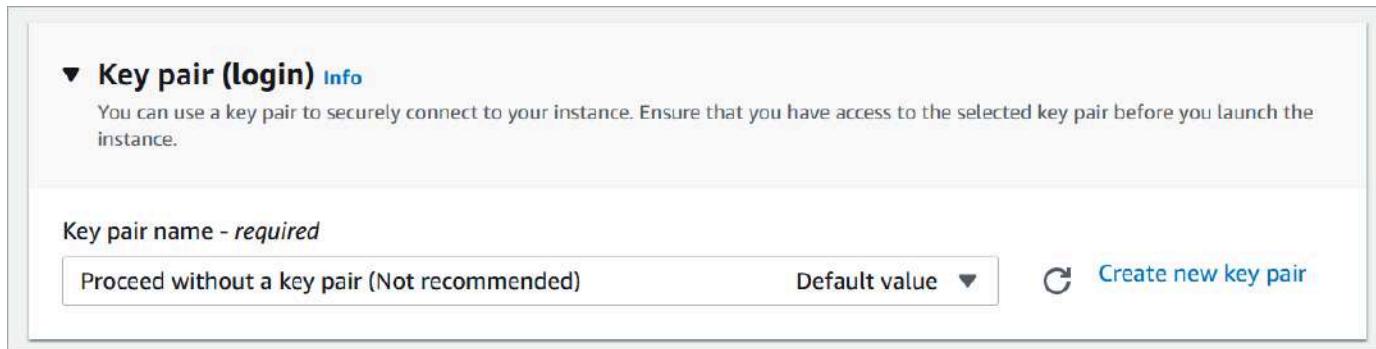
Now your test worker will only render jobs that are submitted by the administrator user.

Step 2: Launch an instance for AMI creation

To launch the instance for AMI creation outside of your Nimble Studio environment, follow these steps:

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **AMIs** in the left navigation pane in **Images**.
3. In the search field, enter the **AMI ID** from the **Launch a test worker** section of [Step 1: Prepare your test environment](#).
 - If you see a message that says No AMIs found matching your filter criteria, open the dropdown menu to the left of the search field and choose one of these two options:
 - i. If this is your first time updating this AMI, select **Public images**
 - ii. If you've updated this AMI before, select **Owned by me**.
4. Make sure that the AMI is selected in the search results and choose **Launch**.
5. For **Name and tags**, give the instance a name so that you can easily find it later, such as **worker_AMIBUILD**.
6. For **Instance Type**, select **c5.large**.
7. Leave **VPC** as **vpc-<id>** (default).

8. Leave **Subnet** as No preference (default subnet in any Availability Zone).
9. Set **Auto-assign Public IP** to **Enable** so that your instance receives a public IP address that you will use when connecting to it later.
10. From the IAM role dropdown, choose the **Nimble_Studio_Build_AMI** profile that you created in [Prerequisites](#).
11. Leave the storage settings as default.
12. Leave **Create a new security group** selected.
 - a. You can remove the security group rule by choosing the box to the left of the rule.
 - b. You will be using port forwarding to connect to this instance and won't need any security group rules to connect.
 - c. You don't need any inbound rules, since you will use Session Manager to connect to this instance.
13. For **Key pair (login)** choose **Proceed without a key pair** from the first dropdown.



- A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance. You will use Session Manager to connect so you don't need a key pair.

 14. Choose **Launch**.
 15. Return to **EC2**.
 16. Choose **Instances** in the left navigation pane.

You will now see both of your worker instances running.

Step 3: Update the TEST worker

In this section, you will update the **TEST worker** with new software. Notice all of the steps that you take to install your software because you will need to repeat them when you update the instance for AMI creation.

To download installers on the TEST worker

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select the instance called `worker_TEST`.
4. Choose **Connect**.
5. Select **Session Manager** and then choose **Connect**.
6. A new tab will open and you'll be connected to your TEST worker instance.

To install applications using installers on Amazon S3

1. Make a temporary folder in `/tmp` for your downloaded installer.

```
sudo mkdir /tmp/installer  
sudo chmod a+w /tmp/installer
```

2. Sign in to the AWS Management Console and open the [Amazon S3](#) console.
3. Navigate to the installer that you want to install on your worker.
4. Select the installer's name.
5. Choose the copy icon next to the **S3 URI** to the left side of the **Object overview**.
6. Return to the **Session Manager** for your instance to download the installer.
7. Run the following command with the S3 URI updated to match the one that you copied from your S3 bucket.

```
cd /tmp/installer  
aws s3 cp S3 URI .
```

8. Install the application into the correct location on the instance.

⚠️ Important

Make sure to run installers using the sudo command, otherwise they might not install correctly.

9. For example, after downloading the installer for Blender 2.90.0, run these commands to install it:

```
sudo tar xf blender*.xz -C /opt/
sudo rm -rf /usr/local/Blender
sudo ln -s /opt/blender-2.90.0-linux64 /usr/local/Blender
```

- Remove the temporary folder by running the following command: sudo rm -rf /tmp/installer

Step 4: Validate the update

Verify that the updates that you made to your render farm are working.

1. Switch back to the NICE DCV window that is connected to the workstation that you launched earlier.
2. Launch a render using the application that you just installed on the **TEST worker** instance.
3. On your workstation, submit a render.
 - When you launch your render, set your **Group** to **UPDATING-AMI-admin-only**.
 - This render will get picked up by your TEST instance.
4. If your render successfully completes, repeat the installation steps from the TEST instance on this AMIBUILD instance.

After you validate that the installations and changes to your TEST worker instance are working, you will need to repeat the steps on the AMIBUILD worker instance before creating a new worker AMI in [Step 5: Deploy studio with StudioBuilder](#).

Step 5: Update the AMIBUILD instance

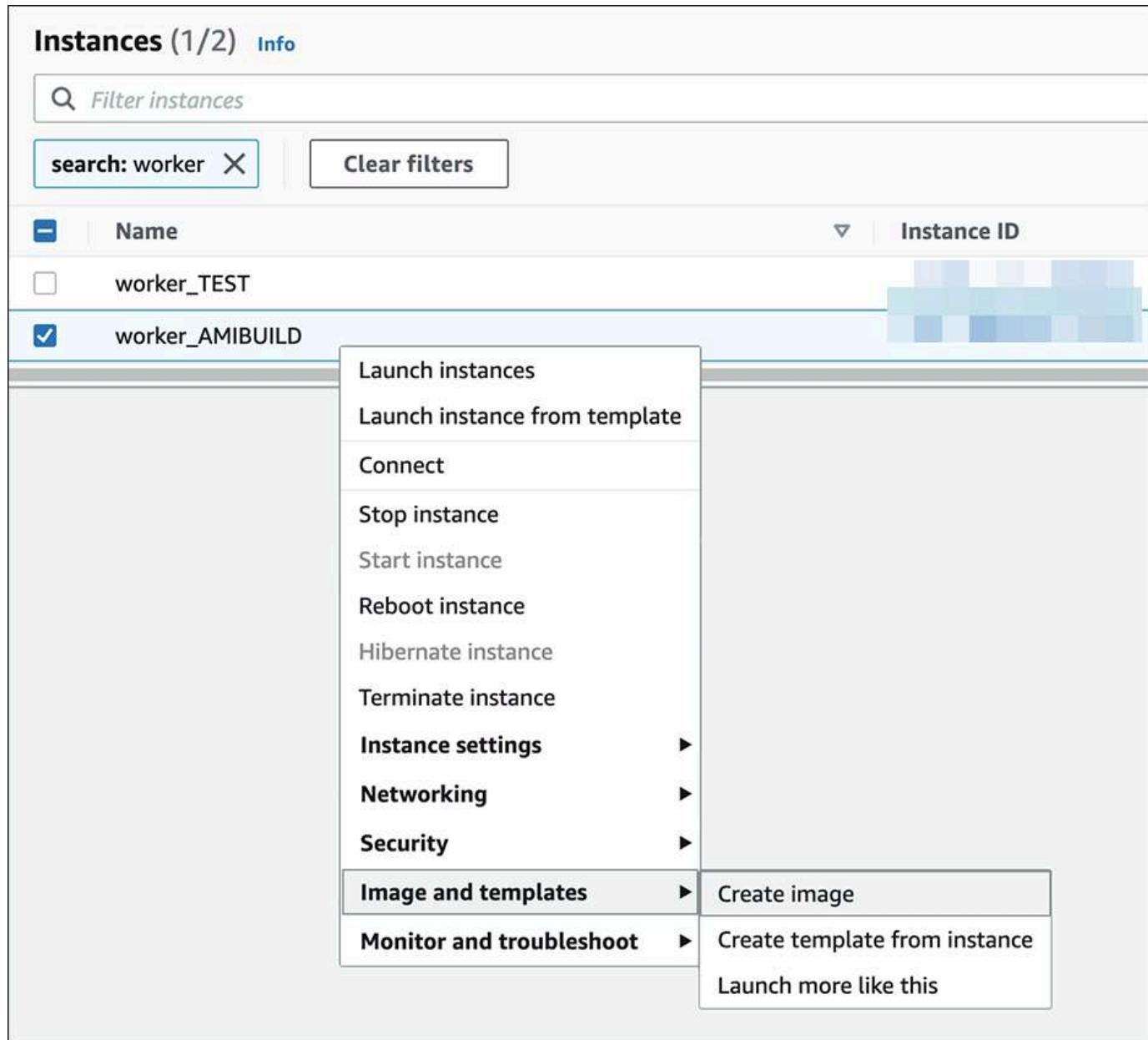
1. Perform the same installation steps on the AMIBUILD instance that you performed on your TEST worker instance [Step 3: Update the TEST worker](#).
2. Remove the temporary folder by running the following command: `sudo rm -rf /tmp/installer`

Step 6: Create the new AMI

In this step, you will create a new AMI that your farm will use, so that you can render with the updated software.

Creating a new worker AMI

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Right-click on the AMIBUILD instance.
4. Hover over **Image and templates**.



5. Choose **Create image**.

6. Give your AMI a name:

- To help you keep track of the different AMIs that you create, it's a good idea to give them descriptive names. Descriptive names should include the operating system, intended use (worker), and a date or version number.
 - Example: <your studio name>-linux-worker-2021-03-11

7. Give your AMI a good description:

- To make your image description, you can include what you changed on this AMI, what makes it unique, or the new software that you installed.
 - Example: Linux worker with Blender 2.92.0
8. Increase the **Volume** size, if necessary.
 9. Navigate to the bottom, then choose **Create image**.
 10. Choose **AMIs** in the left navigation pane in **Images**.
 11. Your new AMI will be in the list with a **Status of pending**. When the status changes to **available**, you can continue to the next step. This process can take 10-20 minutes, depending on the amount of software installed on your instance.
 12. You might also want to add a name to your AMI by hovering over the **Name** field and choosing the **edit icon**.
 13. Notice the **AMI ID** for your AMI because you will use it in the next section of the tutorial.

Step 7: Use StudioBuilder to update your render farm fleet

To update your render farm fleet to use the new AMI that you created, you will use **StudioBuilder** to modify the fleet properties.

Connect to the StudioBuilder instance

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select your StudioBuilder instance.
 - If you don't have an instance of StudioBuilder running, follow the tutorial [Deploying a new studio with StudioBuilder](#) tutorial to get an instance of StudioBuilder running in your account.
4. Connect to your instance using one of the techniques specified in the [Deploying a new studio with StudioBuilder](#) tutorial, either through SSH or EC2 Instance Connect.
5. Make sure to connect as ec2-user.

Start StudioBuilder to update your resources

1. When you connect to your instance, a **Welcome back** prompt will display, giving you options to work with your studio.
 - If you don't see the prompt, type `studio_builder` and press the enter (or return) key to launch the StudioBuilder tool.
2. Use the arrow keys to choose **Update an existing resource**.
3. StudioBuilder will display your current configuration.
4. Press **Y** to edit the configuration and press the enter (or return) key to continue.
5. StudioBuilder will remember most of your previous selections, so you can press the enter (or return) key to accept earlier choices. Continue until you get to the first question about the farm.
6. **Would you like to modify or delete fleet: <farmname>? Please select an option.**
 - Choose **Modify**
7. **Which type of fleet would you like?**
 - To accept your previous choice press the enter (or return) key.
8. **Which Operating System will this fleet use?**
 - To accept your previous choice press the enter (or return) key.
9. **Enter the AMI ID for this Render Worker fleet.**
 - a. Delete the **AMI-ID** that is listed.
 - b. Enter the **AMI-ID** of your new worker AMI.
10. Continue through the rest of questions and press the enter (or return) key to accept the defaults unless you would like to change them.

Review

- **Would you like to generate a studio configuration with your selections?**
 - If you're happy with all the selections you made in the preceding steps, enter **Y** and press the enter (or return) key to proceed. Enter **N** and press the enter (or return) key to go back and make changes.

Ready to deploy your studio build

- Please type **BUILD MY STUDIO** (and then press enter) to continue, or type **QUIT** (and then press enter) to exit
 - Type **BUILD MY STUDIO** and press the enter (or return) key to continue.
 - i. StudioBuilder will run the deploy to update all the components of your studio.
 - ii. It will take approximately 10 minutes to complete updating your studio.

Once your update is complete

After StudioBuilder has finished running, you will be asked what you want to do next. After that, you can close the StudioBuilder browser tab and [terminate your StudioBuilder instance](#). We recommend terminating unused instances to prevent incurring costs.

Step 8: Test your deploy

Now, when you submit renders to your farm, the new AMI that you created will be used to spin up the render workers. To test whether the correct AMIs are being used, we will launch a test render. During this process, we will check the worker to verify that it's using the correct AMI.

1. Launch a new render following the workflow from the [Creating your first render on the farm](#) tutorial.
 - Wait a few minutes for workers to spin up.
2. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
3. Choose **Instances** in the left navigation pane.
4. A few workers will be spinning up in the list of instances. Select one of the new workers.
5. Choose the **Details** tab.
6. Open **Instance Details** to see more information about the selected instance.
7. Find the **AMI ID** for your instance. It should match the ID of the AMI that you used in StudioBuilder.

Step 9: Terminate the Worker_AMIBUILD and Worker_TEST instances

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.

3. Select the **Worker_AMIBUILD** and the **worker_TEST** instances.
4. Choose **Instance state**. Then choose **Terminate**.

Troubleshooting

The render worker that I created doesn't appear in the Deadline Monitor

Consult the [AWS ThinkboxDeadline Documentation](#). The launch logs might provide additional information about why you are receiving errors. You can find the launch logs in the /var/log/cloud-init-output.log file.

You can also perform a test launch [Step 2: \(Optional\) Perform a test launch](#) to verify that there aren't any problems with the launch profile.

My render worker's status is Offline in the Deadline Monitor.

If your render worker is idle for too long, the worker application running on the instance will shut down, and the worker will be listed as offline in the Deadline Monitor. To bring it back online, follow these steps:

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select the instance called **worker_TEST**.
4. Choose **Connect**.
5. Select **Session Manager** and then choose **Connect**.
6. In the new tab with Session Manager running, enter these commands:

```
sudo -i  
cd /opt/Thinkbox/Deadline10/bin  
../deadlineworker --nogui
```

Software specific installation tips

Your studio's Amazon Machine Images (AMIs) have a suite of digital content creation applications that your studio uses. Some software and applications require certain steps during or after installation. Take these steps so that the software and applications install and run correctly on the virtual workstations that your artists use.

Note

Amazon operates using the [Shared Responsibility Model](#), wherein the customer assumes responsibility and management of the guest operating system (including updates and security patches), associated application software, and firewall configuration.

The following list of tutorials will help make it easier for you to customize your AMIs.

Contents

- [Adobe Creative Cloud](#)
- [Autodesk](#)
- [Black Magic Design](#)
- [Chaos group](#)
- [Foundry](#)
- [NVIDIA](#)
- [SideFX](#)
- [Substance](#)
- [Set up Adobe Creative Cloud on Nimble Studio](#)
- [Set up Blackmagic Design DaVinci Resolve 17 on Nimble Studio](#)
- [Set up Perforce Helix Core on Nimble Studio](#)
- [Set up Incredibuild on Nimble Studio](#)

Adobe Creative Cloud

Follow the tutorial for [Set up Adobe Creative Cloud on Nimble Studio](#) to verify that it installs and runs properly.

Autodesk

Instructions for downloading and installing Autodesk products: [Download & Install](#)

Black Magic Design

Follow the tutorial for [Set up Blackmagic Design DaVinci Resolve 17 on Nimble Studio](#) so that it installs and runs properly.

Chaos group

Instructions for installing Vray: [Installation and Licensing](#)

Foundry

Instructions for installing Nuke: [Install Nuke](#)

Note

If you plan to use Nuke on a Windows workstation, follow the instructions in [How to resolve the Nuke dependency issue in Windows Server](#) in the Foundry Knowledge Base. This guide resolves an issue with Nuke on a Windows Server where you get the error message, failed to load studio-#.dll, when you launch Nuke.

Instructions for installing Katana: [Install Katana](#)

NVIDIA

To use G5 streaming sessions, confirm that the GRID version is 13.1 or later. We also suggest you have the latest version of NVIDIA drivers in the AMI. You can either update your custom AMIs by using the latest Workstation AMIs, or by updating the NVIDIA drivers by yourself.

To update the NVIDIA drivers, follow *Option 2: Public NVIDIA drivers* or *Option 3: GRID drivers (G5, G4dn, and G3 instances)* in the [Install NVIDIA drivers on Linux instances](#) tutorial or in the [Install NVIDIA drivers on Windows instances](#) tutorial. Choose the tutorial that matches the operating system of your AMI.

SideFX

Instructions for installing Houdini for Linux: [Installing Houdini Linux](#)

Instructions for installing Houdini for Windows: [Installing Houdini Windows](#)

Substance

Information about downloading and installing the Substance Launcher: [Installing Substance Launcher](#)

When running Substance Painter on a virtual workstation you need to update the registry to keep the GPU drivers from crashing when there are long computations. After installing Substance Painter, follow the instructions in [GPU drivers crash with long computations \(TDR crash\)](#) to update the registry.

Set up Adobe Creative Cloud on Nimble Studio

To successfully install Adobe Creative Cloud on an Amazon Elastic Compute Cloud (Amazon EC2) instance and have it work as part of an Amazon Machine Image (AMI), there are some specific steps that you need to follow. This tutorial describes those steps.

Contents

- [Prerequisites](#)
- [Step 1: Enable JavaScript](#)
- [Step 2: Adjust Internet Explorer security settings](#)
- [Step 3: Enable the Secondary Login Service](#)
- [Step 4: Install software](#)
- [Step 5: Remove Adobe Notification client](#)
- [Step 6: Complete the AMI creation process](#)
- [Related resources](#)

Prerequisites

- You need access to the login credentials for an Adobe Creative Cloud subscription plan.
- Sign in to a Windows virtual workstation that you launched from the default Windows AMI. For help, see the [Update Windows workstation AMI](#) tutorial for instructions.
 - Install Firefox.

Important

We discourage using Internet Explorer because other browsers offer better security. We recommend that you install Firefox before you create AMIs or launch an instance.

Step 1: Enable JavaScript

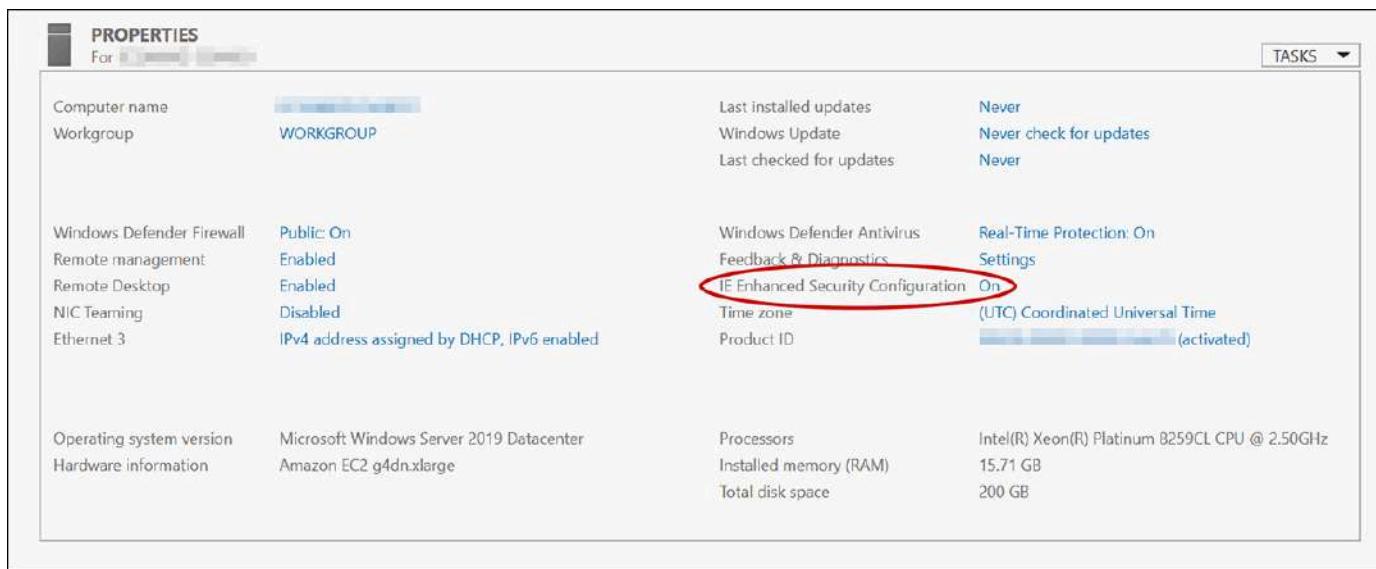
To successfully install Adobe Creative Cloud, you'll need to have JavaScript enabled in Microsoft Internet Explorer on your virtual workstation.

1. On your Windows virtual workstation, follow the [Enable JavaScript](#) instructions from Adobe.
2. Close the Internet Explorer window after you've enabled JavaScript.

Step 2: Adjust Internet Explorer security settings

Disable the Internet Explorer Enhanced Security Configuration so that websites that are accessed during the Adobe Creative Cloud installation process can display correctly.

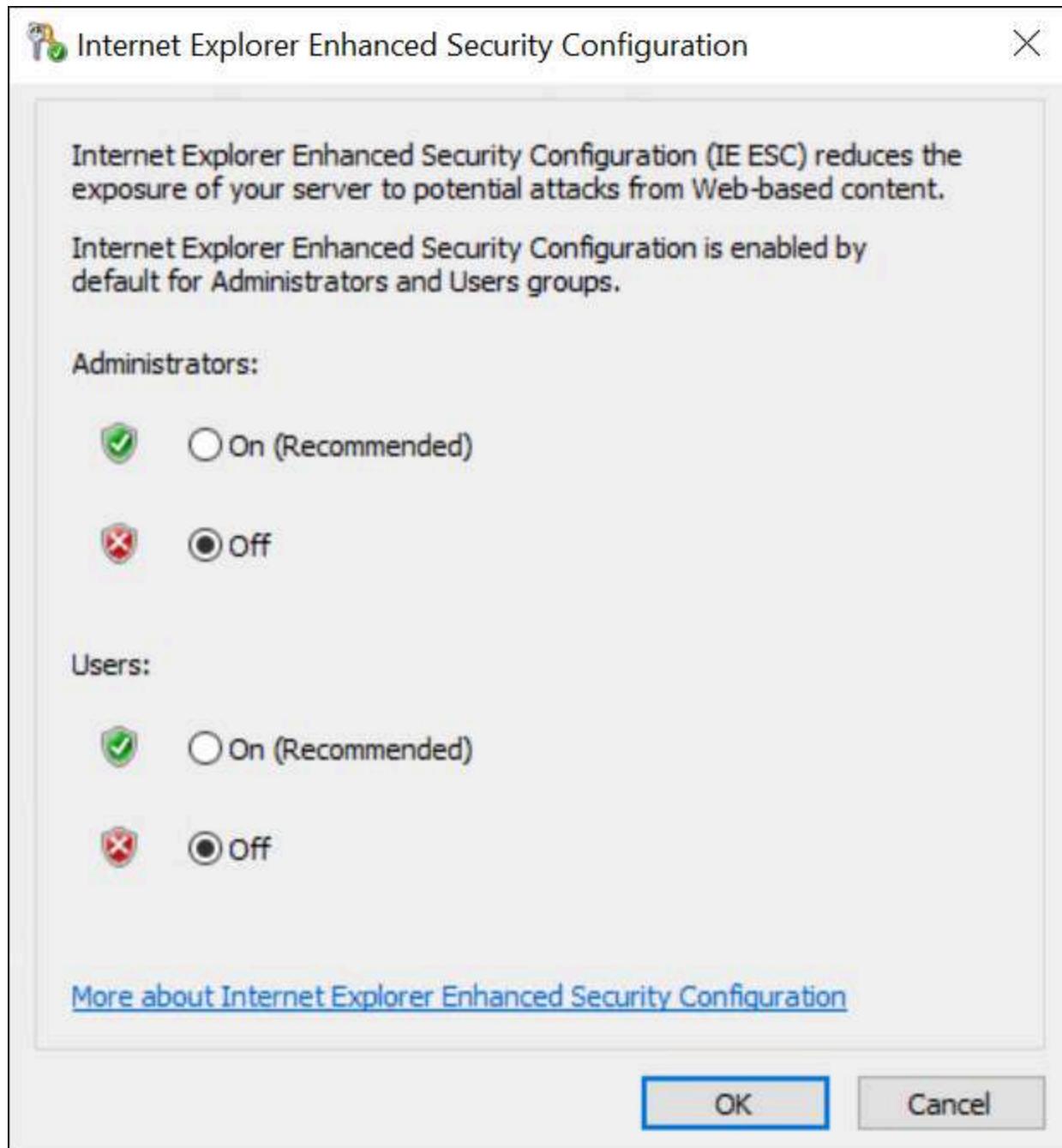
1. Choose the **Start Menu** and search for **Server Manager**.
2. Select **Server Manager** from the list.
3. In the **Server Manager** window, choose **Configure this local server**.
4. On the top of the menu bar, in the Properties section, look for a setting labeled **IE Enhanced Security Configuration (IE ESC)** and choose **On**.



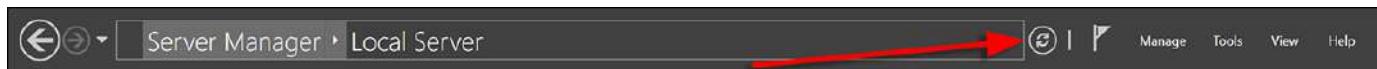
5. In the window that appears, select **Off** for both **Administrators** and **Users**, and then choose **OK**.

Note

Deactivating IE ESC poses a security risk if you don't have an alternate browser installed. Before you create AMIs or launch an instance, we recommend that you install Firefox. We don't recommend using Internet Explorer.



6. Choose the **refresh button** in the top menu of the **Server Manager** window.



7. After refreshing, the **IE Enhanced Security Configuration** should be set to **Off**.
8. Close **Server Manager**.

Step 3: Enable the Secondary Login Service

To install Adobe products, it's important to enable the Secondary Login service, otherwise the install might error out.

1. On your Windows desktop, choose the **Windows icon** and enter **services**.
2. Open the context menu (right-click) for the application **Services** and choose **Run as administrator**.
 - If prompted for your user name and password, enter those.
3. Navigate through the list of services until you find **Secondary Login** for the **Name Column**.
4. Open (double-click) the Secondary Login name to open the Secondary Login Properties window.
5. For **Startup type**, choose **Automatic**.
6. Choose **OK**.
7. Close the **Services** window.

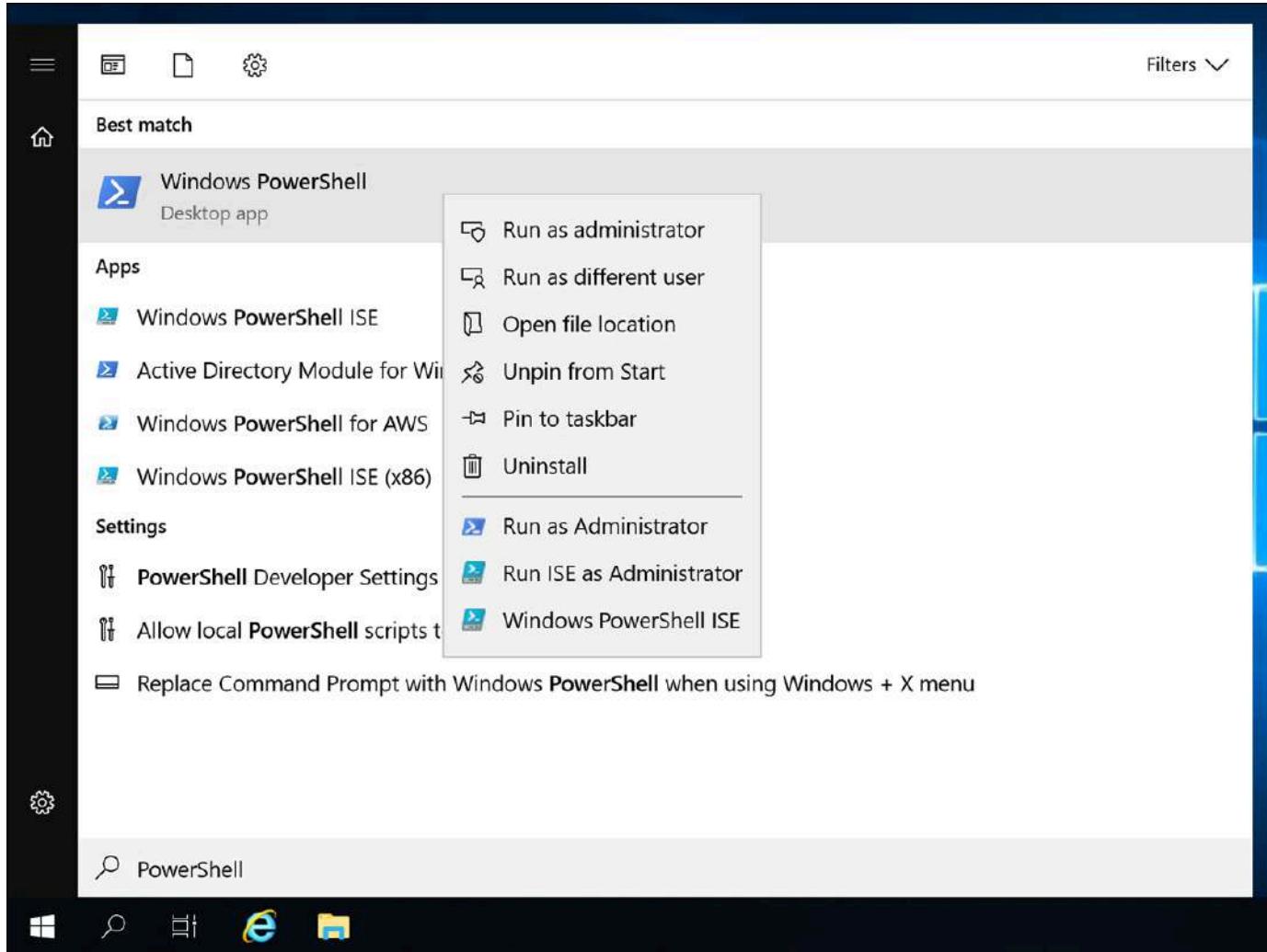
Step 4: Install software

1. Open (double-click) **Firefox** using the shortcut on the desktop.
2. Sign in to your [Adobe Creative Cloud](#) account.
3. Download the installers for the Adobe Creative Cloud applications that you want on the virtual workstation.
4. Install the applications.
 - Make sure to run the Adobe Creative Cloud application as an administrator by choosing the context menu (right-click) on the Creative Cloud application and choosing **Run as administrator**.

Step 5: Remove Adobe Notification client

The Adobe Notification client that's installed as part of Adobe Creative Cloud causes the AMI creation process to fail. As a result, remove it from your virtual workstation.

1. Choose the **Start Menu** and search for **PowerShell**.
2. Open the context menu (right-click menu) on **Windows PowerShell**, then choose **Run as administrator**.



3. Run the following command in PowerShell: `Get-AppxPackage -Name AdobeNotificationClient | Remove-AppxPackage`

Step 6: Complete the AMI creation process

Now that you have your Adobe Creative Cloud applications installed, return to [Step 2: Launch an instance with Windows AMI](#) to finish preparing your virtual workstation for AMI creation.

Related resources

- [Adobe - Enable JavaScript](#)

Set up Blackmagic Design DaVinci Resolve 17 on Nimble Studio

[Blackmagic Design DaVinci Resolve 17](#) combines non-linear editing, color correction, visual effects, motion graphics, and audio post production into a single powerful software package. This tutorial will take you through the process of installing DaVinci Resolve 17 on an Amazon Elastic Compute Cloud (Amazon EC2) instance for use with Amazon Nimble Studio.

Important

Visiting and downloading from third-party sites is done at your own discretion and risk, and you will be solely responsible for any damage, loss, or risks that you experience as a result of downloading, installing, using, modifying, or distributing software from third-party sites.

Contents

- [Prerequisites](#)
- [Step 1: Download the DaVinci Resolve 17 software](#)
- [Step 2: Extract the installation file](#)
- [Step 3: Install the software](#)
- [Step 4: Update NICE DCV settings](#)
- [Step 5: Complete the AMI creation process](#)
- [Troubleshooting](#)
- [Related resources](#)

Estimated time: 35 minutes

Prerequisites

- Sign in to a Windows virtual workstation that you launched from the default Windows Amazon Machine Image (AMI). For help, see the [Update Windows workstation AMI](#) tutorial.

Step 1: Download the DaVinci Resolve 17 software

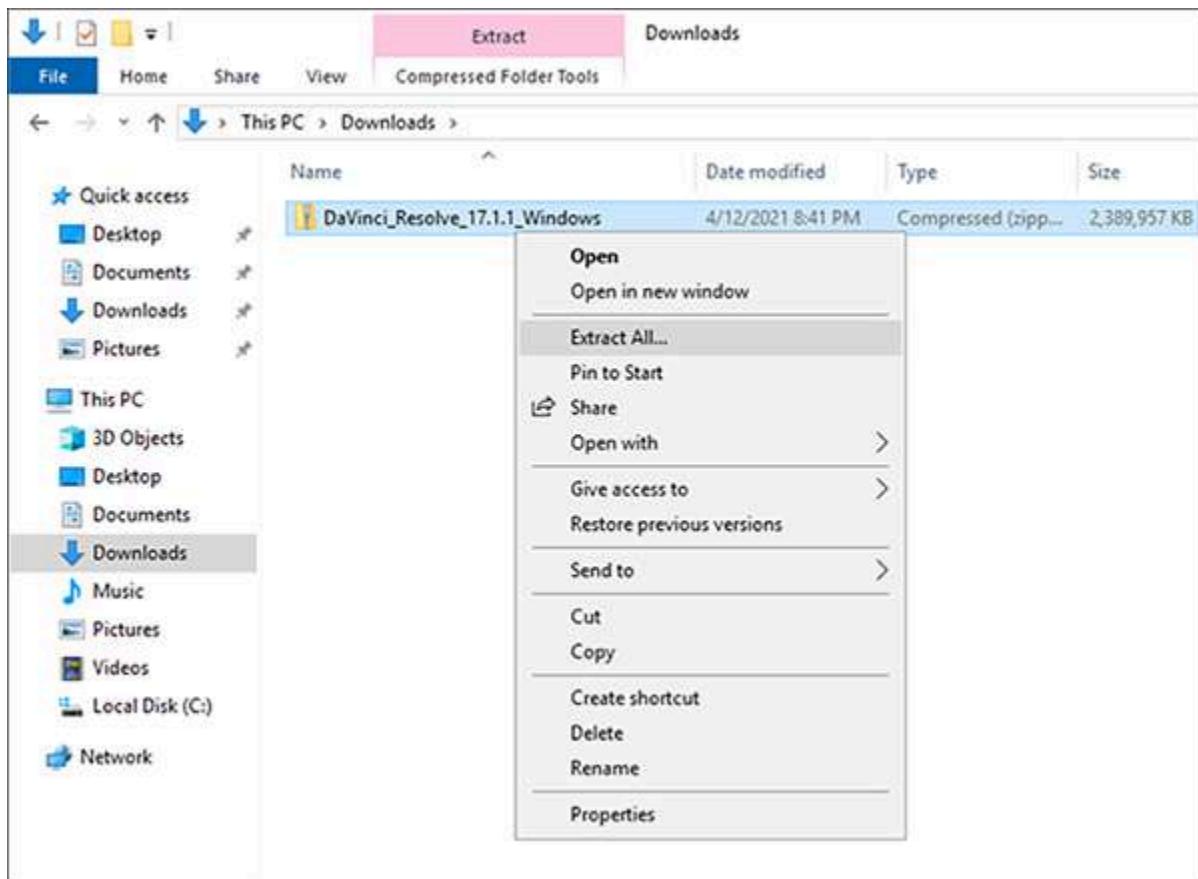
In this step you will register with Blackmagic Design and download the software installer zip file.

1. Open (double-click) **Firefox** using the shortcut on the desktop.
2. Navigate to the [Blackmagic Design DaVinci Resolve website](#) and download **DaVinci Resolve 17 for Windows**.
3. Enter the required information to register with Blackmagic Design.
4. Choose **Register & Download**.
5. Save the file to your **Downloads** folder.

Step 2: Extract the installation file

To install DaVinci Resolve 17, you will extract the contents of a zip file to your **Downloads** folder.

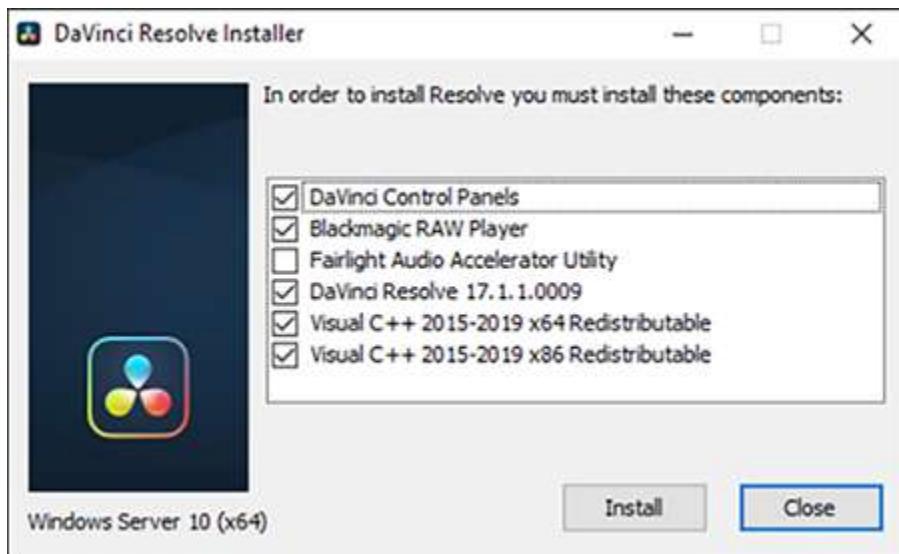
1. Choose the **File Explorer** icon on the toolbar.
2. Choose **Downloads** in **Quick Access**.
3. Open the context menu (right-click) on the **DaVinci_Resolve_17** file, then choose **Extract All...** to open the Extract window.
4. Choose **Extract** to extract files to the specified folder.



Step 3: Install the software

After the files are finished extracting, the DaVinci Resolve installer will display.

1. Right-click on the DaVinci_Resolve_17.x.x_Windows installer and choose **Run as Administrator**.
2. After the **DaVinci Resolve Installer** window appears, choose the appropriate options and then choose **Install**.

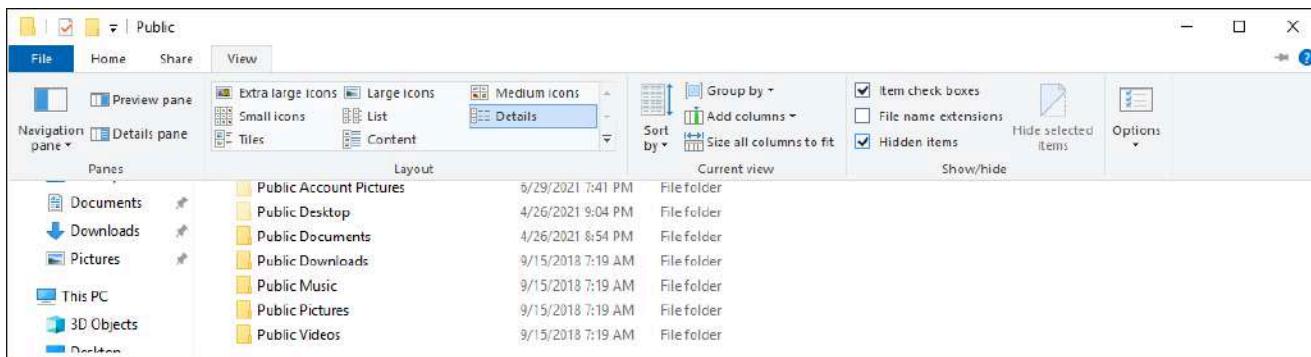


The install process will begin. Let it continue until the **DaVinci Resolve Setup Wizard** appears.

3. Choose **Next** to begin the Setup Wizard.
4. Select **I accept the terms in the License Agreement**, then choose **Next**.
5. Choose your install location, then choose **Next**.
6. Choose **Install** to begin installing.
7. After the install is complete, choose **Finish** to exit the Setup Wizard.
 - a. The install process will continue with the rest of the items.
 - b. Open PowerShell as an administrator and copy and paste the following code to check that icons are available.

```
Copy-Item -Path "$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Blackmagic Design\DaVinci Resolve." -Destination "$env:ALLUSERSPROFILE\Microsoft\Windows\Start Menu\Programs\Blackmagic Design\DaVinci Resolve"
```

- c. Copy the DaVinci Resolve shortcut on the desktop to C:\Users\Public\Public Desktop so it will have an icon on the desktop.
- d. The "Public Desktop" folder must be visible so that you can copy the shortcut. To do this, navigate to Explorer++, choose **View**, then select the **Hidden Items** check box.



8. After the install is finished, choose **Yes** to complete the install and restart your instance.
 - The connection to your instance through NICE DCV will shut down. You can close NICE DCV.
9. If you would like to continue installing software, update your NICE DCV settings, or if you're ready to complete the AMI creation process, re-connect to the instance using the same process from the beginning of the tutorial. Wait a few minutes for the instance to reboot before you try. If you're having trouble reconnecting, consult the **Troubleshoot** section of the [Update Windows workstation AMI](#) tutorial.

Step 4: Update NICE DCV settings

You can adjust some server settings to customize your interaction with DaVinci Resolve. For detailed instructions, see [Managing The NICE DCV Server](#).

- [Enable USB Remotization](#) - NICE DCV enables clients to use some specialized USB devices, such as 3D pointing devices or graphic tablets. The devices are physically connected to their computer to interact with an application running on a NICE DCV server.
- [Configure Multi-Channel Audio](#) - NICE DCV supports up to 7.1 audio channels when using the NICE DCV native clients. The web browser clients support stereo 2.0 audio channels only.

Step 5: Complete the AMI creation process

Now that you have Blackmagic Design DaVinci Resolve installed, return to [Step 4: Connect with NICE DCV](#) in the [Update Windows workstation AMI](#) tutorial. This will help you finish preparing your virtual workstation for AMI creation.

Troubleshooting

Can't connect to my instance after installing DaVinci Resolve.

Your computer might still be in the process of restarting.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Look for the instance that you launched in the [Prerequisites](#) section of this tutorial.
 - a. Review the **Status check** column for your instance. If it's not **2/2 checks passed**, wait a few minutes for the status check to complete.
 - b. After the **Status check** column for your instance is **2/2 checks passed**, reconnect using the process from the beginning of this tutorial.

Related resources

- [Blackmagic Design DaVinci Resolve 17](#)
- [Managing The NICE DCV Server](#)

Set up Perforce Helix Core on Nimble Studio

[Perforce Helix Core](#) is a popular version control software in games development technology.

When you host Perforce on AWS and access the server from Amazon Nimble Studio streaming sessions, you get innate cloud native benefits. Streaming sessions exist in the same Amazon Virtual Private Cloud (Amazon VPC) as the Perforce server. This makes connections to the server, and syncing files from the server, fast and secure. In the AWS network, your files are protected because they don't travel over the public internet.

This guide explains how to set up a Perforce Helix Core server on AWS that is accessible to users within Nimble Studio. This guide also includes setup instructions for [Perforce Helix Swarm](#), which is a code review tool used with Perforce Helix Core.

Contents

- [Prerequisites](#)
- [Step 1: Deploy Perforce server infrastructure to your AWS account](#)
- [Step 2: Create a custom streaming image with the Perforce client](#)

- [Step 3: Create a custom streaming image with the Perforce client](#)
- [Step 4: Launch a streaming session and connect to Perforce](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- You also need the following tools:
 - Install [git](#).
 - Install [Python 3.x](#).
 - [Install](#) and [configure](#) the AWS CLI.
 - Install the [AWS Cloud Development Kit \(AWS CDK\)](#).

Step 1: Deploy Perforce server infrastructure to your AWS account

In this step, you will use the GitHub aws-sample [NimbleStudioPerforceServer](#) to deploy the infrastructure required for hosting Perforce Helix Core in your AWS account.

1. Go to the [NimbleStudioPerforceServer](#) GitHub repository.
2. Follow the instructions in the README file to clone the repository, synthesize the application, and deploy the resources into your account.
 - This needs to be deployed into the same region as your studio.

After successful deployment, continue to the next step.

Step 2: Create a custom streaming image with the Perforce client

In this step, you will retrieve the Perforce Server private record name. This private record name was created during infrastructure deployment from [Step 1: Deploy Perforce server infrastructure to your AWS account](#).

1. Sign in to the AWS Management Console and open the [AWS CloudFormation](#) console.
2. Select the stack named NimbleStudioPerforceServerStack.
3. Select the **Outputs** tab of the stack.

4. Find the key named **P4PrivateRecordName** and notice the value associated with the **P4PrivateRecordName** key. This is the record name for the Perforce Server.

Step 3: Create a custom streaming image with the Perforce client

In this step, you will create a custom streaming image for use with your Nimble Studio. It will have the Perforce clients installed for interacting with the server.

1. Follow the [Update AMIs: Setting up](#) tutorial to launch an Amazon Elastic Compute Cloud (Amazon EC2) instance with a Nimble Studio compatible base AMI for workstations.
 - After you get to [Step 4: Connect with NICE DCV](#) for Windows workstation AMIs or to [Step 4: Connect with NICE DCV](#) for Linux workstation AMIs, continue to the next step in this guide.
2. [Download the Perforce Helix Visual client](#).
 - a. Choose the latest available version for your operating system.
 - b. Download the installer on to your Amazon EC2 instance. You can do this by using the public internet or through use of an Amazon S3 bucket as described in the Nimble Studio documentation.
3. The Perforce client installer recommends installing four individual Helix client applications: Helix Visual Client (P4V), Merge and Diff Tool (P4Merge), Administration Tool (P4Admin), and Command-Line Client (P4).
 - a. This guide assumes that all four clients will be installed.
 - b. You can customize installation based on your needs.
 - c. For example, you can create an Admin streaming image that includes the Administration Tool. You can also create a separate user streaming image that includes the Helix Visual Client and Merge and Diff Tool.
4. For the client configuration, you can input the name of the Perforce Helix Core Server that clients will connect to.
 - a. This is the value of the record name retrieved in *step 5a* of [Step 2: Create a custom streaming image with the Perforce client](#), with an additional prefix and port number suffix.
 - b. The input should resemble the following: `ssl:<perforce-record-name>:1666`

5. Finish installation of the clients. After that, follow the remaining Nimble Studio documentation for creating the encrypted AMI, and updating a launch profile with the new streaming image AMI.

Step 4: Launch a streaming session and connect to Perforce

Follow these steps to launch a streaming session and connect to the Perforce server using the visual client. You will use the new streaming image that you created in [Step 3: Create a custom streaming image with the Perforce client](#).

1. Sign in to the Nimble Studio portal by following the instructions in [Logging in to the Nimble Studio portal](#).
2. Launch a streaming session for the new streaming image created with the appropriate launch profile.
3. After you sign in to the streaming session, open the P4V application.
4. Connect to the server using the default administrator account and password.
 - a. The User is perforce.
 - b. To find the password, go to [Secrets Manager](#). Search for the secret with the description Perforce Helix Core Password. Use the secret value that you retrieve as the password.
5. After opening the connection using the P4V client, enter the value for the secret's plaintext as the password.
6. The P4V client log tab should indicate if Perforce was able to connect to the Helix Swarm server by displaying the message: Connected to Swarm Version 'SWARM/2021.2/2182579 (2021/09/13)'

Connecting to the Perforce Server verifies that the Perforce Server setup is complete. After completion, you can configure the Perforce Server. For example, you can use the [P4Admin](#) tool to create additional users, or to configure Perforce depots.

Set up Incredibuild on Nimble Studio

[Incredibuild](#) is software acceleration technology for builds, tests, and other development processes to run in parallel over a distributed network. When you deploy Incredibuild in your AWS account, it

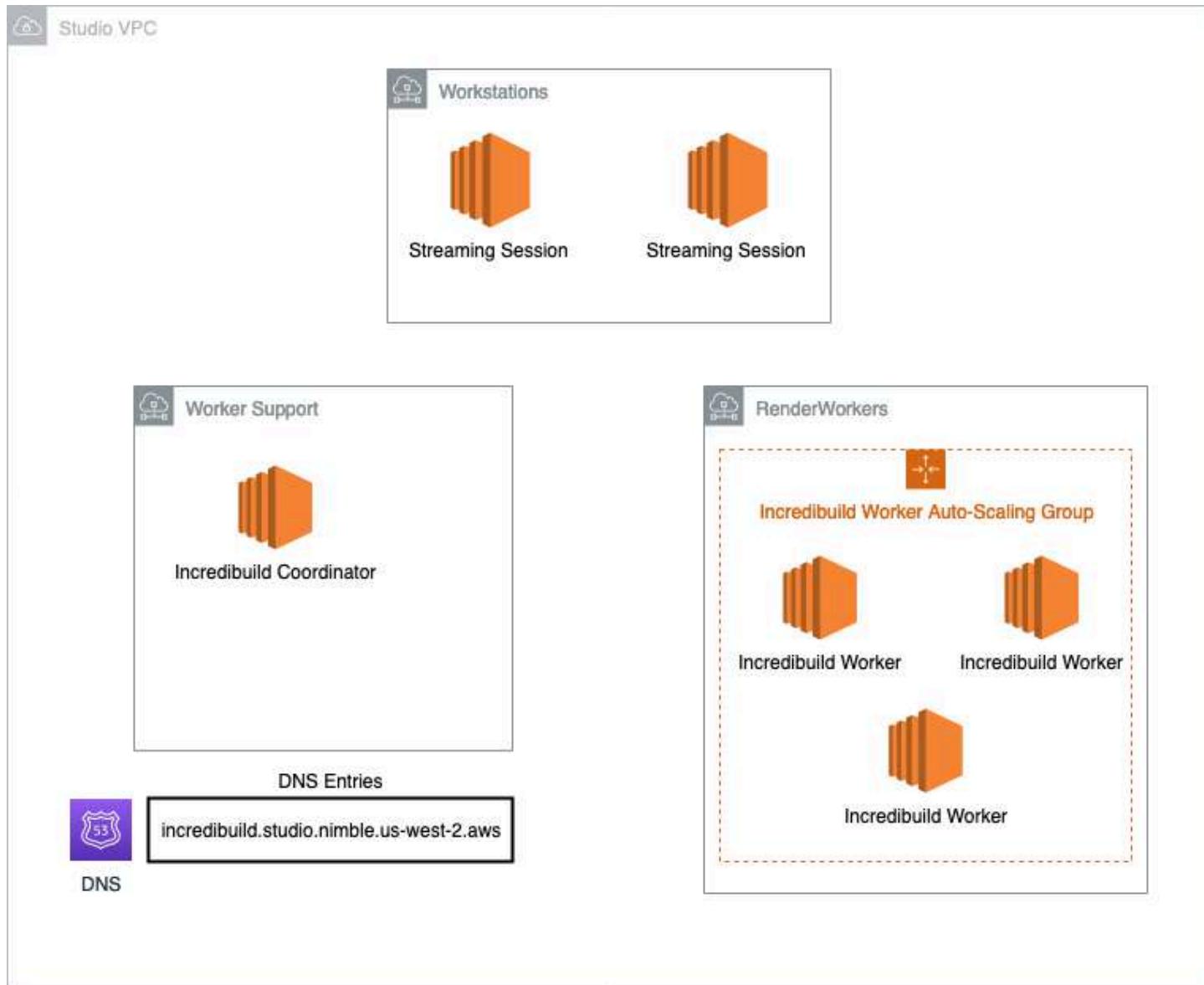
works with the Amazon Virtual Private Cloud (Amazon VPC) distributed network. This tutorial will show you how to set up Incredibuild on Nimble Studio and deploy it to your AWS account.

 **Note**

Incredibuild only supports Windows, so the following information is only for the Windows workstation AMI.

When you configure Incredibuild with your Nimble Studio, the following resources are created:

- A t3.large Windows EC2 instance running the Incredibuild Coordinator.
- A Route 53 record for your Incredibuild Coordinator named `incredibuild.studioId.nimble.region.aws`.
- An Auto Scaling group for increased Incredibuild capacity to accelerate builds further than the allowed number of Nimble Studio workstations.
- Security groups and network ACL configurations for Incredibuild to work with Nimble Studio and Amazon VPC, so that workstations and Incredibuild can connect to each other.
- A studio component that can attach to launch profiles so that Nimble Studio workstations can connect to the Incredibuild Coordinator automatically.



Contents

- [Prerequisites](#)
- [Step 1: Deploy Incredibuild server infrastructure to your AWS account](#)
- [Step 2: Retrieve the Incredibuild private record name](#)
- [Step 3: Create a custom streaming image with the Incredibuild Agent](#)
- [Step 4: Connect Incredibuild to your launch profiles](#)
- [Step 5: Connect Incredibuild to your launch profiles](#)
- [Step 6: Manage your Incredibuild Coordinator](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- You also need the following tools:
 - Install [git](#).
 - Install [Python 3.x](#).
 - [Install](#) and [configure](#) the AWS CLI.
 - Install the [AWS Cloud Development Kit \(AWS CDK\)](#).
 - (Optional) Install the [Session Manager plugin](#).

Step 1: Deploy Incredibuild server infrastructure to your AWS account

In this step, you will use the GitHub aws-sample [NimbleStudioBuildFarm](#) to deploy the infrastructure required for hosting Incredibuild in your AWS account.

1. Go to the [NimbleStudioBuildFarm](#) GitHub repository.
2. Follow the instructions in the README file to clone the repository, synthesize the application, and deploy the resources into your account.
 - This needs to be deployed into the same Region as your studio.

After successful deployment, continue to the next step.

Step 2: Retrieve the Incredibuild private record name

In this step, you will retrieve the Incredibuild private record name created as a part of the infrastructure deployment from [Step 1: Deploy Incredibuild server infrastructure to your AWS account](#).

1. Sign in to the AWS Management Console and open the [AWS CloudFormation](#) console.
2. Select the stack named NimbleStudioBuildFarm.
3. Select the **Outputs** tab of the stack.
4. Find the **Key** named IncredibuildPrivateRecordName.

- Notice the value associated with that key, which is the record name for the Incredibuild server.

Step 3: Create a custom streaming image with the Incredibuild Agent

In this step, you'll create a custom streaming image for use with your Nimble Studio. It will have the Incredibuild clients installed for interacting with the server.

1. Follow the [Update AMIs: Setting up](#) tutorial to launch an Amazon Elastic Compute Cloud (Amazon EC2) instance with a Nimble Studio compatible base AMI for workstations.
 - After you get to [Step 4: Connect with NICE DCV](#) for Windows workstation AMIs, continue to the next step in this guide.
2. Download the [Incredibuild](#) installer. The link should be in the email you received from Incredibuild when you got your license.
 - a. Choose the latest available version for Windows.
 - b. Download the installer.
3. Install the Incredibuild agent.
 - Use the URL that you retrieved from CloudFormation in *step 5a* of [Step 2: Retrieve the Incredibuild private record name](#).
4. Finish installation of the Incredibuild Agent, then follow the remainder of the Nimble Studio docs for creating the encrypted AMI, and updating a launch profile with the new streaming image AMI.

Step 4: Connect Incredibuild to your launch profiles

As part of the deploy, a studio component automatically configures Nimble Studio workstations to connect to the Incredibuild coordinator. The following steps will allow you to use the studio component.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the radio button for the launch profile that you want to change.
4. Choose **Action**. Then choose **Edit**.

5. Select the new streaming image that you created in [Step 3: Create a custom streaming image with the Incredibuild Agent](#).
6. Select the check box next to **Incredibuild** in the **Launch profile components** section.
7. Select **Update launch profile**.

Now any newly launched streaming workstation using this launch profile will be able to work with Incredibuild.

Step 5: Connect Incredibuild to your launch profiles

An Auto Scaling group (which defaults to zero instances) is created when you install Incredibuild. The Auto Scaling group allows you to scale up (or down) the number of Incredibuild agents that are able to accelerate builds in your Nimble Studio.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. At the bottom left, select **Auto Scaling Groups**.
3. Select the Auto Scaling group with the name **NimbleStudioBuildFarm-IncredibuildWorkers**.
4. On the **Details** tab, select **Edit**.
5. Select **Update** and the number of active Incredibuild Agents will increase or decrease to match your **Desired capacity**.
 - a. Specify the number of Incredibuild Agents that you want by modifying **Desired capacity**.
 - b. You can increase the **Maximum capacity** if that number is lower than your **Desired capacity**.
 - c. You can reduce (or keep) the **Desired capacity** at 0 so that the only Incredibuild Agents are the streaming workstations.
6. Select **Update**.

Step 6: Manage your Incredibuild Coordinator

You can view the Incredibuild Coordinator settings from any streaming workstation. However, to change anything, like remove old streaming workstations, connect to the Incredibuild Coordinator.

The Incredibuild Coordinator doesn't have a public IP address, so there are two ways that you can connect. Either use Remote Desktop Protocol (RDP) from a streaming workstation, or use a combination of AWS Session Manager and RDP from your local computer.

Get credentials to sign in to the instance via RDP

In the following steps, you will create a password so that you can connect to the Incredibuild Coordinator. The Incredibuild Coordinator isn't joined to your AWS Managed Microsoft AD domain, so AWS Managed Microsoft AD credentials won't work.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Find the instance named Incredibuild Coordinator, and select its instance ID.
3. Select **Connect** at the top right of this page.
4. Make sure that you are on the **Session Manager** tab, and select **Connect**.
5. Enter **net user Administrator *** and enter a password that you will use to connect through RDP.
6. You'll have to reenter the password again to confirm.

Now you have the credentials to sign in to the instance through RDP. When you use a streaming workstation, you can connect through RDP. The URL is what you retrieved earlier from CloudFormation in [Step 2: Retrieve the Incredibuild private record name](#).

Connect to the instance from your local machine

To connect to the instance directly from your local machine, you'll need the AWS CLI installed, plus you will need the [Session Manager plugin](#) installed. Once done, you can follow these instructions:

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Find the instance named **Incredibuild Coordinator**.
 - Notice its instance ID because you will need it in the next step.
3. Run the following command in a local command prompt: `aws ssm start-session --document-name AWS-StartPortForwardingSession --parameters "localPortNumber=55678,portNumber=3389" --target instance-id`
 - The output should look similar to:

```
Starting session with SessionId: session-id
Port 55678 opened for sessionId session-id
Waiting for connections...
```

Keep that process running in the background. Now you can use your local RDP client and connect to localhost:55678. The Session Manager plugin will route all traffic between your local port 55678 and the RDP port 3389 on the Incredibuild Coordinator EC2 instance. In both cases, you can sign in to your Incredibuild Coordinator instance as Administrator using the password that you entered earlier. After that, you can load Incredibuild Coordinator.

Working with license servers

The tutorials in this section teach you how to create and set up license servers. License servers are used to host licenses that the artists on your team will use to do their work within Amazon Nimble Studio. They allow you to set up environment variables that are necessary for the applications your artists are running to find license files.

Topics

- [Creating a license server](#)
- [Setting up a Nuke license server](#)
- [Setting up an RLM license server](#)

Creating a license server

This tutorial shows how to create a license server and configure a license service so that artists on your team can access license files within Amazon Nimble Studio. A license server allows you to set up environment variables that are necessary for the applications your artists are running to find license files. When you have created the license server, you can install licenses on it and add it to Nimble Studio as a studio component. Adding it as a studio component allows virtual workstations and render workers launched in your studio to communicate to the license server.

Contents

- [Prerequisites](#)
- [Step 1: Upload an installer to an S3 bucket](#)
- [Step 2: Create an instance role for the license server](#)
- [Step 3: Launch an instance](#)
- [Step 4: Set up an Elastic Network Interface \(ENI\)](#)
- [Step 5: Connect to your license server instance](#)

- [Step 6: Add a license server component to your studio](#)
- [Step 7: Add a license server component to your launch profiles](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.

Step 1: Upload an installer to an S3 bucket

Note

Complete [Step 1: Upload an installer to an S3 bucket](#) and [Step 2: Create an instance role for the license server](#) to create the required AWS Identity and Access Management (IAM) role `Nimble_Studio_LicenseServer` and restrict it to one Amazon Simple Storage Service (Amazon S3) bucket.

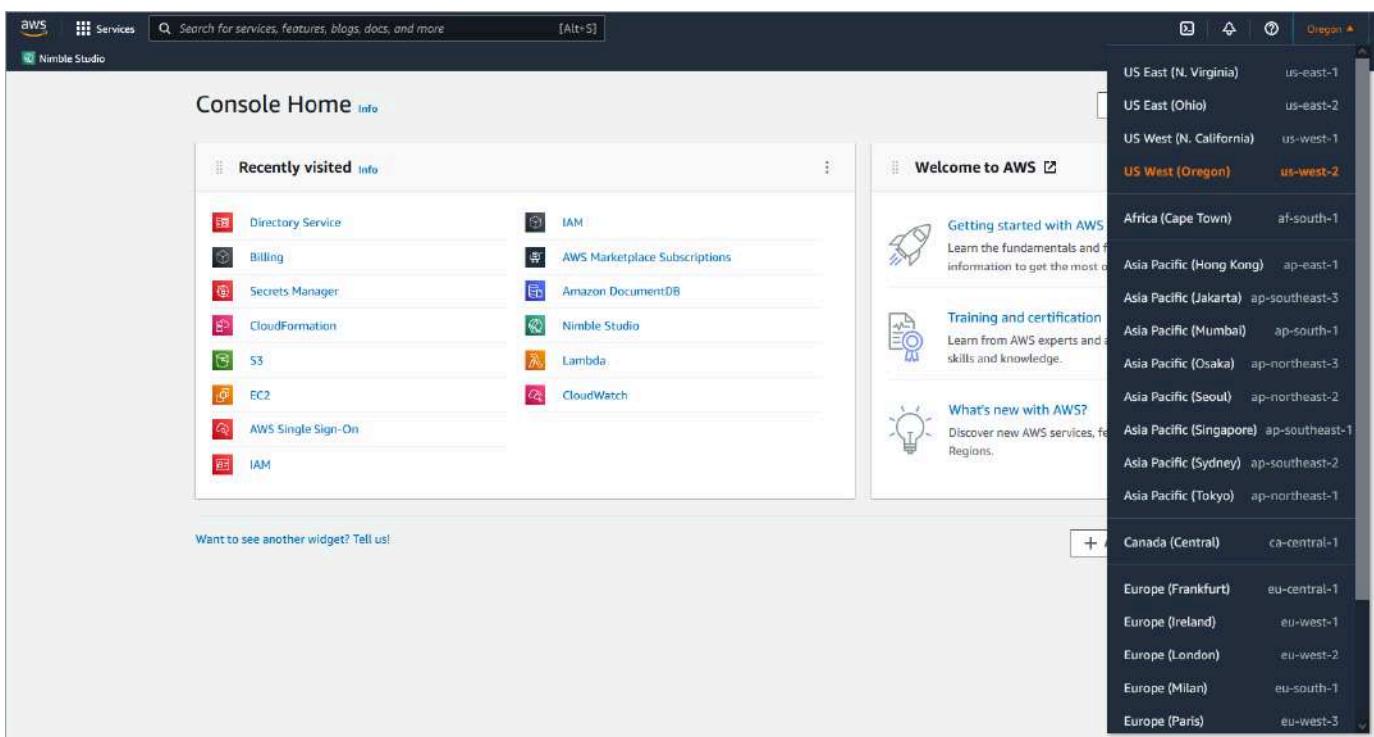
Before you set up an Amazon Elastic Compute Cloud (Amazon EC2) instance as a license server, create an Amazon Simple Storage Service (Amazon S3) bucket to store your installers and license files. After that, you can connect to your license server instance, copy the files from Amazon S3, and install the licenses on the instance. For more information about Amazon S3, see the [Amazon S3](#) overview.

Upload installers and license files to an Amazon S3 bucket.

The storage containers used by Amazon S3 are called *buckets*. Buckets are similar to folders or directories. In the following steps, you will create an Amazon S3 bucket to store your installers and license files.

1. Sign in to the AWS Management Console and open the [Amazon S3](#) console.
2. Choose **Create bucket**.
3. For **Bucket name**, enter a name that makes sense to you. For example, `studiolicensefiles`.
 - To prevent an error, don't use uppercase letters for your bucket name.

4. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



5. Configure the bucket settings so that they're blocking public access.
6. Choose **Create bucket**.
7. Upload installers and license files to your bucket.

Step 2: Create an instance role for the license server

Note

This step is required to create the IAM instance role, `Nimble_Studio_LicenseServer`, and restrict it to the Amazon Simple Storage Service (Amazon S3) bucket from [Step 1: Upload an installer to an S3 bucket](#). You can only attach one IAM role to an instance, but you can attach the same role to many instances.

In this step, you will create the IAM role that grants permissions to download installers and license files from an Amazon S3 bucket onto the license server. This role also allows you to connect to the license server using Session Manager.

The IAM role that you create for Amazon EC2 grants permissions to applications running on instances that need to use your Amazon S3 bucket. It also allows you to connect to your instance using session manager.

To grant Amazon EC2 role read-only access to one Amazon S3 bucket

Follow the instructions in the [Creating policies on the JSON tab](#) tutorial in the IAM User Guide while using the following information.

1. Choose **Create policy**.
2. Replace the JSON data in the text field with the following JSON text.
 - Change the two example lines in the following code DOC-EXAMPLE-BUCKET to the name of the bucket that you created in [Step 1: Upload an installer to an S3 bucket](#) .

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListBucket"  
            ],  
            "Resource": "arn:aws:s3:::<BUCKET-NAME>"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3GetObject"  
            ],  
            "Resource": "arn:aws:s3:::<BUCKET-NAME>/*"  
        }  
    ]  
}
```

3. Name your new policy **licenseServerS3BucketAccess**.

After you've created your policy, follow the instructions in the [Creating a role for an AWS service \(console\)](#) tutorial in the **IAM User Guide** while using the following information.

1. For the service that you want to allow to assume this role, select **EC2**.

2. Search for the policy that you just created and select the check box next to the **licenseServerS3BucketAccess** policy.
3. Search for **SSM** and select the check box next to the **AmazonSSMManagedInstanceCore** policy.
4. (Optional) Enter **Studio** as the key and <your-studio-name> as the value.
5. For **Role name**, enter **Nimble_Studio_LicenseServer**.
6. Enter a description.
7. Review your choices and choose **Create role**.

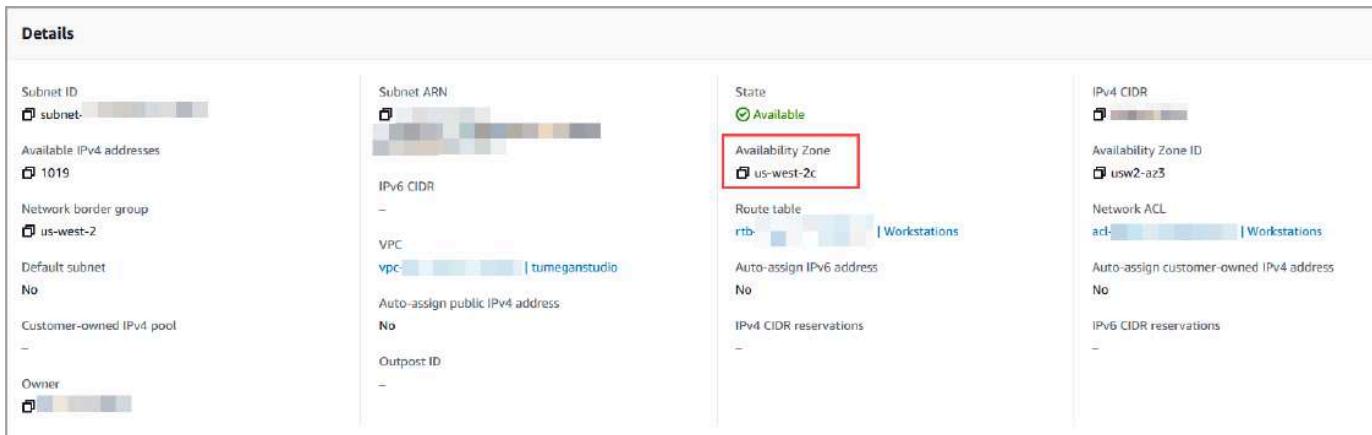
Step 3: Launch an instance

Important

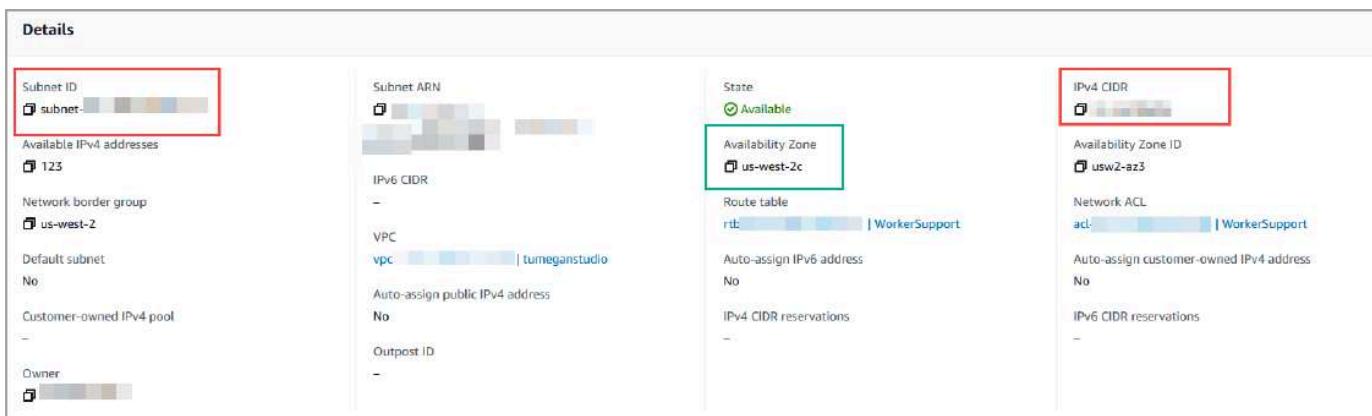
Before you can begin this section, complete [Step 1: Upload an installer to an S3 bucket](#) and [Step 2: Create an instance role for the license server](#), so that you have the IAM instance role, Nimble_Studio_LicenseServer, and the S3 bucket for it. This role is necessary to set the instance configuration in this section.

Follow these instructions to launch an Amazon Elastic Compute Cloud (Amazon EC2) instance that will act as your license server.

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Subnets** in the left navigation pane.
3. Choose the subnet named **Workstations**.
4. In the **Details** section, notice the **Availability Zone**. This information will be used to find the **WorkerSupport** subnet ID.



5. Select **Subnets** in the left navigation pane again.
6. There are two subnets named **WorkerSupport**. Do the following for both subnets.
 - a. Select the subnet.
 - b. In the **Details** section, find the **Availability Zone**.
7. If the **Availability Zone** matches the **Workstations** subnet's **Availability Zone**, notice the **Subnet ID**. This subnet will be used to configure the instance details in step 15b.



8. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
9. Choose **Instances** in the left navigation pane.
10. Choose **Launch instances**.
11. For **Name and tags**, enter <your-studio-name>_licenseServer.
12. (Optional) Choose **Add additional tags**. Enter **Studio** as the key and <your-studio-name> as the value.
13. In **Application and OS Images (Amazon Machine Image)**, choose **Amazon Linux**. Then choose **Amazon Linux 2 AMI (HVM), SSD Volume Type** from the **Amazon Machine Image (AMI)** dropdown.

- Leave 64-bit (x86) as the selected option for **Architecture**.

The screenshot shows the 'Application and OS Images' section on the left and the 'Summary' panel on the right.

Application and OS Images (Amazon Machine Image) Info:

- An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Summary:

- Number of instances: 1
- Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more
ami-03eddedff12e34e59e
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB
- Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Launch Instance button is visible at the bottom right of the summary panel.

14. For **Instance Type**, select **t3.medium** from the list.
15. For **Key pair (login)** choose **Proceed without a key pair** from the first dropdown.

Key pair (login) Info:

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required:

Proceed without a key pair (Not recommended) Default value ▾ Create new key pair

- A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance. You will use EC2 Instance Connect so you don't need a key pair.

16. On **Network Settings**, choose **Edit**.

17. Provide the following information in the specified fields.

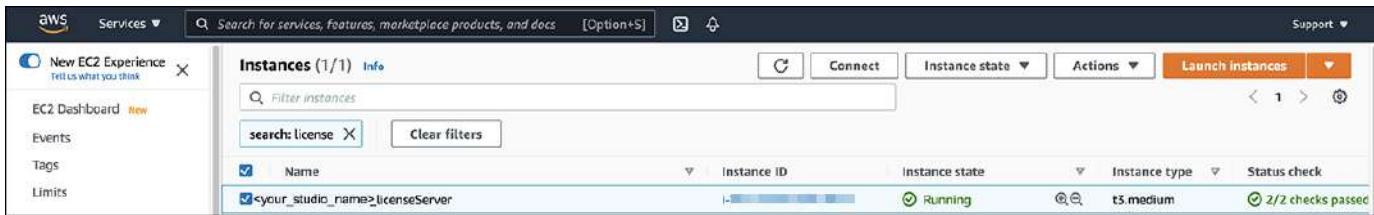
- a. **VPC:** Choose your Studio's VPC.
 - This is named <your-studio-name>
- b. **Subnet:** WorkerSupport.
 - Select the WorkerSupport subnet that you found in step 9.
- c. **Firewall (security groups):** Choose **Create security group**
- d. **Common security groups:** Choose <your-studio-name>Network-LicenseServers

18. In **Configure storage**, increase the size of the root volume.

- The amount of storage you need depends on how much storage you will need for all of your installers and license files. For example, increase the size to 100 GB.

19. In **Advanced details**, choose **Nimble_Studio_LicenseServer** for **IAM instance profile**.

20. Choose **Launch Instance**.



- A warning message might pop up that says you can't connect to your instance because port 22 isn't open. Ignore this message. You will be connecting to the instance using Session Manager and NICE DCV, neither of which require port 22 to be open.

Step 4: Set up an Elastic Network Interface (ENI)

Some vendors associate MAC addresses with license files that they produce (for example: [RLM](#) and [FLEXLM](#)). In these cases, generate an elastic network interface. With a network interface, you can attach/detach the interface from any instance and retain the same MAC address. This will prevent you from having to resubmit requests for a new license file if the license server instance needs to be terminated and a new one needs to be relaunched.

Note

Don't assign a public IP to the ENI that you create.

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Network Interfaces** in the left navigation pane in **Network & Security**.
3. Choose **Create network interface**.
4. Enter a description for your network interface.
 - For example: **License server elastic network interface**.
5. Search for the subnet called **WorkerSupport** and select it.
6. **Auto-assign** a Private IPv4 address.
7. In the security group section, search for **License**.
8. Choose the **<your-studio-name>Network-LicenseServers** security group.
9. Choose **Add new tag**.
 - a. Enter **Name** as the key and **LicenseServerNetworkInterface** as the value.
 - b. (Optional) Add **Studio** as the key and **<your-studio-name>** as the value.
10. Choose **Create network interface**.
11. Select the new network interface: Select the box next to the name that you previously created as the tag value.
12. Choose **Actions** and then choose **Attach**.
13. Choose your license server instance.
14. Choose **Attach**.
15. Navigate to the **Details** tab.
16. Find the **Private IPv4 address**.
 - Write this down because you will need it later.
17. In the Private IPv4 address you will find the **MAC address**.
 - You might need to key your licenses to the MAC address.

Step 5: Connect to your license server instance

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select the instance called **<your-studio-name>_licenseServer**.
4. Choose **Connect**.

5. Select **Session Manager** and then choose **Connect**.
6. A new tab will open and you'll be connected to your license server instance.

Copy files from Amazon S3 to license server instance.

1. Make a temporary folder in /tmp for your downloaded license files.

```
sudo mkdir /tmp/license_files
```

```
sudo chmod a+w /tmp/license_files
```

2. To pull your license file or installer from S3, you will need to find the S3 URI for that file:
 - a. Sign in to the AWS Management Console and open the [Amazon S3](#) console.
 - b. Select **Buckets** in the left navigation pane to view a list of buckets.
 - c. Navigate to the installer or license file that you want to pull onto your worker.
 - d. Choose the file's Name.
 - e. Choose the copy icon next to the **S3 URI** to the right of the **Object overview**.
3. Return to the **Session Manager** for your instance to download the license file.
4. To run the following command, replace **S3 URI** with your own URI that you just copied from your S3 bucket.

```
cd /tmp/license_files
```

```
aws s3 cp S3 URI filename
```

5. Install the license server on the instance according to the instructions provided by the independent software vendor.

Step 6: Add a license server component to your studio

Now that you've installed the license server on the instance, you can add it to your studio as a license service component.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. Choose **Add** in the **License service** studio resource type.
4. In the **License service info** section, complete the fields as follows:

- a. Leave the Region as default.
 - This should already be set to the Region that your studio is deployed in.
- b. Choose a name for your license server.
 - This might be a generic name, or include a description for the type of license hosted on it. Example: <your-studio-name>_licenseServer
- c. (Optional) Give your license server a description. Example: License Server hosting floating Houdini licenses

The screenshot shows the 'License service info' configuration page. It includes the following fields:

- Region Info:** A dropdown menu showing 'us-west-2'.
- License server name:** An input field containing '<your_studio_name>_licenseServer'. Below it is a note: 'The name must be 3 to 128 characters.'
- License server description - optional:** A text area containing 'License Server hosting floating Houdini licenses'. Below it is a note: 'The description can be 3 to 256 characters.'

5. In the **License service configuration** section, enter the **Endpoint** for your license service.
 - Set this to the Private IPv4 IP address setup by the network interface you created earlier.
6. In the **Initialization scripts** section, add commands and set environment variables that are necessary for the virtual workstations in your studio to find license files.
 - a. For Windows:
 - i. Enter the commands as a **system** initialization script.
 - ii. Enter PowerShell commands to configure your environment variables during initialization. For example: `setx /M PROJ Z:/project`
 - iii. You can also enter commands that might be necessary for your licensed applications to connect to the license server. For example, for Houdini: `hserver -S <private_IP_address>`
 - b. For Linux:

- i. Enter the commands as a **system** initialization script.
- ii. Enter bash commands to configure your environment variables during initialization.
For example:

```
cat > /etc/profile.d/license-service-$studioComponentId.sh
<<ENDOFSCRIPT

# – Set Environment Variables

export PROJ=/mnt/fsxshare/project

ENDOFSCRIPT
```

- iii. You can also enter commands that might be necessary for your licensed applications to connect to the license server. For example, for Houdini:

```
# – Run Additional System Commands

# - Specify License server for Houdini and connect to it

/opt/hfs<version_number>/bin/hserver -h <private_IP_address>

# Need to wait 30 seconds before connecting to license server or
it will fail

/usr/bin/sleep 30

/opt/hfs<version_number>/bin/hserver -S <private_IP_address>
```

Initialization scripts

Use the variables \$linuxMountPoint, \$windowsMountDrive, and \$shareName to access the file storage configuration values entered below.

Windows

Windows system initialization script [Info](#)

Enter PowerShell commands to run during Windows system initialization.

```
1 setx /M RLM_LICENSE 5053@  
2 hserver -S [REDACTED]
```

Windows user initialization script [Info](#)

Enter PowerShell commands to run during Windows user initialization.

```
1 # N/A
```

Linux

Linux system initialization script [Info](#)

Enter shell commands to run during Linux system initialization.

```
1 cat > /etc/profile.d/license-service-$studioComponentId.sh <<ENDOFSRIPT  
2  
3 # – Set Environment Variables  
4  
5 export RLM_LICENSE=[REDACTED]  
6  
7 ENDOFSRIPT  
8  
9 # – Run Additional System Commands  
10  
11 # – Specify License server for Houdini and connect to it  
12 /opt/hfs18.5/bin/hserver -h [REDACTED]  
13  
14 # Need to wait 30 seconds before connecting to license server or it will fail  
15 /usr/bin/sleep 30  
16 /opt/hfs18.5/bin/hserver -S [REDACTED]
```

7. Choose the **LicenseServers** security group.
8. (Optional) Add tags if you're using tags to track your AWS resources.
9. Read the terms and conditions and if you agree:
 - Select the check box next to **I understand that Nimble Studio will access my existing license server.**
10. Choose **Save connection parameters**.

Tags - optional [Info](#)

Tags help you identify, search, filter, and track resources. You can add or remove tags anytime.

Key	Value - optional	Remove
<input type="text" value="Studio"/> X	<input type="text" value="<your_studio_name>"/> X	Remove

[Add new tag](#)

You can add up to 49 more tags.

I understand that Nimble Studio will access my existing license server. [Info](#)

[Cancel](#) [Save connection parameters](#)

Step 7: Add a license server component to your launch profiles

These steps explain how to attach the license service to an existing launch profile. To create a new launch profile, navigate to [Creating launch profiles](#) and follow the steps in that tutorial before returning to this step.

1. Return to the Studio Manager.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the dot to the left (beginning) of the launch profile name.
4. Choose **Action**. Then choose **Edit**.
5. Navigate to **Launch profile components**.
6. Select the check box next to the **Studio license server component** that you just created.
7. Choose **Update launch profile**.
8. Repeat these steps for all launch profiles that you want to have access to the license server. For your render workers to have access to the license server, edit the default launch profile to include the license server component.

Now that you've completed the steps for creating a license server, you can install licenses on it and add it to Nimble Studio as a studio component.

Setting up a Nuke license server

This administrator tutorial will show you how to set up a Nuke license server on your studio in Amazon Nimble Studio.

[Nuke](#) is a node-based digital compositing and visual effects application that is used for television and film post-production. You can use the Foundry Licensing Utility with Nuke to install and view licenses, install server tools, and help troubleshoot licensing problems.

Contents

- [Prerequisites](#)
- [Step 1: Launch an instance](#)
- [Step 2: Connect with DCV](#)
- [Step 3: Install the Foundry Licensing Utility](#)

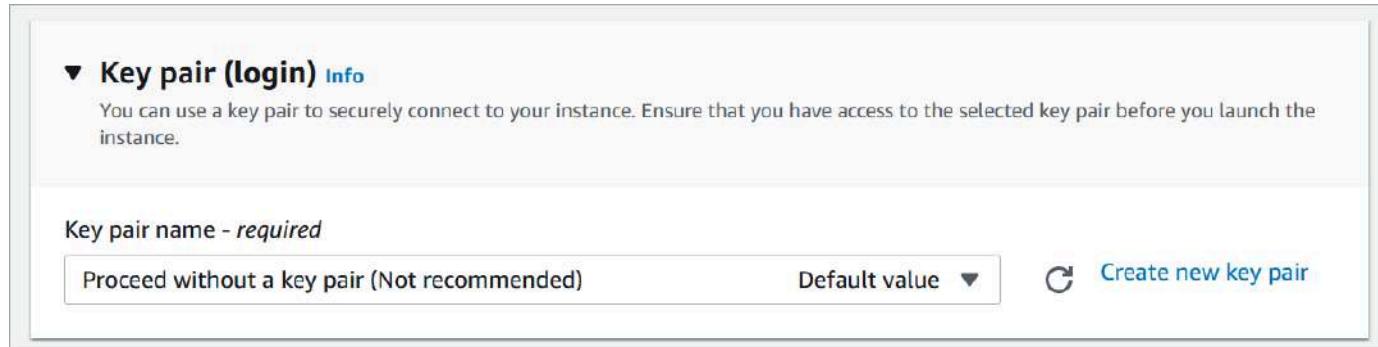
Prerequisites

- To complete this tutorial, create a license server IAM role and an Amazon S3 bucket for license files by following the instructions in [Step 1: Launch an instance](#) and [Step 2: Connect with DCV](#).

Step 1: Launch an instance

Follow the instructions in the [Launch an instance using defined parameters](#) tutorial in the *Amazon EC2 User Guide for Windows Instances* while using the following information.

1. For **Name and tags**, give the instance a name so that you can easily find it later, such as <your-studio-name>_licenseServer.
2. (Optional) Enter **Studio** as the key and <your-studio-name> as the value.
3. Search for NICE DCV for Amazon Linux 2 in **Application and OS Images (Amazon Machine Image)**.
4. Choose the **Select** next to **NICE DCV for Amazon Linux 2**.
5. For the **Instance type**, choose **t3.medium**.
6. For **Key pair (login)** choose **Proceed without a key pair** from the first dropdown.



- A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance. You will use EC2 Instance Connect so you don't need a key pair.

7. On **Network settings**, choose **Edit**.
8. For **Network**, choose your Studio's VPC.
 - This is named <your-studio-name>
9. For **Subnet**, choose **WorkerSupport**.
 - Select the **WorkerSupport** subnet that you found in step 9.
10. Choose **Select an existing security group** for **Firewall (security groups)**.
11. Choose the security group with a name beginning with <your-studio-name>Network-LicenseServers.
12. Depending on how much storage you will need for all of your installers and license files, increase the size of the root volume in the **Configure storage** section.
 - For example, increase the size to 100 GB.
13. In **Advanced details**, select <your-studio-name>_Studio_LicenseServer
14. Choose **Launch instance**.
 - A warning message might pop up that says you can't connect to your instance because port 22 isn't open. Ignore this message. You will be connecting to the instance using Session Manager and NICE DCV, neither of which require port 22 to be open.
15. On the **Launch Status** page, choose **View instances**.

Step 2: Connect with DCV

To connect to your Linux instance with a GUI, you'll need to modify the DCV setup on your instance to allow you to connect with the DCV client. Connecting to your Linux instance with a GUI makes it easier to install and use the Foundry License Utility.

Set up DCV session

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select the license server.
4. Choose **Connect**.
5. Select **Session Manager** and then choose **Connect**.
6. A new tab will open and you'll be connected to your license server instance.
7. Run the following command to restart the DCV server: `sudo systemctl restart dcvserver`
8. Run the following command to set the password for the CentOS user: `sudo passwd root`
 - Remember your password because you will be using it in the **Connect with DCV** section.
9. Run the following command to create a DCV session that you can connect to as root: `sudo dcv create-session --type=virtual --owner root --user root virt`
10. Close the Session Manager browser tab.

Install the Session Manager plugin on your local computer

1. For instructions about installing Session Manager, see [Install the Session Manager plugin for the AWS CLI](#).
 - Install the correct version for your local operating system (OS).
2. After the plugin is installed, run the following command in Terminal or in PowerShell: `aws configure`
3. Enter your credentials. Select the AWS Region that your studio is deployed in..
4. Run the following command that matches your local machine's operating system to update the instance-id with your Nuke license server instance ID.

Linux & macOS

```
aws ssm start-session //  
--target instance-id //  
--document-name AWS-StartPortForwardingSession //  
--parameters '{"portNumber":["8443"], "localPortNumber":["8443"]}' //
```

Windows

```
aws ssm start-session ^  
--target instance-id ^  
--document-name AWS-StartPortForwardingSession ^  
--parameters portNumber="8443",localPortNumber="8443"
```

Connect with DCV

1. Run the **DCV client** on your local machine.
2. Enter **localhost** in the **Hostname/IP Address** field.
3. In the window that says “Your connection is not secure,” choose **Trust** (or **Proceed** for MacOS) to trust the connection and to continue.
 - By default, DCV generates a self-signed certificate that it uses to secure traffic between the DCV client on your local computer and the server on your workstation. If you prefer to use your own certificate, choose **Go Back**, and see the instructions in [Changing the TLS Certificate - NICE DCV](#).
4. For **Username**, enter **root** and then enter the password that you set in the **Setup DCV session** section.
5. Choose **Login**.

Connect with DCV later

The next time that you want to work on the license server instance, follow these steps before connecting to DCV.

1. Run the following command that matches your local machine’s operating system.

Linux & macOS

```
aws ssm start-session //  
--target instance-id  
--document-name AWS-StartPortForwardingSession //  
--parameters '{"portNumber":["8443"], "localPortNumber":["8443"]}' //
```

Windows

```
aws ssm start-session ^  
--target instance-id ^  
--document-name AWS-StartPortForwardingSession ^  
--parameters portNumber="8443",localPortNumber="8443"
```

2. (Optional) If you need to start a new DCV session, connect to the license server instance with Session Manager and run the following command: `sudo dcv create-session --type=virtual --owner root --user root virt`

Troubleshooting

To close an existing running session, run the following command: `sudo dcv close-session virt`

To list running sessions, run the following command: `sudo dcv list-sessions`

Step 3: Install the Foundry Licensing Utility

Note

Request the **VM Enable** license file, which is a license for a Virtual Machine (VM).

Upload the installer to S3

1. Download the installer for [Foundry Licensing Utility \(FLU\) 8.1.3](#)
 - As of June 30, 2021 the FLU version 8.1.6 for Linux doesn't work.
2. Upload the installer to the S3 bucket that you created in the **Prerequisites** section of this tutorial.

3. Notice the S3 URI for the file. You will need this in *step 3* of the next section.

Copy files from Amazon S3 to license server instance

1. In DCV, open a terminal window.
2. Make a temporary folder in /tmp for your downloaded license files by running the following commands:

```
sudo mkdir /tmp/license_files  
sudo chmod a+w /tmp/license_files
```

3. Pull the Red Hat Package Manager File (RPM) file onto your license server instance by running the following commands.
 - Replace <S3_URI> with the S3 URI that you found in *step 3* of the previous section.
4. Run the following command to install the Foundry Licensing Utility: sudo yum localinstall <downloaded file>
 - The FLU gets installed in /opt/FoundryLicensingUtility by default.
5. Open a file browser and select **+ Other Locations**.
6. Go to **Computer**. Then choose **opt**, and **FoundryLicensingUtility**.
7. Select **foundry-licensing-utility**.
8. The Foundry Licensing Utility application will open.

Redeem your licenses

1. Open the FLU and select **System ID**.
2. Notice the system ID and use your Activation Key to redeem your license key. You will need this in *step 5* of this section.
3. In the left navigation pane, expand **Licenses**.
4. Choose **Install**.
5. Enter the license key into the text field.

6. Follow the instructions to install the license server.

Use your Nuke license

1. Complete [Step 6: Add a license server component to your studio](#) and [Step 7: Add a license server component to your launch profiles](#).
2. If you're using a Windows machine, open the C:\Windows\System32\drivers\etc\hosts file and add the following line: private_IP_address my_machine_name
3. In the license service component, set up the Foundry license environment variable in the system initialization scripts by running the following command that matches your local machine's operating system.

Linux

```
cat > /etc/profile.d/license-service-$studioComponentId.sh <<ENDOFSRIPT  
  
# – Set Environment Variables  
export foundry_LICENSE=4101@private_IP_address  
  
ENDOFSRIPT
```

Windows

```
setx /M foundry_LICENSE 4101@private_IP_address
```

Troubleshooting the Foundry Licensing Utility

1. Reconnect to your license server instance.
2. Open a file browser and select **+ Other Locations**.
3. Go to **Computer**. Then choose **opt**, and **FoundryLicensingUtility**.
4. Double-click on **foundry-licensing-utility**.
5. The Foundry Licensing Utility application will open.
6. Go to **License Server**.
7. Select **Control**.
8. Select **Start**.

Setting up an RLM license server

The Reprise License Manager (RLM) is a flexible license manager with the power to serve enterprise users. There are several industry standard DCCs that use RLM for licensing, such as Yeti and SpeedTree.

These instructions are for setting up an RLM license server.

Contents

- [Prerequisites](#)
- [Step 1: Install RLM](#)
- [Step 2: Start the license server](#)
- [Step 3: Set up the license service component](#)
- [Troubleshooting](#)

Prerequisites

Before you begin, create a license server IAM role and an Amazon S3 bucket for license files by following the instructions in **Step 1** through **Step 4** of [Creating a license server](#).

Step 1: Install RLM

First, upload the .tar file to the Amazon S3 bucket that you set up for license files in the Creating a license server tutorial.

Upload the installer to S3

1. Go to the [RLM License Administration Bundle Download](#) page.
2. Read the license agreement and if you agree, choose **I Agree**.

License Administrator License Agreement

RLM License Administration Bundle Download

Please Read the Software Licensing Agreement, scroll to the bottom, and click "I AGREE" if you agree to the terms.

REPRISE LICENSE ADMINISTRATOR LICENSE AGREEMENT FOR RLM LICENSE ADMINISTRATION BUNDLE

PLEASE READ THE FOLLOWING TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD NOT DOWNLOAD THE SOFTWARE. LOADING OF THE SOFTWARE ONTO A COMPUTER INDICATES YOUR ACCEPTANCE OF THE FOLLOWING TERMS AND CONDITIONS:

LICENSE: Reprise Software, Inc. ("Reprise") hereby grants to you a nonexclusive, nontransferable license to use the RLM End-User Bundle software (the "Software") and related documentation for your own internal purposes at the location to which the Software is downloaded. You may copy the Software for backup and archival purposes. Each copy of the Software made hereunder must include all copyright, trademark, and restricted rights notices. Reprise (or its licensor) retains all right, title, and interest in the Software and documentation (and any copy thereof).

RESTRICTIONS: Reproduction, disclosure, reverse engineering, disassembly, modification, use for any purpose other than internal use, and/or distribution by any means of the Software are prohibited. Any attempt to transfer any of the rights or obligations hereunder is void. You may not rent, lease, loan or resell the Software. No product downloaded from this site may be used in any product which has as its primary purpose license management. Unauthorized copying of the Software or documentation is not permitted.

TERM: This Agreement and each license granted hereunder, will remain in effect unless and until terminated by mutual agreement of the parties or as follows: (a) you may terminate this Agreement or any license at any time, and (b) Reprise will have the right to terminate this Agreement or a particular license hereunder if you fail to perform any obligation under this Agreement. You agree upon termination to cease using the Software, promptly destroy the Software and documentation, and provide Reprise with notice that you have done so.

WARRANTY DISCLAIMER: THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS. REPRISE EXPRESSLY DISCLAIMS ALL WARRANTIES, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. REPRISE DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED, ERROR-FREE, OR VIRUS-FREE.

LIMITATION OF LIABILITY: IN NO EVENT WILL REPRISE BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING LOST PROFITS OR DATA, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE OR ANY DATA SUPPLIED THEREWITH, EVEN IF REPRISE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. In no case will Reprise's liability for damages hereunder exceed fifty dollars (US \$50).

EXPORT LAW ASSURANCES: None of the Software or underlying information or technology may be downloaded or otherwise exported or reexported in violation of any applicable laws or regulations.

FOR USERS IN A U.S. GOVERNMENT AGENCY: The Software and related documentation are provided as Commercial Computer Software or restricted computer software. Use, duplication, or disclosure by the U.S. Government or a U.S. Government subcontractor is subject to the restrictions set forth in 48.C.F.R. Section 12.212 or 48 C.F.R 227.2702, as applicable, or to successor provisions. The manufacturer is Reprise Software, Inc., 1530 Meridian Ave., San Jose, CA 95125.

GENERAL: This Agreement will be governed by the laws of the State of California, USA, except for the body of law dealing with conflicts of law and the United Nations Convention on Contracts for the Sale of Goods. If any provision of this Agreement is held to be unenforceable, that provision will be removed and the remaining provision will remain in full force.

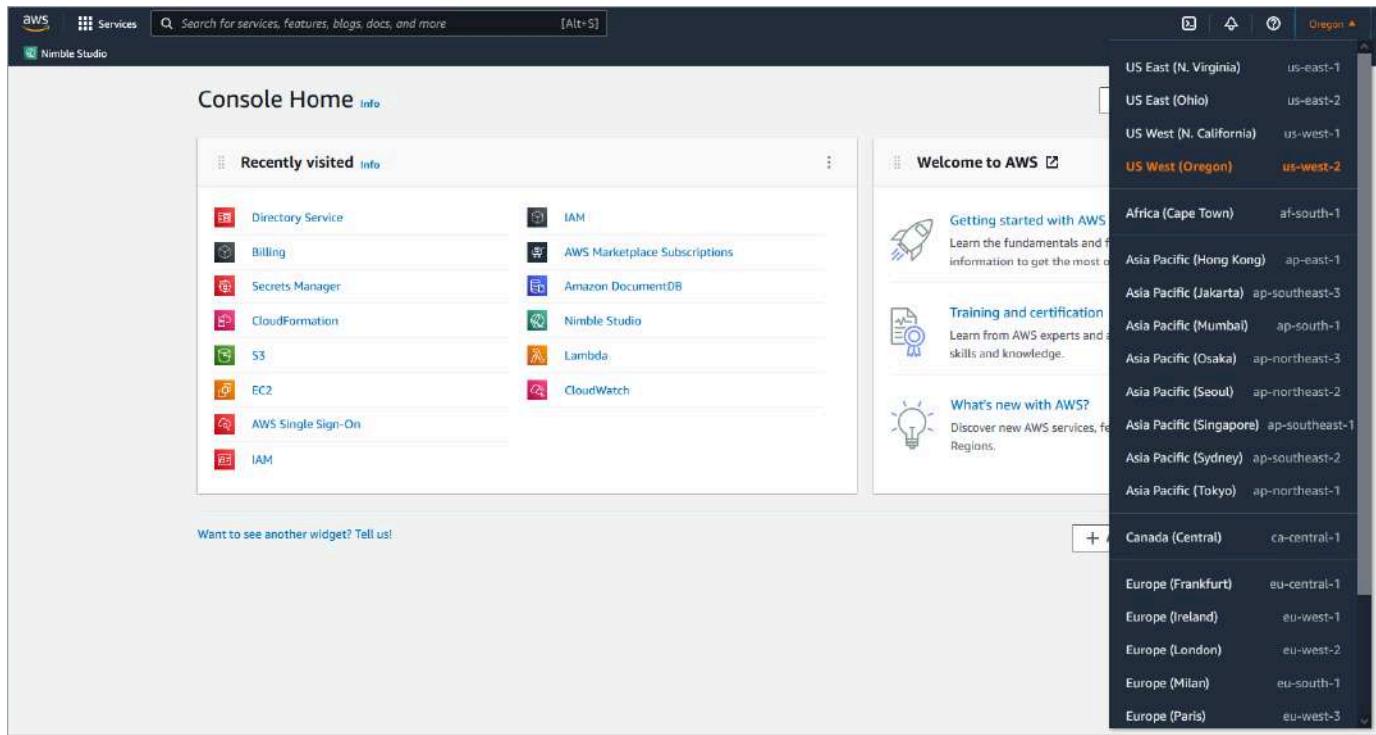
For International Users: This Agreement has been written in the English language. By loading the Software you waive any rights you may have under the law of your country to have this Agreement written in the language of that country.

Should you have any questions concerning this Agreement, you may contact Reprise at support@reprisesoftware.com or via telephone at (408) 907-6756.

I AGREE

I DO NOT AGREE

3. Choose the download link for **Linux x64**.
4. Sign in to the AWS Management Console and open the [Amazon S3 console](#).
5. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



6. Select the bucket that you created in [Step 1: Upload an installer to an S3 bucket](#) of the *Creating a license server* tutorial.

7. Select the **install/** file and choose **Upload**.

8.

The screenshot shows the 'Objects' list in the Amazon S3 console. There are two objects listed: 'install/' and 'LIC/'. The 'install/' object is selected, indicated by a blue border around its row. The 'Actions' dropdown menu is open, and the 'Upload' option is highlighted with a red box. Above the table, there is a search bar labeled 'Find objects by prefix' and several action buttons: 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. The table has columns for Name, Type, Last modified, Size, and Storage class.

9. Choose the installer file that you downloaded in step 3 of this section.

10. Choose **Upload**.

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (1 Total, 5.7 MB)

All files and folders in this table will be uploaded.

[Remove](#)

[Add files](#)

[Add folder](#)

[Find by name](#)

< 1 >

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	x64_l1.admin.tar.gz	-	application/gzip	5.7 MB

Destination

Destination



▶ Destination details

Bucket settings that impact new objects stored in the specified destination.

▶ Permissions

Grant public access and access to other AWS accounts.

▶ Properties

Specify storage class, encryption settings, tags, and more.

[Cancel](#)

11. When the upload succeeds, choose **Close**.

Connect to your instance

1. Sign in to the AWS Management Console and open the [Amazon EC2](#) console.
2. Choose **Instances** in the left navigation pane.
3. Select the instance named `<studio_name>_licenseServer`.
4. Choose **Connect**.

5. Select **Session Manager** and then choose **Connect**.

A new tab will open and you'll be connected to your license server instance.

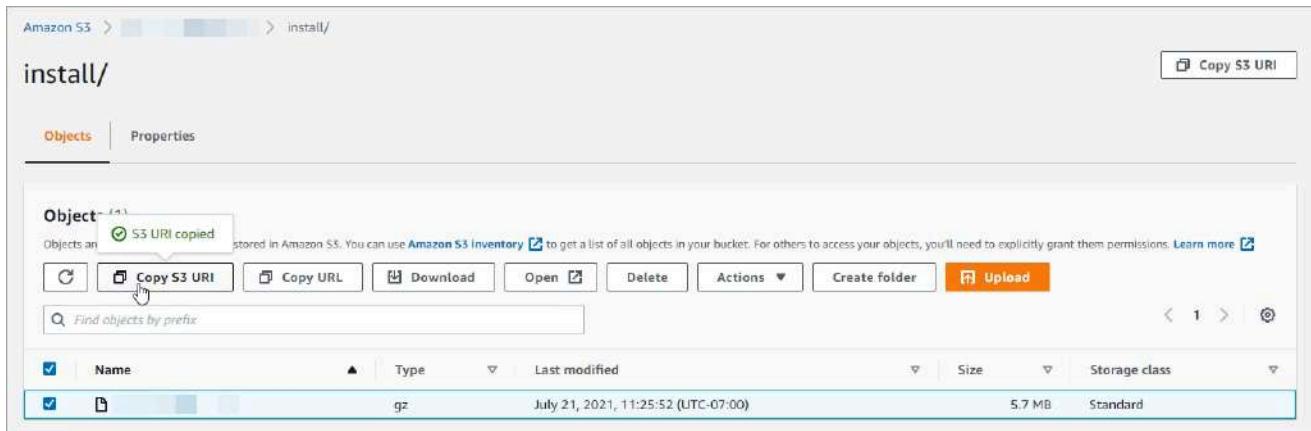
Copy files from Amazon S3 to license server instance

- Run the following commands to make a temporary folder in /tmp for your downloaded license files.

```
sudo mkdir /tmp/license_files  
sudo chmod a+w /tmp/license_files  
cd /tmp/license_files
```

- To pull your license file or installer from S3, find the **S3 URI** for that file.

- Sign in to the AWS Management Console and open the [Amazon S3](#) console.
- Select **Buckets** in the left navigation pane.
- Navigate to the installer or license file that you want to pull on your worker.
- Select the file's **Name**.
- Notice the **S3 URI** to the right of the **Object** overview. You will need this for step 3.



- Run the following command to pull the tar file onto your license server instance: `aws s3 cp <S3_URI> .`
 - Replace <S3_URI> with the **S3 URI** that you found in step 2f of this section.
- Run the following command to untar (extract) the file: `tar xf <PACKAGENAME>.tar.gz`
 - Replace <PACKAGENAME> with the name of the installer or license file that you want to put on your worker.

5. Run the following commands to create new folder and move the directory.

```
cd /opt/
mkdir rlm
mv /tmp/license_files/<PACKAGENAME> /opt/rlm
```

Step 2: Start the license server

Next, start the license server. To do this, add the license files to the `/opt/rlm` folder and set the environment variable.

1. Add license files to the `/opt/rlm` folder by running the following commands to copy the **S3 URI** for each license file and copy it to the license server.
 - Replace `<S3_URI>` with the **S3 URI** that you found in **step 2f** of the previous section.

```
cd /opt/rlm
aws s3 cp <S3_URI> .
```

2. Run the following command to set the RLM environment variable:
`RLM_LICENSE=5053@localhost`
3. Run the following command to start the license server: `./rlm`

Step 3: Set up the license service component

1. Create a license service component in your studio and attach it to your artist's launch profiles by completing **Step 6** and **Step 7** of the [Creating a license server](#) tutorial.
2. Run the following command to set the environment variable in the license service component:
`RLM_LICENSE=5053@<LICENSE_SERVER_PRIVATE_IP_ADDRESS>`

Troubleshooting

I need to end an RLM process and restart the server.

1. Run the following command to see if a RLM process is running: `ps ax | grep rlm`
2. Find the process number.
 - This is the number in the left column followed by a question mark.

3. Run the following command: `kill <PROCESS_NUMBER>`
4. Run the following commands to restart RLM:

```
cd /opt/rlm/x64_11.admin/  
./rlm
```

The license server isn't working.

- Run the following command to verify that the RLM_LICENSE environment variable is set on both the license server and the workstation: `printenv RLM_LICENSE`
 - a. If the output isn't `5053@localhost`, run the following command:
`RLM_LICENSE=5053@localhost`
 - b. Verify that any necessary license files are copied to the location of the executable by running the following commands:

```
cd /opt/rlm  
ls
```

- c. If the license files aren't copied to that location, follow the instructions in the [Copy files from Amazon S3 to license server instance](#) section.

You've now successfully set up an RLM license server.

Rendering with Amazon Nimble Studio

With Amazon Nimble Studio, you can use your studio's render farm to create rendered images on virtual workstations. Certain applications require you to set up a license server before they can be used on virtual workstations and render workers. In addition, some software applications require you to follow specific steps after installation so that they will work properly.

The following topics show how to render in a Nimble Studio cloud studio. You'll learn how to install and configure applications on virtual workstations, and how to set up and maintain a render farm.

Related resources

The Linux and Windows worker AMIs are for the render worker nodes that Deadline spins up.

- [Update a Windows worker AMI](#)
- [Update a Windows worker AMI](#)

Configuring AWSThinkboxDeadline

Deadline is the software that manages the render farm that is created when your cloud studio deploys with StudioBuilder. Before your artists can render using your studio's render farm, configure Deadline according to the render farm choices you made during your cloud studio deploy.

First, you will create a render group for each render fleet that you created with StudioBuilder. After that, you will adjust some settings on your render workers to allow you to connect to them and monitor their progress as they work on render jobs.

If you're using Windows virtual workstations to create content and submit it to a Linux render farm, you will also need to set mapped paths so that your render workers know how to map file paths from the Windows convention of something like Z to the Linux convention of something like /mnt/fsxshare.

Note

If you're only using Linux virtual workstations to submit to your farm, you can skip the path mapping section of this tutorial.

This tutorial will show you how to launch a virtual workstation in your studio, open Deadline, and change the necessary settings.

Contents

- [Prerequisites](#)
- [Step 1: Sign in to Nimble Studio portal as Admin](#)
- [Step 2: Accept the EULA](#)
- [Step 3: Launch a virtual workstation](#)
- [Step 4: Open Deadline and create groups](#)
- [Step 5: Adjust worker settings](#)
- [Step 6: \(Windows only\) Set mapped paths](#)
- [Related resources](#)

Estimated time: 1 hour

Prerequisites

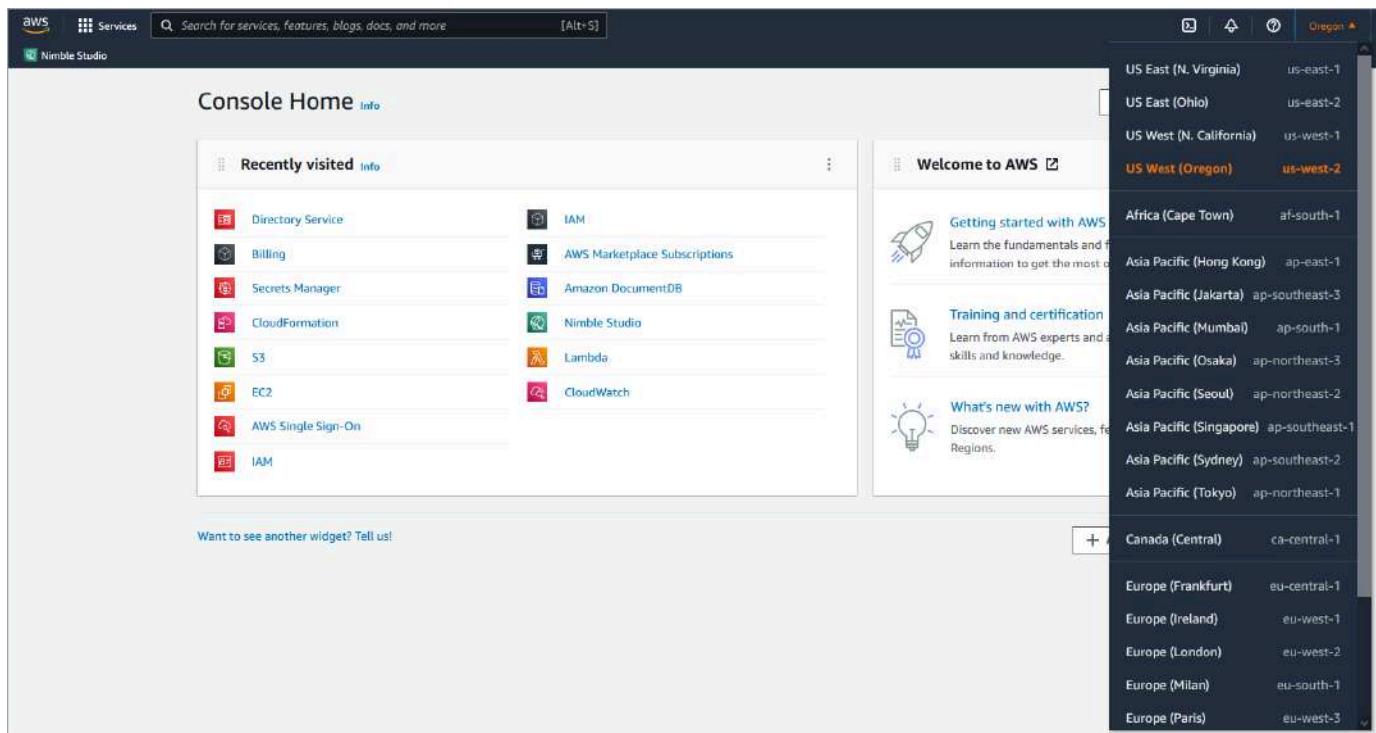
- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- You need the administrator password for your studio's AWS Managed Microsoft AD.

Step 1: Sign in to Nimble Studio portal as Admin

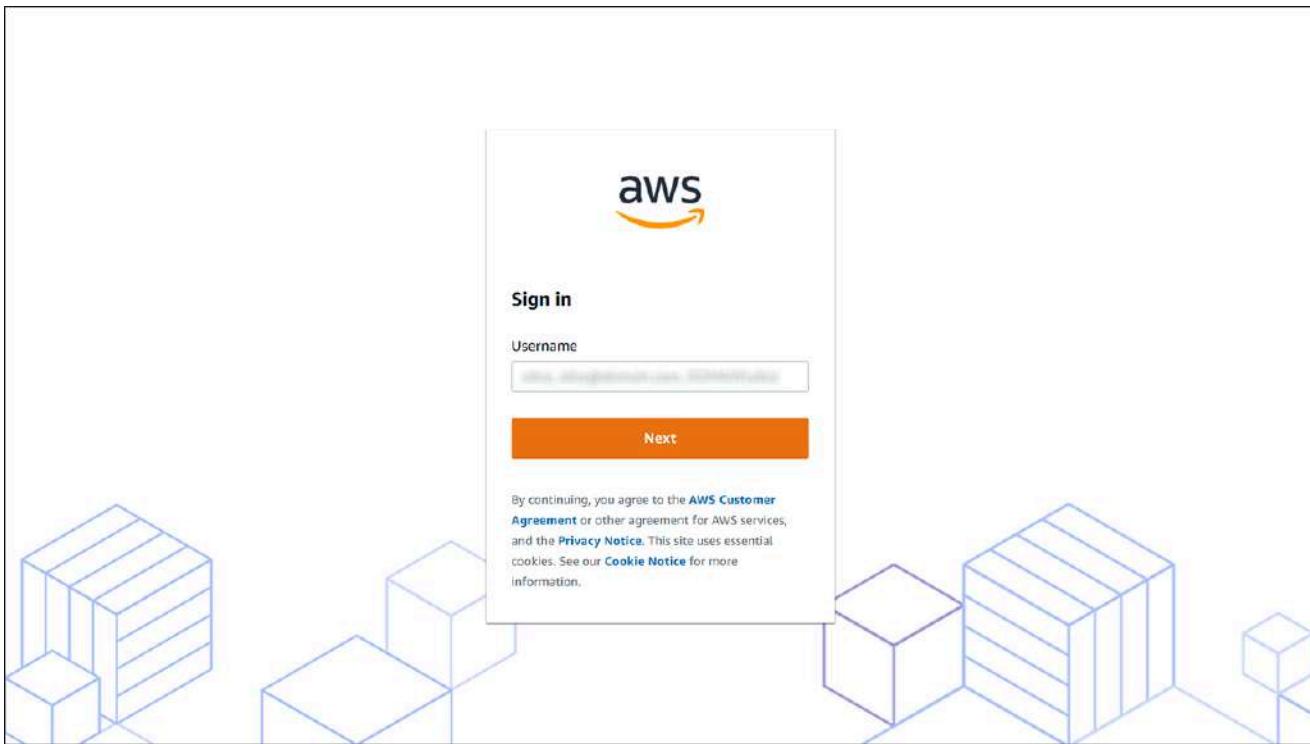
The first step to **configuring Deadline** is to sign in to the Nimble Studio portal as **Admin** and launch a Windows virtual workstation (an instance).

To connect to the Nimble Studio portal

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.

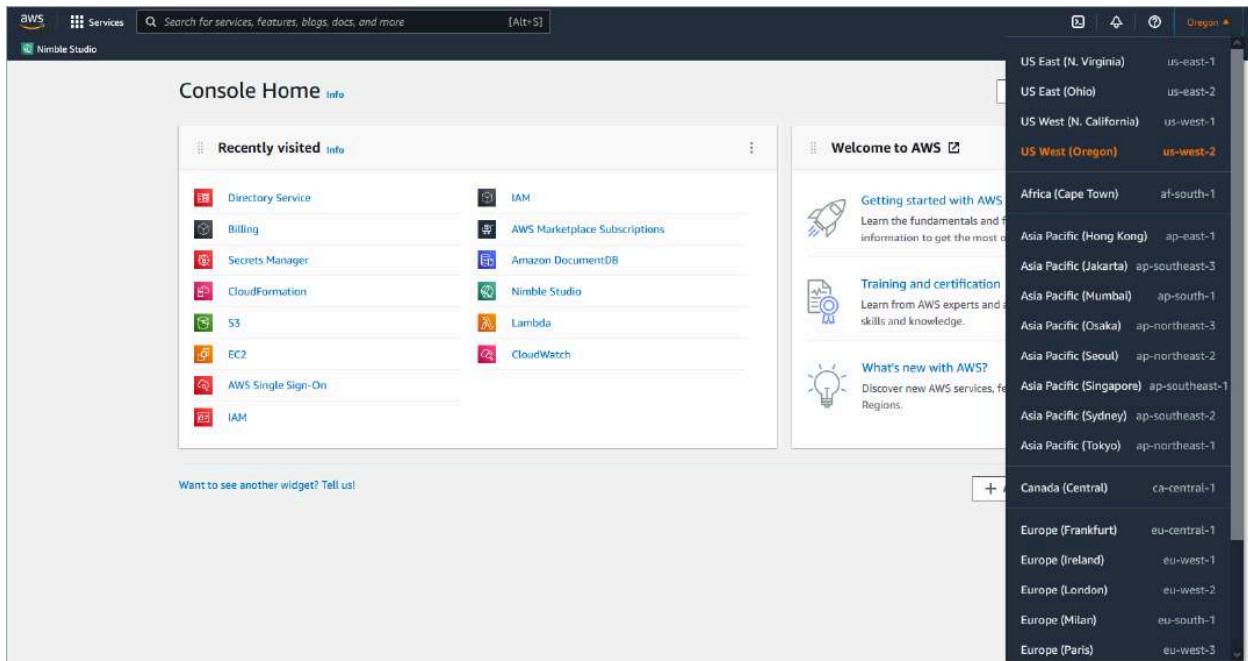


3. Choose **Studio manager** in the left navigation pane.
4. On the **Studio manager** page, choose **Go to Nimble Studio portal**.
5. Sign in to the Nimble Studio portal using your AWS Managed Microsoft AD administrator credentials.
 - a. Enter **Admin** as the user name. Use the password that you set up during StudioBuilder deployment.



b. If you forgot your password, do the following:

- i. Sign in to the AWS Management Console and open the [AWS Directory Service](#) console.
- ii. In the AWS Region selector (top-right navigation bar), verify that the Region for your studio is selected.



- iii. Select the **Directory ID** for your studio's Active Directory.
 - iv. Choose **Reset user password**.
6. Bookmark your portal's URL so that you can get to your studio directly, later.

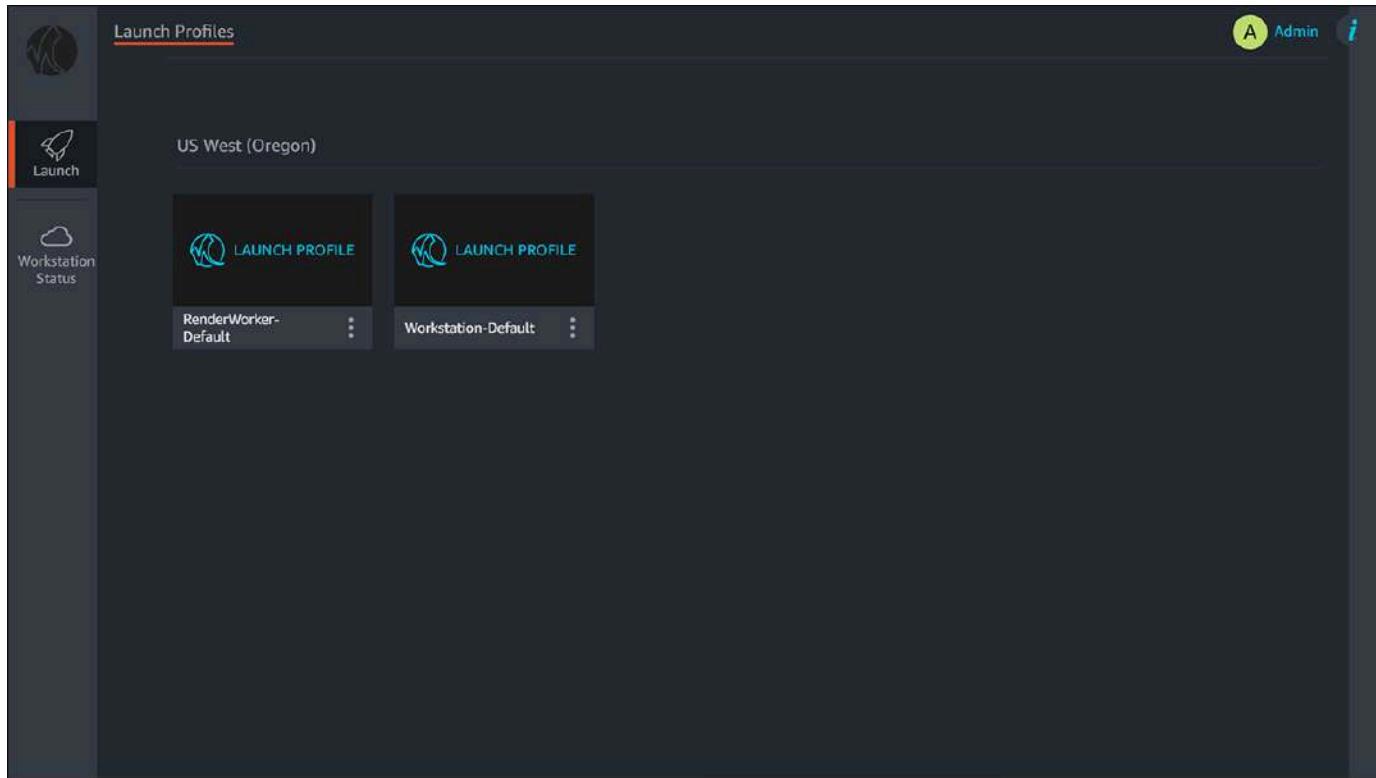
Step 2: Accept the EULA

Before using Nimble Studio, accept the End User License Agreements. You can access these agreements on the settings page in the Nimble Studio portal.

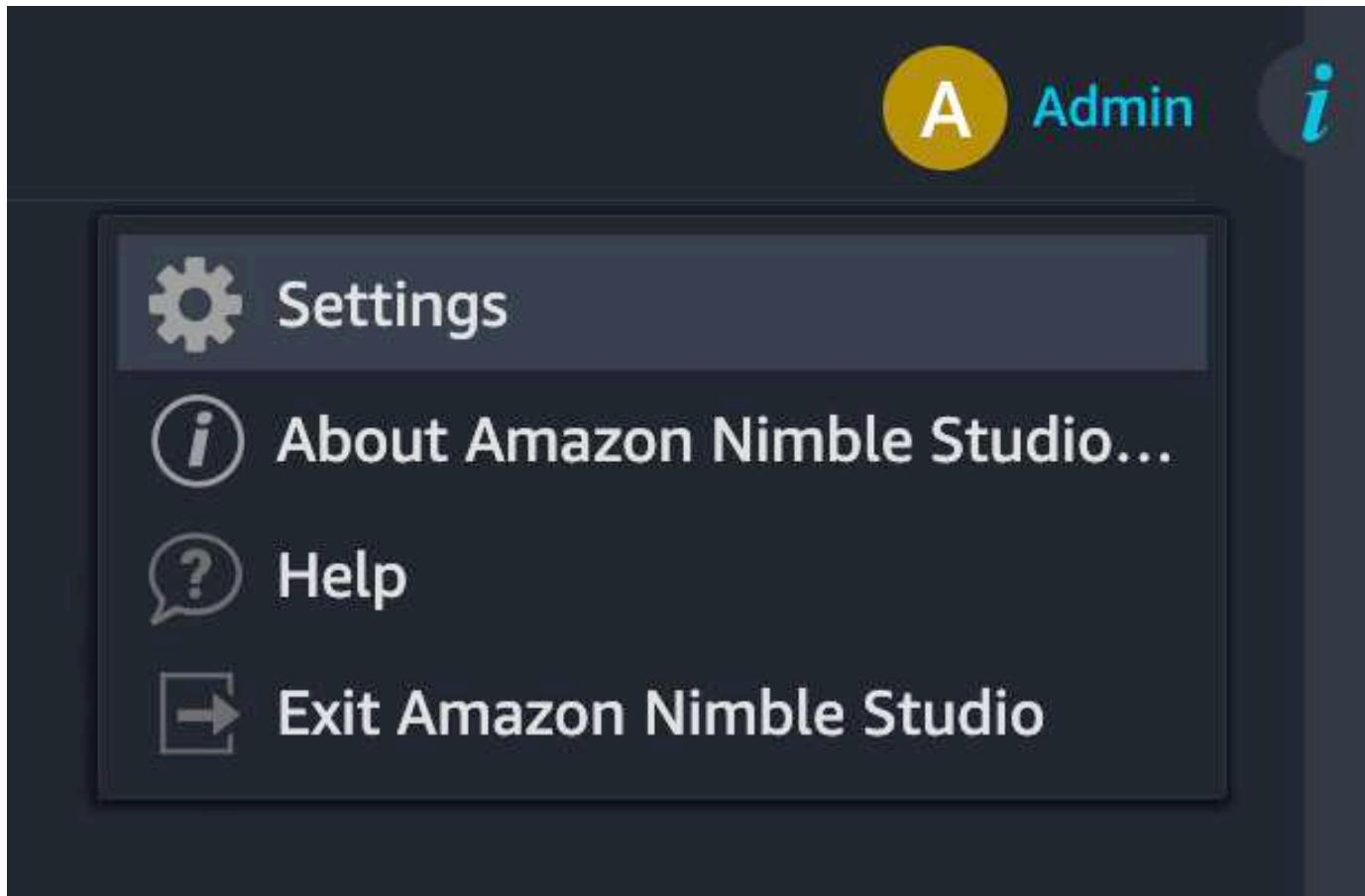
 **Note**

If you have already completed the [Adding studio users](#) tutorial, then most likely you have already accepted the EULA and can skip this step and continue to [Step 3: Launch a virtual workstation](#). If you have not completed the other tutorial, continue to the next step.

1. Choose the **Launch** tab from the left navigation pane.



2. In the upper right-hand corner of the Nimble Studio portal, choose your **user name**.
3. Choose **Settings** from the dropdown menu.



4. Choose **EULA** from the left navigation pane.

The screenshot shows the 'EULA' section of the 'Settings' menu. It displays a table of agreements with their status and usage. A checkbox at the bottom indicates agreement to the terms.

AGREEMENT	STATUS	USED BY
DJV-DJV	Not accepted	NimbleStudioStreamWindowsImage NimbleStudioStreamLinuxImage
EpicGames-Unreal	Not accepted	NimbleStudioStreamWindowsImage NimbleStudioStreamLinuxImage
SideFX-Houdini	Not accepted	NimbleStudioStreamWindowsImage NimbleStudioStreamLinuxImage
ChaosGroup-Vray	Not accepted	NimbleStudioStreamWindowsImage NimbleStudioStreamLinuxImage
Foundry-Nuke	Not accepted	NimbleStudioStreamWindowsImage NimbleStudioStreamLinuxImage
		NimbleStudioStreamWindowsImage

I have read and agree to the terms of the End User License Agreements above.

CONFIRM

5. Open and read each agreement in the list.
6. After you read all of the agreements, select the check box next to **I have read and agree to the terms of the End User License Agreements above.**
7. Choose **Confirm**. The status of each of the agreements will change to Accepted.

Step 3: Launch a virtual workstation

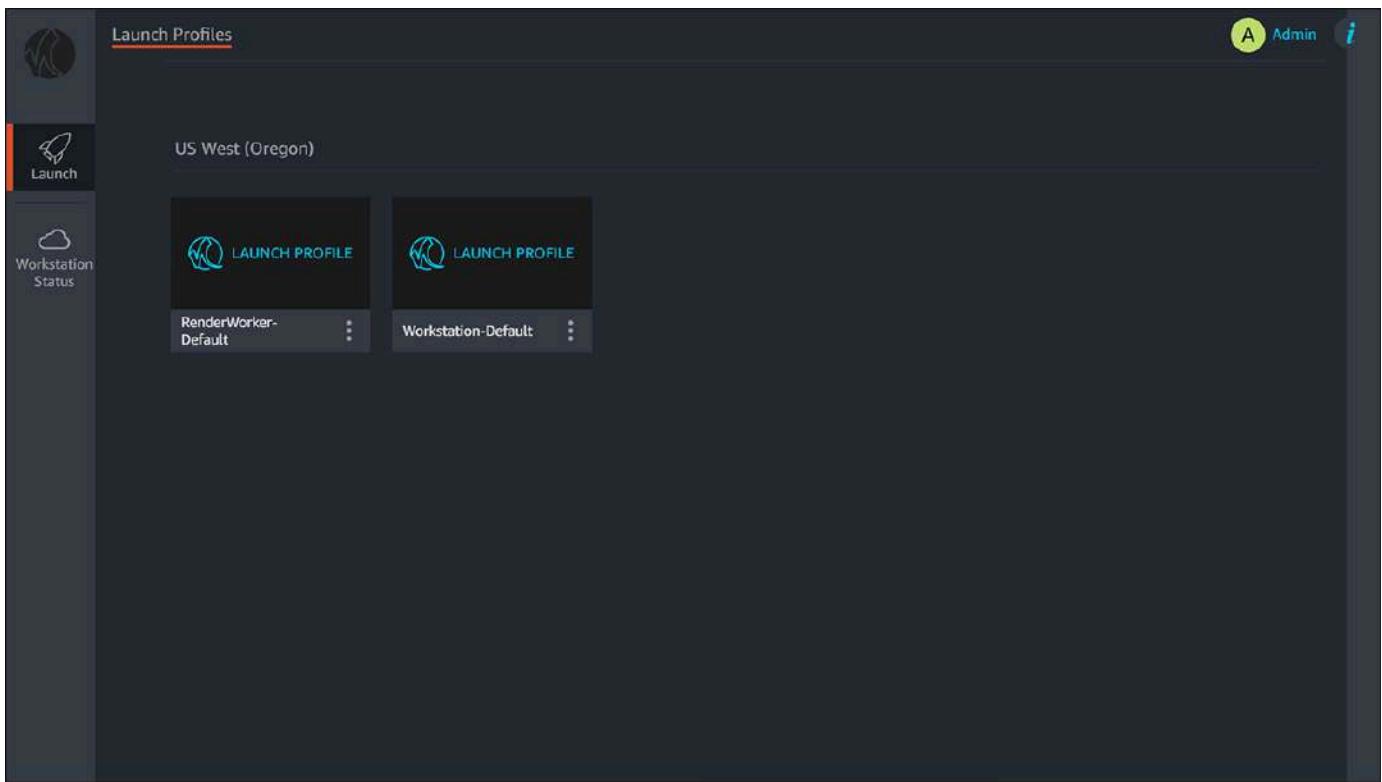
Once you've accepted the EULA, you can proceed to launching a virtual workstation.

Note

Before you can launch a virtual workstation, first install the latest [DCV client](#).

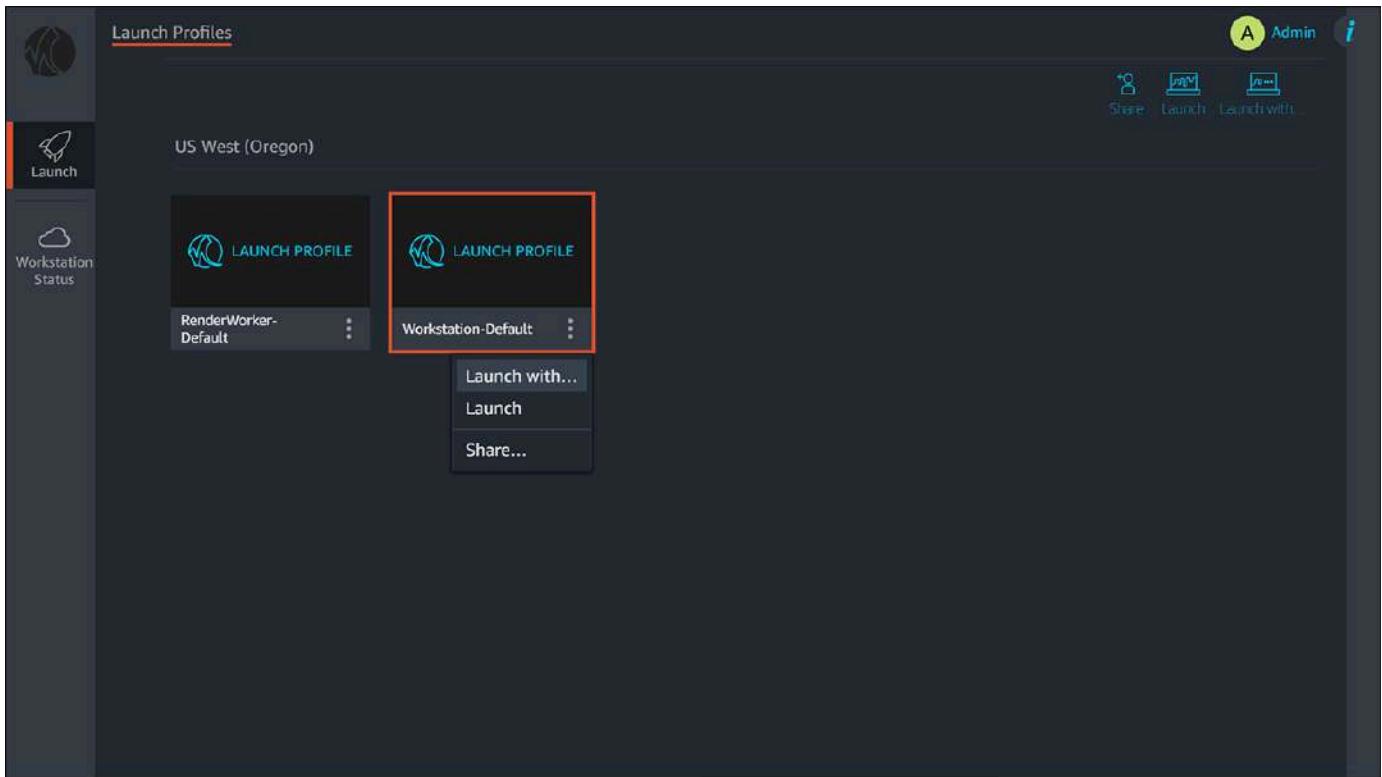
To launch a virtual workstation

1. Choose the **Launch** tab from the left navigation pane.



2. Select the vertical ellipsis

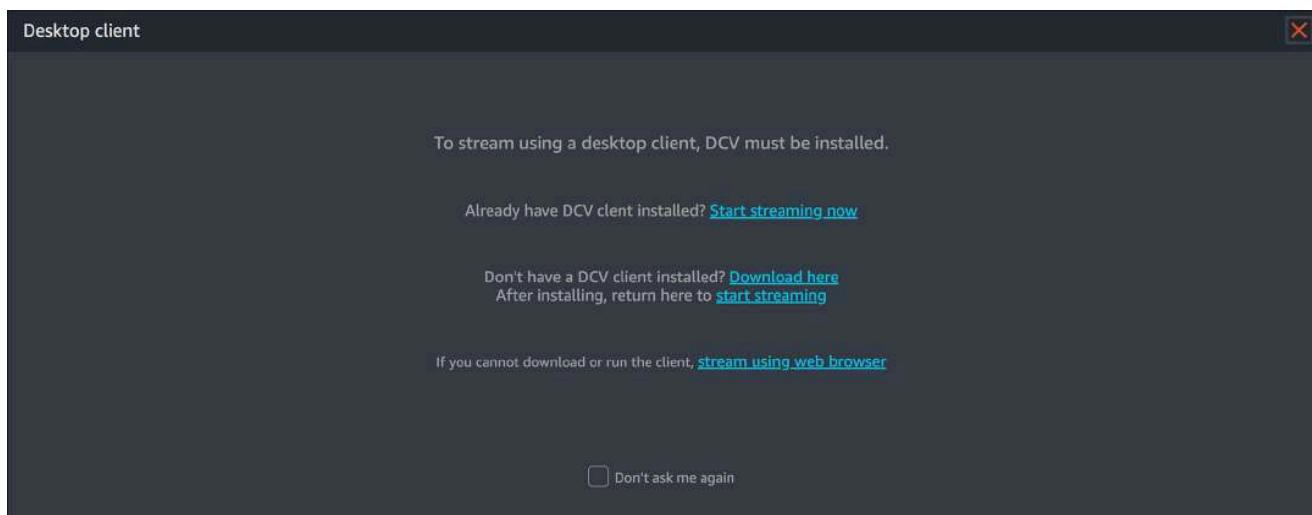
(⋮) on the card to open a dropdown menu.



3. Choose **Launch with...**
4. For **Instance Type**, keep it at the default setting.
5. For **Amazon Machine Image**, verify that **NimbleStudioWindowsStreamImage** is selected.
6. For **Streaming Preference**, choose your streaming preference.
 - a. For the best performance, we recommend choosing **Launch native client**.
 - b. You must download the NICE DCV client before connecting to your workstation. For more information about the DCV client, as well as links to download, see NICE DCV clients [NICE DCV clients](#).
7. Choose **Launch**.
8. A status bar will appear that shows you the progress of launching your virtual workstation. This might take up to 10 minutes.

To connect to the virtual workstation

1. When your virtual workstation is ready, a new window appears reminding you that the client must be installed.
2. Choose **Start streaming now**.
 - If you haven't installed the NICE DCV desktop client, choose **Download here** and install the client first.



3. When your browser displays a window that prompts you to open NICE DCV, choose **Open** to continue. The exact wording of this might vary depending on what browser you're using.

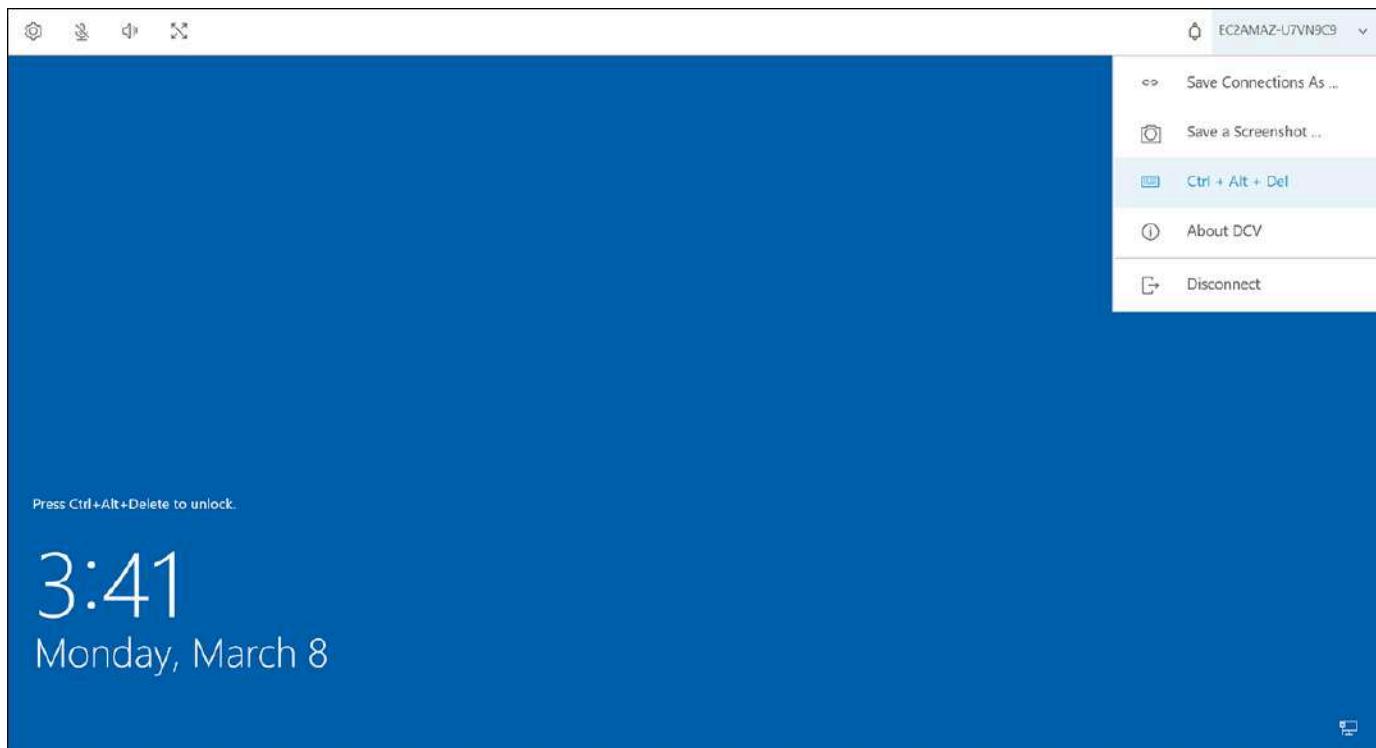
Note

The NICE DCV web browser client runs inside a web browser. You don't need to install the web client. We recommend using the Google Chrome browser to avoid latency. For more information, go to the [Web browser client](#) page in the *NICE DCV User Guide*.

4. After the NICE DCV client application opens in a new window, the Windows login screen will display.
5. Open the instance menu near the top right of the screen and choose **Ctrl + Alt + Del**. For an OS X DCV client, open the **Connection** dropdown menu and select **Send Ctrl + Alt + Del**.

Important

Don't enter **Ctrl+Alt+Delete** on your keyboard. Doing so sends the command to your local computer, not to your workstation.



6. For **User name**, enter **Admin**. For **Password**, enter the password that you created during your studio deploy. Then press the enter (or return) key.

You're now connected to your virtual workstation.

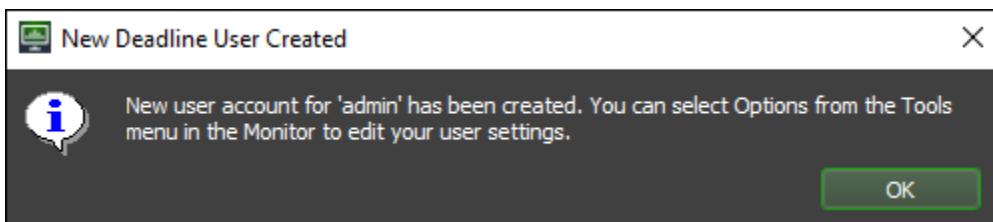
Step 4: Open Deadline and create groups

Now that you're logged in to your virtual workstation, the first thing you will do is open Deadline Monitor. This is the main interface that you and your artists will use to change your render farm settings and monitor the status of renders on your farm.

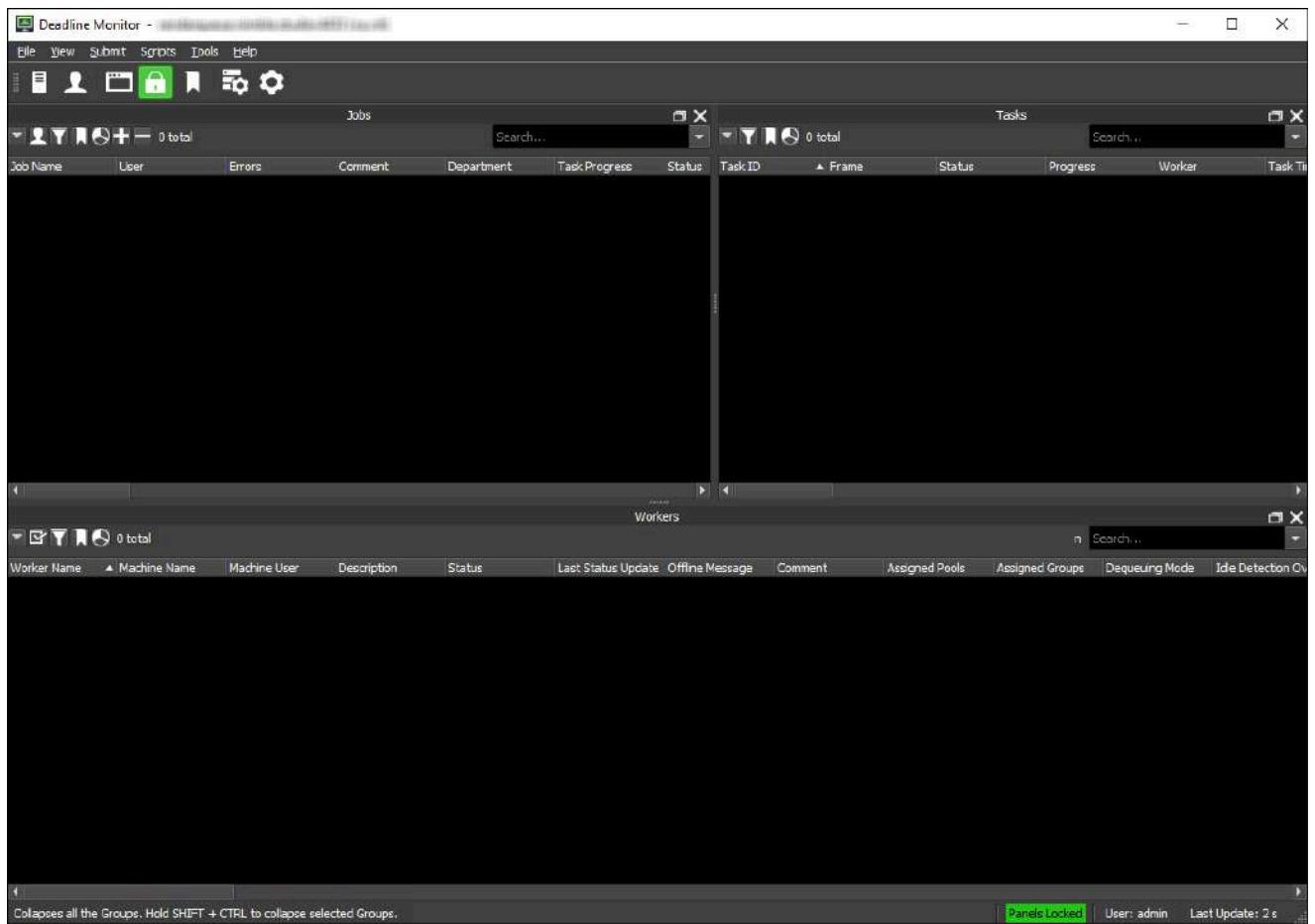
In Deadline Monitor, create a group for each render fleet in your studio. By default, you only create one render fleet during your deploy with StudioBuilder, but if you created more than one, repeat these steps for each fleet that you created.

To open Deadline Monitor

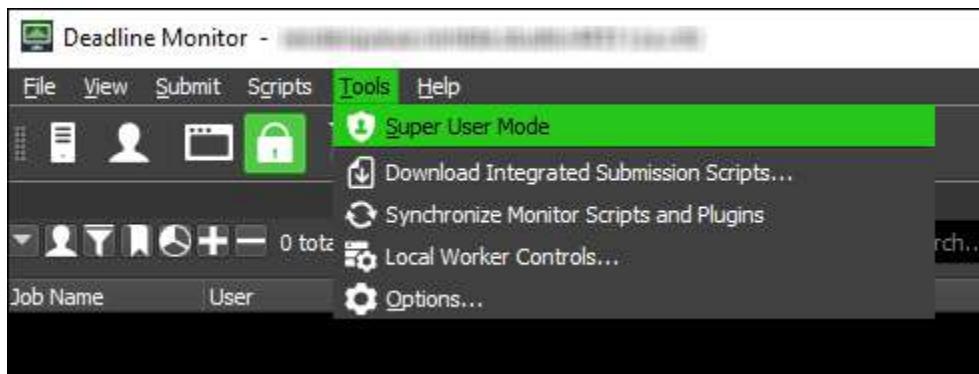
1. Open the **Start** menu. Choose **Thinkbox**. Then choose **Deadline Monitor**.
 - When Deadline Monitor starts opening, a window will appear. This indicates that the system has created a new user account.
2. Choose **OK** to continue.



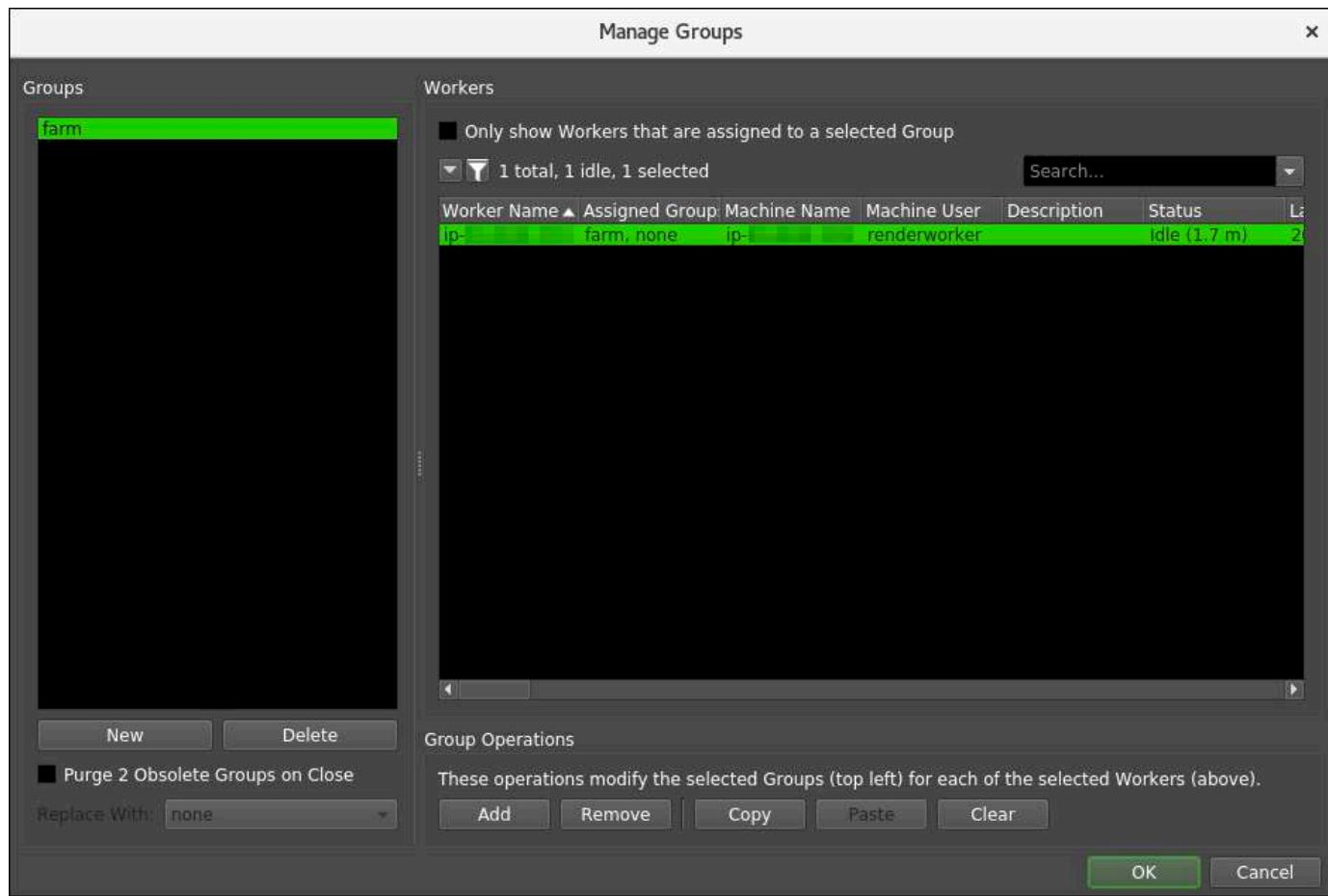
- The Deadline Monitor window will open.



3. Choose **Tools**. Then choose **superuser Mode**.



4. Next, choose **Tools**. Then choose **Manage Groups...**
5. In the window that appears, choose **New**.
6. Enter the **name of your render fleet** and then choose **OK**.

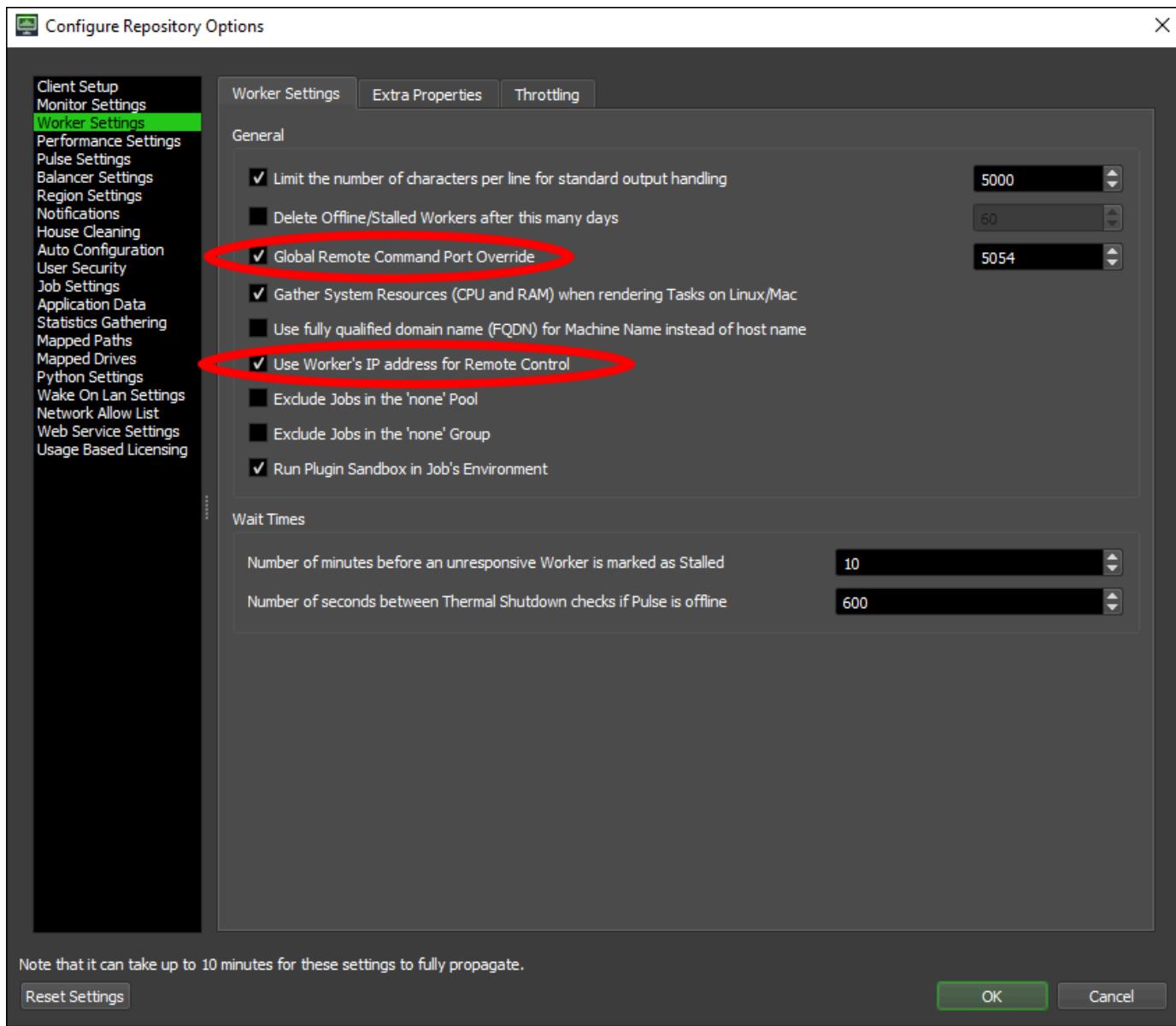


7. Repeat for any additional render fleets that you created during your deploy with StudioBuilder.
8. Choose **OK** when you're finished adding groups.

Step 5: Adjust worker settings

Next you will adjust settings on your render workers to allow you or your artists to connect to them and view the worker logs. This will make it possible to view more log information for each worker and will help you diagnose any problems that might occur with a particular worker in your render fleet.

1. Choose **Tools**. Then choose **Configure Repository Options...**
2. In the Configure Repository Options window, choose **Worker Settings** from the list on the left.
3. Select the check box next to **Global Remote Command Port Override**.
4. Select the check box next to **Use Worker's IP address for Remote Control**.



Step 6: (Windows only) Set mapped paths

Note

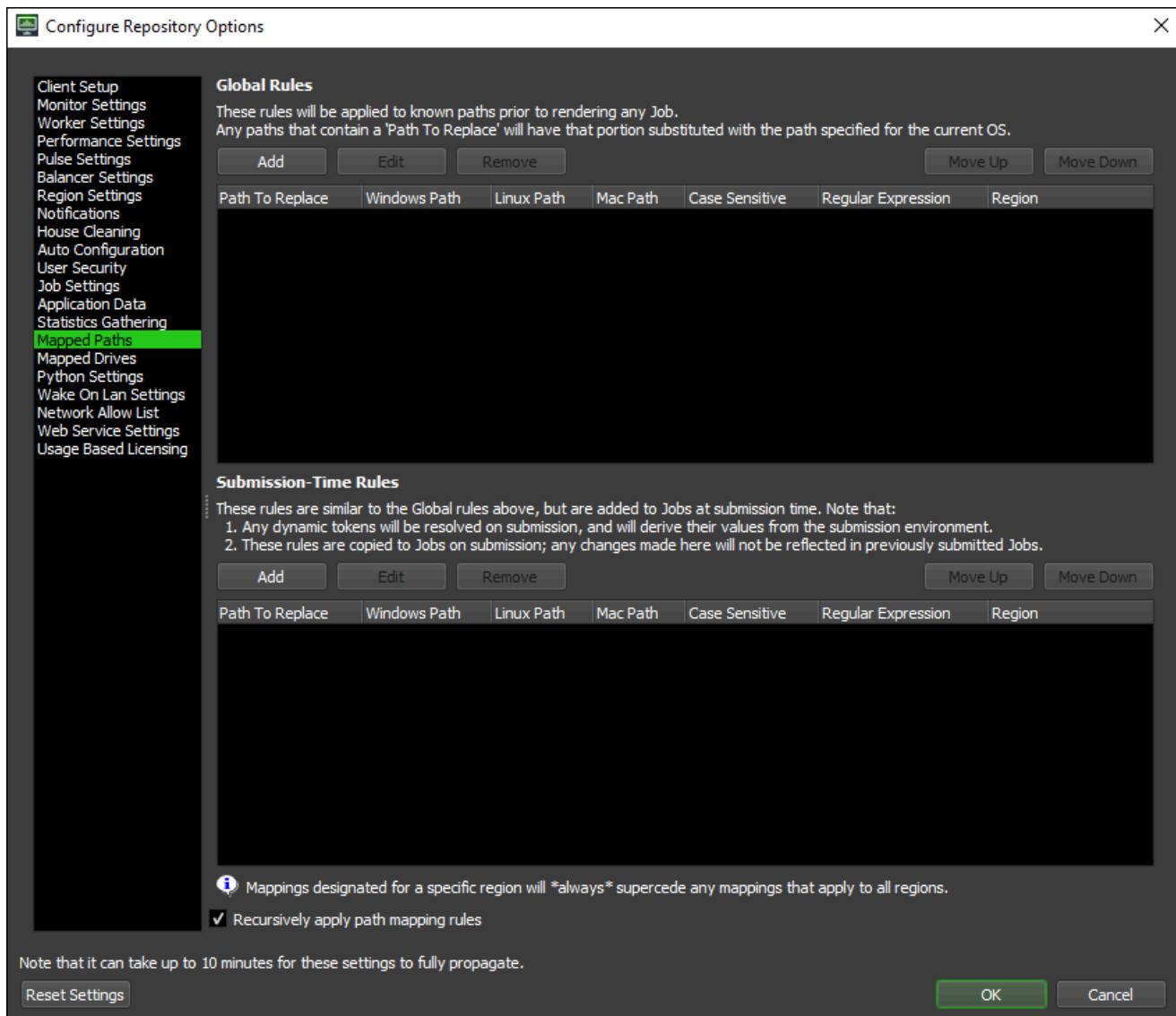
You only need to complete this step if you're using Windows virtual workstations to submit renders to your farm. If you're only using Linux virtual workstations, you can skip this step.

Next, you will set mapped paths for the shared storage that connects to your Windows workstations. By default, StudioBuilder will create one Amazon FSx file system, the Z: drive, when

you deploy your studio. If you've added additional storage to your studio, such as a second Amazon FSx file system, you will need to map paths for it as well.

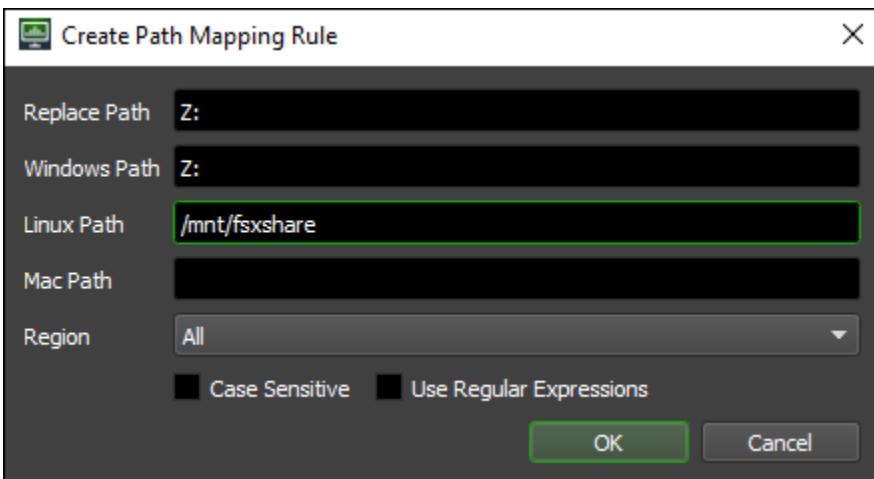
1. While still in the Configure Repository Options window, choose **Mapped Paths** from the list on the left.

- The mapped paths options will appear.



2. In **Global Rules**, choose **Add**.
3. In the window that appears, enter the following:
 - a. For **Replace Path** enter Z:
 - b. For **Windows Path** enter Z:

c. For **Linux Path** enter **/mnt/fsxshare**



4. Confirm your entries and then choose **OK**.

Note

The previous entries are for mapping the default Amazon FSx file system that is automatically created when you deploy your studio with StudioBuilder. If that is the only file system in your studio, then you're done and your artists should be ready to render.

However, if you created additional file systems using the instructions in the [Setting up an Amazon FSx Windows file system](#) tutorial, you will need to add another rule to Deadline for those file systems and enter the appropriate path files. See the following instructions.

Add global rules for additional file systems

First, you will need to find the path values for your additional file systems in the Nimble Studio console.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. In the **File storage** section, choose the **Component ID** for the additional file system that you're interested in. For example, you might have added another file system to your studio and named it **Prod**.
4. On the **File storage details** page for the file system, navigate to the **File storage configuration** section and notice the **Linux mount point** and the **Windows mount drive**.

File storage configuration

Linux mount point

/mnt/prod

Windows mount drive

X

Share name

share

5. Go back to the **Configure Repository Options** window in the **Deadline Monitor** on your virtual workstation.
6. In **Global Rules**, choose **Add** and enter the following information.
 - a. Use the **Windows mount drive** as both the **Replace Path** and the **Windows Path**, and add a colon to the end. For example, X:.
 - b. Use the **Linux mount point** as the **Linux Path**.

Related resources

- [NICE DCV clients - NICE DCV](#)

Working with render farms

A render farm is a computer system that is built to render computer generated images (CGI). They're typically used to create animated films, visual effects, and architectural visualizations. This administrator tutorial series explains how render farms work with Amazon Nimble Studio, and how to set up and maintain render farms.

When you create your cloud studio with Amazon Nimble Studio, you have the option to bring an existing render farm, or to create a new one using AWS resources. After you create your render farm, you can customize it.

Topics

- [Setting up Deadline Usage Based Licensing with Nimble Studio](#)
- [Mounting the Deadline Repository file system on Linux based studios](#)

Setting up Deadline Usage Based Licensing with Nimble Studio

[Usage Based Licensing \(UBL\)](#) is a service from Deadline that gives render workers access to Deadline and third party licenses purchased from the [AWS Thinkbox Marketplace](#). UBL connects to the Cloud License Server (CLS) through license forwarders on a [Remote Connection Server \(RCS\)](#) without directly accessing the external internet. By using UBL, you can have multiple instances render at the same time, and you'll pay for what you use.

This tutorial explains how to configure your Amazon Nimble Studio so that you can use it with Deadline Usage Based Licensing.

Contents

- [Prerequisites](#)
- [Step 1: Enable UBL support in your studio](#)
- [Step 2: Purchase UBL](#)
- [Step 3: Download UBL certificates](#)
- [Step 4: Upload certificates to Secrets Manager](#)
- [Step 5: Restart the UBL cluster](#)
- [Step 6: Restart workers in the fleet](#)
- [Step 7: Enable UBL on Deadline Monitor](#)
- [Step 8: Set limits](#)
- [Related resources](#)

Prerequisites

- To complete this tutorial, you need to have an AWS Thinkbox Marketplace account. If you don't have a Thinkbox Marketplace account, follow the instructions in [The Thinkbox Marketplace tutorial](#).

Step 1: Enable UBL support in your studio

You can enable UBL when you deploy a new studio using StudioBuilder, or when you update and edit an existing studio. To enable UBL, choose **Yes, I want to enable UBL support for my studio**. The CDK will output a command that is similar to the following:

```
aws secretsmanager put-secret-value --secret-id  
arn:aws:secretsmanager:<studio_region>:<account_id>:secret:StudioBuilder-  
UBLLicense-<UBL_license> --secret-binary fileb://[file-name]
```

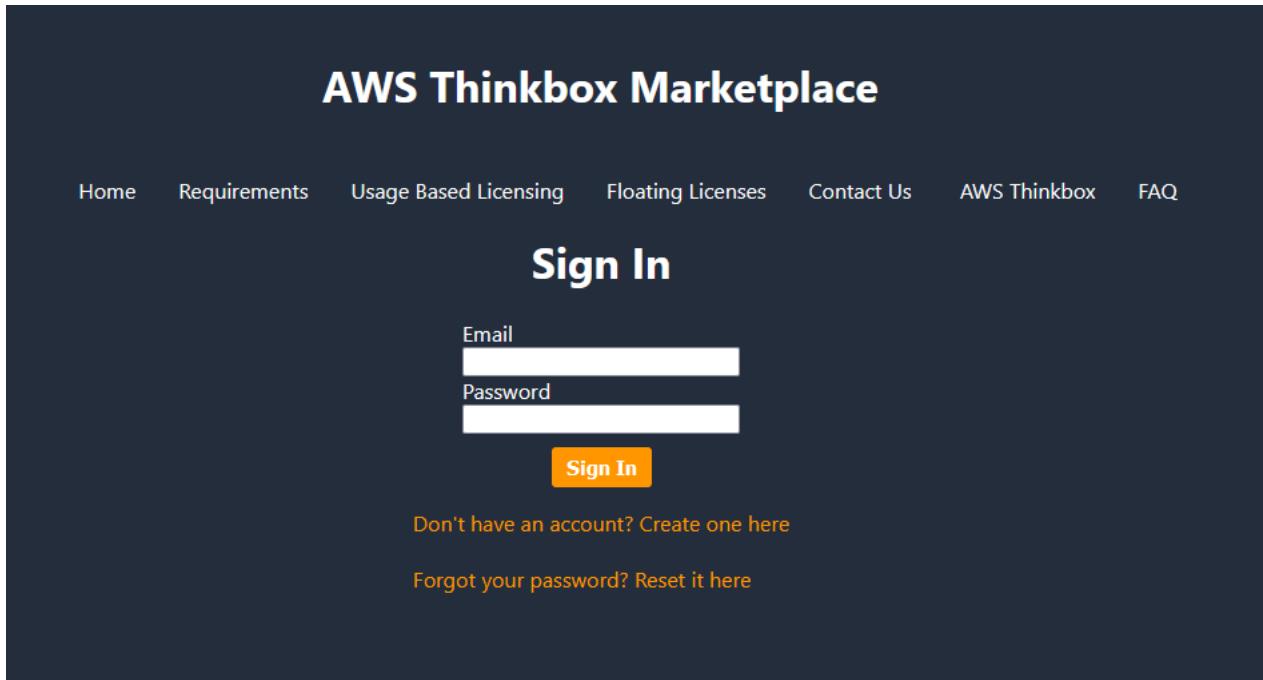
Copy the command that the CDK outputs. You will use the secret ARN from it in [Step 4: Upload certificates to Secrets Manager](#).

Step 2: Purchase UBL

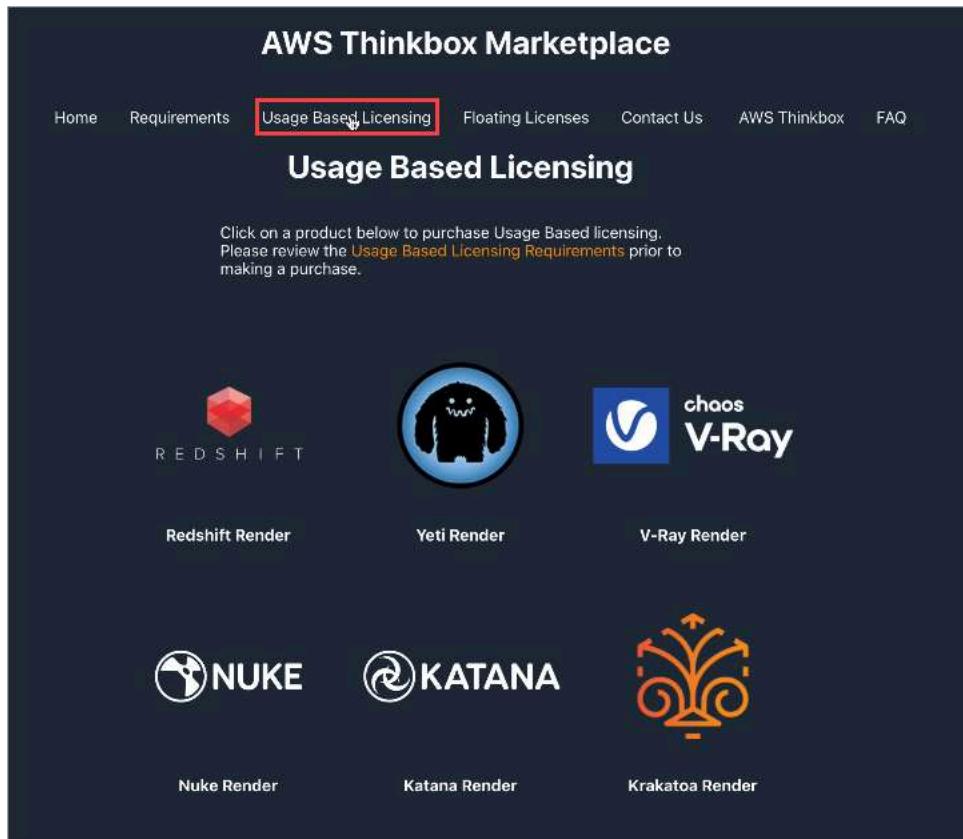
In this section, you will purchase the UBL from the AWS Thinkbox Marketplace.

1. Go to the [AWS Thinkbox Marketplace](#).

- Sign in with the email and password that you used when you created your AWS Thinkbox Marketplace account.



2. Choose **Usage Based Licensing**.



3. Select the application that you'd like to purchase.
4. Select the number of hours that you would like to purchase.
5. Choose **Add to cart**.
6. Select **Cart** at the top-right corner of the menu bar.
7. Read the terms and conditions. If you agree to the terms and conditions for Arnold and Thinkbox Marketplace, select both boxes.
A dark rectangular dialog box contains two checkboxes:
 - I agree with the [terms and conditions](#) for Arnold
 - I agree with the [terms and conditions](#) for the Thinkbox MarketplaceBelow the checkboxes is a yellow 'Checkout' button with a cursor icon pointing to it.
8. Choose **Checkout**.
9. Select **Confirm**.
10. You will receive an email with a URL and license server. Save this email. The URL and license server will be used in [Step 7: Enable UBL on Deadline Monitor](#).

- a. The URL will look something like this: <https://thinkbox.compliance.flexnetoperations.com/instances/XXXXXXXXXXXXX/request>
- b. The license server will be a 16 digit number. Example: XXXX-XXXX-XXXX-XXXX

Step 3: Download UBL certificates

Next, download the zip file for UBL certificates. You will upload this file to AWS Secrets Manager later, in [Step 4: Upload certificates to Secrets Manager](#).

1. Go to the [Thinkbox Customer Portal](#).
 - Sign in to AWS Thinkbox Marketplace.
2. Choose **Home** in the left navigation pane.
3. Choose **certificates**.

The screenshot shows the 'Product Information' page for certificates. On the left, there's a sidebar with 'Software & Services' and 'Devices' sections. The 'Home' link in the top-left corner is highlighted with a red box. The main area shows a table with one row labeled 'certificates'. Below the table, there are 'New Versions' and 'Release Archive' buttons.

4. In the **File Name** section, download the UBL certificates zip file.

The screenshot shows the 'Product Download' page for certificates. It has tabs for 'Files', 'License Keys', 'Download Log', 'Notes', 'Notification Log', and 'Restrict Access'. A note at the top states: 'The software you are about to download is subject to export control laws and regulations. By downloading this software, you agree that you will not knowingly, without prior written authorization from the competent government authorities, export or reexport - directly or indirectly - any software downloaded from this website to any prohibited destination, end-user, or end-use.' Below this, a table lists files with columns for 'File Description', 'File Size', and 'File Name'. The 'File Name' column shows a file named 'certificates_e1bb00296-6cf3-4b1c-ba3f-0407a0cd29d1.zip', which is highlighted with a red box.

Step 4: Upload certificates to Secrets Manager

Because Thinkbox Marketplace and Nimble Studio are two separate entities, you will first upload your UBL certificates to AWS Secrets Manager so that Nimble Studio can fetch them.

Upload certificates using CloudShell

Note

CloudShell isn't supported in the eu-west-2 or ca-central-1 regions. If you're in either of those regions, follow the instructions in [Upload certificates using the AWS CLI](#).

1. Sign in to the AWS Management Console.
2. Sign in to the AWS Management Console.
3. Go to **Services** and then select **CloudShell**.



4. Wait for the CloudShell session to load.
5. Wait for the CloudShell session to load.
6. Choose **Actions**. Then choose **Upload**.
7. Choose the UBL certificates zip file.
8. Run the following command: `aws secretsmanager put-secret-value --secret-id <secret-arn> --secret-binary fileb://<file-name>`
 - a. Replace the <secret-arn> with the ARN provided by StudioBuilder. This is the `arn:aws:secretsmanager:<studio_region>:<account_id>:secret:StudioBuilder-UBLLicense-<UBL_license>` value that you found in [Step 1: Enable UBL support in your studio](#).
 - b. Replace <file-name> with the name of the UBL certificates zip file that you downloaded in [Step 3: Download UBL certificates](#).

Upload certificates using the AWS CLI

If you're in the eu-west-2 or ca-central-1 regions, use the AWS CLI to update the UBL certificates. The following steps instruct you about how to install and configure the AWS CLI.

1. To install or upgrade the AWS CLI on your local machine, follow the instructions in [Installing the AWS Command Line Interface version 2](#) in the *AWS Command Line Interface User Guide*.
2. Configure the AWS CLI by following the instructions in [Setting up new configuration and credentials](#).
3. Verify the installation or upgrade by running `aws nimble help`. This command displays a list of available Nimble Studio commands.
4. Run the following command: `aws secretsmanager put-secret-value --secret-id <secret-arn> --secret-binary fileb://<file-name>`
 - a. Replace the `<secret-arn>` with the ARN provided by StudioBuilder. This is the `arn:aws:secretsmanager:<studio_region>:<account_id>:secret:StudioBuilder-UBLLicense-<UBL_license>` value that you found in [Step 1: Enable UBL support in your studio](#).
 - b. Next, replace `<file-name>` with the name of the UBL certificates zip file that you downloaded in [Step 3: Download UBL certificates](#).

Step 5: Restart the UBL cluster

To reload the updated license file, restart the UBL cluster on the Elastic Container Service (ECS) console.

1. Sign in to the AWS Management Console and open the [Amazon ECS](#) console.
2. In the left navigation pane, choose **Clusters**.
3. Select the cluster named `<your-studio-name>-UsageBasedLicensingCluster`.
4. Select the **Tasks** tab.
5. Select all running tasks.
6. Select Stop.

The UBL service will create a new task to reload the certificate.

Step 6: Restart workers in the fleet

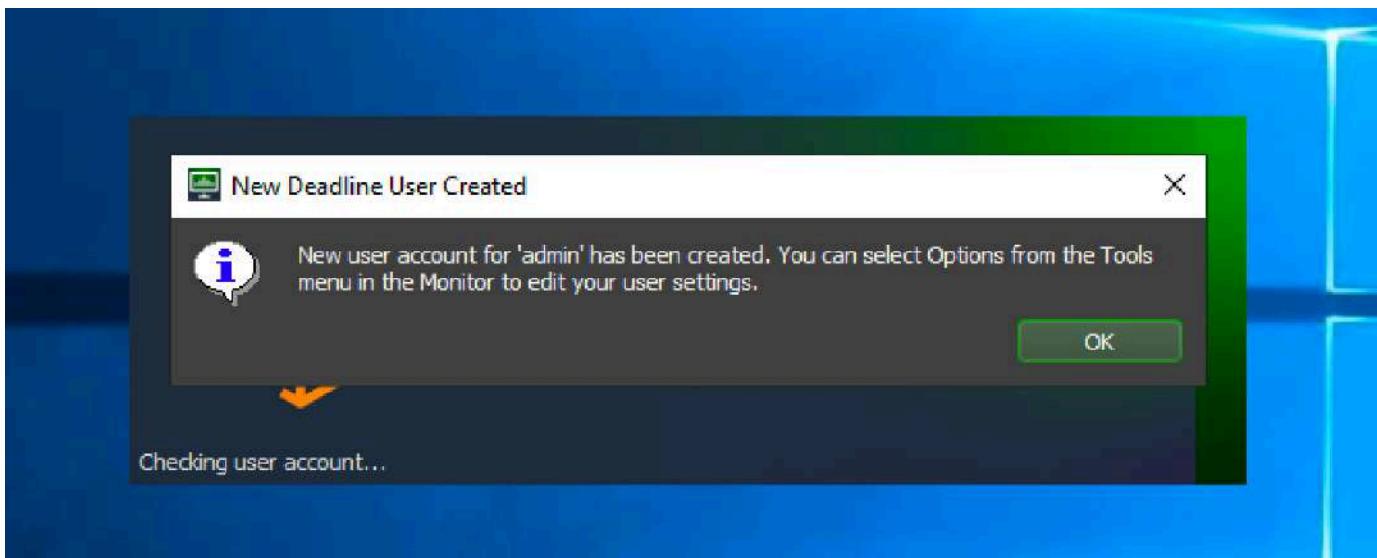
Manually restart your workers in the fleet to register them with Deadline Secrets Management in RCS. Follow the [Remote Controlling Workers, Pulses, and Balancers](#) tutorial to restart your workers in the fleet.

Note

This step is only required if you're adding UBL support to an existing studio with fleet instance type **On Demand**.

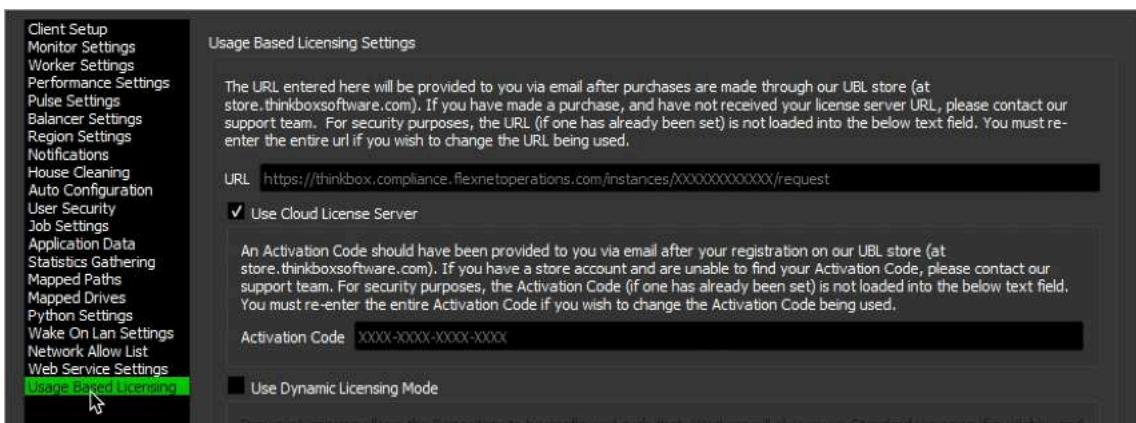
Step 7: Enable UBL on Deadline Monitor

1. Sign into the Nimble Studio portal by following the instructions in [Step 2: Accept the EULA](#).
2. Launch a workstation by following the instructions in [Step 4: Add users to AWS Managed Microsoft AD](#).
3. Choose the **Start** menu, located in the lower-left corner of your desktop.
4. Enter **Deadline** to search for **Deadline Monitor**, and then choose it from the top of the search results.
5. If this message pops up choose **OK**.



6. In the **Tools** menu, go to **Tools**. Then select **Super User Mode**.
7. Next go to **Tools**. Then select **Configure Repository Options**.
8. In the left navigation pane, choose **Usage Based Licensing**.

9. Enter your license server URL in the **URL** field. Enter your activation code in the **Activation Code** field. This is the information that was emailed to you in [Step 2: Purchase UBL](#).



10. Select **OK**.

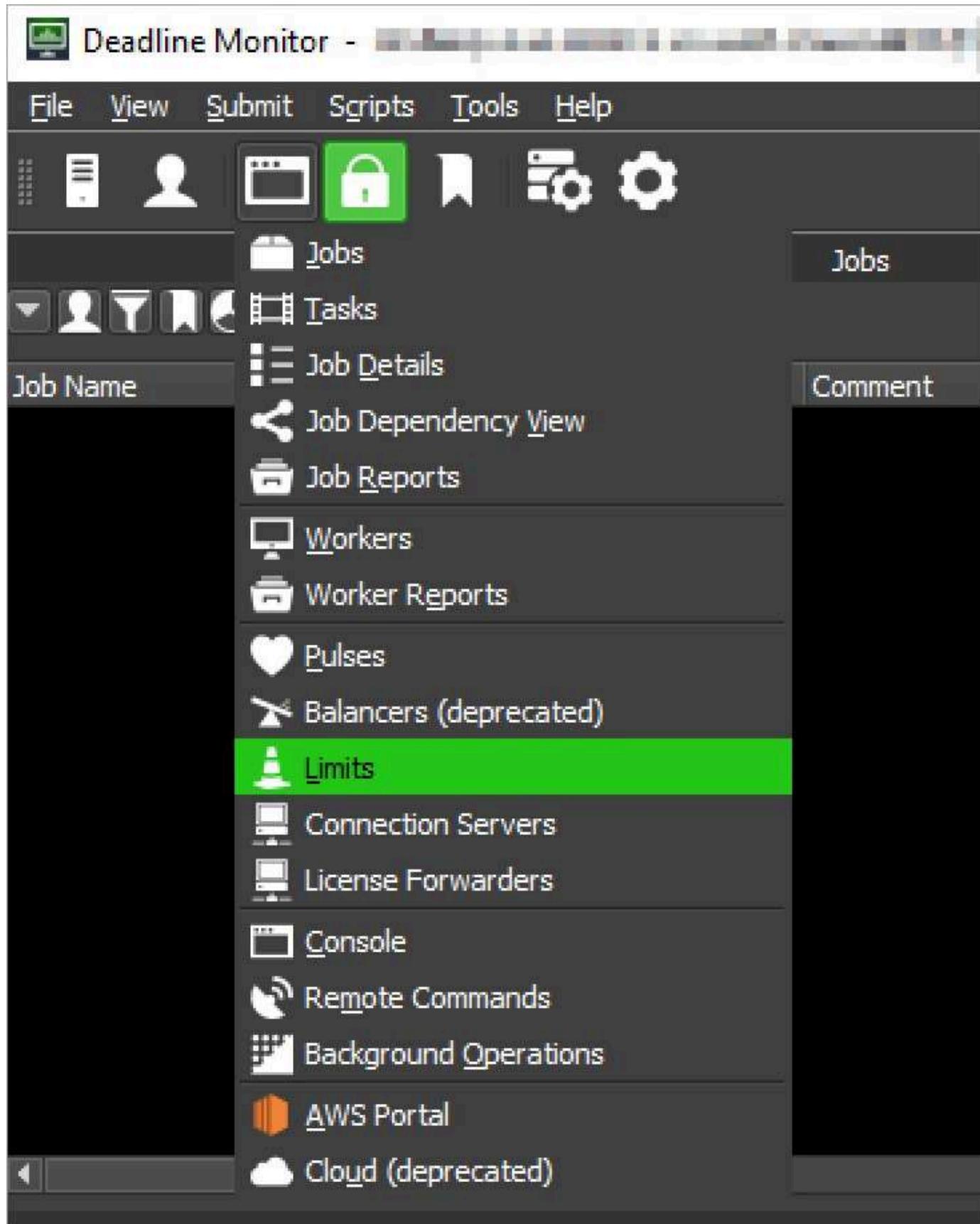
11. Deadline will ask you for your Deadline Secrets Management admin credentials.

- You can obtain your administrator credentials by following the instructions in the [Accessing Administrator Credentials](#) tutorial in the [Render Farm Deployment Kit on AWS Developer Guide](#).

Step 8: Set limits

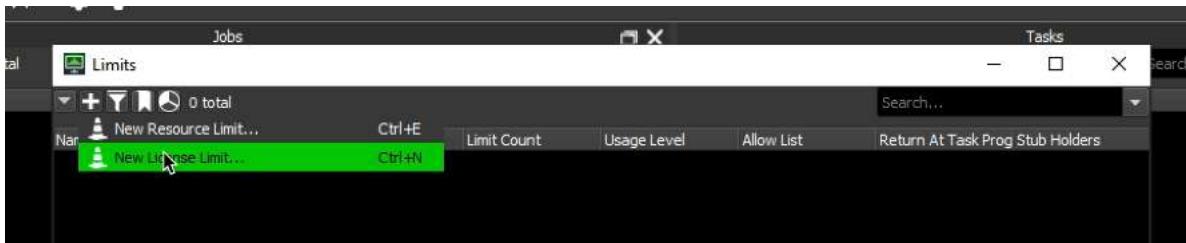
Studio owners can specify how many licenses they want to provide permission for by setting license limits. For more information about Deadline limits, see the [Deadline documentation](#).

1. In the monitor choose the **Panels** icon.
2. Choose **Limits**.



3. Select the + icon to create a new limit.

4. Choose **New License Limit**.

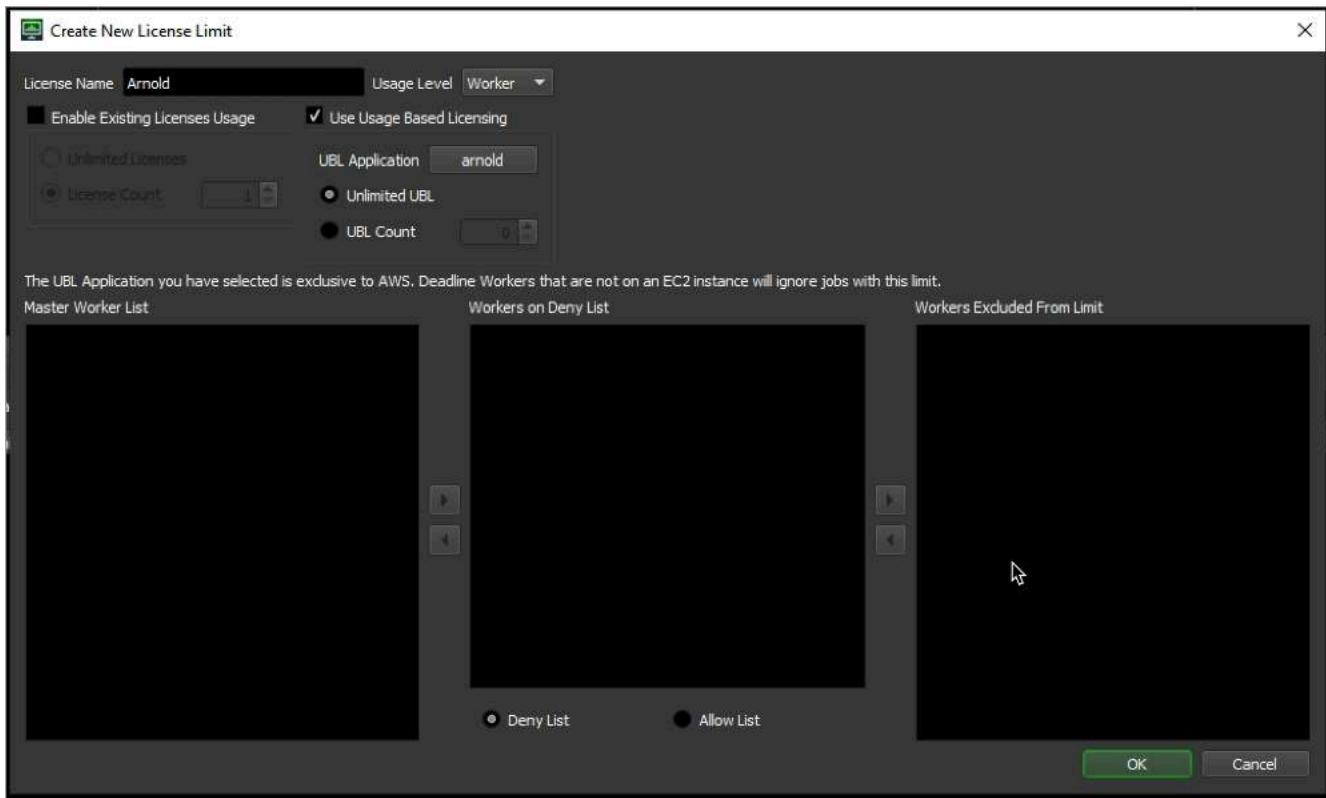


5. Set **License Name** to the application that you want to render with. Example: Arnold.

6. Check the box next to **Use Usage Based Licensing**

7. Set **UBL Application** to the application you entered in step 5.

8. Select **Unlimited UBL**.



When you **Submit** a render, specify the limit for the application that you're using.

Related resources

- [Configuring Deadline](#)

Mounting the Deadline Repository file system on Linux based studios

The Deadline Repository is a global file system component that stores the plugins, scripts, logs, and any auxiliary files (like scene files) that are submitted with jobs. To access the file system, mount the Deadline Repository file system to your Nimble Studio cloud studio.

This administrator tutorial explains how to mount the Deadline Repository file system on Linux based studios.

Contents

- [Prerequisites](#)
- [Step 1: Create a new security group](#)
- [Step 2: Update the security group for the Repository file system](#)
- [Step 3: Update the Backend network ACL](#)
- [Step 4: Add a new file storage component to your studio](#)
- [Step 5: Update launch profiles with the new file storage component](#)

Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.

Step 1: Create a new security group

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Security Groups** from the left navigation pane.
3. Choose **Create security group**.
4. Provide the following information in the specified fields.
 - a. **Security group name:** Enter your studio's name, using the following format: <your-studio-name>_DeadlineRepoFilesystem_EFS_SG
 - b. **Description:** Enter **Connected to Deadline Repository EFS**
 - c. **VPC:** Remove the default Amazon VPC (Amazon VPC) by choosing the VPC that you created for your studio.

- Your studio's VPC is called <your-studio-name>
- d. Keep all default inbound and outbound rules.
- e. **Optional:** Choose **Add new tag** and enter **Name** as the key and <your-studio-name>_DeadlineRepoFilesystem_EFS_SG as the value.
 - This makes it easier to find the security group in the console.

5. Choose **Create security group**.

The screenshot shows the 'Create security group' wizard. It has three main sections: 'Basic details', 'Outbound rules', and 'Tags - optional'.
- In 'Basic details':

- 'Security group name' field contains '<your-studio-name>_DeadlineRepoFilesystem_EFS_SG'.
- 'Description' field contains 'Connected to Deadline Repository EFS'.
- 'VPC' dropdown shows 'your-studio-name' selected.
- A note below says 'This security group has no inbound rules.'
- A 'Add rule' button is present.

- In 'Outbound rules':

- A note says 'This security group has no outbound rules.'
- A 'Add rule' button is present.

- In 'Tags - optional':

- A note says 'A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.'
- A table shows one tag:

Key	Name	Value - optional	<your-studio-name>_DeadlineRepoFilesystem_EFS_SG	X	Remove
Add new Tag					
- A note at the bottom says 'You can add up to 48 more tag'.

At the bottom right are 'Cancel' and 'Create security group' buttons.

Step 2: Update the security group for the Repository file system

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Security Groups** from the left navigation pane.
3. Select **Deadline EFS File System** from the list of security groups.

Security Groups (1/20) Info							
Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules count
Render Worker Access	[REDACTED]	[REDACTED]	[REDACTED]	Controls Nimble Studi...	[REDACTED]	1 Permission entry	1 Permission entry
-	[REDACTED]	[REDACTED]	[REDACTED]	launch-wizard-1 create...	[REDACTED]	1 Permission entry	1 Permission entry
-	[REDACTED]	[REDACTED]	[REDACTED]	AWS created security g...	[REDACTED]	18 Permission entries	1 Permission entry
Deadline EFS File System	[REDACTED]	[REDACTED]	[REDACTED]	Controls access to EFS ...	[REDACTED]	3 Permission entries	1 Permission entry
FSx File Systems	[REDACTED]	[REDACTED]	[REDACTED]	Controls access to the ...	[REDACTED]	12 Permission entries	1 Permission entry
Workstation access to Dea...	[REDACTED]	[REDACTED]	[REDACTED]	Controls Nimble Studi...	[REDACTED]	0 Permission entries	4 Permission entries
-	[REDACTED]	[REDACTED]	[REDACTED]	Automatic security gro...	[REDACTED]	0 Permission entries	1 Permission entry
Workstation access to file ...	[REDACTED]	[REDACTED]	[REDACTED]	Controls Nimble Studi...	[REDACTED]	0 Permission entries	1 Permission entry
Deadline Repository Instal...	[REDACTED]	[REDACTED]	[REDACTED]	Controls access to the ...	[REDACTED]	0 Permission entries	7 Permission entries

4. Choose the **Inbound rules** tab.
5. Choose **Edit inbound rules**.
6. Choose **Add rule**.
7. Provide the following details in the specified fields.
 - a. **Type:** Enter NFS
 - b. **Source:** Enter the security group that you created in [Step 1: Create a new security group](#) using this format: <your-studio-name>_DeadlineRepoFilesystem_EFS_SG
 - c. **(Optional) Description:** Enter a description, such as Allow workstations to access Deadline Repository file system.
8. Choose **Save rules**.

The screenshot shows the 'Inbound rules' tab of the AWS VPC Security Group configuration. The 'Source' dropdown is set to the security group created in Step 1. The 'Description' field is populated with the required text. The 'Save rules' button is prominently displayed at the bottom right of the form.

Step 3: Update the Backend network ACL

1. Sign in to the AWS Management Console and open the [Amazon VPC](#) console.
2. Select **Network ACLs** from the left navigation pane.
3. Select the Network ACL named **Backend**.

Network ACLs (1/10) Info												
Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count	Outbound rules count	Owner					
RenderWorkers			No		9 Inbound rules	5 Outbound rules						
Public			No		7 Inbound rules	5 Outbound rules						
WorkerSupport		2 Subnets	No		4 Inbound rules	6 Outbound rules						
ActiveDirectory		2 Subnets	No		6 Inbound rules	9 Outbound rules						
ServiceEndpoints		2 Subnets	No		13 Inbound rules	7 Outbound rules						
<input checked="" type="checkbox"/> Backend		2 Subnets	No		8 Inbound rules	13 Outbound rules						
Workstations		4 Subnets	No		7 Inbound rules	5 Outbound rules						
-		-	Yes		2 Inbound rules	2 Outbound rules						
Filesystems		-	No		16 Inbound rules	9 Outbound rules						
-		-	Yes		2 Inbound rules	2 Outbound rules						

4. Choose the **Inbound rules** tab.
5. Choose **Edit inbound rules**.
6. Choose **Add new rule**.
7. Provide the following details in the specified fields.
 - a. **Rule number:** Choose a number between 22000 and 30000.
 - b. **Type:** Enter **NFS (2049)**
 - c. **Source:** Enter the IP address range for your **Workstations** subnet.
 - i. If you chose the default CIDR block for your VPC during your studio deploy, set the source to **10.0.40.0/22**.
 - ii. If you didn't choose the default CIDR block for your VPC during your studio deploy, check the CIDR block by doing the following:
 - A. Open a new browser tab and navigate to the [VPC Dashboard](#).
 - B. Choose **Subnets** from the left navigation pane.
 - C. Select the **Workstations** subnet from the list.
 - D. In the **Details** tab, look for the **IPv4 CIDR**.
8. Choose **Save changes**.
9. Choose the **Outbound rules** tab from the lower pane.
10. Choose **Edit outbound rules**.
11. Choose **Add new rule**.
12. Provide the following details in the specified fields.
 - a. **Rule number:** Choose a number between 22000 and 30000.
 - b. **Type:** Custom TCP
 - c. **Port range:** 1024 - 65535

- d. **Destination:** Use the same IP address that you used in *step 7c* of this section.
13. Choose **Save changes**.

Step 4: Add a new file storage component to your studio

1. Sign in to the AWS Management Console and open the [Amazon EFS](#) console.
2. Select the <your-studio-name>Data/RepositoryFS file system.
 - If there are two, select the first one.
3. Choose **View details**.
4. Choose the **Network tab**.
5. Notice the IP address. You will need this for *step 9h* of this section.
6. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
7. Choose **Studio resources** in the left navigation pane.
8. Choose **Add in File Storage**.
9. Provide the following information in the specified fields.
 - a. **Region:** Select the AWS Region that your studio is deployed in.
 - b. **File storage name:** Enter **DeadlineRepoFilesystem**
 - c. (Optional) **File storage description:** Give your custom component a description. For example, **Deadline Repository EFS**.
 - d. **Storage type:** Select **Custom**
 - e. **Linux mount point:** Enter **/mnt/DeadlineRepo**
 - f. **Windows mount drive:** Enter **R** or any letter.
 - The letter that you enter won't be used by your Windows workstation.
 - g. **Share:** Enter **share**
 - The share value must be present to create the component but isn't used for anything else.
 - h. **Endpoint:** IP for the existing Deadline Repo EFS file system that you found in *step 5* of this section.
 - i. **Windows system initialization script:** Enter # N/A

- You won't be using any Windows scripts because this component is purely for Linux virtual workstations. However, entering # N/A means that the Windows portion of the initialization script gets ignored.

j. **Linux system initialization script:** Enter the following text.

```
#  
# Add mount entry to fstab if it doesn't exist to ensure it's  
# remounted if the instance is rebooted  
#  
if ! grep "DeadlineRepoFilesystem" /etc/fstab > /dev/null; then  
    echo "Adding mount for /mnt/DeadlineRepo"  
    cat <<EOF >> /etc/fstab  
# Linux-Home on Amazon EFS  
$endpoint:/ /mnt/DeadlineRepo nfs4  
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 0  
0  
EOF  
fi  
  
#  
# Perform the mount  
#  
mkdir /mnt/DeadlineRepo  
mount /mnt/DeadlineRepo  
  
cat > /etc/profile.d/file-storage-$studioComponentId.sh <<ENDOFSRIPT
```

10. **Security groups:** Choose the <your-studio-name>_DeadlineRepoFilesystem_EFS_SG security group that you created in [Step 5: Update launch profiles with the new file storage component](#).

11. (Optional) Add tags if you're using tags to track your AWS resources.

12. Read the terms and conditions and if you agree:

- Select the check box next to **I understand that Nimble Studio will access my existing file storage.**

13. Choose **Save connection parameters.**

Step 5: Update launch profiles with the new file storage component

Be careful about which launch profiles you add the storage studio component to. Users with this studio component attached to their launch profile can get access to project data that they otherwise don't have access to.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profile that you want to update.
4. Choose **Action**. Then choose **Edit**.
5. In the **Launch profile components** section, choose the check box next to the **DeadlineRepoFilesystem** component.
6. Navigate to the bottom and choose **Update launch profile**.
7. Repeat these steps for all of the launch profiles that you want to access the **DeadlineRepoFilesystem** custom component that you created.

Studio users can now launch a Linux workstation and access the Deadline Repository file system by opening a terminal and navigating to /mnt/DeadlineRepo.

Creating your first render on the farm

This tutorial will show you how to use your studio's render farm to create your first rendered images on Amazon Nimble Studio. Since Blender is installed by default on Nimble Studio virtual workstations, this tutorial shows you how to use Blender as the content creation application.

The steps for other applications are similar to Blender, but licensed applications might require your studio administrator to set up a license server before those applications can be used on virtual workstations and render workers. In addition, some software might need your administrator to follow specific steps after installation in order to work properly.

Tutorials for software that requires specific installation instructions can be found in the [Software specific installation tips](#) tutorial.

Contents

- [Prerequisites](#)
- [Step 1: Launch Blender](#)

- [Step 2: Create a Blender scene](#)
- [Step 3: Save your scene to shared storage](#)
- [Step 4: Configure the scene to render on the farm](#)
- [Step 5: Enable AWSThinkboxDeadline submitter add-on](#)
- [Step 6: Submit render to Deadline](#)
- [Step 7: Check progress in Deadline Monitor](#)
- [Related resources](#)

Prerequisites

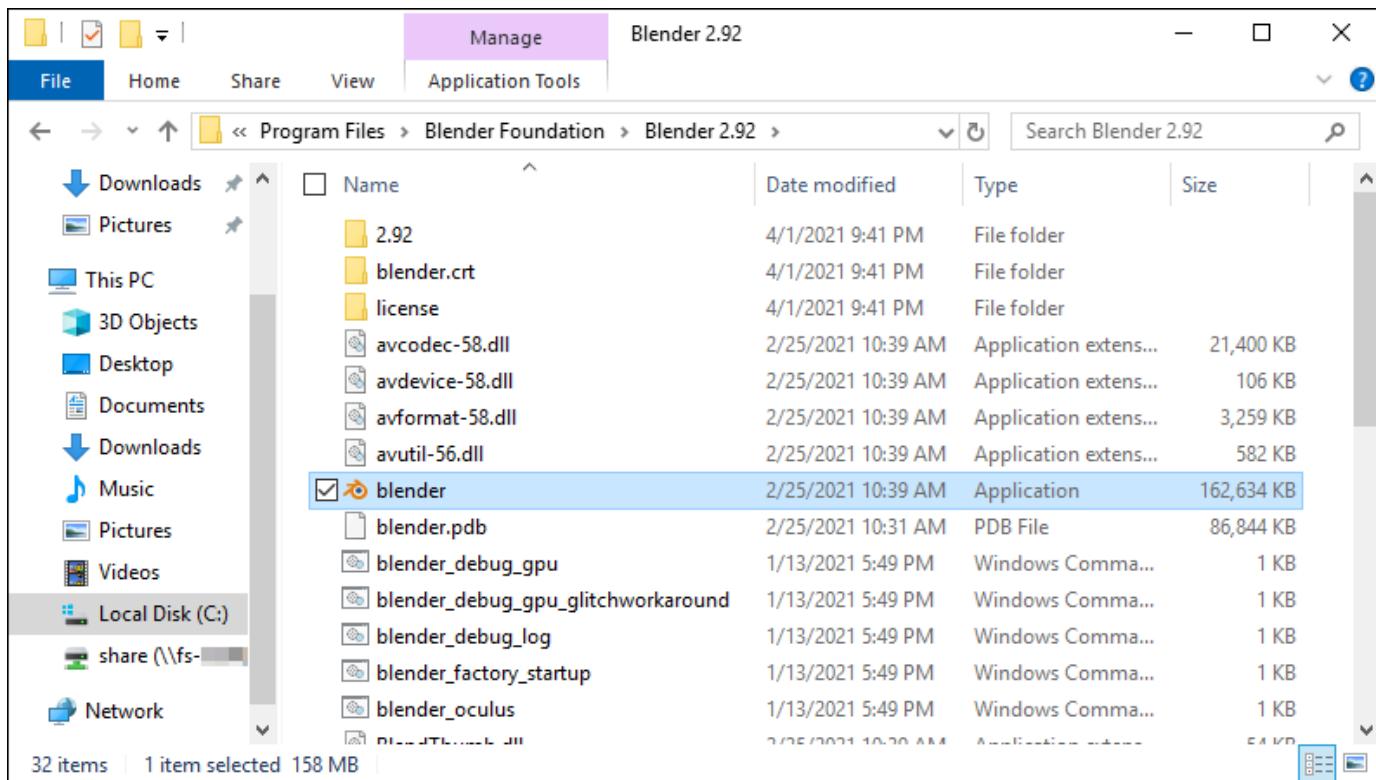
- Launch a streaming session and be logged in to a virtual workstation, as outlined in [Launching a virtual workstation](#).
- A basic working knowledge of Blender is also required to complete this tutorial. If you haven't used Blender before, you can find links to documentation and tutorials on the [Blender Support](#) webpage.
- In addition, your studio administrator must have completed the steps in the [Configuring AWSThinkboxDeadline](#) tutorial in order for you to successfully render your scene.

Step 1: Launch Blender

You'll start by launching Blender, so that you can create a scene to render.

To launch Blender using Windows

1. Open **File Explorer**.
2. Choose **This PC** in the navigation pane.
3. Open (double-click) **Local Disk (C:)** in **Devices and drives**.
4. Navigate to **C:\Program Files\Blender Foundation\Blender 2.92**. Your version of Blender might be different.
5. Open (double-click) the Blender application file.



To launch Blender using Linux

1. Launch a file browser by opening (double-clicking) the **Home** folder on the desktop.
2. In the left navigation pane, choose **+ Other Locations**.
3. Choose **Computer**.
4. Navigate to **/opt/blender-2.92.0-linux64**. Your version of Blender might be different.
5. Open (double-click) the Blender application file.

Step 2: Create a Blender scene

Now that you're in Blender, you need a scene to render. You can use the default scene, create your own, or download a demo scene. This tutorial makes use of a demo scene, but the steps for rendering will be the same no matter what kind of scene you use.

- If you use the default scene or create your own scene, you can skip to [Step 3: Save your scene to shared storage](#).
- To download a demo scene instead, see the following procedure.

To download a demo scene

1. Launch **Mozilla Firefox** on your virtual workstation.
2. Download a demo file of your choosing from the [Blender download](#) page.
 - a. This tutorial uses one of the files from the **Physics** section.
 - b. When prompted to open the file or save it, choose **Save File**.
3. Open a file browser (Windows or Linux) and navigate to your **Downloads** folder.
 - a. **Windows:** C:\Users\ **your-user-name** \Downloads.
 - b. **Linux:** \$Home/Downloads.
4. Select the Blender scene file from the file browser and enter **Ctrl+C** to copy it.

Step 3: Save your scene to shared storage

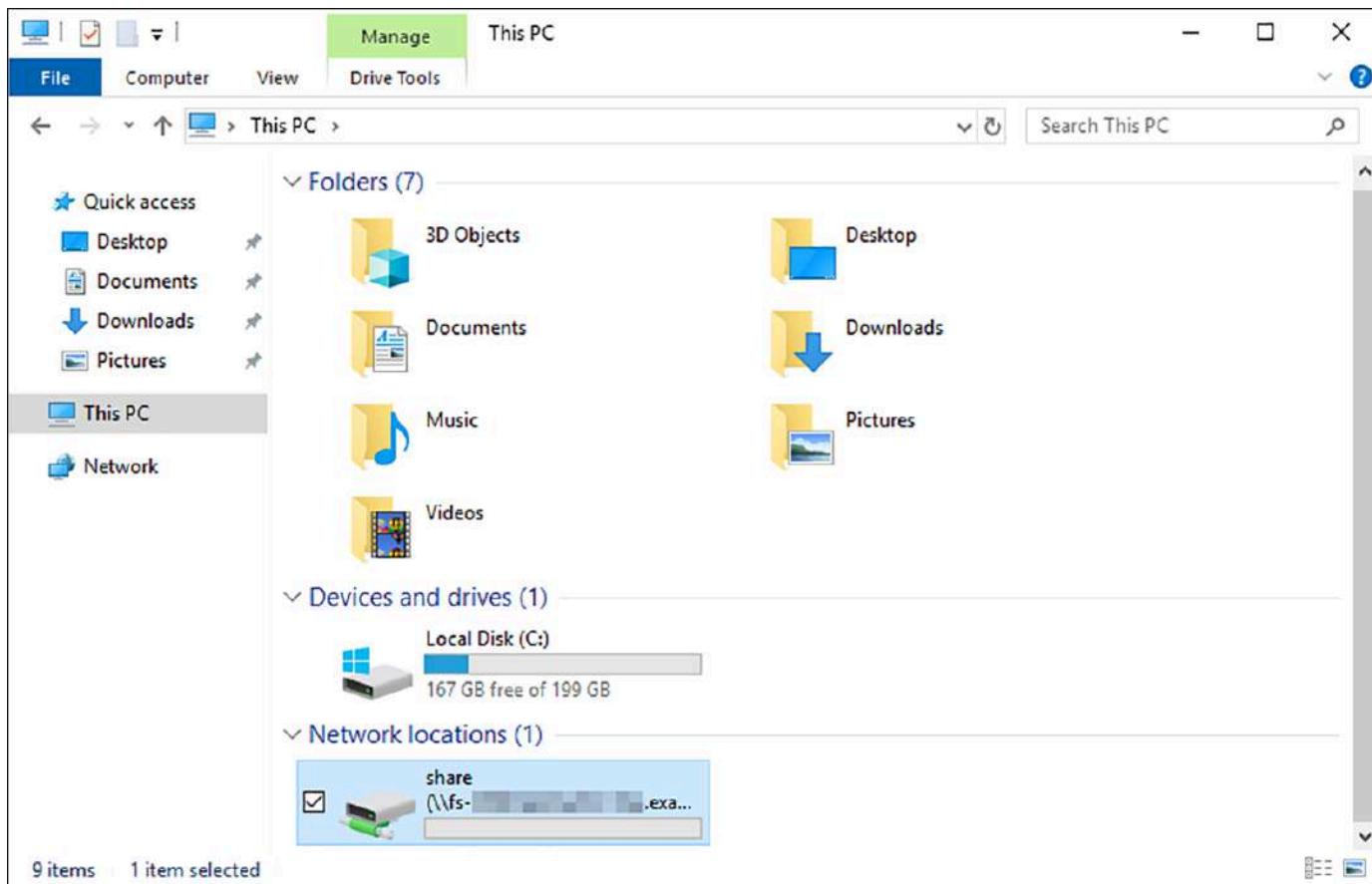
In this step, you will save your Blender scene file to the default shared storage of your studio. We recommend saving your scene file to this shared storage so it's more easily accessible. For more information about file storage, see the [Setting up an Amazon FSx Windows file system](#) tutorial.

To save your Blender scene file to the default shared storage of your studio, use the following steps.

Find the shared storage drive on your virtual workstation

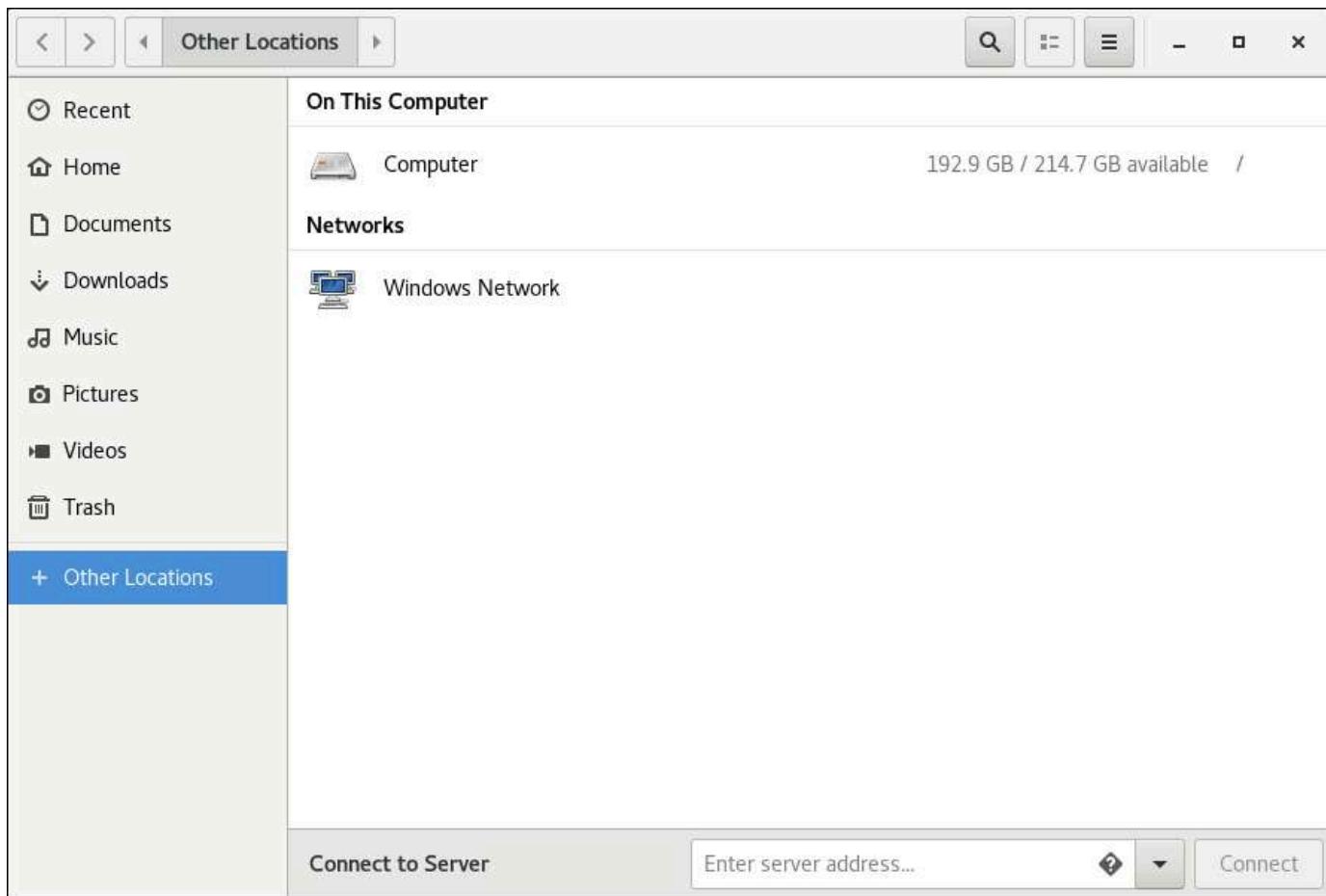
To find shared storage on Windows

1. In a new **File Explorer** window, choose **This PC** from the navigation pane.
2. Open (double-click) **share** in **Network locations**. This is also referred to as the **Z:** drive.



To find shared storage on Linux

1. In a file browser, choose **+ Other Locations** from the left navigation pane.
2. Choose **Computer**.

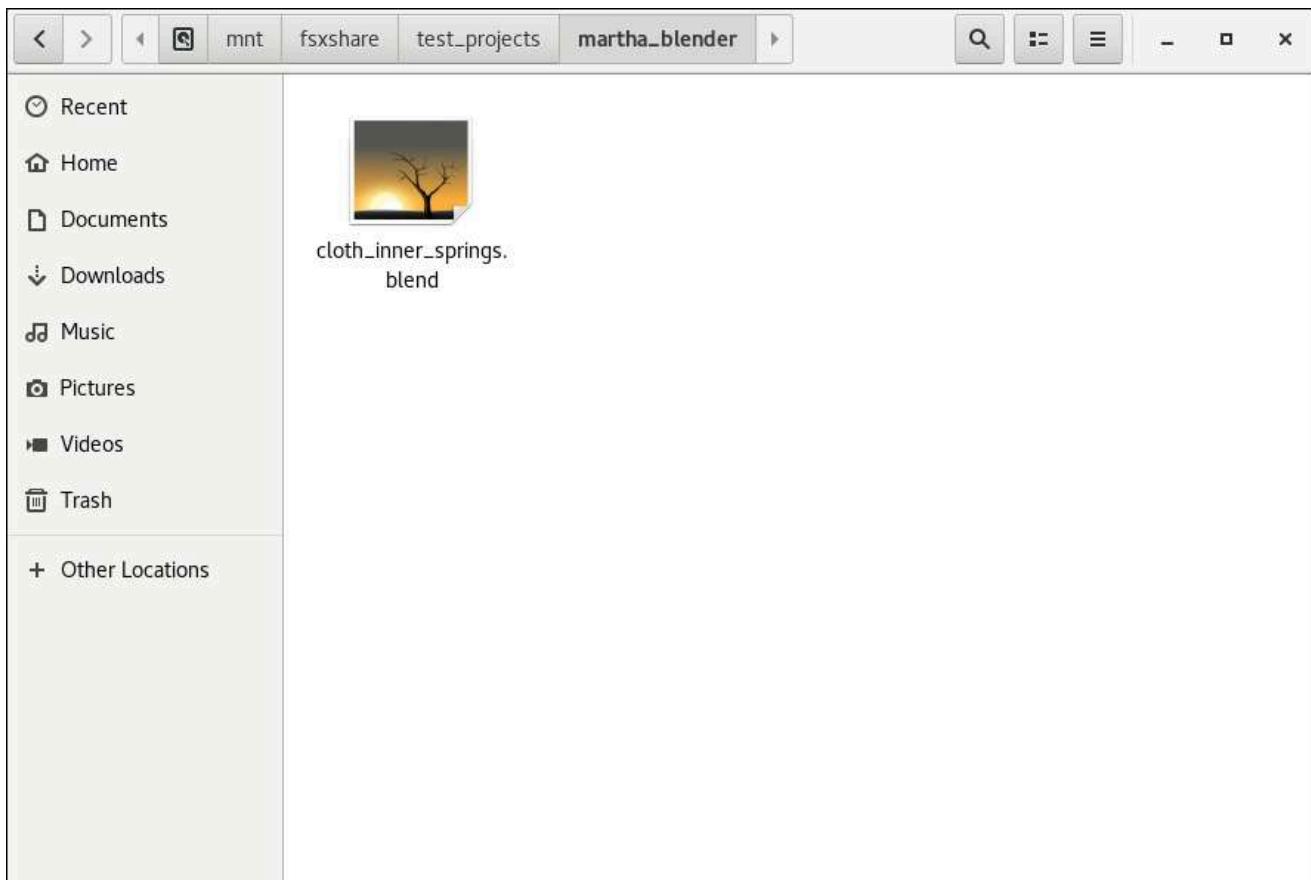


3. Navigate to `/mnt/fsxshare`.

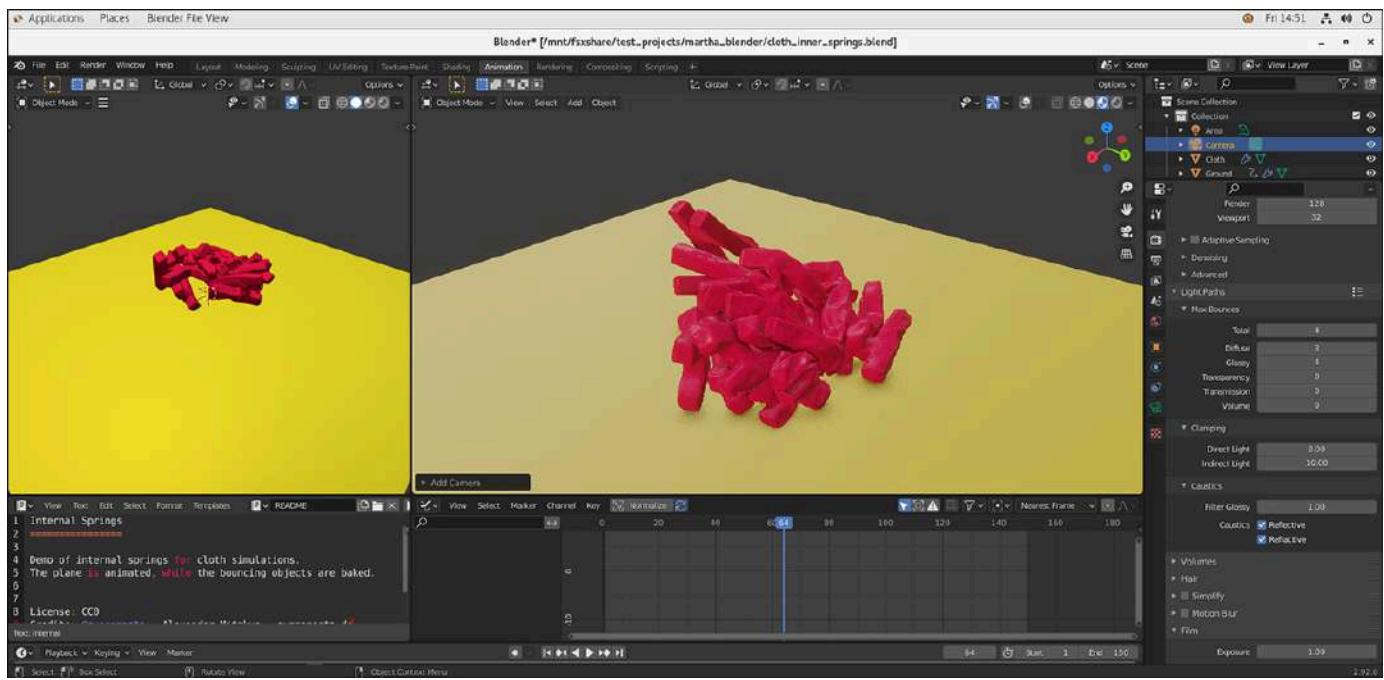
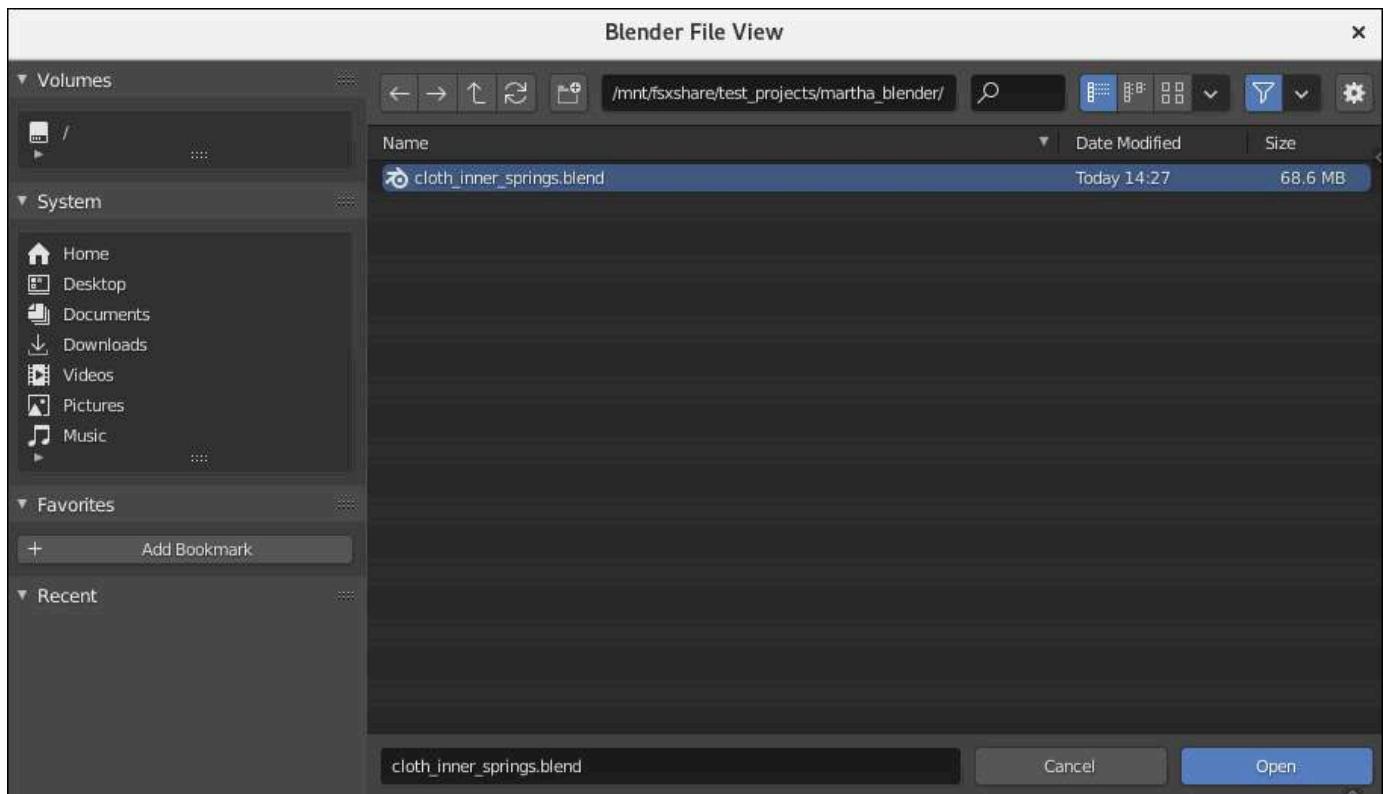
Save your Blender file to the shared storage drive

1. If there is already a folder to store test projects, open it. If not, create a new folder by opening the context menu (right-click) and choosing **New and Folder** (Windows) or **New Folder** (Linux).
 - Name your new folder something that identifies it as a place for test projects, for example: `test_projects`.
2. In the test projects folder, open the context menu (right-click) and choose **New and Folder** (or **New Folder**) again.
3. Name the new folder something that identifies it as your folder specifically, for example: `your-user-name _blender`.
4. Move your Blender file to the current folder.

- a. If you downloaded a demo file in the last step, paste it or move it from your Downloads folder to this location.
- b. If you created your own file and saved it somewhere else, you should copy and paste it (or move it) to this location.
- c. The following is an example of directory structure in a shared storage drive. The structure contains a Blender demo file in the interface's center screen. The side navigation includes links to Home, Recent (files), Downloads, Documents, and various media.



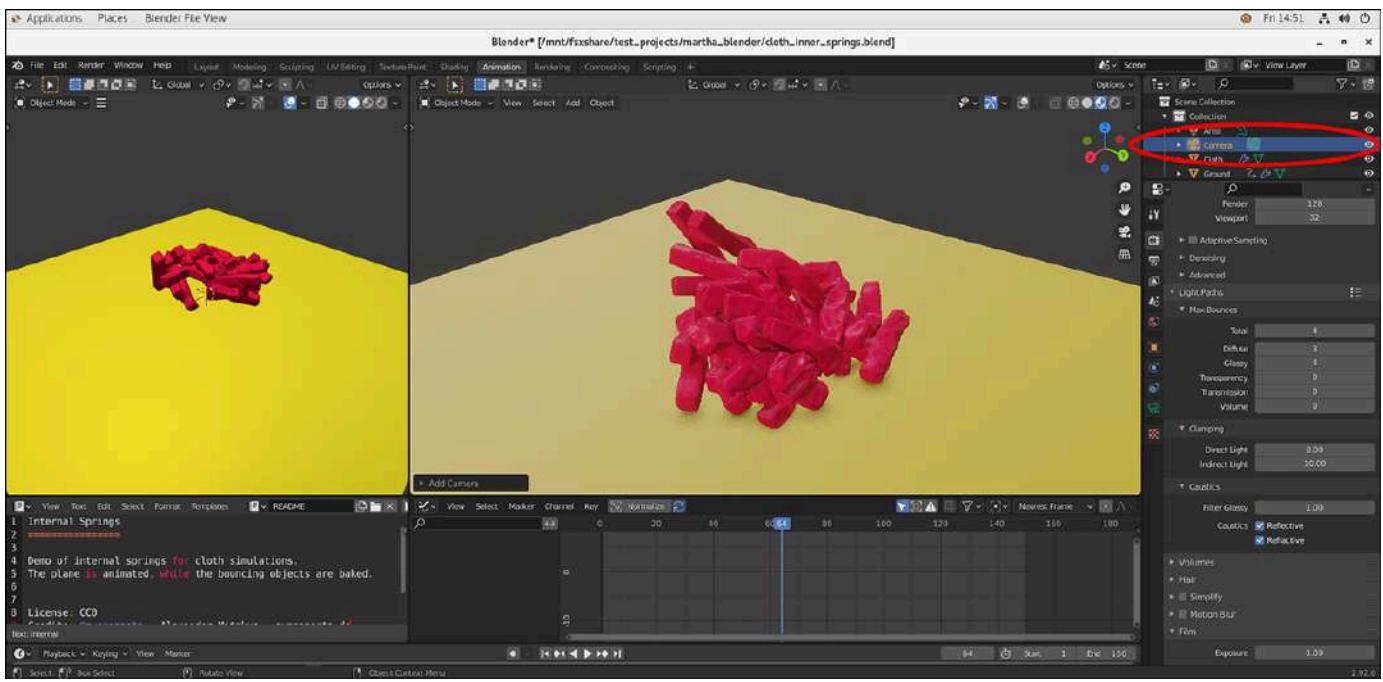
5. In Blender, open the file on the shared storage drive by choosing **File** and **Open...** and navigating to the location of your demo file. For example, Z:\test_projects\martha_blender (Windows) or /mnt/fsxshare/test_projects/martha_blender (Linux).



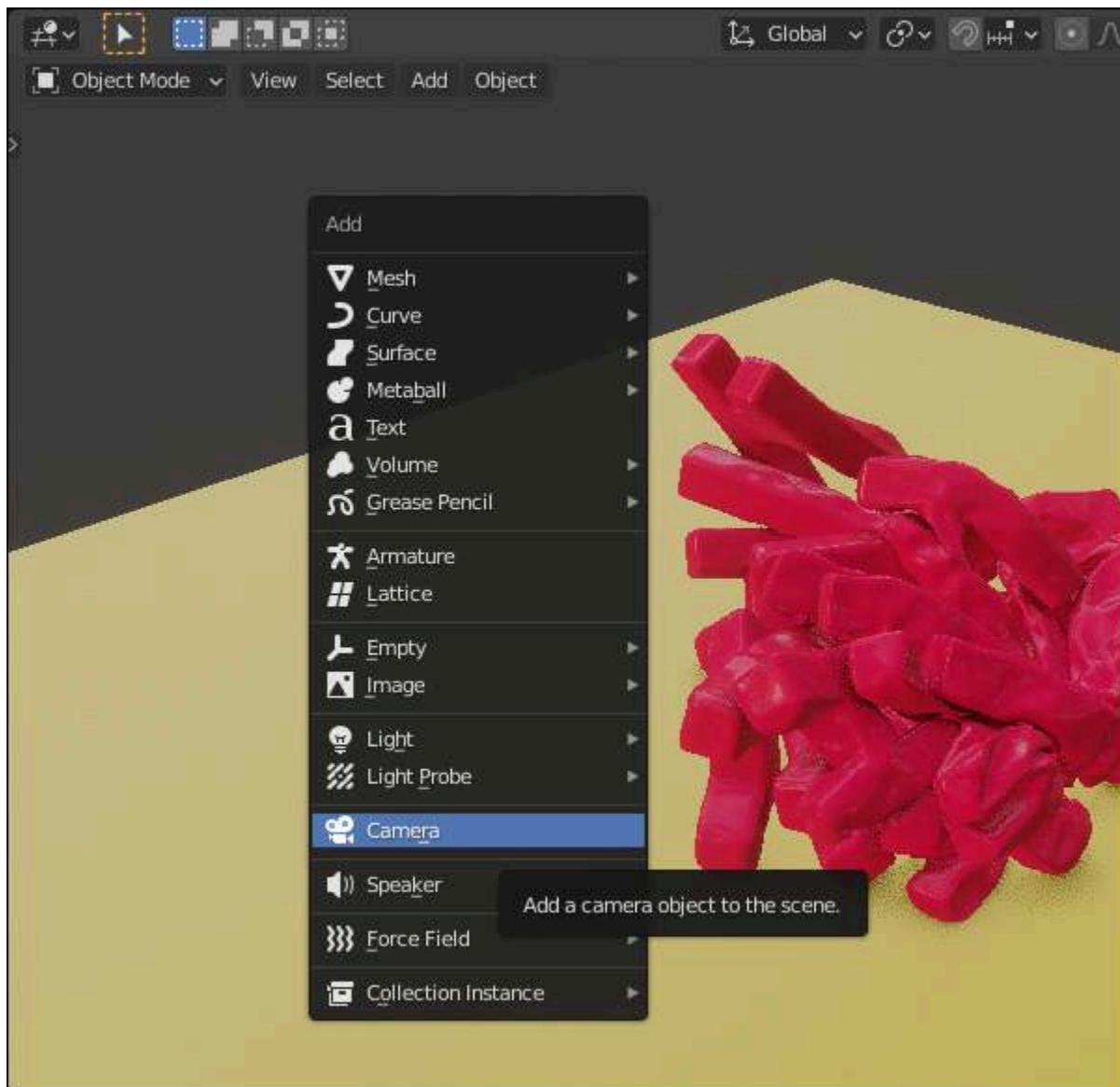
Step 4: Configure the scene to render on the farm

When you have a scene that you want to render, you must configure the scene so that it will render on the farm.

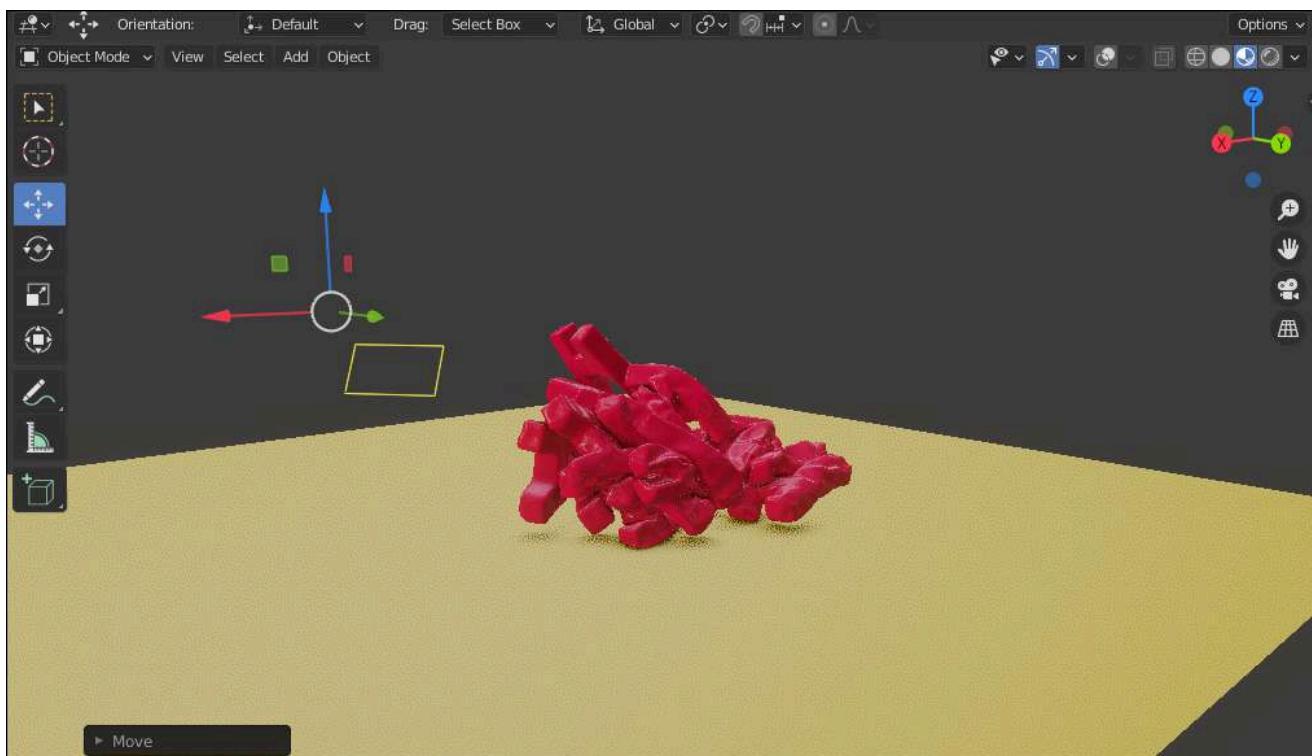
- Verify that a camera is listed in the Outliner. You must have a camera in your Blender scene in order to render it.



- If you don't see a camera in the Outliner, press **Shift+A** in a viewport to add one.



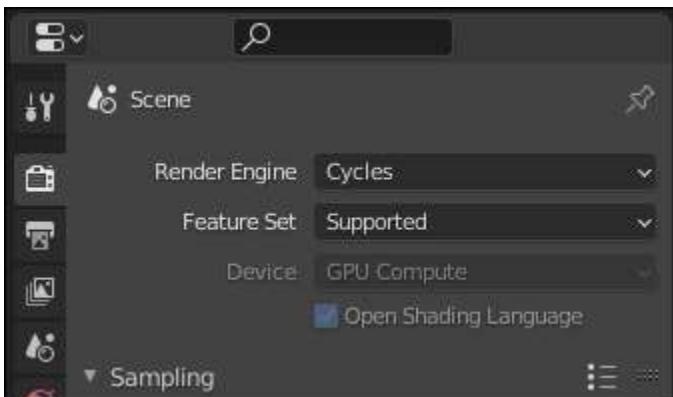
- b. Press **T** in a viewport to show the tools.
- c. Select a transform tool and move your camera into position.



2. In the **Properties** panel on the right, choose **Render Properties** (the icon looks like the back of an SLR camera).



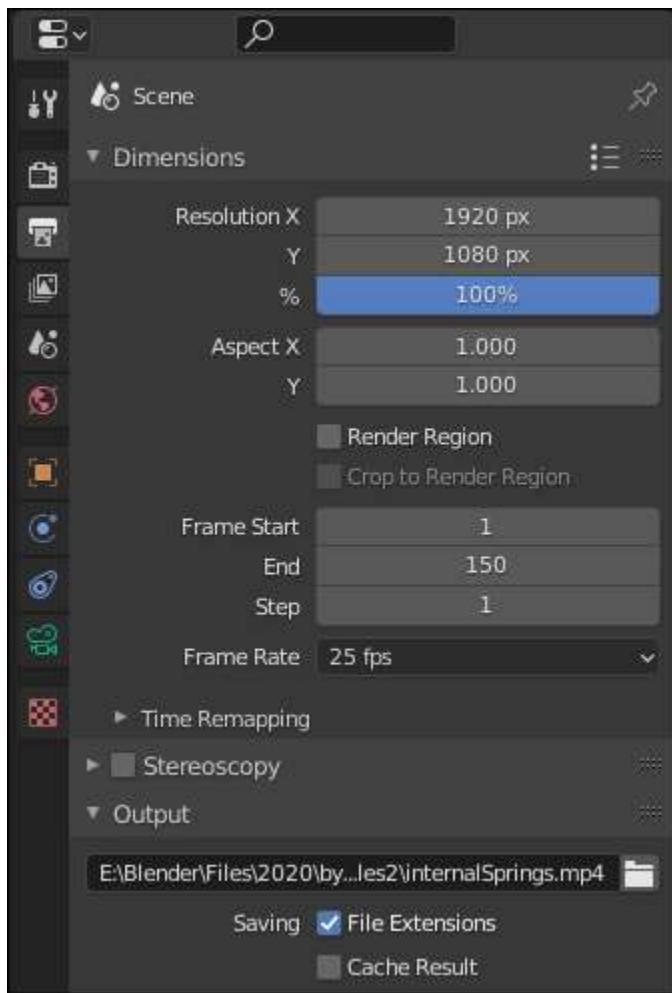
3. For **Render Engine**, choose **Cycles**.



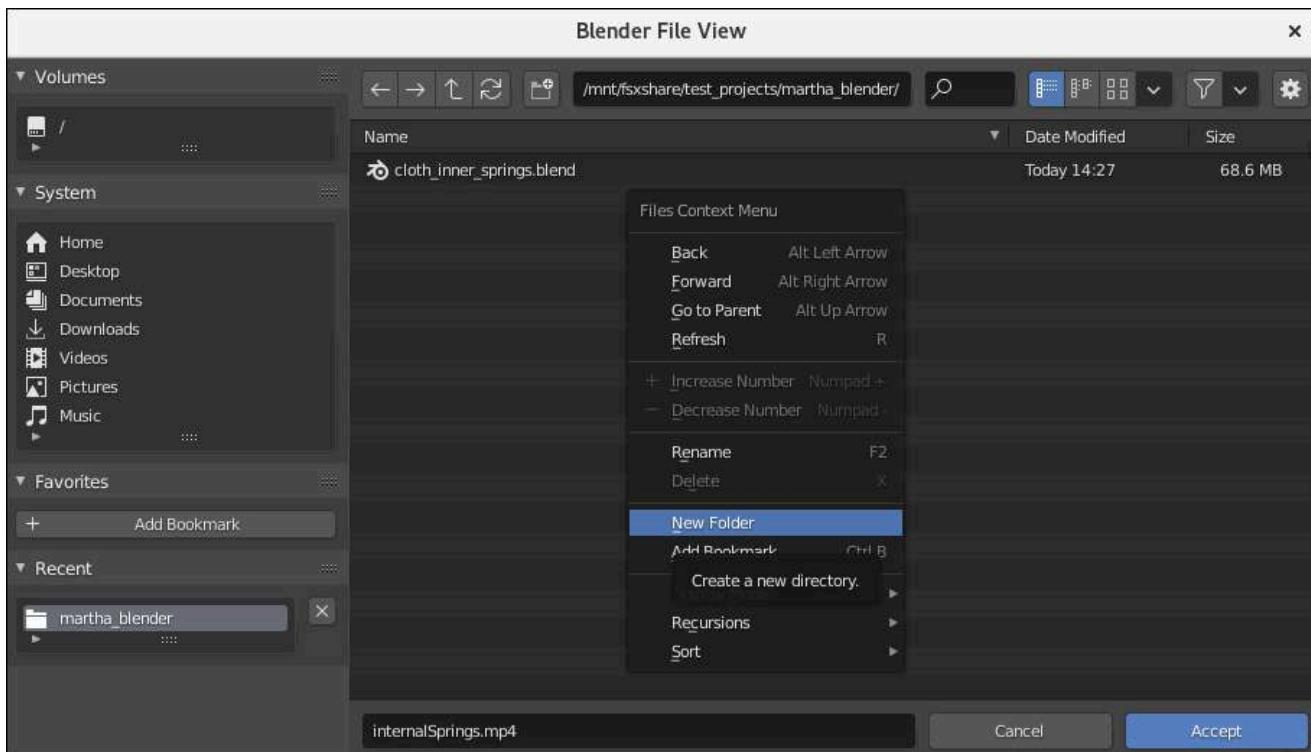
4. Choose **Output Properties** (the icon looks like an inkjet printer).



5. Set your output path in **Output**.



- a. Choose the **folder icon** next to the output field.
- b. In the window that appears, navigate to the location of your Blender file. For example, Z:\\test_projects\\martha_blender (Windows) or /mnt/fsxshare/test_projects/martha_blender (Linux).
- c. Create a new folder by opening the context menu (right-click) in the file list and choosing **New Folder**.

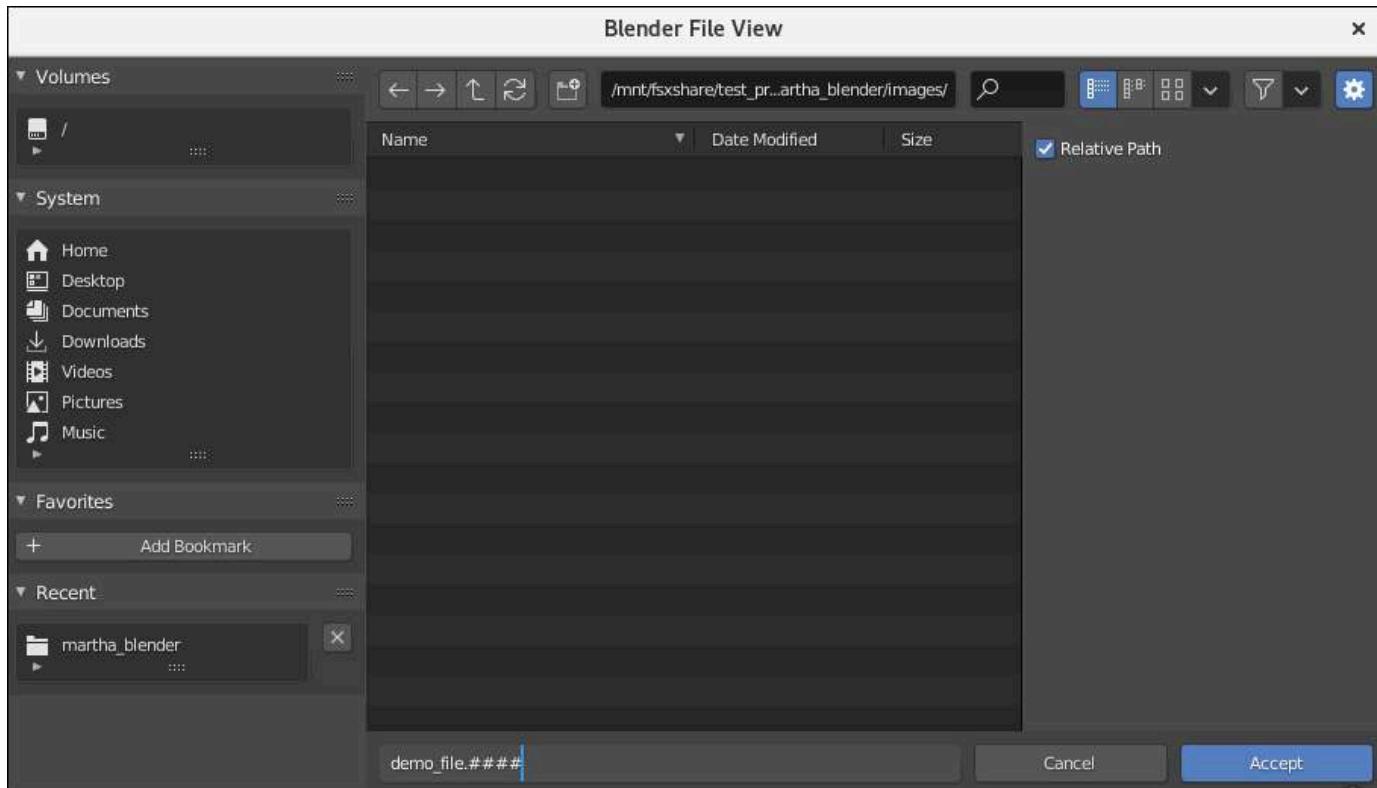


- d. Name the new folder **images** and open it.
- e. Choose the **gear icon** in the top-right corner of the window.
- f. Choose the check box next to **Relative Path**.
- g. Enter a **name** for the output files in the entry field at the bottom of the screen. Use the hash sign <#> notation in your name to indicate where you would like frame numbers inserted into the output file names.

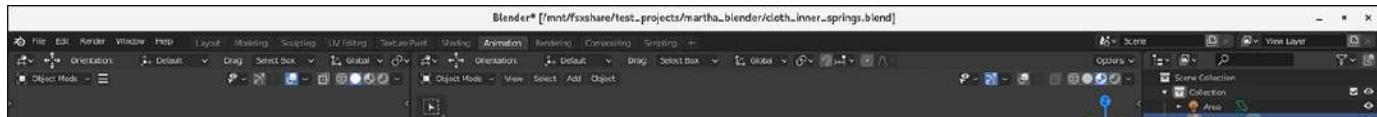
For example:

demo_file.####

6. Choose **Accept**.



7. Return to the **Output Properties** panel. Use the **File Format dropdown** to select the format for your output images. We recommend choosing the **OpenEXR** format.
8. Verify that your scene file is located on shared storage and not the local drive of your virtual workstation by checking the scene file path in the title bar at the top of the main Blender window. For example, Z:\test_projects\martha_blender\<file_name> (Windows) or /mnt/fsxshare/test_projects/martha_blender/<file_name> (Linux).

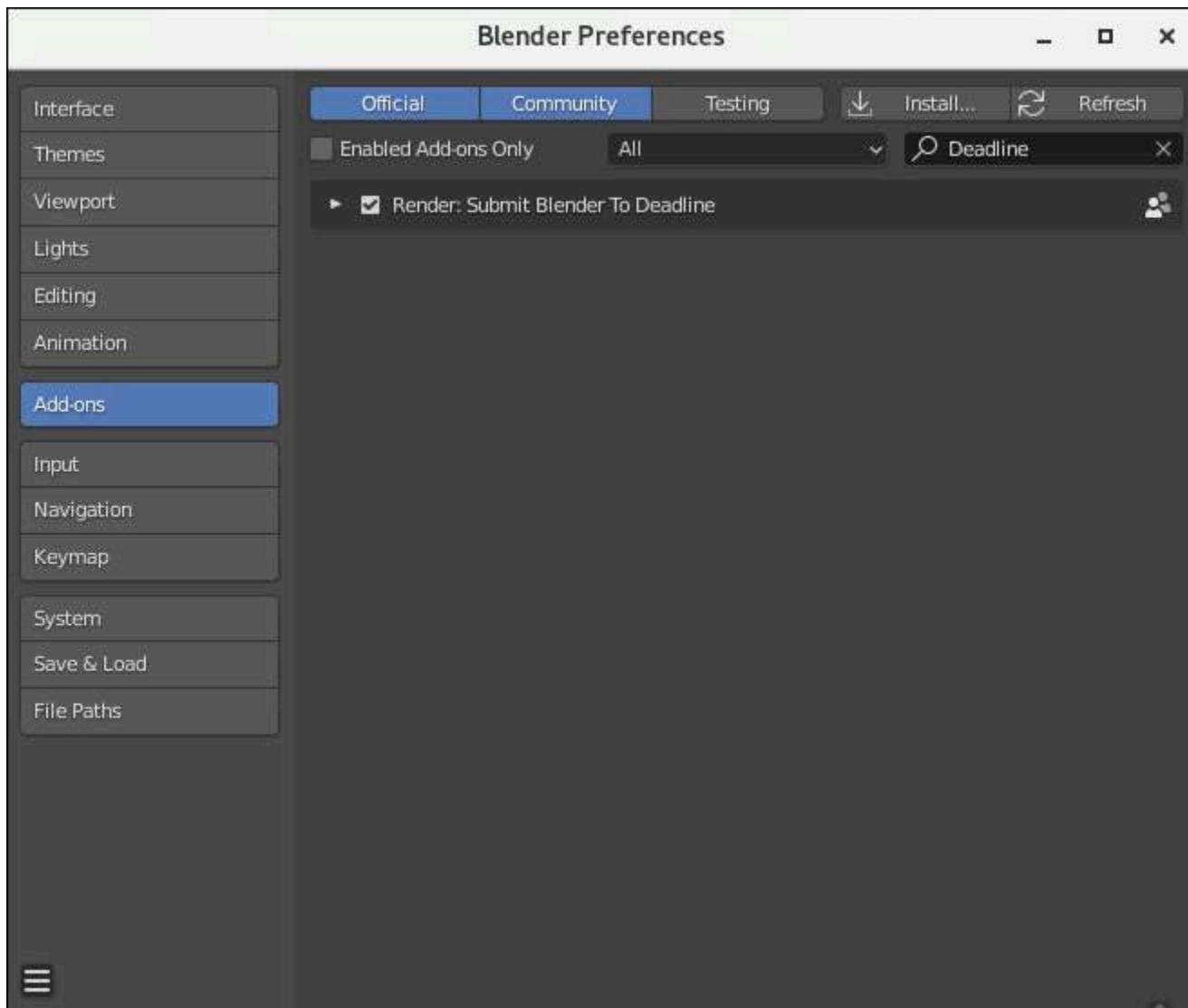


9. Save all the changes that you made to your Blender scene file by choosing **File** and **Save** from the main Blender menu.

Step 5: Enable AWSThinkboxDeadline submitter add-on

To submit your scene file to the render farm, you must enable the Blender submitter add-on for AWS Thinkbox Deadline. Deadline is a compute management toolkit that handles the running of your render farm on Nimble Studio. The submitter add-on allows you to submit a render to your farm directly from within Blender.

1. Enter **Deadline** in the search field to search for the Deadline submitter.
2. Select the check box next to **Render: Submit Blender to Deadline** to enable the submitter add-on and then close the preferences window.

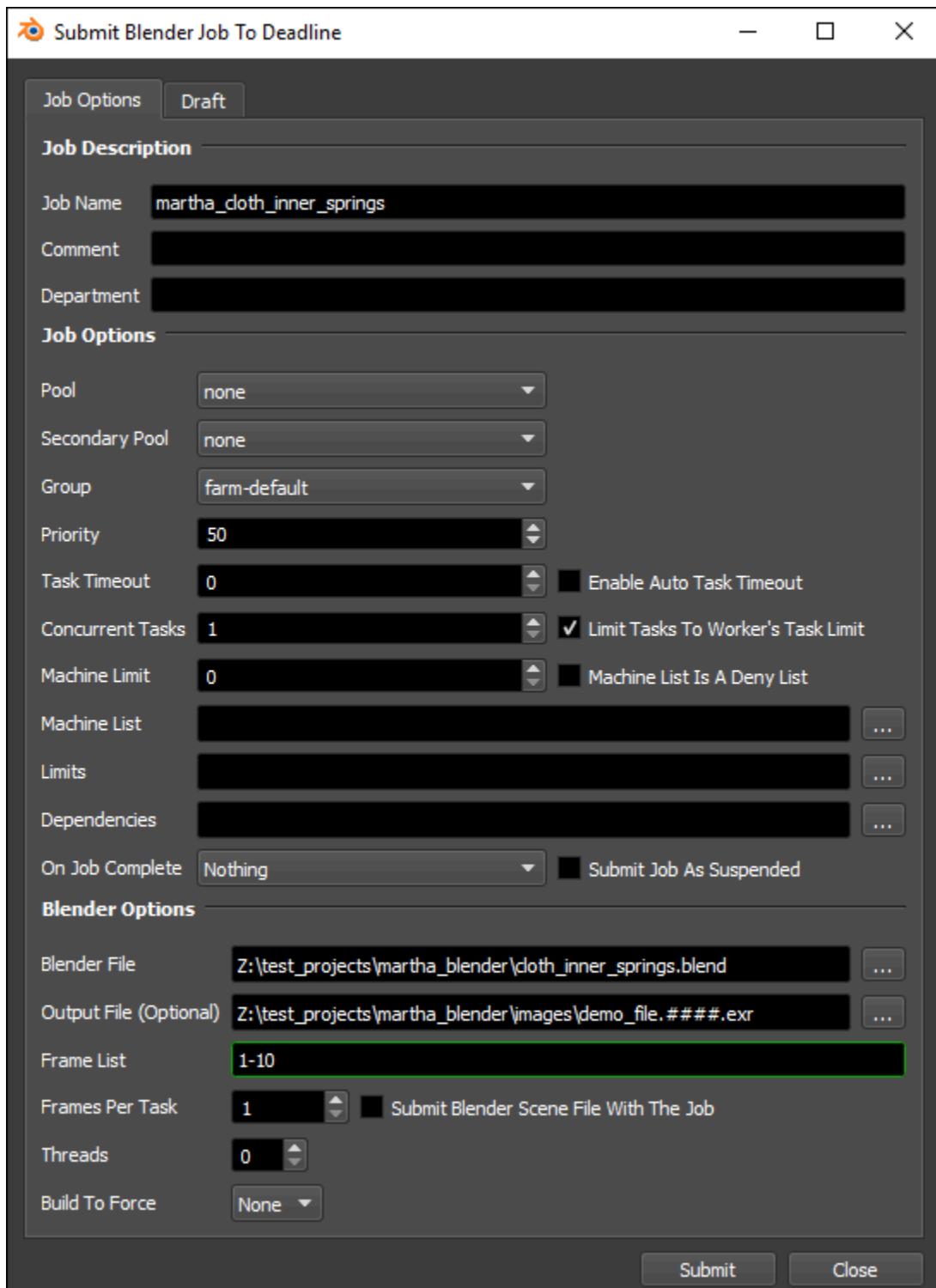


Step 6: Submit render to Deadline

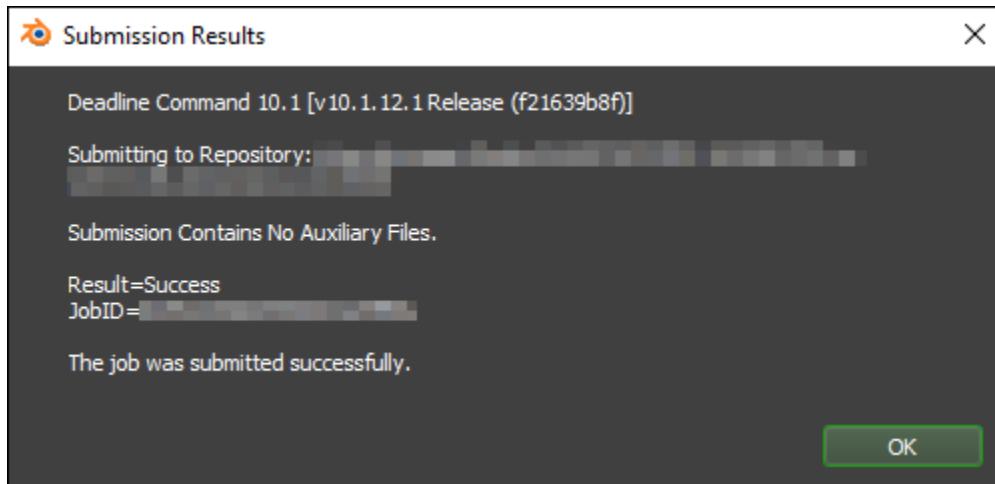
Now that the Deadline submitter add-on is installed, you're ready to submit your render to the farm.

1. In Blender, open the **Render** menu and choose **Submit To Deadline**.
 - This will take a minute to open, because Blender has to connect to Deadline.

2. When the Submission Scripts window opens, look at the **Job Name**. It will be set to the name of your file by default. You can add your user name to the beginning of the name to make your render job easier to identify.
3. Check that **Pool** is set to **none**.
4. Open the dropdown next to **Secondary Pool** and choose **none**.
5. Open the dropdown next to **Group** and choose the name of the render fleet that you want to use.
 - a. This might be named something like **farm-default**, but your studio administrator might have chosen a different name. If you have questions about what to choose, ask your studio administrator or manager.
 - b. If you don't see any groups listed in the pulldown, check that your studio administrator has completed the [Configuring AWSThinkboxDeadline](#) tutorial, in which they set up Deadline groups.
6. Review the value of **Frame List**. By default, this will be set to the start and end frames of your scene, but you can change this to just a few frames for your first render.



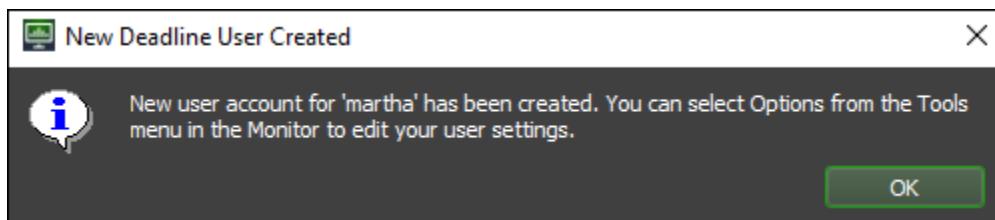
7. Check that all the values that you entered are correct and then choose **Submit**.
8. After your render has been submitted, a confirmation window will appear. Choose **OK** to close the window.



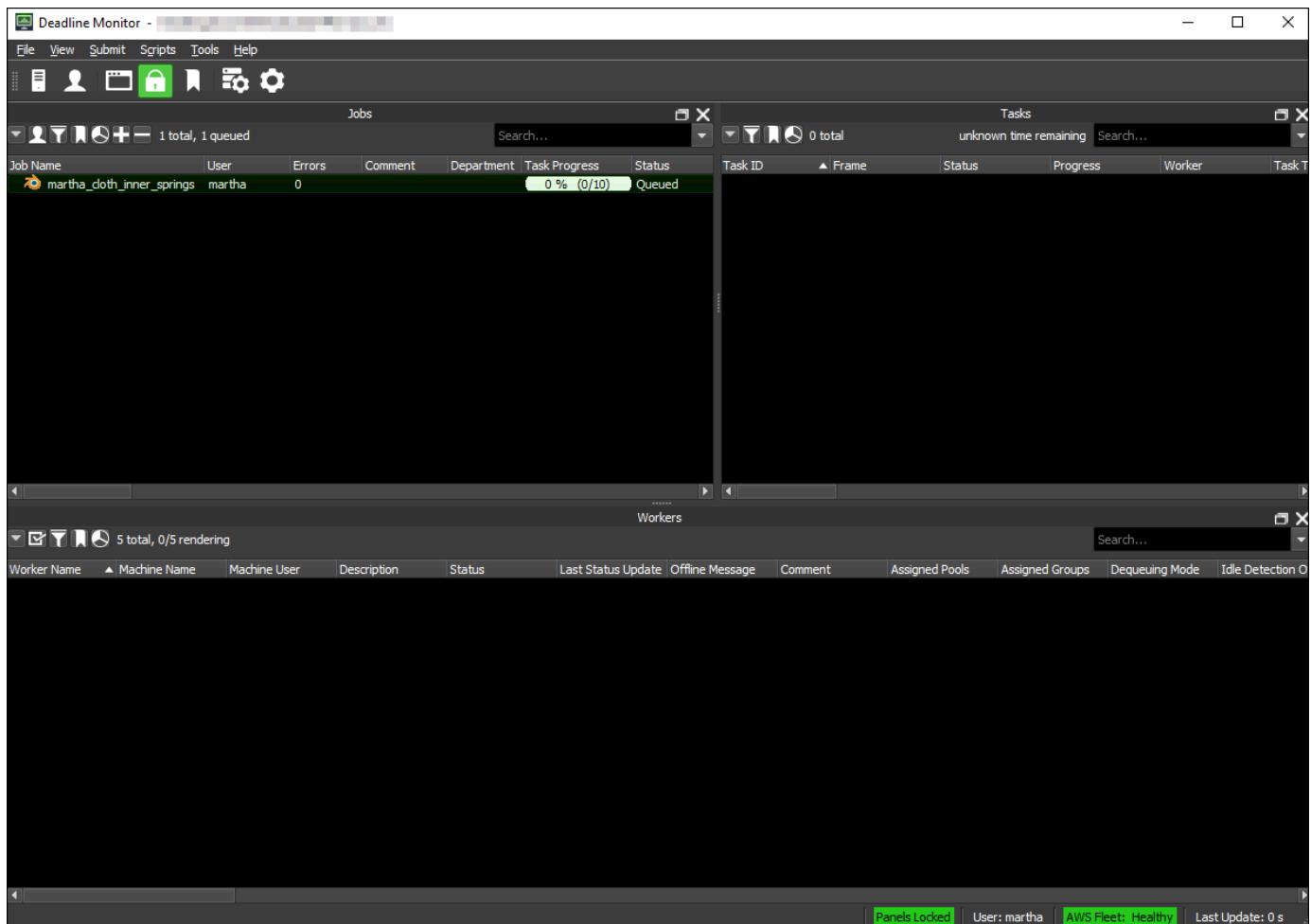
Step 7: Check progress in Deadline Monitor

After you submit your render, you can watch its progress in the Deadline Monitor.

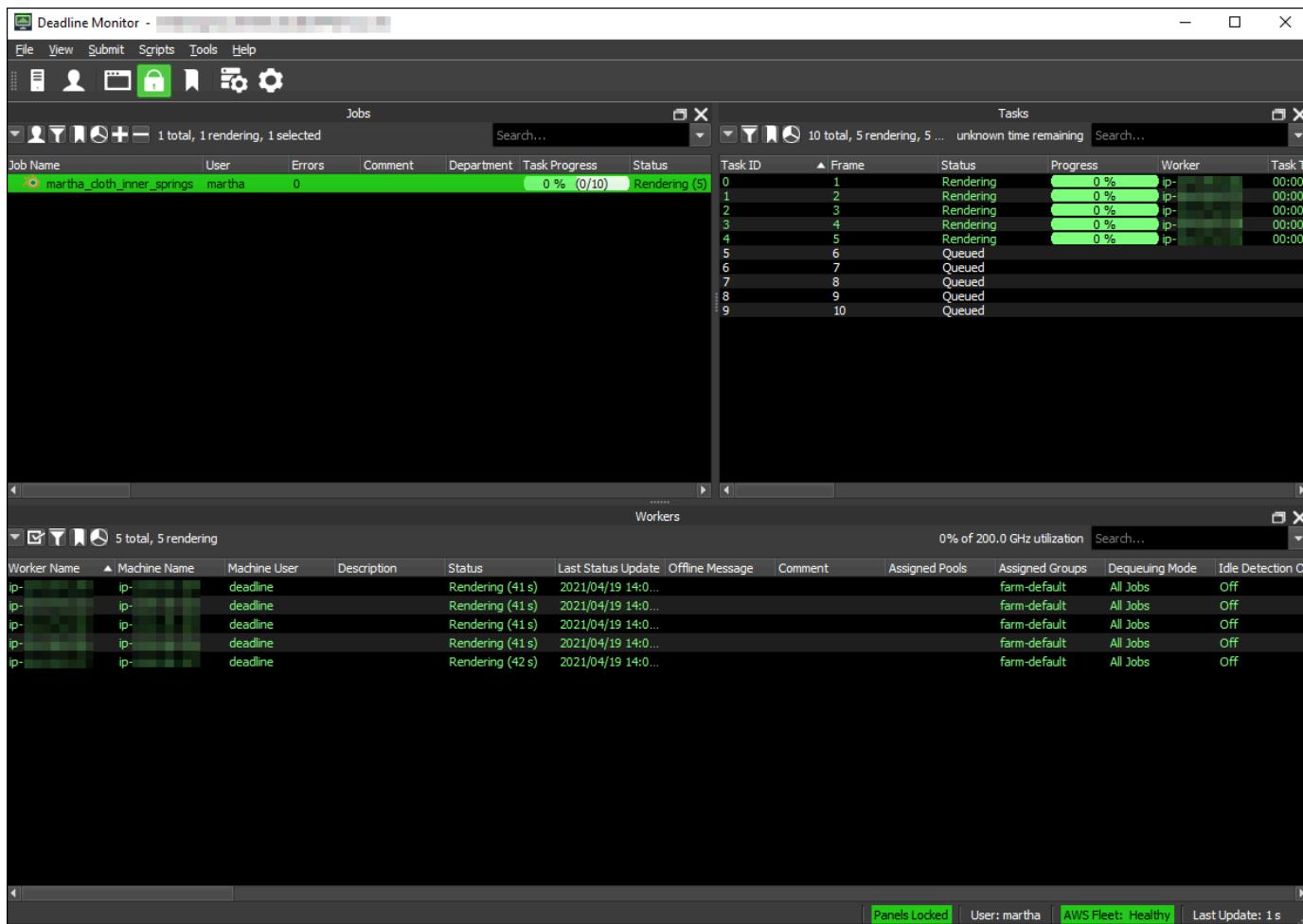
1. Launch the Deadline Monitor.
 - a. **Windows:** Open the **Start** menu and choose **Thinkbox**. Then choose **Deadline Monitor**.
 - b. **Linux:** Choose **Applications**. Then choose **Other** and **Deadline Monitor**.
2. Deadline Monitor starts opening, and a window pops up that indicates that a new account has been created for you. Choose **OK** to continue.



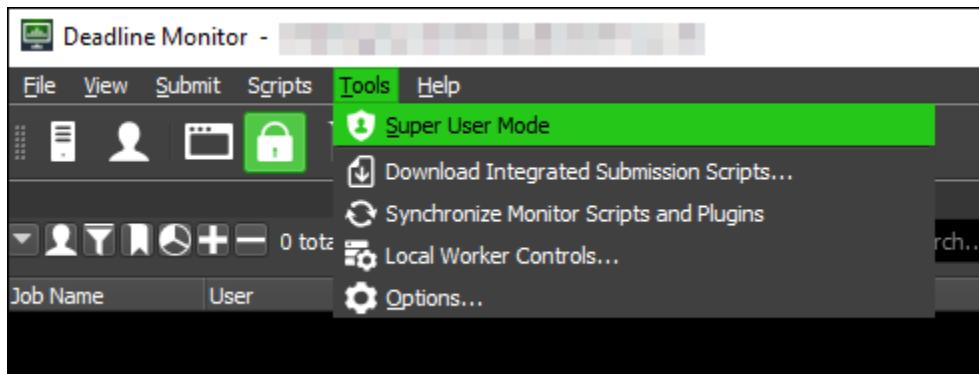
- Now the Deadline Monitor will open.
3. Your render job will appear in the list in the upper left section of the Deadline Monitor window.



- If your farm doesn't currently have any render workers running, or if they are all busy, your job will have a **Queued** status. It will take a few minutes for new render workers to launch. After they launch and pick up your job, the status will change to **Rendering**.
4. Choose your render job to see more details about all the tasks for that job. Usually this will be a list of the frames to be rendered.

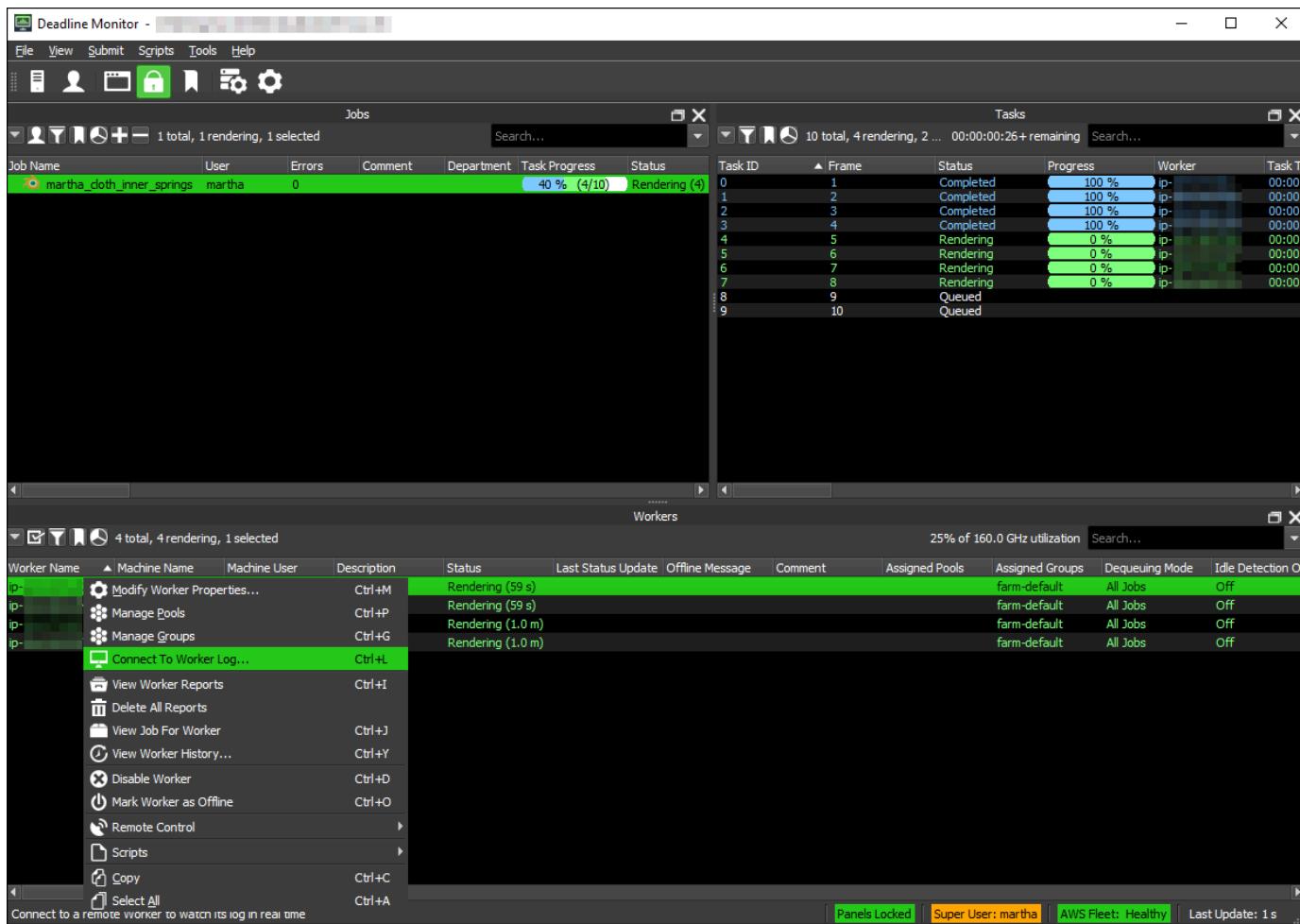


5. Choose Tools. Then choose Super User Mode.

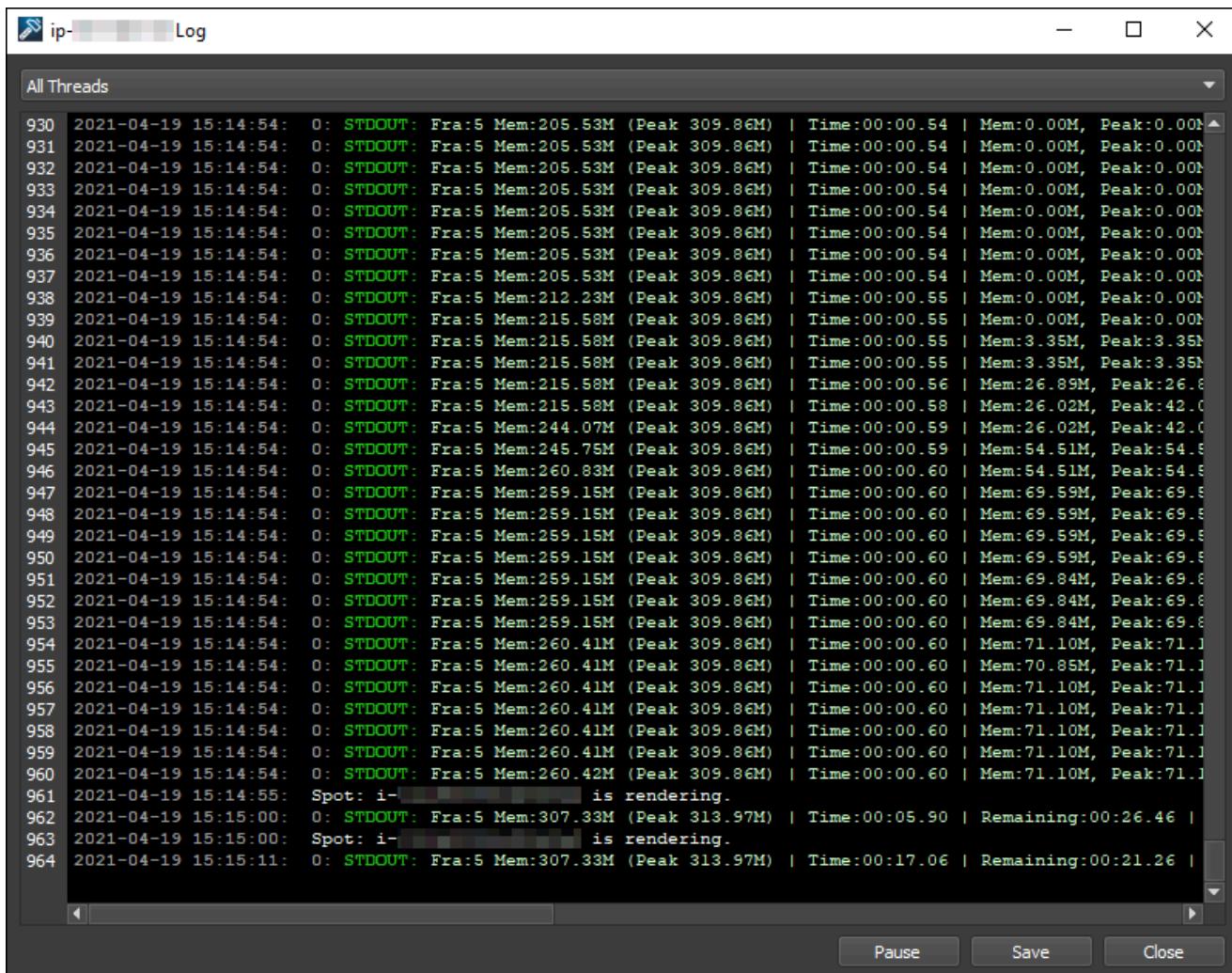


- superuser mode will allow you access to more options, like the ability to check the logs of individual render workers.

- Next, choose a render worker from the list at the bottom of the Deadline Monitor window.
- Open the context menu (right-click) and choose **Connect to Worker Log...**



- A new window will open that displays the log for the worker you connected to. Here you can monitor the status of a particular render worker in your farm.

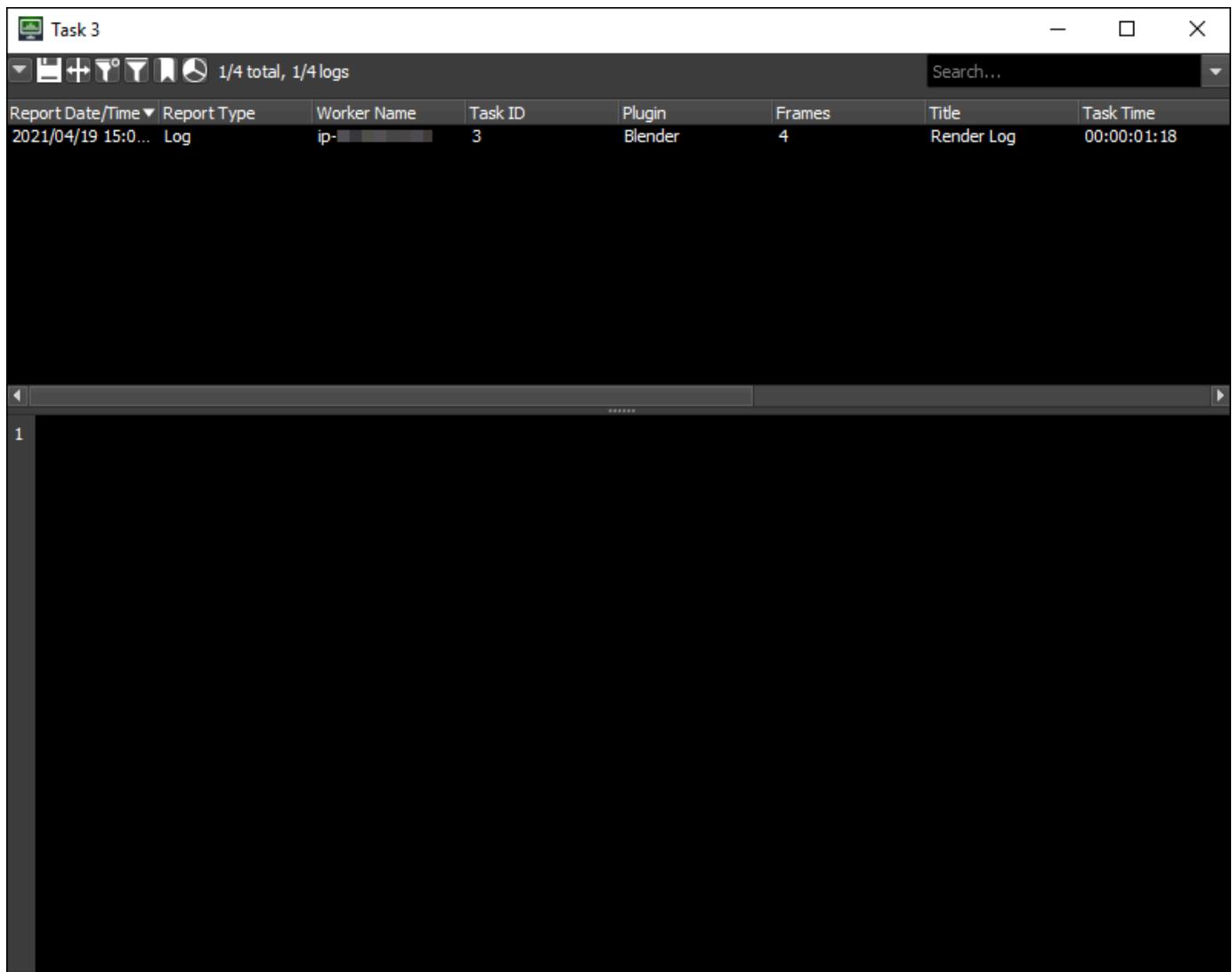


The screenshot shows a log window titled "ip-[REDACTED] Log". The window displays a list of log entries under the heading "All Threads". The log entries are timestamped and show various system and application activity. Key entries include:

- Threads 930 through 934: "STDOUT: Fra:5 Mem:205.53M (Peak 309.86M) | Time:00:00.54 | Mem:0.00M, Peak:0.00M".
- Threads 935 through 940: "STDOUT: Fra:5 Mem:205.53M (Peak 309.86M) | Time:00:00.54 | Mem:0.00M, Peak:0.00M".
- Threads 941 through 942: "STDOUT: Fra:5 Mem:215.58M (Peak 309.86M) | Time:00:00.55 | Mem:3.35M, Peak:3.35M".
- Threads 943 through 944: "STDOUT: Fra:5 Mem:215.58M (Peak 309.86M) | Time:00:00.55 | Mem:26.02M, Peak:42.0".
- Threads 945 through 946: "STDOUT: Fra:5 Mem:244.07M (Peak 309.86M) | Time:00:00.59 | Mem:54.51M, Peak:54.5".
- Threads 947 through 948: "STDOUT: Fra:5 Mem:245.75M (Peak 309.86M) | Time:00:00.59 | Mem:69.59M, Peak:69.5".
- Threads 949 through 950: "STDOUT: Fra:5 Mem:260.83M (Peak 309.86M) | Time:00:00.60 | Mem:69.84M, Peak:69.8".
- Threads 951 through 952: "STDOUT: Fra:5 Mem:259.15M (Peak 309.86M) | Time:00:00.60 | Mem:69.84M, Peak:69.8".
- Threads 953 through 954: "STDOUT: Fra:5 Mem:259.15M (Peak 309.86M) | Time:00:00.60 | Mem:71.10M, Peak:71.1".
- Threads 955 through 956: "STDOUT: Fra:5 Mem:260.41M (Peak 309.86M) | Time:00:00.60 | Mem:70.85M, Peak:71.1".
- Threads 957 through 958: "STDOUT: Fra:5 Mem:260.41M (Peak 309.86M) | Time:00:00.60 | Mem:71.10M, Peak:71.1".
- Threads 959 through 960: "STDOUT: Fra:5 Mem:260.41M (Peak 309.86M) | Time:00:00.60 | Mem:71.10M, Peak:71.1".
- Threads 961 through 962: "STDOUT: Fra:5 Mem:307.33M (Peak 313.97M) | Time:00:05.90 | Remaining:00:26.46 | Spot: i-[REDACTED] is rendering".
- Threads 963 through 964: "STDOUT: Fra:5 Mem:307.33M (Peak 313.97M) | Time:00:17.06 | Remaining:00:21.26 | Spot: i-[REDACTED] is rendering".

At the bottom of the window are buttons for "Pause", "Save", and "Close".

8. Back in the main Deadline Monitor window, open (double-click) a task to see more details.



9. In the task window that appears, choose a report from the list to see the render log for that task. In this case, the render log is displaying the status of the rendering job for a particular frame of your render.

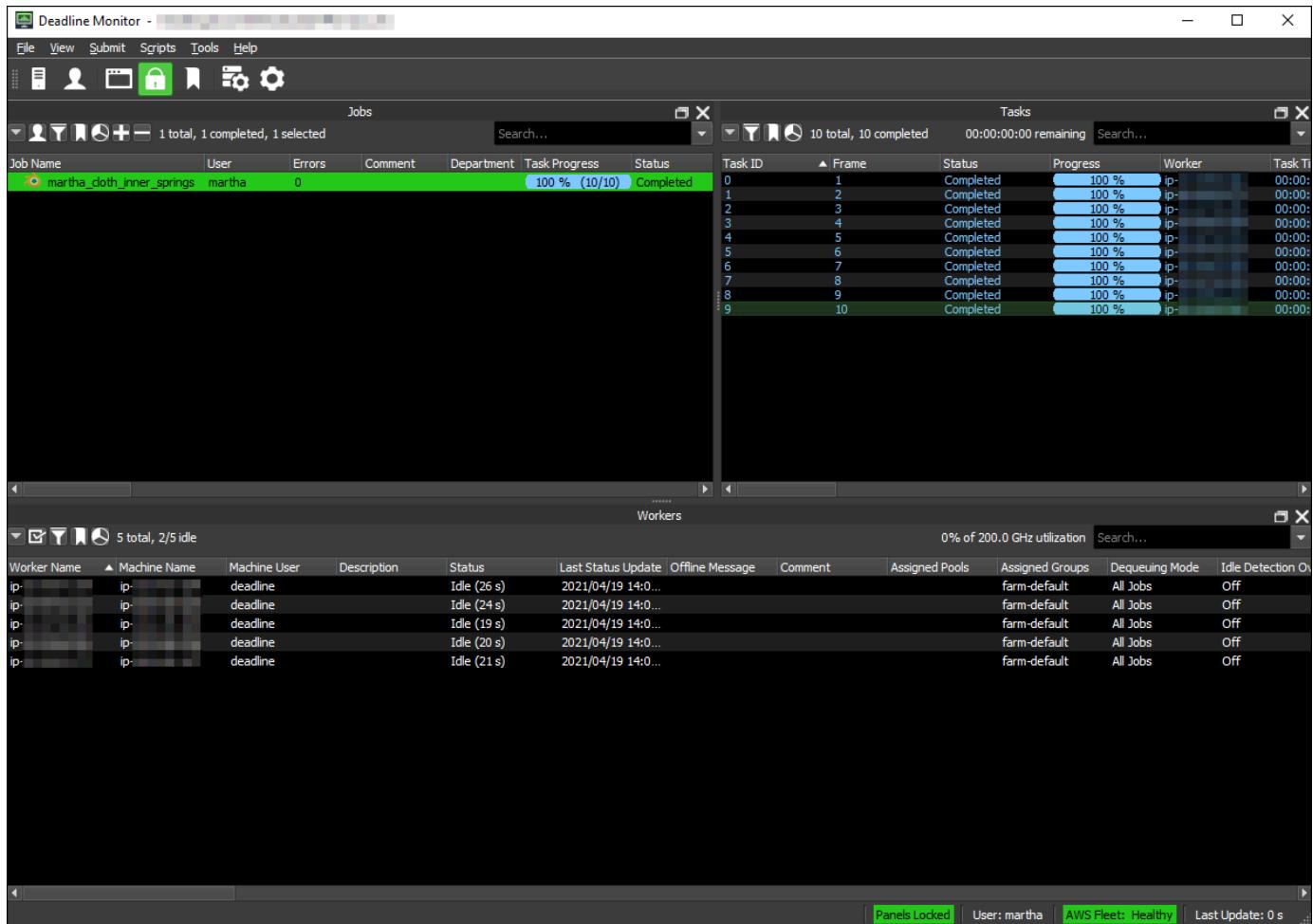
The screenshot shows the Deadline Monitor interface. At the top, there's a header bar with icons for file operations and a search bar labeled "Search...". Below the header is a table with columns: Report Date/Time, Report Type, Worker Name, Task ID, Plugin, Frames, Title, and Task Time. A single row is selected, showing "2021/04/19 15:0..." under Report Date/Time, "Log" under Report Type, "ip-..." under Worker Name, "3" under Task ID, "Blender" under Plugin, "4" under Frames, "Render Log" under Title, and "00:00:01:18" under Task Time. The main area below the table is a large text window titled "Task 3" containing the log output. The log starts with a header:

```
1 =====
2 Log
3 =====
```

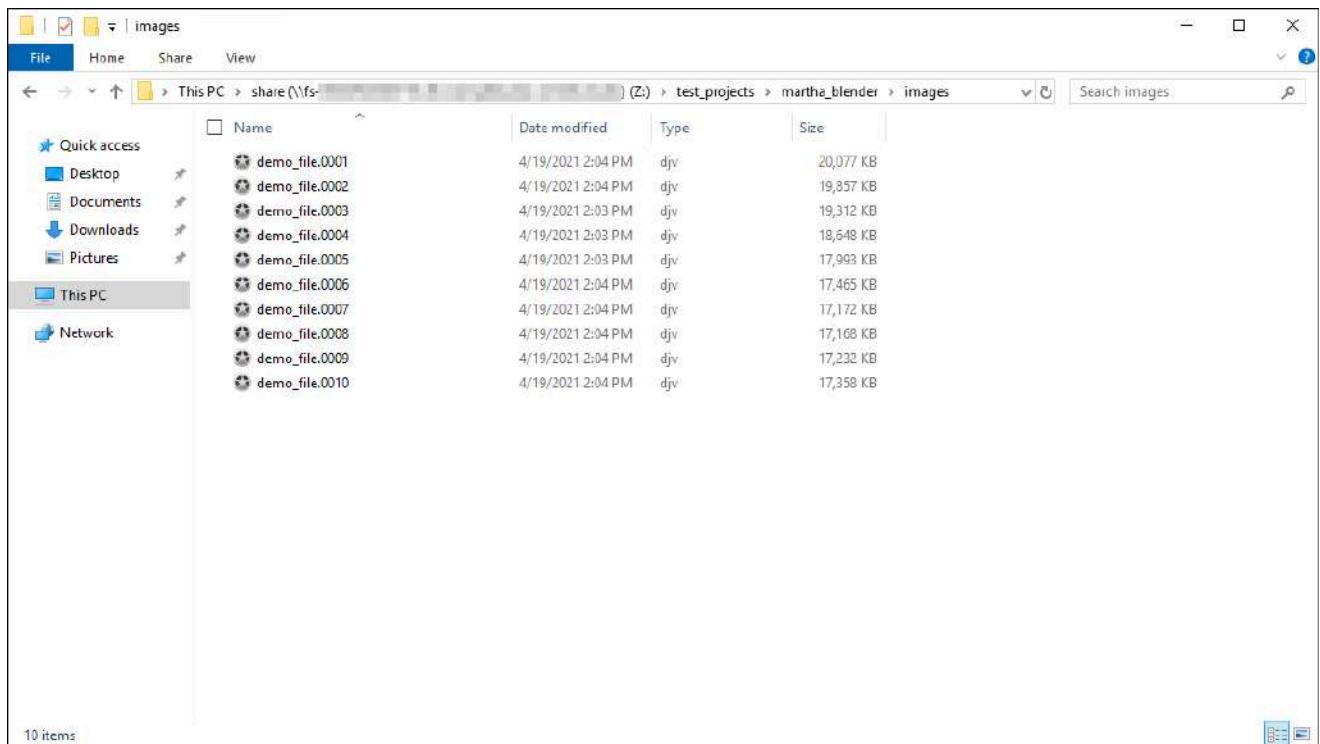
and continues with the following log entries:

```
4 2021-04-19 14:59:41: 0: Loading Job's Plugin timeout is Disabled
5 2021-04-19 14:59:41: 0: WARNING: Python version for 'Blender' plugin is not specified! Defaulting to Py2.
6 2021-04-19 14:59:41: 0: SandboxedPlugin: Render Job As User disabled, running as current user 'deadline'
7 2021-04-19 14:59:43: 0: DEBUG: The Worker is also capable of decompressing responses using Brotli
8 2021-04-19 14:59:43: 0: Executing plugin command of type 'Initialize Plugin'
9 2021-04-19 14:59:43: 0: INFO: Executing plugin script '/var/lib/Thinkbox/Deadline10/workers/ip...
10 2021-04-19 14:59:43: 0: INFO: About: Blender Plugin for Deadline
11 2021-04-19 14:59:43: 0: INFO: The job's environment will be merged with the current environment before ren...
12 2021-04-19 14:59:43: 0: Done executing plugin command of type 'Initialize Plugin'
13 2021-04-19 14:59:43: 0: Start Job timeout is disabled.
14 2021-04-19 14:59:43: 0: Task timeout is disabled.
15 2021-04-19 14:59:43: 0: Loaded job: martha_cloth_inner_springs (...)
16 2021-04-19 14:59:44: 0: Executing plugin command of type 'Start Job'
17 2021-04-19 14:59:44: 0: DEBUG: S3BackedCache Client is not installed.
18 2021-04-19 14:59:44: 0: INFO: Executing global asset transfer preload script '/var/lib/Thinkbox/Deadline10...
19 2021-04-19 14:59:44: 0: INFO: Looking for legacy (pre-10.0.26) AWS Portal File Transfer...
20 2021-04-19 14:59:44: 0: INFO: Looking for legacy (pre-10.0.26) File Transfer controller in /opt/Thinkbox/S...
21 2021-04-19 14:59:44: 0: INFO: Could not find legacy (pre-10.0.26) AWS Portal File Transfer.
22 2021-04-19 14:59:44: 0: INFO: Legacy (pre-10.0.26) AWS Portal File Transfer is not installed on the system
23 2021-04-19 14:59:44: 0: Done executing plugin command of type 'Start Job'
24 2021-04-19 14:59:44: 0: Plugin rendering frame(s): 4
25 2021-04-19 14:59:44: 0: Executing plugin command of type 'Render Task'
```

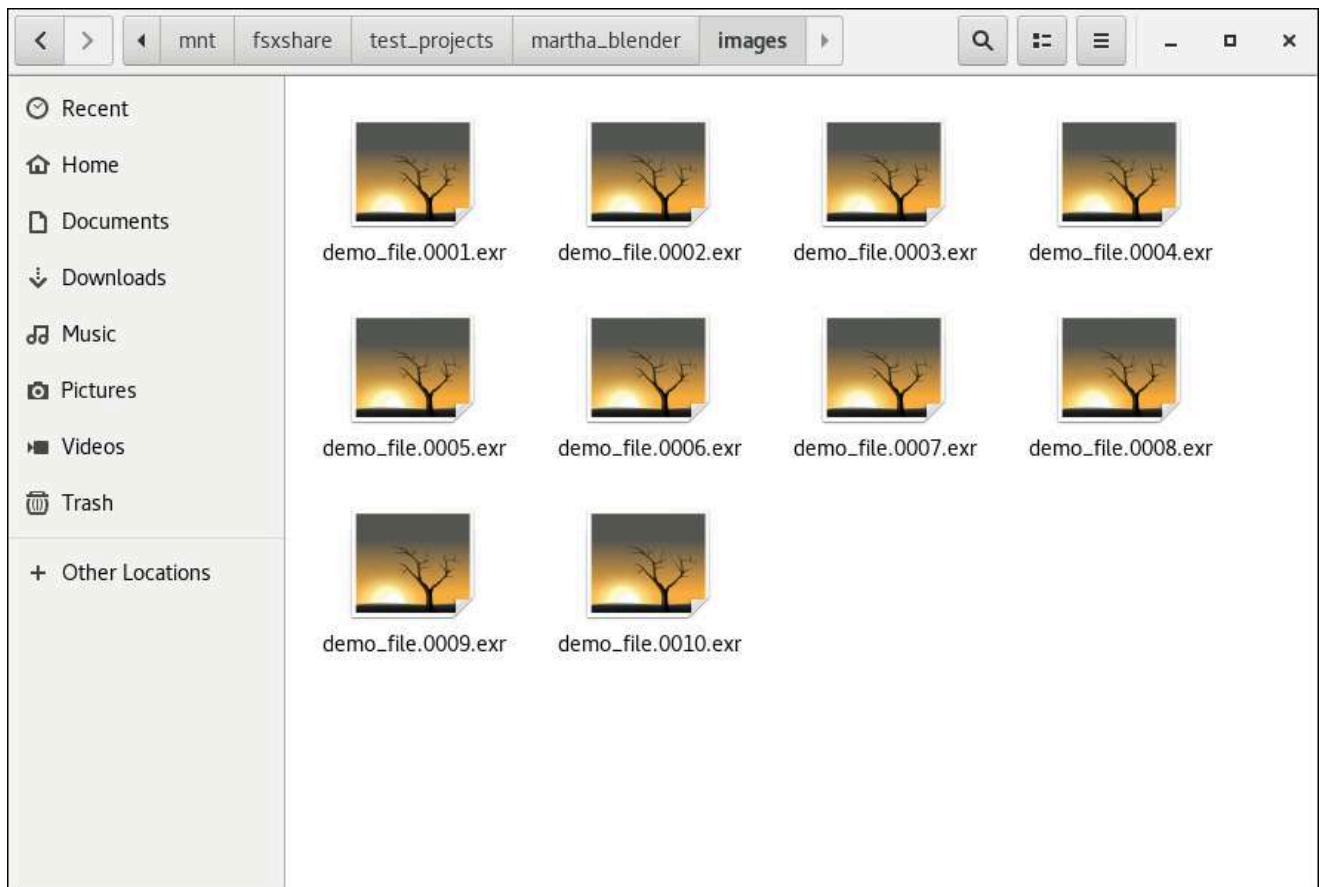
10. Back in the main Deadline Monitor window, check the status of your render job. When your render has finished, the status will change to **Completed**.



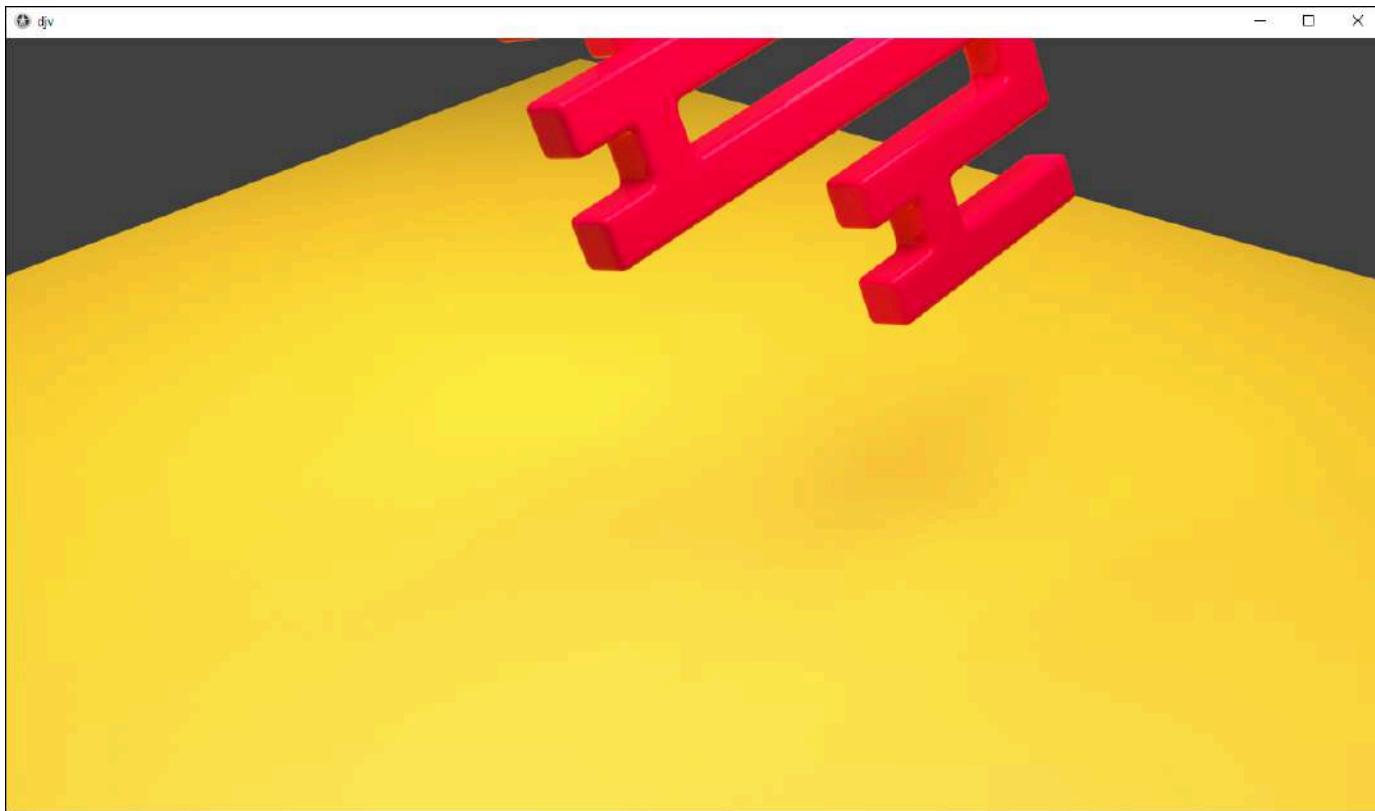
11. In a file browser, navigate to your output directory to view your rendered images.
 - a. On **Windows**, this might be something like: Z:\\test_projects\\<your user name>_blender\\images.



- b. On **Linux**, this might be something like: /mnt/fsxshare/test_projects/<your user name>_blender/images.



12. Open (double-click) on an image to open it in the DJV image and movie viewer.



You've now completed your first render on Amazon Nimble Studio.

Related resources

- [blender.org - Home of the Blender project - Free and Open 3D Creation](https://blender.org)

How to delete a render farm built by StudioBuilder

To delete your render farm created by StudioBuilder, follow the steps in this tutorial. This process involves manually deleting resources and updating to the latest version of StudioBuilder.

⚠ Important

Terminate all workstations before deleting your render farm.

Complete steps one and two in this tutorial before you delete your render farm. If you don't complete these steps, you might not be able to successfully delete your render farm.

Contents

- [Step 1: Remove compute farm studio component](#)
- [Step 2: Update to latest StudioBuilder](#)
- [Step 3: Delete your render farm](#)

Step 1: Remove compute farm studio component

For StudioBuilder to successfully delete your render farm, remove the compute farm studio component from the associated launch profiles that you created.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. For each launch profile that you created associated with the compute farm studio component, do the following:
 - a. Select the button next to the launch profile that has the compute farm studio component attached.
 - b. Choose **Action**. Then choose **Edit**.
 - c. In the **Launch profile components** section, clear the **RenderFarm** check box.
 - d. Select **Update launch profile**.

Step 2: Update to latest StudioBuilder

The first step to deleting your render farm is to follow the instructions in [Update to latest StudioBuilder version](#). After you reach *Step 2 substep 14*, return to this tutorial.

Step 3: Delete your render farm

Important

Before starting *Step 3*, complete all of the previous steps of this tutorial. Deleting a render farm will delete render job history, Deadline configuration, and render workers. Your data on your Amazon FSx fileservers will be unaffected.

StudioBuilder will run and ask you the following questions about how to configure your render farm.

1. **Would you like to delete the farm that StudioBuilder created for you?**
 - Use the arrow keys to choose **Yes, I want to delete my render farm.**
2. **Deleting farms will delete render job history, Deadline configuration, and render workers. Your data on your Amazon FSx drivers will be unaffected. Do you want to delete your existing render farm?**
 - a. To delete your farm, enter **DELETE MY FARM** and press the enter (or return) key.
 - b. If you don't want to delete your farm, enter **QUIT** and press the enter (or return) key to exit. The CLI will continue updating your studio without deleting your render farm.

Your render farm is now deleted. If you would like to create a new render farm, see the [Update to latest StudioBuilder version](#) tutorial.

Artist tutorials for Amazon Nimble Studio

Welcome to the artist guide for Amazon Nimble Studio classic.

The artist tutorials in this guide show how studio artists in VFX, animation, and media production use a cloud studio in Amazon Nimble Studio.

You'll learn how to launch a virtual workstation and use Nimble Studio to collaborate and create remotely. To get started, you must have an active Nimble Studio account and login credentials. If you don't, tell your administrator to follow the instructions in [Adding studio users](#).

When your team's cloud studio is set up in Amazon Nimble Studio, you can sign in to your account and launch a virtual workstation. Your studio administrator configures your workstation to have all of the applications, storage, project data, and rendering resources that you need to do your work. If any resources that you need are missing, or you'd like updates to your working environment, ask your studio administrator for help.

If you don't have an active account or account credentials, ask your studio administrator to set up your account and resources.

 **Important**

Artists can't install new software on a virtual workstation because the system won't retain the software the next time that you launch. To install new software successfully, contact your studio administrator. Ask them to update the AMI for your workstation by following the instructions in the [Update AMIs: Setting up](#) tutorial.

Logging in to the Nimble Studio portal

This tutorial describes how to use Amazon Nimble Studio for the first time as an artist.

Contents

- [Prerequisites](#)
- [Sign in to Nimble Studio portal for the first time](#)
- [Log out of Nimble Studio portal](#)

Prerequisites

As an artist, before you can use Nimble Studio, you need to have the following:

- **Your login information** for Nimble Studio. Your studio administrator or manager will have given this to you. It should include the following:
 - Your user name
 - A link to your Nimble Studio portal
- **A supported web browser** installed on the computer that you will use to access Nimble Studio. Currently, Google Chrome and Mozilla Firefox are the supported web browsers for Nimble Studio. Follow these links to download [Google Chrome](#) or [Mozilla Firefox](#), if needed.

Sign in to Nimble Studio portal for the first time

As an artist, your main interface to your studio will be the Nimble Studio portal. You can access the portal through a web browser using the link that was sent to you, along with your login information. Follow these steps to sign in for the first time.

1. Open the link to Nimble Studio portal that you received with your login information. If you don't have your login information, contact your manager or studio administrator for help.
2. You're directed to a sign-in page. Enter the **Username** that was sent to you and choose **Next**.



Sign in

Username

Remember username

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

3. Choose the **Forgot password** link.



4. Enter the characters in the image to verify that you're a real person and choose **Next**.
5. A confirmation message will appear that says **Reset password email sent**.
 - a. You must associate an email address to this account that you can access. If you don't, your studio administrator will receive the email.
 - b. If you don't have an email address associated with your account, talk to your studio administrator, so that they can add it.



Reset password email sent

Please check your inbox. If you did not receive a password reset email, confirm that your username is correct, or ask your administrator to check your registered email.

Continue

6. Check the email address associated with your Nimble Studio account for a message with the subject **AWS Directory Service Reset Password Request**.
7. Choose the **Reset Password** link in the email message.
8. Enter your new password twice.



Reset password

Please choose a new password

Username: martha

New password

••••••••••••••

Confirm password

••••••••••••••

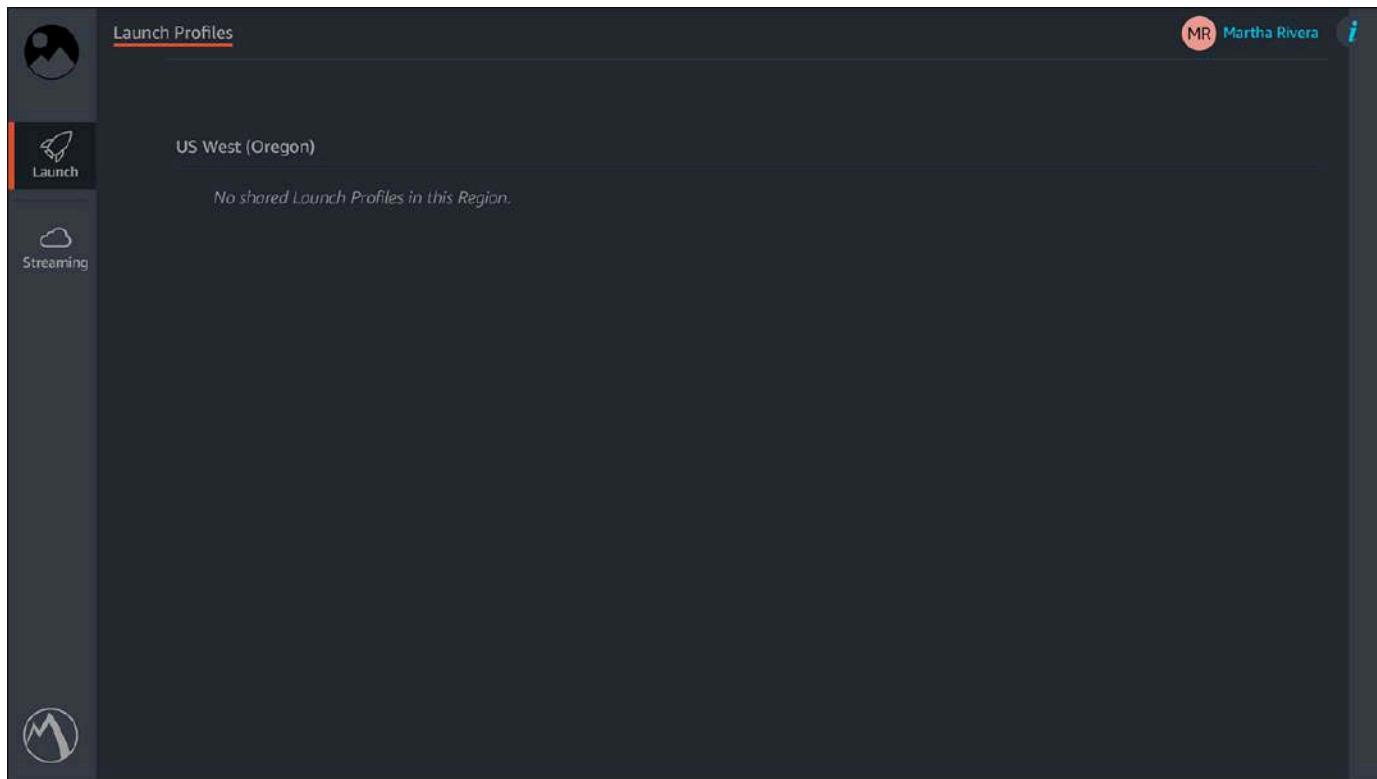
Show password Matches

Set new password

9. Choose **Set new password**.

10. You're redirected to the sign-in page and can now sign in with your new password.

- If, after logging in, you're directed to an AWS IAM Identity Center page, choose the **Nimble Studio card** to continue to the Nimble Studio portal.
11. After logging in, you're directed to the **Launch tab** of the Nimble Studio portal.



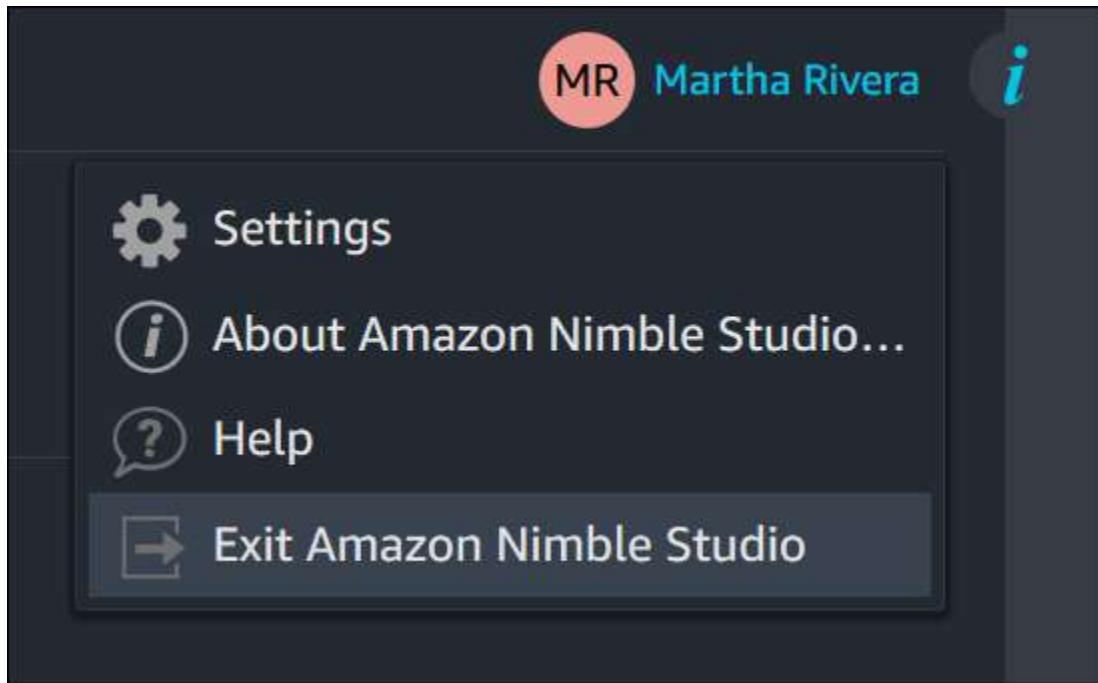
What are launch profiles?

The first time you sign in to Nimble Studio, a message will display that says “No shared Launch Profiles in this Region”.

A launch profile is created by your studio administrator. The profile controls what studio resources you have access to. These resources can include types of workstations, render farms, and shared file systems. Your studio administrator can't share launch profiles with you until you sign in for the first time. This is why the launch tab is blank. Now that you've logged in, you can let your studio administrator know, so that they can share launch profiles with you.

Log out of Nimble Studio portal

1. Log out of Nimble Studio portal by choosing **your name** in the upper right corner of the window, then choose **Exit Amazon Nimble Studio**.



2. On the **Single Sign-On** page, choose **Sign out** near the top right of the window.

Launching a virtual workstation

This tutorial guides you through the process of launching a virtual workstation (streaming session) in Amazon Nimble Studio. Your virtual workstation has access to the software applications, storage, and the render farm that your studio administrator has configured for you to use. Depending on the configuration that your studio administrator has set up, you might have the option to choose your operating system, the size of the virtual workstation, and which tools you can install.

Contents

- [Prerequisites](#)
- [Step 1: Launch a virtual workstation](#)
- [Step 2: Sign in to the virtual workstation](#)
- [Troubleshooting](#)
- [Related resources](#)

Prerequisites

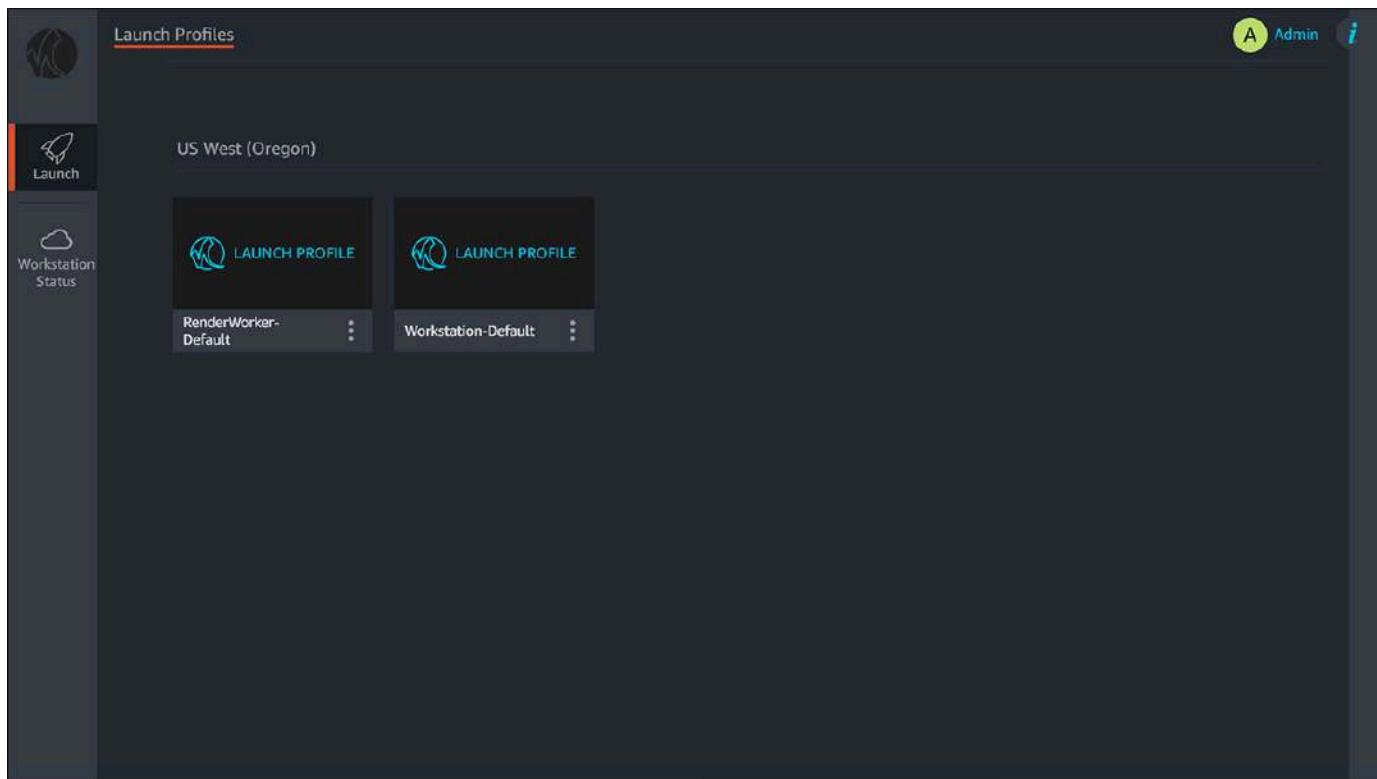
- Before you can launch a virtual workstation, retrieve your user name from your studio administrator and followed the steps in [Logging in to the Nimble Studio portal](#).

Step 1: Launch a virtual workstation

Note

Before you can launch a virtual workstation, first install the latest [DCV client](#).

- Sign in to your Nimble Studio account.
 - If you don't know how to sign in, follow the instructions in [Logging in to the Nimble Studio portal](#).
- Choose the **Launch** tab from the left navigation pane.



- Select the launch profile that you want to use to launch your virtual workstation.

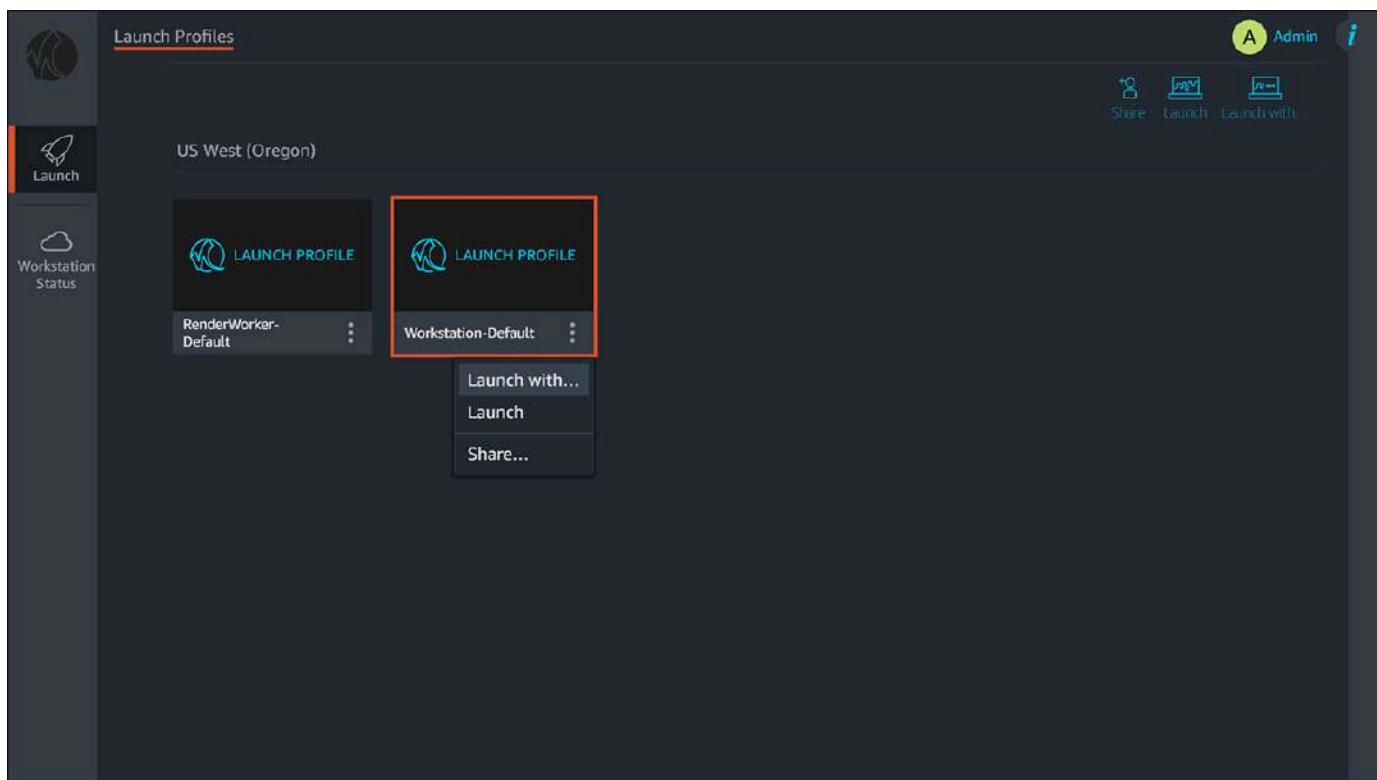
- a. For more information about a particular launch profile, select it and choose the **info** icon in the upper right-hand corner. The info icon is represented by the italicized **i**.
- b. Alternatively, ask your studio administrator for information about your launch profiles.

4. Select the vertical ellipsis

(⋮)

)

on the card to open a dropdown menu.



5. If you have the option, choose an **Instance Type**.

- a. For simpler tasks, choose a smaller instance type such as Small (g4dn.2xlarge). For more complex tasks, choose a larger instance type such as XLarge (g4dn.16xlarge).
- b. To learn more about how many vCPUs, GPUs, and how much memory the different instance sizes have, consult the table in the Product Details section on the [Amazon EC2 G4 Instances](#) page.

6. If you have the option, choose an **Amazon Machine Image**.

- a. Depending on how your studio administrator configured your launch profile, you might have the choice between a Linux or Windows workstation.
- b. This tutorial shows how to sign in to a Windows workstation.

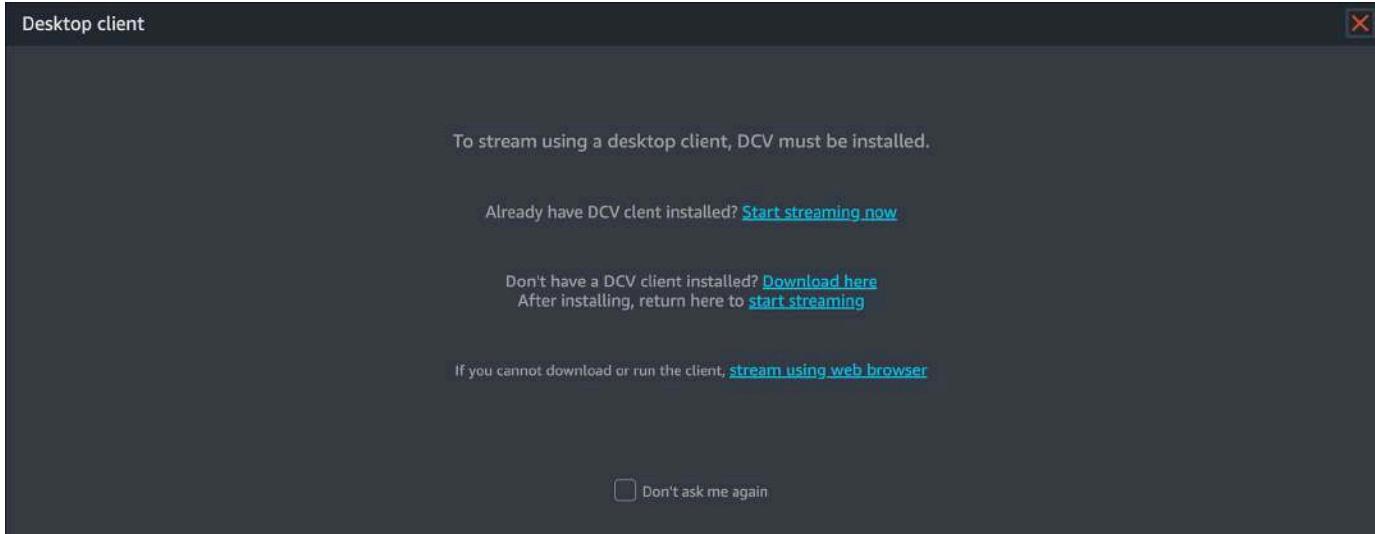
7. Choose your **Streaming Preference**.

- a. This tutorial shows the steps for connecting with the desktop client.
 - b. If you choose **Stream through browser**, you won't have to download a local client, but you will be compromising on streaming quality.
 - c. If you choose **Launch native client**, you will have to download and install the DCV desktop client, but you will get better streaming performance.
8. For Nimble Studio to remember your launch settings for next time, select **Remember my decision and don't ask again**.
- If you choose this check box, the next time that you launch the profile, you can choose the **Launch** option instead of **Launch with...**
9. Choose **Launch**.
10. A status bar will appear that shows you the progress of launching your virtual workstation. Also, the streaming icon in the left panel will change to say **Starting**, and the text color will turn to blue during this process.

Step 2: Sign in to the virtual workstation

When your virtual workstation is ready, a new window will appear reminding you to install the client. If you haven't installed the DCV desktop client yet, choose **Download here** and install the client first.

1. In the new window, choose **Start streaming now**.



- When your browser displays a window that prompts you to open NICE DCV, choose **Open** to continue. The exact wording of this might vary depending on what browser you're using.

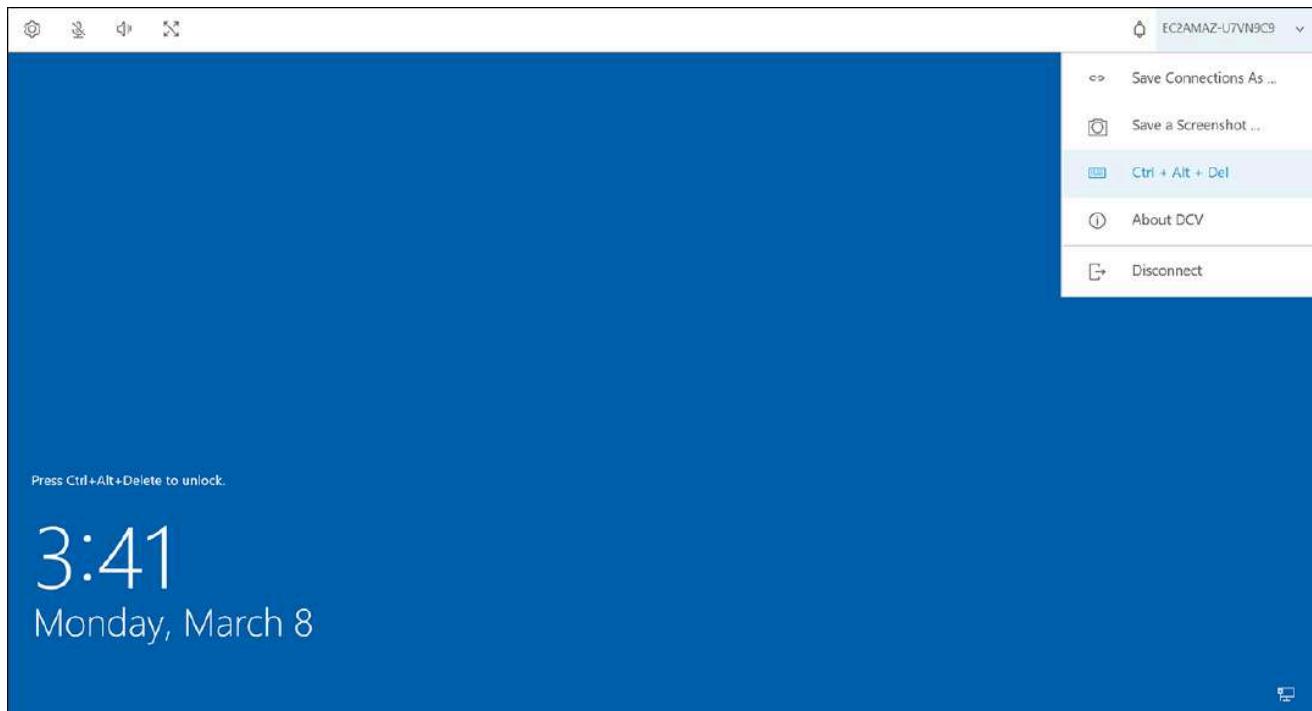
 **Note**

The NICE DCV web browser client runs inside a web browser. You don't need to install the web client. We recommend using the Google Chrome browser to avoid latency. For more information, go to the [Web browser client](#) page in the *NICE DCV User Guide*.

- After the NICE DCV client application opens in a new window, the Windows login screen will display.
- Open the instance menu near the top right of the screen and choose **Ctrl + Alt + Del**.
 - If you launched a Linux workstation you can swipe up the reveal the login fields.

 **Important**

Don't enter **Ctrl+Alt+Delete** on your keyboard. Doing so sends the command to your local computer, not to your workstation.



- Enter your user name.

6. For Password, enter the password that you set up in [Logging in to the Nimble Studio portal](#).

You're now connected to your virtual workstation and you can begin doing your work.

Important

Artists can't install new software on a virtual workstation, because the system will not retain the software the next time you launch. To successfully install new software, contact your studio administrator and ask them to update the AMI for your workstation by following the instructions in the [Update AMIs: Setting up](#).

Troubleshooting

I couldn't launch a virtual workstation because I haven't accepted the End User License Agreement for my AMI.

To solve this error, contact your studio administrator and have them follow the steps in [Adding studio users](#) for accepting the End User License Agreement (EULA).

Related resources

- [Creating your first render on the farm](#)
- [Starting and stopping workstations](#)
- [Uploading files to your virtual workstation](#)

Starting and stopping workstations

This tutorial is for artists who use workstations and want to learn how to start and stop them. For information about how admins can allow artists to start and stop workstations, see [Starting and stopping workstations](#).

The first time a workstation is launched, it's started from the specified Amazon Machine Image (AMI). Depending on your operating system and the size of the AMI that you choose, this process can take 5-25 minutes. After the workstation launches, you can choose between stopping or terminating the workstation. Terminating a workstation means that you will lose the data on the

root volume and that the instance won't be available to use later. To launch the workstation again, you need to spin up a new instance from a new AMI.

When you stop your workstations instead of terminating, you can maintain workstation customization over subsequent logins. You can also reduce time spent on starting and setting up workstations.

Stopping a workstation is equivalent to shutting down a physical machine, while keeping both the root volume and the instance.

The benefits of stopping a workstation include the following:

- Startup times of less than 2 minutes
- Persistent root volume that keeps changes made to it
- Persistent instance with the same IP address
- Saved configurations and data

Stopped workstations don't lose configurations like locally installed applications, operating systems updates, or custom settings. This means that you can restart your workstation with all of your data intact.

Stopping a workstation

You can disconnect, stop, or terminate your workstation from the Nimble Studio portal.

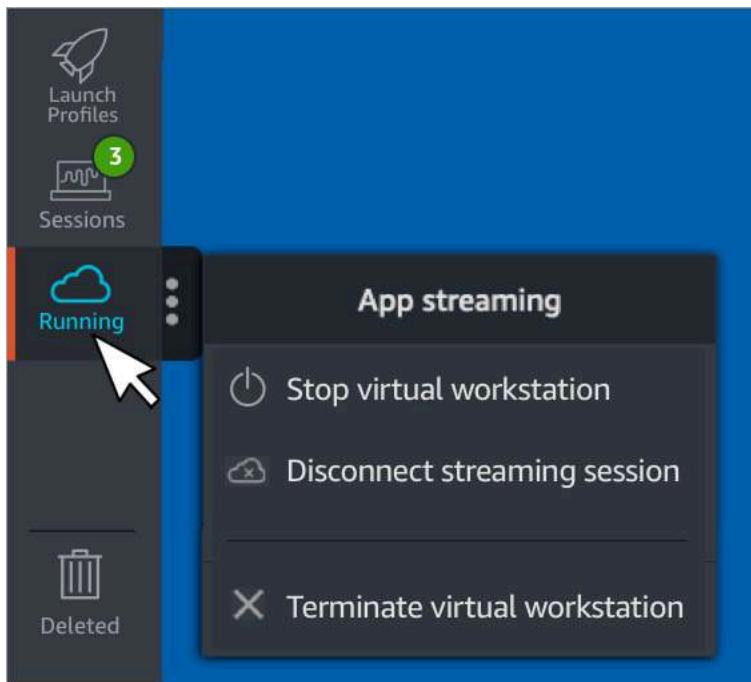
Important

Your studio administrator must turn on [persistence](#) for you to use this feature.

To disconnect, stop, or terminate a workstation from the Nimble Studio portal

1. Start a streaming session. If you don't have a streaming session running, follow the [Launching a virtual workstation](#) tutorial.
2. Select **Running**.
3. Choose one of the following three options.
 - a. Choosing **Stop virtual workstation** stops your workstation. You won't need to reinstall any software or reconfigure your workstation.

- b. Choosing **Disconnect streaming session** disconnects you from your workstation, but leaves the workstation running.
- c. Choosing **Terminate virtual workstation** terminates the workstation. You will need to set up a new workstation next time you launch a workstation.



Uploading files to your virtual workstation

This tutorial is for artists who want to upload files to their virtual workstations. For information about how admins can enable uploads to virtual workstations, see [Enabling uploads to Nimble Studio workstations](#).

If uploads are enabled in your launch profile, you can upload files from your local computer to your remote workstation to streamline your workflow. You can upload things like files defining Maya hotkeys and preferences, a new version of a python script to add panels to Blender, and a folder full of Photoshop brushes. You can only upload to the location designated by your launch profile.

Contents

- [Upload files to your virtual workstation](#)

Upload files to your virtual workstation

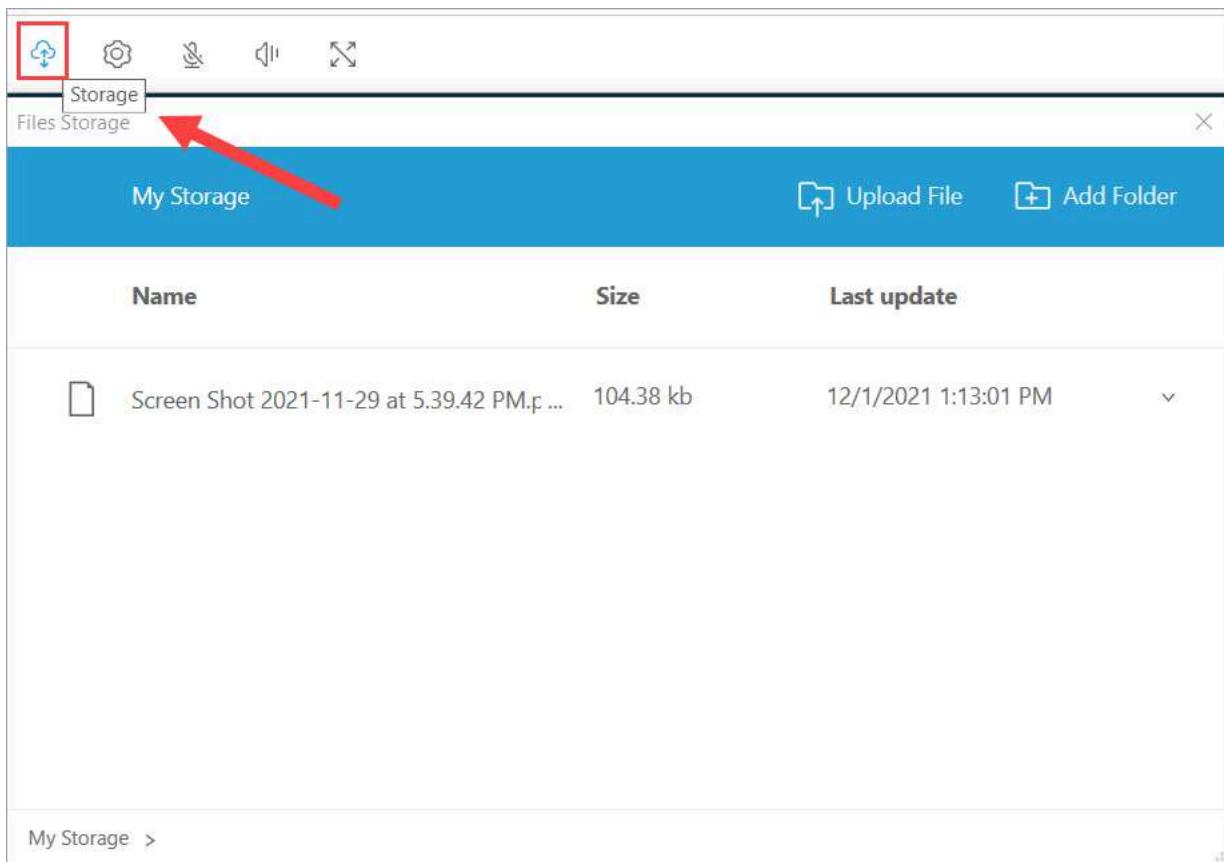
Note

If you launch a virtual workstation using NICE DCV, use DCV server version 2021.2.11445 or later.

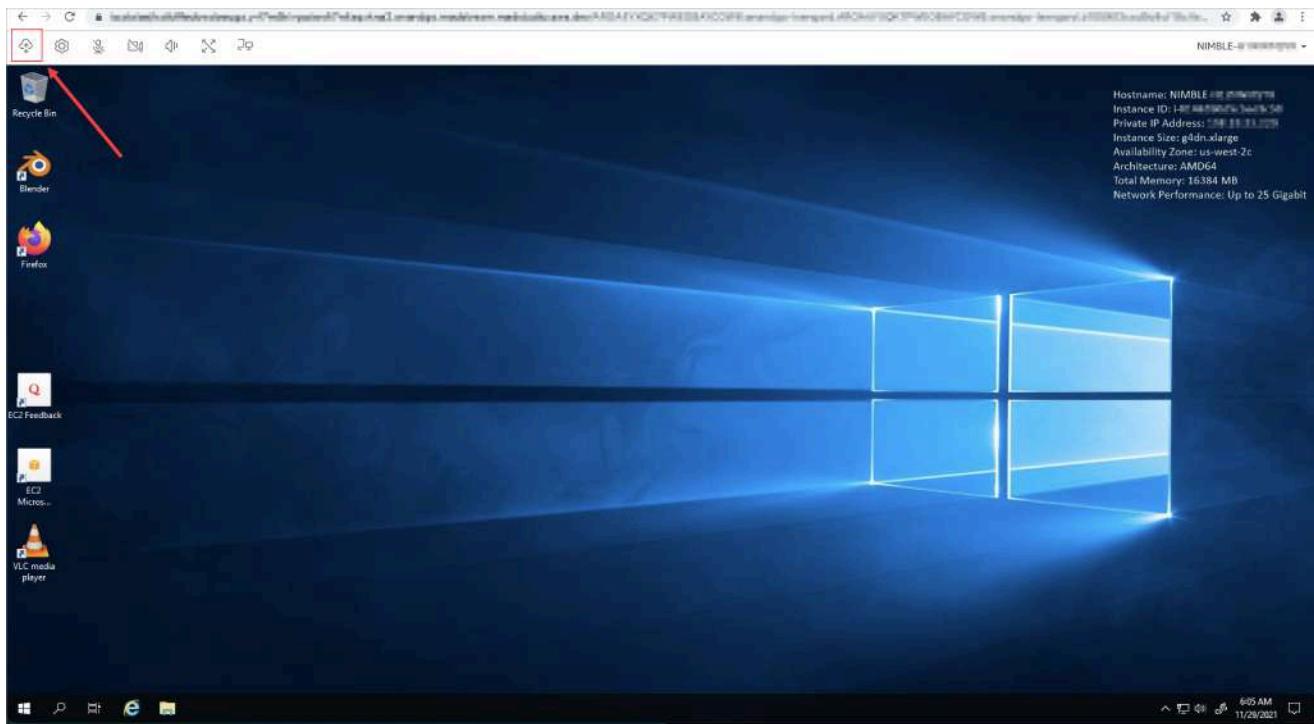
1. Start a streaming session. If you don't have a streaming session running, follow the [Launching a virtual workstation](#) tutorial.
2. Open the file storage window.
 - a. If you launched your workstations using the macOS desktop client for DCV, select **Connection**, then choose **File Storage** from the **DCV Viewer** menu.



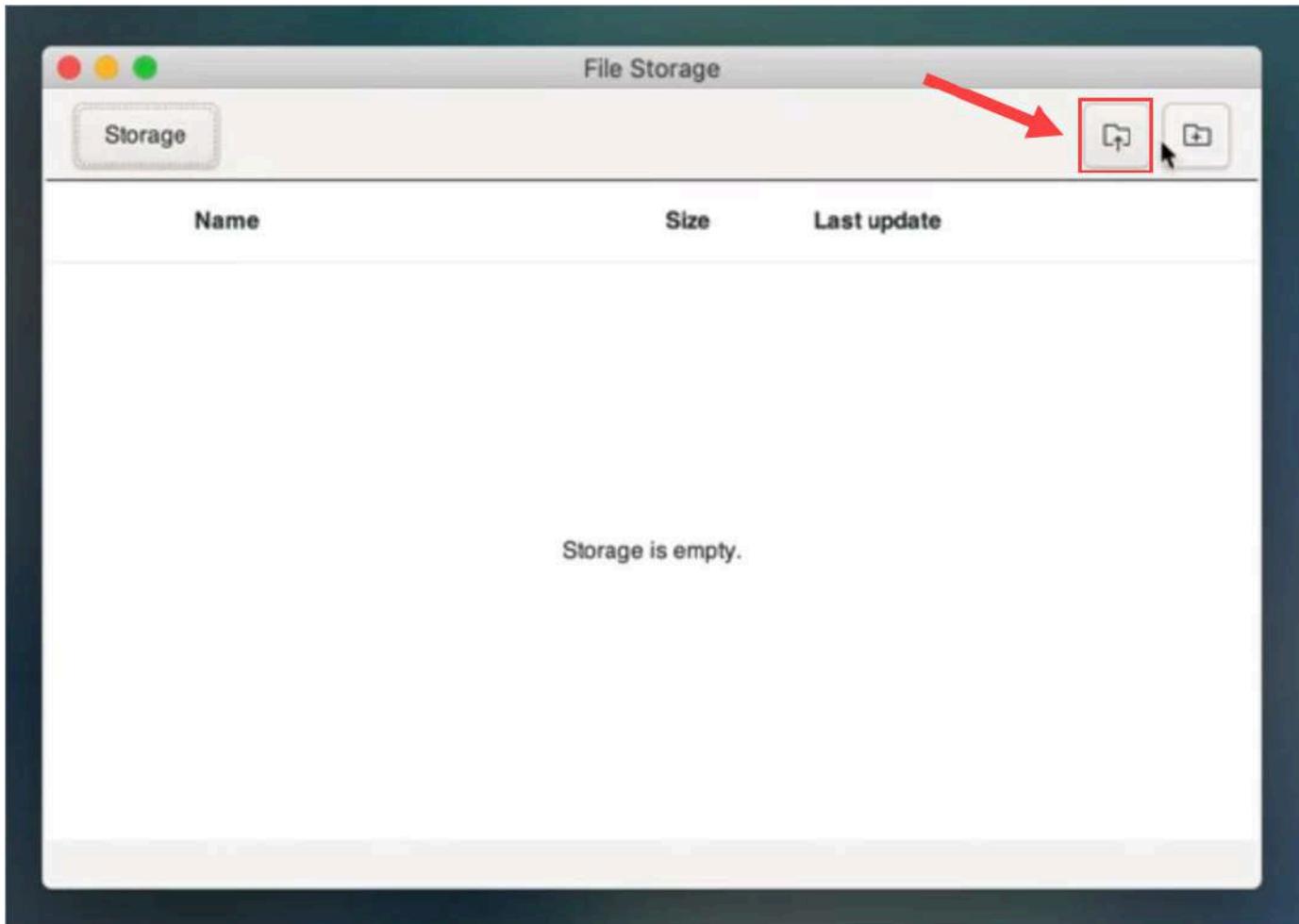
- b. If you launched your workstations using the Windows desktop client for DCV, select **Storage** in the upper-left corner of the menu bar.



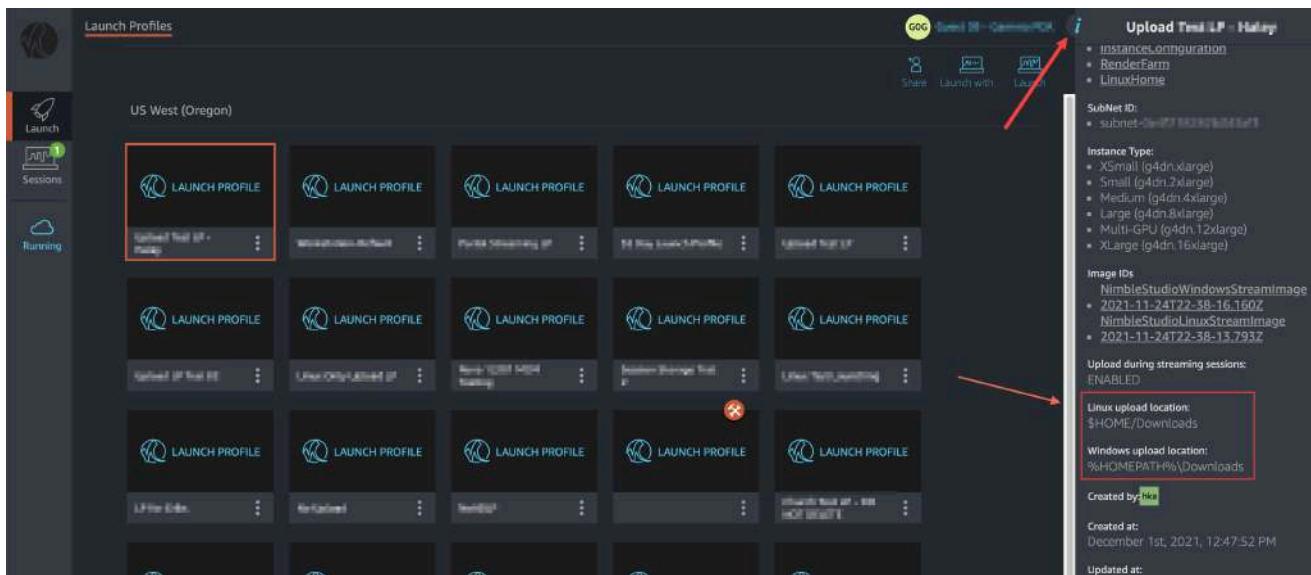
- c. If you launched your workstation using a browser, select the **Storage** in the upper-left corner of the menu bar.



3. Select Upload File.



4. Select the file that you want to upload to your workstation. Choose **Open**.
5. When the file appears in the **File Storage** window, it has been uploaded to the default storage location.
 - a. For Linux, the default location is \$HOME/Downloads unless otherwise specified by the launch profile.
 - b. For Windows the default location is %HOMEPATH%\Downloads unless otherwise specified by the launch profile.
6. To find your upload locations, select your launch profile in the portal. Next, choose the information icon (**i**) in the upper-right corner.
 - In the information pane on the right, the default locations are titled **Linux upload location** and **Windows upload location**.



After you upload your files to the workstation, you can move them to other locations. If you can't move your files, contact your administrator and have them change the upload location to something you have access to.

If your session stops before your files finish uploading, the files won't be uploaded to your workstation.

If the files that you upload don't persist after a session, ask your administrator to attach the LinuxHome studio component to your launch profile. For instructions, see the [Setting up Linux home directories](#) tutorial.

Session auto backup

This tutorial shows artists how to restore their streaming sessions from a backup when their workstation reaches an unusable state. Admins who use or manage workstations can back up and restore sessions by following the [Session auto backup](#) tutorial.

When session auto backup is turned on, Nimble Studio automatically backs up your streaming session storage when your streaming sessions are Stopped. Nimble Studio will also back up your streaming session storage every four hours that your streaming sessions are Running.

Contents

- [Prerequisites](#)
- [Restore from backup](#)

Prerequisites

Before you begin this tutorial, complete the following prerequisites:

- To use this feature, ask your administrator to turn on [persistence](#) and turn on [auto backup](#).
- Your session must be Stopped. For information about how to stop your workstation, see [Starting and stopping workstations](#).

Restore from backup

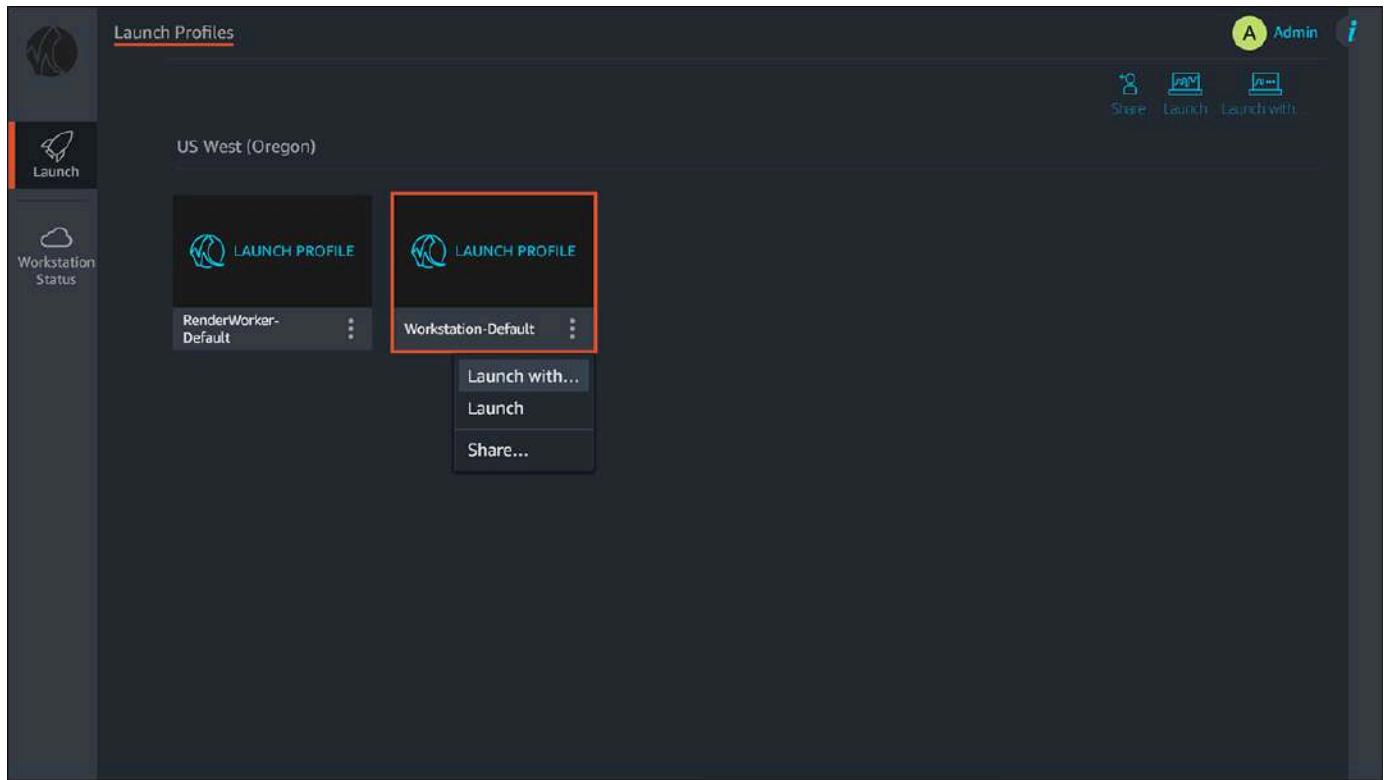
You can restore your streaming session from a backup from the Nimble Studio portal.

Important

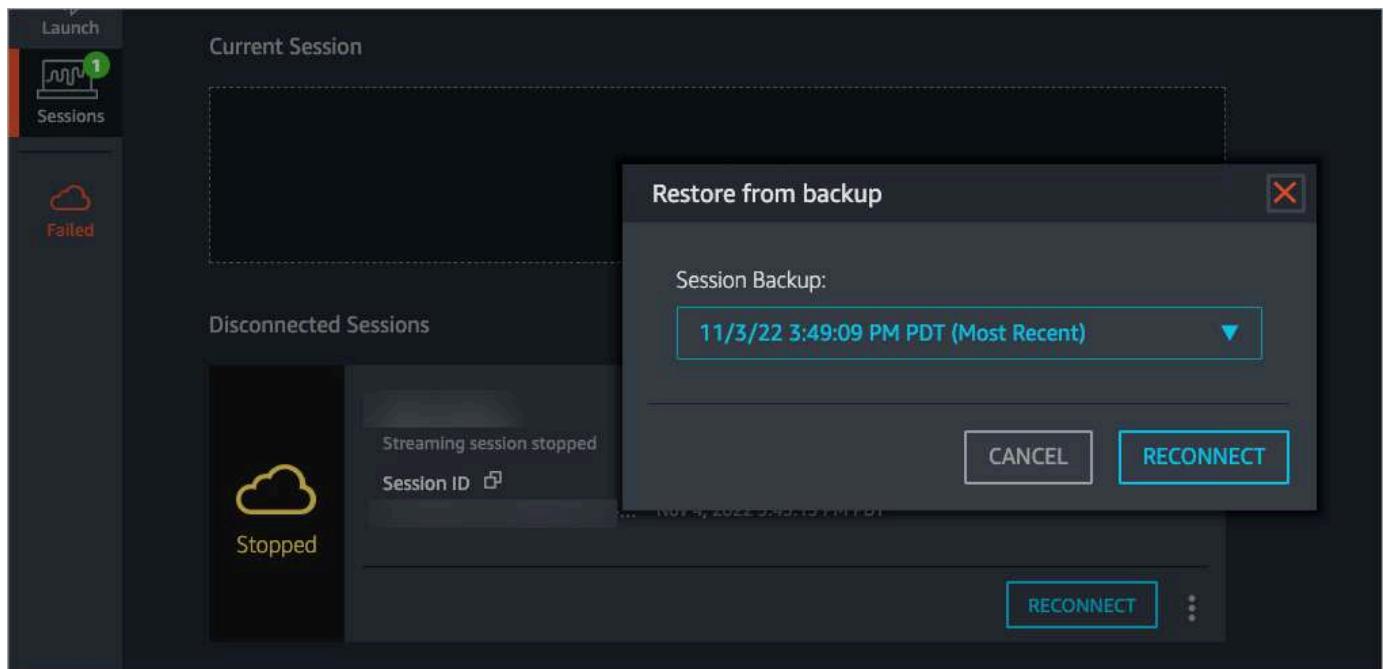
If you shut down your session, you will no longer be able to restore from backups. This means that all backups will be gone and unable to be retrieved.

To restore from backup

1. Sign in to your Nimble Studio account.
 - If you don't know how to sign in, follow the instructions in [Logging in to the Nimble Studio portal](#).
2. Select **Sessions** and find the session that you want to restore from backup. This session must be in the Stopped state.
3. Select the vertical ellipsis
() on the card to open a dropdown menu.



4. Choose **Restore from backup**.
5. Select a backup to restore to.
6. Choose **Reconnect**.

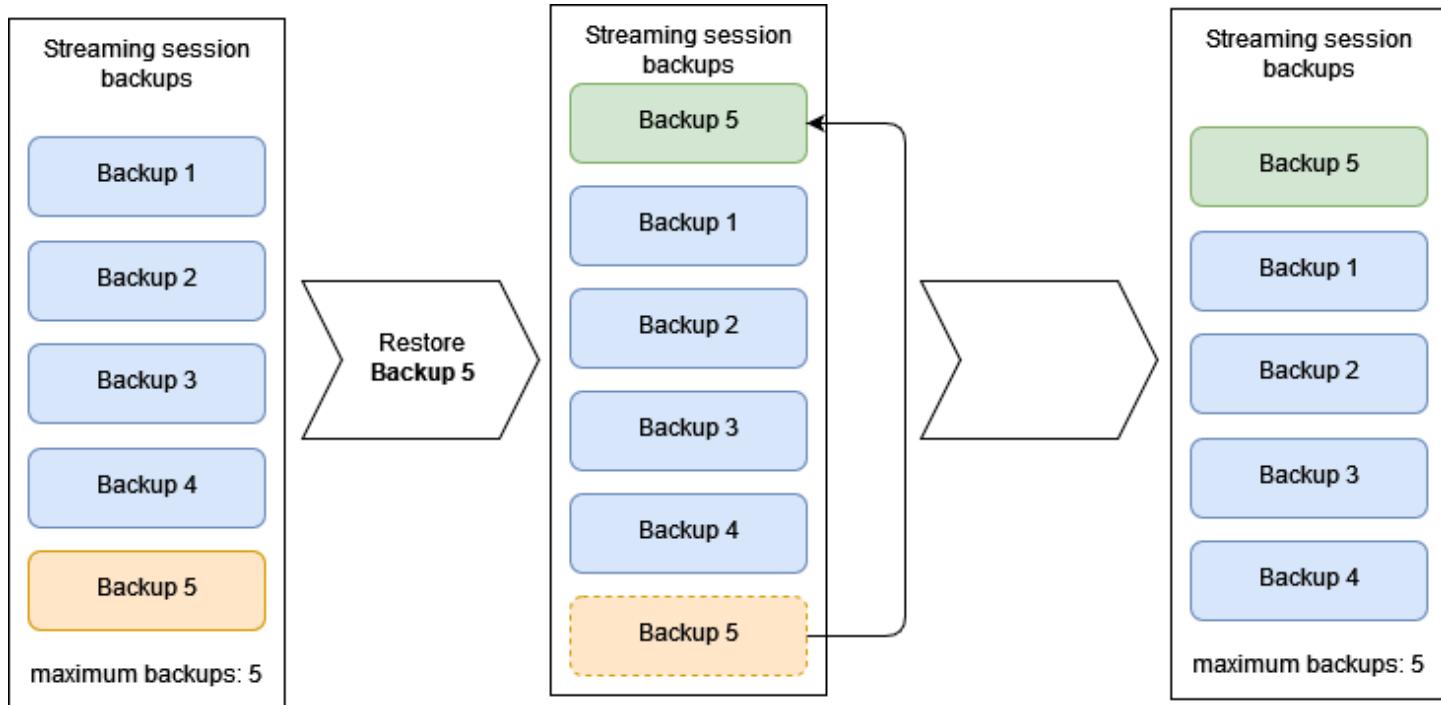


You've now restored your streaming session to a previous state and reconnected to that streaming session. It might take several minutes before the backup is fully restored, depending on the size of the backup. You might also notice some latency issues when working on your workstation while it's restoring from a backup.

Note

Auto backup creates a backup every time that a session is Stopped. If you restore from a backup and stop the session, Nimble Studio will create a backup of that session's current state.

The following diagram depicts restoring from a backup at the maximum number of five backups. When the oldest backup (Backup 5) is restored, a new backup is created by copying it. Then, that oldest backup is deleted, and the new copy becomes the most recent backup.



Testing with AWS assets using Autodesk Maya 2020

This tutorial is for artists. It takes you through the process of downloading high-quality assets that Amazon Nimble Studio provides for testing in your studio.

Contents

- [Prerequisites](#)
- [Step 1: Downloading assets](#)
- [Step 2: Setting up the scene](#)
- [Step 3: Rendering the asset](#)
- [Providing Feedback](#)
- [Related Resources](#)

Estimated time: 30 minutes

Prerequisites

- Before starting this tutorial, launch a streaming session and sign in to a Windows virtual workstation, as outlined in [Launching a virtual workstation](#).
- Your Windows virtual machine must have Autodesk Maya 2020 installed. You also need to connect to a license server for Autodesk Maya 2020, or have an Autodesk account that you can sign in to that provides a license to Autodesk Maya 2020.
- To use the Yeti grooms that come applied to the characters, you will need a license for Peregrine*Labs Yeti version 3.6.0, and Arnold 4.0.1. You can find more information about the Yeti plugin at the [Peregrine*Labs](#) website.
- Optionally, you can use the geometry hair in the character assets and not use the Yeti grooms. This means that you won't need the Yeti license.
- Since this tutorial uses Autodesk Maya 2020, a basic working knowledge of Autodesk Maya is also required. If you haven't used Autodesk Maya before, you can find links to documentation and tutorials on the [Autodesk Area](#) webpage.

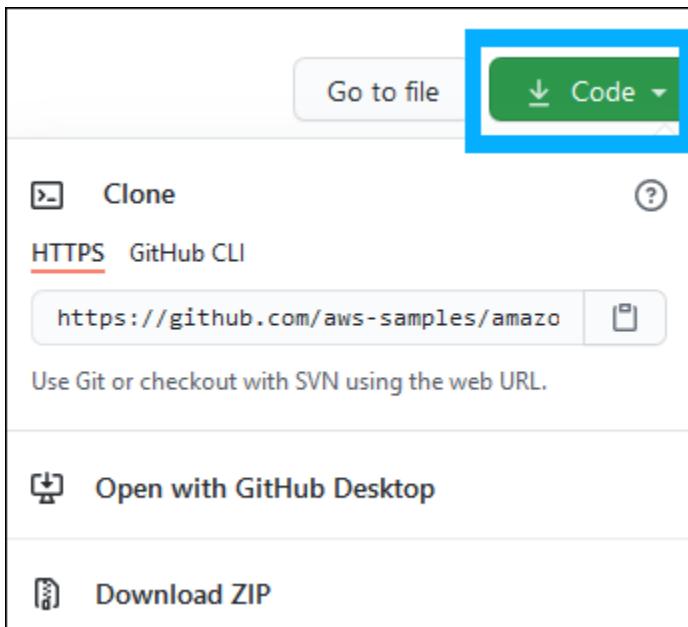
Step 1: Downloading assets

In this step, you will download the assets from the short film Spanner created by FuzzyPixel, an AWS creative team. These assets will help you to test-run your studio. After downloading, you will place them into a project directory on your virtual workstation.

Download assets from GitHub

1. On your virtual workstation, open either **Google Chrome** or **Mozilla Firefox** and navigate to the [Nimble Studio Assets](#) on GitHub.

- When you're on that page, select the **Code** dropdown and choose **Download ZIP**.



- In the window that pops up, select **Save File** and choose **OK**. Notice the location where you saved the file, so that you can find it for use in the next section.

Extract the assets

- In **File Explorer**, navigate to the Z: drive and right-click to open the context menu, then select **New** and choose **Folder**.
- Name this folder `spinnerAssets`.
- In **File Explorer**, navigate to where the `nimblestudio-demo-assets.zip` was saved. For example `C:\Users\your-user-name\Downloads\`.
- Right-click to open the context menu and choose **Extract All...**
- In the pop-up window, you can select a location where you want to extract the assets. To do so, choose **Browse** and navigate to `Z:\spinnerAssets\`.
- Choose **Select Folder**.
- Choose **Extract**.

Note

You will need to maintain the folder structure created by the **nimblestudio-demo-assets.zip** so that rendering works when using the scripts provided in this tutorial.

Step 2: Setting up the scene

In this step, you will open a lighting scene, import assets into the scene, and create a render locally.

Before you begin the following steps, connect to your license server for Autodesk Maya 2020. The method for connecting to the license server should be provided to you from your studio admin. Optionally, if you have an Autodesk account that has an active Maya 2020 license, you can sign in to that account after launching Maya 2020.

Launch AutodeskMaya 2020

1. Choose the **Start Menu** in the lower left-hand corner of your virtual workstation.
2. Enter **Maya** and choose **Maya 2020**.
3. If the Autodesk **Sign in** page appears, enter your account information to verify your license to use Maya. After this is complete, Maya will launch. If you've already connected to a license server for your studio, then Maya will launch without needing an additional sign in.

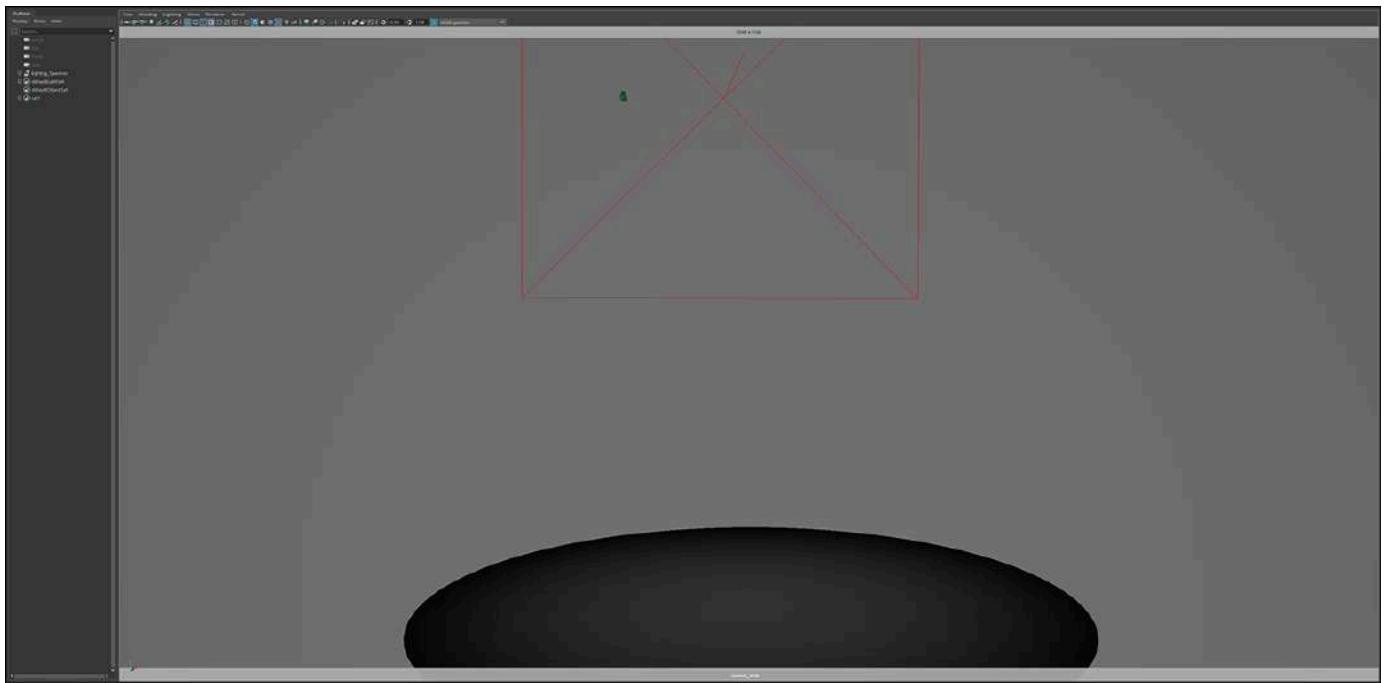
Open a lighting setup

1. Nimble Studio provides a lighting setup for use with the assets.

Spanner - a late afternoon sunset lighting setup



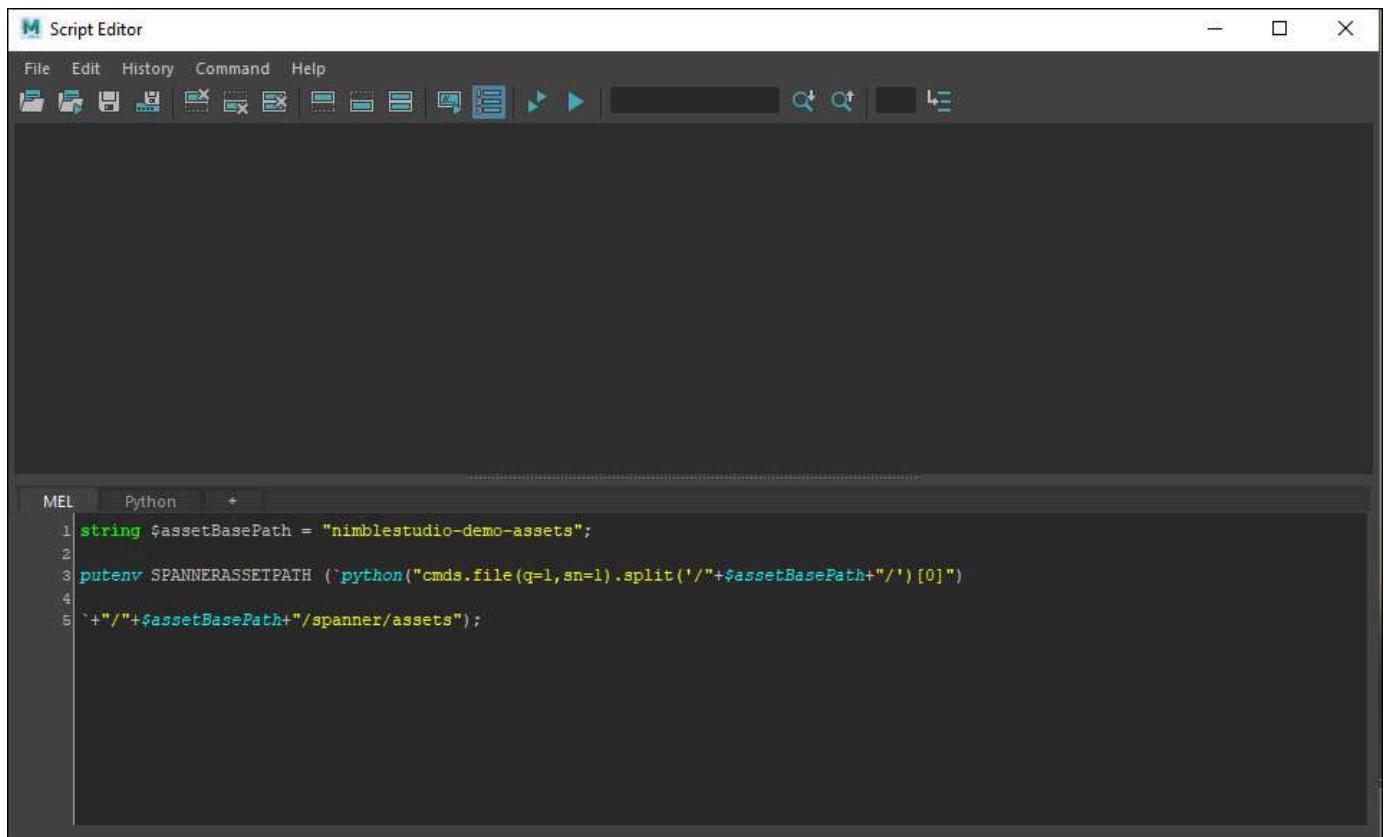
2. In Maya, select the **File** menu at the top left of the application and choose **Open Scene...**
3. In the Maya explorer window that opens, navigate to where you extracted the assets. The files are located within the folder structure where you unzipped the assets. For example: Z:\spannerAssets\nimblestudio-demo-assets\spanner\lightRigs\spanner.
4. Select **lightrig_spanner.mb** and choose **Open**.
5. A menu will open asking to save the existing scene. You haven't done anything with this scene, so choose **Don't Save**.
6. After the scene has loaded, you will be looking through the **camera_wide** in a single view port.



Reconnect the textures

1. In **Maya** open the **Script Editor** by selecting **Windows**, **General Editors**, and **Script Editor**.
2. In the **MEL** tab of the **Script Editor** paste the following:

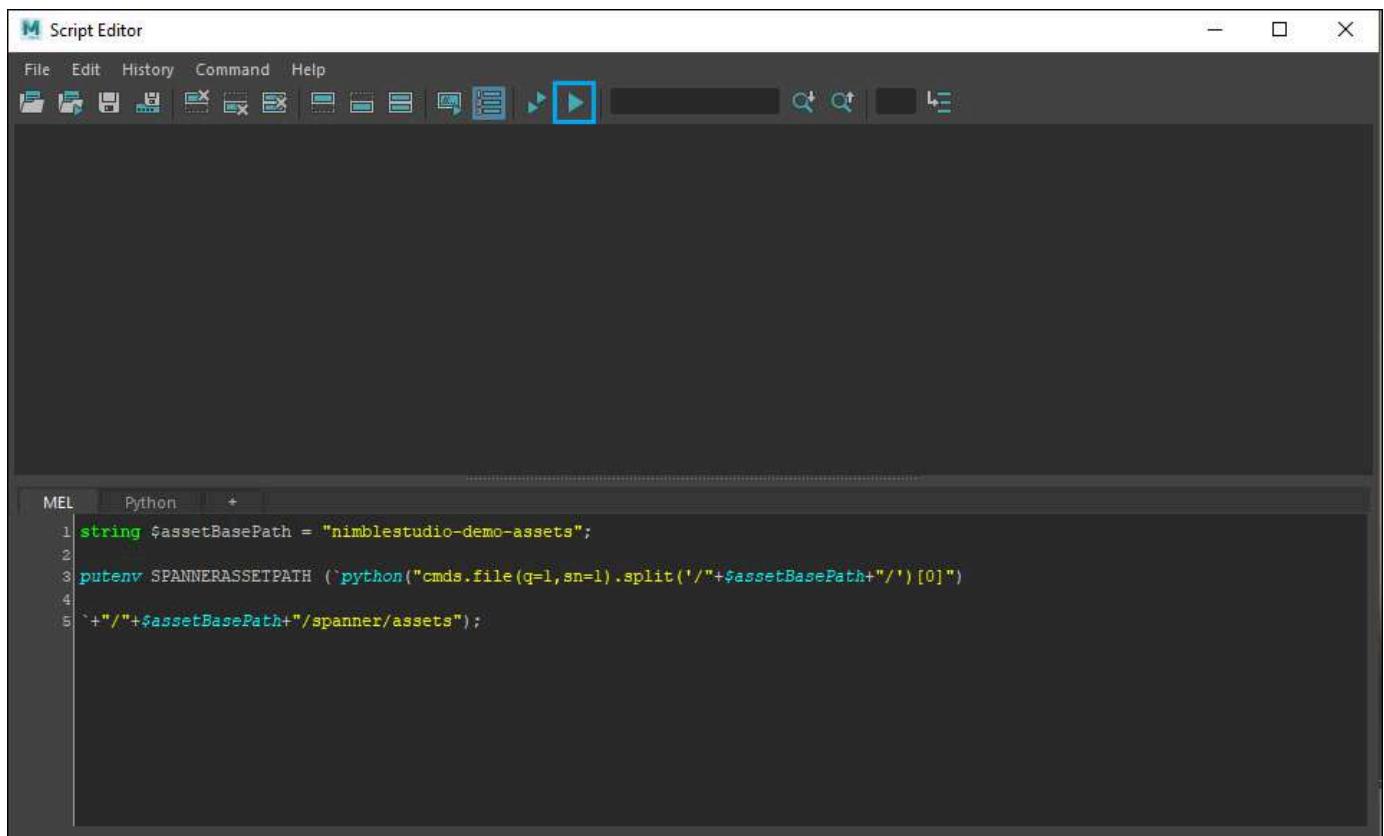
```
string $assetbasePath = "nimblestudio-demo-assets";  
  
putenv SPANNERASSETPATH (`python("cmds.file(q=1,sn=1).split('/"+  
$assetbasePath+"/')[0]")  
  
`+"//"+$assetbasePath+"/spanner/assets");
```



The screenshot shows the Amazon Nimble Studio Classic Script Editor window. The menu bar includes File, Edit, History, Command, and Help. The toolbar contains various icons for file operations like Open, Save, and Undo. The main editor area has tabs for MEL, Python, and a third tab that is currently active. The Python code is as follows:

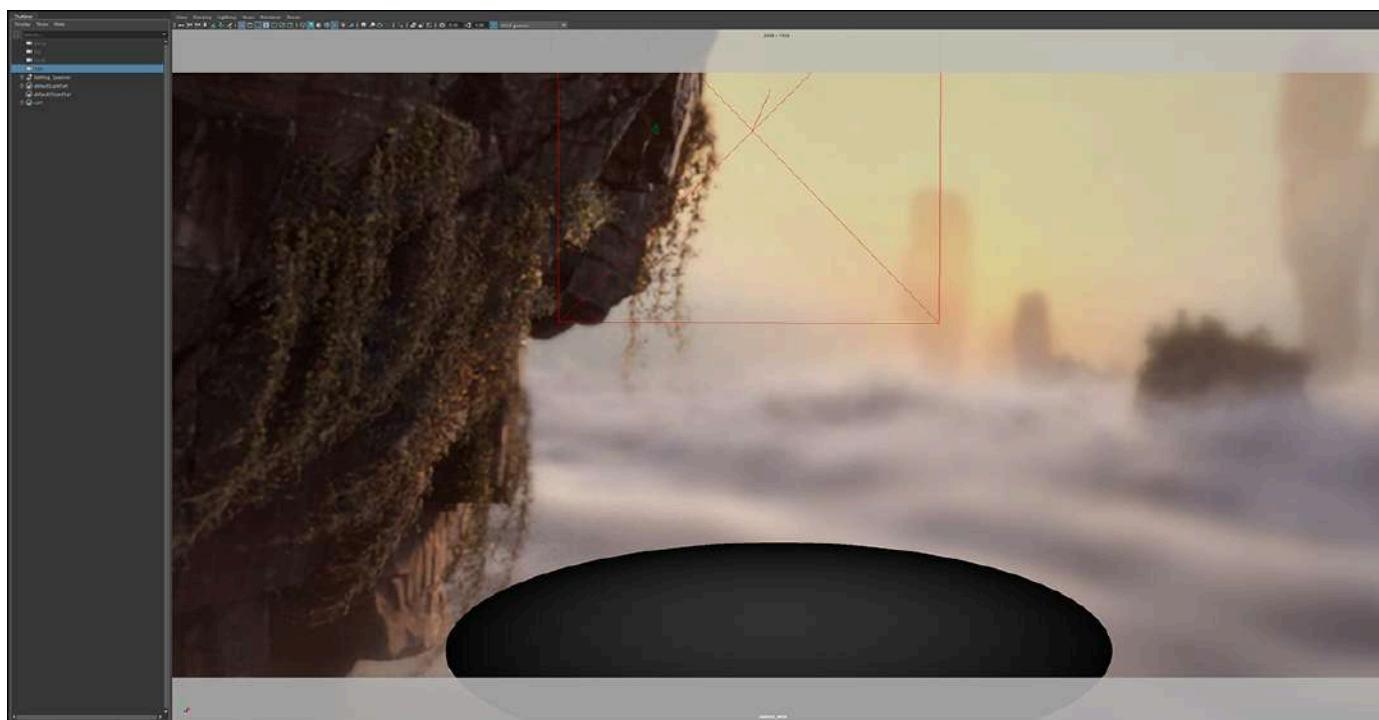
```
1 string $assetBasePath = "nimblestudio-demo-assets";
2
3 putenv SPANNERASSETPATH (`python("cmds.file(q=1,sn=1).split('/'+$assetbasePath+'/')[0]")
4 `+"/*"+$assetbasePath+*/spanner/assets");
```

3. Now choose **Execute**. This is the single triangle button on the menu bar.



The screenshot shows the same Script Editor window as the previous one, but the Execute button in the toolbar is highlighted with a blue border. The Python code remains the same as in the first screenshot.

4. After you run this command, return to the camera view and press **6**. This should make the view port display the textured images and the background.



These steps will correctly path all of the textures for the lightrig and any additional assets you import into the scene.

Note

If you close the file and reopen it, you will have to rerun the script from the [Reconnect the textures](#) section in the local Maya scene. This script has been placed by default into the three lightrig scenes in **Render Settings**, **Render Options**, and **Pre render MEL**. This will allow you to render on a farm.

Import an asset

1. In **Maya**, select **File** in the upper left of the application, and choose **Import**.
2. In the window that pops up, navigate to Z:\spannerAssets\nimblestudio-demo-assets\spanner\assets\noa.
3. Select the **charNoa_rig.mb** file, then choose **Import** in the lower right of the window.

Note

The cameras that are set up in the scene are meant for characters in the default pose. If you pose the character or only want to render props, you will likely need to adjust the camera position to render the scene.

Set up the character's hair

1. If you don't have a license of Yeti or you want to enable the geometry hair, continue with the following instructions. However, if you do have a license for Yeti, you can use the Yeti groom (hair style) for the character and skip to [Step 3: Rendering the asset](#)
2. To enable the geometry hair, select **Windows** and **Outliner** from the Maya on menu bar.
3. In the Outliner, open the following **charNoa_rig:noa_grp** group by selecting the plus icon next to the group name.
4. Select **charNoa_rig:m_groom_grp** and the **charNoa_rig:nhair_grp** group then press **ctrl+H** to hide the Yeti groom and the simulation curves.
5. Select **charNoa_rig:m_hair_grp** group and press **shift+H** to show the geometry hair. Your character will now render with the geometry hair instead of the Yeti groom.

Step 3: Rendering the asset

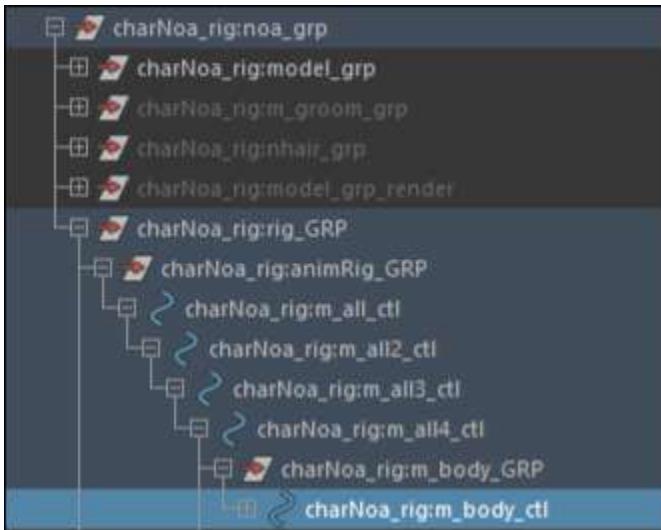
In this step, you will learn to do an Arnold render of Noa. Arnold is a rendering engine built into Maya. For more information about using Arnold in Maya, see the [Arnold for Maya user guide](#).

For each asset (in this case the character Noa), toggle **ON Show Render Geo** before rendering the asset. This toggle will switch between the shaders optimized for the view port, which is useful when doing animation tasks, and the shaders used for rendering.

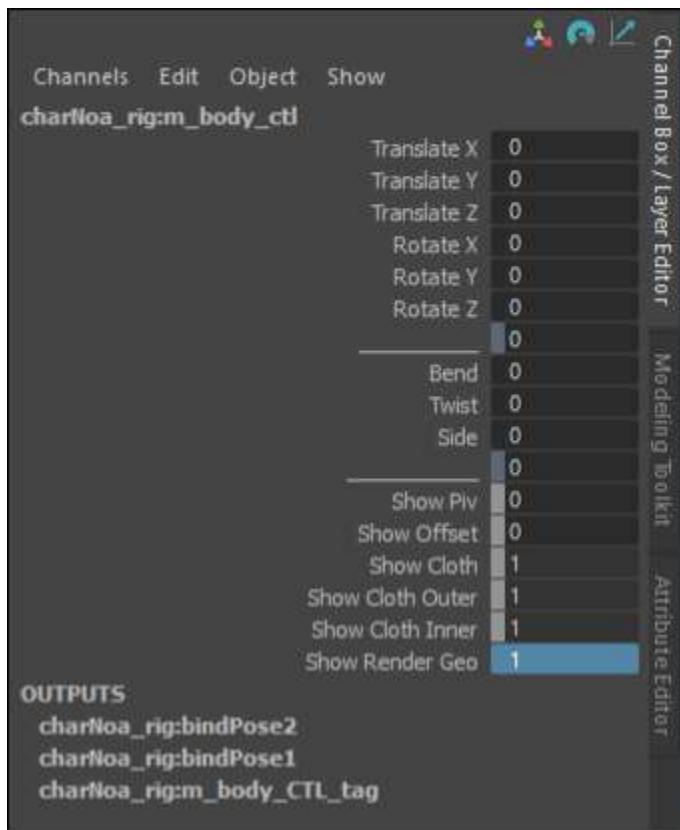
Prepare the asset for rendering

1. In Maya, open the **Outliner** by selecting **Windows** and **Outliner** in the top menu bar.
2. In the Outliner tab, select the **charNoa_rig:m_body_ctl** by navigating through the following groups:
 - a. **charNoa_rig:rig_GRP**

- b. charNoa_rig:animRig_GRP
- c. charNoa_rig:m_all_ctl
- d. charNoa_rig:m_all2_ctl
- e. charNoa_rig:m_all3_ctl
- f. charNoa_rig:m_all4_ctl
- g. charNoa_rig:m_body_GRP
- h. charNoa_rig:m_body_ctl



3. With the **m_body_ctl** selected, go to the **Channel Box / Layer Editor** tab on the right of the interface and change **Show Render Geo** from 0 (which is the view port shader setting) to 1 (which is the render shader setting). Press **Enter**.

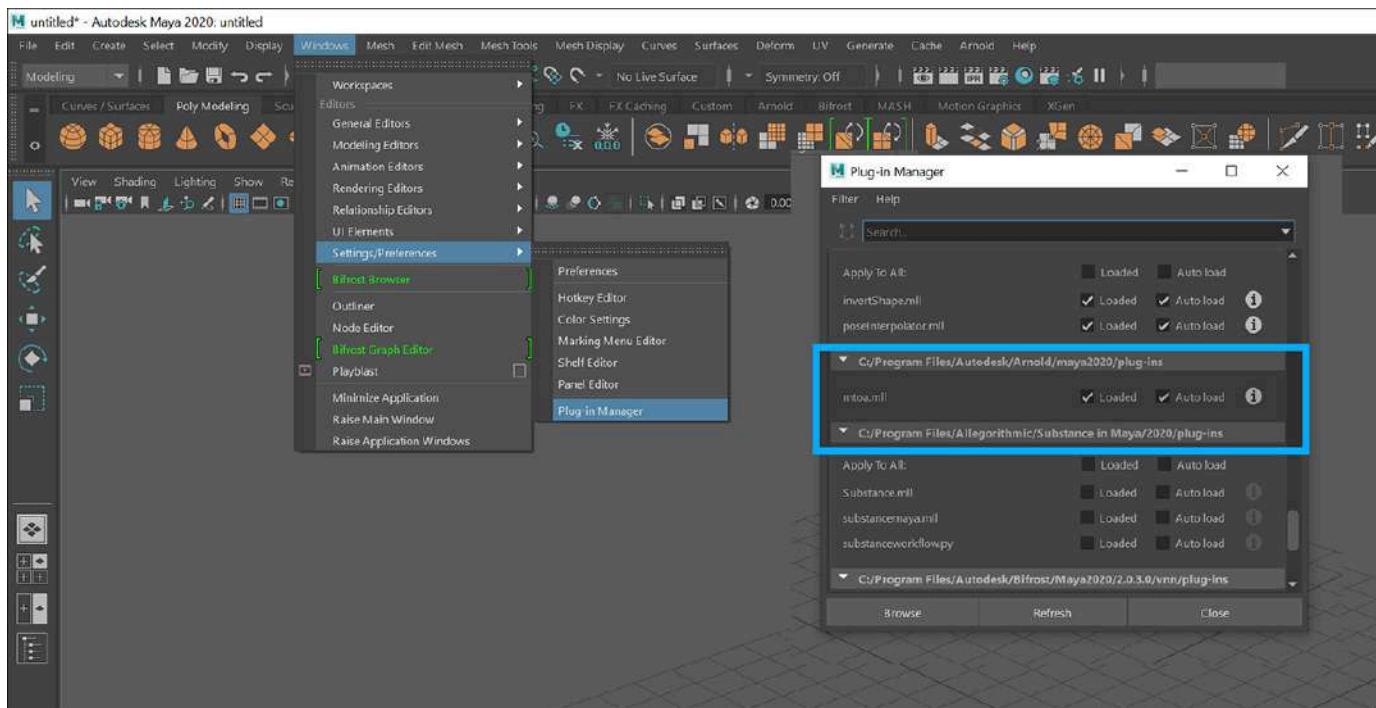


4. When you have updated the Show Render Geo value to 1, the character's appearance will update in the view port.

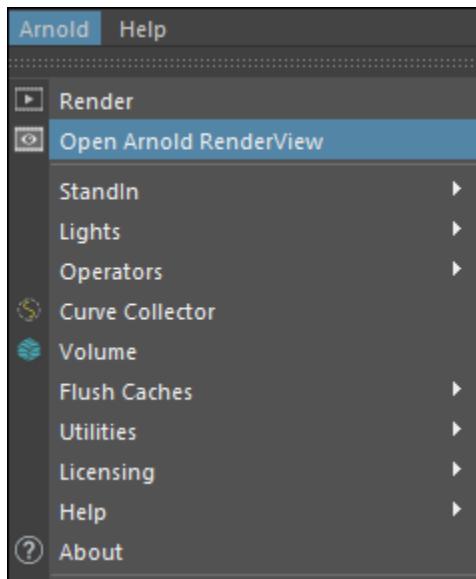


Render the asset

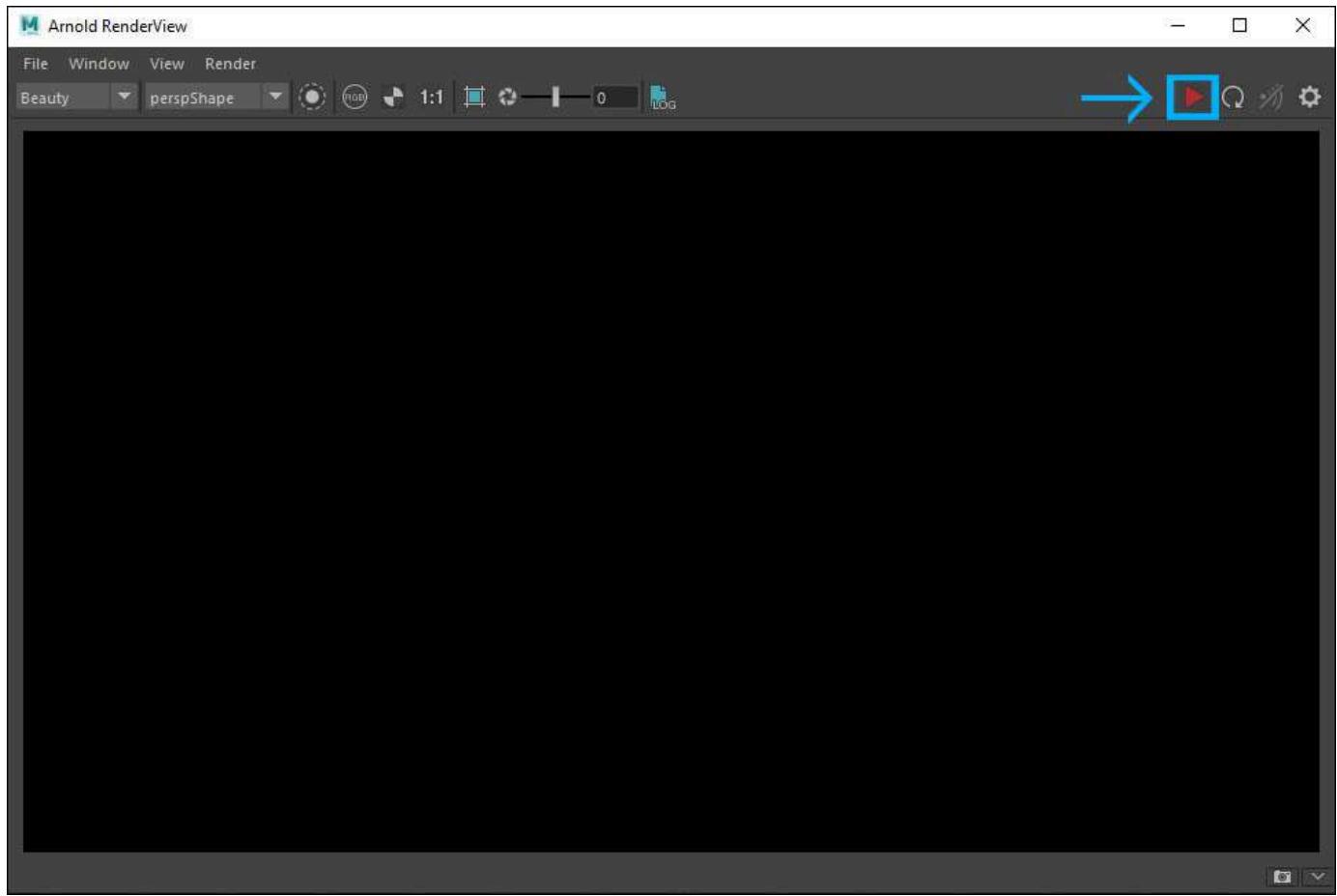
1. Make sure that the Arnold rendering plugin is loaded. In the Maya top bar menu, select **Windows**. Then choose **Settings/Preferences** and **Plug-in Manager**.
2. In the Plug-in Manager Window that pops up, search for **mtoa**.
3. Establish whether **Loaded** and **Auto load** are enabled for the mtoa plugin. The enabled state is denoted by a check mark in the check box to the right of the plugin name. If either of these boxes isn't checked, select the empty check box to enable them, and then choose **Close**.



4. In the Maya view port, select anywhere inside the camera view. This will initialize the render camera to be the one in the view port selected. In this case, it should be the `camera_wide`.
5. In the Maya top bar menu, select **Arnold**. Then select **Open Arnold Render View**.



6. In the camera selection dropdown, the selection should be `camera_wide`. If it's not, select the dropdown and choose `camera_wide`.
7. Choose **Run a live IPR Session** (red triangle) near the upper right of the Arnold RenderView window.



8. After a few minutes (depending on your instance type), the render will start to appear in the Arnold RenderView. Let the render complete.



Providing Feedback

To give feedback about using Amazon Nimble Studio assets, let us know what you think in the **Discussions** section of our [GitHub repo](#).

Related Resources

Check out the following resources for more information about topics in this tutorial.

- [Maya tutorials and support](#)
- [Arnold for Maya user guide](#)
- [Peregrine Labs Yeti](#)

Testing with AWS assets: Blender – Gettin' Fuzzy

This tutorial is for artists. In it, we'll take you through the process of downloading assets that Amazon Nimble Studio provides for testing in your studio. Then, we'll cover how to open a lighting scene, import an asset, and render an image.

Contents

- [Prerequisites](#)
- [Step 1: Downloading assets](#)
- [Step 2: Setting up the scene](#)
- [Step 3: Rendering an image](#)
- [Providing Feedback](#)
- [Related Resources](#)

Prerequisites

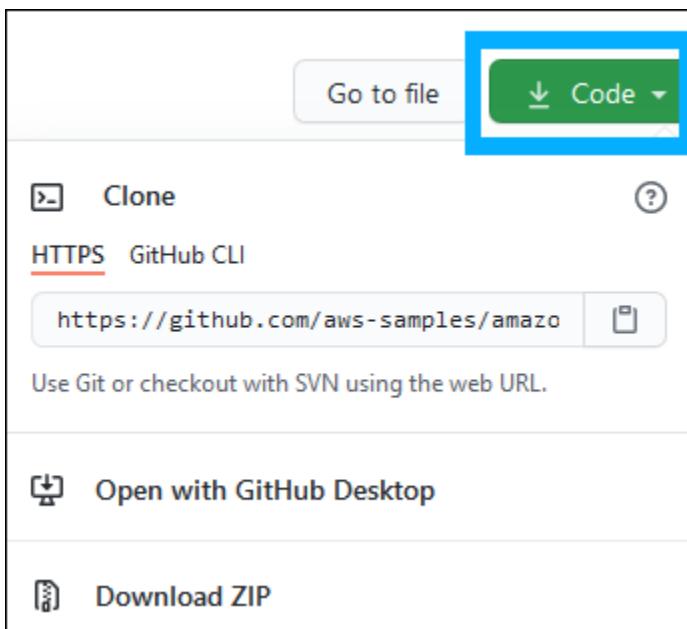
- Before starting this tutorial, launch a streaming session and sign in to a Windows virtual workstation, as outlined in [Launching a virtual workstation](#).
- Your Windows virtual machine must have Blender 2.8 or later installed. Blender should be installed by default. A basic knowledge of how to use Blender is also recommended.

Step 1: Downloading assets

In this step, you will download Blender assets created by FuzzyPixel, an AWS creative team. These assets will help you to test using assets in your studio. After downloading, you will place them into a project directory on your virtual workstation.

To download assets from GitHub

1. On your virtual workstation, open either **Google Chrome** or **Mozilla Firefox** and navigate to the [Nimble Studio Assets](#) on GitHub.
2. When you're on that page, select the **Code** dropdown and choose **Download ZIP**.



3. In the window that pops up, select **Save File** and choose **OK**. Notice the location where you saved the file, so that you can find it for use in the next section.

To extract the assets

1. In **File Explorer**, navigate to the Z: drive and right-click to open the context menu, then select **New** and choose **Folder**.
2. Name this folder nimbleAssets.
3. In **File Explorer**, navigate to where the **nimblestudio-demo-assets.zip** was saved. For example: C:\Users\your-user-name\Downloads\.
4. Right-click to open the context menu and choose **Extract All...**

5. In the pop-up window, you can select a location where you want to extract the assets. To do so, choose **Browse** and navigate to Z:\nimbleAssets\.
6. Choose **Select Folder**.
7. Choose **Extract**.

Step 2: Setting up the scene

In this step, you will learn to open a lighting scene, link in a character, and enable the character to be posed.

Open the lighting scene

1. Choose the **Search Icon** in the lower left-hand corner of your virtual workstation.
2. Enter **Blender** and choose **Blender**.
3. Nimble Studio provides a base lighting setup with lights and a background plane setup for your convenience. After Blender has launched, select **File** in the Blender menu and choose **Open**.
4. In the pop-up menu that opens, choose **Don't Save**.
5. In the **Blender File View** window that opens, navigate to where the lighting scene is in the nimbleAssets folder you created earlier. For example: Z:\nimbleAssets\gettingfuzzy\assets\lightRig.
6. Select **lighting_main.blend** and choose **Open**.

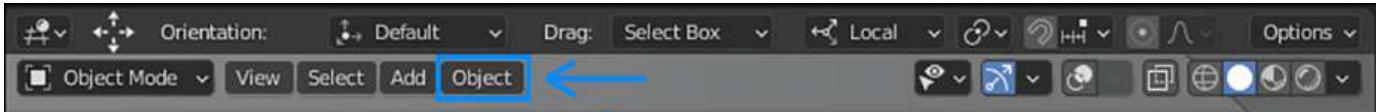
Import the Fuzzy Pixel character

In the following steps, you will be working with FuzzyPixel's mascot, the character Fuzzy Pixel.

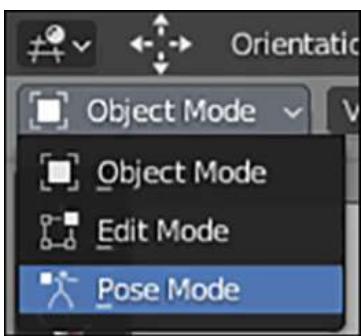
1. In **Blender**, select **File** in the upper left of the application, and choose **Link**.
2. In the Blender File View window that pops up, navigate to Z:\nimbleAssets\nimblestudio-demo-assets\gettingfuzzy\assets\char\fuzzypixel\rig\.
3. Select the **fuzzypixel_rig.blend** file, then choose **Link** in the lower right of the window.
4. In the folder structure that opens, select **Collection**.
5. Then select **rig_SET_fuzzypixel** and choose **Link**.

Pose the character

1. In the Blender Outliner Window*, * select **rig_SET_fuzzyPixel**.
2. In the Header Menu of the camera view, select **Object**.

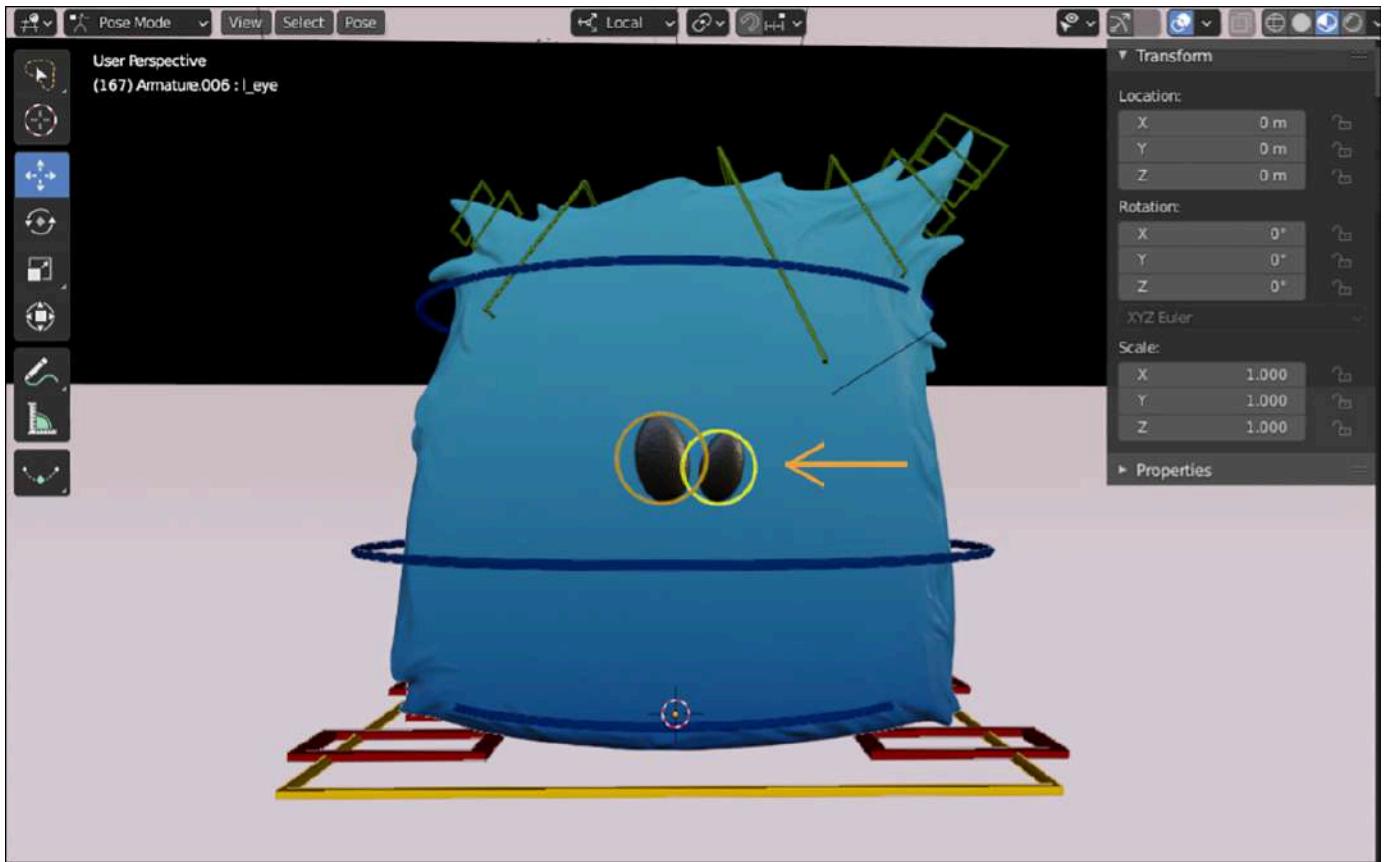


3. Now, select **Relations, Make Library Override....** Then choose **Make Library Override**. This will allow you to select the rig of the Fuzzy Pixel character so that you can pose it.
4. Select the **Armature** for Fuzzy Pixel.
5. Next, in the header menu of the camera view, select the interaction dropdown menu and choose **Pose Mode**.

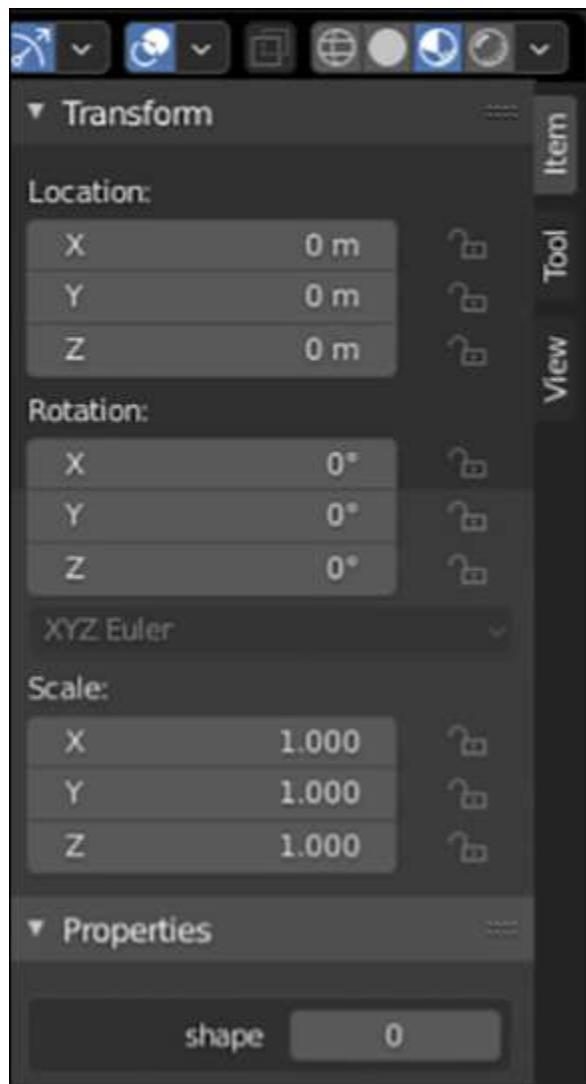


Now that your character is in Pose Mode, you can select the different controls of the Armature and move them to pose the character.

6. To change the shape of the eyes, select the eye control. To do so, select the circle named **I_eye** that goes around the eye.



7. Next, press **N** to toggle open the **Side Bar**. If the Side Bar is already open, this will close the window. If the Side Bar is removed, just press **N** again to open the Side Bar.
8. Open the **Properties** menu and change the value of the **shape** attribute to a value of 0-19. There are 20 different shapes to choose from to add different expressions to Fuzzy Pixel.



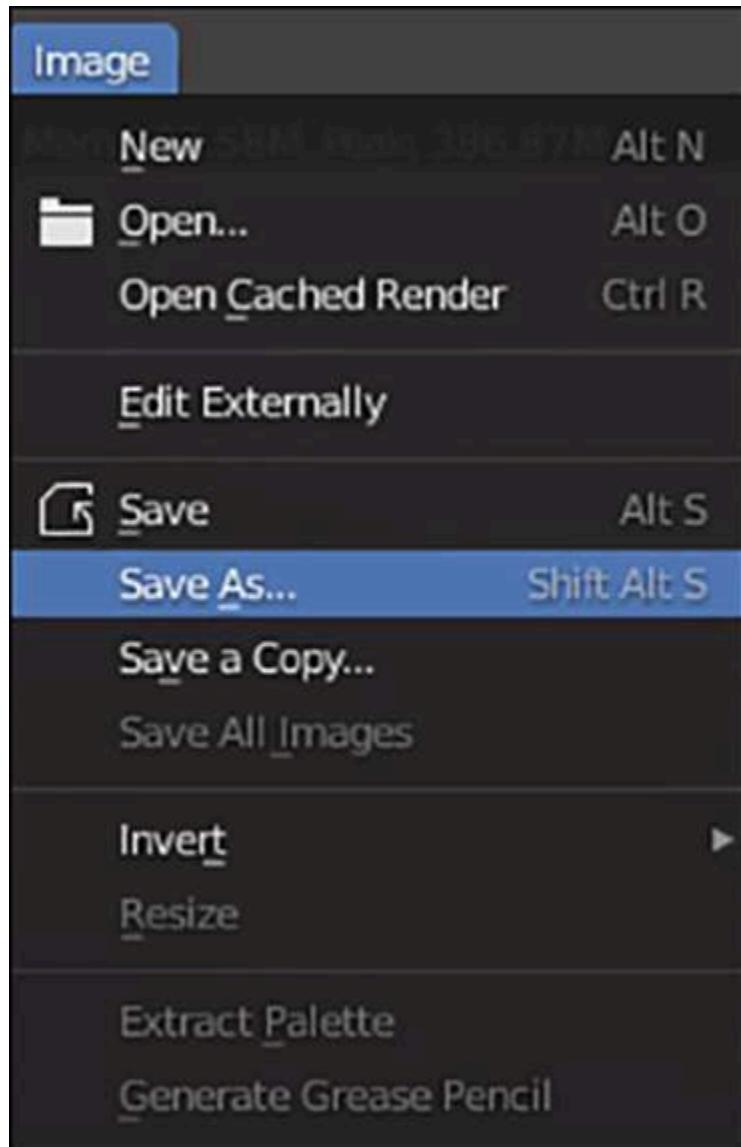
At this point, you can take some time to pose the character or move on to the next step.

Step 3: Rendering an image

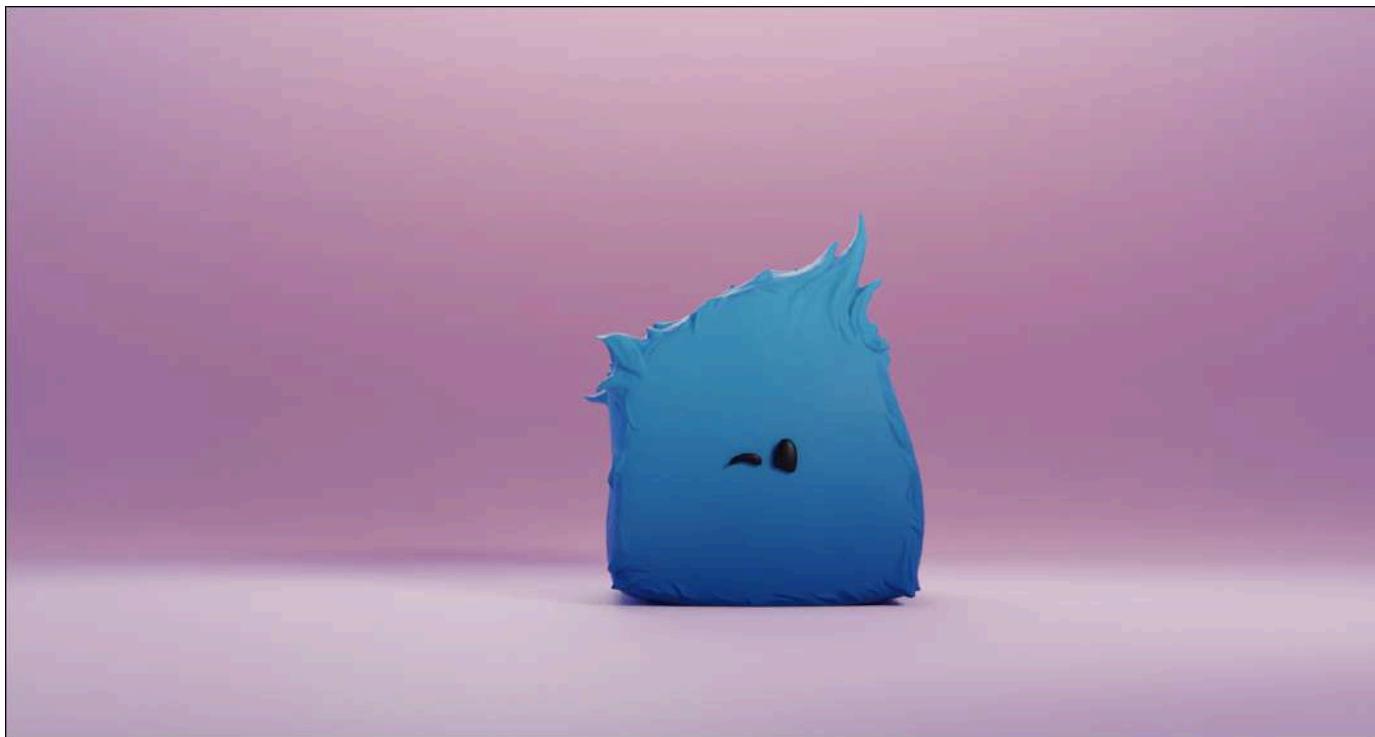
In this step, you will learn how to render a still image of your posed character.

To render a still image

1. In the **Info Window** at the top of the interface select **Render, Render Image**.
2. The rendering time of the image will vary depending on the CPU resources available to your virtual workstation. After the image has finished rendering, you can save the image by selecting **Image, Save As...**



3. In the Blender File View, you can then decide where to save the image and the file format you desire.



Providing Feedback

To give feedback about using Amazon Nimble Studio assets, let us know what you think in the **Discussions** section of our [GitHub repo](#).

Related Resources

If you need help getting started with Blender, you can find [Blender tutorials](#) on [Blender.org](#).

Testing with AWS assets: Blender - Shockingly Fuzzy

This tutorial is for artists. In it, we'll take you through the process of downloading assets that Amazon Nimble Studio provides for testing in your studio. Then, we'll cover how to open a lighting scene, import an asset, and render an image.

Contents

- [Prerequisites](#)
- [Step 1: Downloading assets](#)
- [Step 2: Setting up the scene](#)

- [Step 3: Simulating the cloud and hair](#)
- [Step 4: Rendering an image sequence](#)
- [Providing feedback](#)
- [Related resources](#)

Prerequisites

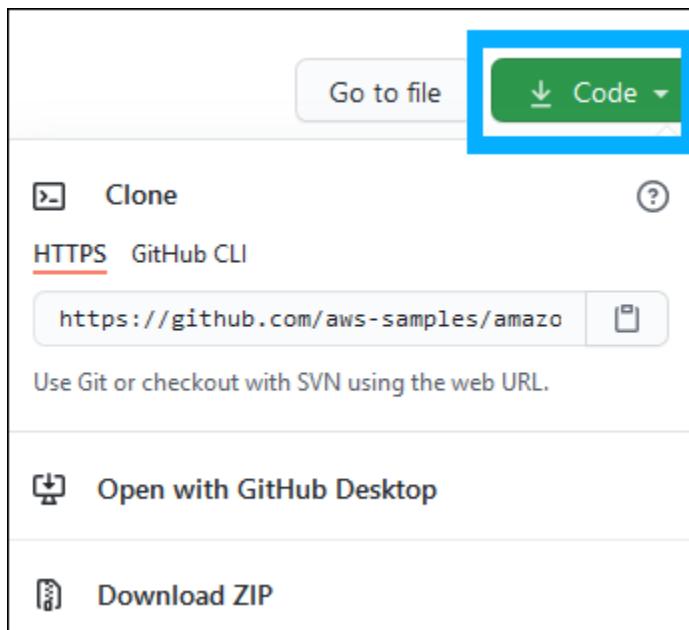
- Before starting this tutorial, confirm that your Windows virtual machine has Blender 2.8 or later installed.
- Launch a streaming session and sign in to a Windows virtual workstation, as outlined in [Launching a virtual workstation](#).
- We recommend that you have a basic knowledge of how to use Blender.

Step 1: Downloading assets

In this step, you will download Blender assets created by FuzzyPixel, an AWS creative team. These assets will help you test using assets in your studio. After downloading, you will place them into a project directory on your virtual workstation.

Download assets from GitHub

1. On your virtual workstation, open either **Google Chrome** or **Mozilla Firefox** and navigate to the [Nimble Studio Assets](#) on GitHub.
2. On the GitHub page, select the **Code** dropdown and choose **Download ZIP**.



3. In the window that pops up, select **Save File** and choose **OK**. Notice the location where you saved the file, so that you can find it for use in the next section.

Extract the assets

1. In **File Explorer**, navigate to the Z: drive and right-click to open the context menu, then select **New**, and choose **Folder**.
2. Name this folder nimbleAssets.
3. In **File Explorer**, navigate to where the **nimblestudio-demo-assets.zip** was saved. For example: C:\Users\your-user-name\Downloads\.
4. Right-click to open the context menu and choose **Extract All...**
5. When a pop-up window opens, select a location where you want to extract the assets. Then choose **Browse** and navigate to Z:\nimbleAssets\.
6. Choose **Select Folder**.
7. Choose **Extract**.

Step 2: Setting up the scene

In this step, you will open a lighting scene, link in a character, and enable the character to be posed.

Open the lighting scene

1. Choose the **Search Icon** in the lower left corner of your virtual workstation.
2. Enter **Blender** and choose **Blender**.
3. Nimble Studio provides a base lighting setup with lights and a background plane setup for your convenience. After Blender has launched, select **File** in the Blender menu and choose **Open**.
4. In the pop-up menu that opens, choose **Don't Save**.
5. When the **Blender File View** window opens, navigate to the lighting scene in the nimbleAssets folder that you created earlier. For example: Z:\nimbleAssets\shockinglyfuzzy\assets\lightRig.
6. Select **lighting_main.blend** and choose **Open**.

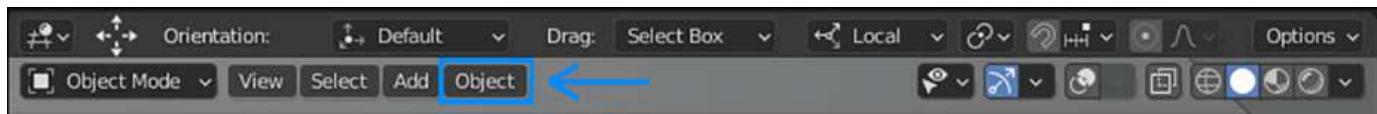
Import the Fuzzy Pixel character

In the following steps, you will be working with FuzzyPixel's mascot, the character Fuzzy Pixel.

1. In **Blender**, select **File** in the upper left of the application, and choose **Link**.
2. In the Blender File View window that pops up, navigate to Z:\nimbleAssets\nimblestudio-demo-assets\shockinglyfuzzy\assets\char\fuzzypixel\rig\.
3. Select the **fuzzypixel_rig.blend** file, and then choose **Link** in the lower-right corner of the window.
4. In the folder structure that opens, select **Collection**.
5. Select **rig_SET_fuzzypixel_hairy** and choose **Link**.

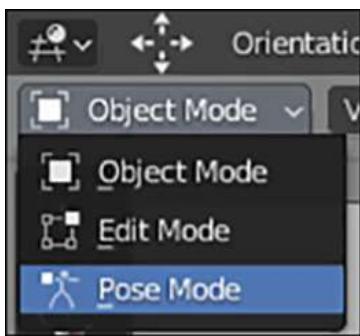
Pose the Fuzzy Pixel character

1. In the **Blender Outliner Window**, select **rig_SET_fuzzyPixel_hairy**.
2. In the **Header** menu of the camera view, select **Object**.



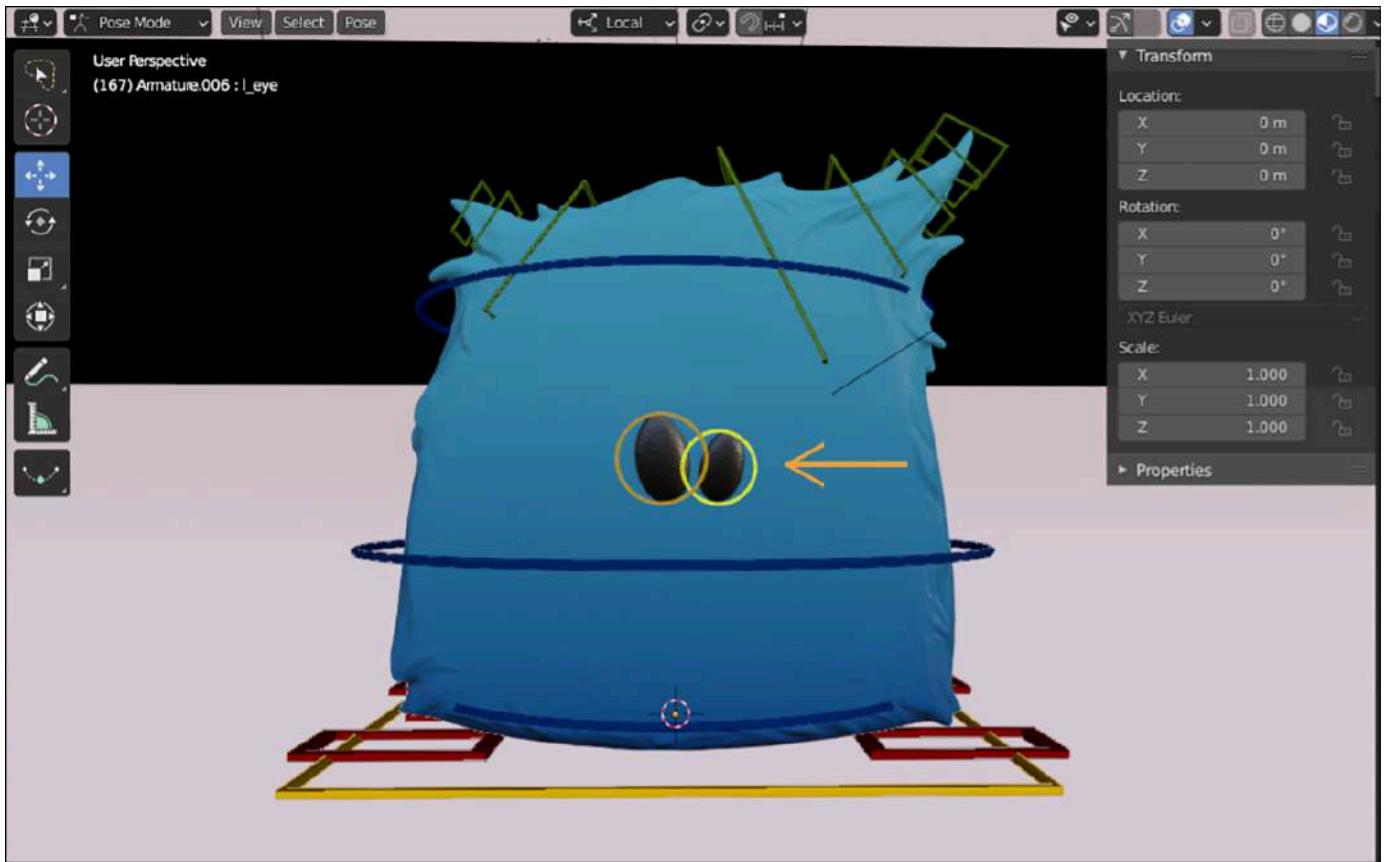
3. Now, select **Relations, Make Library Override...** and then choose **Make Library Override**. This will allow you to select the rig of the Fuzzy Pixel character so that you can pose it.
4. Select the **Armature** for Fuzzy Pixel

5. Next, in the **Header** menu of the camera view, select the interaction dropdown menu and choose **Pose Mode**.

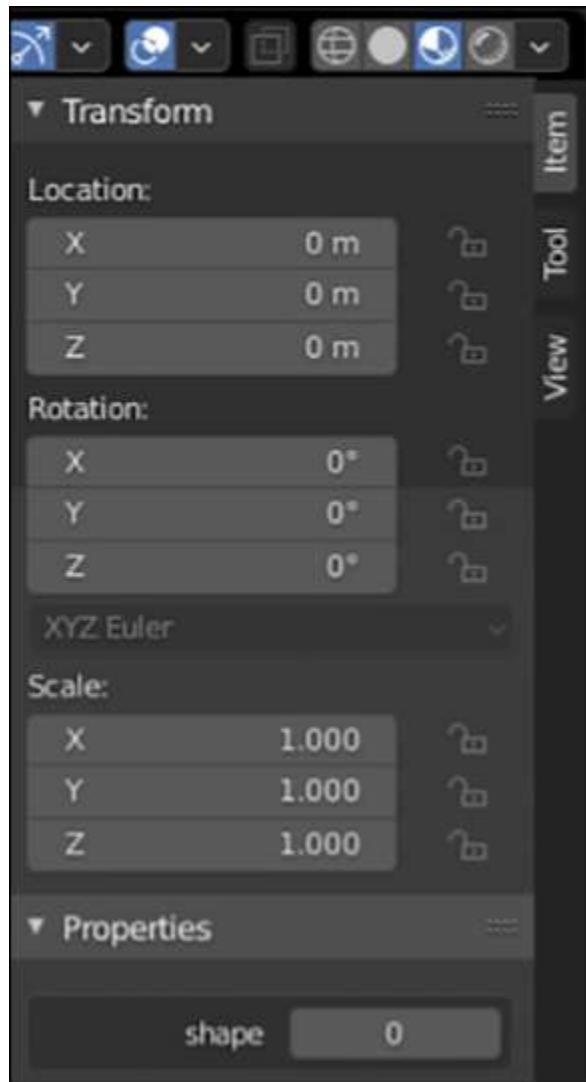


Now that your character is in **Pose Mode**, you can select the different controls of the armature and move them to pose the character.

6. To animate the character, first, choose the correct frame in the **Timeline** bar at the bottom of the window.
 - a. While in **Pose Mode**, with a control selected, open the **Insert Keyframe Menu** by entering capital **I**.
 - b. Then, to create a keyframe for the selected control at the current frame, choose **Location & Rotation**.
7. Advance to another frame further in the timeline, and then change the location or rotation of the keyframed control.
 - Enter capital **I** again, then choose **Location & Rotation** from the **Insert Keyframe** menu. This will insert another keyframe in the time line.
8. To see your animation, change the timeline to frame 1, and hit the **Play** button. Your animation will play and interpolate the keyframes you have placed. Exit **Pose Mode** and return to **Object Mode** after you're satisfied with your animation.
9. To change the shape of the eyes, select the eye control. To do so, select the circle named **L_eye** that goes around the eye.



10. Press **N** to toggle open the **Side Bar**. If the side bar is already open, this will close the window. If the side bar isn't visible, press **N** again to open it.
11. Open the **Properties** menu and change the value of the **shape** attribute to a value of 0-19. To add different expressions to the Fuzzy Pixel, you can choose from 20 different shapes.



After you animate the character, you can move on to the next step.

Step 3: Simulating the cloud and hair

Simulate the Hair

1. In the Blender Outline Window, select **m_fuzzyPixel_geo**.
2. In the Header Menu of the camera view, select **Object**.
3. Next, select **Relations, Make Local..., Selected Objects**. This will allow you to change the dynamic properties of the hair.
4. In the Blender **Properties** window, select the **Particle Properties** tab on the left. This will take you to the settings for the hair on the character.

5. There are two hair systems attached to the provided character, **hairLong** and **hairBase**. You will be simulating them both, and you can simulate them in either order.
6. Navigate to the section called **Hair Dynamics** and verify that the check box next to it is ticked. In the **Collisions** and **Structure** sections of the Hair Dynamics feature, you can change the dynamic properties of the hair.
7. After you've set the values that you want, navigate to **Cache** and set the **Simulation Start** at frame "1". We recommend that the **End** frame is no higher than "48" because the simulation will need to calculate to that frame.
8. Select **Bake**. Blender will now simulate the hair system with your current set values over the frame range selected. After your simulation is finished, you can move through the timeline to see the results.
9. If you're satisfied with the result, return to the top of the **Particle Properties** window and select the other hair system. Repeat these steps, ensuring that you match the **Simulation Start** and **End** frames.

Importing the cloud

1. To simulate the cloud, you will first need to import the cloud model and FX assets. Go to **File**. Then choose **Append**
2. In the Blender File View Window, navigate to Z:\nimbleAssets\nimblestudio-demo-assets\shockinglyfuzzy\assets\cloud\.
3. Select the **cloud.blend** file and choose **Append** in the lower right.
4. In the folder structure that opens, select **Collection**.
5. Then shift-select both **Cloud** and **Forces**.

Simulating the cloud

1. To animate the cloud, in the Outliner Window, select **Cloud** and follow the steps above to keyframe animation for it. You will be keyframing the cloud outside of pose mode because it doesn't have the same type of controls as the character.
2. When you're ready to simulate, in the **Outliner** window, select **Domain**. Then, in the **Properties** window, select the **Physics Properties** tab. This object is where you will determine

most of the dynamic properties of the volume simulation, as well as the bounds of the simulation itself.

- To increase the fidelity of the volume simulation, increase the **Resolution Divisions** property.

 **Note**

Increasing this value will increase the simulation time exponentially, so the default value of 64 is recommended for most test cases. To change how quickly the volume dissipates, enable **Dissolve**. The **Dissolve** value is measured in frames, assuming 24 frames per second.

3. To add or decrease the turbulence of the volume simulations, select any of the **Turbulence** objects from the **Forces** collection in the Outliner window and adjust their **Strength** and **Noise**. These have been made with some baseline values that will work in most situations
4. To simulate and view your volume simulation, navigate back to the **Domain** object and navigate to the **Properties** window to the **Cache** section.
 - Before you can simulate, first set a cache location, as well as a **Frame Start** and **End**. This is where you will save the individual cache frames.
5. Set the **Type** to **All**, and then select **Bake All**.
 - a. To allow simulation, first save the Blender file.
 - b. Blender will now simulate the volume. This can take some time, depending on the complexity of the simulation settings and the available CPU resources of your virtual workstation.

Step 4: Rendering an image sequence

In this step, you will learn how to render a series of images.

1. In the **Properties Window**, navigate to the **Output** tab and adjust the **Frame Start** and **End** to match the values for your animation and simulations.

2. In the **Output** section of the same tab, select a location to save your images by selecting the folder icon and in the new **File View** window.
3. Navigate to a location on the **Z:** drive where you want to save your images, and then hit **Accept**.
4. You can output a movie file by selecting one of the **Movie** settings in the **File Format** selection box.
5. In the **Info Window** at the top of the interface, select **Render**. Then select **Render Animation**.
6. The rendering time of the images will vary depending on the CPU resources available to your virtual workstation.



Providing feedback

Let us know what you think on the [GitHub repo](#).

Related resources

If you need help getting started with [Blender](#), from the basics to scripting, check out [Blender tutorials](#).

Migrating your studio

Use the following tutorials to help transition your studio and studio data. If you're not sure where to start, contact [Support](#).

Important

If you're a new Amazon Nimble Studio customer, see the [Nimble Studio Administrator Guide](#).

- [Migrating Amazon EBS volume data](#)

Migrating Amazon EBS volume data

This tutorial shows how to migrate Amazon Elastic Block Store (Amazon EBS) volume data for a single workstation session. You can use this volume to set up a new studio environment in another AWS service, such as Amazon EC2 or Amazon WorkSpaces.

EBS volumes are storage blocks that talk to your workstation partitions. You can reserve the storage, or only pay when you're using it. Nimble Studio uses this feature for [Session auto backup](#). If session auto backup isn't enabled and you terminate a workstation session, Nimble Studio automatically deletes the storage. EBS volumes are self-managed. This means that you can detatch the EBS volume from an EC2 instance, terminate that EC2 instance, and attach the persistent EBS volume to a new EC2 instance. For more information about Amazon EBS, see [Getting Started with Amazon EBS](#) and [Amazon Elastic Block Store \(Amazon EBS\)](#).

The tutorial steps provide two options for getting a copy of your data in an EBS volume. The following options aren't mutually exclusive. Choose the best solution or solutions for you.

Contents

- [Getting started](#)
- [Option 1: Manually copy data to an Amazon WorkDocs Drive](#)
- [Option 2: Copy files from EBS volume to Amazon FSx or Amazon EFS](#)
- [Option 3: \(EC2 only\) Create AMIs from a streaming session](#)
- [Option 4: \(EC2 only\) Turn on session auto backup and export snapshots](#)

Getting started

The following prerequisites are based on different use cases. Choose the use case that applies to your specific needs.

To migrate to Amazon EC2, complete the following tasks.

- Contact Nimble Studio [Support](#) and request a copy of your studio's Amazon Machine Images (AMIs).
- After you have a copy of your studio's AMIs, terminate all of your streaming sessions.

To migrate to WorkSpaces, complete the following tasks.

- Copy all of your data to a network attached storage (NAS) like Amazon FSx, Amazon Elastic File System (Amazon EFS), or an Amazon WorkDocs Drive.
- Stop all streaming sessions.
- (Optional) Complete the following tasks so that you have a copy of your data.
 - Contact Nimble Studio [Support](#) and request a copy of your studio's AMIs.
 - Contact the Nimble Studio team and request a copy of your studio's Amazon EBS snapshot. The shared AMIs and Amazon EBS snapshots will only be available for 90 days. Copy the resources into your account within the 90 day period or you will lose access to the shared AMIs and EBS snapshots.
- After you have a copy of your studio's AMIs and snapshots, terminate all of your streaming sessions.

Option 1: Manually copy data to an Amazon WorkDocs Drive

In this option, the studio administrator must create an Amazon WorkDocs Drive for the artists to manually copy their files to. Compared to [Option 1: Manually copy data to an Amazon WorkDocs Drive](#), this gives you better access control. Compared to [Option 3: \(EC2 only\) Create AMIs from a streaming session](#), the data can be accessed in any operating system.

Each Active Directory (AD) has 1TB of personal storage. AD users can't read or copy data from storage that doesn't belong to them. This method works for all destinations, including Amazon EC2 and WorkSpaces.

⚠️ Important

For more information about pricing, see the [Amazon WorkDocs pricing page](#).

Set up an Amazon WorkDocs for all studio users by following the instructions in [Getting started with Amazon WorkDocs](#).

- In **step 2**, choose **My sites**. On the **Manage your WorkDocs sites** page, choose **Create a WorkDocs site**.
- Choose **Select a directory**.
- In **step a of To use an existing directory**, select the existing AWS Managed Microsoft AD that StudioBuilder created for your studio in Nimble Studio.

After you create an Amazon WorkDocs Drive, studio users can [Get started with Amazon WorkDocs](#) and [install the Amazon WorkDocs drive](#) or use the [web client](#) on their new machine. They might require administrator permission to install the Amazon WorkDocs client on their machine. For more information about how to use Amazon WorkDocs, see [Using Amazon WorkDocs Drive](#).

After the installation is complete, your artists can manually copy and paste their data by following the instructions in [Using Amazon WorkDocs Drive](#).

If you encounter issues when copying system files to an Amazon WorkDocs Drive, do the following. Use [Option 3: \(EC2 only\) Create AMIs from a streaming session](#) to export the entire AMI.

Option 2: Copy files from EBS volume to Amazon FSx or Amazon EFS

This option allows your artists to manually copy the files from their workstation session's EBS volume. Artists can then add these files to Amazon FSx or Amazon EFS. This is convenient if you already mounted Amazon FSx or Amazon EFS to your studio. This option also works if you migrate to Amazon EC2 or to WorkSpaces.

Mount the Amazon FSx or Amazon EFS volume to Amazon EC2 or in WorkSpaces. Your artists can then copy their data from their workstation and paste the files into new instance or WorkSpace.

Option 3: (EC2 only) Create AMIs from a streaming session

This option makes a copy of all data from an artist's Amazon EBS volume without requiring them to manually back up their data. You can use these AMIs to set up Amazon EC2 instances for your artists to work on after migration.

You can't use snapshots or AMIs directly in WorkSpaces. You must first launch an instance and copy your data to another network attached storage (NAS) before you can download it to WorkSpaces.

Important

This option might cause an issue with the data integrity. This is because the AMI creation process might happen when your artists are still using their workstation. For example, if you download a file while AWS is backing up your data, the backing up process might back up part of the downloading file.

Follow the instructions in [Create an AMI from an Amazon EC2 Instance](#). Use the copy of the AMI that you received in the [Getting started](#).

To create an AMIs from a streaming session

1. Verify that your streaming session is **Stopped** or **Ready**.
2. Contact Nimble Studio [Support](#) and request a copy of your studio's Amazon Machine Images (AMIs).
3. (Optional) After you have a copy of your studio's AMIs, terminate all of your streaming sessions by following the instructions in [Start, stop, or terminate a workstation](#).

Option 4: (EC2 only) Turn on session auto backup and export snapshots

Similar to [Option 3: \(EC2 only\) Create AMIs from a streaming session](#), artists will have an entire copy of their data in their original Amazon EBS volume. Your artists won't need to manually back up their data. If you have multiple EBS volumes, you must attach those volumes one by one to the instance.

Note

This option is only for workstation sessions that already have [Session auto backup](#) enabled. This is because session auto backup is only applied to newly created workstations. If you turn on session auto backup, your workstations that are already running won't be affected.

You can't use snapshots or AMIs directly in WorkSpaces. You must first launch an instance and copy your data to another network attached storage (NAS) before you can download it to WorkSpaces.

To turn on session auto backup and export snapshots

1. Make sure that you have [Session auto backup](#) enabled.
 - a. To enable session auto backup, follow the instructions in [Turn on session auto backup](#).
 - b. After you turn on session auto backup, terminate all of your streaming sessions. Then you must start new sessions so that the new settings take effect. Terminating the session might cause data loss because the EBS volumes attached to the session are deleted when the session are terminated.
2. Stop your sessions.
3. Contact Nimble Studio [Support](#) and ask for the latest snapshot of the session.
 - a. Nimble Studio creates snapshots for all EBS volumes attached to the streaming sessions.
 - b. You can also request previously created backups by providing support with the timestamp of when the backup was created. You can find the timestamp in the Nimble Studio portal when you restore a session from a backup.
4. (Optional) To restore or access your data in the snapshot that you requested in step 2, follow the instructions in [Restoring from an Amazon EBS snapshot or an AMI](#).

Exporting AMIs and backups from streaming sessions

You can export Amazon Machine Images (AMIs) and backups from your streaming sessions. This tutorial shows how to request the export of AMIs and backups from Nimble Studio support.

Contents

- [Export streaming session AMI](#)
- [Export streaming session backup](#)

Export streaming session AMI

This section describes how to export data from a Nimble Studio streaming session. Nimble Studio support will help you back up all of the data in a **Stopped** or **Ready** streaming session. They will export and share an AMI that contains data in all volumes.

To export the AMI

1. Contact Nimble Studio [Support](#).
2. Provide the following information to the Nimble Studio support.
 - a. The AWS Region where your Nimble Studio cloud studio resides.
 - b. The ID of the session that you want to export.
 - i. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
 - ii. Choose **Launch profiles** in the left navigation pane. Then choose **Streaming sessions**.
 - iii. Choose the settings icon
).
 - c. Provide the Amazon Resource Name (ARN) of a AWS KMS key that belongs to you. This KMS key is used to encrypt the resulting snapshot that will be shared with you. At minimum, the KMS key must contain the following trust policy where `<region>` is replaced by the AWS Region that your studio resides in.

```
```json
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "nimble.<region>.amazonaws.com"
 },
 "Action": "kms:DescribeKey",
 "Resource": "*",
 "Condition": {

```

```
 "StringEquals": {
 "kms:ViaService": "ec2.<region>.amazonaws.com"
 }
 },
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "nimble.<region>.amazonaws.com"
 },
 "Action": "kms:CreateGrant",
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "kms:GrantIsForAWSResource": "true",
 "kms:ViaService": "ec2.<region>.amazonaws.com"
 },
 "ForAllValues:StringEquals": {
 "kms:GrantOperations": [
 "Decrypt",
 "Encrypt"
],
 "ForAnyValue:StringEquals": {
 "kms:EncryptionContextKeys": "aws:ebs:id"
 }
 }
 },
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "nimble.<region>.amazonaws.com"
 },
 "Action": "kms:GenerateDataKeyWithoutPlaintext",
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "kms:ViaService": "ec2.us-west-2.amazonaws.com"
 },
 "ForAnyValue:StringEquals": {
 "kms:EncryptionContextKeys": "aws:ebs:id"
 }
 }
 }
 }
}
```

```
]
}
...
```

3. After you provide this information, a Nimble Studio engineer will export the AMI into your account. When the export is complete, the engineer will provide the AMI ID or IDs that they shared to your account.

An AMI is accessible for 90 days after it's exported. During this time, you can copy the AMI by following the instructions in [Copy an AMI](#). After 90 days, the original AMI or AMIs are deleted and irrecoverable.

## Export streaming session backup

This section describes how to export data from a streaming session backup. Nimble Studio support will export and share an Amazon EBS snapshot that contains your data.

Before you can export a session backup, verify that the following information is true.

- The session has backups enabled. To check that the session has backups enabled, follow these steps:
  1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
  2. Choose **Launch profiles** in the left navigation pane.
  3. Open the launch profile that you used to create the session.
  4. If the **Auto backup** field isn't **Off**, then session backups are enabled. If the **Auto backup** field is **Off**, then backups aren't enabled.
    - a. To export session data for an existing session that doesn't have backups enabled, follow the instructions in [Export streaming session AMI](#). Terminating an existing session could cause data loss.
    - b. You can turn on backups by following the instructions in [Turn on session auto backup](#). When you enable backups, the setting will only take effect for any new sessions that are launched with that launch profile. It will not create backups for existing sessions.
- At least one backup exists for the session. For sessions with backups enabled, backups are created automatically every four hours. You can manually stop the session to create a backup. To view existing session backups, follow these steps:

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio manager** in the left navigation pane.
3. On the **Studio manager** page, choose **Go to Nimble Studio portal**.
4. Go to the **Sessions** page.
5. Select the vertical ellipsis  
(⋮) )

and choose **Restore from backup**. The earliest backups are automatically deleted when newer ones are created. Therefore, continued use of the session while waiting for the backup to be exported could cause the target backup to delete unexpectedly. Terminating the session will cause all existing backups for that session to be deleted.

After you verify the previous information, you can request an export of your backup.

### To export a streaming session backup

1. Contact Nimble Studio [Support](#).
2. Provide the following information to the Nimble Studio service team.
  - a. Provide the ID of the backup to be exported. This can only be retrieved by using the AWS CLI or SDK, or the ID of the session containing the backup to be exported and the exact timestamp of the backup to be exported. To retrieve the timestamp of the backup, follow these instructions.
    - i. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
    - ii. Choose **Launch profiles** in the left navigation pane. Then choose **Streaming sessions**.
    - iii. Select the vertical ellipsis  
(⋮) ).Then select **Restore from backup**.
  - b. Provide the Amazon Resource Name (ARN) of an AWS KMS key that belongs to you. This KMS key is used to encrypt the resulting snapshot that will be shared with you. At minimum, the KMS key must contain the following trust policy where `<region>` is replaced by the AWS Region that your studio resides in.

```
```json
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "nimble.<region>.amazonaws.com"
        },
        "Action": "kms:DescribeKey",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "kms:ViaService": "ec2.<region>.amazonaws.com"
            }
        }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "nimble.<region>.amazonaws.com"
        },
        "Action": "kms>CreateGrant",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "kms:GrantIsForAWSResource": "true",
                "kms:ViaService": "ec2.<region>.amazonaws.com"
            },
            "ForAllValues:StringEquals": {
                "kms:GrantOperations": [
                    "Decrypt",
                    "Encrypt"
                ]
            },
            "ForAnyValue:StringEquals": {
                "kms:EncryptionContextKeys": "aws:ebs:id"
            }
        }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "nimble.<region>.amazonaws.com"
        },
        "Action": "kms:GenerateDataKeyWithoutPlaintext",
```

```
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "kms:ViaService": "ec2.us-west-2.amazonaws.com"
            },
            "ForAnyValue:StringEquals": {
                "kms:EncryptionContextKeys": "aws:ebs:id"
            }
        }
    }
}
```
}
```

3. After you provide this information, a Nimble Studio engineer will export the backup into your account. When the export is complete, the engineer will provide the Amazon EBS snapshot ID or IDs that they shared to your account.

Snapshots are accessible for 90 days after they're exported. During this time, you can copy snapshots by following the instructions in [Copy an Amazon EBS snapshot](#). After 90 days, the original snapshots are deleted and irrecoverable.

## Setting up EC2 instances like workstations

This tutorial describes how to set up an Amazon EC2 instance that is similar to a Nimble Studio Windows or Linux workstation.

### Contents

- [Step 1: Prepare the subnets](#)
- [Step 2: Prepare the security groups](#)
- [Step 3: \(Linux Only\) Prepare AWS Managed Microsoft AD for a seamless domain join](#)
- [Step 4: Prepare the AMI](#)
- [Step 5: Create an EC2 instance](#)
- [Step 6: Validate that your instance is joined to the Active Directory](#)
- [Troubleshooting](#)

## Step 1: Prepare the subnets

This tutorial creates an Amazon EC2 instance in the **Public** subnet. This makes it simpler to connect to the RDP or the NICE DCV client from the external public internet.

The **Public** subnet that StudioBuilder creates has all required ports open already. For security on your instance, you must restrict access to the **Public** subnet to your IP address.

You must also update the network ACL rules of the following subnets:

- Open the **FileSystems** subnet to inbound traffic from the **Public** subnet. Duplicate the current allowed ports of the **WorkStations** subnet from **Public**.
- Open the **ActiveDirectory** subnet to all inbound traffic from the **Public** subnet. For security purposes, restrict the ports that are listed in [Configure your VPC subnets and security groups](#).
- Allow all outbound traffic from the **ActiveDirectory** subnet.

The Nimble Studio **Public** subnet has a CIDR mask of 27. This means it is 10.0.0.0/27 and has room for 27 IP addresses. To support more instances, create an additional public subnet, and use the same [network ACL](#) and [route tables](#).

Alternatively, to use a private subnet, you can use the **Workstations** subnet. For instructions about using DCV over a private subnet, see [Getting started with managing NICE DCV sessions secured behind a NICE DCV Connection Gateway](#) for Windows. Also, follow the instructions in [Setting up NICE DCV Session Manager](#) for Linux.

## Step 2: Prepare the security groups

Allow connecting over RDP or DCV by creating a security group for Amazon EC2 workstations. You can create a security group in the existing virtual private cloud (VPC) that Nimble Studio created by following the instructions in [Create a security group](#).

- In *step 4*, name the security group **EC2 workstations**.
- In *step 6*, add the following rules.

| Inbound/Outbound | Type        | Protocol | Port range | Source/Destination | Notes                              |
|------------------|-------------|----------|------------|--------------------|------------------------------------|
| Inbound          | Custom TCP  | TCP      | 3389       | Your IP address    | Only add this rule if you use RDP. |
| Inbound          | Custom TCP  | TCP      | 8443       | Your IP address    | Only add this rule if you use DCV. |
| Inbound          | Custom UDP  | UDP      | 8443       | Your IP address    | Only add this rule if you use DCV. |
| Outbound         | All traffic | All      | All        | 0.0.0.0/0          |                                    |

## Step 3: (Linux Only) Prepare AWS Managed Microsoft AD for a seamless domain join

 **Important**

This step is only for seamlessly joining a Linux instance to the domain.

To prepare AWS Managed Microsoft AD for a seamless join, see [Seamlessly join a Linux EC2 instance to your AWS Managed Microsoft AD directory](#).

## Step 4: Prepare the AMI

You can request an exported Amazon Machine Image (AMI) that is shared by the Nimble Studio service team, or create your own AMI and install DCV on it.

### Request an AMI

This option is to request a AMI that the Nimble Studio team creates. You will ask the Support team to create an AMI from your running streaming session. Then they will share this AMI with you.

To request an AMI, follow the instructions in [Option 3: \(EC2 only\) Create AMIs from a streaming session.](#)

## Choose a new AMI

To build a new AMI, you can use one of the following AMIs as the base AMI. Then you can install DCV onto the AMI.

- Windows: Nimble Studio offers both Windows 2019 and 2022 Server workstations.
  - Microsoft Windows Server 2019 Base
  - Microsoft Windows Server 2022 Base
- Linux: Nimble Studio Linux workstations are CentOS 7.
  - CentOS 7 (x86\_64) - with Updates HVM

Follow the instructions in [Installing the NICE DCV Server](#) to install the DCV server on an instance. Choose the correct guide for your operating system.

## Step 5: Create an EC2 instance

Follow the instructions in [Launch an instance using defined parameters](#).

- For **Application and OS Images (Amazon Machine Image)**, you have two options. You can use the AMI that the Nimble Studio team shared with you, or create your own AMI and install DCV onto the AMI. For more information, see [Step 4: Prepare the AMI](#).
- For **Instance type**, choose the default instance type.
  - To use this instance type, you can request an Amazon EC2 service quota increase. For instructions, see [Request a quota increase for On-Demand Instances \(G and VT\)](#).
- For **Key pair (login)**, create or select an existing RSA key for the instance.
- For **Network settings**, enter the following information:
  - **VPC**: Choose the VPC that StudioBuilder created. It has the same name as your studio ID.
  - **Subnet**: Select the subnet that you prepared in [Step 1: Prepare the subnets](#) in the primary Availability Zone of the Nimble Studio VPC.
  - **Auto-assign public IP**: Choose **Enable**.
  - **Firewall (security groups)**: Choose **Select existing security group** and select **EC2 workstations** and **WorkstationEgress**. EC2 workstations is the security group that you built in [Step 2: Prepare the security groups](#).

- For **Advanced Details**, enter the following information:
  - Configure the **Domain join directory** by selecting the Active Directory that StudioBuilder created.
  - Configure the Audit Manager instance profile. Create and select the following IAM instance profile for the workstation. We recommend that you create a new role by following the instructions in [Creating a role for an AWS service](#). If you're using Linux seamless domain join, use the **LinuxEC2DomainJoin** role that you created in step 3.
    - In step 4, choose **EC2**.
    - In step 5, to allow the instance to seamlessly join the domain, choose the **AmazonSSMManagedInstanceCore** and **AmazonSSMDirectoryServiceAccess** policies.
    - In step 8, enter **Domain Joining**.

## Step 6: Validate that your instance is joined to the Active Directory

This last step shows how to validate that your instance is joined to the Active Directory.

### Windows

1. Follow the instructions in [Connecting to a NICE DCV session](#).
2. Open a command prompt.
  - a. Enter **Windows Key + R** and then enter **cmd** in the *Run window* that appears.
  - b. Search the start menu for **Command Prompt**.
3. Enter **systeminfo | findstr /B "Domain"** in the command prompt window, and press the enter (or return) key.
  - a. If you see something other than **WORKGROUP**, then you are likely joined to a domain.
  - b. If you aren't joined to a domain, you will see **Domain: WORKGROUP**.

### Linux

1. Follow the instructions in [Connecting to a NICE DCV session](#).
2. Open a command prompt.
3. Run the following command: **realm list**
4. The domain name should be returned in the output.

## Troubleshooting

If you can't seamlessly join the Active Directory, follow the instructions in [How do I use Amazon EC2 Systems Manager to join a running EC2 Windows instance to my AWS Directory Service domain?](#).

# Security in Amazon Nimble Studio

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS Compliance Programs](#). To learn about the compliance programs that apply to Amazon Nimble Studio, see [AWS Services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

## Important

It's highly recommended that you read and familiarize yourself with the [Security Pillar - AWS Well-Architected Framework](#). This article contains key principles to securing your AWS infrastructure.

This documentation helps you understand how to apply the shared responsibility model when using Nimble Studio. The following topics show you how to configure Nimble Studio to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Nimble Studio resources.

## More Information

- [Security Pillar - AWS Well-Architected Framework](#)
- [Security for the AWS Cloud Development Kit \(AWS CDK\) \(AWS CDK\)](#)
- [Security in Amazon Virtual Private Cloud](#)
- [AWS security credentials](#)

- Security in Amazon EC2
  - [Linux](#)
  - [Windows](#)

## Data protection in Amazon Nimble Studio

The AWS [shared responsibility model](#) applies to data protection in Amazon Nimble Studio. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the [Data Privacy FAQ](#). For information about data protection in Europe, see the [AWS Shared Responsibility Model and GDPR](#) blog post on the [AWS Security Blog](#).

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Nimble Studio or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

The AWS [shared responsibility model](#) applies to data protection in Amazon Nimble Studio. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You're responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use.

For more information about data privacy, see the [Data Privacy FAQ](#). For information about data protection in the European Union, visit the [GDPR Center](#).

## Encryption at rest

Nimble Studio protects sensitive studio data by encrypting it at rest using encryption keys stored in [AWS Key Management Service \(AWS KMS\)](#). Encryption at rest is available in all AWS Regions where Nimble Studio is available. The studio data that we encrypt includes the name and descriptions of all resource types, as well as studio component scripts, script parameters, mount points, share names, and other data.

Encrypting data means that sensitive data that is saved on disks isn't readable by any user or application without a valid key. Encrypted data can be securely stored at rest and can be decrypted only by a party with authorized access to the managed key.

For information about how Nimble Studio uses AWS KMS for encrypting data at rest, see [Key management for Amazon Nimble Studio](#).

## Using grants with AWS KMS keys

A grant is a policy instrument that allows [AWS principals](#) to use AWS KMS keys in cryptographic operations. It can also let them view a KMS key with the command `DescribeKey`, and create and manage grants.

Grants are commonly used by AWS services that integrate with AWS KMS to encrypt your data at rest. The service creates a grant on behalf of a user in the account, uses its permissions, and retires the grant as soon as its task is complete.

When Nimble Studio creates your studio, we provide two roles for Nimble Studio portal users: user and administrator roles. Nimble Studio creates grants on customer managed keys for these roles to provide them access to studio encrypted data.

**⚠️ Important**

If you delete a grant, the Nimble Studio portal will be unusable for users, until the administrator creates a new grant.

For details about how AWS services use grants, see [How AWS services use AWS KMS or the Encryption at rest](#) topic in the service's user guide or developer guide.

## Encryption in transit

The following table provides information about how data is encrypted in transit. Where applicable, other data protection methods for Nimble Studio are also listed.

| Data                                           | Network path                                                                                                                                  | Protection                                                                                              |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| Web assets such as images and JavaScript files | The network path is between Nimble Studio users and Nimble Studio.                                                                            | Data encryption uses TLS 1.2 or later.                                                                  |
| Pixel and related streaming traffic            | The network path is between Nimble Studio users and Nimble Studio.                                                                            | Encrypted using 256-bit Advanced Encryption Standard (AES-256), and transported using TLS 1.2 or later. |
| API traffic                                    | The path is between Nimble Studio users and Nimble Studio.                                                                                    | Encrypted using TLS 1.2 or later. Requests to create a connection are signed using SigV4.               |
| Nimble Studio managed traffic                  | The path is between AWS services and resources in your AWS account, Nimble Studio streaming instances, and Nimble Studio management services. | Encrypted using TLS 1.2; requests to create a connection are signed using SigV4, where applicable.      |

## Application Access

By default, Nimble Studio enables the applications that you specify in your Amazon Machine Image (AMI) to launch other applications and executable files on the image builder and fleet instance. This means that applications with dependencies on other applications (for example, an application that launches the browser to navigate to a product website) function as expected.

Make sure that you configure your administrative controls, security groups, and other security software to grant users the minimum permissions required to access resources and transfer data between their local computers and fleet instances.

Nimble Studio uses [NICE DCV](#) to securely deliver remote desktops and application streaming from any cloud or data center to any device, over varying network conditions. A Transport Layer Security (TLS) is used to encrypt sensitive data in transit between your user's clients and their streaming instances.

Amazon FSx data encryption is the responsibility of the customer. For more information, view [Amazon FSx documentation](#).

## Key management for Amazon Nimble Studio

When creating a new studio, you can choose one of the following keys to encrypt your studio data:

- AWS owned KMS key – Default encryption type. The key is owned by Nimble Studio (no additional charge).
- Customer managed KMS key – The key is stored in your account and is created, owned, and managed by you. You have full control over the key. AWS KMS charges apply.

### Important

If you delete your customer managed KMS key that was used on a custom AMI, the AMI is unrecoverable.

Deleting a customer managed KMS key in AWS Key Management Service (AWS KMS) is destructive and potentially dangerous. It irreversibly deletes the key material and all metadata associated with the key. After a customer managed KMS key is deleted, you can no longer decrypt the data that was encrypted by that key. This means that the data becomes unrecoverable.

This is why AWS KMS gives customers a waiting period of up to 30 days before deleting the key. The default waiting period is 30 days.

## About the waiting period

Because it's destructive and potentially dangerous to delete a customer managed KMS key, we require you to set a waiting period of 7 – 30 days. The default waiting period is 30 days.

However, the actual waiting period might be up to 24 hours longer than the one you scheduled. To get the actual date and time when the key will be deleted, use the [DescribeKey](#) operation. You can also see the scheduled deletion date of a key in the [AWS KMS console](#) on the key's detail page, in the **General configuration** section. Notice the time zone.

During the waiting period, the customer managed key's status and key state is **Pending deletion**.

- A customer managed KMS key that is pending deletion can't be used in any [cryptographic operations](#).
- AWS KMS doesn't [rotate the backing keys](#) of customer managed AWS KMS keys that are pending deletion.

For more information about deleting a customer managed AWS KMS key see [Deleting customer master keys](#).

## Data security measures

For data protection purposes, we recommend that you protect AWS account credentials and set up individual accounts with AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put sensitive identifying information, such as customer account numbers, into free-form fields such as a **Name** field. This includes when you work with Amazon Nimble Studio or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into Amazon Nimble Studio or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

## Diagnostic data and metrics

During the deployment and deletion of StudioBuilder, Amazon Nimble Studio collects certain metrics that we use to diagnose issues and improve Nimble Studio's features and user experience.

### Types of metrics collected

- Usage information – The generic commands and subcommands that are run.
- Errors and diagnostic information – The status and duration of commands that are run, including exit codes, internal exception names, and failures.
- System and environment information – The Python version, operating system (Windows, Linux, or macOS), and environment in which StudioBuilder is run.

## Identity and Access Management for Amazon Nimble Studio

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. Administrators control who can be **authenticated** (signed in) and **authorized** (have permissions) to use Amazon Nimble Studio resources. IAM is an AWS service that you can use with no additional charge.

### Topics

- [Audience](#)
- [Authenticating with identities](#)
- [Managing access using policies](#)
- [How Amazon Nimble Studio works with IAM](#)
- [Identity-based policy examples for Amazon Nimble Studio](#)
- [AWS managed policies for Amazon Nimble Studio](#)
- [Cross-service confused deputy prevention](#)
- [Troubleshooting Amazon Nimble Studio identity and access](#)

## Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Nimble Studio.

**Service user** – If you use the Nimble Studio service to do your job, then you are a service user. In this case, your administrator will provide you with the credentials and permissions that you need to access your assigned resources. As you use more Nimble Studio features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you can't access a feature in Nimble Studio, see [Troubleshooting Amazon Nimble Studio identity and access](#).

**Service administrator** – If you're in charge of Nimble Studio resources at your company, you probably have full access to Nimble Studio. It's your job to determine which Nimble Studio features and resources your employees should access. Then, submit requests to your administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Nimble Studio, see [How Amazon Nimble Studio works with IAM](#).

**Administrator** – If you're an administrator, you can learn details about how you can write policies to manage access to Nimble Studio. To view example Nimble Studio identity-based policies that you can use in IAM, see [Identity-based policy examples for Amazon Nimble Studio](#).

## Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see [Signing in to the AWS Management Console as an IAM user or root user](#) in the [IAM User Guide](#).

You need to be **authenticated** (signed in to AWS) as the AWS account root user, a user, or by assuming an IAM role. You can also use your company's single sign-on authentication or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you're assuming a role indirectly.

To sign in directly to the [AWS Management Console](#), use your password with your root user email address or your username. You can access AWS programmatically using your root user or user access keys.

**AWS provides SDK and command line tools to cryptographically sign your request using your credentials.** If you don't use AWS tools, sign the request yourself. Do this using **Signature Version 4**, a protocol for authenticating inbound API requests. For more information about authenticating requests, see [Signature Version 4 signing process](#) in the AWS General Reference .

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see [Using multi-factor authentication \(MFA\) in AWS](#) in the IAM User Guide.

## AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the **AWS account root user** and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the [best practice of using the root user only to create your first IAM user](#). Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

## Users and groups

A [user](#) is an identity within your AWS account that has specific permissions for a single person or application. A user can have long-term credentials or a set of access keys. To learn how to generate access keys, see [Managing access keys for IAM users](#) in the IAM User Guide. When you generate access keys for a user, view and securely save the key pair. You can't recover the secret access key in the future. Instead, generate a new access key pair.

An [IAM group](#) is an identity that specifies a collection of users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named **IAMAdmins** and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see [When to create a user \(instead of a role\)](#) in the IAM User Guide.

## IAM roles

An [IAM role](#) is an identity within your AWS account that has specific permissions. It is similar to a user, but isn't associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by [switching roles](#). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see [Using IAM roles](#) in the IAM User Guide.

IAM roles with temporary credentials are useful in the following situations:

- **Temporary user permissions** – A user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – Instead of creating a user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as **federated users**. AWS assigns a role to a federated user when access is requested through an [identity provider](#). For more information about federated users, see [Federated users and roles](#) in the IAM User Guide.
- **Membership** – Nimble Studio uses a concept called 'membership' to provide a user access to a particular launch profile. Membership allows studio administrators to delegate resource access to users, without having to write, or understand, IAM policies. When a Nimble Studio administrator creates a membership for a user in a launch profile, the user is authorized to perform IAM actions that are required to use a launch profile, such as viewing its properties and starting a streaming session using that launch profile.
- **Service role** – A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. Service roles provide access only within your account and can't be used to grant access to services in other accounts. An administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the IAM User Guide.
- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Nimble Studio doesn't support service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using](#)

[an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the IAM User Guide.

To learn whether to use IAM roles or users, see [When to create an IAM role \(instead of a user\)](#) in the IAM User Guide.

## Managing access using policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. You can sign in as the root user or a user, or you can assume an IAM role. When you then make a request, AWS evaluates the related identity-based or resource-based policies. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see [Overview of JSON policies](#) in the IAM User Guide.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and for what **conditions**.

Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

### Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as a user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and for what conditions. To learn how to create an identity-based policy, see [Creating IAM policies](#) in the IAM User Guide.

Identity-based policies can be further categorized as **inline policies** or **managed policies**. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone

policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see [Choosing between managed policies and inline policies](#) in the IAM User Guide.

## Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are **IAM role trust policies** and **Amazon S3 bucket policies**. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and for what conditions. [Specify a principal](#) in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

## Access control lists (ACLs) in Nimble Studio

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they don't use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see [Access control list \(ACL\) overview](#) in the **Amazon Simple Storage Service Developer Guide**.

## Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of the entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the Principal field aren't limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see [Permissions boundaries for IAM entities](#) in the IAM User Guide.

- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in Organizations. Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see [How SCPs work](#) in the AWS Organizations User Guide.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see [Session policies](#) in the IAM User Guide.

## Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see [Policy evaluation logic](#) in the IAM User Guide.

## How Amazon Nimble Studio works with IAM

Before you use IAM to manage access to Nimble Studio, learn what IAM features are available to use with Nimble Studio.

### IAM features you can use with Amazon Nimble Studio

| IAM feature                                                  | Nimble Studio support |
|--------------------------------------------------------------|-----------------------|
| <a href="#">Identity-based policies for Nimble Studio</a>    | Yes                   |
| <a href="#">Resource-based policies within Nimble Studio</a> | No                    |
| <a href="#">Policy actions for Nimble Studio</a>             | Yes                   |
| <a href="#">Policy resources for Nimble Studio</a>           | Yes                   |
| <a href="#">Policy condition keys for Nimble Studio</a>      | Yes                   |
| <a href="#">Access control lists (ACLs) in Nimble Studio</a> | No                    |

| IAM feature                                                              | Nimble Studio support |
|--------------------------------------------------------------------------|-----------------------|
| <a href="#">Attribute-based access control (ABAC) with Nimble Studio</a> | Yes                   |
| <a href="#">Using temporary credentials with Nimble Studio</a>           | Yes                   |
| <a href="#">Cross-service principal permissions for Nimble Studio</a>    | Yes                   |
| <a href="#">Service roles for Nimble Studio</a>                          | Yes                   |
| <a href="#">Service-linked roles for Nimble Studio</a>                   | No                    |

To get a high-level view of how Nimble Studio and other AWS services work with most IAM features, see [AWS services that work with IAM](#) in the IAM User Guide.

## Identity-based policies for Nimble Studio

|                                  |     |
|----------------------------------|-----|
| Supports identity-based policies | Yes |
|----------------------------------|-----|

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as a user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and for what conditions. To learn how to create an identity-based policy, see [Creating IAM policies](#) in the IAM User Guide.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions for which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it's attached. To learn about all of the elements that you can use in a JSON policy, see [IAM JSON policy elements reference](#) in the IAM User Guide.

## Identity-based policy examples for Amazon Nimble Studio

To view examples of Nimble Studio identity-based policies, see [Identity-based policy examples for Amazon Nimble Studio](#).

## Resource-based policies within Nimble Studio

|                                  |    |
|----------------------------------|----|
| Supports resource-based policies | No |
|----------------------------------|----|

Nimble Studio doesn't support resource-based policies or cross-account access. Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM **role trust policies** and Amazon S3 **bucket policies**. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and for what conditions. [Specify a principal](#) in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

## Policy actions for Nimble Studio

|                         |     |
|-------------------------|-----|
| Supports policy actions | Yes |
|-------------------------|-----|

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and for what **conditions**.

The Action element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as **permission-only actions** that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called **dependent actions**.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Nimble Studio actions, see [Actions defined by Amazon Nimble Studio](#) in the [Service Authorization Reference](#).

Policy actions in Nimble Studio use the following prefix before the action:

nimble

To specify multiple actions in a single statement, separate them with commas.

"Action": [

```
"nimble:action1",
"nimble:action2"
]
```

To view examples of Nimble Studio identity-based policies, see [Identity-based policy examples for Amazon Nimble Studio](#).

## Policy resources for Nimble Studio

|                           |     |
|---------------------------|-----|
| Supports policy resources | Yes |
|---------------------------|-----|

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and for what **conditions**.

The Resource JSON policy element specifies the object or objects to which the action applies. Statements must include either a Resource or a NotResource element. As a best practice, specify a resource using its [Amazon Resource Name \(ARN\)](#). You can do this for actions that support a specific resource type, known as **resource-level permissions**.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (\*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To view examples of Nimble Studio identity-based policies, see [Identity-based policy examples for Amazon Nimble Studio](#).

## Policy condition keys for Nimble Studio

|                                |     |
|--------------------------------|-----|
| Supports policy condition keys | Yes |
|--------------------------------|-----|

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and for what **conditions**.

The Condition element (or Condition`block) lets you specify conditions in which a statement is in effect. The `Condition element is optional. You can create conditional expressions that use [condition operators](#), such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple Condition elements in a statement, or multiple keys in a single Condition element, AWS evaluates them using a logical AND operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an user permission to access a resource only if it's tagged with their username. For more information, see [IAM policy elements: variables and tags](#) in the [IAM User Guide](#).

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see [AWS global condition context keys](#) in the [IAM User Guide](#).

To view examples of Nimble Studio identity-based policies, see [Identity-based policy examples for Amazon Nimble Studio](#).

## Access control lists (ACLs) in Nimble Studio

|               |    |
|---------------|----|
| Supports ACLs | No |
|---------------|----|

Nimble Studio doesn't support access control lists (ACLs). ACLs control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they don't use the JSON policy document format.

## Attribute-based access control (ABAC) with Nimble Studio

|                                  |     |
|----------------------------------|-----|
| Supports ABAC (tags in policies) | Yes |
|----------------------------------|-----|

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called **tags**. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they're trying to access.

To control access based on tags, you provide tag information in the [condition element](#) of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

For more information about ABAC, see [What is ABAC?](#) in the IAM User Guide. To view a tutorial with steps for setting up ABAC, see [Use attribute-based access control \(ABAC\)](#) in the IAM User Guide.

## Using temporary credentials with Nimble Studio

|                                |     |
|--------------------------------|-----|
| Supports temporary credentials | Yes |
|--------------------------------|-----|

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see [AWS services that work with IAM](#) in the [IAM User Guide](#).

You're using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see [Switching to a role \(console\)](#) in the IAM User Guide.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see [Temporary security credentials in IAM](#).

## Cross-service principal permissions for Nimble Studio

|                                |     |
|--------------------------------|-----|
| Supports principal permissions | Yes |
|--------------------------------|-----|

## Service roles for Nimble Studio

|                        |     |
|------------------------|-----|
| Supports service roles | Yes |
|------------------------|-----|

A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. Service roles provide access only within your account and can't be used to grant access to services in other accounts. An administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the IAM User Guide.

**⚠️ Warning**

Changing the permissions for a service role might break Nimble Studio functionality. Edit service roles only when Nimble Studio provides guidance to do so.

## Service-linked roles for Nimble Studio

|                               |    |
|-------------------------------|----|
| Supports service-linked roles | No |
|-------------------------------|----|

Nimble Studio doesn't support service-linked roles. A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see [AWS services that work with IAM](#). Find a service in the table that includes a Yes in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

## Identity-based policy examples for Amazon Nimble Studio

By default, users and roles don't have permission to create or modify Nimble Studio resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An administrator must create IAM policies that grant users and roles permission to perform actions on the resources that they need. The administrator must then attach those policies to the users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see [Creating policies on the JSON tab](#) in the IAM User Guide.

### Topics

- [Policy best practices](#)
- [Using the Nimble Studio console](#)
- [Allow users to view their own permissions](#)

## Policy best practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete Nimble Studio resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started using AWS managed policies** – To start using Nimble Studio quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see [Get started using permissions with AWS managed policies](#) in the IAM User Guide.
- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see [Grant least privilege](#) in the **IAM User Guide**.
- **Enable MFA for sensitive operations** – For extra security, require users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see [Using multi-factor authentication \(MFA\) in AWS](#) in the IAM User Guide.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions that your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see [IAM JSON policy elements: Condition](#) in the IAM User Guide.

## Using the Nimble Studio console

To access the **Nimble Studio console**, you need a minimum set of permissions. To use the **Nimble Portal**, you'll also need a set of permissions.

These permissions must allow you to list and view details about the Nimble Studio resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you're trying to perform.

Attach the Nimble Studio ConsoleAccess or ReadOnly AWS managed policy to the entities so that users and roles can still use the Nimble Studio console. For more information, see [Adding permissions to a user](#) in the IAM User Guide.

## Allow users to view their own permissions

This example shows how you might create a policy that allows users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ViewOwnUserInfo",
 "Effect": "Allow",
 "Action": [
 "iam:GetUserPolicy",
 "iam>ListGroupsForUser",
 "iam>ListAttachedUserPolicies",
 "iam>ListUserPolicies",
 "iam GetUser"
],
 "Resource": ["arn:aws:iam::*:user/${aws:username}"]
 },
 {
 "Sid": "NavigateInConsole",
 "Effect": "Allow",
 "Action": [
 "iam:GetGroupPolicy",
 "iam GetPolicyVersion",
 "iam GetPolicy",
 "iam>ListAttachedGroupPolicies",
 "iam>ListGroupPolicies",
 "iam>ListPolicyVersions",
 "iam>ListPolicies",
 "iam>ListUsers"
],
 }
]
}
```

```
 "Resource": "*"
 }
]
}
```

## AWS managed policies for Amazon Nimble Studio

To add permissions to users, groups, and roles, it's easier to use AWS managed policies than to write policies yourself. It takes time and expertise to [create IAM customer managed policies](#) that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see [AWS managed policies](#) in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services don't remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see [AWS managed policies for job functions](#) in the *IAM User Guide*.

Your end users will access Amazon Nimble Studio primarily using the Nimble Studio portal. When creating your studio using StudioBuilder or the Nimble Studio console, one IAM role is created for each studio persona: the studio administrator and the studio user. Each has the respective IAM managed policy attached. The Nimble Studio portal provides an experience where users can only list and use the resources that they have permission to access.

The Nimble Studio portal provides an experience where users can only list and use the resources to which they have access and the portal depends on the content of these policies to operate correctly. Nimble Studio end users will use the portal to access their cloud studio. So, when admins create their studio using StudioBuilder, one IAM role is created for each person who needs to access the studio. This includes the studio administrator and the studio user, each with their respective IAM managed policy attached.

For a list and descriptions of job function policies, see [AWS managed policies for job functions](#) in the IAM User Guide.

## AWS managed policy: AmazonNimbleStudio-LaunchProfileWorker

You can attach the [AmazonNimbleStudio-LaunchProfileWorker](#) policy to your IAM identities.

Attach this policy to EC2 instances created by Nimble Studio Builder to grant access to resources needed by Nimble Studio launch profile workers.

### Permissions details

This policy includes the following permissions.

- ds - Allows LaunchProfile workers to discover connection information about the AWS Managed Microsoft AD associated with a LaunchProfile.
- ec2 - Allows LaunchProfile workers to discover security group and subnet information for connecting to a LaunchProfile.
- fsx - Allows LaunchProfile workers to discover connection information to Amazon FSx volumes associated with a LaunchProfile.

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ec2:DescribeNetworkInterfaces",
 "ec2:DescribeSecurityGroups",
 "fsx:DescribeFileSystems",
 "ds:DescribeDirectories"
],
 "Resource": [
 "*"
],
 "Condition": {
 "StringEquals": {
 "aws:CalledViaLast": "nimble.amazonaws.com"
 }
 },
 "Sid": "GetLaunchProfileInitializationDependencies"
 }
]
}
```

```
],
 "Version": "2012-10-17"
}
```

## AWS managed policy: AmazonNimbleStudio-StudioAdmin

You can attach the [AmazonNimbleStudio-StudioAdmin](#) policy to your IAM identities.

Attach this policy to the Admin role associated with your studio to grant access to Amazon Nimble Studio resources associated with the studio administrator and related studio resources in other services.

## Permissions details

This policy includes the following permissions.

- nimble - Allows Studio Users access to Nimble resources that have been delegated to them by StudioAdmins.
  - sso - Allows Studio Users the ability to view the names of other users in the studio.
  - identitystore - Allows Studio Users the ability to view the names of other users in the studio.
  - ds - Allows Nimble Studio to add virtual workstations to the AWS Managed Microsoft AD associated with the studio.
  - ec2 - Allows Nimble Studio to attach virtual workstations to your configured VPC.
  - fsx - Allows Nimble Studio to connect virtual workstations to your configured Amazon FSx volumes.
  - cloudwatch - Allows Nimble Studio to retrieve CloudWatch metrics.

```
"nimble>DeleteStreamingSession",
"nimble>ListStreamingSessionBackups",
"nimble>GetStreamingSessionBackup",
"nimble>ListEulas",
"nimble>ListEulaAcceptances",
"nimble>GetEula",
"nimble>AcceptEulas",
"nimble>ListStudioMembers",
"nimble>GetStudioMember",
"nimble>ListStreamingSessions",
"nimble>GetStreamingImage",
"nimble>ListStreamingImages",
"nimble>GetLaunchProfileInitialization",
"nimble>GetLaunchProfileDetails",
"nimble>GetFeatureMap",
"nimble>PutStudioLogEvents",
"nimble>ListLaunchProfiles",
"nimble>GetLaunchProfile",
"nimble>GetLaunchProfileMember",
"nimble>ListLaunchProfileMembers",
"nimble>PutLaunchProfileMembers",
"nimble>UpdateLaunchProfileMember",
"nimble>DeleteLaunchProfileMember"
],
"Resource": "*"
},
{
"Effect": "Allow",
"Action": [
"sso-directory:DescribeUsers",
"sso-directory:SearchUsers",
"identitystore:DescribeUser",
"identitystore>ListUsers"
],
"Resource": [
"*"
]
},
{
"Effect": "Allow",
"Action": [
"ds>CreateComputer",
"ds>DescribeDirectories",
"ec2>DescribeSubnets",
```

```
 "ec2:CreateNetworkInterface",
 "ec2:DescribeNetworkInterfaces",
 "ec2:DeleteNetworkInterface",
 "ec2:CreateNetworkInterfacePermission",
 "ec2:DeleteNetworkInterfacePermission",
 "ec2:DescribeSecurityGroups",
 "fsx:DescribeFileSystems"
],
"Resource": [
 "*"
],
"Condition": {
 "StringEquals": {
 "aws:CalledViaLast": "nimble.amazonaws.com"
 }
}
},
{
 "Effect": "Allow",
 "Action": "cloudwatch:GetMetricData",
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "cloudwatch:namespace": "AWS/NimbleStudio"
 }
 }
},
],
"Version": "2012-10-17"
}
```

## AWS managed policy: AmazonNimbleStudio-StudioUser

You can attach the [AmazonNimbleStudio-StudioUser](#) policy to your IAM identities.

Attach this policy to the User role associated with your studio to grant access to Amazon Nimble Studio resources associated with the studio user and related studio resources in other services.

### Permissions details

This policy includes the following permissions.

- nimble - Allows Studio Users access to Nimble resources that have been delegated to them by StudioAdmins.
- sso - Allows Studio Users the ability to view the names of other users in the studio.
- identitystore - Allows Studio Users the ability to view the names of other users in the studio.
- ds - Allows Nimble Studio to add virtual workstations to the AWS Managed Microsoft AD associated with the studio.
- ec2 - Allows Nimble Studio to attach virtual workstations to your configured VPC.
- fsx - Allows Nimble Studio to connect virtual workstations to your configured Amazon FSx volumes.

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ds>CreateComputer",
 "ec2:DescribeSubnets",
 "ec2>CreateNetworkInterfacePermission",
 "ec2:DescribeNetworkInterfaces",
 "ec2>DeleteNetworkInterfacePermission",
 "ec2>DeleteNetworkInterface",
 "ec2>CreateNetworkInterface",
 "ec2:DescribeSecurityGroups",
 "fsx:DescribeFileSystems",
 "ds:DescribeDirectories"
],
 "Resource": [
 "*"
],
 "Condition": {
 "StringEquals": {
 "aws:CalledViaLast": "nimble.amazonaws.com"
 }
 }
 },
 {
 "Effect": "Allow",
 "Action": [
 "sso-directory:DescribeUsers",
 "sso-directory:SearchUsers",
]
 }
]
}
```

```
 "identitystore:DescribeUser",
 "identitystore>ListUsers"
],
"Resource": [
 "*"
]
},
{
"Effect": "Allow",
"Action": [
 "nimble>ListLaunchProfiles"
],
"Resource": "*",
"Condition": {
 "StringEquals": {
 "nimble:requesterPrincipalId": "${nimble:principalId}"
 }
}
},
{
"Effect": "Allow",
"Action": [
 "nimble>ListStudioMembers",
 "nimble>GetStudioMember",
 "nimble>ListEulas",
 "nimble>ListEulaAcceptances",
 "nimble>GetFeatureMap",
 "nimble>PutStudioLogEvents"
],
"Resource": "*"
},
{
"Effect": "Allow",
"Action": [
 "nimble>StartStreamingSession",
 "nimble>StopStreamingSession",
 "nimble>DeleteStreamingSession",
 "nimble>GetStreamingSession",
 "nimble>CreateStreamingSessionStream",
 "nimble>GetStreamingSessionStream",
 "nimble>ListStreamingSessions",
 "nimble>ListStreamingSessionBackups",
 "nimble>GetStreamingSessionBackup"
],

```

```
"Resource": "*",
"Condition": {
 "StringEquals": {
 "nimble:ownedBy": "${nimble:requesterPrincipalId}"
 }
},
],
"Version": "2012-10-17"
}
```

## Nimble Studio updates to AWS managed policies

View details about updates to AWS managed policies for Amazon Nimble Studio since this service began tracking these changes.

| Change                                                                                     | Description                                                                                         | Date              |
|--------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-------------------|
| <a href="#"><u>AWS managed policy: AmazonNimbleStudio-StudioUser - Updated policy</u></a>  | Amazon Nimble Studio updated a policy to allow studio users to view their workstation backups.      | December 20, 2022 |
| <a href="#"><u>AWS managed policy: AmazonNimbleStudio-StudioAdmin - Updated policy</u></a> | Amazon Nimble Studio updated the policy to allow studio admins to view their workstation backups.   | December 20, 2022 |
| <a href="#"><u>AWS managed policy: AmazonNimbleStudio-StudioUser - Updated policy</u></a>  | Amazon Nimble Studio updated a policy to allow studio admins to retrieve CloudWatch metrics.        | November 11, 2021 |
| <a href="#"><u>AWS managed policy: AmazonNimbleStudio-StudioUser - Updated policy</u></a>  | Amazon Nimble Studio updated the policy to allow studio users to start and stop their workstations. | November 1, 2021  |
| <a href="#"><u>AWS managed policy: AmazonNimbleStudio-</u></a>                             | Amazon Nimble Studio updated the policy to allow                                                    | November 1, 2021  |

| Change                                                                                         | Description                                                                                                                                                                          | Date            |
|------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#"><u>StudioAdmin</u></a> - Updated policy                                            | studio admins to start and stop their workstations.                                                                                                                                  |                 |
| <a href="#"><u>AWS managed policy: AmazonNimbleStudio-StudioUser</u></a> - Updated policy      | Amazon Nimble Studio updated the policy to conditionally allow access to streaming-session resources based on <code>nimble:ownedBy</code> instead of <code>nimble:createdBy</code> . | August 16, 2021 |
| <a href="#"><u>AWS managed policy: AmazonNimbleStudio-StudioUser</u></a> – New policy          | Amazon Nimble Studio added a new policy that allows access to resources associated with the studio user and related studio resources in other services.                              | April 28, 2021  |
| <a href="#"><u>AWS managed policy: AmazonNimbleStudio-StudioAdmin</u></a> - New policy         | Amazon Nimble Studio added a new policy that allows access to resources associated with the studio administrator and related studio resources in other services.                     | April 28, 2021  |
| <a href="#"><u>AWS managed policy: AmazonNimbleStudio-LaunchProfileWorker</u></a> – New policy | Amazon Nimble Studio added a new policy that allows access to resources needed by Nimble Studio launch profile workers.                                                              | April 28, 2021  |
| Amazon Nimble Studio started tracking changes                                                  | Amazon Nimble Studio started tracking changes for its AWS managed policies.                                                                                                          | April 28, 2021  |

## Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the calling service) calls another service (the called service). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it shouldn't otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that Identity and Access Management (IAM) gives Amazon Nimble Studio to access your resources. If you use both global condition context keys, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must use the same account id when used in the same policy statement.

The value of `aws:SourceArn` must be the studio's ARN and `aws:SourceAccount` must be your account id. You won't know what the studio id is until the studio is created because it's generated by Nimble Studio. Once your studio is created, you can update the trust policy with the final studio id set as the `aws:SourceArn`.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. If you don't know the full ARN of the resource or if you're specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcards (\*) for the unknown portions of the ARN. For example, `arn:aws:nimble::123456789012:*`.

Your end users assume your studio role when they sign in to the Nimble Studio portal. When you create your studio, AWS configures the role and evaluates the policy. AWS evaluates the policy every subsequent time one of your users logs in to the Nimble Studio portal. When you create a studio, you can't modify the `aws:SourceArn`. After you finish creating your studio, you can use your `studioArn` for the `aws:SourceArn`.

The following example is an assume role policy that shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in Nimble Studio to prevent the confused deputy problem.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "identity.nimble.amazonaws.com"
 },
 "Action": [
 "sts:AssumeRole",
 "sts:TagSession"
],
 "Condition": {
 "StringEquals": {
 "aws:SourceAccount": "123456789012"
 },
 "StringLike": {
 "aws:SourceArn": "arn:aws:nimble:us-west-2:123456789012:studio/*"
 }
 }
 }
]
```

## Troubleshooting Amazon Nimble Studio identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Nimble Studio and IAM.

### Topics

- [I'm not authorized to perform an action in Nimble Studio.](#)
- [I'm not authorized to perform iam:PassRole.](#)
- [I want to view my access keys.](#)
- [I'm an administrator and want to allow others to access Nimble Studio.](#)
- [I want to allow people outside of my AWS account to access my Nimble Studio resources.](#)

### I'm not authorized to perform an action in Nimble Studio.

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the mateojackson IAM user tries to use the console to view details about a fictional *my-example-widget* resource but doesn't have the fictional nimble:*GetWidget* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
nimble:GetWidget on resource: my-example-widget
```

In this case, the policy for the mateojackson user must be updated to allow access to the *my-example-widget* resource by using the nimble:*GetWidget* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

## I'm not authorized to perform iam:PassRole.

If you receive an error that you aren't authorized to perform the iam:PassRole action, then contact your administrator for assistance. Ask them to update your policies to allow you to pass a role to Nimble Studio.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you need permissions to pass the role to the service.

The following example error occurs when an user named johndoe tries to use the console to perform an action in Nimble Studio. However, the action requires the service to have permissions granted by a service role. John doesn't have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/johndoe is not authorized to perform: iam:PassRole
```

In this case, John asks his administrator to update his policies to grant permission to perform the iam:PassRole action.

## I want to view my access keys.

Amazon Nimble Studio doesn't provide access keys. To learn about secret access keys, see Managing access keys in the [IAM User Guide](#).

### Important

Don't provide your access keys to a third party, even to help [find your canonical user ID](#). By doing this, you might give someone permanent access to your account.

When you create an access key pair, you're prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, add new access keys to your user. You can have a maximum of two access keys. If you already have two, delete one key pair before creating a new one. To view instructions, see [Managing access keys](#) in the IAM User Guide.

## I'm an administrator and want to allow others to access Nimble Studio.

To allow others to access Nimble Studio, create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. Then, attach a policy to the entity that grants them the correct permissions.

Nimble Studio provides you with three different policies in the AWS Management Console. The IT administrator who manages the Console uses these policies to grant studio access to others. These are:

- AmazonNimbleStudio-StudioAdmin
- AmazonNimbleStudio-StudioUser
- AmazonNimbleStudio-LaunchProfileWorker

For a tutorial about using the administrator policy, view the [Setting up to use Nimble Studio](#) guide. To learn how to attach existing policies to users, like user and launch profile policies, see [Creating IAM users \(console\)](#).

For information about importing policies, see Creating your first IAM delegated user and group in the [IAM User Guide](#).

## I want to allow people outside of my AWS account to access my Nimble Studio resources.

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Nimble Studio supports these features, see [How Amazon Nimble Studio works with IAM](#).

- To learn how to provide access to your resources across AWS accounts that you own, see [Providing access to an IAM user in another AWS account that you own](#) in the IAM User Guide.
- To learn how to provide access to your resources to third-party AWS accounts, see [Providing access to AWS accounts owned by third parties](#) in the IAM User Guide.
- To learn how to provide access through identity federation, see [Providing access to externally authenticated users \(identity federation\)](#) in the IAM User Guide.
- To learn the difference between using roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the IAM User Guide.

## Security event logging and monitoring with Nimble Studio

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon Nimble Studio and your AWS solutions. Collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs.

AWS and Nimble Studio provide tools for monitoring your resources and responding to potential incidents, including [Logging Nimble Studio calls using AWS CloudTrail](#) and [AWS CloudFormation User Guide](#).

For more information about how Amazon Nimble Studio works with AWS CloudFormation, including examples of JSON and YAML templates, see the [Amazon Nimble Studio resource and property reference](#) in the AWS CloudFormation User Guide. To understand how to use CloudFormation templates, see [AWS CloudFormation concepts](#).

### Topics

- [Logging Nimble Studio calls using AWS CloudTrail](#)

## Logging Nimble Studio calls using AWS CloudTrail

Amazon Nimble Studio is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Nimble Studio. CloudTrail captures all API calls for Nimble Studio as events. The calls captured include calls from the Nimble Studio console and code calls to the Amazon Nimble Studio operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Nimble Studio. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by

CloudTrail, you can determine the request that was made to Nimble Studio, the IP address from which the request was made, who made the request, when it was made, and additional details.

## Nimble Studio information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Nimble Studio, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#).

For an ongoing record of events in your AWS account, including events for Nimble Studio, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs.

For more information, see the following:

[Overview for creating a trail](#)

[CloudTrail supported services and integrations](#)

[Configuring Amazon SNS notifications for CloudTrail](#)

[Receiving CloudTrail log files from multiple Regions](#)

[Receiving CloudTrail log files from multiple accounts](#)

Nimble Studio actions are logged by CloudTrail and are documented in the [Amazon Nimble Studio API Reference](#). For example, calls to the CreateStudio, GetStudio and DeleteStudio actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another service.

For more information, see the [CloudTrail user Identity element](#).

## Understanding Nimble Studio log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

This JSON example shows three actions:

- ACTION\_1: CreateStudio
- ACTION\_2: GetStudio
- ACTION\_3: DeleteStudio

```
{
 "eventVersion": "0",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "EXAMPLE-PrincipalID:EXAMPLE-Session",
 "arn": "arn:aws:sts::111122223333:assumed-role/EXAMPLE-UserName/EXAMPLE-
Session",
 "accountId": "111122223333",
 "accessKeyId": "EXAMPLE-accessKeyId",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "EXAMPLE-PrincipalID",
 "arn": "arn:aws:iam::111122223333:role/EXAMPLE-UserName",
 "accountId": "111122223333",
 "userName": "EXAMPLE-UserName"
 },
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2021-03-08T23:25:49Z"
 }
 }
 },
 "eventTime": "2021-03-08T23:25:49Z",
}
```

```
"eventSource": "nimble.amazonaws.com",
"eventName": "CreateStudio",
"awsRegion": "us-west-2",
"sourceIPAddress": "192.0.2.0",
"userAgent": "EXAMPLE-userAgent",
"requestParameters": {
 "displayName": "Studio Name",
 "studioName": "EXAMPLE-studioName",
 "userRoleArn": "arn:aws:iam::111122223333:role/EXAMPLE-ServiceRole-User",
 "adminRoleArn": "arn:aws:iam::111122223333:role/EXAMPLE-ServiceRole-Admin"
},
"responseElements": {},
"requestID": "EXAMPLE-requestID",
"eventID": "EXAMPLE-eventID",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "111122223333"
},
{
"eventVersion": "0",
"userIdentity": {
 "type": "AssumedRole",
 "principalId": "EXAMPLE-PrincipalID:EXAMPLE-Session",
 "arn": "arn:aws:sts::111122223333:assumed-role/EXAMPLE-UserName/EXAMPLE-
Session",
 "accountId": "111122223333",
 "accessKeyId": "EXAMPLE-accessKeyId",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "EXAMPLE-PrincipalID",
 "arn": "arn:aws:iam::111122223333:role/EXAMPLE-UserName",
 "accountId": "111122223333",
 "userName": "EXAMPLE-UserName"
 },
 "webIdFederationData": {}
 },
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2021-03-08T23:44:25Z"
 }
}
},
```

```
"eventTime": "2021-03-08T23:44:25Z",
"eventSource": "nimble.amazonaws.com",
"eventName": "GetStudio",
"awsRegion": "us-west-2",
"sourceIPAddress": "192.0.2.0",
"userAgent": "EXAMPLE-userAgent",
"requestParameters": {
 "studioId": "us-west-2-EXAMPLE-studioId"
},
"responseElements": null,
"requestID": "EXAMPLE-requestID",
"eventID": "EXAMPLE-eventID",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "111122223333"
},
{
"eventVersion": "0",
"userIdentity": {
 "type": "AssumedRole",
 "principalId": "EXAMPLE-PrincipalID:EXAMPLE-Session",
 "arn": "arn:aws:sts::111122223333:assumed-role/EXAMPLE-UserName/EXAMPLE-
Session",
 "accountId": "111122223333",
 "accessKeyId": "EXAMPLE-accessKeyId",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "EXAMPLE-PrincipalID",
 "arn": "arn:aws:iam::111122223333:role/EXAMPLE-UserName",
 "accountId": "111122223333",
 "userName": "EXAMPLE-UserName"
 },
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2021-03-08T23:45:14Z"
 }
 }
},
"eventTime": "2021-03-08T23:44:14Z",
"eventSource": "nimble.amazonaws.com",
```

```
"eventName": "DeleteStudio",
"awsRegion": "us-west-2",
"sourceIPAddress": "192.0.2.0",
"userAgent": "EXAMPLE-userAgent",
"requestParameters": {
 "studioId": "us-west-2-EXAMPLE-studioId"
},
"responseElements": {
 "studio": {
 "adminRoleArn": "arn:aws:iam::111122223333:role/EXAMPLE-ServiceRole-Admin",
 "displayName": "My New Studio Name",
 "homeRegion": "us-west-2",
 "ssoClientId": "EXAMPLE-ssoClientId",
 "state": "DELETING",
 "statusCode": "DELETING_STUDIO",
 "statusMessage": "Deleting studio",
 "studioEncryptionConfiguration": {
 "keyType": "AWS_OWNED_CMK"
 },
 "studioId": "us-west-2-EXAMPLE-studioId",
 "studioName": "EXAMPLE-studioName",
 "studioUrl": "https://sso111122223333.us-
west-2.portal.nimble.amazonaws.com",
 "tags": {},
 "userRoleArn": "arn:aws:iam::111122223333:role/EXAMPLE-ServiceRole-User"
 }
},
"requestID": "EXAMPLE-requestID",
"eventID": "EXAMPLE-eventID",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "111122223333"
}
```

In the example, you'll notice that the events show the Region, IP address, and other "requestParameters" such as the "userRoleArn" and "adminRoleArn" that will help you identify the event. You can see the time and date in the "creationDate", and the source where the request originated, which is marked as "eventSource": "nimble.amazonaws.com".

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in IAM or AWS STS, that activity is recorded in a CloudTrail event along with other AWS service events in Event history. You can view, search, and download recent events in your AWS account.

AWS CloudTrail captures all API calls for IAM and AWS Security Token Service (AWS STS) as events, including calls from the console and API calls. To learn more about using CloudTrail with IAM and AWS STS, see [Logging IAM and AWS STS API calls with AWS CloudTrail](#).

For more information about CloudTrail, see [AWS CloudTrail User Guide](#).

For information about other monitoring services that Amazon offers, see the [Amazon CloudWatch User Guide](#).

## Compliance validation for Amazon Nimble Studio

Amazon Nimble Studio follows the [shared responsibility model](#), and compliance is shared between AWS and our customers.

To learn whether an AWS service is within the scope of specific compliance programs, see [AWS services in Scope by Compliance Program](#) and choose the compliance program that you are interested in. For general information, see [AWS Compliance Programs](#).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading Reports in AWS Artifact](#).

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- [Security and Compliance Quick Start Guides](#) – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

 **Note**

Not all AWS services are HIPAA eligible. For more information, see the [HIPAA Eligible Services Reference](#).

- [AWS Compliance Resources](#) – This collection of workbooks and guides might apply to your industry and location.
- [AWS Customer Compliance Guides](#) – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- [Evaluating Resources with Rules](#) in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- [AWS Security Hub](#) – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see [Security Hub controls reference](#).
- [AWS Audit Manager](#) – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

## Resilience in Amazon Nimble Studio

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

In Nimble Studio, Amazon Elastic Compute Cloud (Amazon EC2) instances that are related to the AWS Thinkbox Deadline render farm are deployed in a single Availability Zone. This includes Linux or Windows render workers and the Deadline Remote Connection Server (RCS).

 **Note**

If you experience an outage, your render farm will stop working.

StudioBuilder provides a second Availability Zone for resources such as Amazon DocumentDB clusters, and file systems from Amazon Elastic File System (Amazon EFS), and Amazon FSx for Windows File Server .

For more information about AWS Regions and Availability Zones, see [AWS global infrastructure](#).

## Infrastructure security in Amazon Nimble Studio

As a managed service, Amazon Nimble Studio is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see [AWS Cloud Security](#). To design your AWS environment using the best practices for infrastructure security, see [Infrastructure Protection in Security Pillar AWS Well-Architected Framework](#).

You use AWS published API calls to access Nimble Studio through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the [AWS Security Token Service](#) (AWS STS) to generate temporary security credentials to sign requests.

## Network connectivity security model

The connection between your users and their Nimble Studio streaming session is routed over the public internet. All other network traffic from streaming sessions is routed over the network interface created by Nimble Studio in your account. This network traffic is subject to the security groups configured with the launch profile selected by the user.

While Nimble Studio creates a network interface in the customers account, the customers control what this interface has access to via launch profiles. Launch profiles allow you to control which network resources a render worker can access, including public internet access. The security configurations in launch profiles work in tandem with standard OS-level security controls to limit reachable network resources from a streaming session.

Customers can also deploy Amazon Virtual Private Cloud (VPC) endpoints to leverage [AWS PrivateLink](#). PrivateLink is a way of communicating to AWS services via private endpoints.

## Security best practices for Nimble Studio

Amazon Nimble Studio provides a number of security features to consider as you develop and implement your own security policies. The following best practices are general guidelines and don't represent a complete security solution. Because these best practices might not be appropriate or sufficient for your environment, treat them as helpful considerations rather than prescriptions.

### Monitoring

Monitoring is an important part of maintaining the reliability, availability, and performance of Nimble Studio and your AWS solutions. For more information about monitoring and responding to events, see [Security event logging and monitoring with Nimble Studio](#).

### Data protection

For data protection purposes, we recommend that you protect AWS account credentials and set up individual accounts with AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields such as a **Name** field. This includes when you

work with Amazon Nimble Studio or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into Amazon Nimble Studio or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

## Permissions

Manage access to AWS resources using users, IAM roles, and by granting the least privilege to users. Establish credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials. For more information, see [IAM Best Practices](#) in the IAM User Guide.

## Internet traffic privacy

Amazon Nimble Studio uses Amazon Virtual Private Cloud (Amazon VPC) to secure connections. Amazon VPC provides features that you can use to increase and monitor the security for your virtual private cloud (VPC).

The connection between your users and their Nimble Studio streaming session is routed over the public internet. All other network traffic from streaming sessions is routed over the network interface that Nimble Studio creates in your account. This network traffic is subject to the security groups that are configured with the launch profile that's selected by the user.

While Nimble Studio creates a network interface in the customers account, the customers control what this interface has access to via launch profiles. Launch profiles allow you to control which network resources a render worker can access, including public internet access. The security configurations in launch profiles work in tandem with standard OS-level security controls to limit reachable network resources from a streaming session.

Customers can also deploy Amazon VPC endpoints to leverage [AWS PrivateLink](#). AWS PrivateLink is a way of communicating to AWS services via private endpoints.

## Traffic between AWS resources in the same AWS Region

A VPC endpoint is a logical entity within a VPC that allows connectivity only to Nimble Studio. The VPC routes requests to Nimble Studio and routes responses back to the VPC. For more information, see [VPC Endpoints](#) in the Amazon VPC User Guide

# Network ACLs for Nimble Studio

This guide lists the default network access control list (ACL) rule numbers created by StudioBuilder for Amazon Nimble Studio.

## Contents

- [Prerequisites](#)
- [What is a network ACL?](#)
- [Network ACL rules](#)
- [Default network ACL](#)

## Prerequisites

- To complete this tutorial, you need an active Nimble Studio cloud studio deployed in your AWS account. If you don't have a cloud studio already deployed, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- This guide contains default network ACL rule numbers created by StudioBuilder. To learn about ACL basics, see [Network ACL basics](#) in the Amazon VPC User Guide.

## What is a network ACL?

A network access control list (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups so that you can add an additional layer of security to your VPC. For more information about the differences between security groups and network ACLs, see [Compare security groups and network ACLs](#).

## Network ACL rules

You can add or remove rules from the default network ACL, or create additional network ACLs for your VPC. When you add or remove rules from a network ACL, the changes are automatically applied to the subnets that it's associated with.

The following are the parts of a network ACL rule:

- Rule number. Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that might contradict it.

- Type. The type of traffic; for example, SSH. You can also specify all traffic or a custom range.
- Protocol. You can specify any protocol that has a standard protocol number. For more information, see [Protocol Numbers](#). If you specify ICMP as the protocol, you can specify any or all of the ICMP types and codes.
- Port range. The listening port or port range for the traffic. For example, 80 for HTTP traffic.
- Source. [Inbound rules only] The source of the traffic (CIDR range).

Destination. [Outbound rules only] The destination for the traffic (CIDR range).

- Allow/Deny. Whether to allow or deny the specified traffic.

## Default network ACL

The default network ACL is configured to allow all traffic to flow in and out of the subnets with which it's associated. Each network ACL also includes a rule whose rule number is an asterisk. This rule verifies that if a packet doesn't match any of the other numbered rules, it's denied. You can't modify or remove this rule.

 **Note**

If you've modified your default network ACL's inbound rules, we don't automatically add an allow rule for inbound IPv6 traffic when you associate an IPv6 block with your VPC.

Similarly, if you've modified the outbound rules, we don't automatically add an allow rule for outbound IPv6 traffic.

The following tables show the Network ACL rule numbers that StudioBuilder creates. If you have added new rules and they conflict with these rules, you will encounter issues if you ever try to upgrade your studio using a newer StudioBuilder.

### ActiveDirectory

| Inbound rules | Outbound rules |
|---------------|----------------|
| 100           | 100            |
| 110           | 110            |

|       |       |
|-------|-------|
| 120   | 120   |
| 20000 | 300   |
| 20010 | 310   |
|       | 320   |
|       | 20000 |
|       | 20010 |

## Backend

| Inbound rules | Outbound rules |
|---------------|----------------|
| 10            | 100            |
| 100           | 110            |
| 110           | 200            |
| 200           | 201            |
| 210           | 202            |
| 20000         | 210            |
| 20010         | 211            |
|               | 212            |
|               | 1000           |
|               | 1010           |
|               | 20000          |
|               | 20010          |

## Filesystems

| Inbound rules         | Outbound rules |
|-----------------------|----------------|
| 100                   | 100            |
| 101                   | 110            |
| 102                   | 200            |
| 103                   | 201            |
| 200                   | 300            |
| 201                   | 301            |
| 202                   | 20000          |
| 203                   | 20010          |
| 204 (new to SB 1.1.3) |                |
| 300                   |                |
| 301                   |                |
| 320                   |                |
| 321                   |                |
| 20000                 |                |
| 20010                 |                |

## Public

| Inbound rules | Outbound rules |
|---------------|----------------|
| 10            | 10             |

|       |       |
|-------|-------|
| 20    | 20    |
| 30    | 20000 |
| 40    | 20010 |
| 20000 |       |
| 20010 |       |

## RenderWorkers

| Inbound rules | Outbound rules |
|---------------|----------------|
| 10            | 10000          |
| 20            | 10100          |
| 400           | 20000          |
| 401           | 20010          |
| 500           |                |
| 510           |                |
| 20000         |                |
| 20010         |                |

## ServiceEndpoints

| Inbound rules | Outbound rules |
|---------------|----------------|
| 100           | 400            |
| 200           | 410            |

|     |     |
|-----|-----|
| 210 | 420 |
| 300 | 430 |
| 400 | 440 |
| 500 | 450 |
| 510 |     |
| 520 |     |
| 530 |     |
| 540 |     |
| 550 |     |

## WorkerSupport

| Inbound rules | Outbound rules |
|---------------|----------------|
| 10            | 10             |
| 20000         | 100            |
| 20010         | 200            |
|               | 20000          |
|               | 20010          |

## Workstations

| Inbound rules | Outbound rules |
|---------------|----------------|
| 10            | 10000          |

|       |       |
|-------|-------|
| 20    | 10100 |
| 400   | 20000 |
| 401   | 20010 |
| 20000 |       |
| 20010 |       |

To learn more about ACLs, see [Network ACLs](#) in the Amazon VPC User Guide.

## Configuration and vulnerability analysis in Nimble Studio

After StudioBuilder is deployed, you will be adding resources whenever you need them. Any infrastructure that customers deploy is outside of our infrastructure security controls, and therefore is the customer's sole responsibility.

Here are some examples of shared responsibilities:

- AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD)
  - AWS maintains AWS Managed Microsoft AD patches.
  - Customers manage AWS Managed Microsoft AD users.
  - Customers must maintain passwords and password rotation policies.
- Amazon FSx
  - AWS manages security for initialization scripts.
  - Customers own the shared responsibility with AWS for Amazon FSx file systems. For more details on security in Amazon FSx, see [Security in Amazon FSx for Windows](#) and [Security in FSx for Lustre](#).

Customers must take full responsibility for the security of:

- Deadline - Customers manage updates and patching against security vulnerabilities.
- Amazon Machine Image (worker AMIs)
  - Customers are responsible for patching against security vulnerabilities.

- Customers are responsible for defining the AMI. If customers fail to patch, their data can be compromised.
- Third-party software, plugins, scripts, operating systems (OS), and other applications
  - Customers manage security, updates, and patching.

 **Important**

AWS and Amazon Nimble Studio aren't responsible for render failures or Deadline infrastructure failures.

## Patching software

The security of your cloud studio and render farm relies on your commitment to a robust patching strategy. The operating system and applications running on your render farm should be regularly patched and monitored for vulnerabilities. Refer to the following documentation based on the operating system of your instances.

- Linux - [Managing software on your Amazon Linux instances](#)
- Windows - [Best practices for Windows on Amazon EC2](#)

## Other considerations

**Managed policies** – If customers modify the managed policies, they will potentially break the ability to sign in, create resources, and consume. Nimble Studio provides a managed authorization model with IAM Identity Center, so that customers don't have to interface with IAM if they don't want to.

**Launch profiles** – The IT Admin, when creating/updating a LaunchProfile from the console, has the option to assign different security groups for their projects. This allows for restricted access to storage resources, permitting only X-ENIs that are created in the context of a specific project to access them.

**Configuration management** – AWS maintains the configuration of its infrastructure devices, but a customer is responsible for configuring their own guest operating systems, databases, firewall, and applications.

**Awareness & training** – Customers must train their own employees.

**Customer specific controls** – These are solely the responsibility of the customer based on the application they're deploying within AWS services.

**Service and communications protection** – (Also known as Zone Security) might require a customer to route or zone data within specific security environments.

# Monitoring Nimble Studio

Monitoring Nimble Studio is important for maintaining the reliability, availability, and performance of your cloud studio and other AWS solutions.

AWS provides the following tools to monitor Nimble Studio, report when something is wrong, and take automatic actions, when appropriate.

- *Amazon CloudWatch* monitors your AWS resources and the applications that you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you when a specified metric reaches a threshold that you specify. For example, CloudWatch can track CPU usage or other metrics of your Amazon EC2 instances, and automatically launch new instances when needed. For more information, see the [Amazon CloudWatch User Guide](#).
- With *Amazon CloudWatch Logs* you can monitor, store, and access your log files from Amazon EC2 instances, CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the [Amazon CloudWatch Logs User Guide](#).
- *AWS CloudTrail* captures API calls and related events made by, or on behalf of, your AWS account. Then, AWS CloudTrail delivers the log files to an Amazon S3 bucket that you specify. CloudTrail logs identify the users and accounts that called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the [AWS CloudTrail User Guide](#).

*Amazon EventBridge* is a serverless event bus that connects your applications with data from a variety of sources. EventBridge delivers a stream of real-time data from your own applications, software as a service (SaaS) applications, and AWS services. Then it routes that data to targets, such as Lambda, so that you can monitor events that happen in services, and build event-driven architectures. For more information, see the [Amazon EventBridge User Guide](#).

## Contents

- [Monitoring Nimble Studio with Amazon CloudWatch](#)
- [Monitoring examples](#)
- [Related resources](#)

# Monitoring Nimble Studio with Amazon CloudWatch

You can monitor Nimble Studio using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These statistics are kept for 15 months, so that you can access historical information and gain perspective on how your web application or service is performing.

You can also set alarms that watch for certain thresholds and send notifications when those thresholds are met. For more information, see the [Amazon CloudWatch User Guide](#).

The Nimble Studio service reports the following metrics in the AWS/NimbleStudio namespace.

| Metric                                   | Description                                                                                                                                                                                                                                                 | Dimensions      |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| RunningStreamingSessions                 | <p>The count of streaming sessions in the READY state for the account or studio.</p> <p>Reporting criteria: There is a nonzero value.</p> <p>Statistics: The most useful statistic is Maximum.</p> <p>Unit: Count</p>                                       | LaunchProfileId |
| StreamingSessionLaunchSuccess            | <p>The count of streaming session creation attempts that successfully launched, and that you can connect to with NICE DCV.</p> <p>Reporting criteria: There is a nonzero value.</p> <p>Statistics: The most useful statistic is Sum.</p> <p>Unit: Count</p> | LaunchProfileId |
| StreamingSessionLaunchConfigurationError | <p>The count of streaming session creation attempts that failed to launch due to a configuration error that the customer can remediate.</p> <p>Reporting criteria: There is a nonzero value.</p> <p>Statistics: The most useful statistic is Sum.</p>       | LaunchProfileId |

| Metric                                    | Description                                                                                                                                                                                                                                                                                                  | Dimensions      |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
|                                           | Unit: Count                                                                                                                                                                                                                                                                                                  |                 |
| StreamingSessionLaunchInternalServerError | <p>The count of streaming session creation attempts that failed to launch due to an internal error that the customer can't remediate themselves with configuration changes.</p> <p>Reporting criteria: There is a nonzero value.</p> <p>Statistics: The most useful statistic is Sum.</p> <p>Unit: Count</p> | LaunchProfileId |

The following dimensions are supported for the Nimble Studio metrics.

| Dimension       | Description                                                        |
|-----------------|--------------------------------------------------------------------|
| LaunchProfileId | The ID of the launch profile used to create the streaming session. |

## Monitoring examples

The following examples show how to use the AWS CLI to respond to a CloudWatch alarm, and to see which WorkSpaces in a directory have connection failures.

### To respond to a CloudWatch alarm

- Determine which directory that the alarm applies to by using the [describe-alarms](#) command.

```
aws cloudwatch describe-alarms --state-value "ALARM"{
 "MetricAlarms": [
 {
 ...
 "Dimensions": [
 {
 "Name": "DirectoryId",
```

```
 "Value": "directory_id"
 }
],
...
}
]
}
```

2. Get the list of WorkSpaces in the specified directory using the [describe-workspace](#) command.

```
aws workspaces describe-workspaces --directory-id directory_id{
 "Workspaces": [
 {
 ...
 "WorkspaceId": "workspace1_id",
 ...
 },
 {
 ...
 "WorkspaceId": "workspace2_id",
 ...
 },
 {
 ...
 "WorkspaceId": "workspace3_id",
 ...
 }
]
}
```

3. Get CloudWatch metrics for WorkSpaces in a specified directory by using the [get-metric-statistics](#) command.

```
aws cloudwatch get-metric-statistics \
--namespace AWS/NimbleStudio \
--metric-name RunningStreamingSessions \
--start-time 2015-04-27T00:00:00Z \
--end-time 2015-04-28T00:00:00Z \
--period 3600 \
--statistics Maximum \
--dimensions "Name=LaunchProfileId,Value=launch_profile_id"{
 "Datapoints" : [
 {

```

```
"Timestamp": "2015-04-27T00:18:00Z",
"Maximum": 5,
"Unit": "Count"
},
{
"Timestamp": "2014-04-27T01:18:00Z",
"Maximum": 2,
"Unit": "Count"
}
],
"Label" : "RunningStreamingSessions"
}
```

## Related resources

- [Metrics](#)
- [Using Amazon CloudWatch metrics](#)
- [Using Amazon CloudWatch alarms](#)

# Troubleshooting Nimble Studio

Use the following information to troubleshoot Amazon Nimble Studio and its resources, including issues with the render farm, home directory, and initialization scripts.

## Contents

- [Troubleshooting session unavailability](#)
- [Troubleshooting render farm issues](#)
- [Troubleshooting home directory issues](#)
- [Troubleshooting system initialization scripts](#)

## Troubleshooting session unavailability

**Problem:** One of my streaming sessions isn't accessible between certain start and end times.

To solve this problem, choose one of the following solutions:

**Solution 1:** If [Starting and stopping workstations](#) is turned on for the launch profile that you launched the streaming session with, follow the steps in [Starting and stopping workstations](#) to restart the session.

**Solution 2:** Terminate the session from the [Nimble Studio console](#) and launch the session again. You might lose data when you terminate the inaccessible session.

**Solution 3:** If you recently updated your Amazon Machine Image (AMI) in the launch profile that you launched the streaming session with, [update the launch profile](#) to use the previous AMI, if available.

After trying these solutions, if you still can't access the streaming session, contact [support](#). Include the following information in your message:

- Briefly describe the sequence of activities that you performed before the session became inaccessible.
- What session behavior did you see before the session became inaccessible?
- Does an inaccessible session always occur when you launch from this launch profile?

# Troubleshooting render farm issues

## Streaming instances can't submit new render jobs

**Problem:** My streaming workstation instances that use the default Nimble Studio workstation AMI can't submit new jobs to the render farm.

To solve this problem, choose one of the following solutions:

**Solution 1:** If you have an existing studio and you want to upgrade to use **Thinkbox Usage Based Licensing (UBL)**: First, update your existing render worker and streaming AMIs to the latest version of Deadline (10.1.19.x or higher) by following the [Update AMIs for your operating system](#) tutorial. After that, [Update to latest StudioBuilder version](#).

**Solution 2:** If your existing studio is using the default workstation AMI: [Update to latest StudioBuilder version](#) to update your existing compute farm component. If you don't, the connection from the Deadline client to the render farm won't work, and rendering on the farm will break.

**Solution 3:** When you don't want to update your studio with StudioBuilder 1.1.5 or later: To continue using the default workstation AMI, use the following instructions to manually update your compute farm component.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. Select the **Compute Farm** component.
4. Choose **Action**. Then choose **Edit**.
5. Edit the **User Initialization scripts** for Windows and Linux in the **Initialization scripts** section.
  - Replace `.*/*deadlinecommand *-*ChangeRepositorySkipValidation`  
`Proxy renderqueue.<studio_name>.nimble.<region>.aws:4433.`  
`<path>` with `./deadlinecommand.exe -SetIniFileSetting ProxyRoot`  
`renderqueue.<studio_name>.nimble.<region>.aws:4433`
6. Select **Update launch profile**.

# Troubleshooting home directory issues

## Multiple Linux home directories sharing the same mount points

**Problem:** I use Linux home directories, and when I updated my studio with StudioBuilder 1.1.5, multiple storage volumes tried to mount to the same mount point.

**Solution:** If you followed the [Setting up Linux home directories](#) tutorial, check your launch profiles for multiple Linux home directory components. If you do have multiples, use the following instructions to remove earlier Linux home directory components or to update the mount points.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. In the **File storage** section, choose the home directory component that you want to update or remove.
  - a. To update the mount point:
    - Choose **Action**. Then choose **Edit**.
  - b. To remove the component:
    - Choose **Action**. Then choose **Remove**.
4. If you chose to update the mount point: Navigate to the **File storage configuration** section and update the **Linux mount point**.
5. Read the terms and conditions and if you agree:
  - Select the check box next to **I understand that Nimble Studio will access my existing file storage**.
6. Choose **Save connection parameters**.

## Deadline render `tls_cert.crt` isn't found after updating studio

**Error:** Repository Connection Error

**Problem:** I followed the [Setting up Linux home directories](#) tutorial, updated my studio with StudioBuilder 1.1.5, and created a Linux home directory environment. Why did I get the Repository Connection Error in Deadline Monitor?

This error happens when your earlier launch profiles try to mount an additional EBS volume and Deadline can't locate the `tls_cert.crt` file in the `/home/deadline/tls_cert.crt` path. Multiple drives trying to mount to the same mount point (in this case `/home`) will conflict, and the most recent one will become the default.

**Solution:** Remove conflicting Linux home directory components from your launch profile.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Studio resources** in the left navigation pane.
3. In the **File storage** section, select the home directory component that you want to remove.
4. Choose **Action**. Then choose **Remove**.

After you've removed the old Linux home directory component, update your launch profiles with the new Linux home directory component by following [Step 3: Update your studio](#). When you get to the **Storage questions**, choose **Modify**. After that, Deadline can locate the `tls_cert.crt` file.

## Troubleshooting system initialization scripts

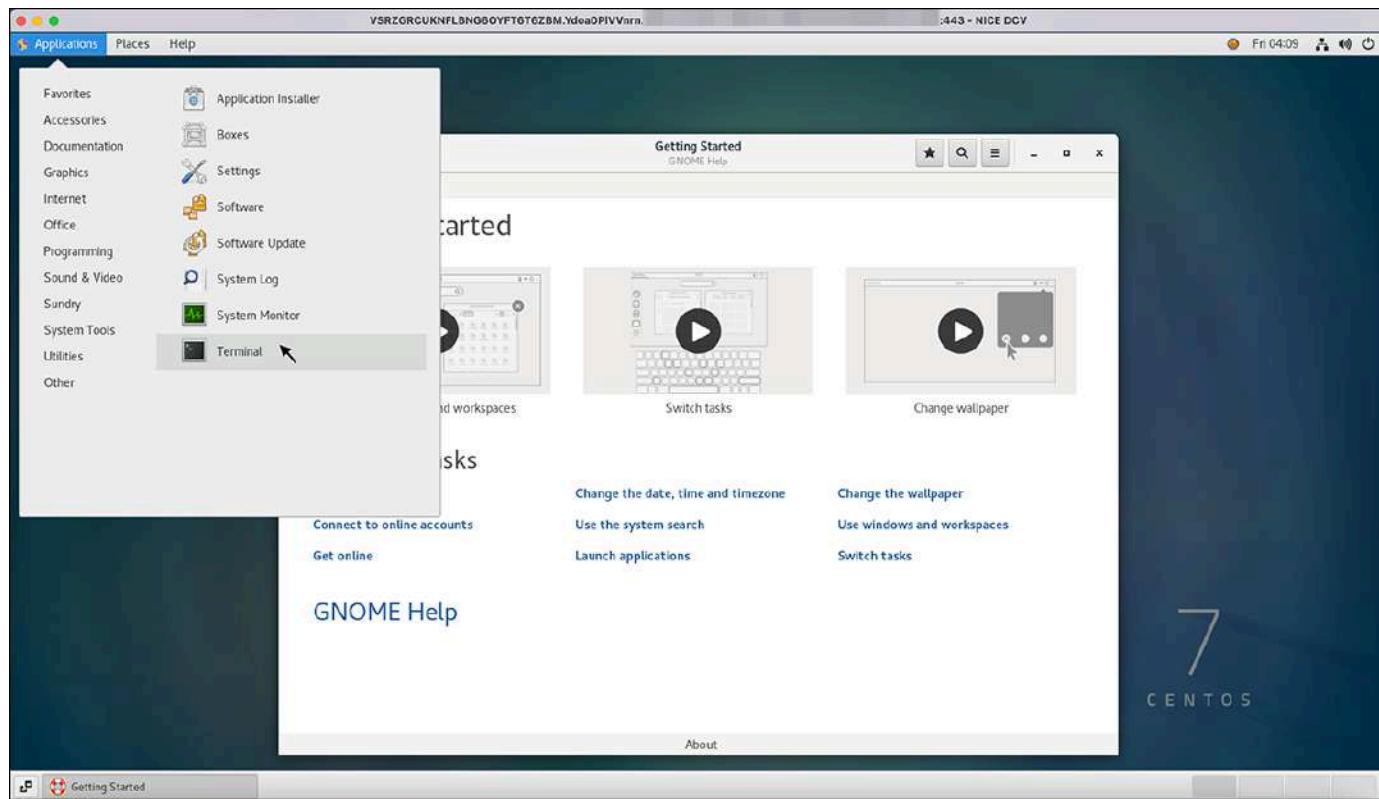
### Linux system initialization script failure

**Problem:** My Linux system initialization scripts keep failing on launch.

#### Step 1: Determine which script failed

Before you can troubleshoot this issue, determine which script failed.

1. Get superuser access to view initialization scripts on Linux by following the [Provide Superuser access for Linux users](#) tutorial.
2. Launch a Linux workstation by following the [Launching a virtual workstation](#) tutorial.
3. Select **Applications** in the menu bar. Then choose **System Tools** and **Terminal**.



4. Run the following command to become a super user: su
5. Enter your user password and press the enter (or return) key.
6. Run the following commands to go to the location of the launch log: cd /var/log/amazon/nimblestudio/ vi launch-profile-system-init.log
7. Examine the list and see which scripts ran successfully and which scripts failed.
8. Notice the failed script.

Now that you know which script failed, you can fix the issue in the following step.

## Step 2: Manually rerun failed script

If the script fails to run on launch, but successfully runs when you manually run it, the script order could be incorrect. If this is the case, it should run after a different initialization script. In this section, you will manually rerun the failed script and replace it.

1. Run the following command to navigate to the location of the initialization script: cd /etc/amazon/nimblestudio/
2. Run the following command to view the initialization scripts: ls

3. Find the system initialization script that corresponds with the failed initialization script that you noted in *step 9* of the previous section.
4. Run the following command to run the script: bash ***LAUNCH\_PROFILE\_INIT\_SCRIPT***
  - a. Replace ***LAUNCH\_PROFILE\_INIT\_SCRIPT*** with the launch profile initialization script that you found in *step 9* of [Step 1: Determine which script failed](#).
  - b. Example: bash `launch-profile-system-init-01-EXAMPLE.sh`

## Step 3: Reorder studio components

If the script runs successfully, you can reorder the components attached to your launch profile. This will make the new initialization script run last when you start up your workstation.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profile that you want to edit.
4. Choose **Action**. Then choose **Edit**.
5. Remove the studio component that you want to move further down on the list.
6. Choose **Update launch profile**.
7. Add the component that you just removed back into the last place in the list. It will be the last initialization script to run when you launch a workstation.

# Amazon Nimble Studio service quotas

Your AWS account has default quotas, formerly referred to as limits, for use with Nimble Studio and other AWS services. Unless otherwise stated, each quota is Region-specific. You can request increases for some quotas, and other quotas can't be increased.

Amazon Nimble Studio Regions:

- **US East (N. Virginia)** us-east-1
- **US West (Oregon)** us-west-2
- **Asia Pacific (Sydney)** ap-southeast-2
- **Canada (Central)** ca-central-1
- **Europe (London)** eu-west-2
- **Asia Pacific (Tokyo)** ap-northeast-1

## To view service quotas

You can view service quotas using the following options:

- Open the [Service endpoints and quotas](#) page in the AWS General Reference. Search for Amazon Nimble Studio, and select the link to go to the page for that service.
- Open the [Service Quotas console](#).
- Use the [list-service-quotas](#) and [list-aws-default-service-quotas](#) AWS CLI commands.

## To request a quota increase

You can request a quota increase using Service Quotas and AWS Support Center. If a service isn't yet available in Service Quotas, use AWS Support Center instead. Increases aren't granted immediately. It might take a couple of days for your increase to become effective.

- (Recommended) Open the [Service Quotas console](#).
- In the navigation pane, choose **AWS services**. Select a service, select a quota, and follow the directions to request a quota increase. For more information, see [Requesting a quota increase](#) in the Service Quotas User Guide.

- Use the [request-service-quota-increase](#) AWS CLI command.
- Open the [AWS Support Center](#) page, sign in if necessary, and choose **Create case**. Choose **Service limit increase**. Complete and submit the form.

For a step-by-step guide to checking Nimble Studio quotas, see [Setting up to use Nimble Studio](#).

# Getting help and support

There are several ways to get the help you need when you encounter an issue deploying or using Amazon Nimble Studio. See the following sections to learn about the different support options that are available to you.

## Contents

- [AWS Support Center](#)
- [Nimble Studio forum](#)
- [Nimble Studio help page](#)
- [AWSThinkboxDeadline Documentation](#)
- [AWS Premium Support plans](#)

## AWS Support Center

The [AWS Support Center](#) gives you access to a variety of resources. There are links to the knowledge center, knowledge center videos, AWS documentation, plus training and certification.

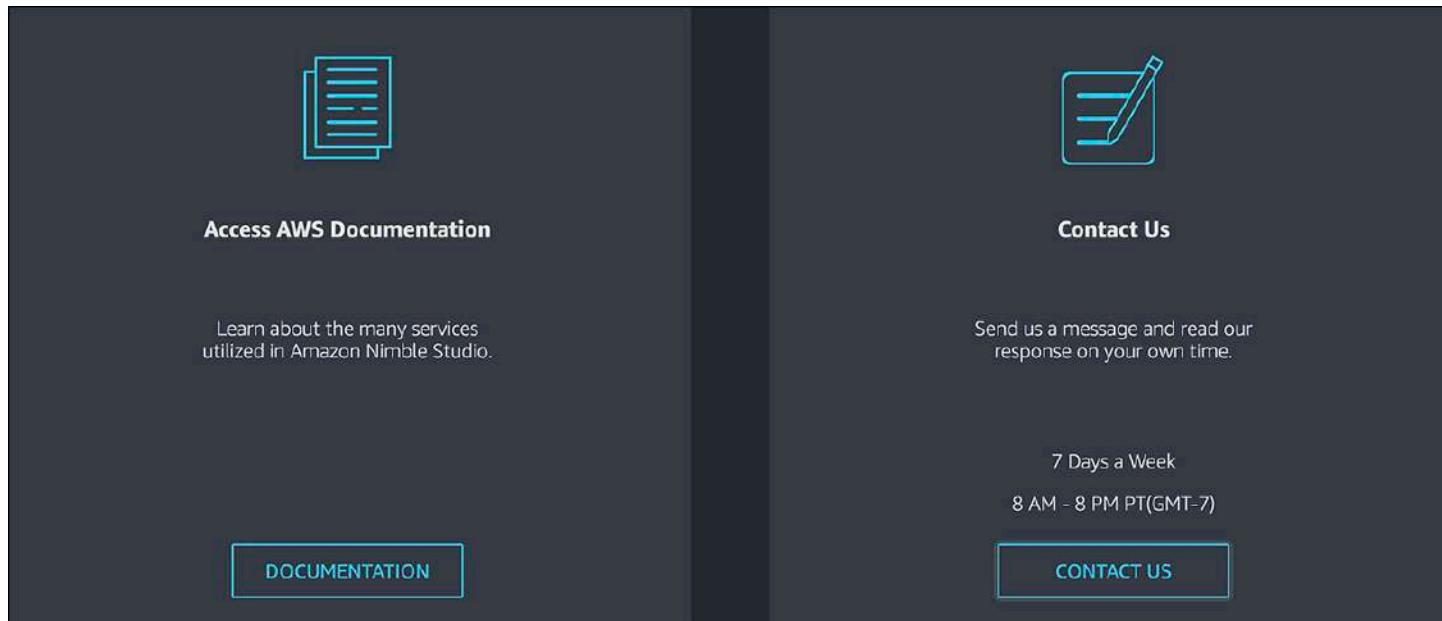
## Nimble Studio forum

If you have a specific question, and would like to get answers from the community and AWS forum moderators, visit the [Nimble Studio forum](#).

## Nimble Studio help page

If you're working in your studio's Nimble Studio portal and you need help, choose your name in the upper right-hand corner, and choose **Help** from the dropdown menu.

The Nimble Studio help page offers you two options for getting help. It links you to the official AWS Documentation for Nimble Studio, and it also links you to a form that will allow you to send a message directly to an AWS sales representative.



## AWSThinkboxDeadline Documentation

If you need help with your render farm or want to learn more about how Deadline works, check out the [AWSThinkboxDeadline Documentation](#). Make sure to look at the documentation for the version of Deadline that your studio is using.

## AWS Premium Support plans

Our Support plans are designed to give you the right tools and access to expertise so that you can be successful with leveraging AWS to help you optimize performance, manage risk, and understand costs. For more information about AWS Support plans, see [Compare AWS Support Plans](#).

For more information about how AWS can support you, visit the [Contact us](#) page.

# Release notes for Amazon Nimble Studio

This section of the table of contents contains all of Nimble Studio's release notes, showing the latest release date first.

## Release dates

- [Nimble Studio AMI release notes](#)
- [StudioBuilder v1.1.13 release notes - May 12, 2023](#)
- [StudioBuilder v1.1.12 release notes - January 25, 2023](#)
- [StudioBuilder v1.1.11 release notes - December 8, 2022](#)
- [StudioBuilder v1.1.10 release notes - October 18, 2022](#)
- [StudioBuilder v1.1.9 release notes - July 27, 2022](#)
- [StudioBuilder v1.1.8 release notes - July 8, 2022](#)
- [StudioBuilder v1.1.7 release notes - April 22, 2022](#)
- [StudioBuilder v1.1.6 release notes - March 28, 2022](#)
- [StudioBuilder v1.1.5 release notes - December 10, 2021](#)
- [StudioBuilder v1.1.4 release notes - October 6, 2021](#)
- [StudioBuilder v1.1.3 release notes - July 22, 2021](#)

## Nimble Studio AMI release notes

This page contains all of Nimble Studio's AMI release notes, showing the latest release date first.

### Contents

- [Nimble Studio Windows 2022 workstation AMI](#)
- [Nimble Studio Windows 2019 workstation AMI](#)
- [Nimble Studio Linux workstation AMI](#)
- [Nimble Studio Windows 2022 worker AMI](#)
- [Nimble Studio Windows 2019 worker AMI](#)
- [Nimble Studio Linux worker AMI](#)

# Nimble Studio Windows 2022 workstation AMI

The following table describes the release history for the Nimble Studio Windows workstation AMI.

| Release          | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| December 4, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to November 15, 2023</li><li>• <a href="#">DCV 2023.1.16220</a></li><li>• <a href="#">Firefox ESR version 115.4.0esr</a></li><li>• <a href="#">NVIDIA driver version 16.2 (537.70)</a></li><li>• <a href="#">VSCode 1.84.2</a></li></ul>                                                                               |
| July 28, 2023    | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to July 25, 2023</li><li>• <a href="#">Blender 3.5.1</a></li><li>• <a href="#">DCV 2023.0.15065</a></li><li>• <a href="#">Firefox ESR version 102.13.0 ESR</a></li><li>• <a href="#">VSCode 1.80.0</a></li><li>• <a href="#">Nvidia Driver version 15.3</a></li><li>• <a href="#">Uses IMDSv2 by default</a></li></ul> |
| April 11, 2023   | This release contains operating system updates for Windows and application                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

| Release           | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to April 11, 2023</li><li>• <a href="#">DCV 2023.0.14852</a></li><li>• <a href="#">Firefox ESR version 102.9.0</a></li><li>• <a href="#">VSCode 1.76.2</a></li></ul>                                                                                                                                                            |
| February 23, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to February 15, 2023</li><li>• <a href="#">DCV 2022.2.14357</a></li><li>• <a href="#">Firefox ESR version 102.7.0</a></li><li>• <a href="#">VSCode 1.75.0</a></li><li>• <a href="#">VLC 3.0.18</a></li><li>• <a href="#">Blender 3.4.1</a></li></ul> |
| December 2, 2022  | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to November 10, 2022</li><li>• <a href="#">Firefox ESR version 102.4.0</a></li><li>• <a href="#">VSCode VSCode 1.73.0</a></li></ul>                                                                                                                  |

| Release          | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| October 21, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"> <li>• Windows security updates current to October 12, 2022</li> <li>• <a href="#">Firefox ESR version 102.2.3</a></li> <li>• <a href="#">VSCode VSCode 1.72.1</a></li> <li>• <a href="#">Blender 3.3.1</a></li> </ul>                                                    |
| October 4, 2022  | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"> <li>• Windows security updates current to September 14, 2022.</li> <li>• <a href="#">Nvidia driver 513.46</a></li> <li>• <a href="#">Firefox ESR version 102.2.0</a></li> <li>• <a href="#">VSCode VSCode 1.71.1</a></li> <li>• <a href="#">Blender 3.3.0</a></li> </ul> |

## Nimble Studio Windows 2019 workstation AMI

The following table describes the release history for the Nimble Studio Windows workstation AMI.

| Release          | Changes                                                                                                                         |
|------------------|---------------------------------------------------------------------------------------------------------------------------------|
| December 4, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update</p> |

| Release       | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <p>workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to December 1, 2023</li><li>• <a href="#">Firefox ESR version 115.4.0esr</a></li><li>• <a href="#">NVIDIA driver version 16.2 (537.70)</a></li><li>• <a href="#">VSCode 1.84.2</a></li></ul>                                                                                                                                                                                                                                                    |
| July 26, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to July 25, 2023</li><li>• <a href="#">Blender 3.5.1</a></li><li>• <a href="#">DCV 2023.0.15065</a></li><li>• <a href="#">Firefox ESR version 102.13.0 ESR</a></li><li>• <a href="#">VSCode 1.80.0</a></li><li>• <a href="#">Nvidia Driver version 15.3</a></li><li>• <a href="#">Uses IMDSv2 by default</a></li></ul> |

| Release           | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| April 3, 2023     | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to April 3, 2023</li><li>• <a href="#">DCV 2023.0.14852</a></li><li>• <a href="#">Firefox ESR version 102.9.0</a></li><li>• <a href="#">VSCode 1.76.2</a></li></ul> |
| February 23, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to February 15, 2023</li><li>• <a href="#">DCV 2022.2.14357</a></li><li>• <a href="#">VSCode 1.75.0</a></li><li>• <a href="#">Blender 3.4.1</a></li></ul>           |

| Release          | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| January 26, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to January 26, 2023</li><li>• <a href="#">DCV 2022.2.14175</a></li><li>• <a href="#">Firefox ESR version 102.7.0</a></li><li>• <a href="#">VSCode VSCode 1.74.3</a></li><li>• <a href="#">VLC 3.0.18</a></li></ul> |
| December 2, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to November 10, 2022</li><li>• <a href="#">Firefox ESR version 102.4.0</a></li><li>• <a href="#">VSCode VSCode 1.73.0</a></li></ul>                                                                                |
| October 21, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to October 12, 2022</li><li>• <a href="#">Blender 3.3.1</a></li></ul>                                                                                                                                              |

| Release            | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| September 27, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to September 20, 2022</li><li>• <a href="#">Firefox ESR version 102.2.0 ESR</a></li><li>• <a href="#">VSCode 1.71.1</a></li><li>• <a href="#">Blender 3.3.0</a></li></ul>                                              |
| August 25, 2022    | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to August 9, 2022, 2022</li><li>• <a href="#">Firefox ESR version 91.12.0 ESR</a></li><li>• <a href="#">VSCode 1.70.0</a></li><li>• <a href="#">Blender 3.2.2</a></li><li>• <a href="#">DCV 2022.1-13300</a></li></ul> |

| Release       | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| July 21, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to July 19, 2022</li><li>• <a href="#">Nvidia driver 512.78</a></li><li>• <a href="#">Firefox ESR version 91.11.0 ESR</a></li><li>• <a href="#">VSCode VSCode 1.69.1</a></li><li>• <a href="#">Blender 3.2.1</a></li><li>• <a href="#">DCV 2022.0.12760</a></li><li>• <a href="#">Deadline 10.1.21.4</a></li></ul> |
| July 6, 2022  | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to June 21, 2022</li><li>• <a href="#">Firefox ESR version 91.10.0 ESR</a></li><li>• <a href="#">VSCode VSCode 1.68.1</a></li><li>• <a href="#">NICE DCV Server version 2022.0.12760</a></li><li>• <a href="#">Blender version 3.2.0</a></li></ul>                                                                 |

| Release      | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| May 23, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to May 17, 2022</li><li>• <a href="#">Nvidia driver 511.65</a></li><li>• <a href="#">Firefox ESR version 91.9.0 ESR</a></li><li>• <a href="#">VSCode VSCode 1.67.1</a></li><li>• <a href="#">VLC 3.0.17.4</a></li></ul> |
| May 4, 2022  | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to May 4, 2022</li><li>• <a href="#">DCV version 2022.0-12123</a></li><li>• <a href="#">Firefox ESR version 91.8.0 ESR</a></li><li>• <a href="#">Blender 3.1.2</a></li><li>• <a href="#">VSCode 1.66.2</a></li></ul>    |

| Release          | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| March 11, 2022   | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"> <li>• Windows security updates current to March 9, 2022</li> <li>• DCV version 2022.0-11954</li> <li>• Firefox ESR version 91.6.0 ESR</li> <li>• Blender 3.0.1</li> <li>• VSCode 1.64.2</li> </ul> |
| January 27, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"> <li>• Windows security updates current to January 27, 2022</li> <li>• Firefox ESR version 91.5.0 ESR</li> <li>• DCV Server version 2021.3</li> </ul>                                               |

## Nimble Studio Linux workstation AMI

The following table describes the release history for the Nimble Studio Linux workstation AMI.

| Release          | Changes                                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| December 4, 2023 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update worksta</p> |

| Release       | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <p>on AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 security updates current to Novemeber 15, 2023</li><li>• <a href="#">DCV 2023.1.16220</a></li><li>• <a href="#">Firefox ESR version 115.4.0esr</a></li><li>• <a href="#">NVIDIA driver version 16.2 (537.70)</a></li><li>• <a href="#">VSCode 1.84.2</a></li></ul>                                                                                                                                                            |
| July 13, 2023 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to July 6, 2023</li><li>• <a href="#">DCV 2023.0.15065</a></li><li>• <a href="#">Firefox ESR version 102.12.0 ESR</a></li><li>• <a href="#">VSCode 1.80.0</a></li><li>• <a href="#">Nvidia Driver version 15.2</a></li><li>• <a href="#">Uses IMDSv2 by default</a></li></ul> |
| April 3, 2023 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to April 3, 2023</li><li>• <a href="#">DCV 2023.0.14852</a></li><li>• <a href="#">Firefox ESR version 102.9.0</a></li><li>• <a href="#">VSCode 1.76.2</a></li></ul>                                                                                                           |

| Release           | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| February 23, 2023 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to February 15, 2023</li><li>• <a href="#">DCV 2022.2.14357</a></li><li>• <a href="#">VSCode 1.75.0</a></li><li>• <a href="#">Blender 3.4.1</a></li></ul>                                                          |
| January 26, 2023  | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to January 26, 2023</li><li>• <a href="#">DCV 2022.2.14175</a></li><li>• <a href="#">Firefox ESR version 102.7.0</a></li><li>• <a href="#">VSCode VSCode 1.74.3</a></li><li>• <a href="#">VLC 3.0.18</a></li></ul> |

| Release            | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| December 2, 2022   | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to November 10, 2022</li><li>• <a href="#">Firefox ESR version 102.4.0</a></li><li>• <a href="#">VSCode VSCode 1.73.0</a></li></ul>                                        |
| October 21, 2022   | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to October 12, 2022</li><li>• <a href="#">Firefox ESR version 102.2.3</a></li><li>• <a href="#">VSCode VSCode 1.72.1</a></li><li>• <a href="#">Blender 3.3.1</a></li></ul> |
| September 27, 2022 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to September 20, 2022</li><li>• <a href="#">Firefox ESR version 102.2.0 ESR</a></li><li>• <a href="#">VSCode 1.71.1</a></li><li>• <a href="#">Blender 3.3.0</a></li></ul>  |

| Release         | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| August 25, 2022 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to August 9, 2022</li><li>• <a href="#">Firefox ESR version 91.12.0 ESR</a></li><li>• <a href="#">VSCode 1.70.0</a></li><li>• <a href="#">Blender 3.2.2</a></li><li>• <a href="#">DCV 2022.1-13300</a></li></ul> |
| August 4, 2022  | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to August 1, 2022</li><li>• Bug fix for fullscreen mode when using g4dn instance types for streaming workstations</li></ul>                                                                                      |

| Release       | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| July 21, 2022 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to July 19, 2022</li><li>• <a href="#">Nvidia driver 510.73.08</a></li><li>• <a href="#">VSCode VSCode 1.69.1</a></li><li>• <a href="#">Blender 3.2.1</a></li><li>• <a href="#">DCV 2022.0.12760</a></li><li>• <a href="#">Deadline 10.1.21.4</a></li></ul> |
| July 6, 2022  | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to June 21, 2022</li><li>• <a href="#">Firefox ESR version 91.10.0 ESR</a></li><li>• <a href="#">VSCode VSCode 1.68.1</a></li><li>• <a href="#">NICE DCV Server version 2022.0.12760</a></li><li>• <a href="#">Blender version 3.2.0</a></li></ul>          |

| Release      | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| May 23, 2022 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to May 17, 2022</li><li>• <a href="#">Nvidia driver 510.47.03</a></li><li>• <a href="#">Firefox ESR version 91.9.0 ESR</a></li><li>• <a href="#">VSCode VSCode 1.67.1</a></li><li>• <a href="#">VLC 3.0.17.2</a></li></ul> |
| May 4, 2022  | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to May 4, 2022</li><li>• <a href="#">DCV version 2022.0-12123</a></li><li>• <a href="#">Firefox ESR version 91.8.0 ESR</a></li><li>• <a href="#">Blender 3.1.2</a></li><li>• <a href="#">VSCode 1.66.2</a></li></ul>       |

| Release          | Changes                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| March 11, 2022   | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to March 9, 2022</li><li>• DCV version 2022.0-11954</li><li>• Firefox ESR version 91.6.0 ESR</li><li>• Blender 3.0.1</li><li>• VSCode 1.64.2</li></ul> |
| January 27, 2022 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to January 27, 2022</li><li>• Firefox ESR version 91.5.0 ESR</li><li>• DCV Server version 2021.3</li></ul>                                             |

## Nimble Studio Windows 2022 worker AMI

The following table describes the release history for the Nimble Studio Windows worker AMI.

| Release          | Changes                                                                                                                                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| December 4, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> |

| Release           | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| July 26, 2023     | <ul style="list-style-type: none"><li>Windows security updates current to November 15, 2023</li></ul> <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to July 19, 2023</li><li><a href="#">Uses IMDSv2 by default</a></li></ul> |
| April 3, 2023     | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to April 3, 2023</li></ul>                                                                                                                                                      |
| February 23, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to February 15, 2023</li></ul>                                                                                                                                                  |

| Release          | Changes                                                                                                                                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| January 26, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to January 26, 2023</li></ul>                                       |
| December 2, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to November 10, 2022</li></ul>                                      |
| October 21, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to October 12, 2022</li><li><a href="#">Blender 3.3.1</a></li></ul> |

| Release         | Changes                                                                                                                                                                                                                                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| October 4, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to September 20, 2022.</li><li>• <a href="#">Blender 3.3.0</a></li></ul> |

## Nimble Studio Windows 2019 worker AMI

The following table describes the release history for the Nimble Studio Windows worker AMI.

| Release          | Changes                                                                                                                                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| December 4, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to November 15, 2023</li></ul>                                              |
| July 26, 2023    | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to July 25, 2023</li><li>• <a href="#">Uses IMDSv2 by default</a></li></ul> |

| Release           | Changes                                                                                                                                                                                                                                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| April 3, 2023     | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to April 3, 2023</li></ul>     |
| February 23, 2023 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to February 15, 2023</li></ul> |
| January 26, 2023  | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to January 26, 2023</li></ul>  |
| December 2, 2022  | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to November 10, 2022</li></ul> |

| Release            | Changes                                                                                                                                                                                                                                                                                                                                                        |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| October 21, 2022   | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to October 12, 2022</li><li>• <a href="#">Blender 3.3.1</a></li></ul>   |
| September 27, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to September 20, 2022</li><li>• <a href="#">Blender 3.3.0</a></li></ul> |
| August 25, 2022    | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to August 9, 2022</li><li>• <a href="#">Blender 3.2.2</a></li></ul>     |

| Release        | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| July 21, 2022  | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to July 19, 2022</li><li>• <a href="#">Blender version 3.2.1</a></li><li>• <a href="#">Houdini 19.0.657</a></li><li>• <a href="#">Deadline 10.1.21.4</a></li></ul> |
| July 6, 2022   | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to June 21, 2022</li><li>• <a href="#">Blender version 3.2.0</a></li></ul>                                                                                         |
| April 28, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• Windows security updates current to April 28, 2022</li><li>• <a href="#">Blender 3.1.2</a></li></ul>                                                                                                |

| Release          | Changes                                                                                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| March 11, 2022   | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to March 9, 2022</li></ul>    |
| January 27, 2022 | <p>This release contains operating system updates for Windows and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>Windows security updates current to January 27, 2022</li></ul> |

## Nimble Studio Linux worker AMI

The following table describes the release history for the Nimble Studio Linux worker AMI.

| Release          | Changes                                                                                                                                                                                                                                                                                                   |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| December 4, 2023 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>CentOS-7 updates current to November 15, 2023</li></ul> |
| July 13, 2023    | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p>                                                                                               |

| Release           | Changes                                                                                                                                                                                                                                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>on AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to July 6, 2023</li><li>• <a href="#">Uses IMDSv2 by default</a></li></ul>                                                                                         |
| April 3, 2023     | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstations on AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to April 3, 2023</li></ul>     |
| February 23, 2023 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstations on AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to February 15, 2023</li></ul> |
| January 26, 2023  | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstations on AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to January 26, 2023</li></ul>  |

| Release            | Changes                                                                                                                                                                                                                                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| December 2, 2022   | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstations AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to November 10, 2022</li></ul>                                          |
| October 21, 2022   | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstations AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to October 12, 2022</li><li>• <a href="#">Blender 3.3.1</a></li></ul>   |
| September 27, 2022 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstations AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to September 20, 2022</li><li>• <a href="#">Blender 3.3.0</a></li></ul> |

| Release         | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| August 25, 2022 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to August 9, 2022</li><li>• <a href="#">Blender 3.2.2</a></li></ul>                                                                                                                                        |
| July 21, 2022   | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to July 19, 2022</li><li>• <a href="#">Blender version 3.2.1</a></li><li>• <a href="#">Houdini 19.0.657</a></li><li>• <a href="#">Deadline 10.1.21.4</a></li><li>• <a href="#">Python 3.7.13</a></li></ul> |
| July 6, 2022    | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstation AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to June 21, 2022</li><li>• <a href="#">Blender version 3.2.0</a></li></ul>                                                                                                                                 |

| Release          | Changes                                                                                                                                                                                                                                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| May 4, 2022      | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstations AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to April 28, 2022</li><li>• <a href="#">Blender 3.1.2</a></li></ul> |
| March 11, 2022   | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstations AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to March 9, 2022</li></ul>                                          |
| January 27, 2022 | <p>This release contains operating system updates for Linux and application updates. For instructions about how to update workstations AMIs, go to the <a href="#">Update AMIs: Setting up</a> tutorial.</p> <ul style="list-style-type: none"><li>• CentOS-7 updates current to January 27, 2022</li></ul>                                       |

## StudioBuilder v1.1.13 release notes - May 12, 2023

Nimble Studio is pleased to announce StudioBuilder version 1.1.13. This new version includes the following changes. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

To avoid issues while you're updating your studio, follow these best practices:

- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that are created by StudioBuilder.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
4. Choose **Actions**, then select **Copy to new**.
  - This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.

## Updates

- Updates StudioBuilder to use AWS Thinkbox Deadline 10.2.1.1.
- Updates StudioBuilder to use AWS Cloud Development Kit (AWS CDK) v2.70.0.
- Updates StudioBuilder to use AWS Render Farm Development Kit 1.2.0.
- Fixes an issue that caused new studio deployments to fail due to recent changes to the [default S3 security settings](#).

## StudioBuilder v1.1.12 release notes - January 25, 2023

Nimble Studio is pleased to announce StudioBuilder version 1.1.12. This new version includes the following changes. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

To avoid issues while you're updating your studio, follow these best practices:

- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that are created by StudioBuilder.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
4. Choose **Actions**, then select **Copy to new**.
  - This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.

## Updates

- Requires the use of [IMDSv2](#) to fetch metadata of EC2 instances spun up by StudioBuilder.

## StudioBuilder v1.1.11 release notes - December 8, 2022

Nimble Studio is pleased to announce StudioBuilder version 1.1.11. This new version includes the changes listed in the [Updates](#) section. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

To avoid issues while you're updating your studio, follow these best practices:

- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that are created by StudioBuilder.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
4. Choose **Actions**, then select **Copy to new**.
  - This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.

## Updates

- Uses the latest version of cfn-signal (v2.0.19) to unblock ADAutoConfigurationInstance deployment failure.
- Customizes the Active Directory name for new studios. Existing studios won't be affected.

# StudioBuilder v1.1.10 release notes - October 18, 2022

Nimble Studio is pleased to announce StudioBuilder version 1.1.10. This new version includes the changes listed in the [Updates](#) section. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

To avoid issues while you're updating your studio, follow these best practices:

- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that are created by StudioBuilder.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
4. Choose **Actions**, then select **Copy to new**.
  - This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.

## Updates

- Introduces Windows2022 and renames Windows to Windows2019 as options when selecting the operating system for a render fleet. This allows customers to leverage new [Windows Server 2022 Deadline Farm Worker AMI](#) that's available in the AWS Marketplace.

## StudioBuilder v1.1.9 release notes - July 27, 2022

Nimble Studio is pleased to announce StudioBuilder version 1.1.9. This new version includes the changes listed in the [Updates](#) section. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

To avoid issues while you're updating your studio, follow these best practices:

- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that are created by StudioBuilder.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

- Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
- Choose **Launch profiles** in the left navigation pane.
- Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
- Choose **Actions**, then select **Copy to new**.

- This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.

## Updates

- Change term “AWS Single Sign-On (AWS SSO)” to “AWS IAM Identity Center”.

## StudioBuilder v1.1.8 release notes - July 8, 2022

Nimble Studio is pleased to announce StudioBuilder version 1.1.8. This new version includes the changes listed in the [Updates](#) section. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

This StudioBuilder update can leave orphaned Spot Fleet Requests in your studio renderfarm if you don't follow these instructions:

- This update will reconfigure the Spot Event Plugin settings in Deadline to use a new Spot Fleet Request configuration. **If you have active Spot Fleet Requests created by the Spot Event Plugin, upgrading to StudioBuilder 1.1.8 and redeploying your render farm will orphan those Spot Fleet Requests. We highly recommend following these instructions to upgrade to StudioBuilder 1.1.8.**
1. Disable the Spot Event Plugin in Deadline. For more information, see the [Spot Event Plugin "State" option in Deadline](#) page in the Deadline documentation
  2. Cancel any Spot Fleet Requests created by the Spot Event Plugin by following the instructions in the [Cancel a Spot Fleet request](#) tutorial in the Amazon EC2 User Guide for Linux Instances.
  3. Upgrade to StudioBuilder 1.1.8 and redeploy your render farm by following the instructions in Update to latest StudioBuilder version [Update to latest StudioBuilder version](#).

- Once the deployment is complete, re-enable the Spot Event Plugin in Deadline. For more information, see the [Spot Event Plugin "State" option in Deadline](#) page in the Deadline documentation.

To avoid issues while you're updating your studio, follow these best practices:

- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that are created by StudioBuilder.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

- Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
- Choose **Launch profiles** in the left navigation pane.
- Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
- Choose **Actions**, then select **Copy to new**.
  - This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.

## Updates

- Update Deadline version to [10.1.21.4](#)
  - If you have a custom AMI, update your Deadline client to 10.1.21.4.
  - Run the new version of StudioBuilder (1.1.8) to update your existing compute farm component if you have an existing studio and are using the default workstation AMI. Rendering on the farm will break with the next workstation AMI release if you have not run the latest version of StudioBuilder.

- Fix the bug that prevented UBL license forwarders from connecting to public 3rd party license servers.

## StudioBuilder v1.1.7 release notes - April 22, 2022

Nimble Studio is pleased to announce StudioBuilder version 1.1.7. This new version includes the changes listed in the [Updates](#) section. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

This StudioBuilder update can leave orphaned Spot Fleet Requests in your studio renderfarm if you don't follow these instructions:

- This update will reconfigure the Spot Event Plugin settings in Deadline to use a new Spot Fleet Request configuration. **If you have active Spot Fleet Requests created by the Spot Event Plugin, upgrading to StudioBuilder 1.1.7 and redeploying your render farm will orphan those Spot Fleet Requests. We highly recommend following these instructions to upgrade to StudioBuilder 1.1.7.**
1. Disable the Spot Event Plugin in Deadline. For more information, see the [Spot Event Plugin "State" option in Deadline](#) page in the Deadline documentation.
  2. Cancel any Spot Fleet Requests created by the Spot Event Plugin by following the instructions in the [Cancel a Spot Fleet request](#) tutorial in the Amazon EC2 User Guide for Linux Instances.
  3. Upgrade to StudioBuilder 1.1.7 and redeploy your render farm by following the instructions in Update to latest StudioBuilder version [Update to latest StudioBuilder version](#).
  4. Once the deployment is complete, re-enable the Spot Event Plugin in Deadline. For more information, see the [Spot Event Plugin "State" option in Deadline](#) page in the Deadline documentation

To avoid issues while you're updating your studio, follow these best practices:

- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that are created by StudioBuilder.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
4. Choose **Actions**, then select **Copy to new**.
  - This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.

## Updates

- Customers can now deploy Nimble Studio infrastructure in the Tokyo AWS Region.
- Deadline SpotEvenPluginFleet now uses EC2 Launch Templates instead of Launch Specifications.

## StudioBuilder v1.1.6 release notes - March 28, 2022

Nimble Studio is pleased to announce StudioBuilder version 1.1.6. This new version includes the changes listed in the [Updates](#) section. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

To avoid issues while you're updating your studio, follow these best practices:

- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that are created by StudioBuilder.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
4. Choose **Actions**, then select **Copy to new**.
  - This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.

## Updates

- Customers can now deploy Nimble Studio infrastructure into a Member account that is part of an AWS Organization.

[AWS Organizations](#) helps you centrally govern your environment as you scale your workloads on AWS. Organizations helps you allocate resources, programmatically create new accounts, create groups of accounts to organize your workflows, apply policies to help govern these groups, and set up a single payment method for all of your accounts. AWS Organizations is integrated with other AWS services. This means you can define central configurations, security mechanisms, and resource sharing across multiple accounts in your organization.

# StudioBuilder v1.1.5 release notes - December 10, 2021

Nimble Studio is pleased to announce StudioBuilder version 1.1.5. This new version includes changes listed in the [Updates](#) section. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

To avoid issues while you're updating your studio, follow these best practices:

- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that are created by StudioBuilder.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
  2. Choose **Launch profiles** in the left navigation pane.
  3. Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
  4. Choose **Actions**, then select **Copy to new**.
    - This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.
- If you have an existing studio and want to upgrade to use **Thinkbox Usage Based Licensing (UBL)**, update your existing render worker and streaming AMIs to the latest Deadline (10.1.19.x or higher). Then update your studio with the new version of StudioBuilder (1.1.5).

- If you have an existing studio and are using the default workstation AMI, you only need to run the new version of StudioBuilder (1.1.5) to update your existing compute farm component. If you don't, rendering on the farm will break.
- If you don't want to update your studio with StudioBuilder 1.1.5 and you want to continue using the default workstation AMI, then you need to update your compute farm component.
  1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
  2. Choose **Studio resources** in the left navigation pane.
  3. Select the **Compute Farm** component.
  4. Choose **Action**. Then choose **Edit**.
  5. Edit the **User Initialization scripts** for Windows and Linux in the **Initialization scripts** section
    - Replace `./deadlinecommand -ChangeRepositorySkipValidation Proxy renderqueue.<studio_name>.nimble.<region>.aws:4433. <path>` with `./deadlinecommand.exe -SetIniFileSetting ProxyRoot renderqueue.<studio_name>.nimble.<region>.aws:4433`
  6. Select **Update launch profile**.

## Updates

- Customers can now deploy infrastructure necessary to use UBL from the [AWS Thinkbox Marketplace](#) directly from StudioBuilder. Customers can purchase render licenses for popular digital content creation (DCC) software applications and use them for Deadline-based render tasks.

 **Note**

For UBL support, customers need to update to the latest render worker and streaming AMIs. This includes updating the Deadline client to version 10.1.19. For more information, see the [Setting up Deadline Usage Based Licensing with Nimble Studio](#).

- Customers can now choose multiple instance types when using Spot rendering on the farm through StudioBuilder.
- Customers can now deploy the necessary infrastructure to use a Linux operating system for content creation from StudioBuilder. This Linux support makes it easier for customers to deploy

FSx for Lustre and the infrastructure needed to support home directories and user profiles when using Linux workstations.

- Customers can deploy new storages from StudioBuilder. Along with FSx for Lustre and FSx for Windows File Server, customers can also deploy Amazon Elastic File System (Amazon EFS) to their Nimble Studio.
- Customers can now deploy resources in the Los Angeles Local Zone (usw2-lax1-az1). The LA Local Zone provides lower latency for customers closer to California than Oregon. This allows them to create content using virtual workstations and storage.
- Uses [Deadline 10.1.19.4](#)

## StudioBuilder v1.1.4 release notes - October 6, 2021

Nimble Studio is pleased to announce StudioBuilder version 1.1.4 on October 6, 2021. This new version includes bug fixes and other changes listed in the [Bug fixes](#) section. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

To avoid issues while you're updating your studio, follow these best practices:

- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that are created by StudioBuilder.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.

3. Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
4. Choose **Actions**, then select **Copy to new**.
  - This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.

## Updates

- Customers can now build a studio without a render farm, and if needed later, can come back to StudioBuilder to deploy a render farm.
- Customers can now delete an existing render farm. Make sure you remove the respective Compute farm component from the launch profiles before deleting the farm. For more details, see [How to delete a render farm built by StudioBuilder](#).
- Deadline server version updated to 10.1.17. Customer are recommended to update Deadline client version to 10.1.17 on any custom AMI they have created.
- Diagnostic data and metrics collection for StudioBuilder.
  - For more information, see **Diagnostic data and metrics** in the [Data protection in Amazon Nimble Studio](#).
- Add GNOME settings to the instance configuration studio component related to the screen blank/lock in Linux workstations.
- Uses [Deadline 10.1.17.4](#)

## Bug fixes

- Path Mapping is now configured in Deadline when a new studio is deployed.
- StudioBuilder displays an error message when the AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) passwords don't match while creating the studio.

# StudioBuilder v1.1.3 release notes - July 22, 2021

Nimble Studio is pleased to announce StudioBuilder version 1.1.3 on July 22, 2021. This new version includes bug fixes, two more instance types, and other changes listed in the [Bug fixes](#) section. To create your new studio or update your existing studio, follow these tutorials:

- To create a new studio, see the [Deploying a new studio with StudioBuilder](#) tutorial.
- To update an existing studio, see the [Update to latest StudioBuilder version](#) tutorial.

## Important notes

To avoid issues while you're updating your studio, follow these best practices:

- If you have added any custom Network Access Control List (ACL) rules, check that your rule numbers are at 30,000 (or greater).
- To review your current network ACLs, go to the [Amazon VPC console](#) and choose **Network ACLs** from the left navigation panel.
- To learn more about the network ACLs that StudioBuilder creates and how they work, see [Network ACLs for Nimble Studio](#).
- StudioBuilder will overwrite any components added to the default **Workstation-Default** launch profile and the **RenderWorker-Default** launch profile that StudioBuilder creates.
- If you have modified your **Workstation-Default** and **RenderWorker-Default** launch profiles, and want to keep your modifications, we recommend that you follow these steps.

 **Note**

All other launch profiles won't be modified.

1. Sign in to the AWS Management Console and open the [Nimble Studio](#) console.
2. Choose **Launch profiles** in the left navigation pane.
3. Select the launch profiles that you want to keep the same (**Workstation-Default** and **RenderWorker-Default**).
4. Choose **Actions**, then select **Copy to new**.

- This will create a copy that you can use for reference. Only the **RenderWorker-Default** launch profile is used by the farm and the copy will just be used for reference if modified from the original.

## Updates

- Default workstation launch profile created by StudioBuilder now includes g4dn.4xlarge and g4dn.8xlarge instance types.
- Diagnostic data and metrics collection for StudioBuilder.
  - For more information, see **Diagnostic data and metrics** in the [Data protection in Amazon Nimble Studio](#).
- Uses [Deadline 10.1.14.5](#)

## Bug fixes

- Allows admins to set render workers to 0.
- Enforces lowercase group names when defining fleets.
- Enforces password policies that align to [AWS Directory Service](#) standards.
- Retains user preferences when using Windows instances.
- Render worker launch templates have been renamed to be more descriptive.
- InstanceConfiguration studio component now applies custom umask configurations for Linux GUI applications and Linux terminal.

# Document History for Nimble Studio Onboarding Guide

The following table describes the documentation for this release of Nimble Studio.

- API version: latest
- Latest documentation update: November 1, 2021

| Change                                                      | Description                                                                                                                                                                                              |                   |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| New page:<br>Create EC2 instances similar to workstations   | Learn how to create an Amazon EC2 instance that has similar properties to Nimble Studio workstations. For more information, see <a href="#">Setting up EC2 instances like workstations</a> .             | July 28, 2022     |
| New page:<br>Export AMIs and backups                        | Learn how to export AMIs and backups. For more information, see <a href="#">Exporting AMIs and backups from streaming sessions</a> .                                                                     | July 25, 2022     |
| Updated guide:<br>Classic guide                             | The Administrator Guide and Artist Guide were combined into the Nimble Studio Classic Guide. A new page was added with information about <a href="#">Migrating your studio</a> .                         | June 19, 2022     |
| New page:<br>Session auto backup                            | Learn how to back up your streaming sessions. For more information, see <a href="#">Session auto backup</a> .                                                                                            | December 21, 2022 |
| Updated page:<br>Start and stop workstations                | Learn how to turn off persistence. For more information, see <a href="#">Turn off persistent workstations</a> .                                                                                          | December 21, 2022 |
| Updated managed policies:<br>AmazonNimbleStudio-StudioAdmin | These managed policies were updated to allow studio admins and studio users to view their workstation backups. For more information, see <a href="#">AWS managed policies for Amazon Nimble Studio</a> . | December 20, 2021 |

| Change                                                       | Description                                                                                                                                                                    |                  |
|--------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| and AmazonNimbleStudio-StudioUser                            |                                                                                                                                                                                |                  |
| New page:<br>Administrator access for Windows users          | Learn how to provide secure user access to your Windows users. For more information, see <a href="#">Provide administrator access for Windows users</a> .                      | October 31, 2022 |
| New page:<br>Removing studio users                           | Learn how remove studio users after completing a project. For more information, see <a href="#">Removing studio users</a> .                                                    | October 10, 2022 |
| New Guide:<br>Nimble Studio Artist Guide                     | The content in <a href="#">Artist tutorials for Amazon Nimble Studio</a> has been moved.                                                                                       | October 1, 2022  |
| Removed page:<br>Deploy a new studio using AWS Organizations | The page, Deploying Amazon Nimble Studio using AWS Organizations, has been removed.                                                                                            | October 1, 2022  |
| New page:<br>Monitoring                                      | Learn how Nimble Studio monitors the reliability, availability, and performance of your cloud studio. For more information, see <a href="#">Monitoring Nimble Studio</a> .     | July 15, 2022    |
| New page:<br>Create custom configurations                    | Learn how to create a custom configuration for your studio, and view custom configuration examples. For more information, see <a href="#">Creating custom configurations</a> . | July 7, 2022     |
| Removed page:<br>Deploy a studio by hand                     | The page, Deploying a new studio with StudioBuilder, has been removed.                                                                                                         | July 6, 2022     |

| Change                                                              | Description                                                                                                                                                                                          |                |
|---------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Updated page:<br>Delete studio                                      | The <a href="#">How to delete a studio</a> page was updated to a more comprehensive tutorial with less likelihood of running into errors.                                                            | June 29, 2022  |
| New page:<br>Provision<br>workstations in<br>multiple AZs or<br>LZs | Learn how to launch workstations in multiple Availability Zones. For more information, see <a href="#">Provisioning workstations in multiple Availability Zones or Local Zones</a> .                 | June 1, 2022   |
| New page: Set<br>up RLM license<br>server                           | Learn how to set up a Reprise License Manager (RLM) license server. For more information, see <a href="#">Setting up an RLM license server</a> .                                                     | May 23, 2022   |
| New page:<br>Rotating<br>certificates                               | Learn how to manually rotate the root CA and the render queue certificate. For more information, see <a href="#">Rotating certificates created in Nimble Studio</a> .                                | April 4, 2022  |
| New page:<br>Deploy a new<br>studio using<br>AWS Organiza<br>tions  | Learn how to deploy a Nimble Studio in a member account. For more information, see <a href="#">Deploying a new studio with StudioBuilder</a> .                                                       | April 4, 2022  |
| New page:<br>Incredibuild                                           | Learn how to set up the Incredibuild software acceleration technology. For more information, see <a href="#">Set up Incredibuild on Nimble Studio</a> .                                              | March 23, 2022 |
| New page:<br>Perforce Helix<br>Core                                 | Learn how to set up a Perforce Helix Core server on AWS which is accessible to users within a Nimble Studio. For more information, see <a href="#">Set up Perforce Helix Core on Nimble Studio</a> . | March 23, 2022 |
| New page:<br>Troubleshooting                                        | Learn how to troubleshoot issues with your render farm, home directory, initialization scripts, and more. For more information, see <a href="#">Troubleshooting Nimble Studio</a> .                  | March 7, 2022  |

| Change                                                       | Description                                                                                                                                                                                                                                                                                            |                   |
|--------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| New page: AMI release notes                                  | Learn about the latest AMI release and read a history of previous AMI releases. For more information, see <a href="#">Nimble Studio AMI release notes</a> .                                                                                                                                            | February 21, 2022 |
| New page: Set up Nuke license server                         | Learn how to set up a Nuke license server. For more information, see <a href="#">Setting up a Nuke license server</a> .                                                                                                                                                                                | January 25, 2022  |
| New page: Mount the Deadline Repository file system on Linux | Learn how to set up mount the Deadline Repository file system on Linux workstations. For more information, see <a href="#">Mounting the Deadline Repository file system on Linux based studios</a> .                                                                                                   | January 24, 2022  |
| New page: Update Windows worker AMI                          | Learn how to update your Windows worker AMIs. For more information, see <a href="#">Update a Windows worker AMI</a> .                                                                                                                                                                                  | January 18, 2022  |
| New feature: Validate your launch profiles                   | You can now verify if your launch profile will launch without launching an instance. For information about how to validate your launch profiles, see <a href="#">Modifying launch profiles</a> .                                                                                                       | January 13, 2022  |
| New feature: Upload files to your workstations               | You can now upload files to your workstations. For information about how to enable this feature, see <a href="#">Enabling uploads to Nimble Studio workstations</a> . For information about how to upload files to your workstation, see <a href="#">Uploading files to your virtual workstation</a> . | December 20, 2021 |
| New page: Set up Weka                                        | Learn how to set up Weka in your studio. For more information, see <a href="#">Setting up Weka in Nimble Studio</a> .                                                                                                                                                                                  | December 20, 2021 |

| Change                                                                              | Description                                                                                                                                                                                                                                                                                                                     |                   |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Updated pages:<br>Deploy a new studio and<br>Update to latest StudioBuilder version | The <a href="#">Deploying a new studio with StudioBuilder</a> and <a href="#">Update to latest StudioBuilder version</a> pages were updated with the new StudioBuilder question flow. For more information about the new version of StudioBuilder, see <a href="#">StudioBuilder v1.1.5 release notes - December 10, 2021</a> . | December 10, 2021 |
| Updated page:<br>Setting up                                                         | The <a href="#">Setting up to use Nimble Studio</a> page was updated with a new section about how to <a href="#">(Optional) Opt in to the LA Local Zone</a> .                                                                                                                                                                   | December 10, 2021 |
| New feature: Set up Deadline UBL                                                    | You can now set up Deadline Usage Based Licensing (UBL) in your studio. For more information, see <a href="#">Setting up Deadline Usage Based Licensing with Nimble Studio</a> .                                                                                                                                                | December 10, 2021 |
| New page:<br>Update Linux worker AMI                                                | Learn how to update your Linux worker AMIs. For more information, see <a href="#">Update a Linux worker AMI</a> .                                                                                                                                                                                                               | December 3, 2021  |
| New page: Set up Qumulo                                                             | Learn how to set up Qumulo in your studio. .                                                                                                                                                                                                                                                                                    | November 19, 2021 |
| New page:<br>Deploy a studio by hand                                                | Learn how to deploy a studio by creating your own resources. For more information, see <a href="#">Deploying a new studio with StudioBuilder</a> .                                                                                                                                                                              | November 12, 2021 |
| New feature:<br>Perform a test launch                                               | You can now verify that your launch profiles can create a streaming session from the console. For more information, see <a href="#">Step 2: (Optional) Perform a test launch</a> .                                                                                                                                              | November 11, 2021 |

| Change                                                                          | Description                                                                                                                                                                                                 |                  |
|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Updated managed policies:<br>AmazonNimbleStudioAdmin and AmazonNimbleStudioUser | These managed policies were updated to allow studio admins and studio users to start and stop their workstations. For more information, see <a href="#">AWS managed policies for Amazon Nimble Studio</a> . | November 1, 2021 |
| New page:<br>Update Windows workstation AMI                                     | Learn how to update your Linux workstation AMIs. For more information, see <a href="#">Update a Linux workstation AMI</a> .                                                                                 | November 1, 2021 |
| New feature:<br>Start and stop your workstations                                | You can now stop your instances after use instead of terminating them. For more information, see <a href="#">Starting and stopping workstations</a> .                                                       | November 1, 2021 |
| Updated page: Provide Superuser access for Linux users                          | The <a href="#">Provide Superuser access for Linux users</a> page was updated with information about the custom configuration component.                                                                    | October 29, 2021 |
| New page:<br>Cross-service confused deputy prevention                           | Learn how to avoid the confused deputy deploy problem. For more information, see <a href="#">Cross-service confused deputy prevention</a> .                                                                 | October 18, 2021 |
| New page: How StudioBuilder works                                               | Learn what resources StudioBuilder creates when it sets up your studio. For more information, see <a href="#">How StudioBuilder works with Amazon Nimble Studio</a> .                                       | October 14, 2021 |

| Change                                                                              | Description                                                                                                                                                                                                                                                                                           |                    |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| Updated pages:<br>Deploy a new studio and<br>Update to latest StudioBuilder version | The <a href="#">Deploying a new studio with StudioBuilder</a> and <a href="#">Update to latest StudioBuilder version</a> pages were updated with a questions. For more information about the new version of StudioBuilder, see <a href="#">StudioBuilder v1.1.4 release notes - October 6, 2021</a> . | October 6, 2021    |
| New feature:<br>Delete a render farm                                                | You can now delete and replace your studio's render farm when you update your studio. For more information, see <a href="#">How to delete a render farm built by StudioBuilder</a> .                                                                                                                  | October 6, 2021    |
| New page: Back up studio data                                                       | Learn how to back up your studio's data using one of three options. For more information, see <a href="#">How to back up your studio data</a> .                                                                                                                                                       | September 22, 2021 |
| New service and guide                                                               | This is the initial release of Amazon Nimble Studio and the <b>Amazon Nimble Studio User Guide</b> .                                                                                                                                                                                                  | April 28, 2021     |

# AWS Glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS Glossary Reference*.