
AWS Edit in the Cloud Implementation Guide



AWS Edit in the Cloud: Implementation Guide

Copyright © 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

Welcome	1
Cost	2
Architecture	3
Components	4
Amazon EC2	4
Teradici	4
Amazon FSx for Windows File Server	4
Security	5
AWS Directory Service	5
IAM roles	5
Security groups	5
Design considerations	6
Quotas	6
AWS CloudFormation template	7
Automated deployment	8
Deployment overview	8
Step 1. Launch the stack	8
Step 2. Add Teradici license to the EC2 instance	11
Obtain the EC2 IP address	11
Activate the Teradici license	12
Connect to the edit host	12
Step 3. Mount the NVMe disk	12
Step 4. (Optional) Load data from Amazon S3	13
To verify that your files were synchronized from S3 to the FSx file share	13
Additional resources	14
AWS services	14
Blog posts	14
Uninstall the solution	15
Using the AWS Management Console	15
Using AWS Command Line Interface	15
Source code	16
Revisions	17
Contributors	18
Notices	19

Build a virtual, video editing environment on AWS using the AWS Edit in the Cloud solution

Publication date: *May 2021 (last update (p. 17): August 2021)*

AWS Edit in the Cloud helps you, the content creator, to build a virtual, video editing environment on AWS. This allows you to reach your editors and creative professionals where they reside and scale up and down elastically. There is no upfront cost for high-priced GPU workstations. And the solution saves you costs associated with transferring large assets to collaborators or shipping unencrypted drives using couriers.

The solution provides the following features:

- An edit host that runs your favorite video editing software for news, sports, promotional, or post-production work.
- A shared pool of high-performance network storage so that you can collaborate with other users on projects.
- A remote streaming application so that you can access your workstation remotely and securely without having to transfer content to on-premises.

This implementation guide describes architectural considerations and configuration steps for deploying AWS Edit in the Cloud in the Amazon Web Services (AWS) Cloud. It includes links to an [AWS CloudFormation](#) template that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for DevOps teams who have practical experience architecting in the AWS Cloud.

Cost

You are responsible for the cost of the AWS services used while running this solution. The total cost will depend on your activity. This depends on how long you keep your EC2 running, how much data you transfer into AWS, and other service costs that are associated with this solution. For example, this solution does not deploy an Amazon S3 bucket; however, it allows you to configure your own S3 bucket for media file storage.

The default EC2 this solution deploys is a g4dn.4xlarge EC2 instance for running Windows. As of May 2021, the cost of running this solution with the default settings in the US East (N. Virginia) Region is **\$1,323.04 per month**. This includes estimated charges for Amazon FSx, and AWS Directory Service. To save money, consider turning off your Windows EC2 instance when not in use. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

AWS service	Quantity	Monthly cost
AWS Directory Service	Base price: Standard x 730 hours (entire month)	\$87.60
Amazon EC2	On-Demand pricing: G4dn.4xlarge x 160 hours (work hours only) Outbound data transfer: 10 Mbps x 160 hours (work hours only) (4.5 GB x 160 hours)	\$192.64 \$64.80
Amazon FSx for Windows	Storage: 5 TB x \$0.130 GB-month Throughput: 16 MBps x \$4.50/MBps Backup: 5 TB x \$0.050/GB	\$650.00 \$72.00 \$256.00
	Total cost	\$1,323.04

Architecture

Deploying this solution with the default parameters builds the following environment in the AWS Cloud.

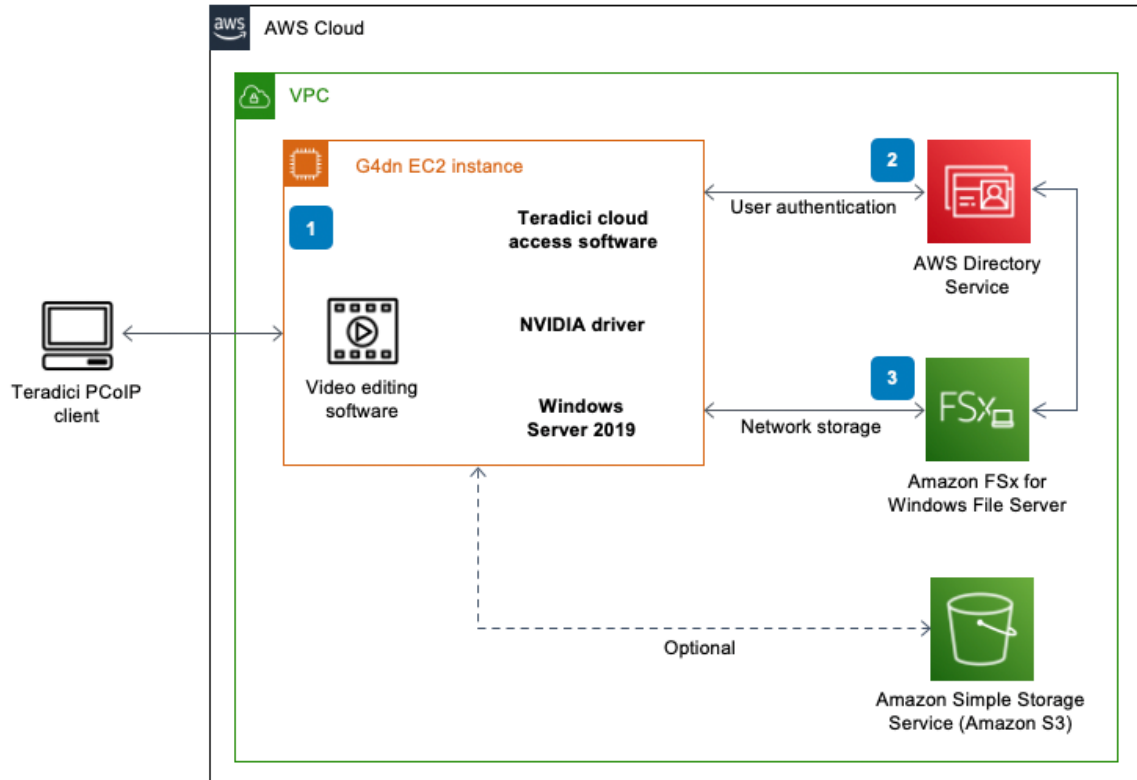


Figure 1: AWS Edit in the Cloud architecture on AWS

The AWS CloudFormation template deploys the following infrastructure:

1. An [Amazon Elastic Compute Cloud](#) (Amazon EC2) instance for Windows Server 2019 with [Teradici Cloud Access Software](#) and NVIDIA T4 GPU drivers for running your Non-Linear Editor (NLE) software of choice. Users will remote into the instance using [Teradici's PC-over-IP](#) (PCoIP) client.
2. [AWS Directory Service](#) for user authentication.
3. [Amazon FSx for Windows File Server](#) to access digital assets via the Amazon EC2 instance through your editor of choice. FSx for Windows File Server will auto-mount the network share upon startup of the Windows Amazon EC2 instance. FSx for Windows File Server will be used for storing your media assets to be used by your non-linear editor (NLE).

Note

This solution does not deploy an Amazon S3 bucket with associated IAM roles for access. However, it allows you to configure your own S3 bucket for media file storage.

Solution components

Amazon EC2

The Amazon EC2 G4 GPU instance family used in this solution provides the latest generation NVIDIA T4 Tensor Core GPU, AWS custom Intel Cascade Lake CPUs, up to 100 Gbps of networking throughput, and up to 1.8 TB of local NVME storage. These instances are well-suited for streaming graphics-intensive applications that rely on NVIDIA GPU libraries such as CUDA.

Teradici

Using [Teradici's PC-over-IP \(PCoIP\) technology](#), you can modify content by connecting to a remote workstation, instead of transferring large files to a local machine. The Teradici client runs on your desktop or laptop and gives access to an [EC2 G4dn instance](#).

We also recommend that you familiarize yourself with [Cloud Access Software](#) from Teradici that runs on the EC2 instance and serves up the PCoIP protocol.

Amazon FSx for Windows File Server

Amazon FSx for Windows File Server provides fully managed, highly reliable, and scalable file storage that is accessible over the industry-standard Server Message Block (SMB) protocol. It delivers a wide range of administrative features such as user quotas, end-user file restore, and Microsoft Active Directory (AD) integration. It also offers single-AZ and multi-AZ deployment options, fully managed backups, and encryption of data at rest and in transit.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This [shared responsibility model](#) reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit [AWS Cloud Security](#).

AWS Directory Service

This solution uses AWS Directory Service for user administration and authentication. Amazon FSx works with Microsoft Active Directory (AD) to integrate with your existing Microsoft Windows environments. Active Directory is the Microsoft directory service used to store information about objects on the network and make this information easy for administrators and users to find and use. These objects typically include shared resources such as file servers and network user and computer accounts.

IAM roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. This solution creates a role attached to the EC2 instance with least privilege, allowing access to specific S3 buckets (when applicable).

Security groups

The security groups created in this solution are designed to control and isolate network traffic between specified public IP ranges and the edit host instance. Security groups are also used to control access to Amazon FSx from the edit host. Microsoft Active Directory and Amazon FSx access is also controlled via security group access. For more information, refer to [Using Amazon FSx with AWS Directory Service for Microsoft Active Directory](#).

Design considerations

Quotas

Before deploying the solution, you must request a vCPU limit increase based on the G4dn instance size you want to run. For details about submitting a limit increase, refer to [How do I calculate an EC2 vCPU limit increase request for my On-Demand Instance?](#)

AWS CloudFormation template

To automate deployment, this solution uses AWS CloudFormation. It includes the following CloudFormation template, which you can download before deployment:

[View
Template](#)

aws-edit-in-the-cloud.template: Use this template to launch the solution and all associated components. The default configuration deploys Amazon Elastic Compute Cloud (Amazon EC2), [Amazon FSx for Windows File Server](#), AWS Directory Service and AWS Lambda. You can customize the template based on your specific needs.

Automated deployment

Before you launch the solution, review the architecture, network security, other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately one hour

Deployment overview

Use the following steps to deploy this solution on AWS. For detailed instructions, follow the links for each step.

[Step 1. Launch the stack \(p. 8\)](#)

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for required parameters: **Availability Zones**, **Edit Host Access CIDR**, **Key Pair Name**, and **Domain Admin Password**.
- Review the other template parameters, and adjust if necessary.

[Step 2. Add Teradici license to the EC2 instance \(p. 11\)](#)

- Obtain the EC2 IP address.
- Activate the Teradici license.

[Step 3. Mount the NVMe Disk \(p. 12\)](#)

- Use Windows PowerShell terminal to mount the NVMe Disk.

[Step 4. Load data from Amazon S3 \(p. 13\)](#)

- Use Windows PowerShell terminal to load data from Amazon S3.

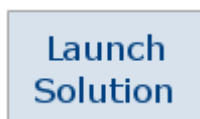
Step 1. Launch the stack

This automated AWS CloudFormation deploys Edit in the Cloud in the AWS Cloud.

Note

You are responsible for the cost of the AWS services used while running this solution. For more details, visit to the [Cost \(p. 2\)](#) section in this guide, and refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and select the button to launch the `aws-edit-in-the-cloud` AWS CloudFormation template.



Alternatively, you can [download the template](#) as a starting point for your own implementation.

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to [IAM and STS Limits](#) in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Network Configuration		
Availability Zones	<Requires input>	Select two Availability Zones to use for the VPC subnets. The template presents all Availability Zones available, select two.
VPC CIDR	10.0.0.0/16	CIDR block for the VPC that contains the Windows EC2 instance.
Private Subnet 1 CIDR	10.0.0.0/19	CIDR block for private subnet 1 located in Availability Zone 1.
Private Subnet 2 CIDR	10.0.32.0/19	CIDR block for private subnet 2 located in Availability Zone 2.
Public Subnet 1 CIDR	10.0.128.0/20	CIDR block for the public DMZ subnet 1 located in Availability Zone 1.
Public Subnet 2 CIDR	10.0.144.0/20	CIDR block for the public DMZ subnet 2 located in Availability Zone 2.
Amazon EC2 Configuration		
Amazon EC2 instance type	g4dn.4xlarge	The Amazon EC2 instance type for the video editing servers. Note: You can change the size of the EC2 instance that you can use for editing; however, the majority of the price is based on the size of the EC2 instance that you select here. Check the Cost (p. 2) section for more information.
Edit Host Access CIDR	<Requires input>	The CIDR block that is used for the security group on your Amazon EC2 instance editing host.

Parameter	Default	Description
		Note: Use a website such as, https://checkip.amazonaws.com to determine your public facing IP address for this step.
Key Pair Name	<Requires input>	The PEM key that allows you to securely connect to your EC2 instance after it launches. You can create a PEM key in the Amazon EC2 console.
Microsoft Active Directory Configuration		
Domain DNS Name	example.com	Fully qualified domain name (FQDN) of the forest root domain.
Domain NetBIOS Name	example	NetBIOS name of the domain (up to 15 characters) for users of earlier versions of Windows.
Domain Admin User	Admin	User name for the account that will be added as Domain Administrator. This is separate from the default Administrator account.
Domain Admin Password	<Requires input>	Password for the Domain Administrator user. Must be at least 8 characters containing letters, numbers, and symbols.
FSx Configuration		
FSx File System Size	1024	The storage capacity of the file system being created. 32 GiB - 65,536 GiB (64 TiB)
FSx Throughput	16	The throughput capacity of an Amazon FSx file system, measured in megabytes per second in powers of 2 (8, 16, 32, .., 1024), with the recommended value based on the file system size: 8 for <800 GiB, 16 for <1,600 GiB, ... 512 for <51,200 GiB, 1,024 for >=51,200 GiB
FSx AZ Deployment Mode	SINGLE_AZ_2	Specifies the file system deployment type.
Other Parameters		
Project	Cloud-video-edit-fsx	Used for billing and resource groups.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the boxes acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation Console in the **Status** column. You should receive a CREATE_COMPLETE status in approximately one hour.

Note

This solution includes the `custom-resource` AWS Lambda function, which runs only during initial configuration or when resources are updated or deleted. Do not delete this function because it is necessary to manage associated resources.

Step 2. Add Teradici license to the EC2 instance

After the stack deploys, use Windows Remote Desktop to add a Teradici license to the EC2 instance. First, obtain the public IP address of the EC2 edit host.

Obtain the EC2 IP address

1. Navigate to the [Amazon Elastic Compute Cloud console](#).
2. In the navigation pane, choose **Instances**.
3. Select the `win-host` EC2 instance
4. On the **Networking** tab, copy the value for **Public IPv4 address**.

Windows connection steps

1. Open **Remote Desktop**. If you can't find it, choose **Start**, enter **MSTSC**, and press **Enter**.
2. When prompted, enter the IP address of the edit host and choose **Connect**.
3. When prompted, enter the username and password. The username is the value that you entered for the **Domain Admin User** parameter (`Admin` by default). The password is the value that you entered for the **Domain Admin Password** parameter when launching the stack.

MacOS connection steps

If you use MacOS, you must download the [Microsoft Remote Desktop Client](#) from the App Store.

1. Open Microsoft Remote Desktop.
2. Select the **+** icon, then select **Add PC**.
3. For **PC Name**, enter the IP address of the edit host. You can specify a friendly name for easy reference, such as *Edit Host*.
4. Choose **Add**.
5. When the new PC appears in your connection options, select it to connect to the EC2 instance.
6. When prompted, enter the username and password. The username is the value that you entered for the **Domain Admin User** parameter (`Admin` by default). The password is the value that you entered for the **Domain Admin Password** parameter when launching the stack.

Activate the Teradici license

In order to use Teradici, you must [purchase a license](#) first.

1. Select the Teradici icon (refer to Figure 2), and choose **Licensing**.

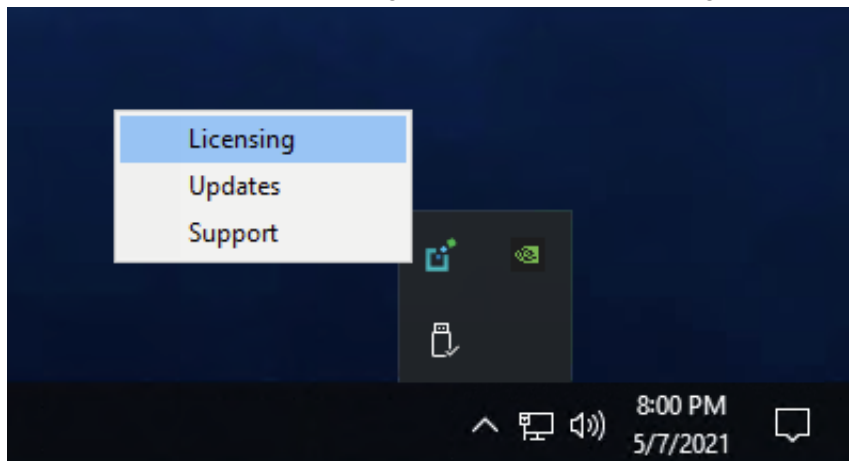


Figure 2: Teradici icon

2. In the **PCoIP Control Panel** box, enter your registration code, and choose **Register**.

Connect to the edit host

After activating your Teradici license, connect to the edit host using a [Teradici PCoIP Client](#).

1. On the connection screen, enter in the IP address of your edit host and a connection name for future reference.
2. When prompted for a username and password, use the same credentials you used earlier to connect to the instance using Windows Remote Desktop.

You are now connected to your edit host using Teradici.

Step 3. Mount the NVMe disk

In this deployment, we will mount ephemeral storage to the edit host for a scratch media edit space. Ephemeral storage provides low-latency to the host as it is physically attached storage.

Important

Any assets stored on this volume will be lost upon edit host reboot or termination. We recommend saving all project assets to the FSx file share.

1. Open Windows Powershell terminal:
 - a. Choose **Start**.
 - b. Enter "powershell" in the Search bar.
 - c. Right click the Windows Powershell icon and choose **Run as Administrator**.
 - d. If prompted with an alert to allow Windows Powershell to make changes on your device, select **Yes**.
2. In the Windows Powershell terminal, enter the following Powershell command to change to the correct directory:

```
cd C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts
```

3. Enter the following Powershell command to mount the NVMe disk:

```
.\InitializeDisks.ps1
```

The FSx file share mounts to the F drive upon user sign in.

Step 4. (Optional) Load data from Amazon S3

Follow these steps to move your media assets from Amazon S3 to Amazon FSx for Windows File Server.

1. Choose **Start**.
2. Enter "powershell" in the Search bar.
3. Select **Windows Powershell for AWS**.
4. Change to the FSx File Share drive mapping:

```
F:
```

5. Create a new directory for media:

```
mkdir Media
```

6. Use the following command to list the contents of your S3 bucket:

```
aws s3 ls s3://<bucket-name>
```

7. Use the following command to load the data from your S3 Bucket into the FSx File Share. *<key-prefix>* is any folder or object you want to load into the file share.

```
aws s3 sync s3://<bucket-name>/<key-prefix> .\Media\
```

To verify that your files were synchronized from S3 to the FSx file share

1. Navigate to your Windows File Explorer, then to your FSx File Share, which is mounted to the F: Drive.
2. From your file share, verify that the **Media** folder is there.
3. Open the **Media** folder and check that the contents of your S3 Bucket have synchronized.

Additional resources

AWS services

- [AWS CloudFormation](#)
- [AWS EC2](#)
- [AWS FSX](#)
- [AWS Directory Service](#)
- [Amazon S3](#)

Blog posts

- [Deploying your favorite post-production applications on AWS Virtual Desktop infrastructure](#)
- [Getting started with AWS cloud video editing](#)

Uninstall the solution

To uninstall this solution, you must delete the CloudFormation stack. If you added additional resources, such as an Amazon S3 bucket, you must delete these manually.

Using the AWS Management Console

1. Sign in to the [AWS CloudFormation console](#) console.
2. Select this solution's installation stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to [What Is the AWS Command Line Interface](#) in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Source code

Visit our solution [GitHub repository](#) to download the source files for this solution and to share your customizations with others. Refer to the [README.md file](#) for additional information.

Revisions

Date	Change
May 2021	Initial release
June 2021	Release v1.0.1: bug fix. For more information, refer to the CHANGELOG.md file in the GitHub repository.
August 2021	Release v2.0.0: Changed the default value of a field in the Active Directory sub-template to save cost. For more information, refer to the CHANGELOG.md file in the GitHub repository.

Contributors

- Leah Siddall
- Eddie Goynes
- Brandon Dold
- Mark Stephens
- Matt Herson

Notices

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

AWS Edit in the Cloud is licensed under the terms of the of the Apache License Version 2.0 available at [The Apache Software Foundation](#).