

Lesson 04 Demo 07

Creating and Mounting EFS on a Linux Server

Objective: To demonstrate the creation, customization, and mounting of an Amazon Elastic File System (EFS) across multiple AWS instances

Tools required: AWS Workspace

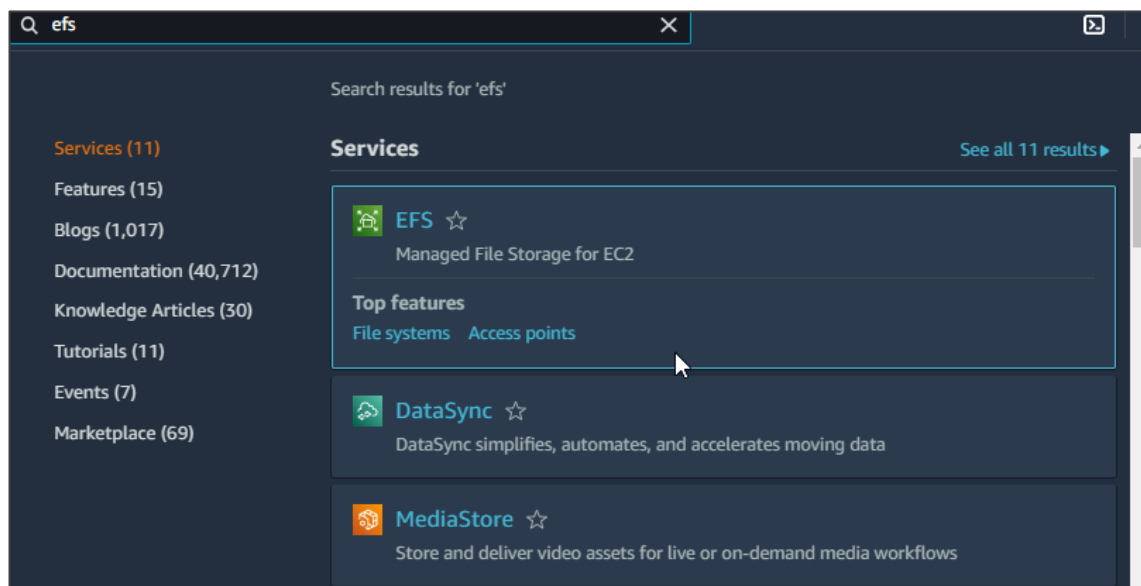
Prerequisites: AWS account with an S3 bucket created

Steps to be followed:

1. Create and customize an EFS
2. Create a security group to configure network access
3. Create AWS instances to access the EFS
4. Install EFS on the created instances

Step 1: Create and customize an EFS

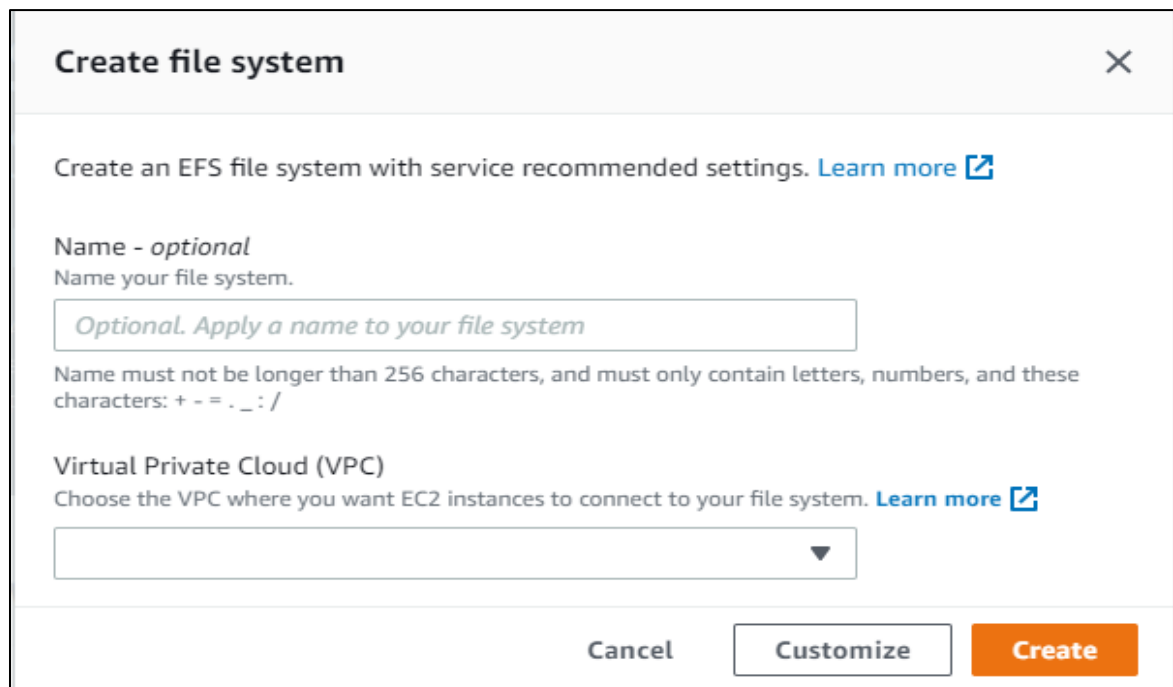
1.1 Navigate to the AWS Management Console homepage and search for the **EFS** service



1.2 Click **Create file system**

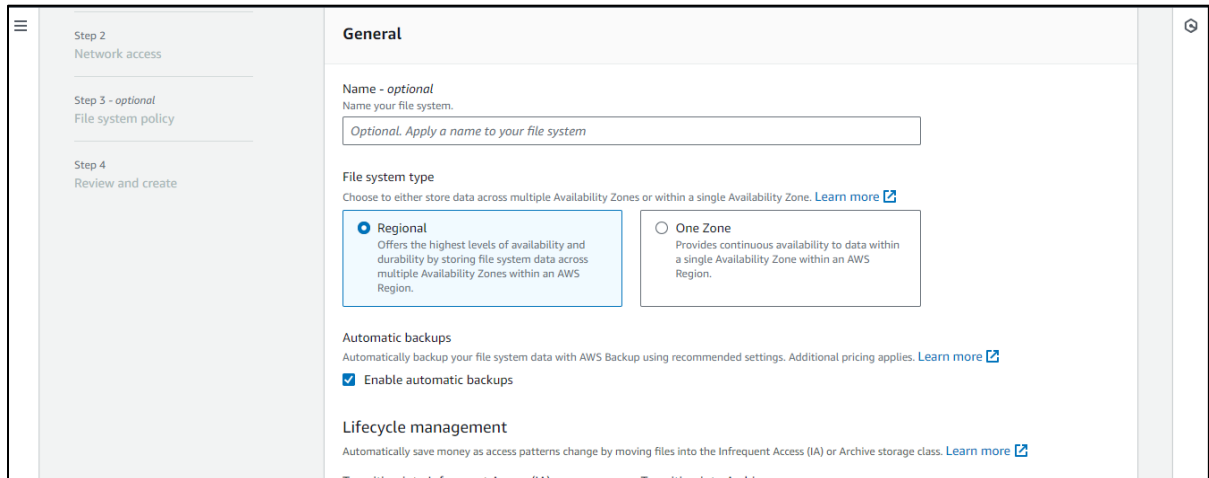


1.3 Click **Customize**



The screenshot shows the 'Create file system' dialog box in the AWS console. The dialog has a title bar with 'Create file system' and a close button. The main content area contains the following text: 'Create an EFS file system with service recommended settings. [Learn more](#)'. Below this, there is a section for 'Name - optional' with the instruction 'Name your file system.' and a text input field with the placeholder text 'Optional. Apply a name to your file system'. Below the input field, there is a note: 'Name must not be longer than 256 characters, and must only contain letters, numbers, and these characters: + - = . _ : /'. The next section is 'Virtual Private Cloud (VPC)' with the instruction 'Choose the VPC where you want EC2 instances to connect to your file system. [Learn more](#)'. Below this, there is a dropdown menu. At the bottom of the dialog, there are three buttons: 'Cancel', 'Customize', and 'Create'.

1.4 Name it **my-efs-demonstration**, select the options **Regional** and **Enhanced**, and then click **Next**



General

Name - optional
Name your file system.

File system type
Choose to either store data across multiple Availability Zones or within a single Availability Zone. [Learn more](#)

☒ **Regional**
Offers the highest levels of availability and durability by storing file system data across multiple Availability Zones within an AWS Region.

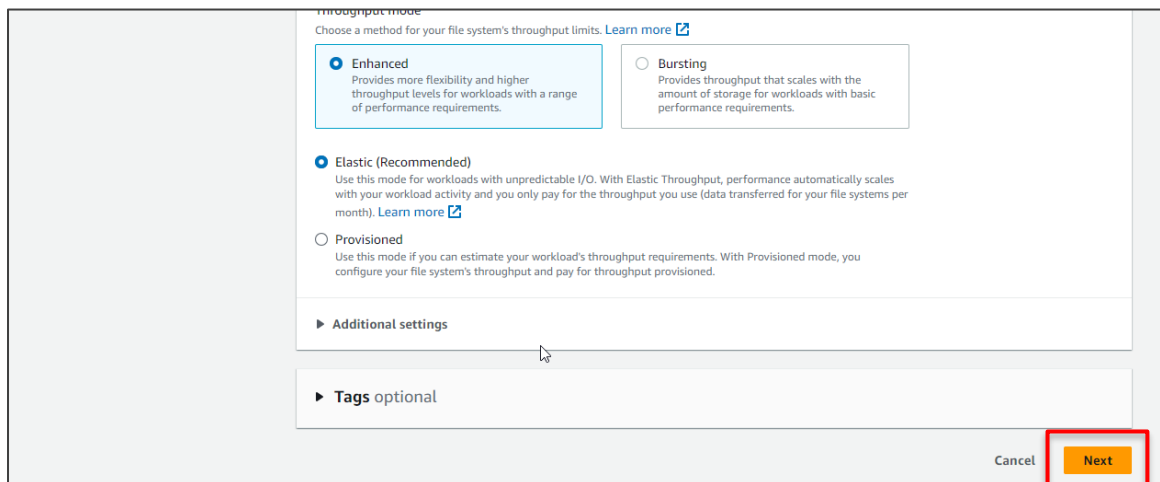
☐ **One Zone**
Provides continuous availability to data within a single Availability Zone within an AWS Region.

Automatic backups
Automatically backup your file system data with AWS Backup using recommended settings. Additional pricing applies. [Learn more](#)

☒ **Enable automatic backups**

Lifecycle management
Automatically save money as access patterns change by moving files into the Infrequent Access (IA) or Archive storage class. [Learn more](#)

[Transition into Infrequent Access \(IA\)](#) [Transition into Archive - new](#)



Throughput mode
Choose a method for your file system's throughput limits. [Learn more](#)

☒ **Enhanced**
Provides more flexibility and higher throughput levels for workloads with a range of performance requirements.

☐ **Bursting**
Provides throughput that scales with the amount of storage for workloads with basic performance requirements.

☒ **Elastic (Recommended)**
Use this mode for workloads with unpredictable I/O. With Elastic Throughput, performance automatically scales with your workload activity and you only pay for the throughput you use (data transferred for your file systems per month). [Learn more](#)

☐ **Provisioned**
Use this mode if you can estimate your workload's throughput requirements. With Provisioned mode, you configure your file system's throughput and pay for throughput provisioned.

► **Additional settings**

► **Tags optional**

Cancel **Next**

1.5 Click **Network access** and set it as default

1.6 Click **Next**

1.7 Review the file system and click **Create**

Step 2

Network access

Step 3 - optional

File system policy

Step 4

Review and create

Step 1: File system settings

Edit

File system

Field	Value	Is editable?
Name	my-efs-demonstration	Yes
Performance mode	General Purpose	No
Throughput mode	Bursting	Yes
Encrypted	Yes	No
KMS Key ID	-	No
Lifecycle management	Transition into IA: 30 day(s) since last access Transition out of IA: None	Yes
Automatic backups	Yes	Yes
VPC ID	vpc-01acb7a7e122e5ae3 (default)	Yes
Availability Zone	Standard	No

Cancel

Previous

Create

Elastic File System

File systems

Access points

AWS Backup

AWS DataSync

AWS Transfer

Documentation

Success!

File system (fs-02d9f17e3dde831d) is available.

View file system

Amazon EFS > File systems

File systems (1)

Filter by property values

Refresh

View details

Delete

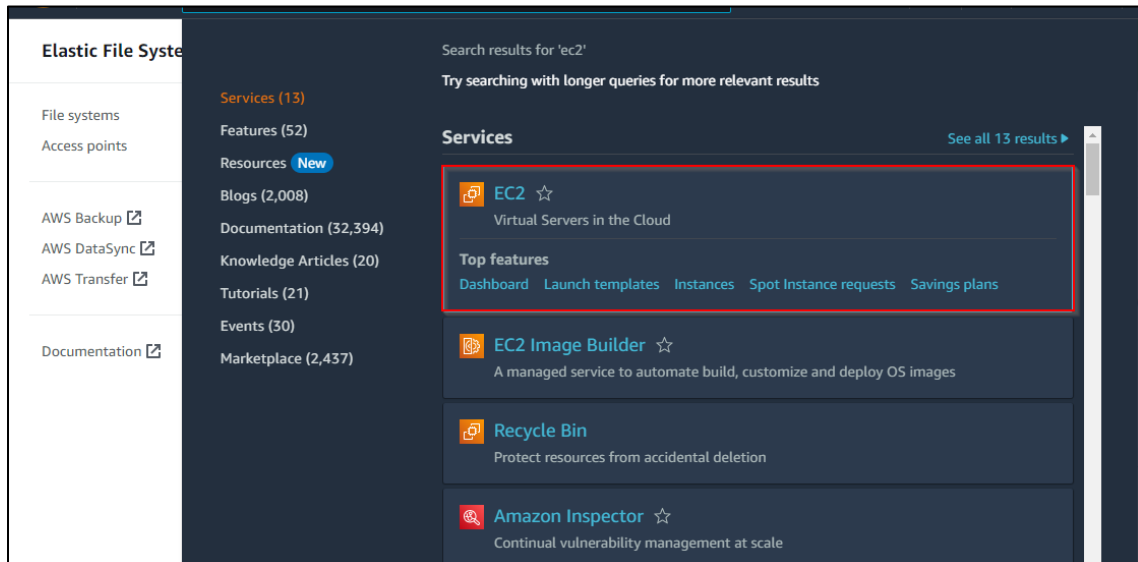
Create file system

	Name	File system ID	Encrypt	Total size	Size in Standard	Size in IA	Size in Archive
	my-efs-demonstration	fs-02d9f17e3dde831d	Encrypt	6.00 KiB	6.00 KiB	0 Bytes	0 Bytes

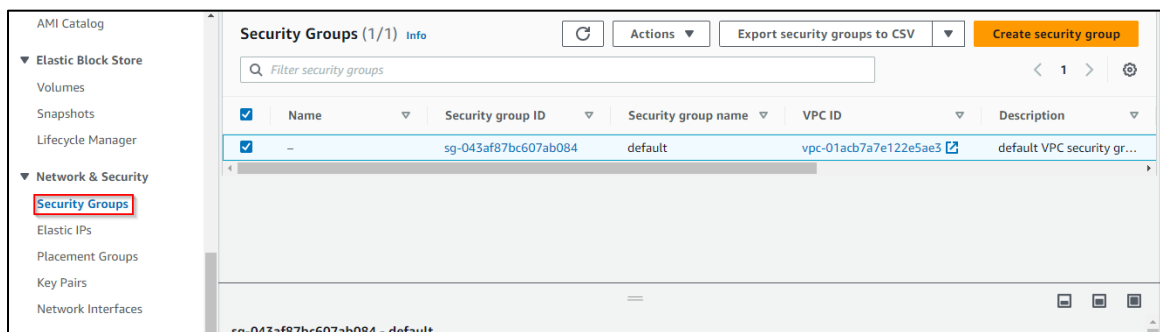
The EFS file has been successfully created.

Step 2: Create a security group to configure network access

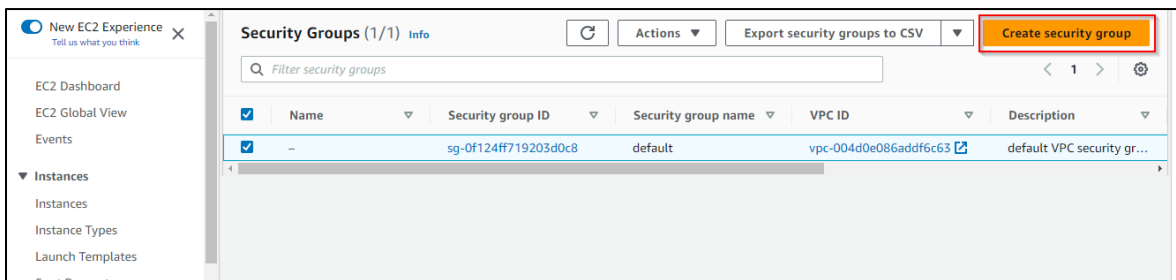
2.1 Navigate to the EC2 section and click on it



2.2 Click on Security Groups



2.3 Click on the **Create security group** button



2.4 Enter the security group name as **my-efs-demonstration** and enter a description

EC2 > Security Groups > Create security group

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, you must specify a name and a description.

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

2.5 Click on **Create security group**

Outbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info	
All traffic ▼	All	All	Custom ▼ <input type="text" value="0.0.0.0/0"/>	<input type="text"/>	Delete

[Add rule](#)

Tags - optional
 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.

No tags associated with the resource.

[Add new tag](#)
 You can add up to 50 more tags

[Cancel](#) [Create security group](#)

Security group (sg-0c3839c366569b724 | my-efs-demonstration) was created successfully

[Details](#)

EC2 > Security Groups > sg-0c3839c366569b724 - my-efs-demonstration

sg-0c3839c366569b724 - my-efs-demonstration [Actions](#)

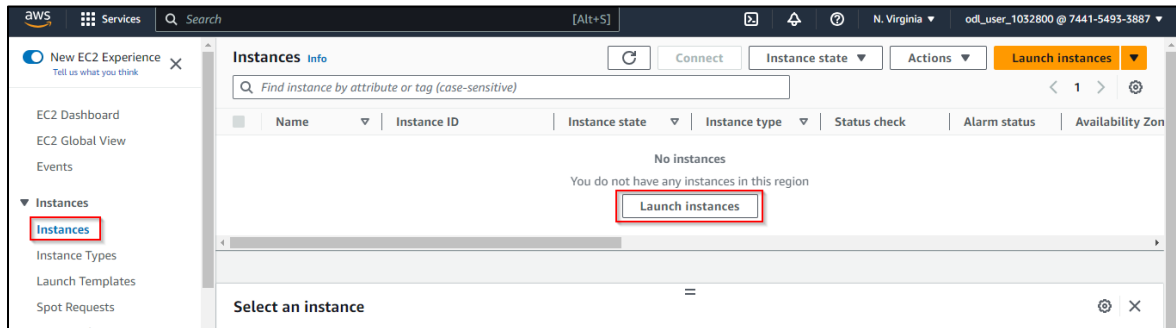
Details

Security group name my-efs-demonstration	Security group ID sg-0c3839c366569b724	Description SG for EFS	VPC ID vpc-01acb7a7e122e5ae3
Owner 744154933887	Inbound rules count 0 Permission entries	Outbound rules count 1 Permission entry	

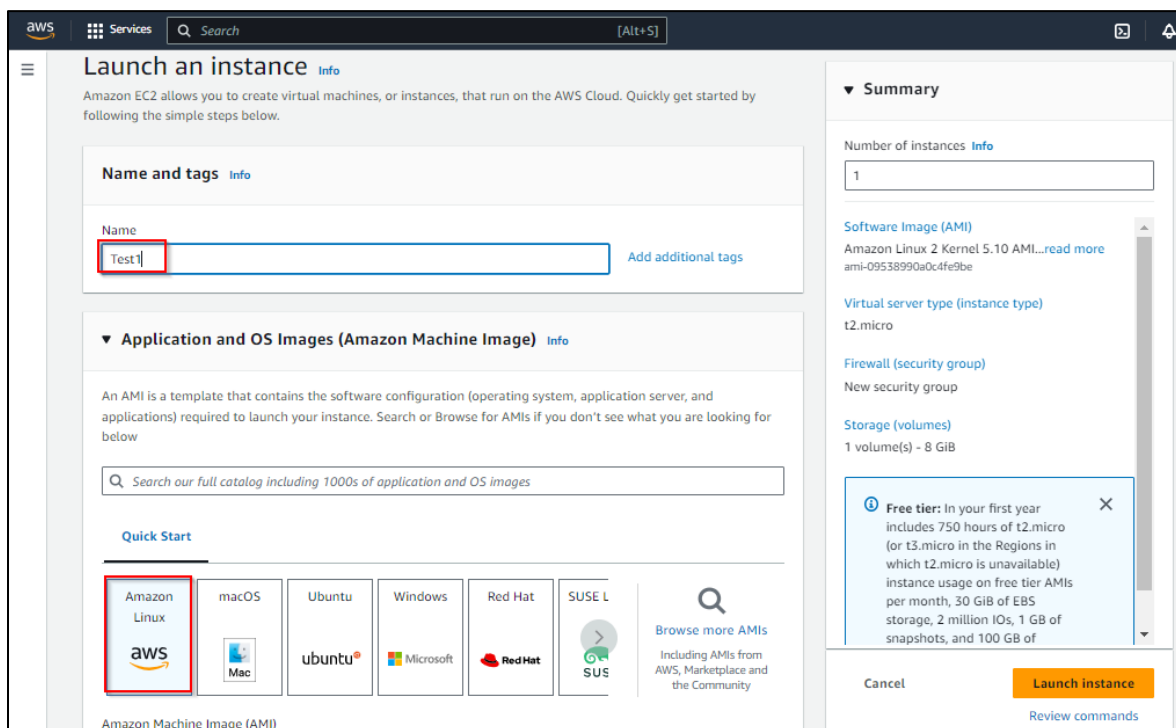
The security group has been created successfully and must be added to the EFS.

Step 3: Create AWS instances to access the EFS

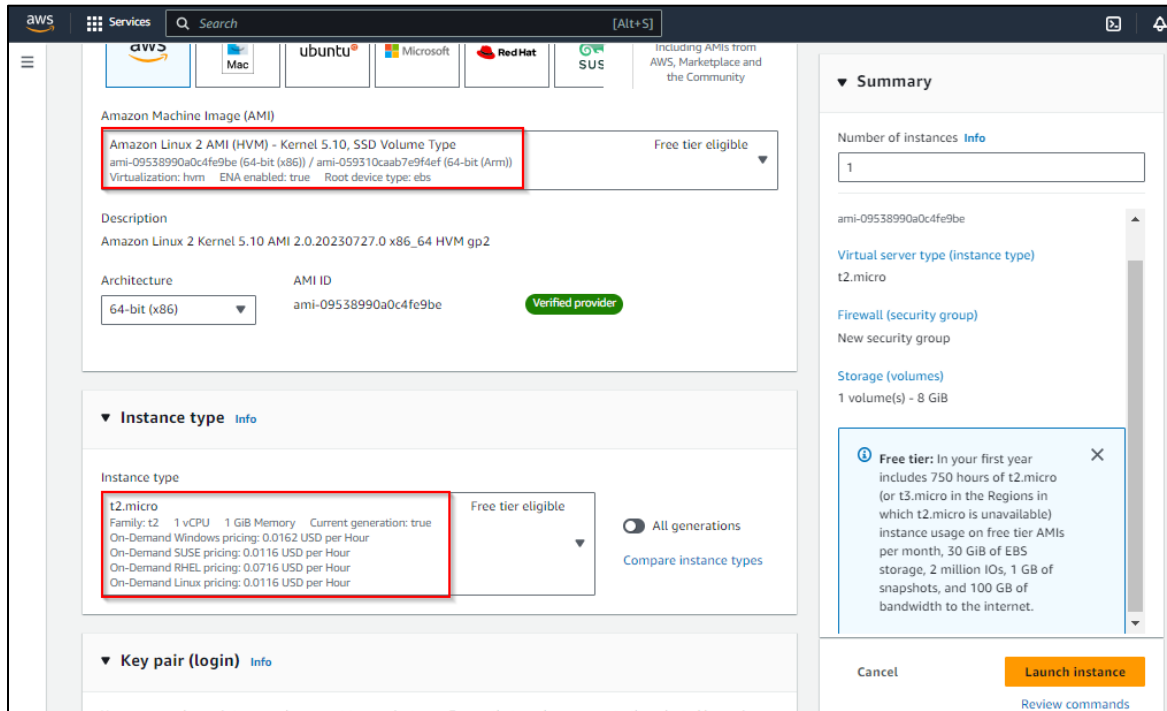
3.1 Navigate to **Instances** and click on **Launch instances**



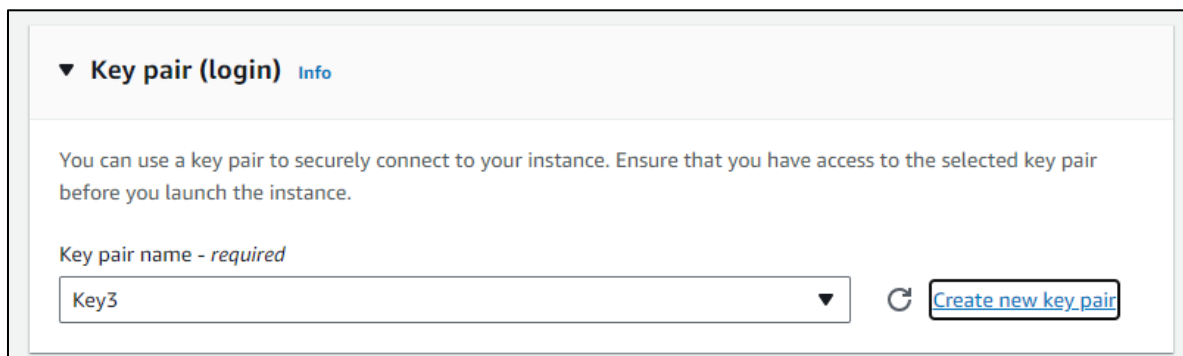
3.2 Enter the name as **Test1** and select **Amazon Linux**



3.3 Select the **Amazon Linux 2 AMI** from the **Amazon Machine Image (AMI)** and **t2.micro** from the **Instance type**



3.4 Click on **Create new key pair**



Note: Download the **pem.key** file

3.5 Under **Network settings**, change the **Subnet** to a different Availability Zone and click on **Create**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Test1	i-026717696dc8b87d3	Running	t2.micro	Initializing	No alarms	us-east-1a

The Test1 instance has been created successfully. Please wait for the initialization to complete.

Note: Repeat steps 3.1 to 3.5 to create an additional instance in a different Availability Zone

3.6 Note down the **Public IPv4** address of the current instance

The screenshot shows the AWS Management Console with the 'Instances' page. A table lists three instances: Test2 (Running), Test1 (Terminated), and Test3 (Running). Test3 is selected. Below the table, the details for 'Instance: i-0cbd62fef5eafa475 (Test3)' are shown. The 'Public IPv4 address' is highlighted as 3.91.151.118.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Test2	i-0301522fa20ea7395	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a
Test1	i-026717696dc8b87d3	Terminated	t2.micro	-	No alarms	us-east-1a
Test3	i-0cbd62fef5eafa475	Running	t2.micro	Initializing	No alarms	us-east-1b

Instance: i-0cbd62fef5eafa475 (Test3)

Public IPv4 address
3.91.151.118 | [open address](#)

Private IPv4 addresses
172.31.25.115

Public IPv4 DNS
ec2-3-91-151-118.compute-1.amazonaws.com | [open address](#)

3.7 Open the **AWS CloudShell** and use SSH to connect to the instance:

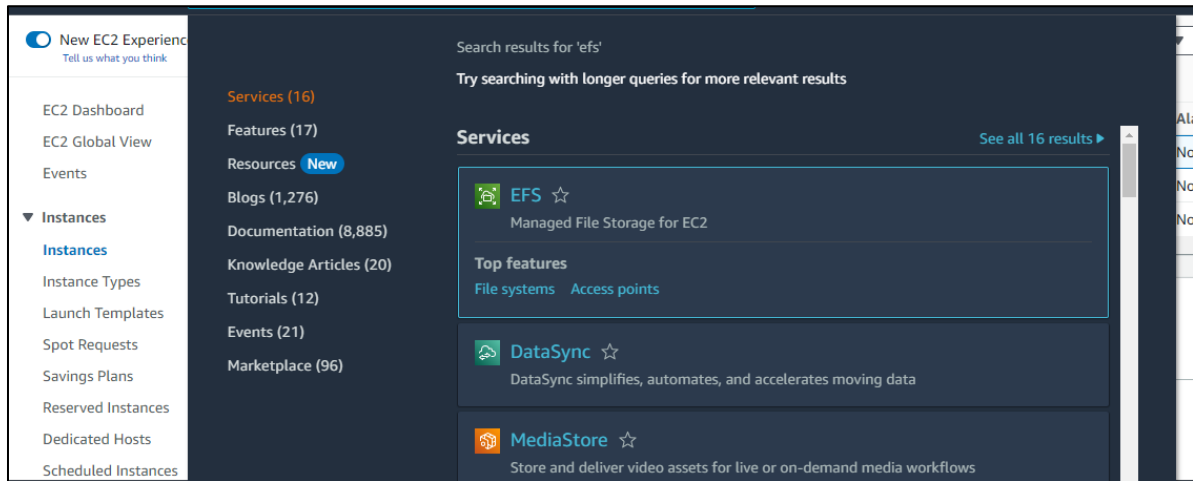
ssh -i my-ec2.pem ec2-user@3.91.151.118

The screenshot shows the AWS CloudShell interface. The command `ssh -i my-ec2.pem ec2-user@3.91.151.118` is entered and highlighted in red. The output shows a warning about the identity file and a confirmation to connect to the host 3.91.151.118.

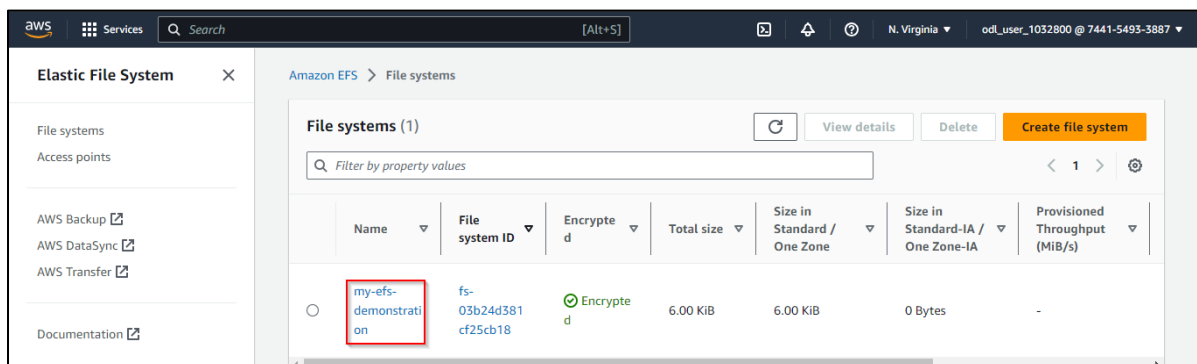
```
[cloudshell-user@ip-10-2-47-231 ~]$ ssh -i my-ec2.pem ec2-user@3.91.151.118
Warning: Identity file my-ec2.pem not accessible: No such file or directory.
The authenticity of host '3.91.151.118 (3.91.151.118)' can't be established.
ECDSA key fingerprint is SHA256:gXbiPG9x6+69iMDQzxXhcXS0QrtT7BtsGcNw4tTjy8A.
ECDSA key fingerprint is MD5:a5:c7:42:b1:f0:eb:24:8a:ce:69:49:db:e2:46:38:31.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '3.91.151.118' (ECDSA) to the list of known hosts.
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[cloudshell-user@ip-10-2-47-231 ~]$
```

Step 4: Install EFS on the created instances

4.1 Open the EFS section in the AWS Management Console



4.2 Access the Elastic File System you created in Step 1



4.3 Click on **Network**, then select **Manage**

Elastic File System ×

Availability zone: Standard

fs-0aa2bd19519fdc177.efs.us-east-1.amazonaws.com

Metered size | Monitoring | Tags | File system policy | Access points | **Network** | Replication

Network ↻ **Manage**

Availability zone	Mount target ID	Subnet ID	Mount target state	IP address	Network interface ID	Security groups
us-east-1a	fsmt-0dd69ff9b539d755c	subnet-059777e57e871fa47	Available	172.31.84.181	eni-09a88b8177712ee40	sg-0a135c0c067572f56 (my-efs-demonstration)
us-east-1b	fsmt-00305f18f5fdf31cd	subnet-0ed9a72f921dc3e7f	Available	172.31.22.199	eni-0c23a73e51734212f	sg-0a135c0c067572f56 (my-efs-demonstration)

4.4 Remove the existing mount target Availability Zones

Elastic File System ×

Amazon EFS > File systems > fs-0aa2bd19519fdc177 > Network access

Availability zone

Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 instances to connect to your file system.

vpc-004d0e086addf5c63
default

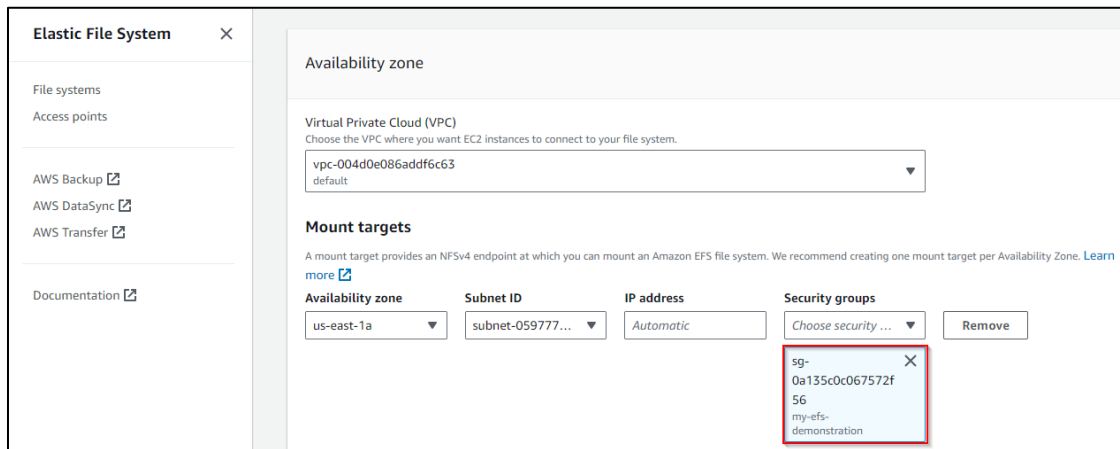
You must delete all existing mount targets in order to change the VPC of your file system.

Mount targets

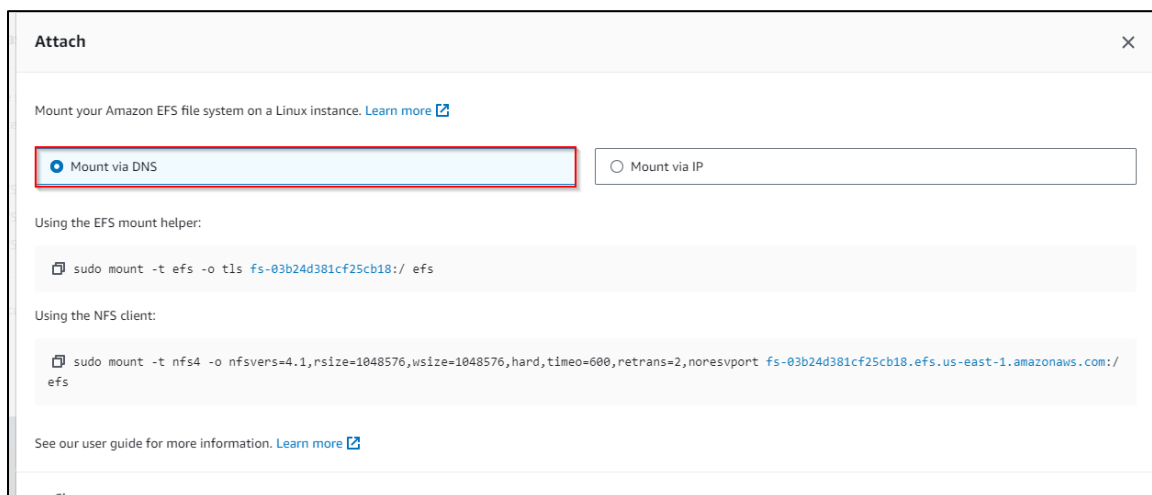
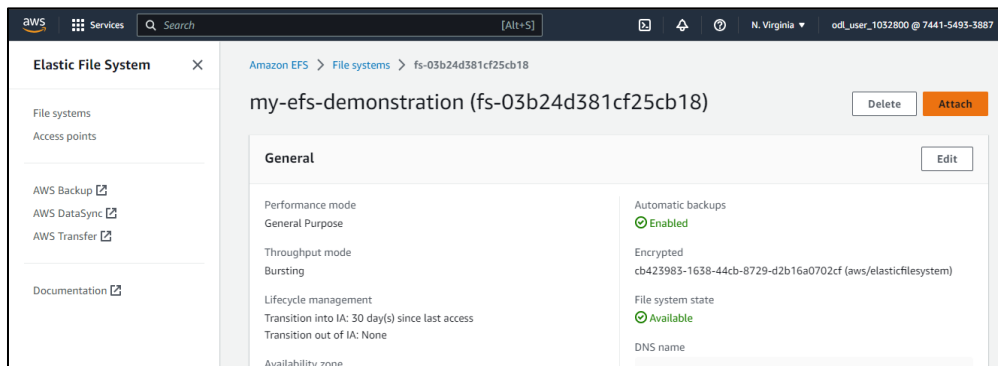
A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone. [Learn more](#)

Availability zone	Subnet ID	IP address	Security groups
us-east-1a	subnet-059777e57e87	172.31.90.77	Choose security ... sg-0f124ff719203d0

4.5 Add the Availability Zone associated with your EFS, **my-efs-demonstration**



4.6 Click **Attach** and then **Mount via DNS**



Note: Follow the guide to install the amazon-efs-utils package on both instances: <https://docs.aws.amazon.com/efs/latest/ug/installing-amazon-efs-utils.html>

4.7 Execute the command **sudo yum install -y amazon-efs-utils** in the **AWS CloudShell** of both instances

```
[ec2-user@ip-172-31-47-151 ~]$ sudo yum install -y amazon-efs-utils
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
---> Package amazon-efs-utils.noarch 0:1.27.1-1.amzn2 will be installed
--> Processing Dependency: stunnel >= 4.56 for package: amazon-efs-utils-1.27.1-1.amzn2.noarch
--> Running transaction check
---> Package stunnel.x86_64 0:4.56-6.amzn2.0.3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

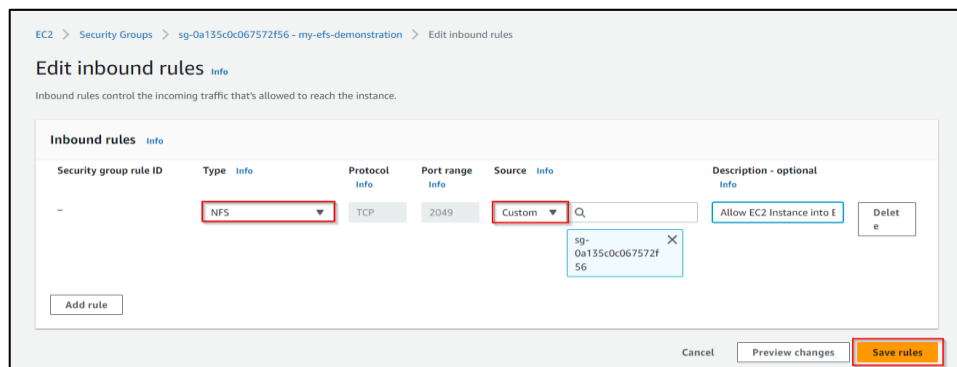
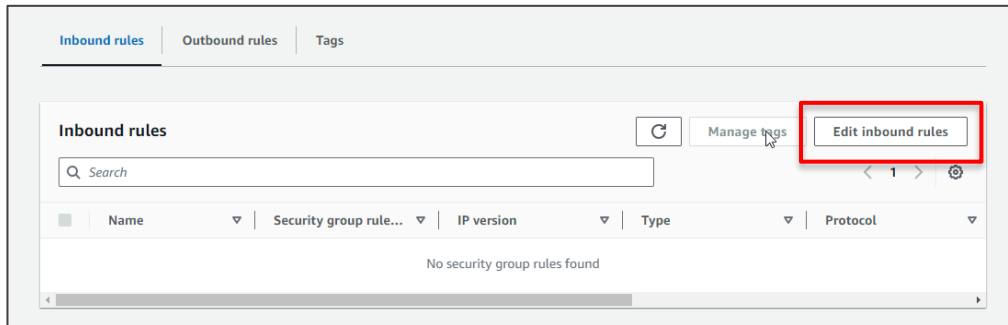
```
[ec2-user@ip-172-31-80-38 ~]$ sudo yum install -y amazon-efs-utils
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
---> Package amazon-efs-utils.noarch 0:1.27.1-1.amzn2 will be installed
--> Processing Dependency: stunnel >= 4.56 for package: amazon-efs-utils-1.27.1-1.amzn2.noarch
--> Running transaction check
---> Package stunnel.x86_64 0:4.56-6.amzn2.0.3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

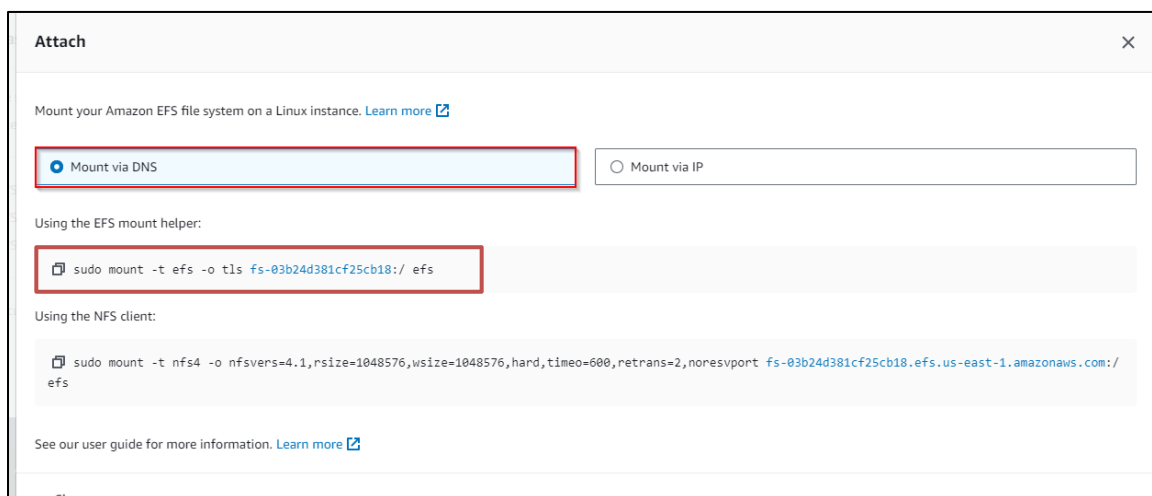
4.8 Create an **efs** directory on both instances using:
mkdir efs

```
[cloudshell-user@ip-10-136-37-33 ~]$ mkdir efs
```


4.9 Navigate to the security group and click on **Edit inbound rules** to add inbound rules for EFS



4.10 On the instances, use the EFS mount helper to mount the EFS

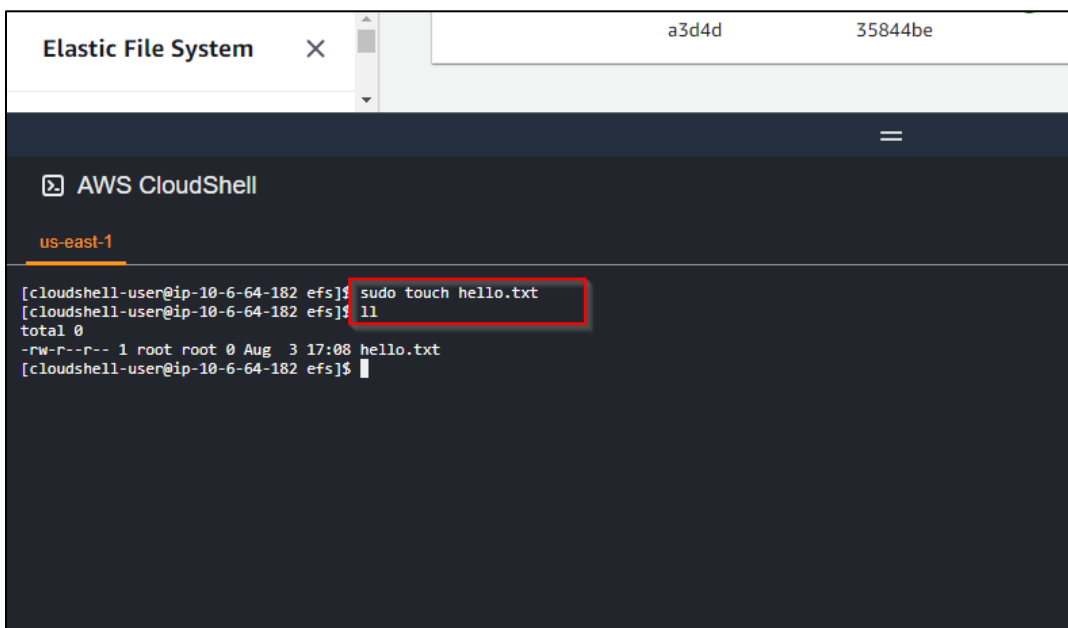


- 4.11 Log into one of the instances and create a file, such as **hello.txt**:

```
cd efs/  
sudo touch hello.txt  
ll
```

```
[ec2-user@ip-172-31-80-38 ~]$ cd efs/  
[ec2-user@ip-172-31-80-38 efs]$ sudo touch hello.txt  
[ec2-user@ip-172-31-80-38 efs]$ ll  
total 4  
-rw-r--r-- 1 root root 0 Sep 15 16:57 hello.txt  
[ec2-user@ip-172-31-80-38 efs]$
```

- 4.12 Log in to the other instance, and you will see the same file, **hello.txt**, in that instance



```
Elastic File System a3d4d 35844be  
AWS CloudShell  
us-east-1  
[cloudshell-user@ip-10-6-64-182 efs]$ sudo touch hello.txt  
[cloudshell-user@ip-10-6-64-182 efs]$ ll  
total 0  
-rw-r--r-- 1 root root 0 Aug 3 17:08 hello.txt  
[cloudshell-user@ip-10-6-64-182 efs]$
```

By following these steps, you have effectively established and mounted an Amazon Elastic File System (EFS) on multiple instances, showcasing the seamless sharing and accessibility of files across your AWS environment.