

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) on multiple AWS instances

Tools required: AWS Workspace

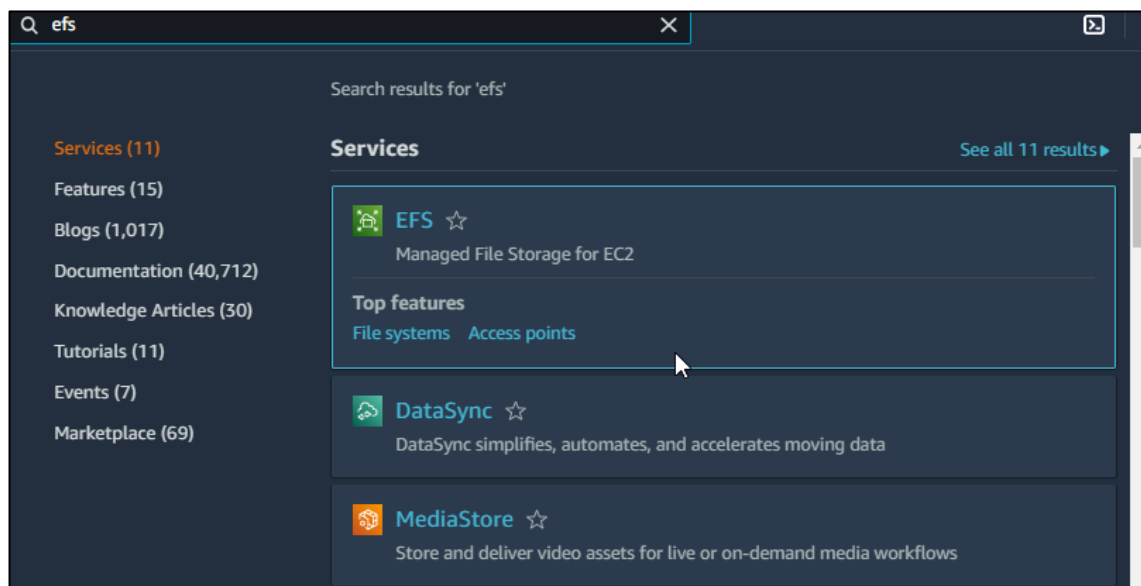
Prerequisites: AWS account with an S3 bucket created

Steps to be followed:

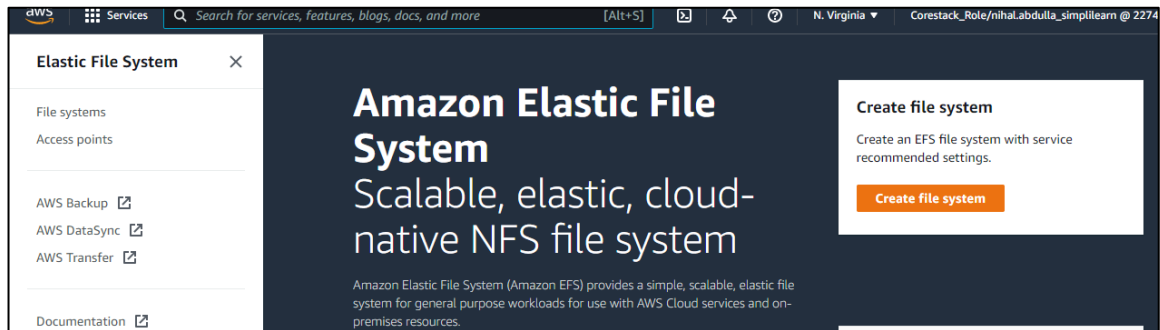
1. Create and customize an EFS
2. Create a security group to configure network access
3. Create AWS instances to access the EFS
4. Install EFS on the created instances

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**

Create file system

Create an EFS file system with service recommended settings. [Learn more](#)

Name - optional
Name your file system.

Name must not be longer than 256 characters, and must only contain letters, numbers, and these characters: + - = . _ : /

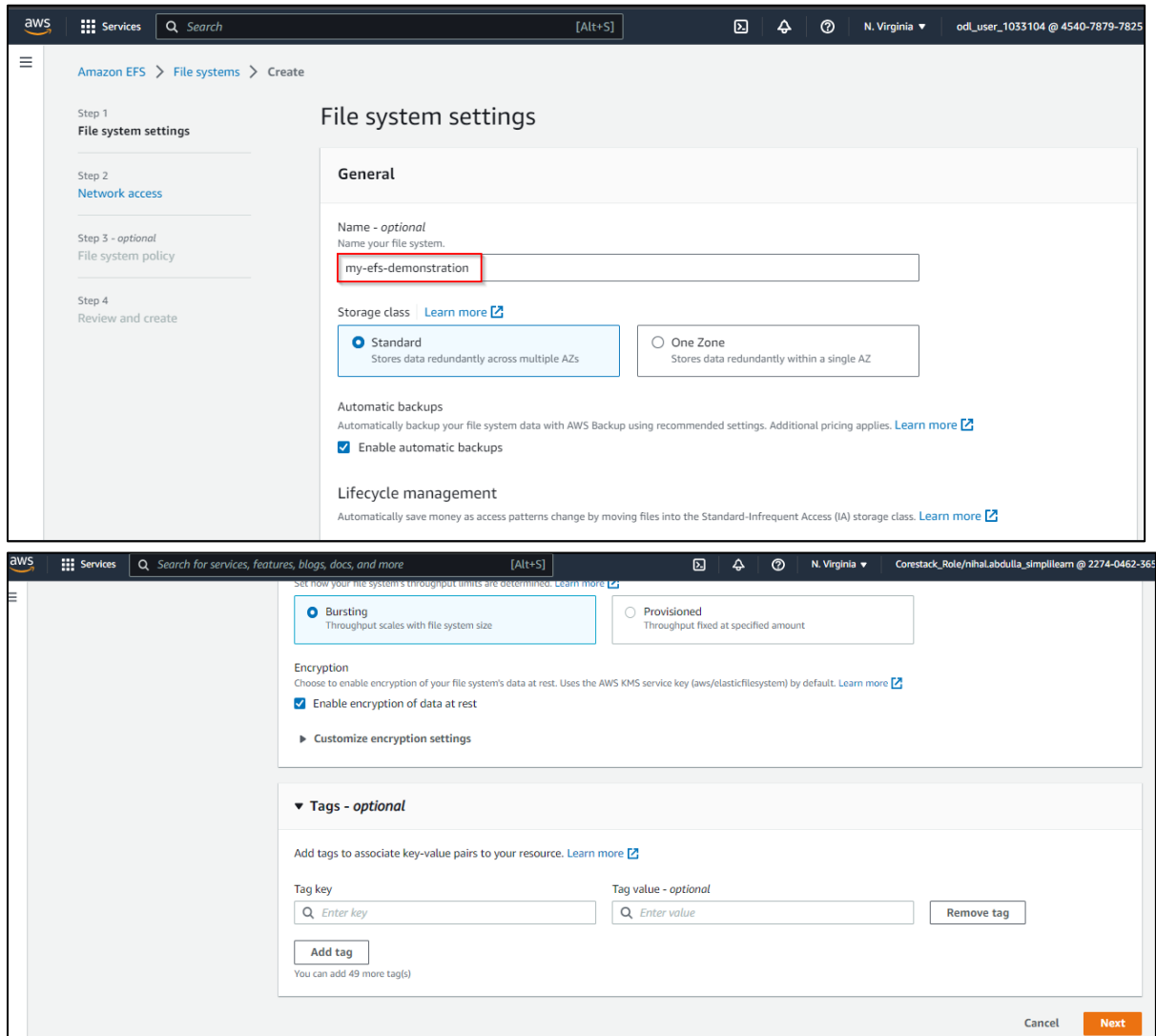
Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 instances to connect to your file system. [Learn more](#)

Cancel

Customize

Create

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



aws Services Search [Alt+S] N. Virginia odl_user_1033104 @ 4540-7879-7825

Amazon EFS > File systems > Create

Step 1
File system settings

Step 2
Network access

Step 3 - optional
File system policy

Step 4
Review and create

File system settings

General

Name - optional
Name your file system.
my-efs-demonstration

Storage class [Learn more](#)

☒ Standard
Stores data redundantly across multiple AZs

☐ One Zone
Stores data redundantly within a single AZ

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 Standard-Infrequent Access (IA) storage class. [Learn more](#)

Set how your file system's throughput limits are determined. [Learn more](#)

☒ Bursting
Throughput scales with file system size

☐ Provisioned
Throughput fixed at specified amount

Encryption
Choose to enable encryption of your file system's data at rest. Uses the AWS KMS service key (aws/elasticfilesystem) by default. [Learn more](#)

☒ Enable encryption of data at rest

► Customize encryption settings

▼ Tags - optional

Add tags to associate key-value pairs to your resource. [Learn more](#)

Tag key

Tag value - optional

You can add 49 more tag(s)

Cancel

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

The screenshot shows the AWS Management Console interface for creating an Amazon EFS file system. The left-hand navigation pane shows the 'Create' wizard with four steps: Step 1 (File system settings), Step 2 (Network access), Step 3 (optional, File system policy), and Step 4 (Review and create). Step 2, 'Network access', is highlighted with a red box. The main content area is titled 'Network access' and contains two sections: 'Network' and 'Mount targets'. In the 'Network' section, a dropdown menu shows 'vpc-01acb7a7e122e5ae3' as the selected VPC. In the 'Mount targets' section, there are four columns: 'Availability zone' (set to 'us-east-1a'), 'Subnet ID' (set to 'subnet-07637...'), 'IP address' (set to 'Automatic'), and 'Security groups'. A dropdown menu for 'Security groups' is open, showing 'sg-043af87bc607ab084' as the selected group. A 'Remove' button is visible next to the security group dropdown.

1.6 Click **Next**

The screenshot shows the AWS Management Console interface for creating an Amazon EFS file system, now at the 'File system policy' step. The left-hand navigation pane shows the 'Create' wizard with four steps: Step 1 (File system settings), Step 2 (Network access), Step 3 (optional, File system policy), and Step 4 (Review and create). Step 3, 'File system policy', is highlighted with a red box. The main content area is titled 'File system policy' and contains two sections: 'Policy options' and 'Policy editor (JSON)'. In the 'Policy options' section, there are four checkboxes: 'Prevent root access by default*', 'Enforce read-only access by default*', 'Prevent anonymous access', and 'Enforce in-transit encryption for all clients'. Below these checkboxes is a link to 'Learn more' and a section for 'Grant additional permissions'. In the 'Policy editor (JSON)' section, there is a text area for editing the policy and a 'Clear' button. At the bottom of the console, there are three buttons: 'Cancel', 'Previous', and 'Next'. The 'Next' button is highlighted with a red box.

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

Step 1: File system settings

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

Amazon EFS > File systems

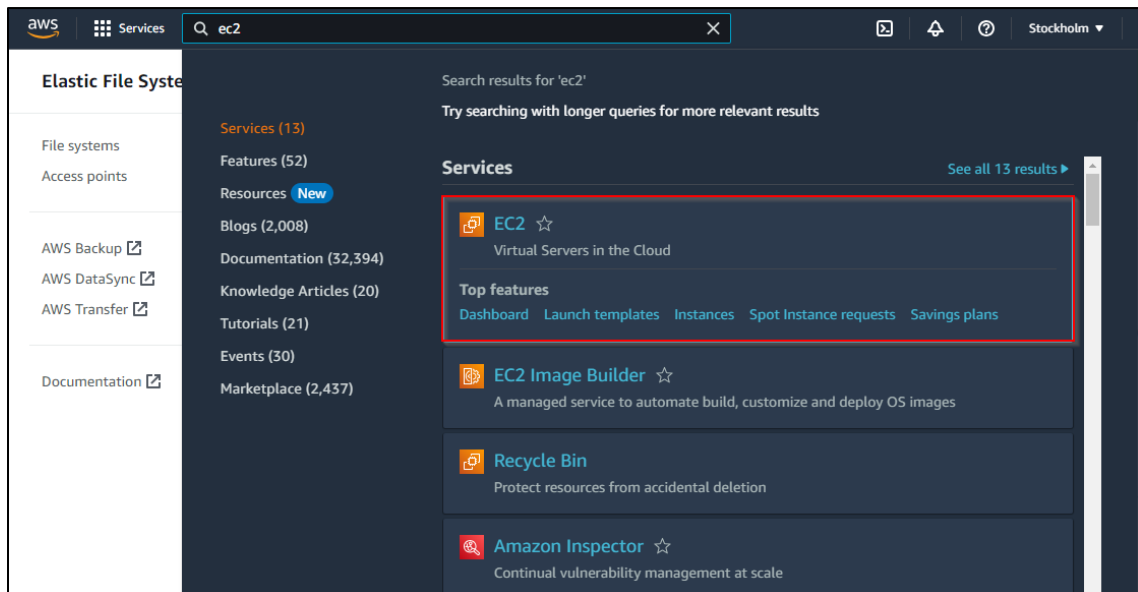
File systems (1)

Name	File system ID	Encrypted	Total size	Size in Standard / One Zone	Size in Standard-IA / One Zone-IA	Provisioned Throughput (MiB/s)
my-efs-demonstration	fs-03b24d381cf25cb18	Encrypted	6.00 KIB	6.00 KIB	0 Bytes	-

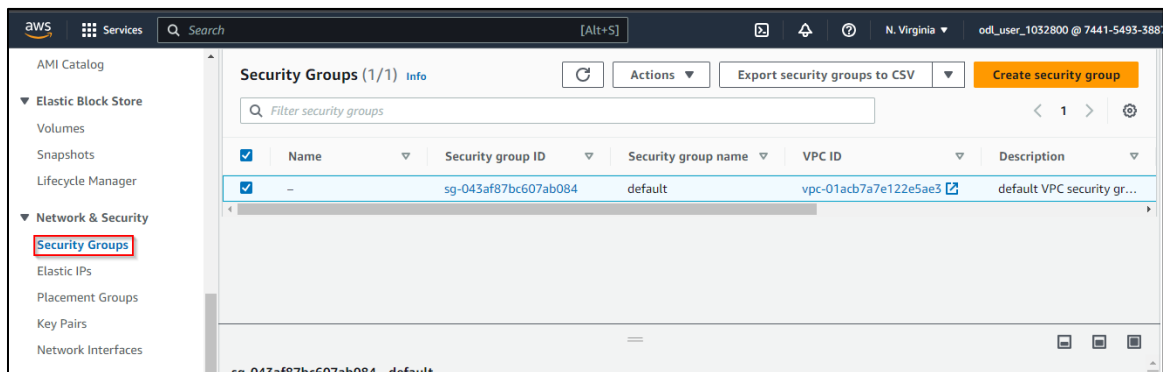
The EFS file is now successfully created.

Step 2: Create a security group to configure network access

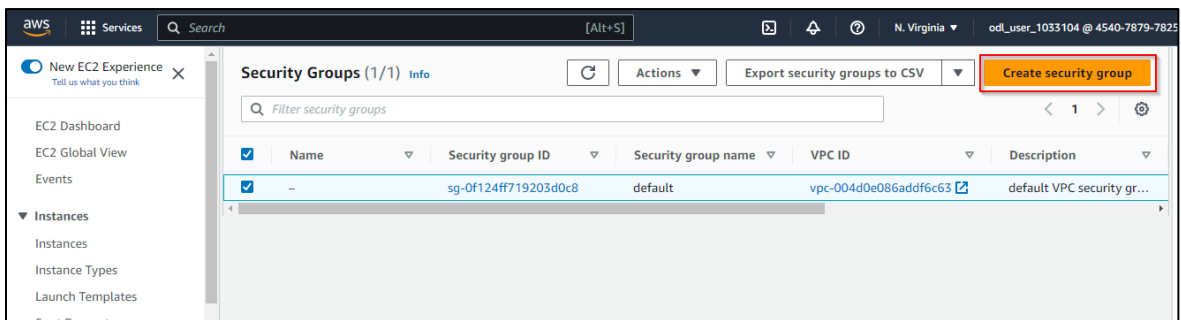
2.1 Navigate to EC2 and click on it



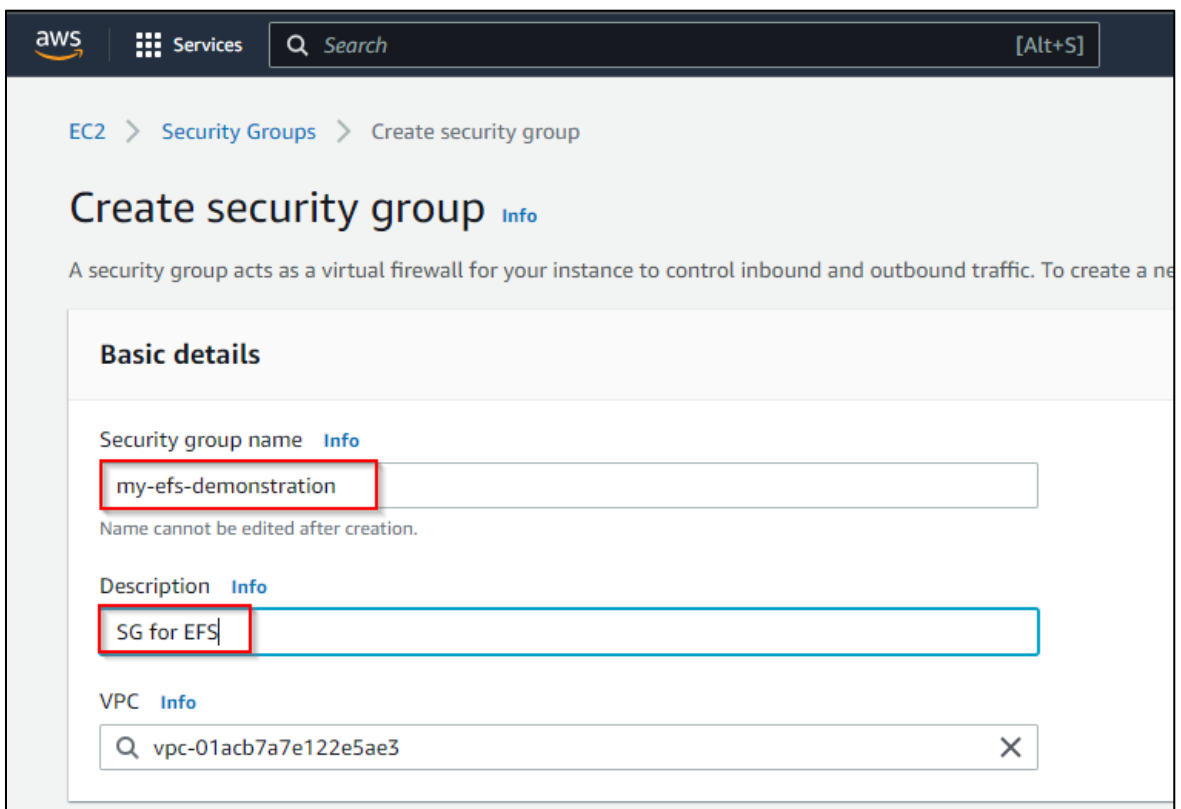
2.2 Click on Security Groups



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



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



2.5 Click on **Create security group**

Outbound rules [Info](#)

Type	Protocol	Port range	Destination	Description - optional	
All traffic	All	All	Custom 0.0.0.0/0		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

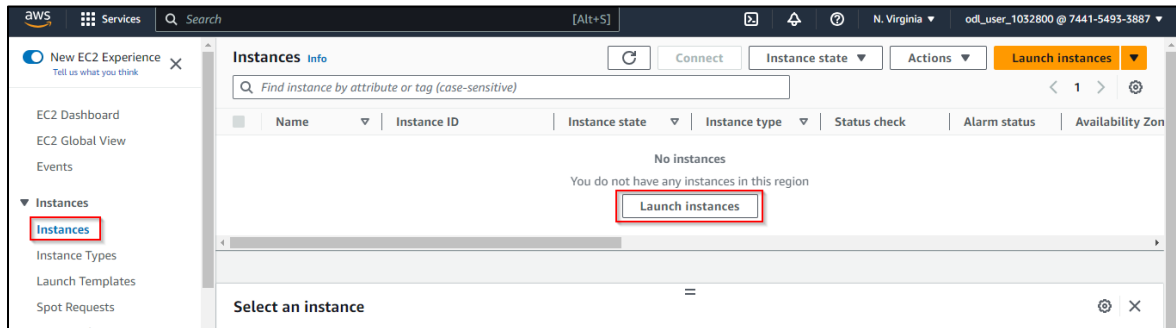
sg-0c3839c366569b724 - my-efs-demonstration [Actions](#)

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

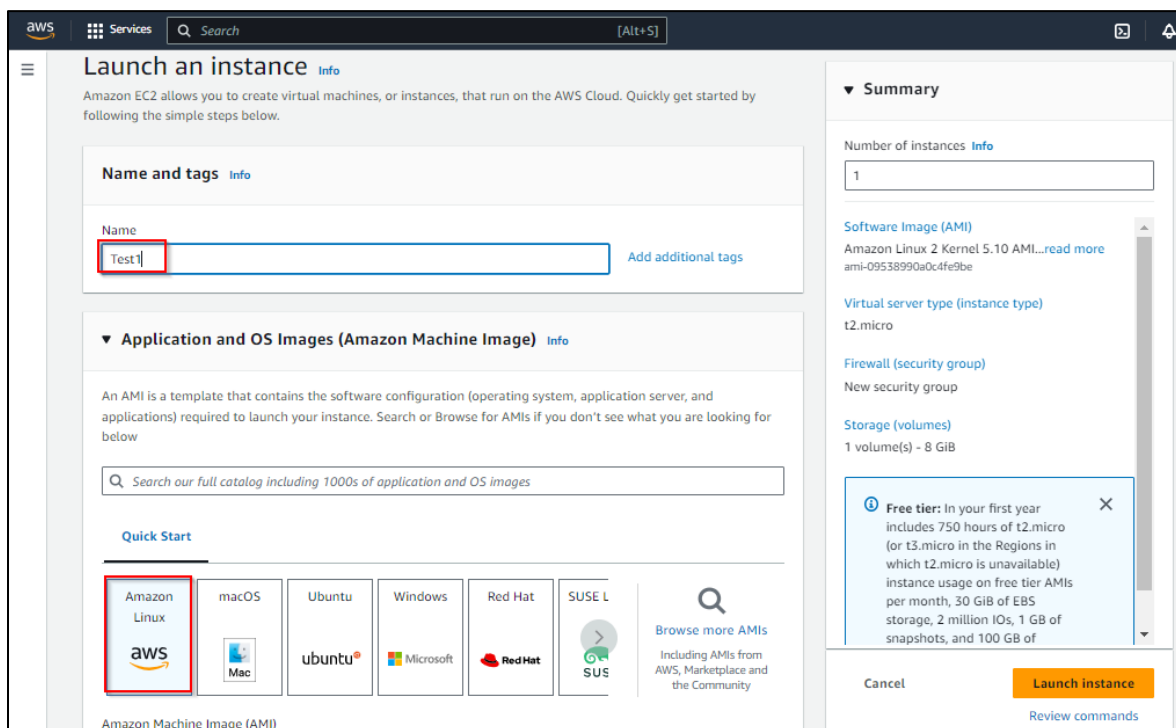
The security group has been created successfully and must be added to the EFS.

Step 3: Create AWS instances to access the EFS

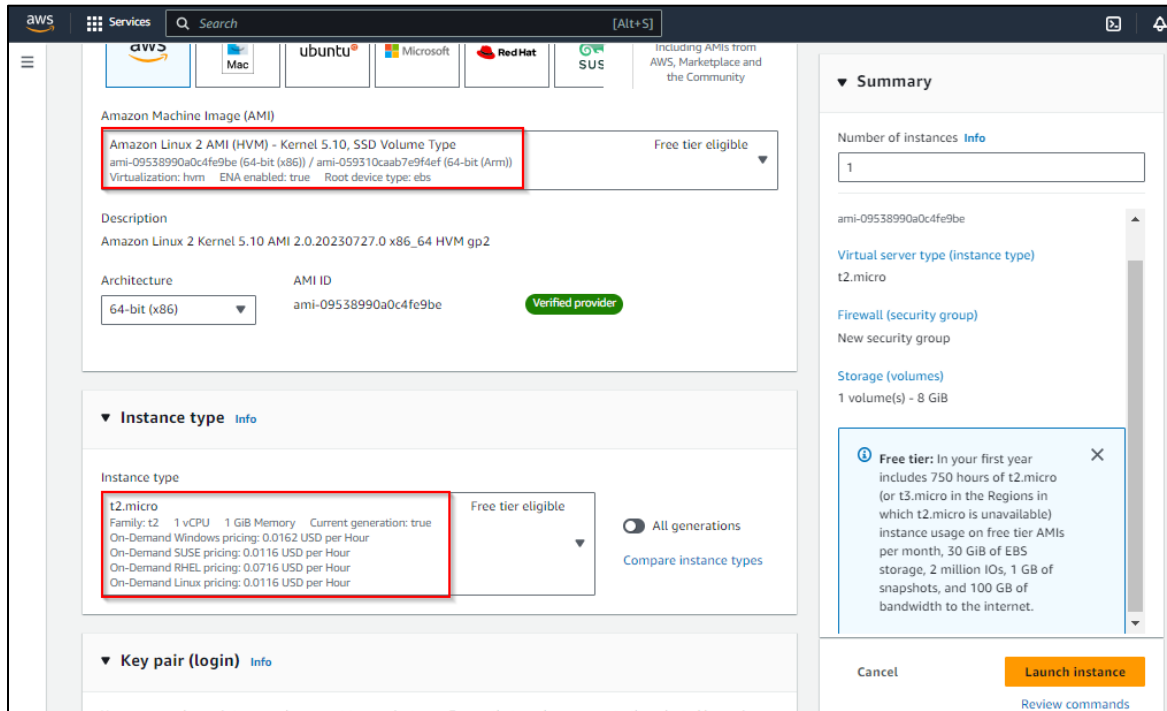
3.1 Navigate to Instances > Launch instances



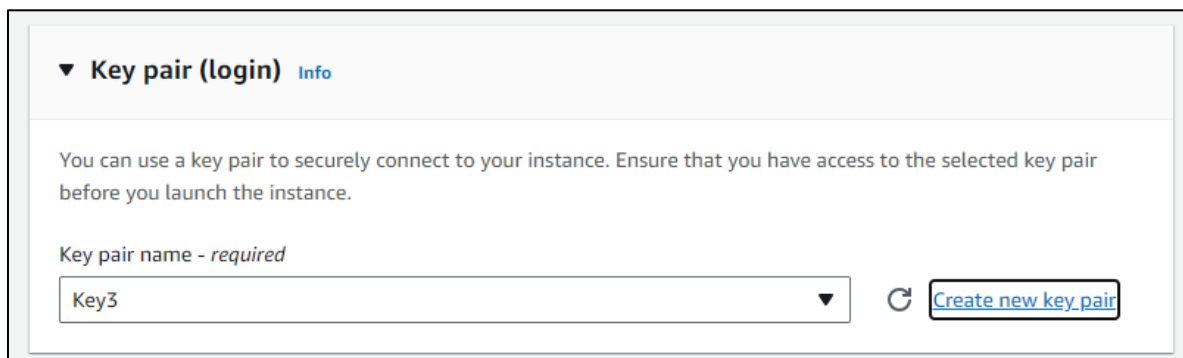
3.2 Enter the name as **Test1** and select **Amazon Linux**



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**

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-01acb7a7e122e5ae3 (default) ↕

172.31.0.0/16

Subnet [Info](#)

subnet-0aaaa33415db257b9

VPC: vpc-01acb7a7e122e5ae3 Owner: 744154933887 Availability Zone: us-east-1b

IP addresses available: 4090 CIDR: 172.31.16.0/20

↕ [Create new subnet](#)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

aws Services Search [Alt+S] N. Virginia ocl_user_1032800 @ 7441-5493-3887

New EC2 Experience Tell us what you think

EC2 Dashboard EC2 Global View Events

▼ Instances

Instances

Instance Types Launch Templates Spot Requests

Instances (1) [Info](#) ↕ Connect Instance state Actions [Launch instances](#)

Find instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Test1	i-026717696dc8b87d3	Running	t2.micro	Initializing	No alarms	us-east-1a

Select an instance

The **Test1** instance has been created successfully. Wait for initialization to be completed.

Repeat steps **3.1** to **3.5** to create another instance with 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 selected. A table lists three instances: Test2 (Running), Test1 (Terminated), and Test3 (Running). Test3 is selected, and its details are shown below. 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)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Instance ID: 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:

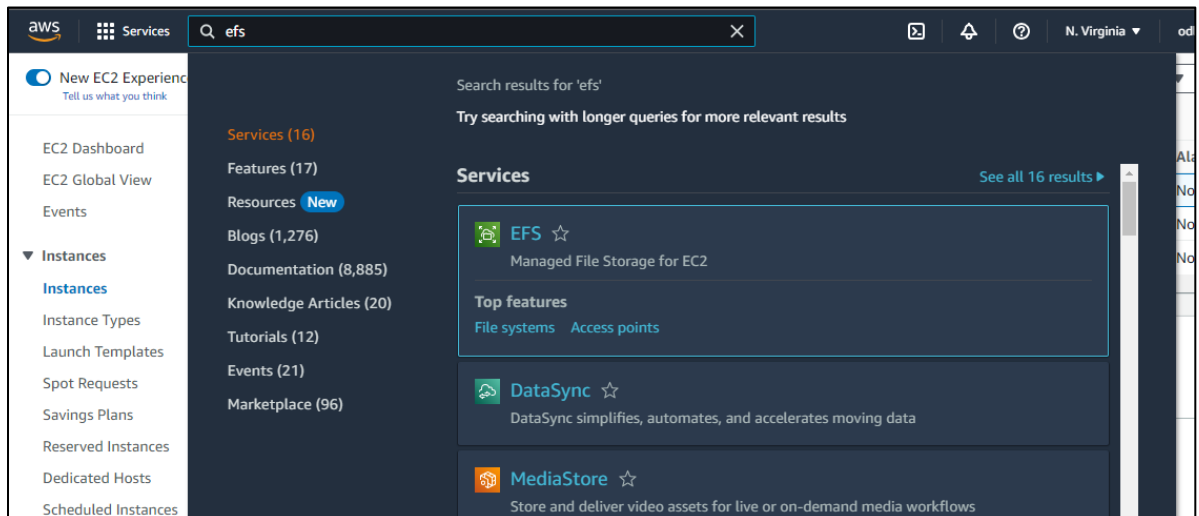
ssh -i my-ec2.pem ec2-user@3.91.151.118

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.

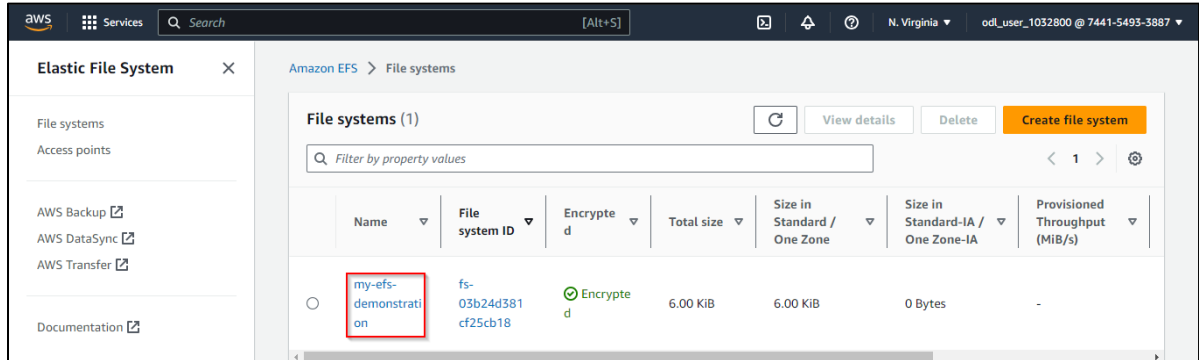
```
[cloudshell-user@ip-10-2-47-231 ~]$ ssh -i my-ec2.pem ec2-user@3.91.151.118
Warning: Identity file my-ec2.pem not accessible: No such file or directory.
The authenticity of host '3.91.151.118 (3.91.151.118)' can't be established.
ECDSA key fingerprint is SHA256:gXbiPG9x6+69iMDQzxXhcXS0QrtT7BtsGcNw4tTjy8A.
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** and then **Manage**

The screenshot shows the AWS Elastic File System console. The left sidebar has 'Elastic File System' selected. The main panel shows the 'Network' tab for the file system 'fs-0aa2bd19519fdc177'. The 'Network' tab is highlighted with a red box, and the 'Manage' button is also highlighted with a red box. Below the tabs, there is a table of mount targets.

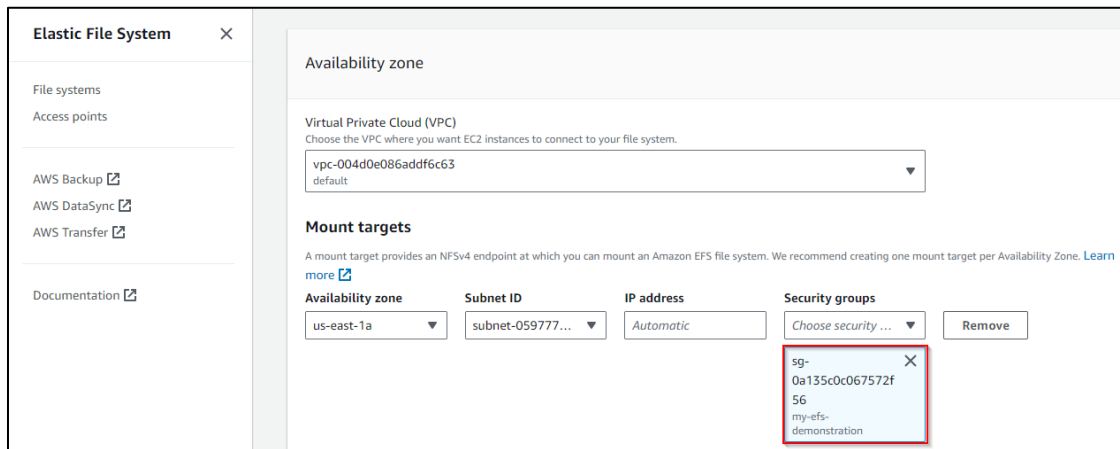
Availability zone	Mount target ID	Subnet ID	Mount target state	IP address	Network interface ID	Security groups
us-east-1a	fsmt-0dd69ff9b539d755c	subnet-059777e57e871fa47	Available	172.31.84.181	eni-09a88b8177712ee40	sg-0a135c0c067572f56 (my-efs-demonstration)
us-east-1b	fsmt-00305f18f5fdf31cd	subnet-0ed9a72f921dc3e7f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

4.4 Remove existing mount targets' availability zones

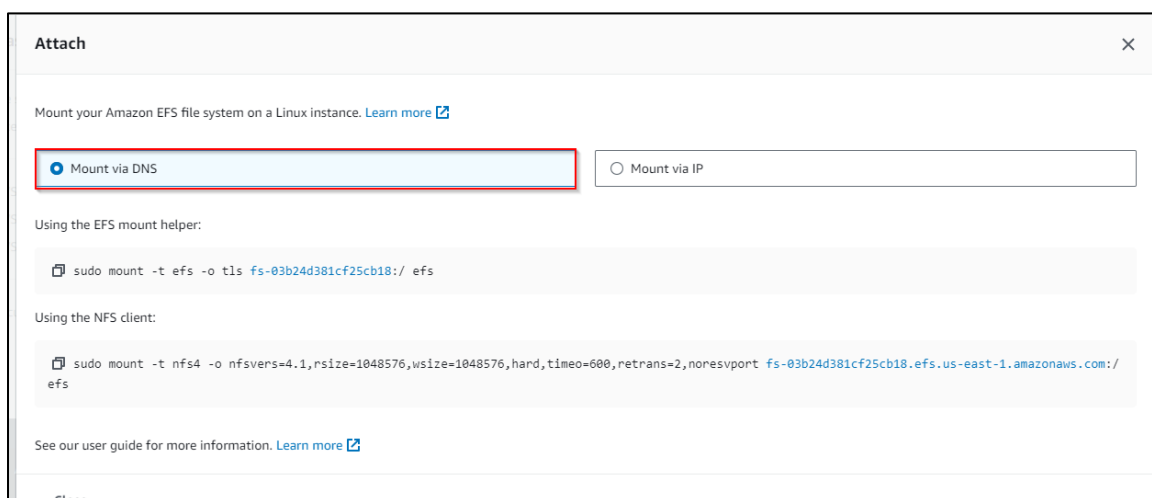
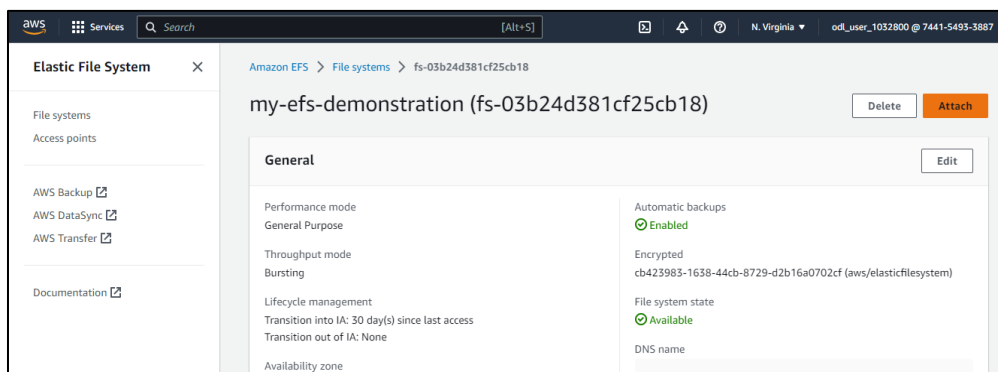
The screenshot shows the AWS Elastic File System console, specifically the 'Mount targets' section. The 'Mount targets' header is highlighted with a red box. Below the header, there is a table of mount targets.

Availability zone	Subnet ID	IP address	Security groups
us-east-1a	subnet-059777e57e87	172.31.90.77	Choose security ... sg-0f124f719203d0

4.5 Add the availability zone of your EFS, my-efs-demonstration



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



4.7 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>

4.8 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
```

4.9 Create an **efs** directory on both instances using:

mkdir efs

4.10 Navigate to the security group and add inbound rules for EFS

EC2 > Security Groups > sg-0a135c0c067572f56 - my-efs-demonstration > Edit inbound rules

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
-	NFS	TCP	2049	Custom	Allow EC2 Instance into EFS

[Add rule](#)

[Cancel](#) [Preview changes](#) [Save rules](#)

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

Attach

×

Mount your Amazon EFS file system on a Linux instance. [Learn more](#)

☒ Mount via DNS

☐ Mount via IP

Using the EFS mount helper:

`sudo mount -t efs -o tls fs-03b24d381cf25cb18:/ efs`

Using the NFS client:

`sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-03b24d381cf25cb18.efs.us-east-1.amazonaws.com:/ efs`

See our user guide for more information. [Learn more](#)

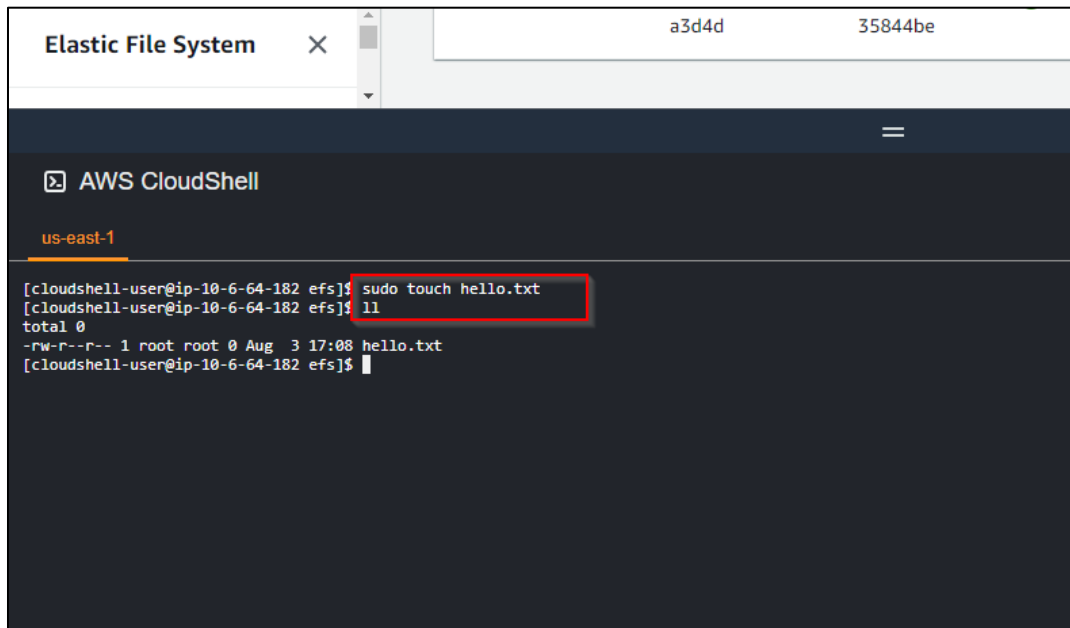
Close

4.12 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]$
```

4.13 Log in to the other instance, and you will be able to see the same file, **hello.txt**, in that instance as well.



The screenshot shows the AWS CloudShell interface. At the top, there's a header for 'Elastic File System' with a close button and two identifiers: 'a3d4d' and '35844be'. Below this, the 'AWS CloudShell' logo is visible, followed by the region 'us-east-1'. The terminal window shows the following commands and output:

```
[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]$
```

The command `sudo touch hello.txt` and its output `ll` are highlighted with a red box.

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.