

## Exercise: Hands on for EFS

### 1) Create a security Group

### Create Security Group

Security group name ⓘ

MyEFSDemoSG

Description ⓘ

Security Group for EFS Demd

VPC ⓘ

vpc-e026569a (default) ▾

Security group rules:

Inbound

Outbound

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
This security group has no rules				

Add Rule

Cancel


Create

### 2) Create EFS

aws

Services ▾ Resource Groups ▾ ★

admin @ 1787-1683-3342 ▾ N. Virginia ▾ Support ▾




## Amazon Elastic File System (EFS)

Amazon EFS provides file storage for use with your EC2 instances.


Create file system

[Getting started guide](#)




### Create

Create an Amazon EFS file system to store your files in the Amazon cloud. A file system grows and shrinks automatically with the files you put in, and you only pay for what you use.



### Access

Write files to and read files from your Amazon EFS file system by using the NFSv4 protocol. Any number of EC2 instances can work with your file system at the same time, and your instances can be in multiple Availability Zones in a region.



### Manage

You can easily administer your file system using the Amazon EFS console, CLI, and SDK.

### Step 1: Configure file system access

Step 2: Configure optional settings

Step 3: Review and create

### Configure file system access

An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system by using a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.

VPC vpc-e026569a (default) ⓘ

### Create mount targets

Instances connect to a file system by using mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.

	Availability Zone	Subnet ⓘ	IP address ⓘ	Security groups ⓘ
<input checked="" type="checkbox"/>	us-east-1a	subnet-98e9cab6 (default) ▾	Automatic ⓘ	sg-085e9d54 - default ✕
<input checked="" type="checkbox"/>	us-east-1b	subnet-2633c76b (default) ▾	Automatic ⓘ	sg-085e9d54 - default ✕
<input checked="" type="checkbox"/>	us-east-1c	subnet-cee8c992 (default) ▾	Automatic ⓘ	sg-085e9d54 - default ✕
<input type="checkbox"/>	us-east-1d			

## Remove the Default Security group and give the security group created for EFS

p 3: Review and create

### Add tags

You can add tags to describe your file system. A tag consists of a case-sensitive key-value pair. (For example, you can define a tag with key-value pair with key = Corporate Department and value = Sales and Marketing.) At a minimum, we recommend a tag with key = Name.

Key	Value	Remove
Name	MyEFSDemo	✕
Add New Key		

### Enable lifecycle management NEW

**Automatically save up to 85%** on your EFS bill as your access patterns change by enabling **Lifecycle Management** for your file system. Based on the policy you choose, any files in your file system that are not accessed for a period of time will automatically move to the EFS Infrequent Access (EFS IA) storage class. EFS IA provides price/performance that's cost-optimized for files not accessed every day. [Learn more](#)

Lifecycle policy 14 days since last access ▾

### Choose throughput mode

We recommend **Bursting** throughput mode for most file systems. Use **Provisioned** throughput mode for applications that require more throughput than allowed by **Bursting** throughput. [Learn more](#)

- ☒ Bursting  
☐ Provisioned

### Choose performance mode

We recommend **General Purpose** performance mode for most file systems. **Max I/O** performance mode is optimized for applications where tens, hundreds, or thousands of EC2 instances are accessing the file system — it scales to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for file operations.

- ☒ General Purpose  
☐ Max I/O

### Enable encryption

If you enable encryption for your file system, all data on your file system will be encrypted at rest. You can select a KMS key from your account to protect your file system, or you can provide the ARN of a key from a different account. Encryption of data at rest can only be enabled during file system creation. Encryption of data in transit is configured when mounting your file system. [Learn more](#)

- ☐ Enable encryption of data at rest

Cancel Previous **Next Step**

## Performance Mode:

--General Purpose – This option is used when only few EC2 instance are connected with EFS

--Max i/O—This option is used when more EC2 instance are connected with EFS

## Review and Create File System:

## Create file system

Step 1: Configure file system access

Step 2: Configure optional settings

Step 3: Review and create

### Review and create

Review the configuration below before proceeding to create your file system.

#### File system access

VPC	Availability Zone	Subnet	IP address	Security groups
vpc-e026569a (default)	us-east-1a	subnet-98e9cab6 (default)	Automatic	sg-039e1380eb8f2cc2c - MyEFSDemoSG
	us-east-1b	subnet-2633c76b (default)	Automatic	sg-039e1380eb8f2cc2c - MyEFSDemoSG
	us-east-1c	subnet-cee8c992 (default)	Automatic	sg-039e1380eb8f2cc2c - MyEFSDemoSG
	us-east-1d	Not configured		
	us-east-1e	Not configured		
	us-east-1f	Not configured		

#### Optional settings

Tags Name: MyEFSDemo

Performance mode General Purpose

Throughput mode Bursting

Encrypted No

Lifecycle policy 14 days since last access

Cancel Previous Create File System

### File systems

AWS DataSync

AWS Backup

### File systems

Success! You have created a file system. You can mount your file system from an EC2 instance with an NFSv4.1 client installed. You can also mount your file system from an on-premises server over an AWS Direct Connect or AWS VPN connection. Click here for EC2 mount instructions, and here for on-premises mount instructions.

Create file system Actions

Name	File system ID	Mounted size	Number of mount targets	Creation date
MyEFSDemo	fs-e9e04db	6.0 KB	3	08/18/2019, 06:39:49 UTC

Other details

Owner ID 178716833342

File system state Available

Performance mode General Purpose

Throughput mode Bursting

Encrypted No

Lifecycle policy 14 days since last access

Tags Name: MyEFSDemo

Manage tags

File system access

DNS name fs-e9e04db.efs.us-east-1.amazonaws.com

Amazon EC2 mount instructions (from local VPC)

Amazon EC2 mount instructions (across VPC peering connection)

On-premises mount instructions

Mount targets

VPC	Availability Zone	Subnet	IP address	Mount target ID	Network interface ID	Security groups	Mount target state
vpc-e026569a (default)	us-east-1a	subnet-98e9cab6 (default)	172.31.84.143	fsmt-a06ab040	eni-099693aed1941efce		Creating
	us-east-1b	subnet-2633c76b (default)	172.31.24.57	fsmt-dffab03f	eni-00f980e6799fa8f09		Creating
	us-east-1c	subnet-cee8c992 (default)	172.31.47.123	fsmt-a16ab041	eni-0ed277871863db4a		Creating

## 2) Create an EC2 instance in one of the availability zone which is provided in EFS

aws Services Resource Groups

admin @ 1787-1683-3342 N. Virginia Support

EC2 Dashboard Events Tags Reports Limits INSTANCES Instances Launch Templates Spot Requests

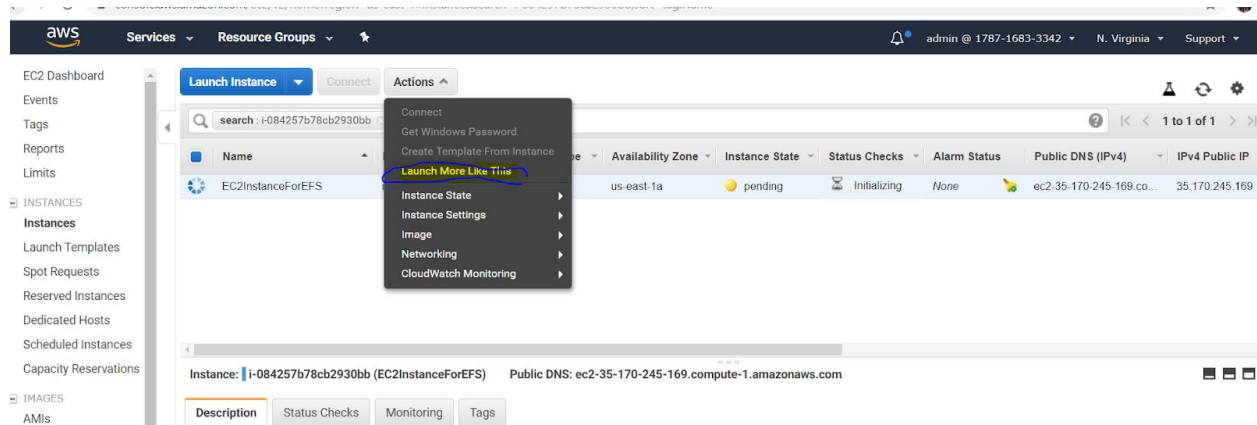
Launch Instance Connect Actions

search: i-084257b78cb2930bb Add filter

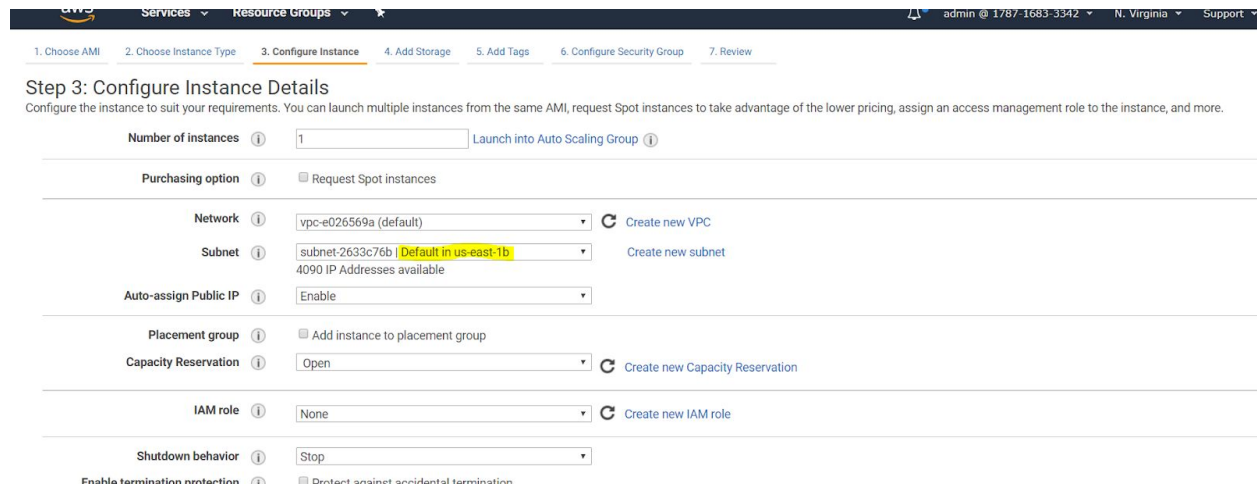
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
EC2InstanceForEFS	i-084257b78cb2930bb	t2.micro	us-east-1a	pending	Initializing	None	ec2-35-170-245-169 co...	35 170 245 169

3) Create another Ec2 instance.

Use option “Launch more like this” option

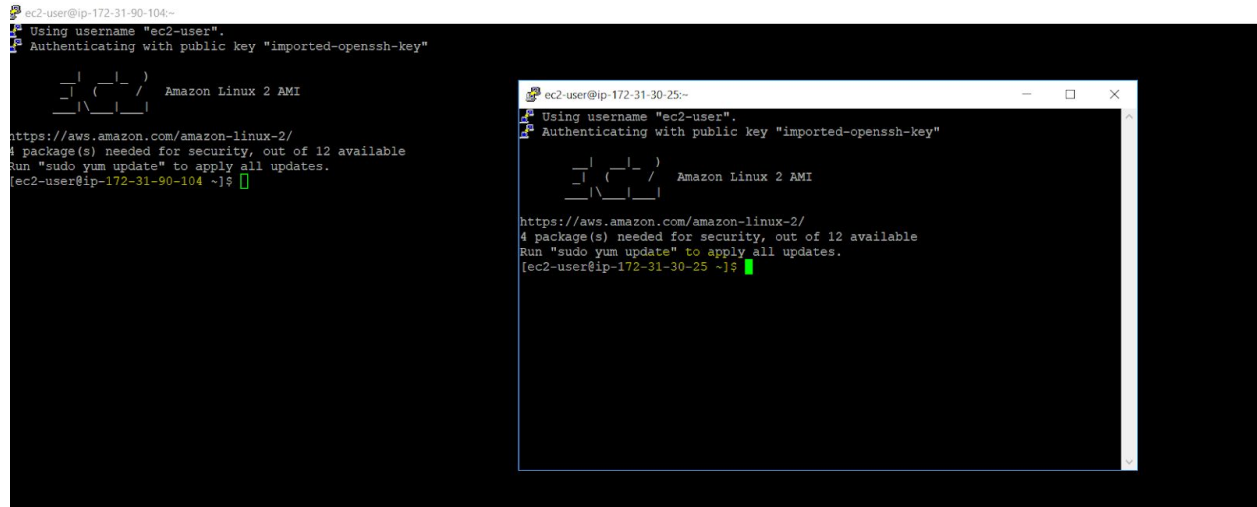


Change the availability zone to another zone which is listed in EFS



Now two instance created in two different zone which is provided in EFS

4) Connect two different session in putty for each of two instance to mount EFS

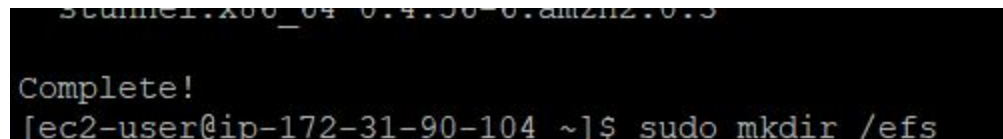


## 5) Install NFS in both instances and Mount EFS

```
sudo yum install -y amazon-efs-utils
```

Create a new directory on your EC2 instance, such as "efs".

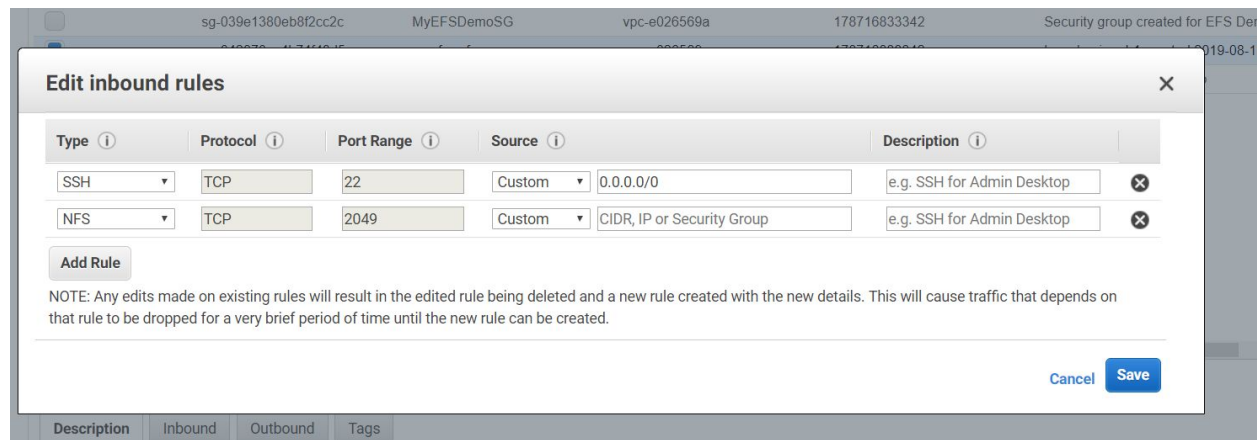
```
sudo mkdir efs
```



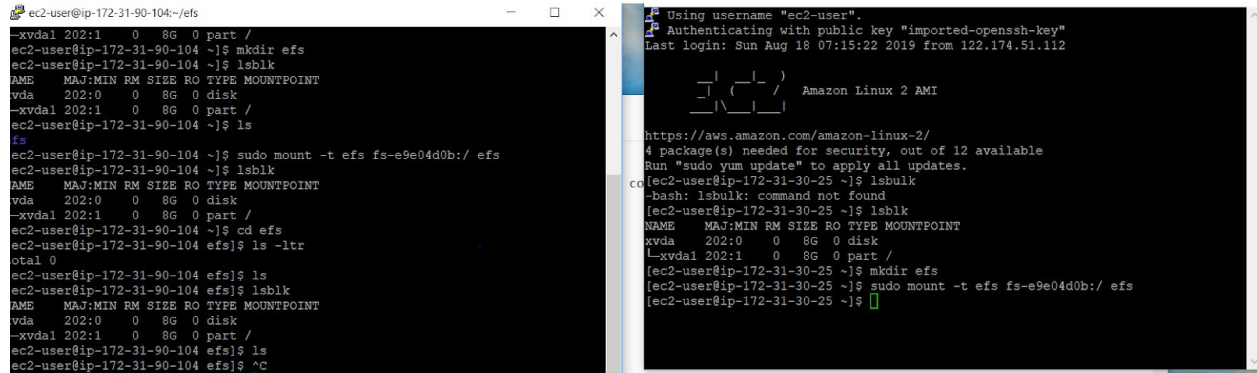
Mount your file system with a method listed following.

```
sudo mount -t efs fs-e9e04d0b:/ efs
```

This will not work as the security group for NFS was not created. Go to security group and add NFS



Mount the efs in both the instance



The image shows two terminal windows side-by-side. The left window is a terminal for an EC2 instance with IP 172-31-90-104, where the user 'ec2-user' is running commands to create an EFS directory, mount it, and verify the disk layout. The right window is a terminal for an EC2 instance with IP 172-31-30-25, showing the login process and the user 'ec2-user' running commands to update packages, create an EFS directory, and mount it.

```
ec2-user@ip-172-31-90-104:~/efs
-xvda1 202:1 0 8G 0 part /
ec2-user@ip-172-31-90-104 ~/$ mkdir efs
ec2-user@ip-172-31-90-104 ~/$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 202:0 0 8G 0 disk
-xvda1 202:1 0 8G 0 part /
ec2-user@ip-172-31-90-104 ~/$ ls
fs
ec2-user@ip-172-31-90-104 ~/$ sudo mount -t efs fs-e9e04d0b:/ efs
ec2-user@ip-172-31-90-104 ~/$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 202:0 0 8G 0 disk
-xvda1 202:1 0 8G 0 part /
ec2-user@ip-172-31-90-104 ~/$ cd efs
ec2-user@ip-172-31-90-104 efs/$ ls -ltr
total 0
ec2-user@ip-172-31-90-104 efs/$ ls
ec2-user@ip-172-31-90-104 efs/$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 202:0 0 8G 0 disk
-xvda1 202:1 0 8G 0 part /
ec2-user@ip-172-31-90-104 efs/$ ls
ec2-user@ip-172-31-90-104 efs/$ ^C

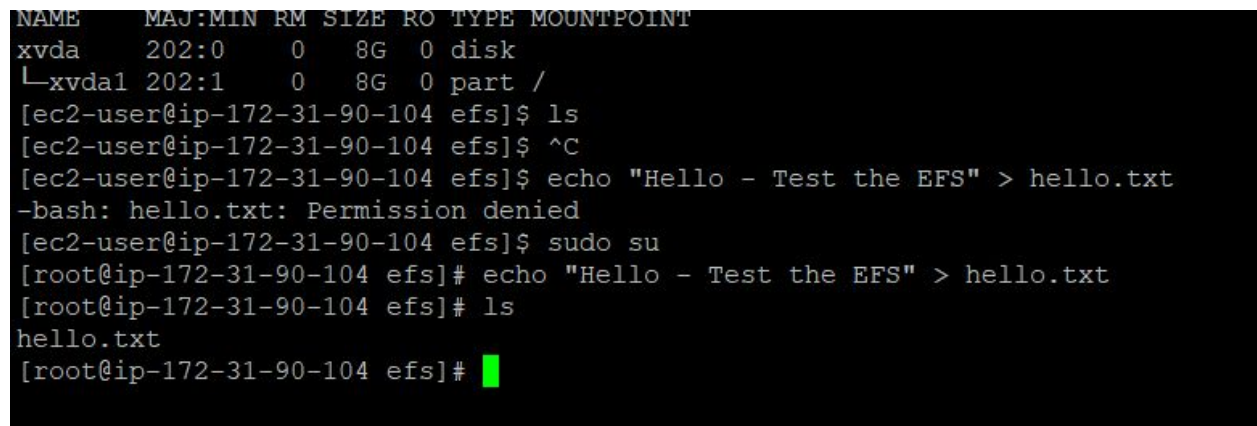
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Sun Aug 18 07:15:22 2019 from 122.174.51.112

_ _ _ _ _
| | | | |
_ _ _ _ _ Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 12 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-30-25 ~]$ lsblk
-bash: lsblk: command not found
[ec2-user@ip-172-31-30-25 ~]$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 202:0 0 8G 0 disk
-xvda1 202:1 0 8G 0 part /
[ec2-user@ip-172-31-30-25 ~]$ mkdir efs
[ec2-user@ip-172-31-30-25 ~]$ sudo mount -t efs fs-e9e04d0b:/ efs
[ec2-user@ip-172-31-30-25 ~]$
```

## 6) Test the file system

Create the file in one instance. The same file should be seen in another instance as well



The image shows a terminal window for an EC2 instance with IP 172-31-90-104. The user 'ec2-user' runs 'ls' to see the EFS directory, then 'echo "Hello - Test the EFS" > hello.txt' which fails with 'Permission denied'. Then, they run 'sudo su' to become root and successfully create the file 'hello.txt'. Finally, they run 'ls' to confirm the file's presence.

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 202:0 0 8G 0 disk
-xvda1 202:1 0 8G 0 part /
[ec2-user@ip-172-31-90-104 efs]$ ls
[ec2-user@ip-172-31-90-104 efs]$ ^C
[ec2-user@ip-172-31-90-104 efs]$ echo "Hello - Test the EFS" > hello.txt
-bash: hello.txt: Permission denied
[ec2-user@ip-172-31-90-104 efs]$ sudo su
[root@ip-172-31-90-104 efs]# echo "Hello - Test the EFS" > hello.txt
[root@ip-172-31-90-104 efs]# ls
hello.txt
[root@ip-172-31-90-104 efs]#
```

This file is visible in another instance where EFS is mounted

```

Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Sun Aug 18 07:15:22 2019 from 122.174.51.112

    _ | _ | _ )
    _ | ( _ | _ /   Amazon Linux 2 AMI
    __| \__| __|

https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 12 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-30-25 ~]$ lsbulk
-bash: lsbulk: command not found
[ec2-user@ip-172-31-30-25 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda         202:0    0   8G  0 disk
└─xvda1      202:1    0   8G  0 part /
[ec2-user@ip-172-31-30-25 ~]$ mkdir efs
[ec2-user@ip-172-31-30-25 ~]$ sudo mount -t efs fs-e9e04d0b:/ efs
[ec2-user@ip-172-31-30-25 ~]$ ls
efs
[ec2-user@ip-172-31-30-25 ~]$ ls -ltr
total 4
drwxr-xr-x 2 root root 6144 Aug 18 11:37 efs
[ec2-user@ip-172-31-30-25 ~]$ cd efs
[ec2-user@ip-172-31-30-25 efs]$ ls
hello.txt
[ec2-user@ip-172-31-30-25 efs]$ █

```

By Selvin