

Lab manual for EFS

Objective:

To create a EFS and mount it on an EC2 instance as a Network file share

Steps:

1. Setup the Pre-Requisite
2. Create the EFS mount point
3. Mount to the EC2 instance and check
4. Create EFS from AWS CLI

Step1:

Pre-Requisite for the EFS Lab Manual

On the VPC, make sure **“DNS hostname”** and **“DNS Resolution”** is enabled.

[VPCs](#) > Edit DNS resolution

Edit DNS resolution

VPC ID vpc-08ed4f70

DNS resolution ☒ enable

* Required

Enable -- DNS Resolution

[VPCs](#) > Edit DNS hostnames

Edit DNS hostnames

VPC ID vpc-08ed4f70

DNS hostnames ☒ enable

* Required

Enable -- DNS Hostnames

Create a separate security Group for the EFS

Create security group Actions ▾

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name ▾	Group ID ▴	Group Name ▾	VPC ID ▾	Type	Description ▾	Owner
<input checked="" type="checkbox"/>		sg-00b7aa1cfb3a2...	EFS-Access	vpc-08ed4f70	EC2-VPC	EFSAccess	558457644648

Security Group: sg-00b7aa1cfb3a271c9

Description Inbound Rules Outbound Rules Tags

Edit rules

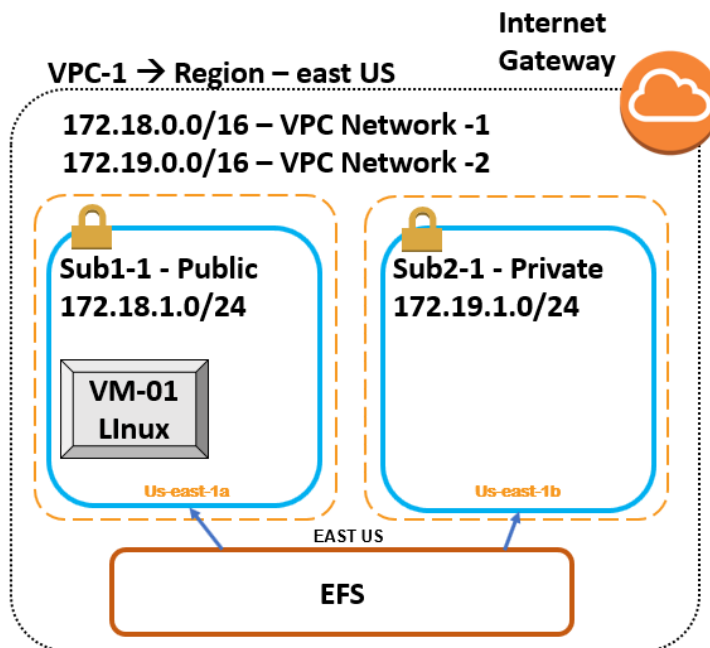
Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
NFS	TCP	2049	172.31.0.0/16	-

VPC CIDR

As my VPC CIDR is “172.31.0.0/16”

Step2: Creating Mount Targets

After you create a file system, you can create mount targets and then you can mount the file system on EC2 instances in your VPC, as shown in the following illustration.



The mount target security group acts as a virtual firewall that controls the traffic. For example, it determines which Amazon EC2 instances can access the file system. This section explains the following:

- Mount target security groups and how to enable traffic.
- How to mount the file system on your Amazon EC2 instance.
- NFS-level permissions considerations.
- Initially, only the root user on the Amazon EC2 instance has read-write-execute permissions on the file system. This topic discusses NFS-level permissions and provides examples that show you how to grant permissions in common scenarios. For more information, see [Network File System \(NFS\)–Level Users, Groups, and Permissions](#).

You can create mount targets for a file system using the console, using AWS Command Line Interface, or programmatically using the AWS SDKs. When using the console, you can create mount targets when you first create a file system or after the file system is created.

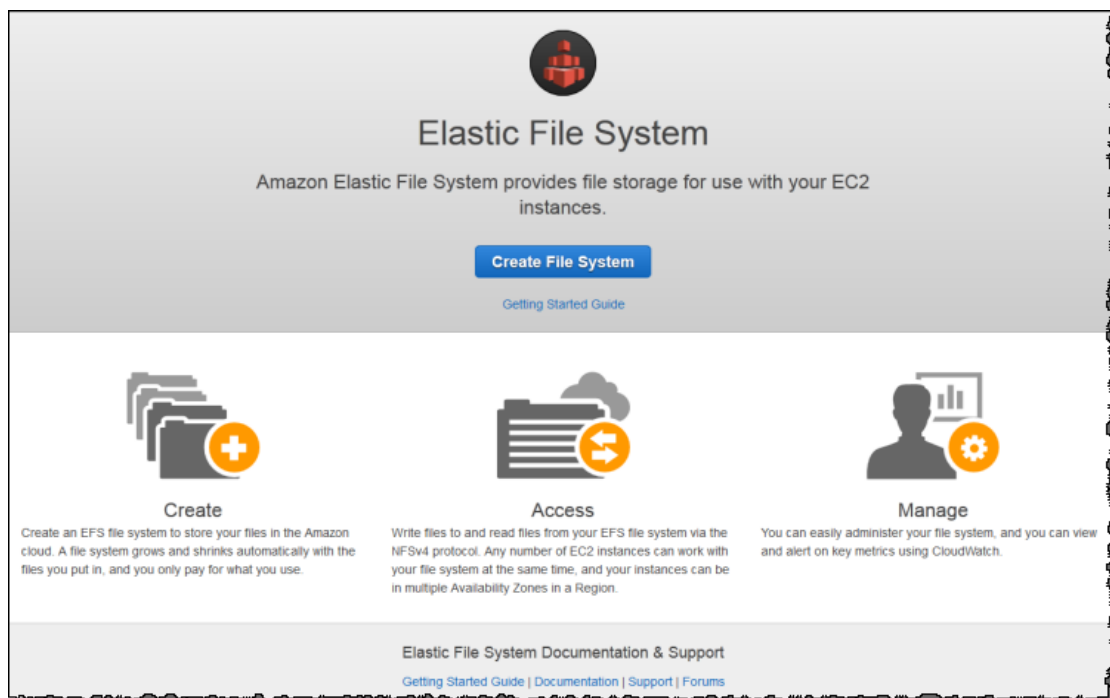
Creating a Mount Target Using the Amazon EFS console

Perform the steps in the following procedure to create a mount target using the console. As you follow the console steps, you can also create one or more mount targets for EFS .

You can create one mount target for each Availability Zone in your VPC.

To create an Amazon EFS file system (console)

1. Sign in to the AWS Management Console and open the Amazon EFS console at <https://console.aws.amazon.com/efs/>.
2. Choose **Create File System**.



Note:-

The console shows the preceding page only if you don't already have any Amazon EFS file systems. If you have created file systems, the console shows a list of your file systems.

On the list page, choose **Create File System**.

3. On the **Step 1: Configure File System Access** page, select the VPC and the Availability Zone in the VPC.

This VPC should be the same Amazon VPC in which you created your Amazon EC2 instance in the preceding section.

- a. Select a Amazon VPC from the **VPC** list.

Warning:--

If the Amazon VPC you want is not listed, verify the region in the global navigation in the Amazon EFS console.

- b. In the **Create Mount Targets** section, select all of the Availability Zones listed.
- c. We recommend that you create mount targets in all Availability Zones. You can then mount your file system on Amazon EC2 instances created in any of the Amazon VPC subnets.

Note: -- You can access a file system on an Amazon EC2 instance in one Availability Zone by using a mount target created in another Availability Zone, but there are costs associated with cross–Availability Zone access.

- d. For each Availability Zone, do the following:
 - Choose a **Subnet** from the list where you want to create the mount target.

- You can create one mount target in each Availability Zone. If you have multiple subnets in an Availability Zone where you launched your Amazon EC2 instance, you don't have to create mount target in the same subnet, it can be any subnet in the Availability Zone.
- Leave **IP Address** select to **Automatic**. Amazon EFS will select one of the available IP addresses for the mount target.
- Specify the **Security Group** you created specifically for the mount target, or the default security group for the default VPC. Both security groups will have the necessary inbound rule that allows inbound access from the EC2 instance security group.
- Click in the **Security Group** box and the console will show you the available security groups. Here you can select a specific security group (**EFS-Access in our case**) and remove the **Default** security group, or leave the default in place, depending on how you configured your Amazon EC2 instance.

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 ⓘ

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	<input type="text" value="subnet-0b8044d04915fd918 (default)"/>	Automatic	<input type="text" value="sg-0e443b1af068e6e20 - EFS-Access *"/>
<input checked="" type="checkbox"/>	us-east-1b	<input type="text" value="subnet-04f51f4d2fe464bc2 (default)"/>	Automatic	<input type="text" value="sg-0e443b1af068e6e20 - EFS-Access *"/>
<input checked="" type="checkbox"/>	us-east-1c	<input type="text" value="subnet-068f945b1fd13595a (default)"/>	Automatic	<input type="text" value="sg-0e443b1af068e6e20 - EFS-Access *"/>
<input checked="" type="checkbox"/>	us-east-1d	<input type="text" value="subnet-0f3b774d3d05f9a3b (default)"/>	Automatic	<input type="text" value="sg-0e443b1af068e6e20 - EFS-Access *"/>
<input type="checkbox"/>	us-east-1e			

Click → Next

- On the **Step 2: Configure optional settings** page, specify a value for the **Name** tag (**MyExampleFileSystem**) and choose your performance mode.
The console prepopulates the **Name** tag because Amazon EFS uses its value as the file system display name.

Configure optional settings

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 Sales and Marketing.) At a minimum, we recommend a tag with key = Name.

Key	Value
<input type="text" value="Name"/>	<input type="text" value="Add New Value"/>
<input type="text" value="Add New Key"/>	<input type="text"/>

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 s that are not accessed for a period of time will automatically move to the EFS Infrequent Access (EFS IA) storage class. EFS IA prov every day. [Learn more](#)

Lifecycle policy

We can also select the required **“Lifecycle”**.

Means, here we have selected **“14 days”** and AWS will move the files in the EFS to an EFS IA (infrequent access) system.

This will reduce the cost incurred for the EFS storage.

We do have few more options as below.

Choose throughput mode

We recommend **Bursting** throughput mode for most file systems. Use **Provisioned** throughput mode for applications that require [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 accessing the file system — it scales to higher levels of aggregate throughput and operations per second with a tradeoff of slightly

- ☒ **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 or 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 always enabled.

- ☐ **Enable encryption of data at rest**

Throughput Mode →

Bursting → will provide best effort basis of bandwidth between EFS and EC2 instance.

Provisioned → This is charged per MB/month for the dedicated bandwidth from EFS to EC2.

Choose throughput mode

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

- ☐ **Bursting**
- ☒ **Provisioned**

Throughput (MiB/s)

Throughput bill can be up to \$60.00/month.

Valid range is 1-1024 MiB/s

Encryption →

We can use the inbuild “AES” or our customer encryption.

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 or a key from a different account. Encryption of data at rest can only be enabled during file system creation. Encryption of data in transit

☒ **Enable encryption of data at rest**

☒ **Select KMS master key**

aws/elasticfilesystem

Mykey1

arn:aws:kms:us-east-1:123456789012:key/6f318a15-8c79-4825-ab54-ed94d05c6ddb

aws/elasticfilesystem

Default key for EFS file systems when no other key is defined

☐ **Enter a KMS key ARN from another account**

This means, complete DATA protection at rest.

5. On the **Step 3: Review and Create** page, choose **Create File System**.

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-0b17f9a14de147cc1 (default)	us-east-1a	subnet-0b8044d04915fd918 (default)	Automatic	sg-0e443b1af068e6e20 - EFS-Access
	us-east-1b	subnet-04f51f4d2fe464bc2 (default)	Automatic	sg-0e443b1af068e6e20 - EFS-Access
	us-east-1c	subnet-068f945b1fd13595a (default)	Automatic	sg-0e443b1af068e6e20 - EFS-Access
	us-east-1d	subnet-0f3b774d3d05f9a3b (default)	Automatic	sg-0e443b1af068e6e20 - EFS-Access
	us-east-1e	Not configured		
	us-east-1f	Not configured		

Optional settings

Tags	No tags added
Performance mode	General Purpose
Throughput mode	Provisioned (10 MiB/s)
Encrypted	Yes
KMS key ARN	arn:aws:kms:us-east-1:573625096654:key/6f318a15-8c79-4825-ab54-ed94d05c6ddb
KMS key alias	aws/elasticfilesystem
Lifecycle policy	14 days since last access

[Cancel](#)
[Previous](#)
[Create File System](#)

The console shows the newly created file system on the **File Systems** page. Verify that all mount targets show the **Life Cycle State** as **Available**. It might take a few moments before the mount targets become available (you can expand/collapse the file system in the EFS console to force it to refresh).

Under **File system access**, you'll see the file system's **DNS name**. Make a note of this DNS name. In the next section, you use the DNS name to mount the file system on the Amazon EC2 instance through the mount target. The Amazon EC2 instance on which you mount the file system can resolve the file system's DNS name to the mount target's IP address.

Now you are ready to mount the Amazon EFS file system on an Amazon EC2 instance.

Output:

Name

File system ID

Metered size

Number of mount targets

Creation date

fs-49a962ab

6.0 KiB

4

08/09/2019, 04:49:02 UTC

Other details

Tags

Owner ID

573625096654

File system state

Available

Performance mode

General Purpose

Throughput mode

Provisioned (10 MIB/s)

Encrypted

Yes

KMS key alias

aws/elasticfilesystem

KMS key ARN

arn:aws:kms:us-east-1:573625096654:key/6f318a15-8c79-4825-ab54-ed94d05c6ddb

Lifecycle policy

14 days since last access

No tags added

DNS name

fs-49a962ab.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-0b17f9a14de147cc1 (default)	us-east-1a	subnet-0b8044d04915fd918 (default)	172.31.16.61	fsmt-6f81578f	eni-0cba59625fd68363b	sg-0e443b1af068e6e20 - EFS-Access	Available
	us-east-1b	subnet-04f51f4d2fe464bc2 (default)	172.31.43.189	fsmt-70815790	eni-0663fa435ab432c7b	sg-0e443b1af068e6e20 - EFS-Access	Available
	us-east-1c	subnet-068f945b1fd13595a (default)	172.31.14.115	fsmt-71815791	eni-0ff7bdbed67fdb14c	sg-0e443b1af068e6e20 - EFS-Access	Available
	us-east-1d	subnet-0f3b774d3d05f9a3b (default)	172.31.89.217	fsmt-74815794	eni-0230c526424ff6f6e	sg-0e443b1af068e6e20 - EFS-Access	Available

The EFS is part of 4 Subnets in our use case.

Wait until the Mount target state == Available.

Step3: Mounting the EFS on the EC2 instance.

1. Login to the EC2 instance.
2. Check with the DNS name of the NFS on the machine by 'ping' to the EFS DNS name

```
[ec2-user@ip-172-31-21-3 ~]$ ping fs-49a962ab.efs.us-east-1.amazonaws.com
PING fs-49a962ab.efs.us-east-1.amazonaws.com (172.31.16.61) 56(84) bytes of data.
^C
--- fs-49a962ab.efs.us-east-1.amazonaws.com ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2033ms

[ec2-user@ip-172-31-21-3 ~]$
```

If you observe, the name is resolving to the EFS private IP , that is part of the Subnet1.

Here the EC2 instance is in subnet1

i-0ee6759c7433b74fd	t2.micro	us-east-1a	running	2/2 checks ...	None
Public DNS: ec2-107-23-155-232.compute-1.amazonaws.com					
Elastic Checks Monitoring Tags					
Instance ID	i-0ee6759c7433b74fd	Public DNS (IPv4)	ec2-107-23-155-232.compute-1.amazonaws.com		
Instance state	running	IPv4 Public IP	107.23.155.232		
Instance type	t2.micro	IPv6 IPs	-		
Elastic IPs		Private DNS	ip-172-31-21-3.ec2.internal		
Availability zone	us-east-1a	Private IPs	172.31.21.3		
Security groups	SSH-HTTP, view inbound rules, view outbound rules	Secondary private IPs			
Scheduled events	No scheduled events	VPC ID	vpc-0b17f9a14de147cc1		
AMI ID	amzn2-ami-hvm-2.0.20190618-x86_64-gp2 (ami-0b898040803850657)	Subnet ID	subnet-0b8044d04915fd91b		

And the ip of the EFS @172.31.16.61 is also in the same subnet

DNS name `fs-49a962ab.efs.us-east-1.amazonaws.com` ?

Mount instructions (from local VPC)

Mount instructions (across VPC peering connection)

Mount instructions

	Availability Zone	Subnet	IP address	Mount target ID
4de147cc1	us-east-1a	subnet-0b8044d04915fd918 (default)	172.31.16.61	fsmt-6f81578f
	us-east-1b	subnet-04f51f4d2fe464bc2 (default)	172.31.43.189	fsmt-70815790
	us-east-1c	subnet-0a8044d04915fd918 (default)	172.31.16.189	fsmt-70815790

3. Create a folder and mount the EFS on the EC2.

```
[ec2-user@ip-192-168-1-138 ~]$ mkdir efs01
[ec2-user@ip-192-168-1-138 ~]$ ls -l
total 0
drwxrwxr-x 2 ec2-user ec2-user 6 Dec  1 04:16 efs01
[ec2-user@ip-192-168-1-138 ~]$
```

To get the mount point.

File system access

DNS name fs-49a962ab.efs.us-east-1.amazonaws.com ⓘ

Amazon EC2 mount instructions (from local VPC)

↩ Click here

[Amazon EC2 mount instructions \(across VPC peering connection\)](#)

[On-premises mount instructions](#)

Mount targets

Mounting your file system

1. Open an SSH client and connect to your EC2 instance. (Find out [how to connect](#)).
2. Create a new directory on your EC2 instance, such as "efs".
 - ```
sudo mkdir efs
```
3. Mount your file system with a method listed following. If you need encryption of data in transit, use the EFS mount helper and the TLS mount option. [Mounting considerations](#)
  - Using the EFS mount helper:

```
sudo mount -t efs fs-49a962ab:/ efs
```
  - Using the EFS mount helper and the TLS mount option:

```
sudo mount -t efs -o tls fs-49a962ab:/ efs
```
  - Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-49a962ab.efs.us-east-1.amazonaws.com:/ efs
```

If you can't to connect, see our [troubleshooting documentation](#).

Close

Note: -- The above screen is part of the actual screen, the first part says to install the nfs-utils

The EC2 instance image we use in your labs already has these software's installed, hence you could skip the first part and move to the **"Mount Target"**.

### Mounting your file system

1. Open an SSH client and connect to your EC2 instance. (Find out [how to connect](#)).
2. Create a new directory on your EC2 instance, such as "efs".

- `sudo mkdir efs`

3. Mount your file system with a method listed following. If you need encryption of data in transit, use the EFS mount helper and the TLS mount option. [Mounting considerations](#)

- Using the EFS mount helper:

```
sudo mount -t efs fs-49a962ab:/ efs
```

- Using the EFS mount helper and the TLS mount option:

```
sudo mount -t efs -o tls fs-49a962ab:/ efs
```

- Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport fs-49a962ab.efs.us-east-1.amazonaws.com:/ efs
```

If you can't to connect, see our [troubleshooting documentation](#).

Close

Copy the one that is highlighted and paste it on the ec2 screen.

Note: -- in our example the folder name on the ec2 instance is "efs01".

Hence after copying the command change the "efs" to "efs01" as below.

```
[ec2-user@ip-192-168-1-138 ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport fs-9a4c5ad0.efs.us-east-1.amazonaws.com:/ efs01
[ec2-user@ip-192-168-1-138 ~]$ df -h
```

| Filesystem                                | Size | Used | Avail | Use% | Mounted on           |
|-------------------------------------------|------|------|-------|------|----------------------|
| devtmpfs                                  | 476M | 0    | 476M  | 0%   | /dev                 |
| tmpfs                                     | 493M | 0    | 493M  | 0%   | /dev/shm             |
| tmpfs                                     | 493M | 444K | 493M  | 1%   | /run                 |
| tmpfs                                     | 493M | 0    | 493M  | 0%   | /sys/fs/cgroup       |
| /dev/xvda1                                | 8.0G | 1.2G | 6.9G  | 15%  | /                    |
| tmpfs                                     | 99M  | 0    | 99M   | 0%   | /run/user/1000       |
| tmpfs                                     | 99M  | 0    | 99M   | 0%   | /run/user/0          |
| fs-9a4c5ad0.efs.us-east-1.amazonaws.com:/ | 8.0E | 0    | 8.0E  | 0%   | /home/ec2-user/efs01 |

```
[ec2-user@ip-192-168-1-138 ~]$
```

Then run

**\$ df -h**

This is to check if the NFS is mounted are not.

We can see that that it is successfully mounted.

Note: --

At this stage if you are not able to see the mount point, which means the “Security Group” on EFS is having some issues.

Or

The VPC of EFS and EC2 instance is different

Or

NACL on that subnets is BLOCKING something

### Step4 : Creating a Mount Target using the AWS CLI

To create a mount target using AWS CLI, use the create-mount-target CLI command (corresponding operation is [CreateMountTarget](#)), as shown following.

```
$ aws efs create-mount-target \
--file-system-id file-system-id \
--subnet-id subnet-id \
--security-group ID-of-the-security-group-created-for-mount-target \
--region aws-region \
--profile adminuser
```

After successfully creating the mount target, Amazon EFS returns the mount target description as JSON as shown in the following example.

```
{
 "MountTargetId": "fsmt-f9a14450",
 "NetworkInterfaceId": "eni-3851ec4e",
 "FileSystemId": "fs-b6a0451f",
 "LifeCycleState": "available",
 "SubnetId": "subnet-b3983dc4",
 "OwnerId": "23124example",
 "IpAddress": "10.0.1.24"
}
```

You can also retrieve a list of mount targets created for a file system using the describe-mount-targets CLI command (corresponding operation is [DescribeMountTargets](#)), as shown following.

```
$ aws efs describe-mount-targets \
--file-system-id file-system-id \
--region aws-region \
--profile adminuser
```