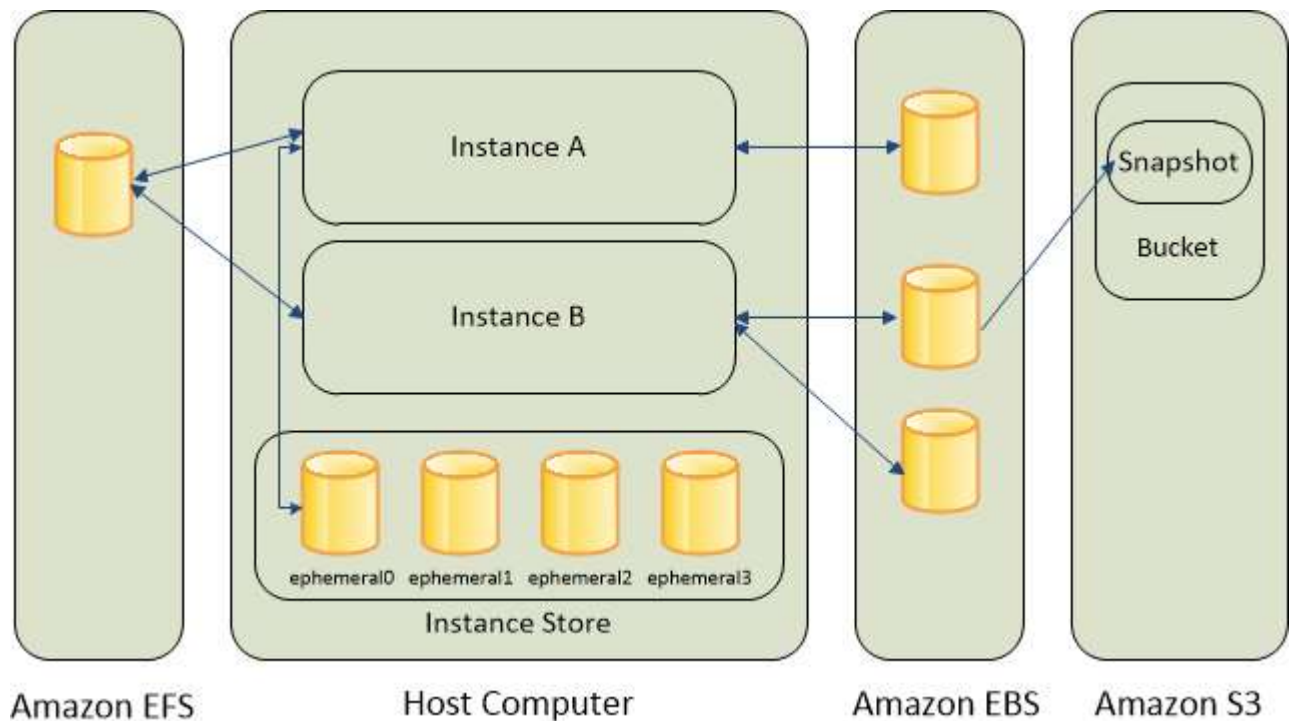


## Amazon Storage

- Amazon EC2 provides you with flexible, cost effective, and easy-to-use data storage options for your instances.
- Types of storage option for EC2:
  - (1) Amazon Elastic Block Storage (EBS)
  - (2) Amazon Elastic File system (EFS)
  - (3) Amazon Simple Storage Service (S3)
  - (4) Amazon EC2 Instance Storage.
- The following figure shows the relationship between these storage options and your instance.



### Amazon EBS:

- Amazon EBS provides durable, block-level storage volumes that you can attach to a running instance.
- After an EBS volume is attached to an instance, you can use it like any other physical hard drive.
- You can use Amazon EBS as a primary storage device for data that requires frequent and granular updates.

- You can also detach an EBS volume from one instance and attach it to another instance.
- EBS volumes can also be created as encrypted volumes using the Amazon EBS encryption feature.
- To keep a backup copy of your data, you can create a snapshot of an EBS volume, which is stored in Amazon S3.
- You can create an EBS volume from a snapshot, and attach it to another instance.

### Amazon EFS:

- Amazon EFS provides scalable file storage for use with Amazon EC2.
- You can create an EFS file system and configure your instances to mount the file system.
- You can use an EFS file system as a common data source for workloads and applications running on multiple instances.
- Amazon EFS is **not supported on Windows instances**.
- You can mount an Amazon EFS file system on instances in **only one VPC at a time**.

### Amazon S3:

- Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web.
- You can use Amazon S3 to store backup copies of your data and applications.
- Amazon EC2 uses Amazon S3 to store EBS snapshots and instance store-backed AMIs.
- you can copy a file to or from Amazon S3 and your instance using method.

```
[ec2-user ~]$ wget https://my_bucket.s3.amazonaws.com/path-to-file
```

- To copy an object from Amazon S3 to your instance.

```
$ aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

- To download an entire Amazon S3 bucket to a local directory on your instance.

```
[ec2-user ~]$ aws s3 sync s3://remote_S3_bucket local_directory
```

### Amazon EC2 Instance Store:

- Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web.
- Many instances can access storage from disks that are physically attached to the host computer. This disk storage is referred to as instance store. Instance store provides temporary block-level storage for instances.
- The data on an instance store volume persists only during the life of the associated instance; if you stop or terminate an instance, any data on instance store volumes is lost.

## Which AWS Cloud Storage Service Is Best?

- Amazon S3 is cheapest for data storage alone. However, there are various other pricing parameters in S3, including cost per number of requests made, S3 Analytics, and data transfer out of S3 per gigabyte. EFS has the simplest cost structure.
- Amazon S3 can be accessed from anywhere. AWS EBS is only available in a particular region, while you can share files between regions on multiple EFS instances.
- EBS and EFS are both faster than Amazon S3, with high IOPS and lower latency.
- EBS is scalable up or down with a single API call. Since EBS is cheaper than EFS, you can use it for database backups and other low-latency interactive applications that require consistent, predictable performance.
- EFS is best used for large quantities of data, such as large analytic workloads. Data at this scale cannot be stored on a single EC2 instance allowed in EBS—requiring users to break up data and distribute it between EBS instances. The EFS service allows concurrent access to thousands of EC2 instances, making it possible to process and analyze large amounts of data seamlessly.

## Amazon Elastic File System (Amazon EFS)

An Amazon EFS file system is excellent as a managed **network file system (NFS)** that can be shared across different Amazon EC2 instances. Amazon EFS works like NAS devices and performs well for big data analytics, media processing workflows, and content management.

It is very important to know which is a more suitable storage service for your specific needs. People decide on a storage service based on their system requirements as well as parameters such as cost, performance and access type.

AWS EFS is a shared, elastic file storage system that grows and shrinks as you add and remove files. It offers a traditional file storage paradigm, with data organized into directories and subdirectories. EFS is useful for SaaS applications and content management systems. You can mount EFS onto several EC2 instances at the same time.

Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. You can use an EFS file system as a common data source for workloads and applications running on multiple instances.

With Amazon EFS, you pay only for the storage used by your file system and there is no minimum fee or setup cost.

Amazon EFS offers **two storage classes**:

1. **Standard**: The Standard storage class is used to store frequently accessed files.
2. **Infrequent Access**. The Infrequent Access (IA) storage class is a lower-cost storage class that's designed for storing long-lived, infrequently accessed files cost-effectively.
3. By simply enabling EFS Lifecycle Management on your file system, files not accessed according to the lifecycle policy you choose will be automatically and transparently moved into EFS IA. The EFS IA storage class costs only \$0.025/GB-month\*.

Amazon EFS supports **two forms of encryption** for file systems:

1. **Encryption in transit**
2. **Encryption at rest.**

You can enable encryption at rest when creating an Amazon EFS file system. If you do, all your data and metadata is encrypted. You can enable encryption in transit when you mount the file system.

Amazon EFS is designed to provide the throughput, IOPS, and low latency needed for a broad range of workloads.

### **NOTE:**

- **Amazon EFS is not supported on Windows instances.**
- You can **mount** an Amazon EFS file system on instances in **only one VPC at a time.**

You can mount an Amazon EFS file system in your VPC, through the Network File System versions 4.0 and 4.1 (NFSv4) protocol.

List of Amazon EC2 Linux Amazon Machine Images (AMIs) that support this protocol,

- Amazon Linux 2
- Amazon Linux 2015.09 or newer
- RHEL 7.3 or newer
- RHEL 6.9 with kernel 2.6.32-696 or newer
- All versions of Ubuntu 16.04
- Ubuntu 14.04 with kernel 3.13.0-83 or newer
- SLES 12 Sp2 or later
- If you are using another distribution or a custom kernel, we recommend **kernel version 4.3 or newer**

For some AMIs, you'll need to install an NFS client to mount your file system on your Amazon EC2 instance.

### Installing the NFS Client:

To mount your Amazon EFS file system on your Amazon EC2 instance, first you need to install an NFS client. To connect to your EC2 instance and install an NFS client, you need the public DNS name of the EC2 instance and a user name to log in. That user name for your instance is typically ec2-user.

1. Connect to your EC2 instance.
2. (Optional) Get updates and reboot.
3. After the reboot, reconnect to your EC2 instance.
4. Install the NFS client.

If you're using an Amazon Linux AMI or Red Hat Linux AMI

```
$ sudo yum -y install nfs-utils
```

If you're using an Ubuntu Amazon EC2 AMI

```
$ sudo apt-get -y install nfs-common
```

**NOTE:** If you

choose Amazon Linux AMI 2016.03.0 or Amazon Linux AMI 2016.09.0 when launching your Amazon EC2 instance, you don't need to install nfs-utils because it's already included in the AMI by default.

## Mounting on Amazon EC2 with a DNS Name

You can mount an Amazon EFS file system on an Amazon EC2 instance using DNS names. You can do this with a DNS name for the file system, or a DNS name for a mount target.

### 1. File system DNS name:

If you have the file system ID, you can construct it using the following convention.

```
$ file-system-id.efs.aws-region.amazonaws.com
```

Using the file system DNS name, you can mount a file system on your Amazon EC2 instance with the following command.

```
$ sudo mount -t nfs -o nfsvers=4.1, rsize=1048576, wsize=1048576,
hard,timeo=600,retrans=2,noresvport file-system-id.efs.aws-
region.amazonaws.com:/ efs-mount-point
```

(Or)

```
sudo mount -t efs fs-12345678:/ /mnt/efs → fs-12345678=File_System_ID
```

If you want to use encryption of data in transit, you can mount your file system with the following command.

```
sudo mount -t efs -o tls fs-12345678:/ /mnt/efs
```

### 2. Mount target DNS name

```
$ availability-zone.file-system-id.efs.aws-region.amazonaws.com
```

In some cases, you might delete a mount target and then create a new one in the same Availability Zone. In such a case, the DNS name for that new mount target in that Availability Zone is the same as the DNS name for the old mount target.

To be able to use a DNS name in the mount command, the following must be true:

- The connecting EC2 instance must be inside a VPC and must be configured to use the DNS server provided by Amazon.
- The VPC of the connecting EC2 instance must have both **DNS Resolution** and **DNS Hostnames enabled**.
- The connecting EC2 instance must be inside the same VPC as the EFS file system.

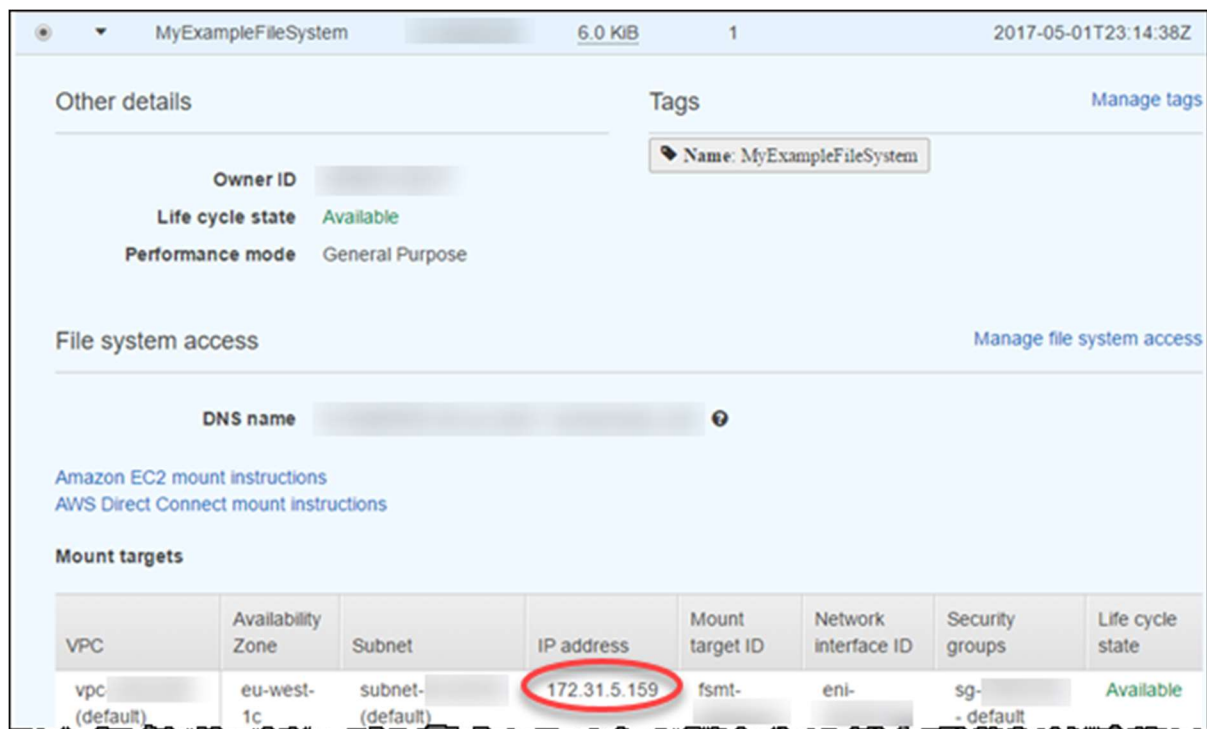
## Mounting with an IP Address

As an alternative to mounting your Amazon EFS file system with the DNS name, Amazon EC2 instances can mount a file system using a mount target's IP address. Mounting by IP address works

in environments where DNS is disabled, such as VPCs with DNS hostnames disabled, and EC2-Classic instances mounting using ClassicLink.

You can get the mount target IP address for your EFS file system through the console using the following procedure.

1. Open the **Amazon Elastic File System** console.
2. Choose the **Name** value of your EFS file system for **File systems**.
3. In the **Mount targets** table, identify the **Availability Zone** that you want to use to mount your EFS file system to your EC2 instance.
4. Make a note of the **IP address** associated with your chosen **Availability Zone**.



You can specify the IP address of a mount target in the mount command, as shown following.

```
$ sudo mount -t nfs -o  
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,nores  
vport mount-target-IP:/ ~/efs-mount-point
```

## Tasks:

- Step-1: Prerequisites
- Step-2: Create an EFS File System
- Step-3: Mount the File System
- Step-4: Test the File System
- Step-5: Clean Up

### Step-1: Prerequisites:

Create a security group to associate with the EC2 instances and EFS mount target, and add the following rules:

- Allow inbound SSH connections to the EC2 instances from your computer (the source is the CIDR block for your network Ex:12.245.2.56/32 ).
- Allow inbound NFS connections to the file system via the EFS mount target from the EC2 instances that are associated with this security group (the source is the security group itself).

### Step-2: Create an EFS File System:

Amazon EFS enables you to create a file system that multiple instances can mount and access at the same time.

1. Open the **Amazon Elastic File System** console at <https://console.aws.amazon.com/efs/>.
2. Choose **Create file system**.
3. On the **Configure file system** access page, do the following:
  - a. For **VPC**, select the VPC to use for your instances.
  - b. For **Create mount targets**, select all the Availability Zones.
  - c. For each Availability Zone, ensure that the value for **Security group** is the security group that you created in Prerequisites.
  - d. Choose **Next Step**.
4. On the **Configure optional settings** page, do the following:
  - a. For the tag with **Key=Name**, type a name for the file system in **Value**.
  - b. For **Choose performance mode**, keep the default option, **General Purpose**.
  - c. Choose **Next Step**.
5. On the **Review and create** page, choose **Create File System**.
6. After the file system is created, **note the file system ID**

### Step-3: Mount the File System

Use the following procedure to launch two t2.micro instances. The user data script mounts the file system to both instances during launch and updates `/etc/fstab` to ensure that the file system is



remounted after an instance reboot. Note that T2 instances must be launched in a subnet. You can use a default VPC or a nondefault VPC.

**Note:** There are other ways that you can mount the volume (for example, on an already running instance).

1. Open the **Amazon EC2 console** at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image** page, select an **Amazon Linux AMI** with the HVM virtualization type.
4. On the **Choose an Instance Type** page, keep the default instance type, **t2.micro** and choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, do the following:
  - a. For **Number of instances**, type 2.
  - b. [Default VPC] If you have a default VPC, it is the default value for **Network**. Keep the default VPC and the default value for **Subnet** to use the default subnet in the Availability Zone that Amazon EC2 chooses for your instances.  
  
[Nondefault VPC] Select your VPC for **Network** and a public subnet from **Subnet**.
  - c. [Nondefault VPC] For **Auto-assign Public IP**, choose **Enable**. Otherwise, your instances do not get public IP addresses or public DNS names.
  - d. Under **Advanced Details**, select **As text**, and paste the following script into **User data**. Update **FILE\_SYSTEM\_ID** with the **ID** of your file system (from **Step-1**). You can optionally update **MOUNT\_POINT** with a directory for your mounted file system.

```
#!/bin/bash
yum update -y
yum install -y nfs-utils
FILE_SYSTEM_ID=fs-xxxxxxx
AVAILABILITY_ZONE=$(curl -s http://169.254.169.254/latest/meta-
data/placement/availability-zone )
REGION=${AVAILABILITY_ZONE:0:-1}
MOUNT_POINT=/mnt/efs
mkdir -p ${MOUNT_POINT}
chown ec2-user:ec2-user ${MOUNT_POINT}
```

```
echo ${FILE_SYSTEM_ID}.efs.${REGION}.amazonaws.com:/
${MOUNT_POINT} nfs4
nfsvers=4.1, rsize=1048576, wsize=1048576, hard, timeo=600, retrans=2, _netdev 0 0 >> /etc/fstab
mount -a -t nfs4
```

- e. Advance to Step 6 of the wizard.
6. On the **Configure Security Group** page, choose **Select** an existing security group and select the security group that you created in **Prerequisites**, and then choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.
8. In the **Select an existing key pair or create a new key pair** dialog box, select **Choose** an existing key pair and choose your key pair. Select the acknowledgment check box, and choose **Launch Instances**.
9. In the navigation pane, choose **Instances** to see the status of your instances. Initially, their status is **pending**. After the status changes to **running**, your instances are **ready for use**.

## Step-4: Test the File System

You can connect to your instances and verify that the file system is mounted to the directory that you specified.

1. Connect to your instances. For more information, see [Connect to Your Linux Instance](#).
2. From the terminal window for each instance, run the **“df -T”** command to verify that the EFS file system is mounted.

```
$ df -T
```

Filesystem	Type	1K-blocks	Used	Available	Use%	Mounted on
/dev/xvda1	ext4	8123812	1949800	6073764	25%	/
devtmpfs	devtmpfs	4078468	56	4078412	1%	/dev
tmpfs	tmpfs	4089312	0	4089312	0%	/dev/shm
<b>efs-dns</b>	nfs4	9007199254740992	0	9007199254740992	0%	/mnt/efs

3. (Optional) Create a file in the file system from one instance, and then verify that you can view the file from the other instance.
  - a) From the first instance, run the following command to create the file:

```
$ sudo touch /mnt/efs/test-file.txt
```

b) From the second instance, run the following command to view the file:

```
$ ls /mnt/efs  
test-file.txt
```

## Step-5: Transfer Files to Amazon EFS Using AWS DataSync

- Now that you have created a functioning Amazon EFS file system, you can use **AWS DataSync** to transfer files from an existing file system to Amazon EFS.
- AWS DataSync is a data transfer service that simplifies, automates, and accelerates moving and replicating data between on-premises storage systems and AWS storage services over the internet or **AWS Direct Connect**.
- AWS DataSync can transfer your file data, and also file system metadata such as ownership, time stamps, and access permissions.

### Before You Begin

In this step, we assume that you have the following:

- A **source NFS file system** that you can **transfer files from**. This **source system needs** to be accessible over **NFS version 3, version 4, or 4.1**.
- A **destination Amazon EFS file system** to **transfer files to**. If you don't have an Amazon EFS file system, create one.
- Your server and network meet the AWS DataSync requirements.

To transfer files from a source location to a destination location using AWS DataSync, you do the following:

- Download and deploy an agent in your environment and activate it.
- Create and configure a source and destination location.
- Create and configure a task.
- Run the task to transfer files from the source to the destination.

## Step-6: Clean UP (Terminate the Instance and Delete the File System)

### To delete the file system

1. Connect to your Amazon EC2 instance.
2. **Unmount** the Amazon EFS file system with the following command.  

```
$ sudo umount efs
```
3. Open the **Amazon Elastic File System** console at <https://console.aws.amazon.com/efs/>.

4. Select the file system to delete.
5. Choose **Actions, Delete file system**.
6. When prompted for confirmation, **type the ID** of the file system and choose **Delete File System**.

### To terminate the instances

1. Open the **Amazon EC2 console** at <https://console.aws.amazon.com/ec2/>.
2. In the **navigation pane**, choose **Instances**.
3. Select the instances to terminate.
4. Choose **Actions, Instance State, Terminate**.
5. Choose **Yes, Terminate** when prompted for confirmation.

### Turn Off the ID Mapper

The NFS utilities in the operating system include a daemon called an **ID Mapper** that manages mapping between user names and IDs.

However, Amazon EFS deals only with numeric IDs. We recommend that you turn this process off on your EC2 instances. On Amazon Linux, the ID mapper is usually disabled, and if it is don't enable it. To turn off the ID mapper, use the commands shown following.

```
$ service rpcidmapd status  
  
$ sudo service rpcidmapd stop
```

### AWS Backup with Amazon EFS

- AWS Backup is a simple and cost-effective way to protect your data by backing up your Amazon EFS file systems.
- AWS Backup is a unified backup service designed to simplify the creation, migration, restoration, and deletion of backups, while providing improved reporting and auditing.
- Amazon EFS is integrated with AWS Backup. You can use AWS Backup to set backup plans where you specify your backup frequency, when to back up, how long to retain backups, and a lifecycle policy for backups.

### How AWS Backup Works with EFS File Systems

- With AWS Backup, first you create a backup plan. The backup plan defines backup schedule, backup window, retention policy, lifecycle policy, and tags.
  - **Schedule** – When the backup occurs
  - **Backup window** – The window of time in which the backup needs to start
  - **Lifecycle** – When to move a recovery point to cold storage and when to delete it
  - **Backup vault** – Used to organize recovery points created by the Backup rule.
- After your backup plan is created, you assign the specific Amazon EFS file systems to the backup plan by using either tags or the Amazon EFS file system ID.
- After a plan is assigned, AWS Backup begins automatically backing up the Amazon EFS file system on your behalf according to the backup plan that you defined.
- You can use the AWS Backup console to manage backup configurations or monitor backup activity.
- AWS Backup performs incremental backups of EFS file systems.
- you can expect the following backup rates with AWS Backup:
  - 100 MB/s for file systems composed of mostly large files
  - 500 files/s for file systems composed of mostly small files
  - The maximum duration for a backup or a restore operation in AWS Backup is seven days.
- Completion window for a backup. This window defines the period of time in which a backup needs to complete.
- Scheduled or on-demand backups may fail if a backup job is already in progress.
- You can restore a recovery point to a new EFS file system or to the source file system. In both cases, your recovery point is restored to the restore directory, `aws-backup-restore_2019-01-07T21-06-22-108Z`, which you can see at the root of the file system when the restore is complete.
- If the restore fails to complete, you can see the directory `aws-backup-failed-restore_2019-01-07T21-06-22-108Z`.
- You need to manually delete this directory when you are through using it.

## Using the `amazon-efs-utils` Tools

- The `amazon-efs-utils` package is an open-source collection of Amazon EFS tools.

- There's no additional cost to use amazon-efs-utils, and you can download these tools from GitHub here: <https://github.com/aws/efs-utils>.
- The amazon-efs-utils package is available in the Amazon Linux package repositories, and you can build and install the package on other Linux distributions.
- The amazon-efs-utils package comes with a mount helper and tooling that makes it easier to perform encryption of data in transit for Amazon EFS.
- A *mount helper* is a program that you use when you mount a specific type of file system.
- The following dependencies exist for amazon-efs-utils and are installed when you install the amazon-efs-utils package:
  - NFS client (the nfs-utils package)
  - Network relay (stunnel package, version 4.56 or later)
  - Python (version 2.7 or later)
  - OpenSSL 1.0.2 or newer
- When using the Amazon EFS mount helper with Transport Layer Security (TLS), the mount helper enforces certificate hostname checking.
- The following Linux distributions support amazon-efs-utils:
  - Amazon Linux 2
  - Amazon Linux
  - Red Hat Enterprise Linux (and derivatives such as CentOS) version 7 and newer
  - Ubuntu 16.04 LTS and newer

## Installing the amazon-efs-utils Package

### a) On Amazon Linux:

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

### b) On Other Linux Distributions

```
git clone https://github.com/aws/efs-utils
sudo yum -y install make
```

## Upgrading Stunnel:

After installing the Amazon EFS mount helper, you can upgrade your system's version of stunnel with the following instructions.

1. In a web browser, go to the stunnel downloads page <https://stunnel.org/downloads.html>.
2. Locate the latest stunnel version that is available in tar.gz format. Note the name of the file as you will need it in the following steps.
3. Open a terminal on your Linux client, and run the following commands in the order presented.

```
sudo yum install -y gcc openssl-devel tcp_wrappers-devel
```

4. Replace `latest-stunnel-version` with the name of the file you noted previously in Step 2.

```
$ sudo curl -o latest-stunnel-version.tar.gz
https://www.stunnel.org/downloads.html/latest-stunnel-
version.tar.gz
$ sudo tar xvfz latest-stunnel-version.tar.gz
$ cd latest-stunnel-version/
$ sudo ./configure
$ sudo make
```

5. The current `amazon-efs-utils` package is installed in `bin/stunnel`. So that the new version can be installed, remove that directory with the following command.

```
sudo rm /bin/stunnel
sudo make install
```

6. The default CentOS shell is `csh`, which has different syntax than the `bash` shell. The following code first invokes `bash`, then runs.

```
bash
if [[ -f /bin/stunnel ]]; then
sudo mv /bin/stunnel /root
fi
$ sudo ln -s /usr/local/bin/stunnel /bin/stunnel
```

7. After you've installed a version of stunnel with the required features, you can mount your file system using TLS with the recommended settings.

## Disabling Certificate Hostname Checking

If you are unable to install the required dependencies, you can optionally disable certificate hostname checking inside the Amazon EFS mount helper configuration. We do not recommend that you disable this feature in production environments. To disable certificate host name checking, do the following:

1. Using your text editor of choice, open the `/etc/amazon/efs/efs-utils.conf` file.
2. Set the `stunnel_check_cert_hostname` value to `false`.
3. Save the changes to the file and close it.

## Enabling Online Certificate Status Protocol

In order to maximize file system availability in the event that the CA is not reachable from your VPC, the Online Certificate Status Protocol (OCSP) is not enabled by default when you choose to encrypt data in transit.

1. Open a terminal on your Linux client.
2. Using your text editor of choice, open the `/etc/amazon/efs/efs-utils.conf` file.
3. Set the `stunnel_check_cert_validity` value to `true`.
4. Save the changes to the file and close it.

## To enable OCSP as part of the `mount` command

Use the following `mount` command to enable OCSP when mounting the file system.

```
$ sudo mount -t efs -o tls,ocsp fs-12345678:/ /mnt/efs
```

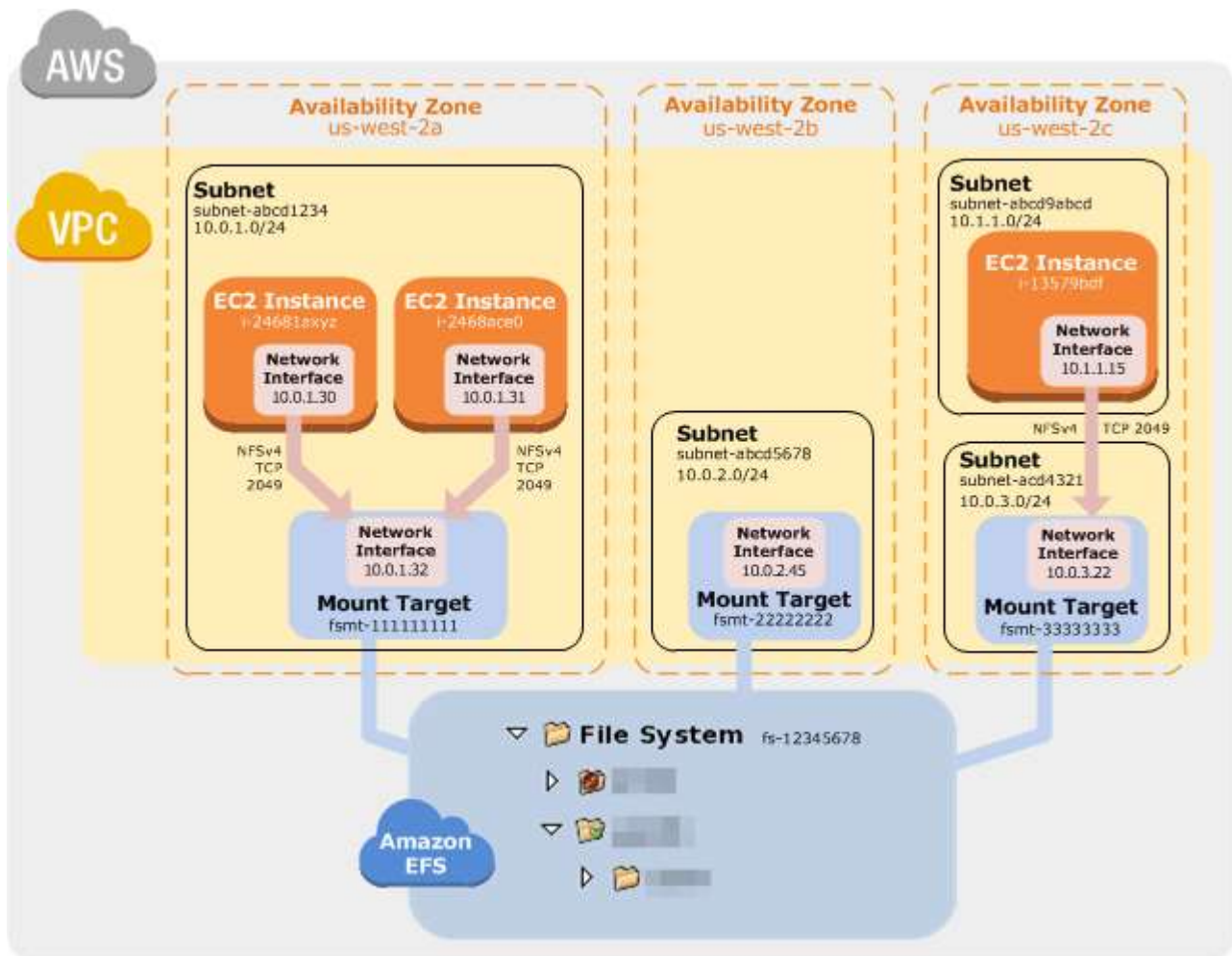
## Managing File System Network Accessibility

You mount your file system on an EC2 instance in your virtual private cloud (VPC) using a mount target that you create for the file system. Managing file system network accessibility refers to managing the mount targets.

The following illustration shows how EC2 instances in a VPC access an Amazon EFS file system using a mount target.

The illustration shows three EC2 instances launched in different VPC subnets accessing an Amazon EFS file system. The illustration also shows one mount target in each of the Availability Zones (regardless of the number of subnets in each Availability Zone).





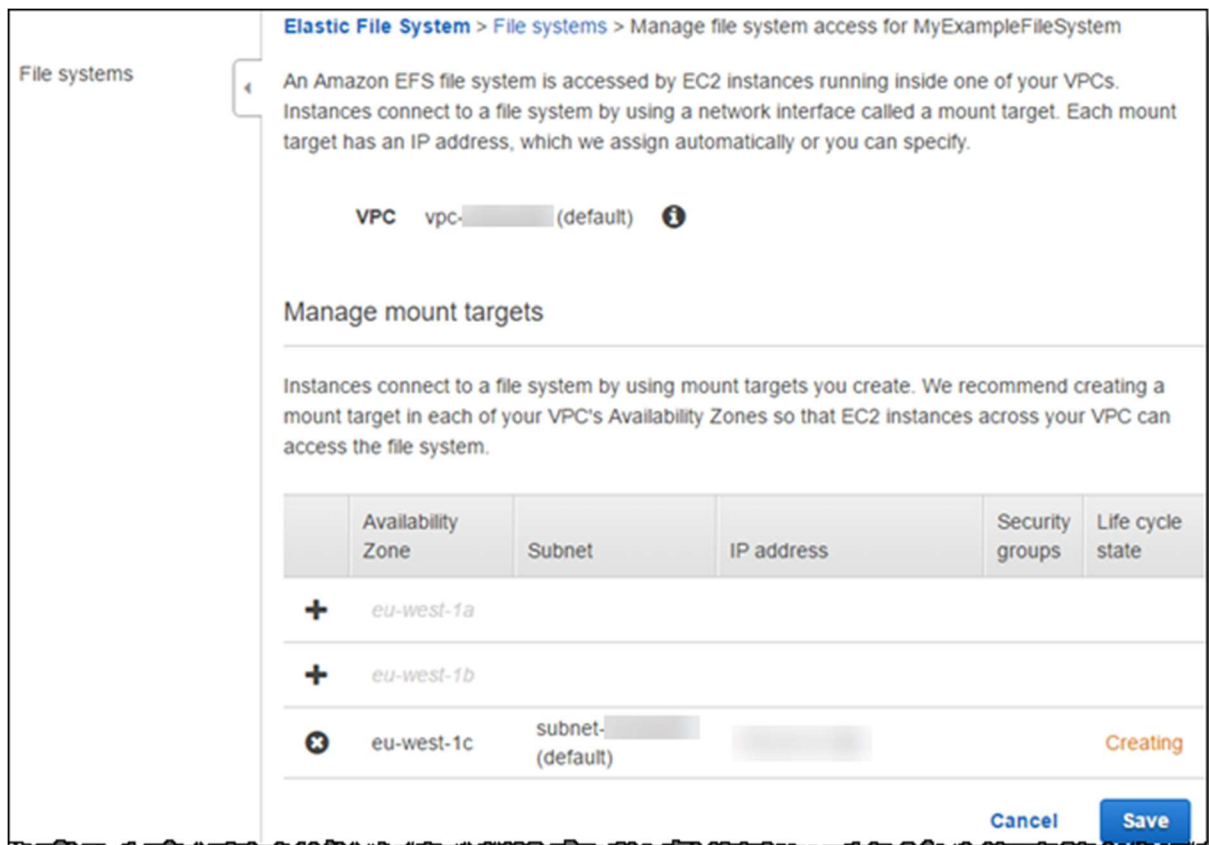
## Creating or Deleting Mount Targets in a VPC

To access an Amazon EFS file system in a VPC, you need mount targets. For an Amazon EFS file system, the following is true:

- You can create one mount target in each Availability Zone.
- If the VPC has multiple subnets in an Availability Zone, you can create a mount target in only one of those subnets. All EC2 instances in the Availability Zone can share the single mount target.

1. In the **Amazon EFS** console, choose the **file system**, and for **Actions** choose **Manage File System Access**.

The console displays the **Manage File System Access** page with a list of file system mount targets you have created in the selected VPC. The console shows a list of Availability Zones and mount target information, if there is a mount target in that Availability Zone.



2. To create **new mount targets**
  - a. Click on the left side in the specific **Availability Zone** row.
  - b. If the Availability Zone has multiple subnets, select a subnet from the **Subnet** list.
  - c. Amazon EFS automatically selects an available IP address, or you can provide another IP address explicitly.
  - d. Choose a Security Group from the list.
3. To **delete a mount target**, choose the **X** next to the Availability Zone from which you want to remove a mount target.
4. Using **AWS CLI**:

```
$ aws efs create-mount-target \
--file-system-id file-system-ID (for which to create the mount
target) \
--subnet-id vpc-subnet-ID (in which to create mount target) \
--security-group security-group IDs (to associate with the mount
target) \
--region aws-region (for example, us-west-2) \
--profile adminuser
```

```
$ aws efs describe-mount-targets \
```

```
--file-system-id file-system-ID \  
--region aws-region-where-file-system-exists \  
--profile adminuser
```

## Changing the VPC for Your Mount Target

You can use an Amazon EFS file system in one VPC based on the Amazon VPC service at a time.

To access the file system from EC2 instances in another VPC, you must first delete the current mount targets and then create new mount targets, as described following.

1. In the **Amazon EFS console**, choose the **file system**, and for **Actions**, choose **Manage File System Access**.

The console displays the **Manage File System Access** page with a list of mount targets that you created for the file system in a VPC.

2. Choose another VPC for **VPC** to choose the VPC.

The console clears all of the mount target information and lists only the Availability Zone.

3. Create mount targets in one or more Availability Zones as follows:
  - a. If the Availability Zone has multiple subnets, choose a subnet for **Subnet**.
  - b. Amazon EFS automatically **selects an available IP address**, or you can **provide another IP address** explicitly.
  - c. Choose the security groups that you want to associate. For inter-region VPC peering, the security groups that you choose need to have a rule that allows inbound traffic over NFS (port 2049) from your other VPC or VPCs.
4. Choose **Save**.

The console first deletes the mount targets from the previous VPC and then creates new mount targets in the new VPC that you selected.

## Updating the Mount Target Configuration

- After you create a mount target for your file system, you might want to update security groups that are in effect.
- You can't change the IP address of an existing mount target. To change an IP address, delete the mount target and create a new one with the new address.
- Deleting a mount target breaks any existing file system mounts.

1. In the **Amazon EFS** console, choose the **file system** and for **Actions**, choose **Manage File System Access**.

The console displays the **Manage File System Access** page with a list of Availability Zones and mount target information, if there is a mount target in the Availability Zone.

2. In the **Security Group** column, you can add or remove security groups. Choose X to remove an existing security group. Choose the **Security Group** box to choose from other available security groups.

If you remove all security groups, Amazon EFS assigns the VPC's default security group.

3. Using AWS CLI:

```
$ aws efs modify-mount-target-security-groups \
--mount-target-id mount-target-ID-whose-configuration-to-update \
--security-groups security-group-ids-separated-by-space \
--region aws-region-where-mount-target-exists \
--profile adminuser
```

#### NOTE:

How to use the stat utility on an empty file to return the file's disk usage.

```
$ /usr/bin/stat --format="%b*%B" file | bc
4096
```

To measure the size of a directory, use the stat utility

```
$ /usr/bin/stat --format="%b*%B" . | bc
4096
```

## Mounting Your Amazon EFS File System Automatically

You can use **fstab** to automatically mount your Amazon EFS file system using the **mount helper** whenever the Amazon EC2 instance it is mounted on reboots.

You can set up automatic mounting in two ways.

1. You can update the `/etc/fstab` file in your EC2 instance after you connect to the instance for the first time
2. You can configure automatic mounting of your EFS file system when you create your EC2 instance.

### To configure your EC2 instance to mount an EFS file system automatically at launch

1. Make sure that you have created your Amazon EFS file system
2. Open the **Amazon EC2 console** at <https://console.aws.amazon.com/ec2/>.
3. Choose **Launch Instance**.
4. In Step 1: Choose an Amazon Machine Image (AMI), find an Amazon Linux AMI at the top of the list and choose Select.
5. In Step 2: Choose an **Instance Type**, choose Next: Configure Instance Details.
6. In Step 3: **Configure Instance Details**, provide the following information:
  1. For **Network**, choose the entry for the same **VPC** that the EFS file system you're mounting is in.
  2. For **Subnet**, choose a default subnet in any Availability Zone.
  3. For **File systems**, choose the **EFS file system that you want to mount**. The path shown next the file system ID is the mount point that the EC2 instance will use, which you can change. Choose **Add to user data** to mount the file system when the EC2 is launched.
  4. Under **Advanced Details**, confirm that the user data is present in **User data**.
7. Choose Next: **Add Storage**.
8. Choose Next: **Add Tags**.
9. Name your instance and choose Next: Configure Security Group.
10. In Step 6: **Configure Security Group**, set Assign a security group to Select an existing security group. Choose the default security group to make sure that it can access your EFS file system.

You can't access your EC2 instance by Secure Shell (SSH) using this security group. For access by SSH, later you can edit the default security and add a rule to allow SSH or a new security group that allows SSH. You can use the following settings:

- Type: SSH
- Protocol: TCP
- Port Range: 22
- Source: Anywhere 0.0.0.0/0

11. Choose **Review and Launch**.
12. Choose **Launch**.
13. Select the check box for the key pair that you created, and then choose **Launch Instances**.

Your EC2 instance is now configured to mount the EFS file system at launch and whenever it's rebooted.

## Updating an Existing EC2 Instance to Mount Automatically

To automatically remount your Amazon EFS file system directory when the Amazon EC2 instance reboots, you can use the file `fstab`. The file `fstab` contains information about file systems, and the command `mount -a`, which runs during instance startup, mounts the file systems listed in the `fstab` file.

Update the `/etc/fstab` file in your EC2 instance

1. Connect to your EC2 instance, and open the `/etc/fstab` file in an editor.
2. Add the following line to the `/etc/fstab` file.

```
fs-12345678:/mnt/efs efs defaults,_netdev 0 0
```

## Warning

- Use the `_netdev` option, used to identify network file systems, when **mounting your file system automatically**.
  - If `_netdev` is missing, your EC2 instance might stop responding. This result is because network file systems need to be initialized after the compute instance starts its networking.
3. Save the changes to the file.

Your EC2 instance is now configured to mount the EFS file system whenever it restarts.

## Creating CloudWatch Alarms to Monitor Amazon EFS

- You can create a CloudWatch alarm that sends an Amazon SNS message when the alarm changes state.
- An alarm watches a single metric over a time period you specify, and performs one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic or Auto Scaling policy.

- One important use of CloudWatch alarms for Amazon EFS is to enforce encryption at rest for your file system.
  - You can enable encryption at rest for an Amazon EFS file system when it's created.
  - To enforce data encryption-at-rest policies for Amazon EFS file systems, you can use Amazon CloudWatch and AWS CloudTrail to detect the creation of a file system and verify that encryption at rest is enabled.
1. Sign in to the **AWS Management Console** and open the **CloudWatch console** at <https://console.aws.amazon.com/cloudwatch/>.
  2. Choose **Create Alarm**. This launches the **Create Alarm Wizard**.
  3. Choose **EFS Metrics** and scroll through the Amazon EFS metrics to locate the metric you want to place an alarm on.
    - To display just the Amazon EFS metrics in this dialog box, search on the file system id of your file system.
    - Select the metric to create an alarm on and choose **Next**.
  4. Fill in the **Name**, **Description**, **Whenever** values for the metric.
  5. If you want CloudWatch to send you an email when the alarm state is reached, in the **Whenever this alarm:** field, choose **State is ALARM**. In the **Send notification to:** field, choose an **existing SNS topic**. If you select **Create topic**, you can set the name and email addresses for a new email subscription list. This list is saved and appears in the field for future alarms.

**Note:**

If you use Create topic to create a new Amazon SNS topic, the email addresses must be verified before they receive notifications. Emails are only sent when the alarm enters an alarm state. If this alarm state change happens before the email addresses are verified, they do not receive a notification.

6. At this point, the **Alarm Preview** area gives you a chance to preview the alarm you're about to create. Choose **Create Alarm**.