

EFS STORAGE

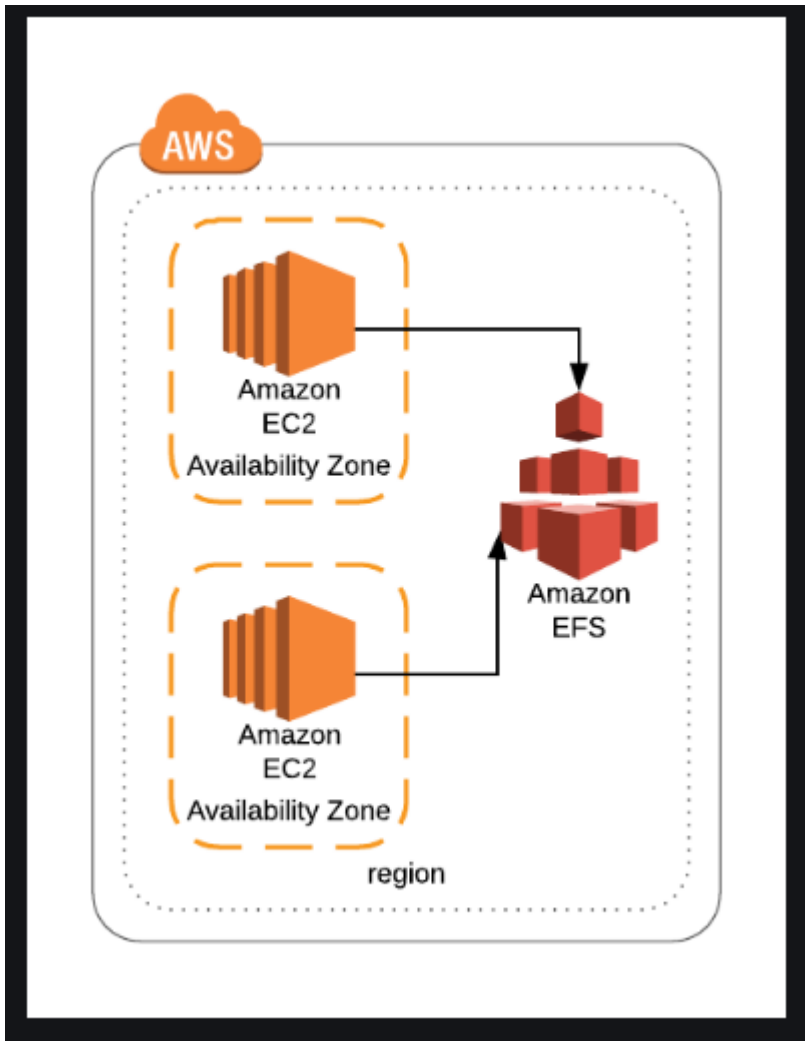
What is EFS storage in AWS?

- Amazon EFS is a regional service storing data within and across multiple Availability Zones (AZs) for high availability and durability.
- Amazon EC2 instances can access your file system across AZs, regions, and VPCs, while on-premises servers can access using AWS Direct Connect or AWS VPN.

How does AWS EFS work?

- You can create a file system, mount the file system on an Amazon EC2 instance, and then read and write data to and from your file system.
- You can mount an Amazon EFS file system in your VPC, through the Network File System versions 4.0 and 4.1 (NFSv4) protocol.

For example:



These steps to be followed for EFS

Step 1:

You should create 2 EC2 instance

White and black

aws Services Resource Groups nishanthanravi1993 Mumbai Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes

Free tier only

aws Services Resource Groups nishanthanravi1993 Mumbai Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

aws Services Resource Groups nishanthanravi1993 Mumbai Support

EC2 Dashboard Events Tags Reports Limits

INSTANCES

Instances

Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations

IMAGES

AMIs Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Launch Instance **Connect** **Actions**

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IP)
white	i-06ad7c468b211bb01	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-52-66-246-
black	i-0bdc1da2bb3056bc	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-13-127-73-

Instance: i-0bdc1da2bb3056bc (black) **Public DNS:** ec2-13-127-73-77.ap-south-1.compute.amazonaws.com

Description **Status Checks** **Monitoring** **Tags**

Instance ID	Instance state	Instance type	Elastic IPs	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Private DNS
i-0bdc1da2bb3056bc	running	t2.micro		ec2-13-127-73-77.ap-south-1.compute.amazonaws.com	13.127.73.77	-	in-172-31-7-270.ap-south-1.compute.internal

Step 2:

You can see EFS under storage

Create one EFS file

aws Services Resource Groups nishanthanravi1993 Mumbai Support

History

Find a service by name or feature (for example, EC2, S3 or VM, storage)

Group **A-Z**

Compute EC2 Lightsail ECR ECS EKS Lambda Batch Elastic Beanstalk Serverless Application Repository	Robotics AWS RoboMaker Customer Enablement AWS IQ Support Managed Services	Analytics Athena EMR CloudSearch Elasticsearch Service Kinesis QuickSight Data Pipeline AWS Glue AWS Lake Formation MSK	Business Applications Alexa for Business Amazon Chime WorkMail End User Computing WorkSpaces AppStream 2.0 WorkDocs WorkLink
Storage S3 EFS FSx S3 Glacier Storage Gateway	Blockchain Amazon Managed Blockchain Satellite Ground Station	Security, Identity, & Compliance IAM Resource Access Manager Cognito	Internet Of Things IoT Core Amazon FreeRTOS IoT 1-Click IoT Analytics

close

Use default

VPC, SG

Create file system

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-7f784317 (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"/>	ap-south-1a	subnet-89b293e1 (default)	Automatic	sg-e4e6ec8b - default ×
<input checked="" type="checkbox"/>	ap-south-1b	subnet-6d51c721 (default)	Automatic	sg-e4e6ec8b - default ×
<input checked="" type="checkbox"/>	ap-south-1c	subnet-40a1793b (default)	Automatic	sg-e4e6ec8b - default ×

Live it as a default

Create file system

Step 1: Configure file system access
Step 2: Configure optional settings
Step 3: Review and create

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 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
<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 92% 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 None

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**

Step 3: Review and create

File system access

VPC	Availability Zone	Subnet	IP address	Security groups
vpc-7f784317 (default)	ap-south-1a	subnet-89b293e1 (default)	Automatic	sg-e4e6ec8b - default
	ap-south-1b	subnet-6d51c721 (default)	Automatic	sg-e4e6ec8b - default
	ap-south-1c	subnet-40a1793b (default)	Automatic	sg-e4e6ec8b - default

Optional settings

Tags	No tags added
Performance mode	General Purpose
Throughput mode	Bursting
Encrypted	No
Lifecycle policy	None

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

Step 3:

Choose EC2 mount

Throughput mode

Bursting

Encrypted

No

Lifecycle policy

None

File system access

Manage file system access

DNS name fs-bec4116f.efs.ap-south-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-7f784317 (default)	ap-south-1c	subnet-40a1793b (default)	172.31.28.75	fsmt-12f80ac3	eni-03716b637128832c2		Creating
	ap-south-1b	subnet-6d51c721 (default)	172.31.5.126	fsmt-13f80ac2	eni-00e4a064b20b65267		Creating
	ap-south-1a	subnet-89b293e1 (default)	172.31.41.66	fsmt-14f80ac5	eni-0bc71448b8bb631a3		Creating

Amazon EC2 mount instructions (from local VPC)

To set up your EC2 instance:

- Using the [Amazon EC2 console](#), associate your EC2 instance with a VPC security group that enables access to your mount target. For example, if you assigned the "default" security group to your mount target, you should assign the "default" security group to your EC2 instance. [Learn more](#)
- Open an SSH client and connect to your EC2 instance. (Find out [how to connect](#).)
- If you're using an Amazon Linux EC2 instance, install the EFS mount helper with the following command:

sudo yum install -y amazon-efs-utils

You can still use the EFS mount helper if you're not using an Amazon Linux instance. [Learn more](#)

If you're not using the EFS mount helper, install the NFS client on your EC2 instance:

- On a Red Hat Enterprise Linux or SUSE Linux instance, use this command:

sudo yum install -y nfs-utils
- On an Ubuntu instance, use this command:

sudo apt-get install nfs-common

Mounting your file system

- Open an SSH client and connect to your EC2 instance. (Find out [how to connect](#).)

Close

Step 4:

Login to both machines

White and black

```
ec2-user@ip-172-31-10-140:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
  _|  _|_ )  
  _| ( _|_ /  Amazon Linux 2 AMI  
  __| \__|__|  
  
https://aws.amazon.com/amazon-linux-2/  
8 package(s) needed for security, out of 15 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-10-140 ~]$
```

```
ec2-user@ip-172-31-7-220:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
  _|  _|_ )  
  _| ( _|_ /  Amazon Linux 2 AMI  
  __| \__|__|  
  
https://aws.amazon.com/amazon-linux-2/  
8 package(s) needed for security, out of 15 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-7-220 ~]$
```

Copy this

- Open an SSH client and connect to your EC2 instance. (Find out [how to connect](#).)
- If you're using an Amazon Linux EC2 instance, install the EFS mount helper with the following command:

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

You can still use the EFS mount helper if you're not using an Amazon Linux instance. [Learn more](#)

If you're not using the EFS mount helper, install the NFS client on your EC2 instance:


```
[ec2-user@ip-172-31-10-140 ~]$ sudo su -
[root@ip-172-31-10-140 ~]# sudo yum install -y amazon-efs-utils
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 2.4 kB 00:00
Resolving Dependencies
--> Running transaction check
--> Package amazon-efs-utils.noarch 0:1.10-1.amzn2 will be installed
--> Processing Dependency: stunnel >= 4.56 for package: amazon-efs-utils-1.10-1.amzn2.noarch
--> Running transaction check
--> Package stunnel.x86_64 0:4.56-6.amzn2.0.3 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

Package	Arch	Version	Repository	Size
Installing:				
amazon-efs-utils	noarch	1.10-1.amzn2	amzn2-core	20 k
Installing for dependencies:				
stunnel	x86_64	4.56-6.amzn2.0.3	amzn2-core	149 k

Transaction Summary


Install 1 Package (+1 Dependent package)

Total download size: 168 k

Installed size: 350 k

Downloading packages:

(1/2): amazon-efs-utils-1.10-1.amzn2.noarch.rpm	20 kB	00:00
(2/2): stunnel-4.56-6.amzn2.0.3.x86_64.rpm	149 kB	00:00

 root@ip-172-31-10-140:~

```
[root@ip-172-31-10-140 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        475M   0    475M   0% /dev
tmpfs           492M   0    492M   0% /dev/shm
tmpfs           492M 400K   492M   1% /run
tmpfs           492M   0    492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.3G   6.8G  16% /
tmpfs           99M   0     99M   0% /run/user/1000
[root@ip-172-31-10-140 ~]#
```

Do the same in black machine

root@ip-172-31-7-220:~

Dependencies Resolved

Package	Arch	Version
Installing:		
amazon-efs-utils	noarch	1.10-1.amzn2
Installing for dependencies:		
stunnel	x86_64	4.56-6.amzn2.0.3

Transaction Summary

Install 1 Package (+1 Dependent package)

Total download size: 168 k
Installed size: 350 k
Downloading packages:
(1/2): amazon-efs-utils-1.10-1.amzn2.noarch.rpm
(2/2): stunnel-4.56-6.amzn2.0.3.x86_64.rpm

Total

Running transaction check
Running transaction test
Transaction test succeeded

Running transaction

Installing : stunnel-4.56-6.amzn2.0.3.x86_64
Installing : amazon-efs-utils-1.10-1.amzn2.noarch
Verifying : stunnel-4.56-6.amzn2.0.3.x86_64
Verifying : amazon-efs-utils-1.10-1.amzn2.noarch

Installed:

amazon-efs-utils.noarch 0:1.10-1.amzn2

Dependency Installed:

stunnel.x86_64 0:4.56-6.amzn2.0.3

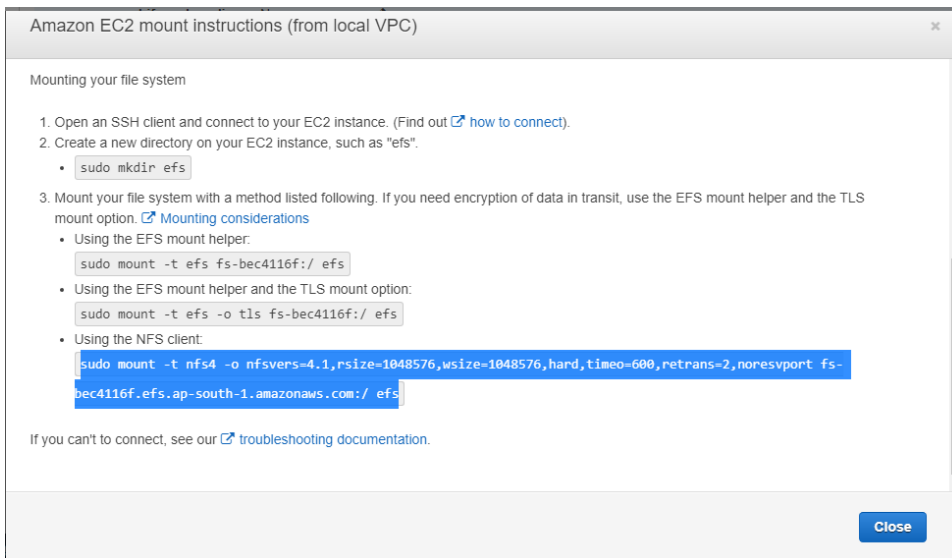
Complete!

root@ip-172-31-7-220:~

```
[root@ip-172-31-7-220 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        475M   0   475M   0% /dev
tmpfs           492M   0   492M   0% /dev/shm
tmpfs           492M 400K   492M   1% /run
tmpfs           492M   0   492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.3G   6.8G  16% /
tmpfs           99M   0    99M   0% /run/user/1000
[root@ip-172-31-7-220 ~]#
```

Step 5:

Copy this



Do in both machines

Before that mkdir efs

```
[root@ip-172-31-10-140 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport fs-bec4116f.efs.ap-south-1.amazonaws.com:/ efs
mount.nfs4: mount point efs does not exist
[root@ip-172-31-10-140 ~]# sudo mkdir efs
[root@ip-172-31-10-140 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport fs-bec4116f.efs.ap-south-1.amazonaws.com:/ efs
[root@ip-172-31-10-140 ~]#
```

```
[root@ip-172-31-10-140 ~]# df -h
Filesystem                                Size  Used Avail Use% Mounted on
devtmpfs                                  475M   0  475M   0% /dev
tmpfs                                      492M   0  492M   0% /dev/shm
tmpfs                                      492M 404K  492M   1% /run
tmpfs                                      492M   0  492M   0% /sys/fs/cgroup
/dev/xvda1                                8.0G  1.3G   6.8G  16% /
tmpfs                                      99M   0   99M   0% /run/user/1000
fs-bec4116f.efs.ap-south-1.amazonaws.com:/ 8.0E   0   8.0E   0% /root/efs
[root@ip-172-31-10-140 ~]#
```

Same in black machine also

```
[root@ip-172-31-7-220 ~]# sudo mkdir efs
[root@ip-172-31-7-220 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport fs-bec4116f.efs.ap-south-1.amazonaws.com:/ efs
[root@ip-172-31-7-220 ~]#
```

```
[root@ip-172-31-7-220 ~]# df -h
Filesystem                                Size  Used Avail Use% Mounted on
devtmpfs                                  475M   0  475M   0% /dev
tmpfs                                      492M   0  492M   0% /dev/shm
tmpfs                                      492M 404K  492M   1% /run
tmpfs                                      492M   0  492M   0% /sys/fs/cgroup
/dev/xvda1                                8.0G  1.3G   6.8G  16% /
tmpfs                                      99M   0   99M   0% /run/user/1000
fs-bec4116f.efs.ap-south-1.amazonaws.com:/ 8.0E   0   8.0E   0% /root/efs
[root@ip-172-31-7-220 ~]#
```

Step 6:

Create a file in white machine

Go to this path

Cd efs

```
[root@ip-172-31-10-140 ~]# cd efs
[root@ip-172-31-10-140 efs]# vi test
[root@ip-172-31-10-140 efs]# cat test
this is my first demo for EFS
[root@ip-172-31-10-140 efs]#
```

```
root@ip-172-31-10-140:~/efs
this is my first demo for EFS
~
~
~
```

Step 7:

Go to black machine

Under

Cd efs

```
[root@ip-172-31-7-220 ~]# cd efs
[root@ip-172-31-7-220 efs]# ls -lrt
total 4
-rw-r--r-- 1 root root 30 Nov 12 03:58 test
[root@ip-172-31-7-220 efs]# cat test
this is my first demo for EFS
[root@ip-172-31-7-220 efs]#
```

Now modify this in black machine

```
root@ip-172-31-7-220:~/efs
this is my first demo for EFS by nishanthan
~
~
~
~
```

```
[root@ip-172-31-7-220 efs]# vi test
[root@ip-172-31-7-220 efs]# cat test
this is my first demo for EFS by nishanthan
[root@ip-172-31-7-220 efs]#
```

Step 8:

Now go to white machine check the file

```
[root@ip-172-31-10-140 ~]# cd efs
[root@ip-172-31-10-140 efs]# vi test
[root@ip-172-31-10-140 efs]# cat test
this is my first demo for EFS
[root@ip-172-31-10-140 efs]# cat test
this is my first demo for EFS by nishanthan
[root@ip-172-31-10-140 efs]#
```

File is changed

Task: Use this in your VPC and SG

PLEASE DELETE AFTER USING