

## Elastic Block Storage

### Step1: Create an Ec2 instance with EBS volumes

Launch an Amazon Linux 2 AMI (HVM), SSD Volume Type - (64-bit Arm)

Change the volume type:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-08be832481aebc7df	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted
EBS	/dev/sdb	Search (case-insensit)	1	General Purpose SSD (gp2)	100 / 3000	N/A	<input type="checkbox"/>	Not Encrypted

Add New Volume

Select the security

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

Security Group ID	Name	Description	Actions
sg-e3e163b9	default	default VPC security group	Copy to new

Create the instance with the extra EBS Volume

### Step2: Connect to the instance using SSH

Check the Volumes :

Lists information about all or the specified block devices. **lsblk**

```
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-23-149 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
└─xvda1     202:1    0   8G  0 part /
xvdb        202:16   0    1G  0 disk
[ec2-user@ip-172-31-23-149 ~]$
[ec2-user@ip-172-31-23-149 ~]$
```

Determine file type -s option causes file to also read argument files which are block or character special files. **sudo file -s /dev/xvda1**

```
[ec2-user@ip-172-31-23-149 ~]$ sudo file -s /dev/xvda1
/dev/xvda1: SGI XFS filesystem data (blksiz 4096, inosz 512, v2 dirs)
[ec2-user@ip-172-31-23-149 ~]$ sudo mkfs -t ext4 /dev/xvdb
mke2fs 1.42.9 (28-Dec-2013)
```

Build a file system. **mkfs -**

Build a Linux file system.

**sudo mkfs -t ext4 /dev/xvdb**

```
[ec2-user@ip-172-31-23-149 ~]$ sudo mkfs -t ext4 /dev/xvdb
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Make a Directory : **sudo mkdir data**

```
[ec2-user@ip-172-31-23-149 ~]$ sudo mkdir data
[ec2-user@ip-172-31-23-149 ~]$ ls
data
```

Mount the file system:

**sudo mount /dev/xvdb data**

```
[ec2-user@ip-172-31-23-149 ~]$ sudo mount /dev/xvdb data
[ec2-user@ip-172-31-23-149 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda         202:0    0   8G  0 disk 
└─xvda1      202:1    0   8G  0 part /
xvdb         202:16   0   1G  0 disk /home/ec2-user/data
[ec2-user@ip-172-31-23-149 ~]$
```

Make a File in new volume : **sudo nano file1.txt**

```
[ec2-user@ip-172-31-23-149 ~]$ vim sample1.txt
[ec2-user@ip-172-31-23-149 ~]$ ls
data sample1.txt
[ec2-user@ip-172-31-23-149 ~]$
```

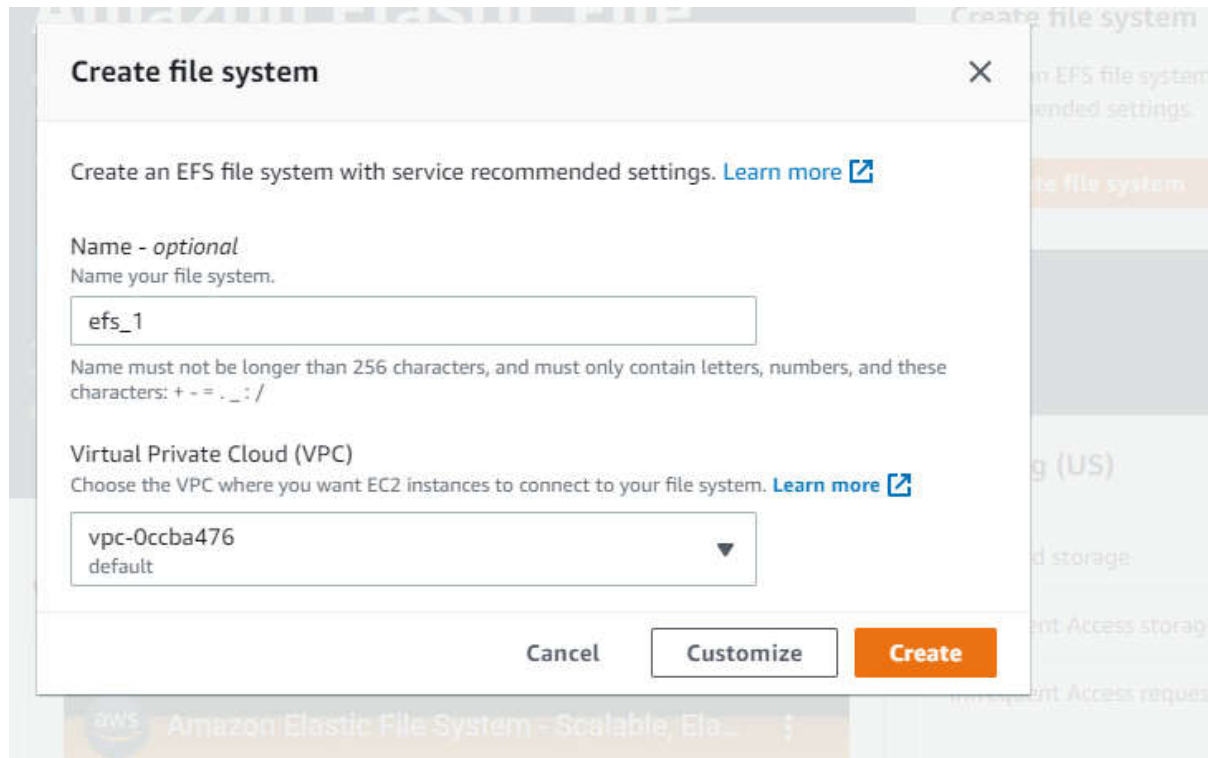
Unmount a file system : **sudo umount -l data**

```
[ec2-user@ip-172-31-23-149 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda         202:0    0   8G  0 disk 
└─xvda1      202:1    0   8G  0 part /
xvdb         202:16   0   1G  0 disk 
[ec2-user@ip-172-31-23-149 ~]$
```

## Creating an EFS

### Step1: Create a file system

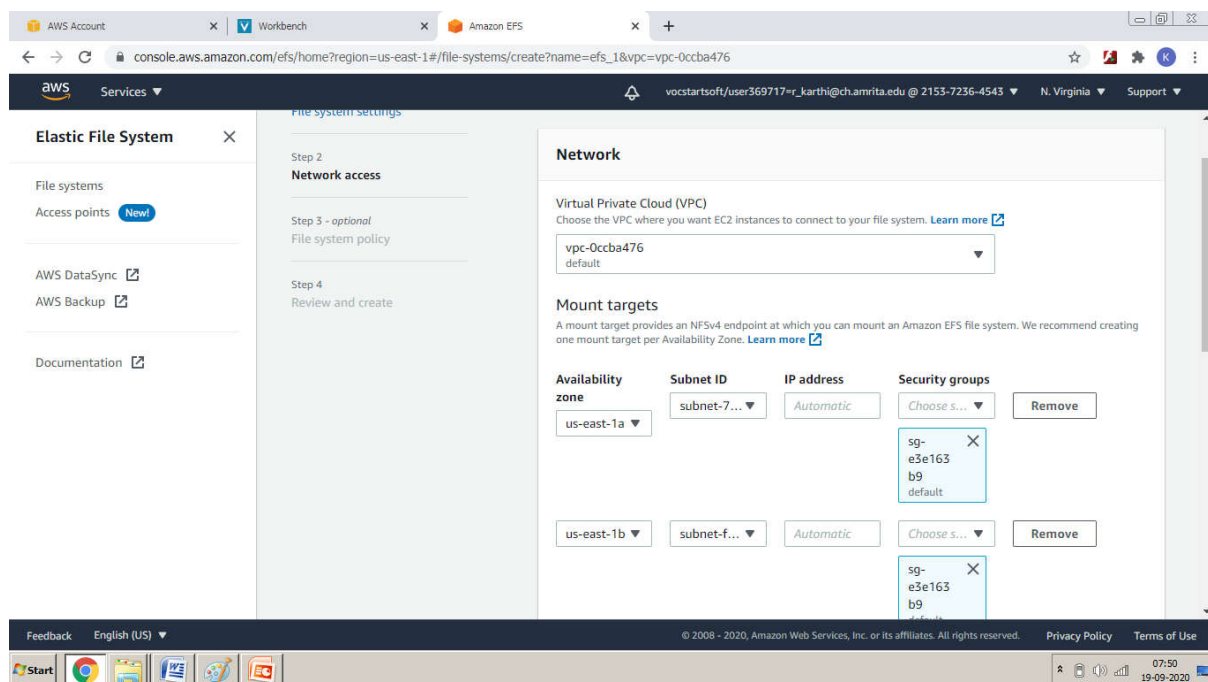
#### Select the default VPC



The screenshot shows the 'Create file system' dialog box in the AWS console. It has a title bar with a close button. The main content area includes a heading 'Create file system' and a sub-heading 'Create an EFS file system with service recommended settings. [Learn more](#)'. Below this, there is a section for 'Name - optional' with the instruction 'Name your file system.' and a text input field containing 'efs\_1'. A note below the input field states: 'Name must not be longer than 256 characters, and must only contain letters, numbers, and these characters: + - = . \_ : /'. The next section is 'Virtual Private Cloud (VPC)' with the instruction 'Choose the VPC where you want EC2 instances to connect to your file system. [Learn more](#)'. Below this is a dropdown menu showing 'vpc-0ccba476' and 'default'. At the bottom of the dialog are three buttons: 'Cancel', 'Customize', and 'Create'.

#### Choose Customise the view the other settings

#### Network Access



The screenshot shows the 'Network' settings page in the AWS console for an EFS file system. The page is titled 'Network' and has a sub-heading 'Virtual Private Cloud (VPC)'. It includes a dropdown menu for 'vpc-0ccba476' and 'default'. Below this is a section for 'Mount targets' with a note: 'A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone. [Learn more](#)'. The 'Mount targets' section has a table with columns: 'Availability zone', 'Subnet ID', 'IP address', and 'Security groups'. The table has two rows. The first row shows 'us-east-1a', 'subnet-7...', 'Automatic', and 'sg-e3e163b9 default'. The second row shows 'us-east-1b', 'subnet-f...', 'Automatic', and 'sg-e3e163b9 default'. Each row has a 'Remove' button. The page also has a left sidebar with 'Elastic File System' and 'File system settings' sections. The bottom of the page shows the AWS footer with '© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and 'Privacy Policy' and 'Terms of Use' links.

## Set all as default and create the EFS

	Name	File system ID	Encrypted	Total size	Size in EFS Standard	Size in EFS IA	Provisioned Throughput (MiB/s)
	efs_1	fs-5543d9d7	Encrypted	6 KiB	6 KiB	0 Bytes	-

## Step2: Create an instance and Connect to EFS

## Launch an Linux instance and set the network to default VPC

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, a dedicated instance, and more.

Number of instances	1	<a href="#">Launch into Auto Scaling Group</a>
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-0ccba476 (default)	<a href="#">Create new VPC</a>
Subnet	No preference (default subnet in any Availability Zone)	<a href="#">Create new subnet</a>
Auto-assign Public IP	Use subnet setting (Enable)	

## Set the security group

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	Anywhere 0.0.0.0, ::/0	e.g. SSH for Admin Desktop
NFS	TCP	2049	Anywhere 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](#)

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## Create the instance

## Step3: Connect to the EFS using ssh

```

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

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-21-112 ~]$ ls
[ec2-user@ip-172-31-21-112 ~]$ ls
```

**Make a directory to mount the EFS drive to the instance**

**mkdir efs2**

```

[ec2-user@ip-172-31-21-112 ~]$ ls
[ec2-user@ip-172-31-21-112 ~]$ mkdir efs2
[ec2-user@ip-172-31-21-112 ~]$ ls
efs2
```

**Get the Mount details from EFS – view details – attach**

File systems (1)							
<div>Filter by property values</div>							
Name	File system ID	Encrypted	Total size	Size in EFS Standard	Size in EFS IA	Provisioned Throughput (MiB/s)	
efs_1	fs-5543d9d7	Encrypted	6 KiB	6 KiB	0 Bytes		

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-5543d9d7:/ efs
```

Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-5543d9d7.efs.us-east-1.amazonaws.com:/ efs
```

**Mount the EFS device to the instance**

```

[ec2-user@ip-172-31-21-112 ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-5543d9d7.efs.us-east-1.amazonaws.com:/ /home/ec2-user/efs2
```

### Make a folder in EFS drive - efsfolder

```
[ec2-user@ip-172-31-21-112 efs2]$ sudo mkdir efsfolder
[ec2-user@ip-172-31-21-112 efs2]$ ls
efsfolder  sample  x.txt
[ec2-user@ip-172-31-21-112 efs2]$ |
```

### Create an other instance and connect to the efs device

```
$ ssh -i "ebs_example.pem" ec2-user@ec2-184-72
-180-5.compute-1.amazonaws.com
load pubkey "ebs_example.pem": invalid format
Last login: Sat Sep 19 02:35:02 2020 from 49.3
7.205.138

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

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 1 packages av
ailable
```

### Make a new folder to mount the efs

```
[ec2-user@ip-172-31-89-56 ~]$ mkdir efsfolder2
```

### Check the folder

```
[ec2-user@ip-172-31-89-56 ~]$ ls
efsfolder2
[ec2-user@ip-172-31-89-56 ~]$ pwd
/home/ec2-user
```

### Mount the EFS to ec2 instance in folder efsfolder2

```
[ec2-user@ip-172-31-89-56 ~]$
[ec2-user@ip-172-31-89-56 ~]$ sudo mount -t nfs4 -o nfsvers=4
.1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,nores
vport fs-5543d9d7.efs.us-east-1.amazonaws.com:/ /home/ec2-us
er/efsfolder2
[ec2-user@ip-172-31-89-56 ~]$ ls
efsfolder2
```

### The same information in ec1 is available for ec2 instance

```
[ec2-user@ip-172-31-89-56 efsfolder2]$ ls
efsfolder sample x.txt
```

In both the EC2 instance the files are available for update.

<pre>er/efsfolder2 [ec2-user@ip-172-31-89-56 ~]\$ ls efsfolder2 [ec2-user@ip-172-31-89-56 ~]\$ cd efsfolder2 [ec2-user@ip-172-31-89-56 efsfolder2]\$ ls efsfolder sample x.txt [ec2-user@ip-172-31-89-56 efsfolder2]\$</pre>	<pre>Run "sudo yum update" to apply all updates. [ec2-user@ip-172-31-21-112 ~]\$ ls efs2 [ec2-user@ip-172-31-21-112 ~]\$ cd efs2 [ec2-user@ip-172-31-21-112 efs2]\$ ls efsfolder sample x.txt [ec2-user@ip-172-31-21-112 efs2]\$</pre>
--	--

- The second floppy drive is named **/dev/fd0**.
- The first hard disk detected is named **/dev/sda**.
- The second hard disk detected is named **/dev/sdb**, and so on.

A new folder is added which is visible in both EC2 instances

<pre>[ec2-user@ip-172-31-89-56 efsfolder2]\$ ls efsfolder sample x.txt [ec2-user@ip-172-31-89-56 efsfolder2]\$ ls efsfolder new sample x.txt [ec2-user@ip-172-31-89-56 efsfolder2]\$</pre>	<pre>[ec2-user@ip-172-31-21-112 efs2]\$ mkdir new mkdir: cannot create directory 'new': Permission denied [ec2-user@ip-172-31-21-112 efs2]\$ sudo mkdir new [ec2-user@ip-172-31-21-112 efs2]\$ ls efsfolder new sample x.txt [ec2-user@ip-172-31-21-112 efs2]\$</pre>
--	---

## GCP using command line

**gcloud auth login**

**gcloud config set project**

**gsutil mb gs://karth**

## for File System

<https://cloud.google.com/filestore/docs/quickstart-console>

## C.4. Device Names in Linux

Linux disks and partition names may be different from other operating systems. You need to know the names that Linux uses when you create and mount partitions. Here's the basic naming scheme:

- The first floppy drive is named **/dev/fd0**.
- The second floppy drive is named **/dev/fd1**.
- The first hard disk detected is named **/dev/sda**.
- The second hard disk detected is named **/dev/sdb**, and so on.



- The first SCSI CD-ROM is named **/dev/scd0**, also known as **/dev/sr0**.

The partitions on each SCSI disk are represented by appending a decimal number to the disk name: **sda1** and **sda2** represent the first and second partitions of the first SCSI disk drive in your system.

Here is a real-life example. Let's assume you have a system with 2 SCSI disks, one at SCSI address 2 and the other at SCSI address 4. The first disk (at address 2) is then named **sda**, and the second **sdb**. If the **sda** drive has 3 partitions on it, these will be named **sda1**, **sda2**, and **sda3**. The same applies to the **sdb** disk and its partitions.

Note that if you have two SCSI host bus adapters (i.e., controllers), the order of the drives can get confusing. The best solution in this case is to watch the boot messages, assuming you know the drive models and/or capacities.

**xvd** means **Xen Virtual Disk** on a Xen Server. **/dev/xvd** is the standard name for **Xen virtual disk**, by analogy with the **hd\*** in IDE and **sd\*** in SCSI. The first virtual disk is **/dev/xvda**, which can be partitioned into **/dev/xvda1** (just as **/dev/sda1** is the first partition of the first SCSI or SCSI-like storage device). In short, treat **xvda1** exactly as you would **sda1** on a regular PC.