**Elastic File System (Amazon EFS)**

* Log in to the AWS Management Console
* Create an Amazon EFS file system
* Log in to an Amazon Elastic Compute Cloud (Amazon EC2) instance that runs Amazon Linux
* Mount your file system to your EC2 instance
* Examine and monitor the performance of your file system

**Prerequisite** : Create VPC with three subnets – private,public in AZ and private in another AZ , create EC2 instance as **EFS Client** in one public subnet with keypair, security group with **port 22** allowed , Route table with route IGW.

## Creating a security group to access your EFS file system

The security group that you associate with a mount target must allow inbound access for **TCP on port 2049 for Network File System (NFS)**. This is the security group that you will now create, configure, and attach to your EFS mount targets.

 In the **AWS Management Console**, on the **Services** menu, choose **EC2**.

 In the navigation pane on the left, choose **Security Groups**.

 Copy the **Security group ID** of the *EFSClient* security group to your text editor.

The Group ID should look similar to *sg-03727965651b6659b*.

 Choose **Create security group** then configure:

* **Security group name:** EFS Mount Target
* **Description:** Inbound NFS access from EFS clients
* **VPC:** *Lab VPC*

 Under the **Inbound rules** section, choose **Add rule** then configure:

* **Type:** *NFS*
* **Source:**
  + *Custom*
  + In the *Custom* box, paste the security group's **Security group ID** that you copied to your text editor
* Choose **Create security group**.

## Creating an EFS file system

EFS file systems can be mounted to multiple EC2 instances that run in different Availability Zones in the same Region. These instances use mount targets that are created in each Availability Zone to mount the file system by using standard NFSv4.1 semantics. You can mount the file system on instances in only one virtual private cloud (VPC) at a time. Both the file system and the VPC must be in the same Region.

* On the **Services** menu, choose **EFS**.
* Choose **Create file system**
* In the **Create file system** window, choose **Customize**
* On **Step 1**:
  + Uncheck Enable automatic backups.
  + **Lifecycle management:** Select *None*
  + In the **Tags** section, configure:
    - **Key:** Name
    - **Value:** My First EFS File System
* Choose **Next**
* For **VPC**, select *Lab VPC*.
* Detach the default security group from each *Availability Zone* mount target by choosing the check box on each default security group.
* Attach the **EFS Mount Target** security group to each Availability Zone mount target (private subnets) by:
* Selecting each **Security groups** check box.
* Choosing **EFS Mount Target**.

A mount target is created for each subnet.

Your mount targets should look like the following example. The diagram shows two mount targets in the **Lab VPC** that use the **EFS Mount Target** security group. In this lab, you should be using the Lab VPC.

* Choose **Next**
* On **Step 3**, choose **Next**
* On **Step 4:**
* Review your configuration.
* Choose **Create**
* created a new EFS file system in your Lab VPC and mount targets in each Lab VPC subnet. In a few seconds, the **File system state** of the file system will change to Available, followed by the mount targets 2–3 minutes later.
* Proceed to the next step after the **Mount target state** for each mount target changes to Available. Choose the screen refresh button after 2–3 minutes to check its progress.
* scroll to the right in the **File systems** pane to find the **File system state**
* Connect to your EFS client - EC2 instance via SSH

## Creating a new directory and mounting the EFS file system

* Amazon EFS supports the NFSv4.1 and NFSv4.0 protocols when it mounts your file systems on EC2 instances. Though NFSv4.0 is supported, we recommend that you use NFSv4.1. When you mount your EFS file system on your EC2 instance, you must also use an NFS client that supports your chosen NFSv4 protocol. The EC2 instance that was launched as a part of this lab includes an NFSv4.1 client, which is already installed on it.
* In your SSH session, make a new directory by entering sudo mkdir efs
* Back in the **AWS Management Console**, on the **Services** menu, choose **EFS**.
* Choose **My First EFS File System**.
* In the **Amazon EFS Console**, on the top right corner of the page, choose **Attach** to open the Amazon EC2 mount instructions.
* Copy the entire command in the **Using the NFS client** section.

The mount command should look similar to this example:

sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-bce57914.efs.us-west-2.amazonaws.com:/ efs

The provided sudo mount... command uses the default Linux mount options.

* In your Linux SSH session, mount your Amazon EFS file system by:
  + Pasting the command
  + Pressing ENTER

1. Get a full summary of the available and used disk space usage by entering:

sudo df -hT

This following screenshot is an example of the output from the following *disk filesystem* command:

df -hT

Notice the *Type* and *Size* of your mounted EFS file system.

* **Eg: fs-1221.efs.us.west-2.amazonaws.com:/ nfs4 8.0E 0 8.0E 0% /home/ec2-user/efs**

### Examining the performance by using Flexible IO

Flexible IO (fio) is a synthetic I/O benchmarking utility for Linux. It is used to benchmark and test Linux I/O subsystems. During boot, fio was automatically installed on your EC2 instance.

* Examine the write performance characteristics of your file system by entering:

sudo fio --name=fio-efs --filesize=10G --filename=./efs/fio-efs-test.img --bs=1M --nrfiles=1 --direct=1 --sync=0 --rw=write --iodepth=200 --ioengine=libaio

fio command will take 5–10 minutes to complete. The output of fio command, specifically the summary status information for this WRITE test.

### Monitoring performance by using Amazon CloudWatch

* In the **AWS Management Console**, on the **Services** menu, choose **CloudWatch**.
* In the navigation pane on the left, choose **Metrics**.
* In the **All metrics** tab, choose **EFS**.
* Choose **File System Metrics**.
* Select the row that has the **PermittedThroughput** Metric Name.

You might need to wait 2–3 minutes and refresh the screen several times before all available metrics, including **PermittedThroughput**, calculate and populate.

* On the graph, choose and drag around the data line. If you do not see the line graph, adjust the time range of the graph to display the period during which you ran the fio command.

The throughput of Amazon EFS scales as the file system grows. File-based workloads are typically spiky. They drive high levels of throughput for short periods of time, and low levels of throughput the rest of the time. Because of this behavior, Amazon EFS is designed to burst to high throughput levels for periods of time. All file systems, regardless of size, can burst to 100 MiB/s of throughput.

* In the **All metrics** tab, uncheck the box for **PermittedThroughput**.
* Select the check box for **DataWriteIOBytes**.

If you do not see DataWriteIOBytes in the list of metrics, use the **File System Metrics** search to find it.

* Choose the **Graphed metrics** tab.
* On the **Statistics** column, select **Sum**.
* On the **Period** column, select **1 Minute**.
* Pause your pointer on the peak of the line graph. Take this number (in bytes) and divide it by the duration in seconds (60 seconds). The result gives you the write throughput (B/s) of your file system during your test.
* The throughput that is available to a file system scales as a file system grows. All file systems deliver a consistent baseline performance of 50 MiB/s per TiB of storage. Also, all file systems (regardless of size) can burst to 100 MiB/s. File systems that are larger than 1T B can burst to 100 MiB/s per TiB of storage. Add data to file system, the maximum throughput that is available to the file system scales linearly and automatically with storage.