

Module 4 - Guided Lab: Introducing Amazon Elastic File System (Amazon EFS)

Lab overview and objectives

This lab introduces you to Amazon Elastic File System (Amazon EFS) by using the AWS Management Console.

After completing this lab, you should be able to:

- 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

Duration

This lab requires approximately **20 minutes** to complete.

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Accessing the AWS Management Console

1. At the top of these instructions, choose Start Lab to launch your lab.

A **Start Lab** panel opens, and it displays the lab status.

ⓘ Tip: If you need more time to complete the lab, restart the timer for the environment by choosing the Start Lab button again.

2. Wait until the **Start Lab** panel displays the message *Lab status: ready*, then close the panel by choosing the **X**.

3. At the top of these instructions, choose AWS.

This action opens the AWS Management Console in a new browser tab. The system automatically logs you in.

⚠ Tip: If a new browser tab does not open, a banner or icon is usually at the top of your browser with the message that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and then choose **Allow pop-ups**.

4. Arrange the **AWS Management Console** tab so that it displays alongside these instructions. Ideally, you will have both browser tabs open at the same time so that you can follow the lab steps more easily.

⚠ Do not change the Region unless specifically instructed to do so.

Task 1: 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.

5. In the **AWS Management Console**, on the **Services** menu, choose **EC2**.
6. In the navigation pane on the left, choose **Security Groups**.
7. Copy the **Security group ID** of the *EFSClient* security group to your text editor.

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

8. Choose **Create security group** then configure:
 - **Security group name:** `EFS Mount Target`
 - **Description:** `Inbound NFS access from EFS clients`
 - **VPC:** *Lab VPC*
9. 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** .

Task 2: 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.

10. On the **Services** menu, choose **EFS**.
11. Choose **Create file system**
12. In the **Create file system** window, choose **Customize**
13. On **Step 1**:
 - Uncheck ☐ Enable automatic backups.
 - **Lifecycle management:** Select *None*
 - In the **Tags** section, configure:
 - **Key:** `Name`
 - **Value:** `My First EFS File System`
14. Choose **Next**
15. For **VPC**, select *Lab VPC*.
16. Detach the default security group from each *Availability Zone* mount target by choosing the **✕** check box on each default security group.
17. Attach the **EFS Mount Target** security group to each *Availability Zone* mount target 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.

	Availability Zone	Subnet	IP address	Security groups
<input checked="" type="checkbox"/>	us-west-2a	subnet-07c695200e91a0682 - Lab VPC Private Subnet 1	Automatic	sg-0a88c746ae4385cd6 - EFS Mount Target
<input checked="" type="checkbox"/>	us-west-2b	subnet-01970d4e8eea29acc - Lab VPC Private Subnet 2	Automatic	sg-0a88c746ae4385cd6 - EFS Mount Target

18. Choose **Next**

19. On **Step 3**, choose **Next**

20. On **Step 4**:

- Review your configuration.
- Choose **Create**

👏 Congratulations! You have 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.

Note: You may need to scroll to the right in the **File systems** pane to find the **File system state**.

Task 3: Connecting to your EC2 instance via SSH

In this task, you will connect to your EC2 instance by using Secure Shell (SSH).

Microsoft Windows users

💬 These instructions are specifically for Microsoft Windows users. If you are using macOS or Linux, [skip to the next section](#).

21. Above these instructions that you are currently reading, choose the **Details** dropdown menu, and then select **Show**

A **Credentials** window opens.

22. Choose the **Download PPK** button and save the **labsuser.ppk** file.

Note: Typically, your browser saves the file to the **Downloads** directory.

23. Note the **EC2PublicIP** address, if it is displayed.

24. Exit the **Details** panel by choosing the **X**.

25. To use SSH to access the EC2 instance, you must use ***PuTTY***. If you do not have PuTTY installed on your computer, [download PuTTY](#).

26. Open **putty.exe**.

27. To keep the PuTTY session open for a longer period of time, configure the PuTTY timeout:

- Choose **Connection**
- **Seconds between keepalives:**

28. Configure your PuTTY session by using the following settings.

- Choose **Session**
- **Host Name (or IP address):** Paste the **EC2PublicIP** for the instance you noted earlier
 - Alternatively, return to the Amazon EC2 console and choose **Instances**
 - Select the instance you want to connect to
 - In the *Description* tab, copy the **IPv4 Public IP** value
- Back in PuTTY, in the **Connection** list, expand **SSH**
- Choose **Auth** and expand **Credentials**

- Under **Private key file for authentication**: Choose **Browse**
- Browse to the *labsuser.ppk* file that you downloaded, select it, and choose **Open**
- Choose **Open** again

29. To trust and connect to the host, choose **Accept**.

30. When you are prompted with **login as**, enter: `ec2-user`

This action connects you to the EC2 instance.

31. Microsoft Windows users: [Choose this link to skip ahead to the next task.](#)

macOS and Linux users

These instructions are specifically for macOS or Linux users. If you are a Windows user, [skip ahead to the next task.](#)

32. Above these instructions that you are currently reading, choose the `Details` dropdown menu, and then select `Show`

A **Credentials** window opens.

33. Choose the **Download PEM** button and save the **labsuser.pem** file.

34. Note the **EC2PublicIP** address, if it is displayed.

35. Exit the **Details** panel by choosing the **X**.

36. Open a terminal window, and change directory to the directory where the *labsuser.pem* file was downloaded by using the `cd` command.

For example, if the *labsuser.pem* file was saved to your **Downloads** directory, run this command:

```
cd ~/Downloads
```

37. Change the permissions on the key to be read-only, by running this command:

```
chmod 400 labsuser.pem
```

38. Run the following command (replace **<public-ip>** with the **EC2PublicIP** address that you copied earlier).

- Alternatively, to find the IP address of the on-premises instance, return to the Amazon EC2 console and select **Instances**
- Select the instance that you want to connect to
- In the **Description** tab, copy the **IPv4 Public IP** value

```
ssh -i labsuser.pem ec2-user@<public-ip>
```

39. When you are prompted to allow the first connection to this remote SSH server, enter `yes`.

Because you are using a key pair for authentication, you are not prompted for a password.

Task 4: Creating a new directory and mounting the EFS file system

i 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.

40. In your SSH session, make a new directory by entering `sudo mkdir efs`

41. Back in the **AWS Management Console**, on the **Services** menu, choose **EFS**.
42. Choose **My First EFS File System**.
43. In the **Amazon EFS Console**, on the top right corner of the page, choose **Attach** to open the Amazon EC2 mount instructions.
44. 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,wsiz=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.

45. In your Linux SSH session, mount your Amazon EFS file system by:

- Pasting the command
- Pressing ENTER

46. 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.

```
[ec2-user@ip-10-0-1-186 ~]$ sudo df -hT
Filesystem                                Type      Size  Used Avail Use% Mounted on
devtmpfs                                 devtmpfs  488M   60K   488M   1% /dev
tmpfs                                     tmpfs     497M    0   497M   0% /dev/shm
/dev/xvda1                               ext4       7.8G  1.2G   6.6G  15% /
fs-531babfa.efs.us-west-2.amazonaws.com:/ nfs4       8.0E    0   8.0E   0% /home/ec2-user/efs
[ec2-user@ip-10-0-1-186 ~]$
```

Task 5: Examining the performance behavior of your new EFS file system

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.

47. 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
```

💬 The `fio` command will take 5–10 minutes to complete. The output should look like the example in the following screenshot. Make sure that you examine the output of your `fio` command, specifically the summary status information for this WRITE test.

```
[ec2-user@ip-10-0-1-205 ~]$ sudo fio --name=fio-efs --filesize=10G --filename=./
efs/fio-efs-test.img --bs=1M --nrfiles=1 --direct=1 --sync=0 --rw=write --iodept
h=200 --ioengine=libaio
fio-efs: (g=0): rw=write, bs=1M-1M/1M-1M/1M-1M, ioengine=libaio, iodepth=200
fio-2.1.5
Starting 1 process
fio-efs: Laying out IO file(s) (1 file(s) / 10240MB)
Jobs: 1 (f=1): [W] [98.1% done] [0KB/0KB/0KB /s] [0/0/0 iops] [eta 00m:11s]
fio-efs: (groupid=0, jobs=1): err= 0: pid=8539: Wed Jan  2 21:53:29 2019
  write: io=10240MB, bw=18085KB/s, iops=17, runt=579800msec
    slat (usec): min=59, max=233, avg=102.74, stdev=17.07
    clat (msec): min=43, max=52587, avg=11321.05, stdev=11796.38
      lat (msec): min=43, max=52587, avg=11321.15, stdev=11796.37
    clat percentiles (msec):
      | 1.00th=[ 1156],  5.00th=[ 2073], 10.00th=[ 2147], 20.00th=[ 2245],
      | 30.00th=[ 2409], 40.00th=[ 2573], 50.00th=[ 2900], 60.00th=[ 5211],
      | 70.00th=[16712], 80.00th=[16712], 90.00th=[16712], 95.00th=[16712],
      | 99.00th=[16712], 99.50th=[16712], 99.90th=[16712], 99.95th=[16712],
      | 99.99th=[16712]
    bw (KB /s): min= 115, max=119635, per=100.00%, avg=19320.80, stdev=26865.56
    lat (msec) : 50=0.01%, 100=0.05%, 250=0.15%, 500=0.21%, 750=0.21%
    lat (msec) : 1000=0.23%, 2000=1.03%, >=2000=98.11%
  cpu          : usr=0.06%, sys=0.14%, ctx=15803, majf=0, minf=9
  IO depths    : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.2%, 32=0.3%, >=64=99.4%
    submit     : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
    complete   : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
    issued    : total=r=0/w=10240/d=0, short=r=0/w=0/d=0
    latency    : target=0, window=0, percentile=100.00%, depth=200

Run status group 0 (all jobs):
  WRITE: io=10240MB, aggrb=18085KB/s, minb=18085KB/s, maxb=18085KB/s, mint=579800msec, maxt=579800msec
[ec2-user@ip-10-0-1-205 ~]$
```

Monitoring performance by using Amazon CloudWatch

48. In the **AWS Management Console**, on the **Services** menu, choose **CloudWatch**.

49. In the navigation pane on the left, choose **Metrics**.

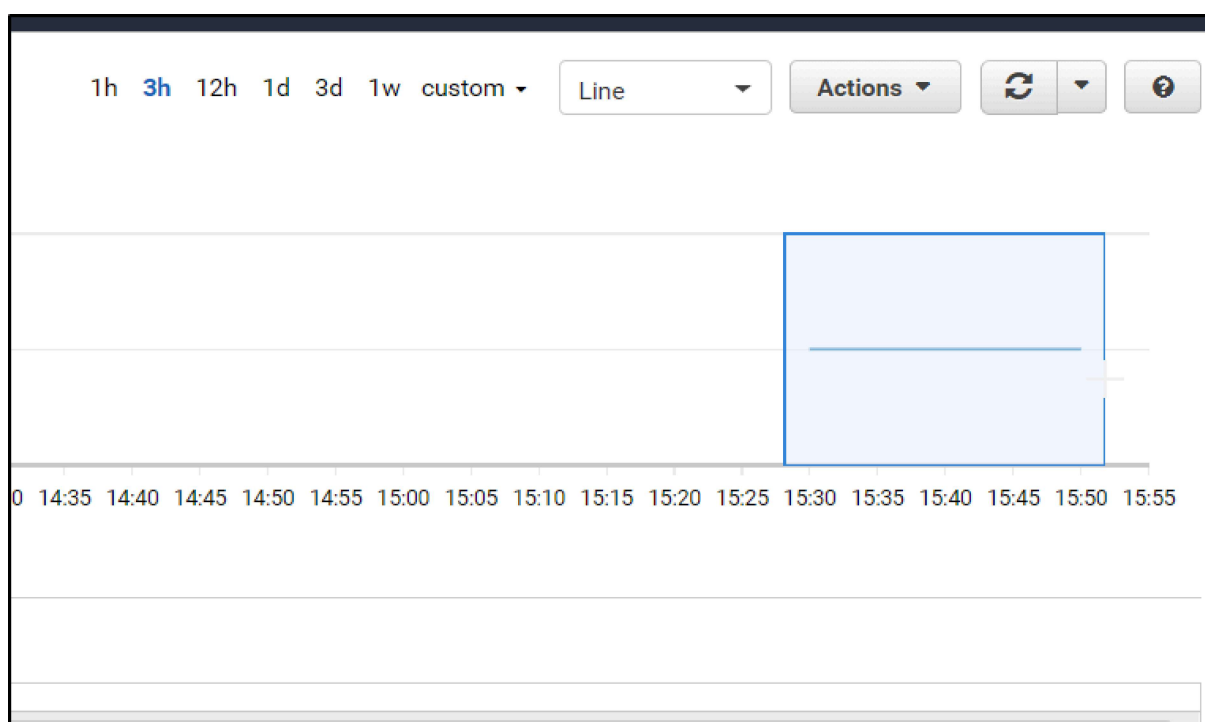
50. In the **All metrics** tab, choose **EFS**.

51. Choose **File System Metrics**.

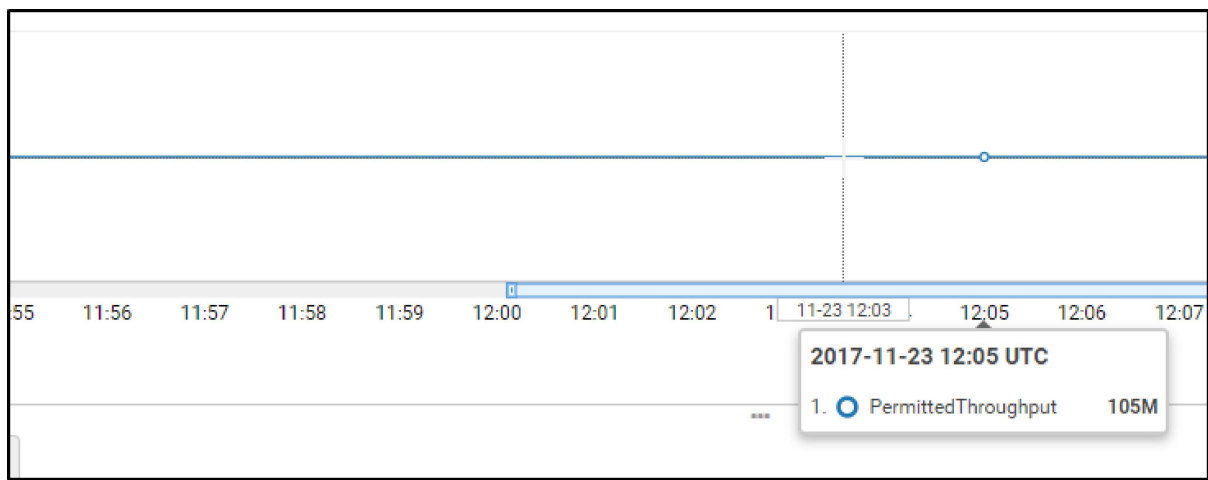
52. Select the row that has the **PermittedThroughput** Metric Name.

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

53. 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.



54. Pause your pointer on the data line in the graph. The value should be **105M**.



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. For more information about performance characteristics of your EFS file system, see the official [Amazon Elastic File System documentation](#).

55. In the **All metrics** tab, *uncheck* the box for **PermittedThroughput**.

56. Select the check box for **DataWriteIOBytes**.

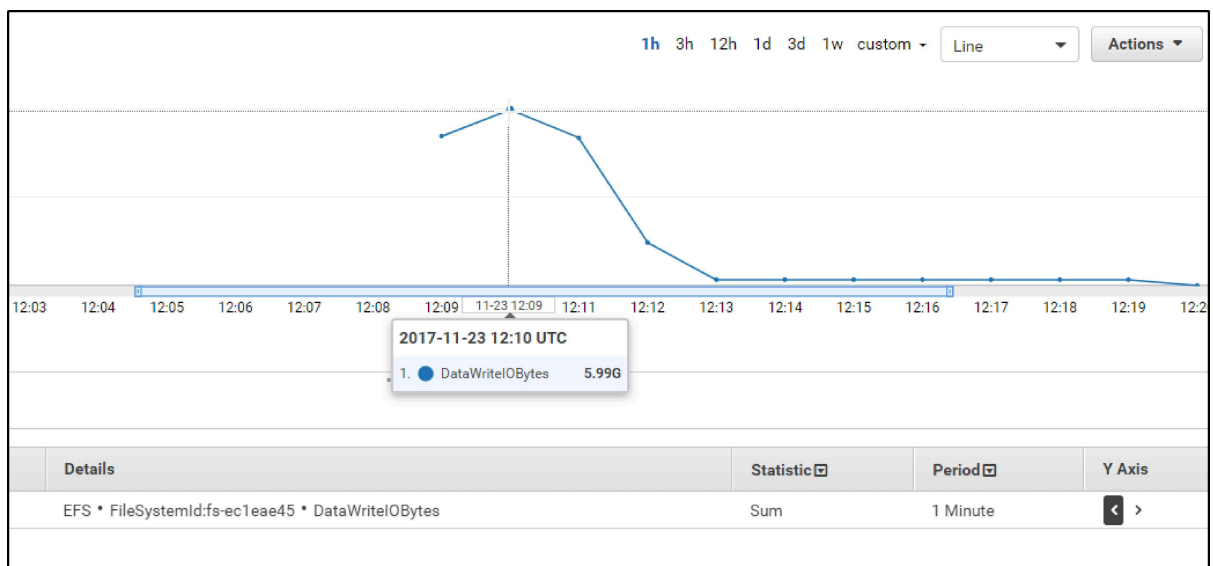
💡 If you do not see *DataWriteIOBytes* in the list of metrics, use the **File System Metrics** search to find it.

57. Choose the **Graphed metrics** tab.

58. On the **Statistics** column, select **Sum**.

59. On the **Period** column, select **1 Minute**.

60. 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. As you add data to your file system, the maximum throughput that is available to the file system scales linearly and automatically with your storage.

File system throughput is shared across all EC2 instances that are connected to a file system. For more information about performance characteristics of your EFS file system, see the official [Amazon Elastic File System documentation](#).

👏 Congratulations! You created an EFS file system, mounted it to an EC2 instance, and ran an I/O benchmark test to examine its performance characteristics.

Submitting your work


61. At the top of these instructions, choose **Submit** to record your progress and when prompted, choose **Yes**.

62. If the results don't display after a couple of minutes, return to the top of these instructions and choose **Grades**

Tip: You can submit your work multiple times. After you change your work, choose **Submit** again. Your last submission is what will be recorded for this lab.

63. To find detailed feedback on your work, choose **Details** followed by ► **View Submission Report**.

Lab complete

 Congratulations! You have completed the lab.

64. Choose **End Lab** at the top of this page, and then select **Yes** to confirm that you want to end the lab.

A panel should appear with this message: *DELETE has been initiated... You may close this message box now.*

65. Select the **X** in the top right corner to close the panel.

©2020 Amazon Web Services, Inc. and its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.