

## Project 7: Introducing Amazon Elastic File System (Amazon EFS)



AWS Elastic File System.

### Group 2

BY / Ahmed Saeed & Engy Maher

#### Access and Configure AWS CLI

1. Open the Lab Environment o Start your lab session as directed.
2. Run the Lab o Initiate the lab session by clicking the "Run Lab" button.
3. Access AWS CLI
  - Navigate to the AWS Details panel.
  - Locate the AWS CLI section and click "Show" to reveal the CLI credentials.

## Task 1: Create a Security Group for EFS Access

### 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 *EFSClients* 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 1.1: Creating a Security Group using AWS CLI

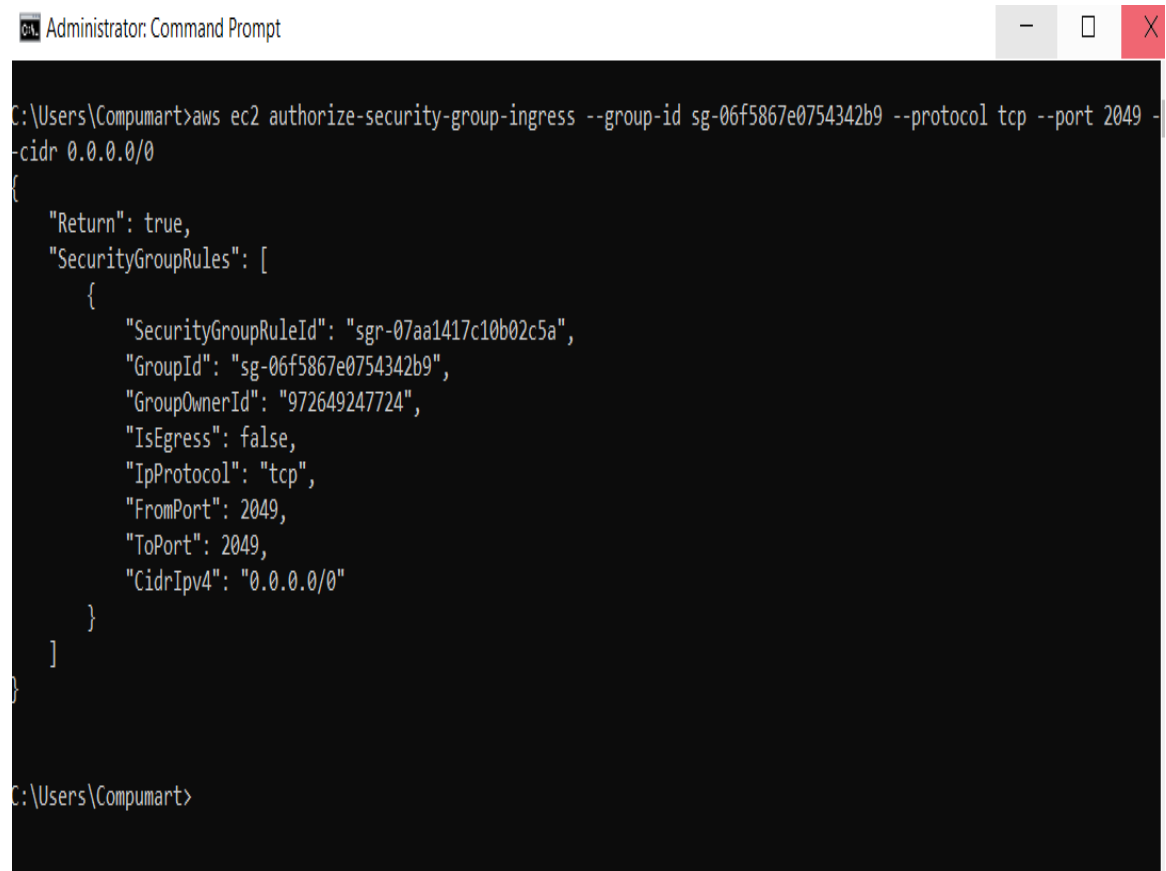
```
Administrator: Command Prompt

Microsoft Windows [Version 10.0.19045.4894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Compumart>aws configure
AWS Access Key ID [*****W0AA]: ASIA6E5TJWPWEWMG37W
AWS Secret Access Key [*****YFrK]: 0tFLhuvhMbRvcVxm0eJRWGSi1+rvBTLn5Qz7/2G
Default region name [us-east-1]: us-east-1
Default output format [json]: json

C:\Users\Compumart>aws ec2 create-security-group --group-name EFS-Mount-Target --description "Security group for EFS Mount Target" --vpc-id vpc-0c7908cfff0a3058
{
  "GroupId": "sg-06f5867e0754342b9"
}
```

## Task 1.2: Adding Inbound Rules to Allow Access via NFS



```
Administrator: Command Prompt

C:\Users\Compumart>aws ec2 authorize-security-group-ingress --group-id sg-06f5867e0754342b9 --protocol tcp --port 2049 -
-cidr 0.0.0.0/0
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-07aa1417c10b02c5a",
      "GroupId": "sg-06f5867e0754342b9",
      "GroupOwnerId": "972649247724",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 2049,
      "ToPort": 2049,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}

C:\Users\Compumart>
```

## Task 2: Creating an Amazon EFS File System

### 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 group
us-east-1a	subnet-0c0f0200000000000 - Lab VPC Private Subnet 1	Automatic IP	sg-0ab00000000000000 - EFS Mount Target
us-east-1b	subnet-01000000000000000 - Lab VPC Private Subnet 2	Automatic IP	sg-0ab00000000000000 - 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**.

```
Administrator: Command Prompt

C:\Users\Compumart>aws efs create-file-system --performance-mode generalPurpose --throughput-mode bursting --encrypted
{
  "OwnerId": "972649247724",
  "CreationToken": "5fff6389-7602-41af-a48f-4c3df1e33895",
  "FileSystemId": "fs-0e2c4ad39e351d769",
  "FileSystemArn": "arn:aws:elasticfilesystem:us-east-1:972649247724:file-system/fs-0e2c4ad39e351d769",
  "CreationTime": "2024-10-06T22:20:13+03:00",
  "LifeCycleState": "creating",
  "NumberOfMountTargets": 0,
  "SizeInBytes": {
    "Value": 0,
    "ValueInIA": 0,
    "ValueInStandard": 0,
    "ValueInArchive": 0
  },
  "PerformanceMode": "generalPurpose",
  "Encrypted": true,
  "KmsKeyId": "arn:aws:kms:us-east-1:972649247724:key/2f92b8c4-6ba2-48b3-be9a-90aba407d578",
  "ThroughputMode": "bursting",
  "Tags": [],
  "FileSystemProtection": {
    "ReplicationOverwriteProtection": "ENABLED"
  }
}

C:\Users\Compumart>
```

## 2.2 Adding Mount Targets for the EFS File System

```
Administrator: Command Prompt
C:\Users\Compumart>aws efs create-mount-target --file-system-id fs-0e2c4ad39e351d769 --subnet-id subnet-049ad398543aaa80
b --security-groups sg-0873e77efc5949439
{
  "OwnerId": "972649247724",
  "MountTargetId": "fsmt-0feedbf378b5c0418",
  "FileSystemId": "fs-0e2c4ad39e351d769",
  "SubnetId": "subnet-049ad398543aaa80b",
  "LifeCycleState": "creating",
  "IpAddress": "10.0.3.186",
  "NetworkInterfaceId": "eni-0cb4aaaac4451e1c4",
  "AvailabilityZoneId": "us-east-1a",
  "AvailabilityZoneName": "us-east-1a",
  "VpcId": "vpc-0c7908cffffd0a3058"
}

C:\Users\Compumart>
```

### Task 3: Connect to the EC2 Instance.

## 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:** 30

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](#).

Connect to the EC2 Instance using Session Manager.





## Task 4: 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.

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,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.

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  100M   60K   100M   1% /dev
tmpfs                                     tmpfs      697M    0   697M   0% /dev/shm
/dev/xvda1                               ext4       7.8G  1.2G  6.6G  15% /
fs-bce57914.efs.us-west-2.amazonaws.com:/ nfs4       8.0E    0   8.0E   0% /home/ec2-user/efs
[ec2-user@ip-10-0-1-186 ~]$
```

```
ec2-user@ip-10-0-1-199:~$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz
e=1048576,hard,timeo=600,retrans=2,noresvport fs-06cd66e4clddada04.efs.us-east-1
.amazonaws.com:/ efs
[ec2-user@ip-10-0-1-199 ~]$ sudo df -hT
Filesystem                                Type      Size  Used Avail Us
et Mounted on
devtmpfs                                 devtmpfs  478M    0  478M
0% /dev
tmpfs                                     tmpfs     486M    0  486M
0% /dev/shm
tmpfs                                     tmpfs     486M  464K  485M
1% /run
tmpfs                                     tmpfs     486M    0  486M
0% /sys/fs/cgroup
/dev/xvda1                               xfs       8.0G  2.2G  5.9G  2
7% /
tmpfs                                     tmpfs     98M    0   98M
0% /run/user/0
tmpfs                                     tmpfs     98M    0   98M
0% /run/user/1000
fs-06cd66e4clddada04.efs.us-east-1.amazonaws.com:/ nfs4    8.0E    0  8.0E
0% /home/ec2-user/efs
[ec2-user@ip-10-0-1-199 ~]$
```

Session ID: Instance ID: i-0af9f675ff86b0e8c  
user3371450=minaakram2002@gmail.com-  
z575kqcvq2korzqht8sj3g5gq

Installing:

amazon-efs-utils

x86\_64

2.0.4-1.amzn2023

amazonlinux

Installing dependencies:

stunnel

x86\_64

5.58-1.amzn2023.0.2

amazonlinux

Transaction Summary

=====

Install 2 Packages

Total download size: 1.5 M

Installed size: 5.5 M

Downloading Packages:

(1/2): amazon-efs-utils-2.0.4-1.amzn2023.x86\_64.rpm

20 MB/s | 1.4 MB 00:00

(2/2): stunnel-5.58-1.amzn2023.0.2.x86\_64.rpm

2.0 MB/s | 156 kB 00:00

-----

Total

13 MB/s | 1.5 MB 00:00

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

Preparing :

Installing : stunnel-5.58-1.amzn2023.0.2.x86\_64

Running scriptlet: stunnel-5.58-1.amzn2023.0.2.x86\_64

Installing : amazon-efs-utils-2.0.4-1.amzn2023.x86\_64

Running scriptlet: amazon-efs-utils-2.0.4-1.amzn2023.x86\_64

Verifying : amazon-efs-utils-2.0.4-1.amzn2023.x86\_64

Verifying : stunnel-5.58-1.amzn2023.0.2.x86\_64

Installed:

amazon-efs-utils-2.0.4-1.amzn2023.x86\_64

stunnel-5.58-1.amzn2023.0.2.x86\_64

Complete!

## Task 5: Examining the Performance of the EFS File System

### 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=200 --filename=/
efs/fio-efs-test.img --bs=1M --xfiles=1 --direct=1 --sync=0 --rw=write --iodepth
=200 --ioengine=libaio
fio-efs: (rw=0): rw=write, bs=1M/1M/1M/1M, ioengine=libaio, iodepth=200
fio-2.1.5
Starting 1 process
fio-efs: Laying out 10 file(s) (1 file(s) / 10240MB)
Jobs: 1 (f=1): [W] [90.1% done] [0MB/0MB/0MB /s] [6/6/0 iops] [eta 00m:11s]
fio-efs: (groupid=0, jobs=1): err= 0: gid=6539: Wed Jan 2 21:53:29 2019
write: io=10240MB, bw=1000562/s, iops=17, runt=579600msec
slat (msec): min=39, max=233, avg=102.74, stdev=17.07
clat (msec): min=43, max=52567, avg=11321.05, stdev=11796.38
lat (msec): min=83, max=52567, avg=11321.15, stdev=11796.37
clat percentiles (msec):
 | 1.00th=[ 1156], 5.00th=[ 2073], 10.00th=[ 2247], 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=19320.60, stdev=26845.56
lat (msec) : 50=0.01%, 100=0.05%, 250=0.15%, 500=0.21%, 750=0.21%
lat (msec) : 1000=0.23%, 2000=1.01%, >=2000=98.11%
cpu        : usr=0.04%, sys=0.14%, ctx=1580, mayf=0, minf=9
io depths  : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.2%, 32=0.2%, >=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=0/0/10240/d=0, short=0/0/0/d=0
latency    : target=0, window=0, percentile=100.00%, depth=200

Run status group 0 (all jobs):
WRITE: io=10240MB, aggrb=1000562/s, mibw=1000562/s, maxb=1000562/s, minb=579600msec, maxt=579600msec
[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.

```
ec2-user@ip-10-0-1-199:~$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz
e=1048576,hard,timeo=600,retrans=2,noresvport fs-06cd66e4clddade04.efs.us-east-1
.amazonaws.com:/ efs
[ec2-user@ip-10-0-1-199 ~]$ sudo df -hT
```

Filesystem	Type	Size	Used	Avail	Us
e% Mounted on					
devtmpfs	devtmpfs	478M	0	478M	
0% /dev					
tmpfs	tmpfs	486M	0	486M	
0% /dev/shm					
tmpfs	tmpfs	486M	464K	485M	
1% /run					
tmpfs	tmpfs	486M	0	486M	
0% /sys/fs/cgroup					
/dev/xvda1	xfs	8.0G	2.2G	5.9G	2
7% /					
tmpfs	tmpfs	98M	0	98M	
0% /run/user/0					
tmpfs	tmpfs	98M	0	98M	
0% /run/user/1000					
fs-06cd66e4clddade04.efs.us-east-1.amazonaws.com:/	nfs4	8.0E	0	8.0E	
0% /home/ec2-user/efs					

```
[ec2-user@ip-10-0-1-199 ~]$ sudo fio --name=fio-efs --filesize=10G --filename=./efs/f
-direct=1 --
fio-efs: (g=0): rw=read, bs=1M-1M/1M-1M/1M-1M, ioengine=psync, iodepth=1
fio-2.14
Starting 1 process
fio-efs: Laying out IO file(s) (1 file(s) / 10240MB)
```

```
[ec2-user@ip-10-0-1-183 ~]$ sudo fio --name=fio-efs --filesize=10G --filename=./efs/fio-efs-test.img --bs=1M --nrfiles=1 --direct=1 --sync=0 --rw=write --iod
ePTH=200 --ioengine=libaio
fio-efs: (g=0): rw=write, bs=(R) 1024KiB-1024KiB, (W) 1024KiB-1024KiB, (T) 1024KiB-1024KiB, ioengine=libaio, iodepth=200
fio-3.32
Starting 1 process
fio-efs: Laying out IO file (1 file / 10240MiB)
Jobs: 1 (f=1): [W(1)][48.2%][w=122MiB/s][w=122 IOPS][eta 00m:44s]
```

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.

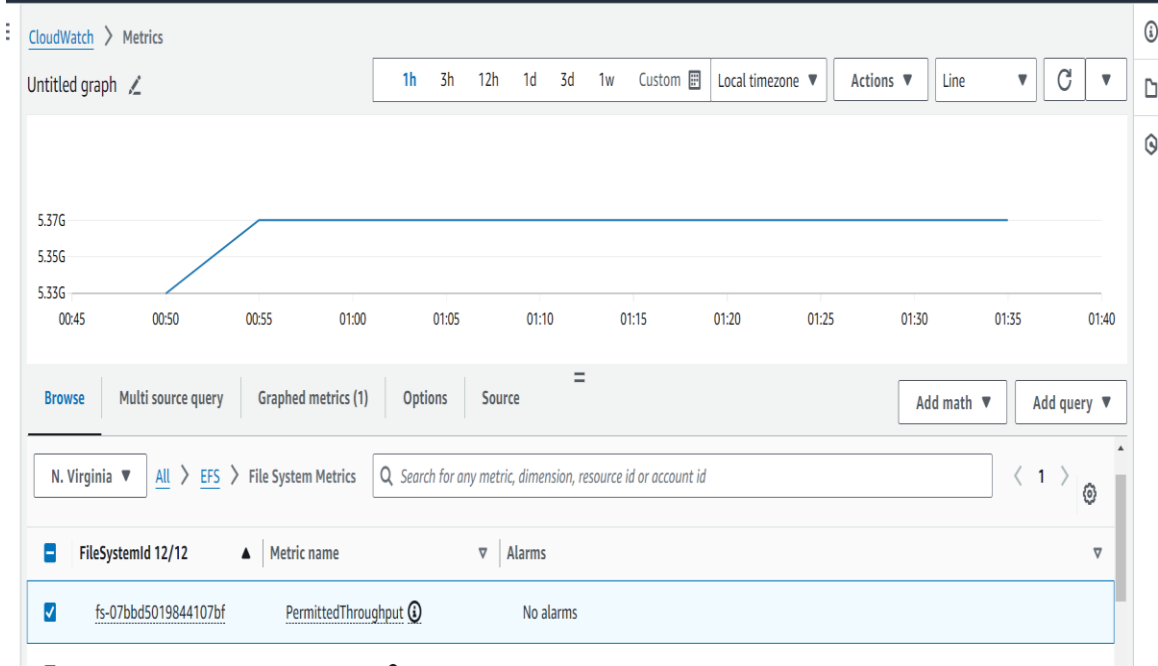
58. On the **Statistics** column, select **Sum**.

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



```
Starting 1 process
fio-efs: Laying out IO file(s) (1 file(s) / 10240MB)
Jobs: 1 (f=1): [R(1)] [100.0% done] [64576KB/0KB/0KB /s] [63/0/0 iops] [eta 00m:00s]
fio-efs: (groupid=0, jobs=1): err= 0: pid=32341: Mon Apr 1 16:44:39 2024
  read: io=10240MB, bw=62739KB/s, iops=61, runt=167133msec
    clat (msec): min=11, max=83, avg=16.31, stdev= 4.26
    lat (msec): min=11, max=83, avg=16.32, stdev= 4.26
    clat percentiles (usec):
      | 1.00th=[12224],  5.00th=[12736], 10.00th=[13120], 20.00th=[13632],
      | 30.00th=[14016], 40.00th=[14528], 50.00th=[15040], 60.00th=[15680],
      | 70.00th=[16768], 80.00th=[18304], 90.00th=[20608], 95.00th=[23680],
      | 99.00th=[35584], 99.50th=[41216], 99.90th=[48384], 99.95th=[49408],
      | 99.99th=[61696]
    lat (msec) : 20=87.53%, 50=12.42%, 100=0.05%
  cpu          : usr=0.07%, sys=0.31%, ctx=10242, majf=0, minf=267
  IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
  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=0.0%
  issued     : total=r=10240/w=0/d=0, short=r=0/w=0/d=0, drop=r=0/w=0/d=0
  latency    : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
  READ: io=10240MB, aggrb=62739KB/s, minb=62739KB/s, maxb=62739KB/s,
  maxt=167133msec
[ec2-user@ip-10-0-1-199 ~]$ sync=0 --rw=write --iodepth=200 --ioe
```



[▶ Start Lab](#)   [■ End Lab](#)   [i AWS Details](#)   [i Details](#)   [✕](#)

Submit

Submission Report

Grades

Total score

15/15

[Task 1] Security Group created

5/5

[Task 2] EFS file system created

5/5

[Task 5] Flexible IO was run

5/5