Introduction to Amazon Elastic File System (EFS)
SPL-151 - Version 2.0

Lab Overview

Overview

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

Topics covered

This lab demonstrates:

- Log into the AWS Management Console
- Create an Amazon EFS file system
- Log into an Amazon Linux EC2 instance
- Mount your file system to your instance
- Examine and monitor the performance of your file system

Other AWS Services

Other AWS Services than the ones needed for this lab are disabled by IAM policy during your access time in this lab. In addition, the capabilities of the services used in this lab are limited to what's required by the lab and in some cases are even further limited as an intentional aspect of the lab design. Expect errors when

accessing other services or performing actions beyond those provided in this lab guide.

What Is Amazon EFS?

Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud. Amazon EFS is easy to use and offers a simple interface that allows you to create and configure file systems quickly and easily. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

When mounted to Amazon EC2 instances, an Amazon EFS file system provides a standard file system interface and file system access semantics, allowing you to seamlessly integrate Amazon EFS with your existing applications and tools. Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, allowing Amazon EFS to provide a common data source for workloads and applications running on more than one Amazon EC2 instance.

It’s designed for high availability and durability, and provides performance for a broad spectrum of workloads and applications, including Big Data and analytics, media processing workflows, content management, web serving, and home directories.

What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

What is Amazon CloudWatch?

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources. Amazon CloudWatch can monitor AWS resources such as Amazon EFS, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to react and keep your application running smoothly. For further information, see the official Amazon Web Services Documentation for CloudWatch at https://aws.amazon.com/documentation/cloudwatch/.

Task 1: Create a Security Group to Access your Amazon EFS File System

Both an EC2 instance and a mount target have associated security groups. These security groups act as a virtual firewall that controls the traffic between them. To enable traffic between an EC2 instance and a mount target (and thus the file system), you must configure the following rules in two security groups:

- Each EC2 instance that mounts the file system must have a security group that allows outbound access to the mount target on TCP port 2049. The

EC2 instance launched when you started the lab has a security group that satisfies this requirements. Remember, security groups are stateful so responses to allowed inbound traffic are allowed to flow outbound regardless of outbound rules, and vice versa.
    - The security group you associate with a mount target must allow inbound access for the TCP protocol on port 2049 for NFS from all EC2 instances on which you want to mount the file system. This is the security group you will create and configure now, and attach it to your EFS mount targets.
- In the **AWS Management Console**, on the **Services** menu, click **EC2**.
- In the navigation pane on the left, click **Security Groups**.
- Click **Create Security Group**.
- In the **Create Security Group** window, configure the following:
    - For **Security group name**, enter
    - For **Description**, enter
    - For **VPC**, select **Lab VPC**.
- Below the **Inbound** tab, configure the following:
    - Click **Add Rule**.
    - For **Type**, select **NFS**.
    - Click the **CIDR, IP or Security Group** box.
    - In the **CIDR, IP or Security Group**, enter

Auto-complete will display a list of security groups.

If you don't see any security groups appear when you enter , click **Cancel** and repeat these steps. The security groups might not be displaying because the CloudFormation template lab may have not had enough time to create them.

- Click the security group that contains **EFSClientSecurityGroup**.
- Click **Create**.

Task 2: Create an Amazon EFS File System

EFS file systems can be mounted to multiple Amazon EC2 instances running in different Availability Zones within the same region. These instances use *mount targets* created in each *Availability Zone*to mount the file system using standard Network File System v4.1 (NFSv4.1) semantics. You can mount the file system on instances in only one Amazon Virtual Private Cloud (Amazon VPC) at a time, and both the file system and VPC must be in the same region.

- On the **Services** menu, click **EFS**.
- Click **Create file system**.
- On **Step 1: Configure file system access**, for **VPC**, select **Lab VPC**.

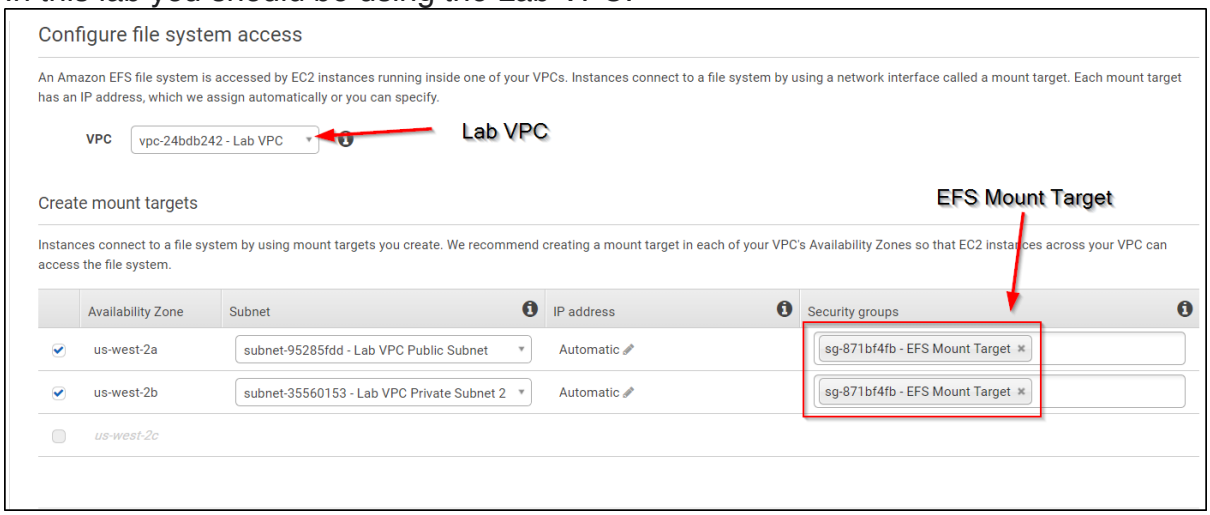
A mount target will be created for each *Lab VPC* subnet in the *Lab* VPC.
Next, you will need to change the **security groups** so that all Mount Targets can use the **EFS Mount Target**.

- Detach the default security group from each mount target by clicking the on each default security group.

This will remove the security groups from the mount targets.

- Attach the **EFS Mount Target** security group to each *Availability Zone* mount target by:
  - Clicking in each **Select Security Group(s)** box.
  - Selecting the **EFS Mount Target** security group.

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



- Click **Next Step**.

Next, you'll add an optional tag to your Amazon EFS file system. Tags are case-sensitive key-value pairs that help describe your file system.

- On **Step 2: Configure optional settings**, configure the following:
  - In the **Add tags** section, in the **Value** field of the **Name** attribute, enter
  - In the **Choose performance mode** section, keep the defaults.

When creating a file system, you also choose a performance mode. There are two performance modes to choose from: **General Purpose** and **Max I/O**. For the majority of use cases, we recommend that you use the *General Purpose* performance mode for your file system.

  - Click **Next Step**.
- On **Step 3: Review and create**, configure the following:
  - Review your configuration.
  - Click **Create File System**.

Congratulations! You have created a new Amazon EFS File System in your Lab VPC and mount targets in each Lab VPC subnet. Within a few seconds the **Life cycle state** of the file system will change to **Available**, followed by the mount targets 2-3 minutes later. Proceed to the next step once the **Life cycle state** for each mount target changes to **Available**. Click the screen refresh button after 2-3 minutes to check on it's progress.

Next you will mount the EFS file system to your EC2 instance and evaluate the performance of your new EFS file system.

Task 3: Connect to Your Amazon EC2 Instance Via SSH.

In this task, you will connect to your Amazon EC2 Instance using SSH. All Amazon EC2 instances are assigned two IP addresses at launch: a *private IP address* (RFC 1918) and a *public IP address*that are directly mapped to each other through Network Address Translation (NAT). Private IP addresses are only reachable from within the Amazon EC2 network. Public addresses are reachable from the Internet. Amazon EC2 also provides an *internal DNS name* and a *public DNS name* that map to the private and public IP addresses, respectively. The internal DNS name can only be resolved within Amazon EC2. The public DNS name resolves to the public IP address outside the Amazon EC2 network and to the private IP address within the Amazon EC2 network.

Windows Users: Using SSH to Connect

These instructions are for Windows users only.
If you are using Mac or Linux, skip to the next section.

- To the left of the instructions you are currently reading, click **Download PPK**.
- Save the file to the directory of your choice.

You will use PuTTY to SSH to Amazon EC2 instances.

If you do not have PuTTY installed on your computer, download it here.

- Open PuTTY.exe

- Configure the PuTTY to not timeout:

    - Click **Connection**
    - Set **Seconds between keepalives** to

This allows you to keep the PuTTY session open for a longer period of time.

- Configure your PuTTY session:
    - Click **Session**
    - **Host name:** Copy and paste the **Ec2IpAddress** shown to the left of these instructions
    - In the **Connection** list, expand **SSH**
    - Click **Auth** (don't expand it)
    - Click **Browse**
    - Browse to and select the PPK file that you downloaded
    - Click **Open** to select it
    - Click **Open**
- Click **Yes**, to trust the host and connect to it.
- When prompted for a **username**, enter:

This will connect to your EC2 instance.

23. Windows Users: Click here to skip ahead to the next task.

Mac and Linux Users

These instructions are for Mac/Linux users only. If you are a Windows user, skip ahead to the next task.

- To the left of the instructions you are currently reading, click **Download PEM**.
- Save the file to the directory of your choice.

- Copy this command to a text editor:

```
chmod 400 KEYPAIR.pem

ssh -i KEYPAIR.pem ec2-user@Ec2IpAddress
```

- Replace *KEYPAIR.pem* with the path to the PEM file you downloaded.
- Replace *Ec2IpAddress* with the values of Ec2IpAddress shown to the left of these instructions.
- Paste the command into the Terminal window and run it.

- Type when prompted to allow a first connection to this remote SSH server.

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

Task 4: Create a New Directory And Mount The EFS File System

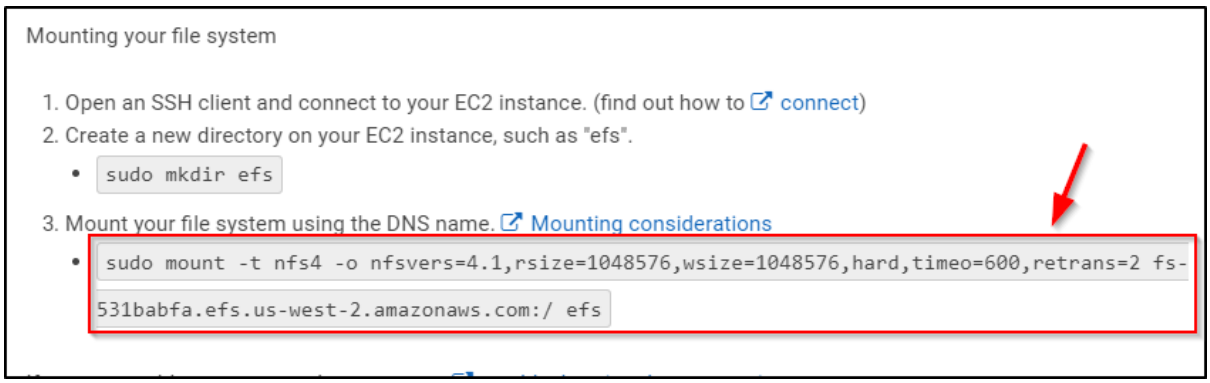
Amazon EFS supports the Network File System version 4.1 (NFSv4.1) and NFSv4.0 protocols when mounting your file systems on Amazon EC2 instances. While NFSv4.0 is supported, we recommend that you use NFSv4.1. Mounting your Amazon EFS file system on your Amazon EC2 instance also requires an NFS client that supports your chosen NFSv4 protocol. The EC2 instance launched as a part of this lab has the NFSv4.1 client already installed.

- In your SSH session, make a new directory by entering

- In the **AWS Management Console**, click **Amazon EC2 mount instructions**.

This link is located below the EFS **File system access** section.

- Scroll down to the **Mounting your file system** section.
- Copy the entire **sudo mount...** command.



The provided "**sudo mount...**" command uses the default Linux mount options. For more information about mounting your EFS file system, see http://docs.aws.amazon.com/efs/latest/ug/mounting-fs-mount-cmd-general.html.

- In your Linux SSH session, mount your Amazon EFS file system by:
  - Pasting the entire sudo mount... command.
  - Pressing enter to run the command.

This is an example of running the sudo mount command.

```
[ec2-user@ip-10-0-1-186 ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz
e=1048576,hard,timeo=600,retrans=2 fs-531babfa.efs.us-west-2.amazonaws.com:/ efs
[ec2-user@ip-10-0-1-186 ~]$
```

- Get a full summary of available and used disk space usage by entering

This is an example of the output from the *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: Examine The Performance Behavior Of Your New EFS File System

Examine Performance Using Flexible IO

Flexible IO (fio) is a synthetic IO benchmarking utility for Linux which is used to benchmark and test Linux IO subsystems. *Fio* was automatically installed on your EC2 instance during boot.

- 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 a 3-5 minutes to complete and the output should look something like the screenshot below. Please examine the output of your *fio* command, specifically the summary status information for this WRITE test.

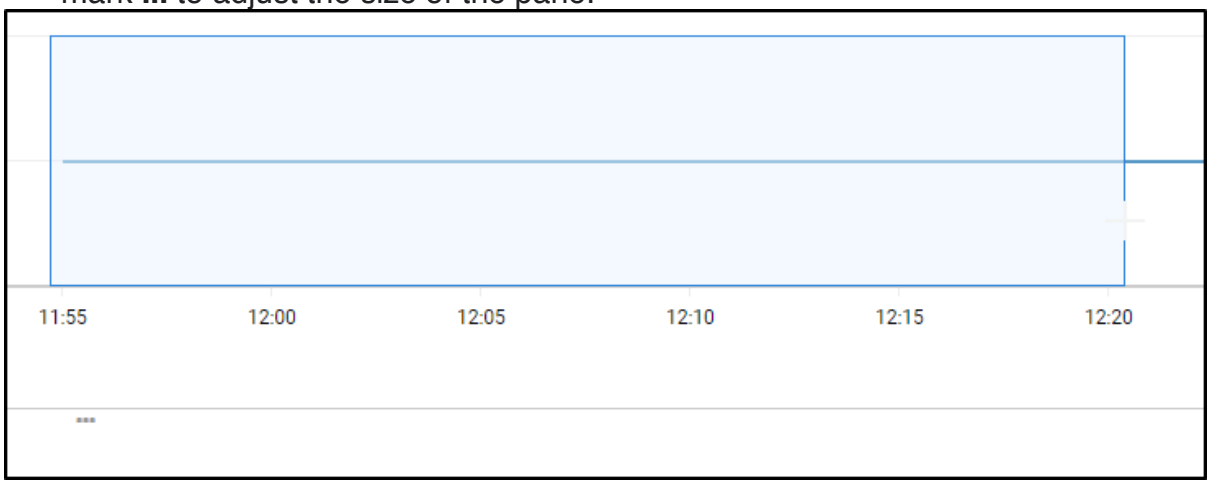
```
[ec2-user@ip-10-0-1-221 ~]$ sudo fio --name=fio-efs --filesize=10G --filename=./efs/fio-ef
fio-efs: (g=0): rw=write, bs=1M-1M/1M-1M/1M-1M, ioengine=libaio, iodepth=200
fio-2.1.5
Starting 1 process
fio-efs: Laying out IO file(s) (1 file(s) / 10240MB)
Jobs: 1 (f=1): [W] [98.1% done] [0KB/0KB/0KB /s] [0/0/0 iops] [eta 00m:11s]
fio-efs: (groupid=0, jobs=1): err= 0: pid=8324: Thu Nov 23 12:20:35 2017
  write: io=10240MB, bw=18095KB/s, iops=17, runt=579482msec
    slat (usec): min=52, max=447, avg=90.56, stdev=16.64
    clat (msec): min=31, max=52603, avg=11314.87, stdev=11825.49
     lat (msec): min=31, max=52603, avg=11314.96, stdev=11825.49
    clat percentiles (msec):
     |  1.00th=[ 1172],  5.00th=[ 2147], 10.00th=[ 2212], 20.00th=[ 2376],
     | 30.00th=[ 2474], 40.00th=[ 2671], 50.00th=[ 2933], 60.00th=[ 3785],
     | 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=  153, max=111524, per=100.00%, avg=19004.40, stdev=26013.56
    lat (msec) : 50=0.01%, 100=0.04%, 250=0.09%, 500=0.23%, 750=0.23%
    lat (msec) : 1000=0.23%, 2000=0.90%, >=2000=98.26%
  cpu          : usr=0.04%, sys=0.14%, ctx=17143, 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=18095KB/s, minb=18095KB/s, maxb=18095KB/s, mint=579482msec, max
[ec2-user@ip-10-0-1-221 ~]$
```

Monitor Performance Using Amazon CloudWatch

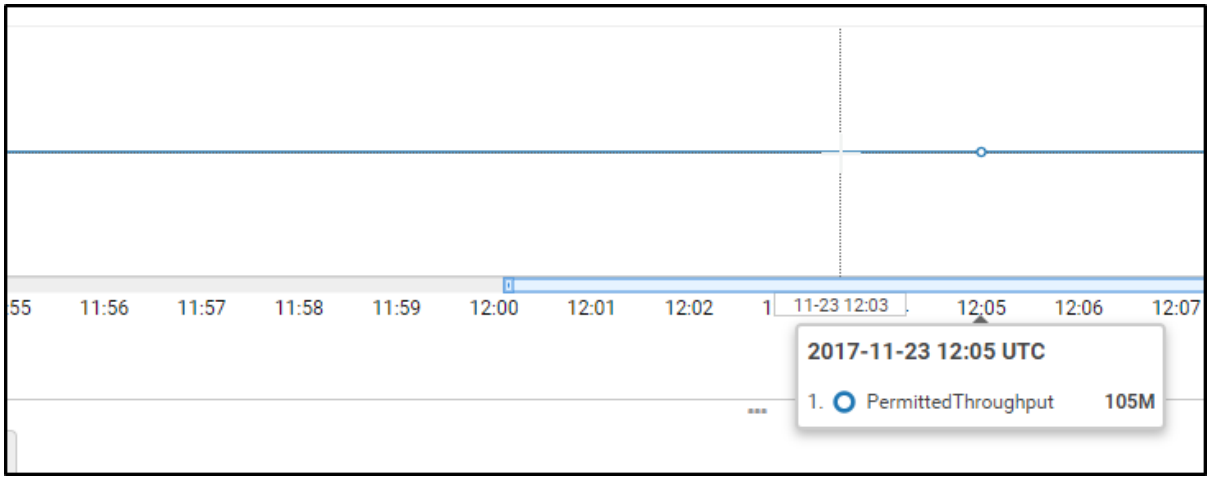
- In the **AWS Management Console**, on the **Services** menu, click **CloudWatch**.
- In the navigation pane on the left, click **Metrics**.
- In the **All metrics** tab, click **EFS**.
- Click **File System Metrics**.
- Check the **FileSystemID** for **PermittedThroughput**.

you may need to wait 2-3 minutes and refresh the screen several times for all the available metrics, including **PermittedThroughput**, to calculate and populate.

- On the graph above, click and drag (up or down) the line just above the elipsis mark **...** to adjust the size of the pane.

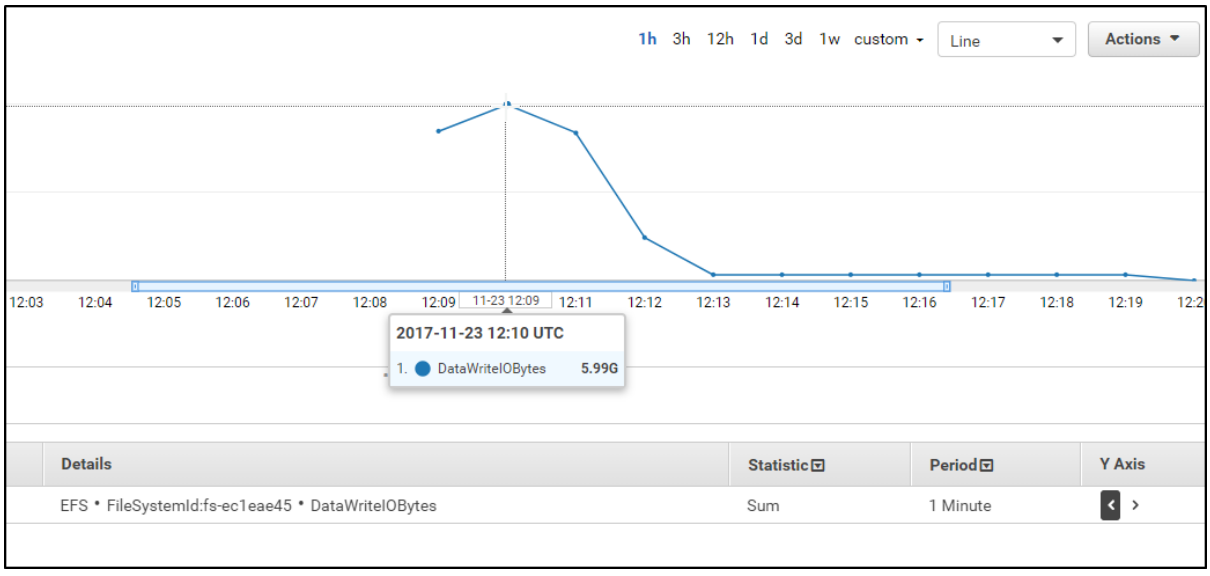


44. Hover your mouse over the data line in the graph. The value should be 105M.



Throughput of Amazon EFS scales as the file system grows. Because file-based workloads are typically spiky, driving high levels of throughput for short periods of time and low levels of throughput the rest of the time, 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 http://docs.aws.amazon.com/efs/latest/ug/performance.html.

- Uncheck the **FileSystemID** for **PermittedThroughput**.
- Check the **FileSystemID** for **DataWriteIOBytes**.
- Click the **Graphed metrics** tab.
- On the **Statistics** column, select **Sum**.
- On the **Period** column, select **1 Minute**.
- Hover over the peak of the line graph. Take this number (in bytes) and divide it by the duration in seconds (60 seconds). This will give you the write throughput (B/s) of your file system during your test.



The throughput 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

and all file systems (regardless of size) can burst to 100 MiB/s. File systems larger than 1TB can burst to 100 MiB/s per TiB of storage. As you add data to your file system, the maximum throughput available to the file system scales linearly and automatically with your storage.

File system throughput is shared across all Amazon EC2 instances connected to a file system. For example, a 1 TiB file system that can burst to 100 MiB/s of throughput can drive 100 MiB/s from a single Amazon EC2 instance, or 10 Amazon EC2 instances can collectively drive 100 MiB/s. For more information about performance characteristics of your EFS file system, see http://docs.aws.amazon.com/efs/latest/ug/performance.html.
Congratulations! You have created an Amazon EFS file system, mounted it to an Amazon EC2 instance, and have run a IO benchmark test to examine its performance characteristics.

End Lab