Introduction to Amazon EFS

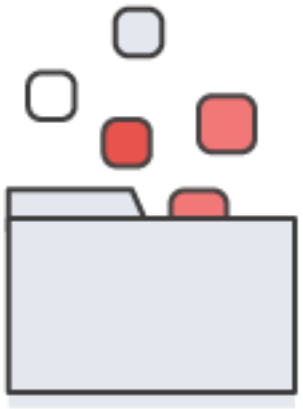
Edward Naim, Head of Product

August 11, 2016

Goals and expectations for this session

- Overall goal: Introduce you to Amazon EFS (what it is, its features, how it can help you)
- Session intended for all levels: We’ll cover both beginner topics and more advanced concepts
- We’ll do Q&A at the end

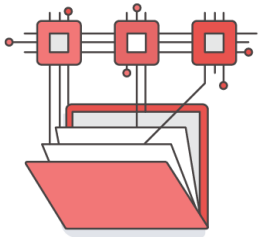
Agenda



1. Provide overview of EFS
2. Introduce EFS technical concepts
3. Walk through creating a file system
4. Review file system security mechanisms
5. Discuss EFS availability and durability properties
6. Share key performance characteristics

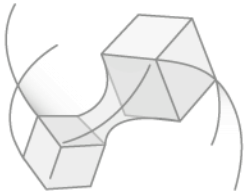
Overview of Amazon EFS

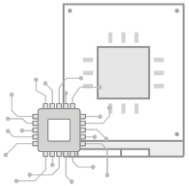
The AWS storage platform



Amazon EFS

File

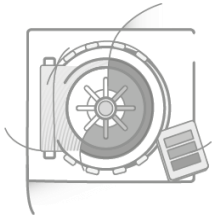




Amazon EBS

Amazon EC2 Instance Store

Block

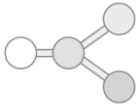


Amazon S3

Amazon Glacier

Object

Data Transfer



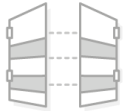
Internet/VPN

AWS Direct Connect

Amazon CloudFront

S3 Transfer Acceleration

ISV Connectors



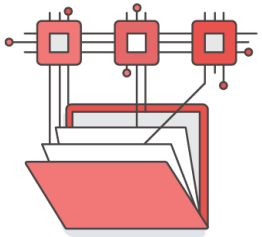
Storage Gateway

AWS Snowball

Amazon Kinesis Firehose

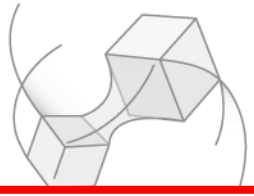
The AWS storage platform

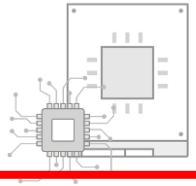
The AWS storage platform



Amazon EFS

File

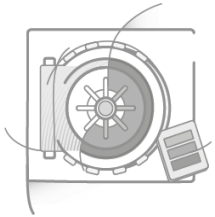




Amazon EBS

Amazon EC2 Instance Store

Block

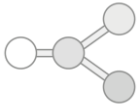


Amazon S3

Amazon Glacier

Object

Data Transfer



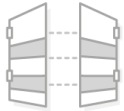
Internet/VPN

AWS Direct Connect

Amazon CloudFront

S3 Transfer Acceleration

ISV Connectors

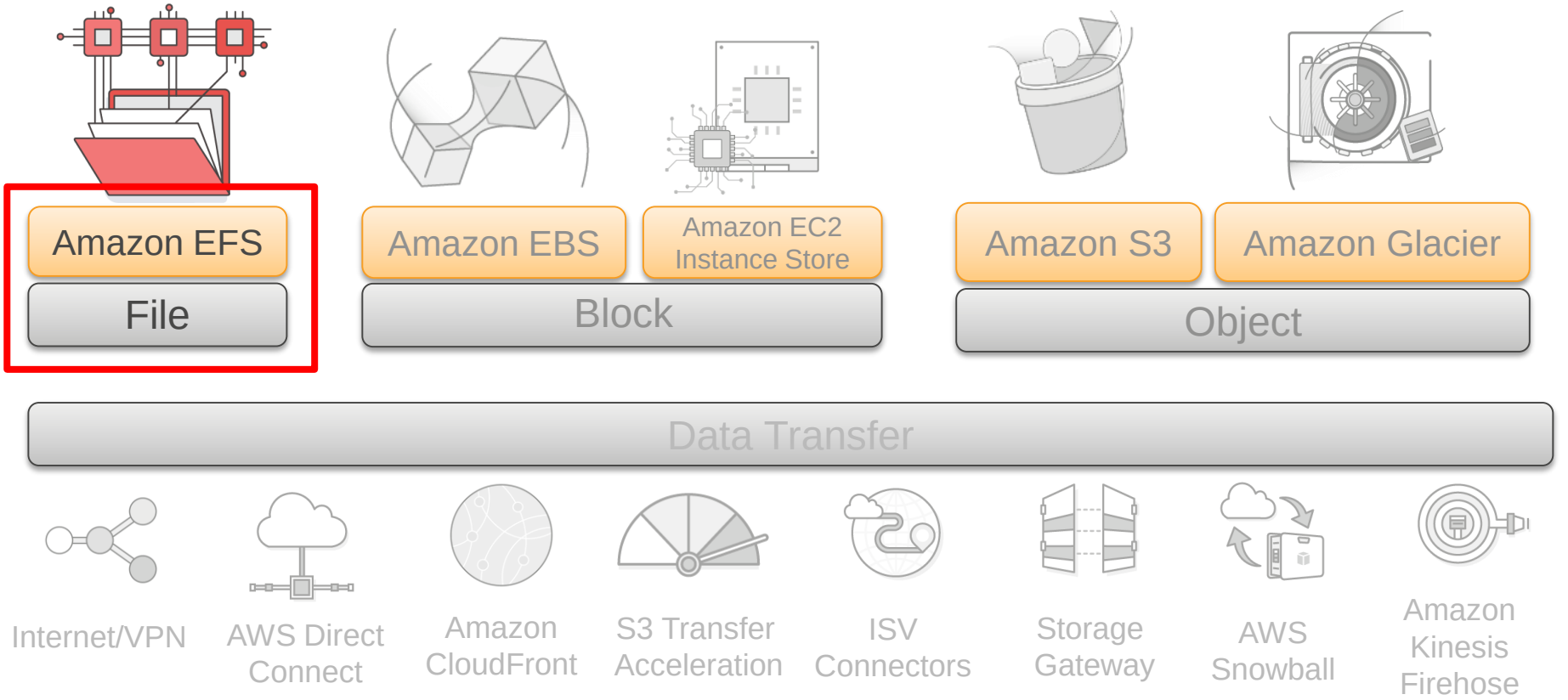


Storage Gateway

AWS Snowball

Amazon Kinesis Firehose

The AWS storage platform



What is Amazon EFS?

- A fully managed file system for Amazon EC2 instances
- Exposes a file system interface that works with standard operating system APIs
- Provides file system access semantics (consistency, locking)
- Sharable across thousands of instances
- Designed to grow elastically to petabyte scale
- Built for performance across a wide variety of workloads
- Highly available and durable

Operating file storage on-prem today is a pain

IT administrator

- Estimate demand
- Procure hardware
- Set aside physical space
- Set up and maintain hardware (and network)
- Manage access and security

Operating file storage on-prem today is a pain

IT administrator

- Estimate demand
- Procure hardware
- Set aside physical space
- Set up and maintain hardware (and network)
- Manage access and security

Application owner or developer

- Provide demand forecasts/business case
- Add lead times and extra coordination to your schedule
- Limit your flexibility and agility

Operating file storage on-prem today is a pain

IT administrator

- Estimate demand
- Procure hardware
- Set aside physical space
- Set up and maintain hardware (and network)
- Manage access and security

Application owner or developer

- Provide demand forecasts/business case
- Add lead times and extra coordination to your schedule
- Limit your flexibility and agility

Business owner

- Make up-front capital investments, over-buy, stay on a constant upgrade/refresh cycle
- Sacrifice business agility
- Distract your people from your business's mission

Building your own on the cloud is too much work and is expensive

Replicate EBS volumes (1 per EC2 instance)

- Substantial management overhead (sync data, provision and manage volumes)
- Costly (one volume per instance)

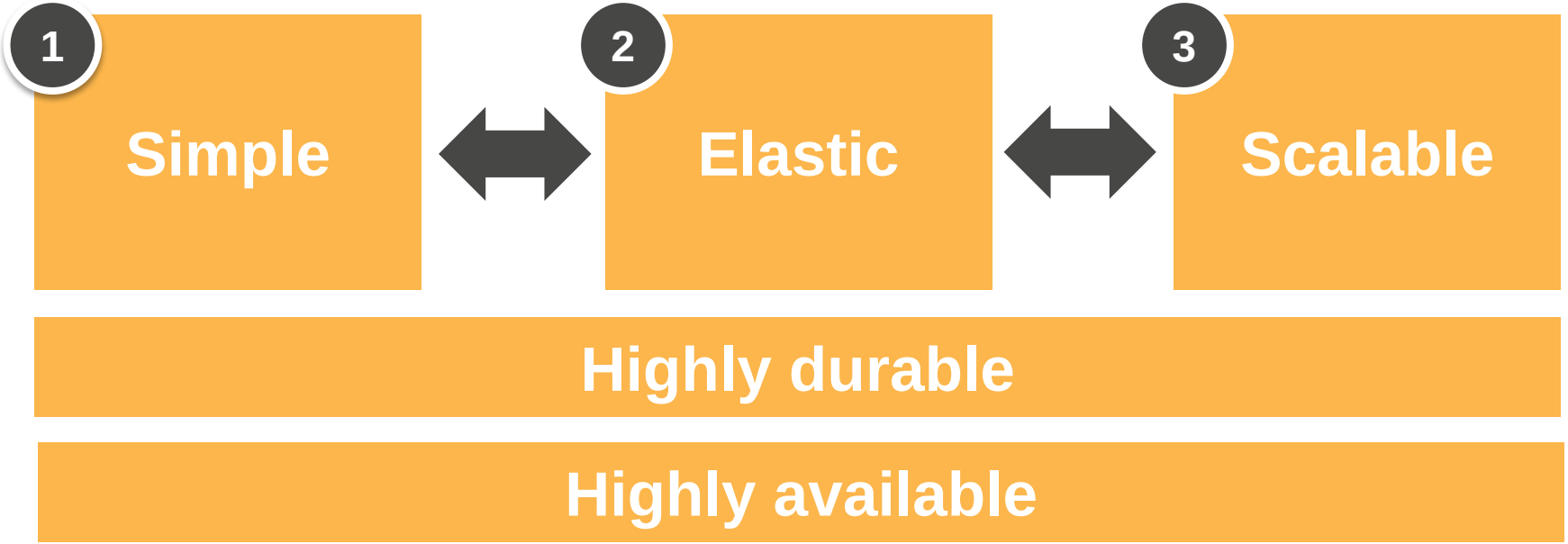
Use a shared file layer

- Complex to set up and maintain
- Scale challenges
- Costly (compute + storage)

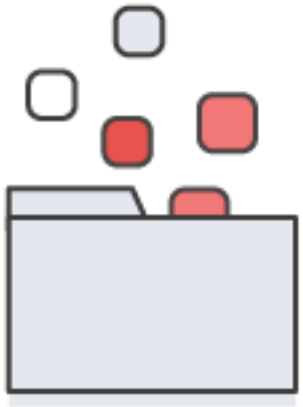
Amazon EFS is useful even for access from a single EC2 instance

- Multi-AZ availability/durability
- Elastically grows – create it and forget about it
- Can later access it from multiple Amazon EC2 instances if needed

We focused on changing the game



1 Amazon EFS is simple



Fully managed

- No hardware, network, file layer
- Create a scalable file system in seconds!

Seamless integration with existing tools and apps

- NFS v4.1—widespread, open
- Standard file system access semantics
- Works with standard OS file system APIs

Simple pricing = simple forecasting

2 Amazon EFS is elastic

File systems grow and shrink automatically as you add and remove files

No need to provision storage capacity or performance

You pay only for the storage space you use, with no minimum fee

3 Amazon EFS is scalable

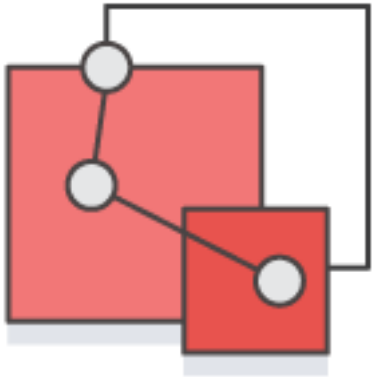
File systems can grow to petabyte scale

Throughput and IOPS scale automatically as file systems grow

Consistent low latencies regardless of file system size

Support for thousands of concurrent NFS connections

Highly durable and highly available



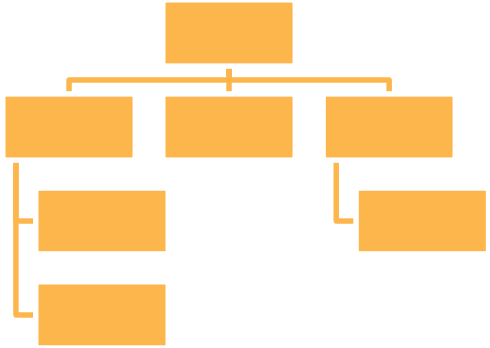
Designed to sustain AZ offline conditions

Superior to traditional NAS availability models

Appropriate for production/tier 0 applications

Diving in

What is a file system?



The primary resource in EFS

Where you store files and directories

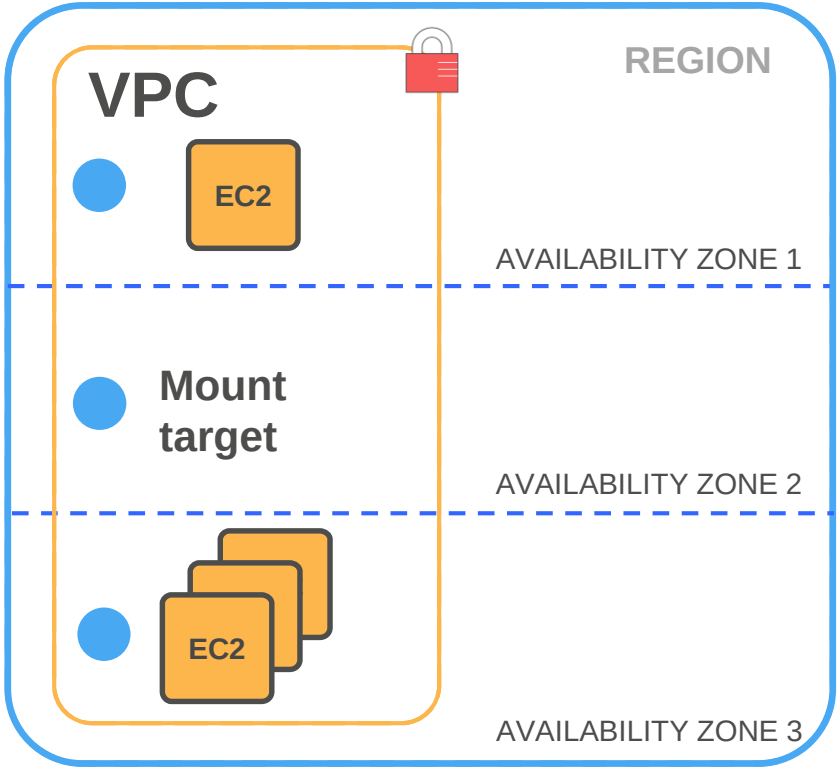
Can create 10 file systems per account

What is a mount target?

To access your file system from instances in a VPC, you create *mount targets* in the VPC

A mount target is an NFS v4 endpoint in your VPC

A mount target has an IP address and a DNS name you use in your mount command



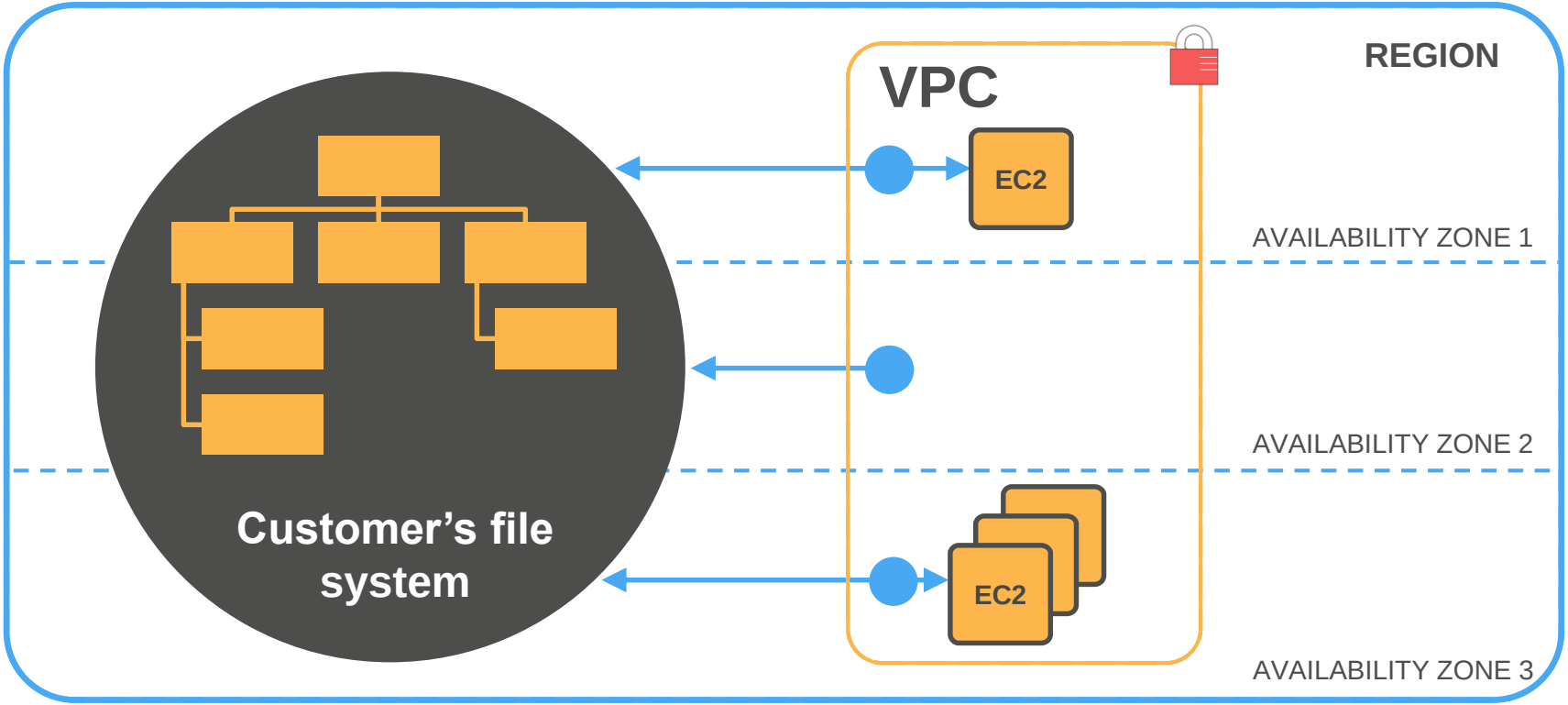
How to access a file system from an instance

You “mount” a file system on an Amazon EC2 instance (standard command) — the file system appears like a local set of directories and files

An NFS v4.1 client is standard on Linux distributions

```
mount -t nfs4 -o nfsvers=4.1
      [file system DNS name]:/
      /[user’s target directory]
```

How does it all fit together?

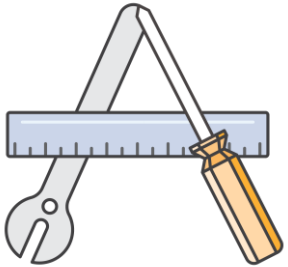


There are three ways to set up and manage a file system

AWS Management Console

AWS Command Line Interface (CLI)

AWS Software Development Kit (SDK)



The AWS Management Console, CLI, and SDK each allow you to perform a variety of management tasks

- Create a file system
- Create and manage mount targets
- Tag a file system
- Delete a file system
- View details on file systems in your AWS account

Setting up and mounting a file system takes under a minute

1. Create a file system
2. Create a mount target in each AZ from which you want to access the file system
3. Enable the NFS client on your instances
4. Run the mount command

Elastic File System Management Console

Elastic File System

Amazon Elastic File System provides file storage for use with your EC2 instances.

Create File System

Getting Started Guide

Create

Create an EFS file system to store your files in the Amazon cloud. A file system grows and shrinks automatically with the files you put in, and you only pay for what you use.

Access

Write files to and read files from your EFS file system via the NFSv4 protocol. Any number of EC2 instances can work with your file system at the same time, and your instances can be in multiple Availability Zones in a Region.

Manage

You can easily administer your file system, and you can view and alert on key metrics using CloudWatch.

File Edit View History Bookmarks Tools Help

Elastic File System Manag...

https://us-west-2.console.aws.amazon.com/efs/home?region=us-west-2#/wizard/1

Create file system

Step 1: Configure file system access

Step 2: Configure optional settings

Step 3: Review and create

Configure file system access

An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system via a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.

VPC vpc-947152fc (default)

Create mount targets

Instances connect to a file system via mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.

	Availability Zone	Subnet	IP address	Security group
✓	us-west-2a	subnet-957152fd (default)	Automatic	sg-a420c2cb - default ×
✓	us-west-2b	subnet-967152fe (default)	Automatic	sg-a420c2cb - default ×
✓	us-west-2c	subnet-977152ff (default)	Automatic	sg-a420c2cb - default ×

Cancel Next Step

File Edit View History Bookmarks Tools Help

Elastic File System Manag...

https://us-west-2.console.aws.amazon.com/efs/home?region=us-west-2#/wizard/2

Create file system

Step 1: Configure file system access

Step 2: Configure optional settings

Step 3: Review and create

Configure optional settings

Add tags

You can add tags to describe your file system. A tag consists of a case-sensitive key-value pair. (For example, you can define a tag with key-value pair with key = Corporate Department and value = Sales and Marketing.) At a minimum, we recommend a tag with key = Name.

Key	Value	Remove
Name	My first EFS file system	
Add New Key		

Choose performance mode

We recommend **General Purpose** performance mode for most file systems. **Max I/O** performance mode is optimized for applications where tens, hundreds, or thousands of EC2 instances are accessing the file system — it scales to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for file operations.

- (●) **General Purpose (default)**
- ( ) **Max I/O**

Cancel | Previous | Next Step

File Edit View History Bookmarks Tools Help

Elastic File System Manag...

https://us-west-2.console.aws.amazon.com/efs/home?region=us-west-2#/wizard/3

Create file system

Step 1: Configure file system access

Step 2: Configure optional settings

Step 3: Review and create

Review and create

Review the configuration below before proceeding to create your file system.

File system access

VPC	Availability Zone	Subnet	IP address	Security group
vpc-947152fc (default)	us-west-2a	subnet-957152fd (default)	Automatic	sg-a420c2cb - default
	us-west-2b	subnet-967152fe (default)	Automatic	sg-a420c2cb - default
	us-west-2c	subnet-977152ff (default)	Automatic	sg-a420c2cb - default

Optional settings

Tags Name: My first EFS file system

Performance mode General Purpose (default)

Cancel Previous Create File System

File Edit View History Bookmarks Tools Help

Elastic File System Manag...

https://us-west-2.console.aws.amazon.com/efs/home?region=us-west-2#/filesystems/fs-096f99a0

	Name	File system ID	Metered size	Number of mount targets	Creation date
○		fs-0809e4a1	6.0 KiB	3	2015-05-22T19:57:16Z
◉	My first EFS file system	fs-096f99a0	6.0 KiB	3	2016-07-20T18:28:35Z

Other details

Owner ID 031817516254

Life cycle state Available

Performance mode General Purpose

Tags

Manage tags

Name: My first EFS file system

File system access

Manage file system access

DNS names

EC2 mount instructions

Mount targets

VPC	Availability Zone	Subnet	IP address	Mount target ID	Network interface ID	Security groups	Life cycle state
vpc-947152fc (default)	us-west-2b	subnet-967152fe (default)	172.31.29.52	fsmt-568474ff	eni-f93c20b5		Creating
	us-west-2a	subnet-957152fd (default)	172.31.34.49	fsmt-578474fe	eni-0a7e5977		Creating
	us-west-2c	subnet-977152ff (default)	172.31.7.161	fsmt-a9857500	eni-d6963189		Creating

EC2 mount instructions

Setting up your EC2 instance

1. Using the Amazon EC2 console, associate your EC2 instance with a VPC security group that enables access to your mount target. For example, if you assigned the "default" security group to your mount target, you should assign the "default" security group to your EC2 instance. (learn more about using VPC security groups with Amazon EFS)
2. Open an SSH client and connect to your EC2 instance. (find out how to connect)
3. Install the nfs client on your EC2 instance.
   - On an Amazon Linux, Red Hat Enterprise Linux, or SuSE Linux instance:
     `sudo yum install -y nfs-utils`
   - On an Ubuntu instance:
     `sudo apt-get install nfs-common`

Mounting your file system

1. Open an SSH client and connect to your EC2 instance. (find out how to connect)
2. Create a new directory on your EC2 instance, such as "efs".
   - `sudo mkdir efs`
3. Mount your file system using the DNS name. The following command looks up your EC2 instance's Availability Zone (AZ)

Close

DNS Names

A file system has a unique DNS name for each Availability Zone that has a mount target.

Availability Zone	DNS Name
us-west-2c	us-west-2c.fs-d694707f.efs.us-west-2.amazonaws.com
us-west-2a	us-west-2a.fs-d694707f.efs.us-west-2.amazonaws.com
us-west-2b	us-west-2b.fs-d694707f.efs.us-west-2.amazonaws.com

You can mount your file system directly using the DNS name(s) above or use a command like the one below to automatically mount your file system using the DNS name for the mount target in the same Availability Zone as the connecting EC2 instance. This command requires a mount target in the same Availability Zone as the instance and that the directory "efs" has already been created on your instance.

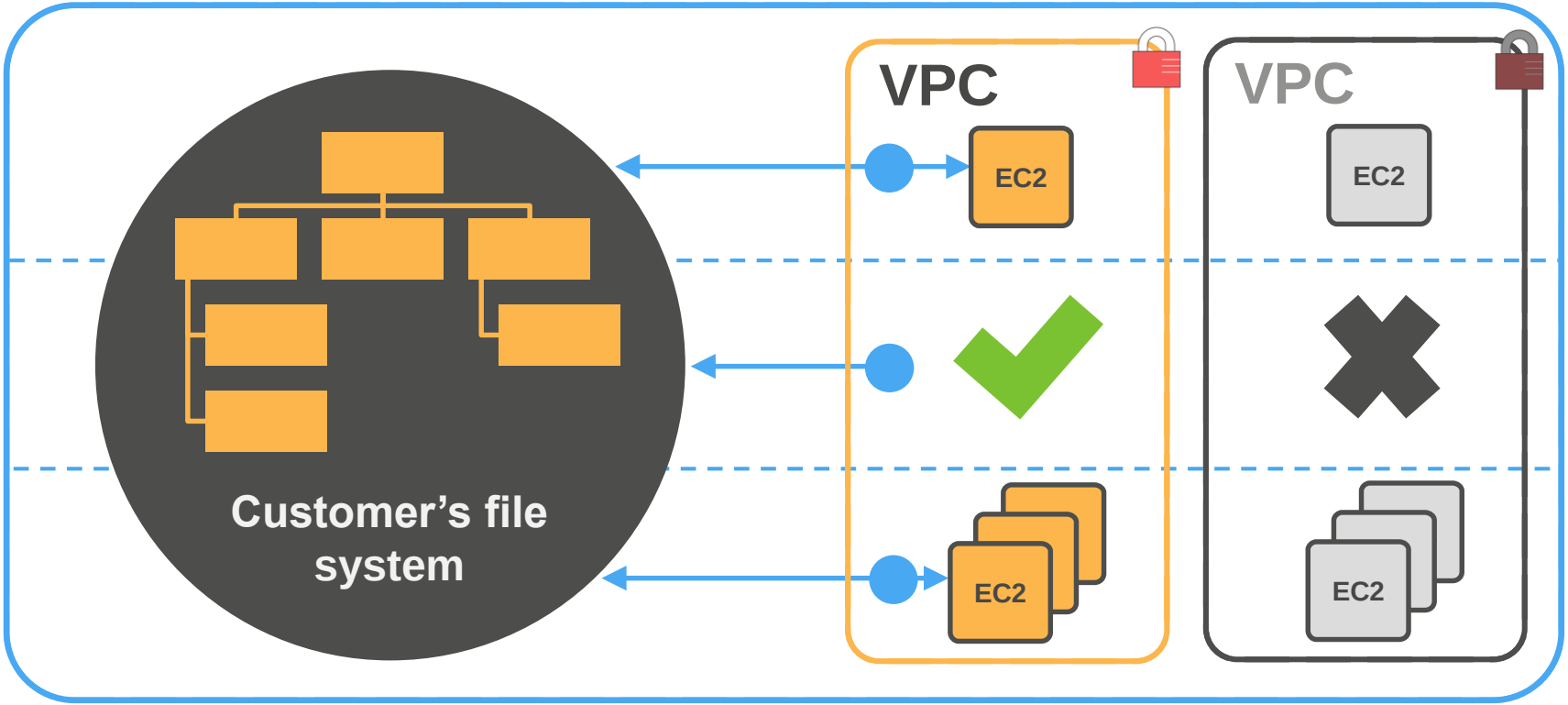
- `sudo mount -t nfs4 $(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone).fs-d694707f.efs.us-west-2.amazonaws.com:/ efs`

Ok

 Privacy Policy Terms of Use

Securing your file system

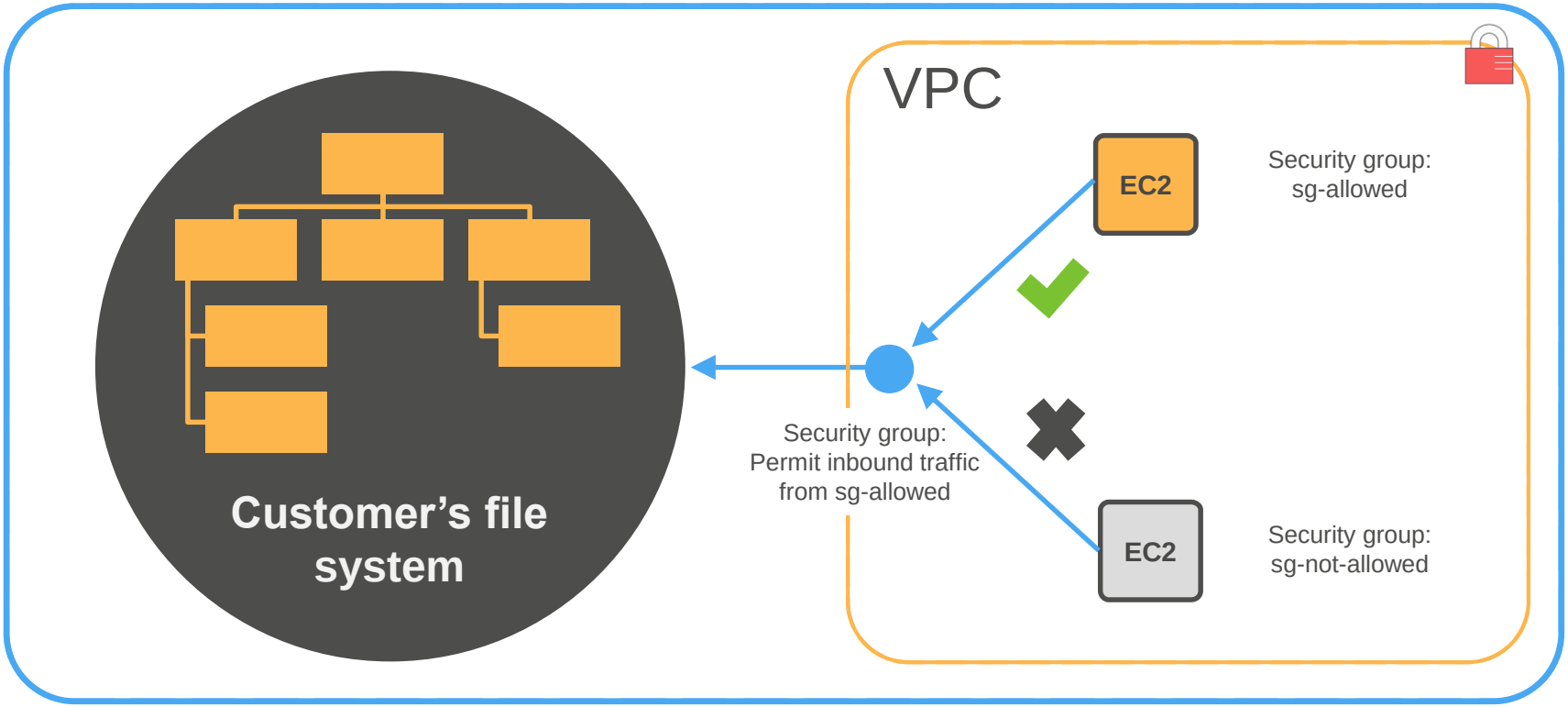
Only EC2 instances in the VPC you specify can access your EFS file system



Several security mechanisms

- **Control network traffic** to and from file systems (mount targets) by using VPC security groups and network ACLs
- **Control file and directory access** by using POSIX permissions
- **Control administrative access** (API access) to file systems by using AWS Identity and Access Management (IAM)

Security groups control which instances in your VPC can connect to your mount targets

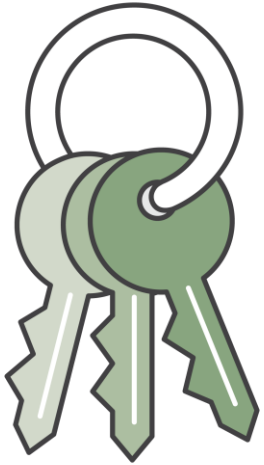


EFS supports POSIX file and directory access permissions

Set file/directory permissions to specify read-write-execute permissions for users and groups

```
drwxr-xr-x  4 root     root     4096 Feb  5 22:37 .
dr-xr-xr-x 25 root     root     4096 Feb  5 22:20 ..
drwxr-xr-x  2 mike     mike     4096 Feb  4 01:18 mike
```

Integration with IAM provides administrative security



Use IAM policies to control who can use the administrative APIs to create, manage, and delete file systems

EFS supports *action-level* and *resource-level* permissions

Availability and durability

In which regions can I use EFS?

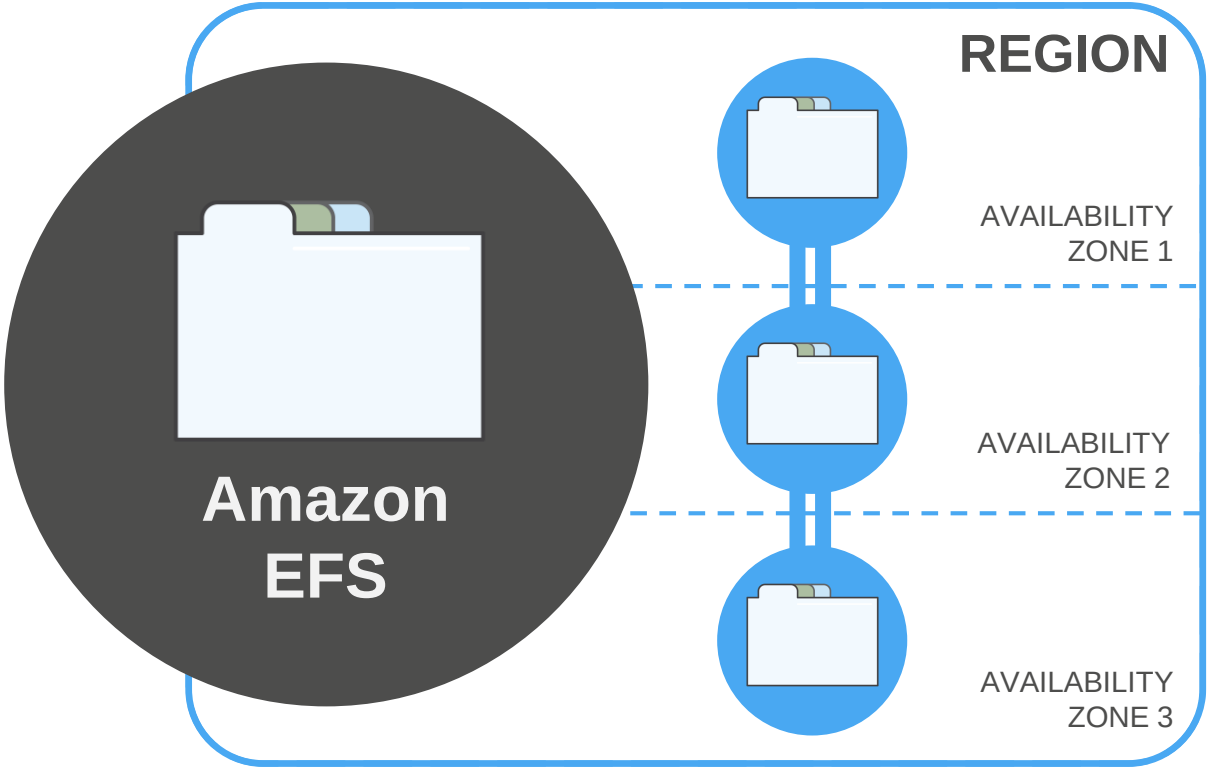
US West (Oregon)

US East (N. Virginia)

EU (Ireland)

Data is stored in multiple AZs for high availability and durability

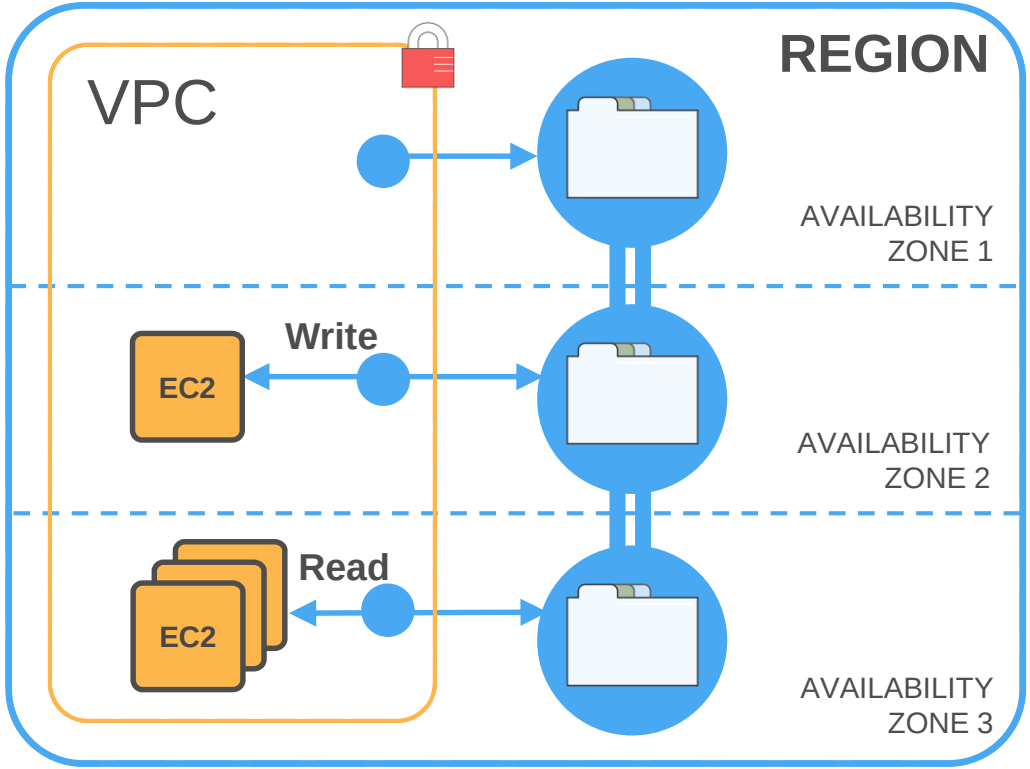
Every file system object (directory, file, and link) is redundantly stored across multiple AZs in a region



Data can be accessed from any AZ in the region while maintaining full consistency

Your EC2 instances can connect to your EFS file system from any AZ in a region

All reads will be fully consistent in all AZs—that is, a read in one AZ is guaranteed to have the latest data, even if the data is being written in another AZ



Performance

Amazon EFS is designed for wide spectrum of use cases

High throughput and parallel I/O

Genomics
Big data analytics
Scale-out jobs

Web serving
Home directories
Content management

Metadata-intensive jobs

Low latency and serial I/O

EFS provides throughput that scales as a file system grows

As a file system gets larger, it needs access to more throughput

Many file workloads are spiky, with peak throughput well above average levels

Amazon EFS scalable bursting model is designed to make performance available when you need it

Bursting model examples

File system size	Read/write throughput
A **1 TB** EFS file system can…	• Drive up to 50 MB/s continuously *or* • Burst to 100 MB/s for up to 12 hours each day*
A **10 TB** EFS file system can…	• Drive up to 500 MB/s continuously *or* • Burst to 1 GB/s for up to 12 hours each day*
A **100 GB** EFS file system can…	• Drive up to 5 MB/s continuously *or* • Burst to 100 MB/s for up to 72 minutes each day*

* Assumes a 75% read/25% write workload, 256 KB I/O size, 50% compressibility, and 1 client

EFS has a distributed data storage design

File systems distributed across unconstrained number of servers

- Avoids bottlenecks/constraints of traditional file servers
- Enables multi-threaded and distributed applications to achieve high levels of aggregate IOPS/throughput

Data also distributed across AZs (durability, availability)

How to think about EFS perf relative to EBS

		Amazon EFS	Amazon EBS PIOPS
Performance	Per-operation latency	Low, consistent	Lowest, consistent
	Throughput scale	Multiple GBs per second	Single GB per second
Characteristics	Data Availability/Durability	Stored redundantly across multiple AZs	Stored redundantly in a single AZ
	Access	1 to 1000s of EC2 instances, from multiple AZs, concurrently	Single EC2 instance in a single AZ
	Use Cases	Big Data and analytics, media processing workflows, content management, web serving, home directories	Boot volumes, transactional and NoSQL databases, data warehousing & ETL

Two performance modes designed to support a broad spectrum of use cases

Default: Recommended for most use cases

General purpose mode

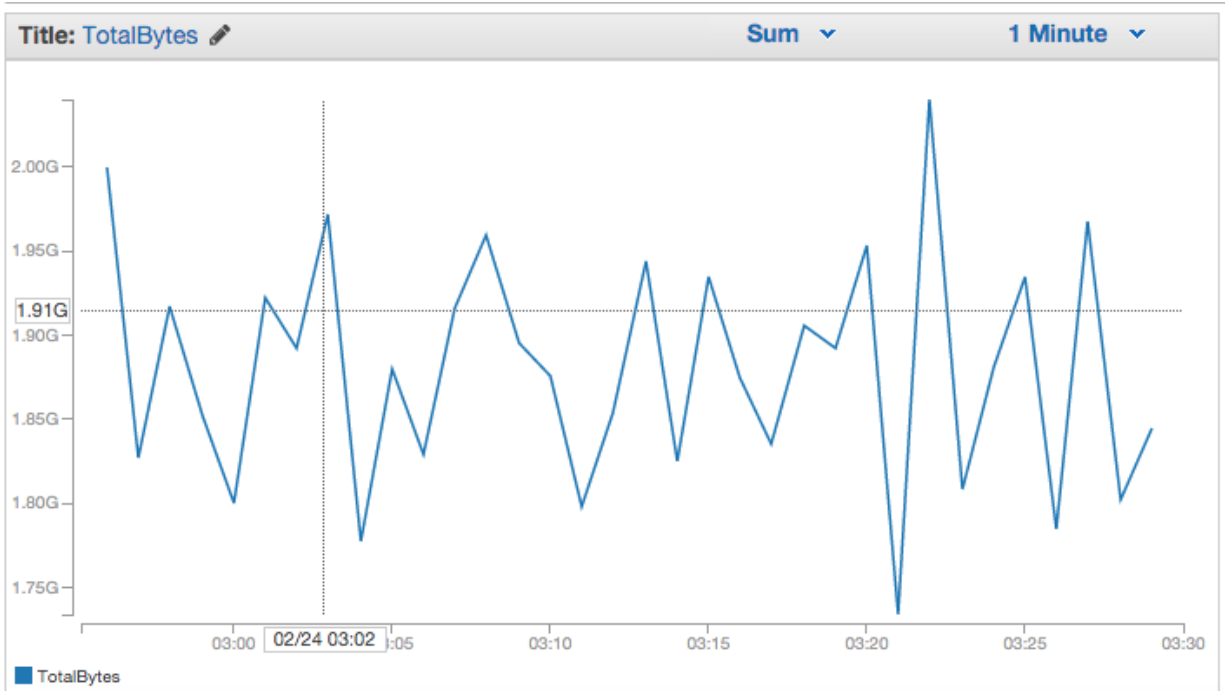
Optimized for latency-sensitive applications and general-purpose file-based workloads – this mode is the best option for the majority of use cases

Max I/O mode

Optimized for large-scale and data-heavy applications where tens, hundreds, or thousands of EC2 instances are accessing the file system — it scales to higher levels of aggregate throughput and ops per second with a tradeoff of slightly higher latencies for file operations

Use CloudWatch to determine whether your application can benefit from Max I/O; if not, you'll get the best performance in general purpose mode

CloudWatch metrics provide visibility into file system performance



Wrapping up

Simple and predictable pricing

With EFS, you pay only for the storage space you use

- No minimum commitments or up-front fees
- No need to provision storage in advance
- No other fees, charges, or billing dimensions

EFS price: **$0.30/GB-month**

TCO example

Let’s say you need to store ~500GB and require high availability and durability

Using a shared file layer on top of EBS, you might provision 1 TB and fully replicate the data to a second AZ for availability/durability

Example GlusterFS cost:

Storage (2x 1TB EBS gp2 volumes):	$205 per month
Compute (2x m4.xlarge instances):	$350 per month
Inter-AZ bandwidth costs (est.):	$30 per month
Total	$585 per month

EFS cost is (500GB * $0.30/GB-month) = $150 per month, with no additional charges

What to do next?

Learn more at aws.amazon.com/efs

Try it out for free!

Thank you!

Q&A next