Storage on AWS

Chetan Agrawal, Solutions Architect

Deven Suri, Account Manager

23-Jul-2020

Agenda

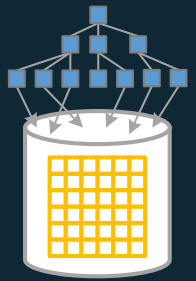
- Introduction
- Storage Primer
- Block Storage
- Shared File Systems
- Object Store
- Demo
- Quiz

Storage Primer

Block vs File vs Object

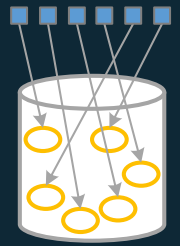
Block Storage

Raw Storage
Data organized as an array of unrelated blocks
Host File System places data on disk
Ex: Hard Disks, Storage Area Network (SAN) Storage Arrays



File Storage

Unrelated data blocks managed by a file (serving) system
Native file system places data on disk
Ex: Network Attached Storage (NAS) Appliances, Windows File Servers



Object Storage

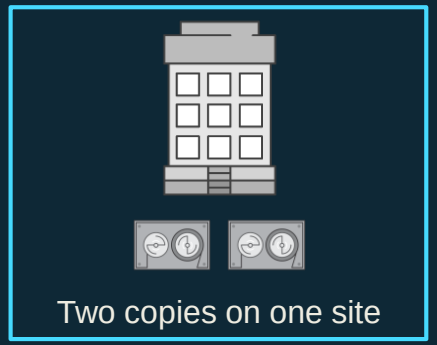
Stores Virtual containers that encapsulate the data, data attributes, metadata and Object IDs
API Access to data
Metadata Driven, Policy-based, etc.
Ex: Ceph, OpenStack Swift

Storage - Characteristics

Some of the ways we look at storage

Durability	Availability	Security	Cost	Scalability	Performance	Integration
Measure of expected data loss	Measure of expected downtime	Security measures for at-rest and in-transit data	Amount per storage unit, e.g. $ / GB	Upward flexibility, storage size, number of users	Performance metrics (bandwidth, latency)	Ability to interact via API or with other services

Understanding Durability



designed for

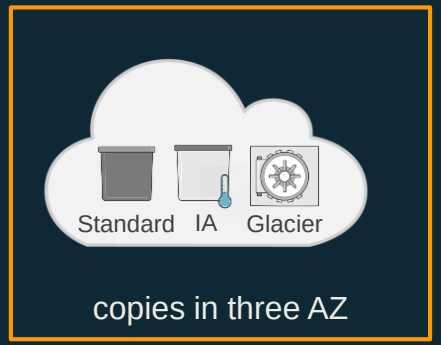
99.99%

durability

designed for

99.999%

durability



designed for

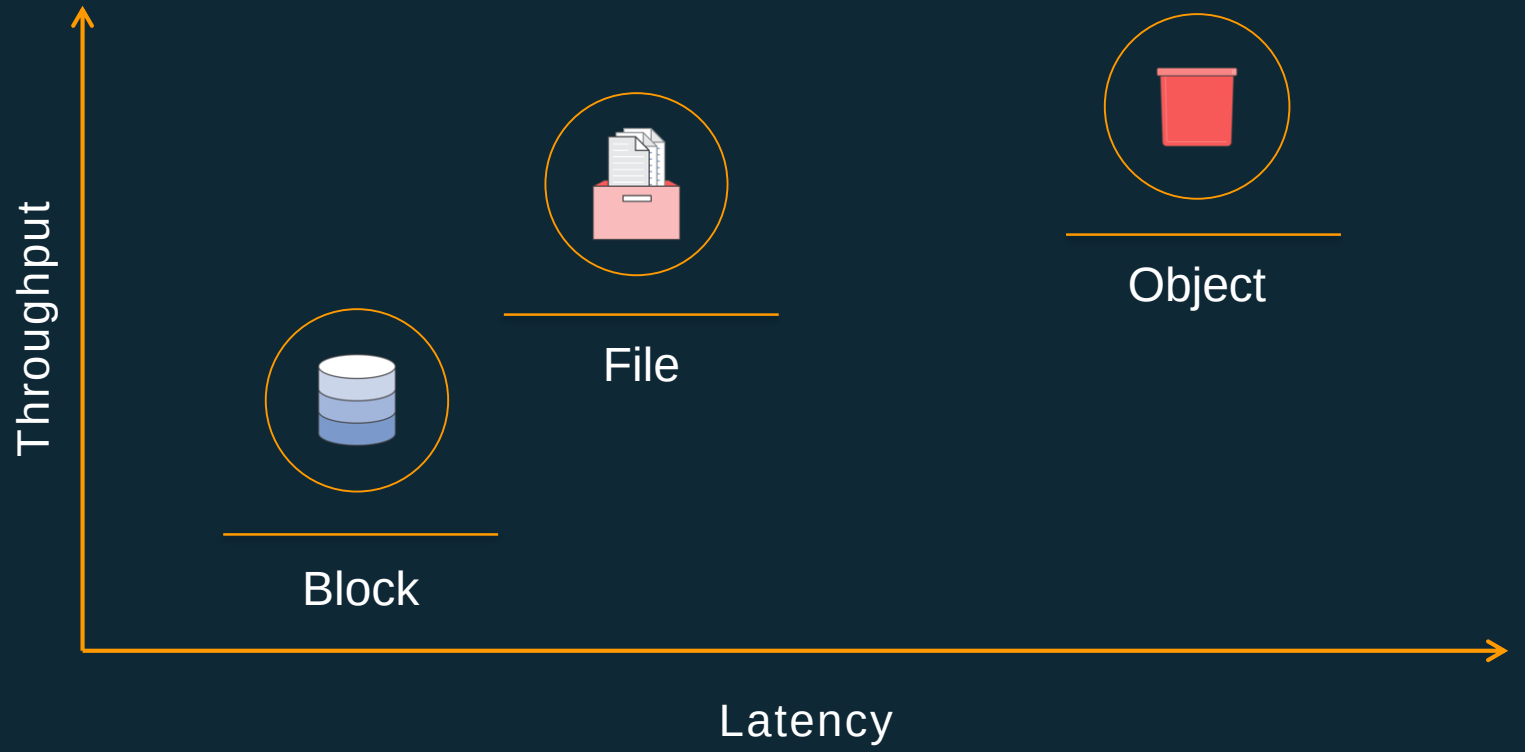
99.999999999%

durability

Availability vs Durability

%	Availability	Durability
99.999	5 minutes 15 seconds	1 in 100,000
99.9999	31 seconds	1 in 1,000,000
99.99999	3 seconds	1 in 10,000,000
99.999999999	300 uSeconds	1 in 100,000,000,000

Performance comparison of storage types



More choice for more applications

Block storage

General Purpose SSD

Provisioned IOPS SSD

Throughput-Optimized HDD

Cold HDD

Backup

AWS Backup

File storage

EFS Standard

EFS Infrequent Access

FSx for Windows

FSx for Lustre

Object storage

S3 Standard

S3 Standard-IA

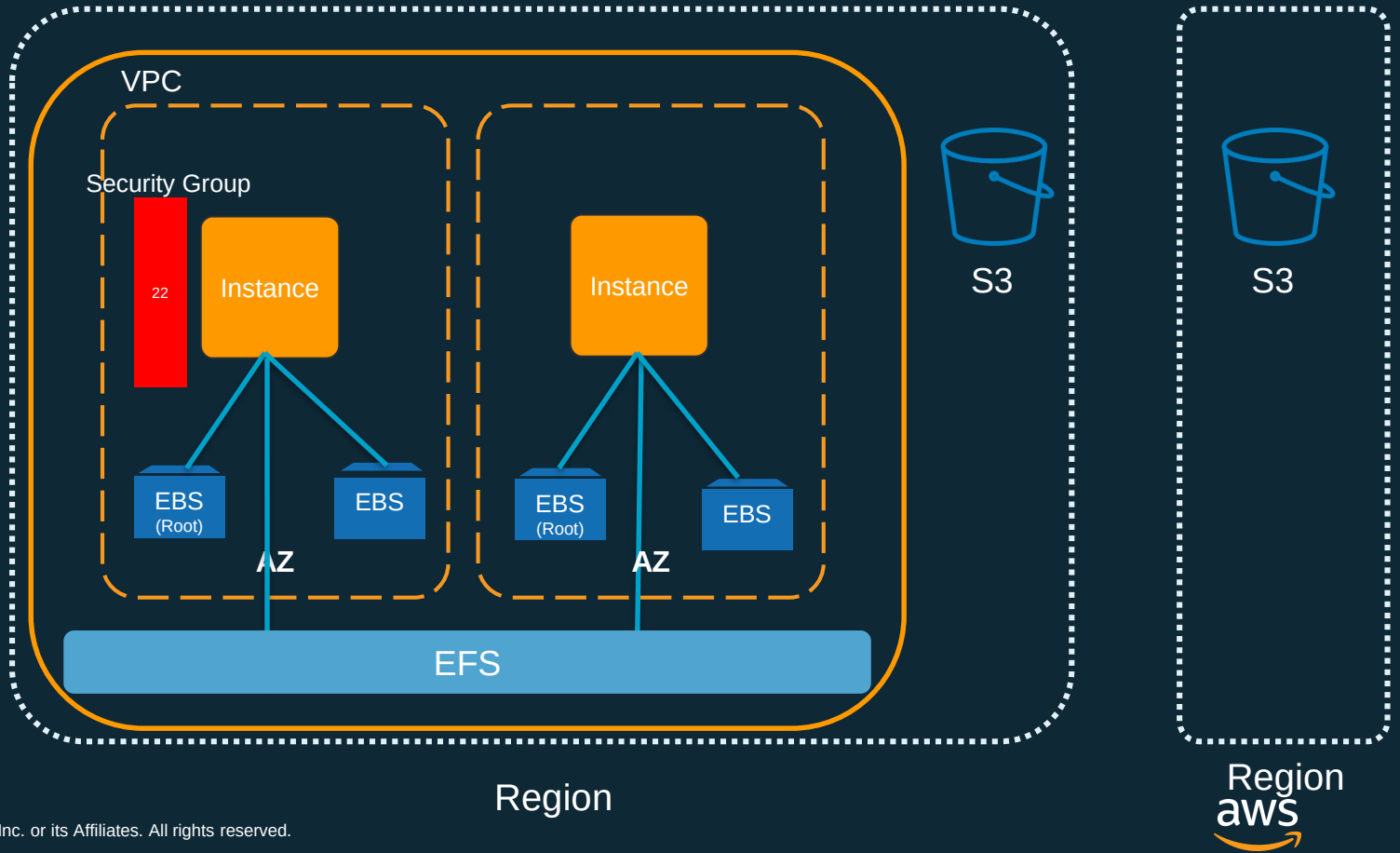
S3 One Zone-IA

S3 Intelligent-Tiering

S3 Glacier

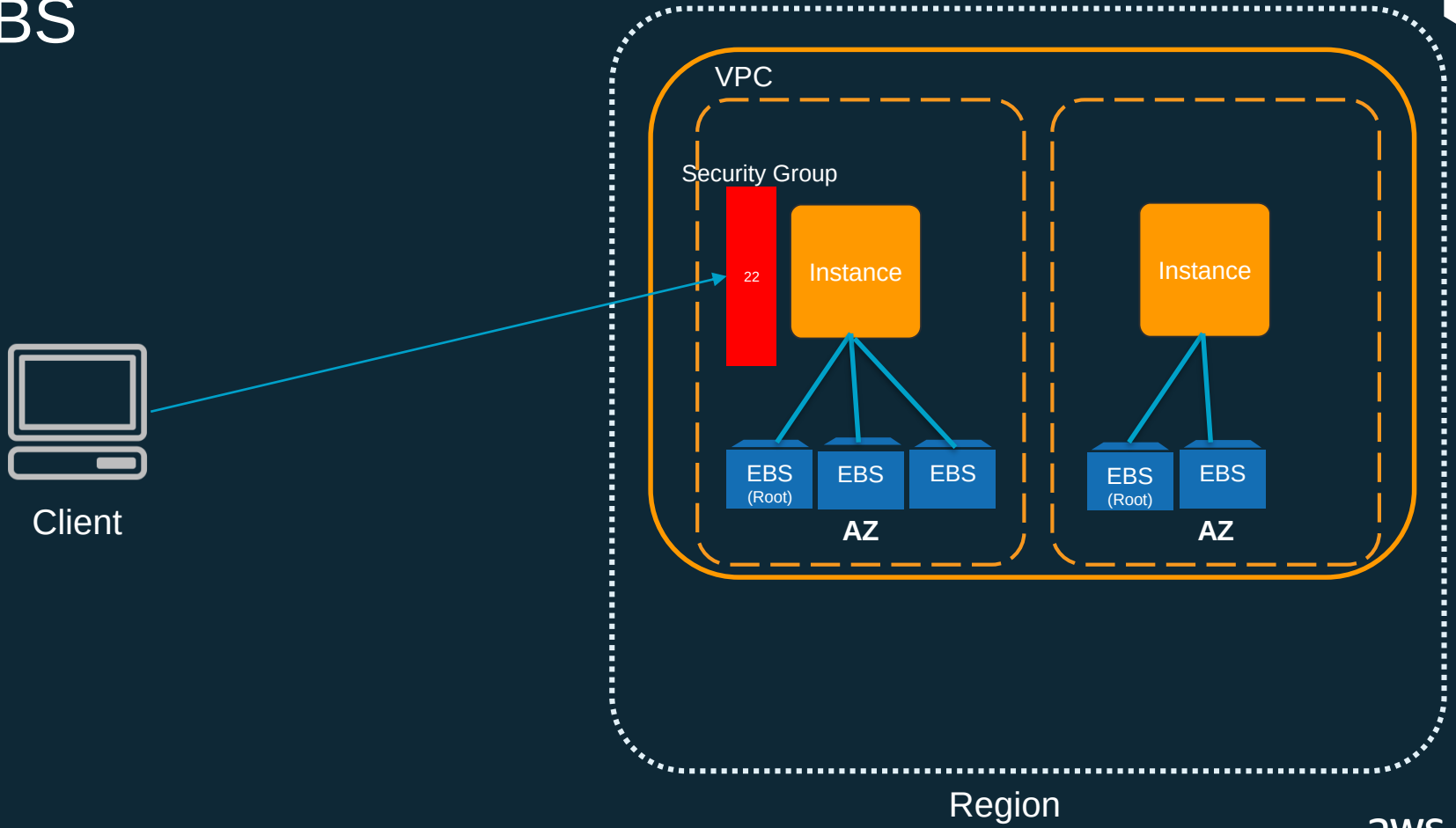
S3 Glacier Deep Archive

EBS vs EFS vs S3

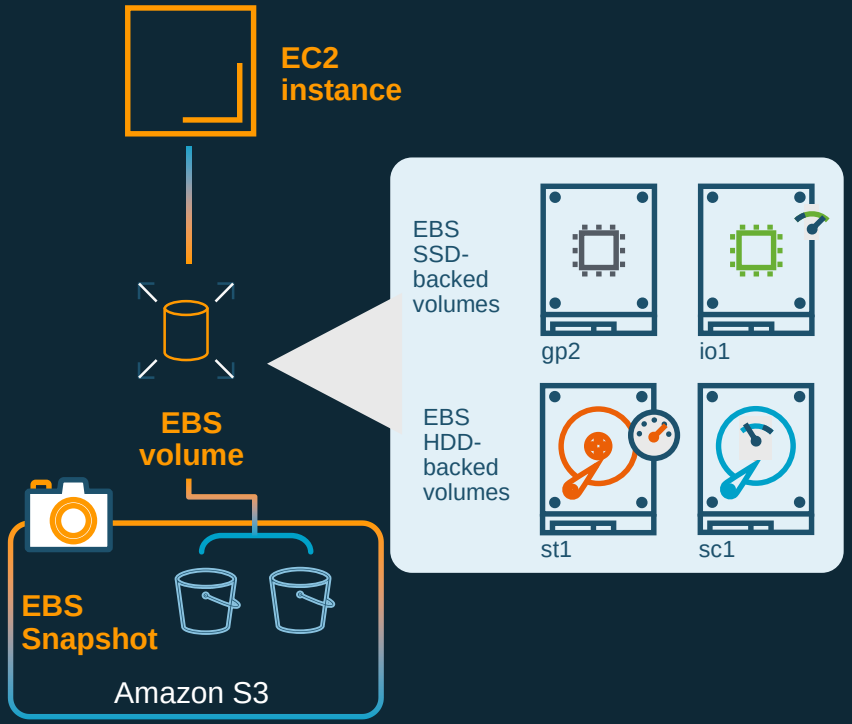


Block Storage - EBS

EBS

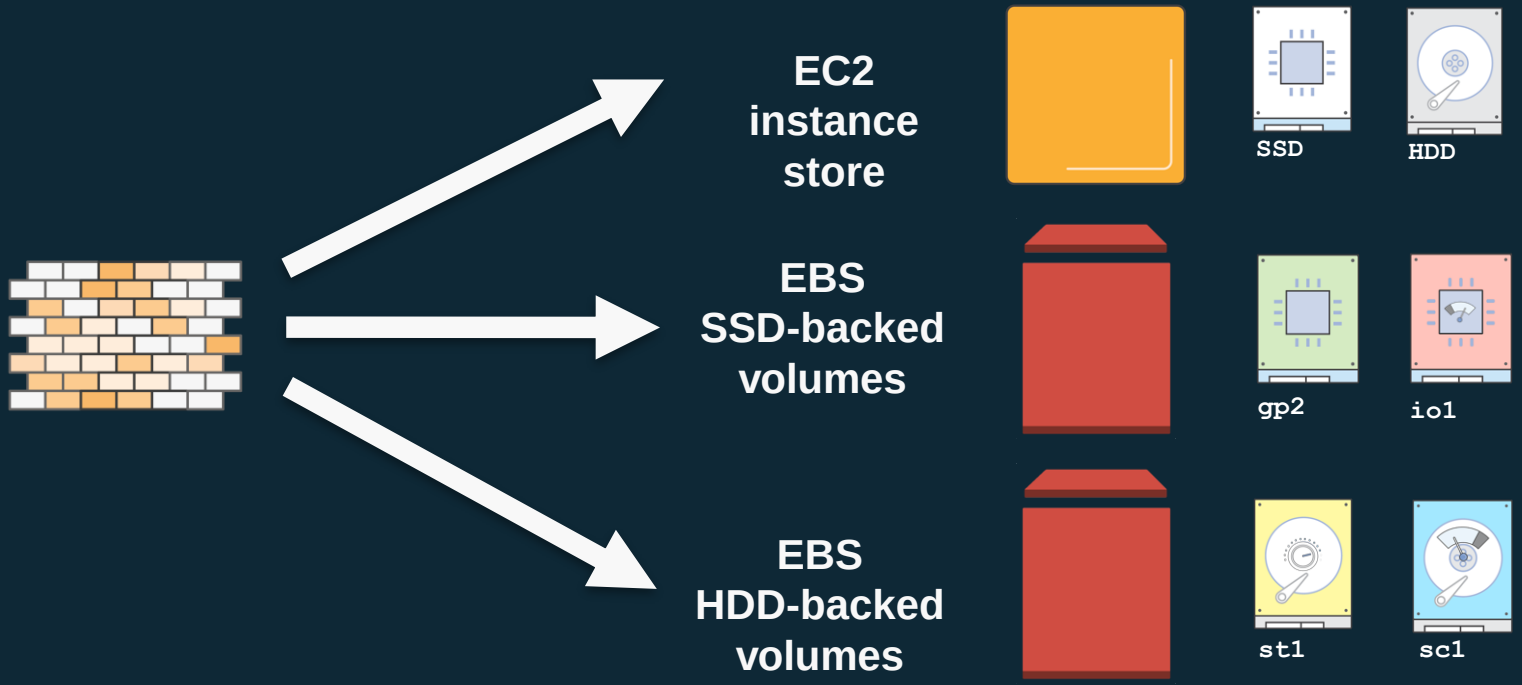


Amazon Elastic Block Store (EBS)

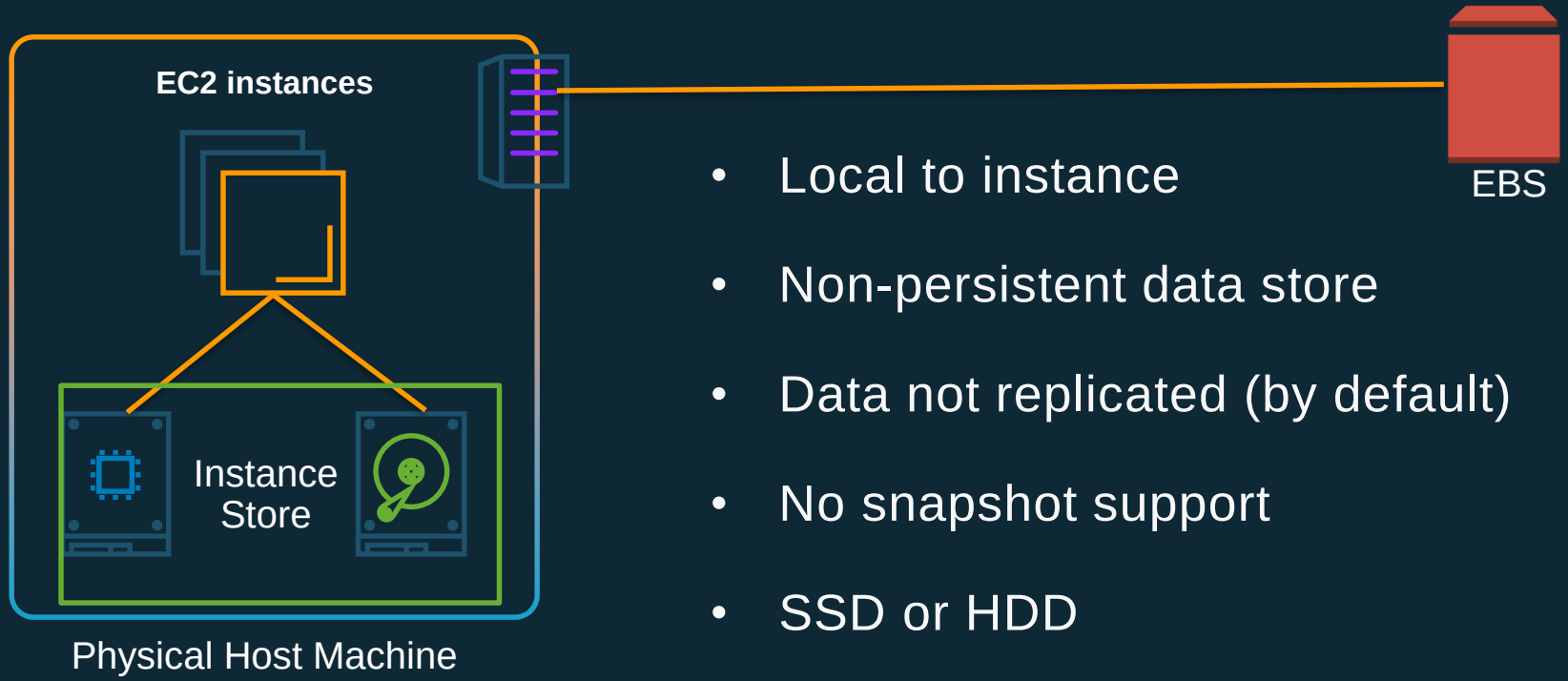


- Block storage as a service
- Create, attach volumes through an API
- Service accessed over the network
- Many volumes can attach to an instance
- Detach and attach between instances
- Supports Snapshots: Point-in-time backup of modified volume blocks
- Choice of magnetic and SSD-based volume types
- Supports Encryption (Data at Rest)

AWS block storage offerings

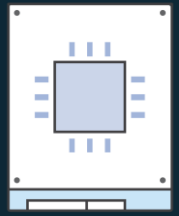


Amazon EC2 instance store volume



- Local to instance
- Non-persistent data store
- Data not replicated (by default)
- No snapshot support
- SSD or HDD

Amazon EBS use cases



SSD

HDD

Relational Databases

MySQL, SQL Server, PostgreSQL, SAP, Oracle

NoSQL Databases

Cassandra, MongoDB, CouchDB

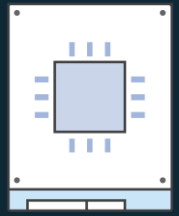
Big Data , Analytics

Kafka, Splunk, Hadoop, Data Warehousing

File / Media

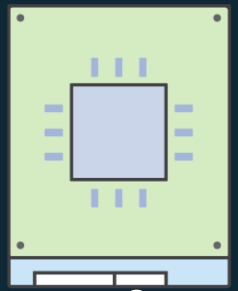
CIFS/NFS, Transcoding, Encoding, Rendering

Amazon EBS volume types



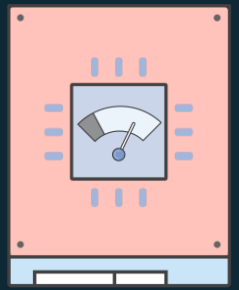
SSD

HDD



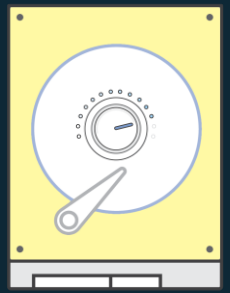
gp2

General Purpose SSD



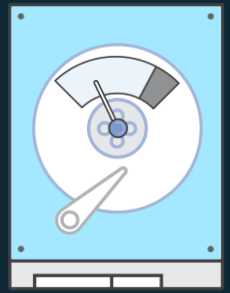
io1

Provisioned IOPS SSD



st1

Throughput Optimized HDD

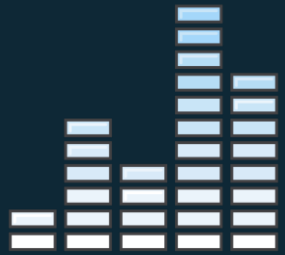


sc1

Cold HDD

Choosing an Amazon EBS volume type

What is more important to your workload?

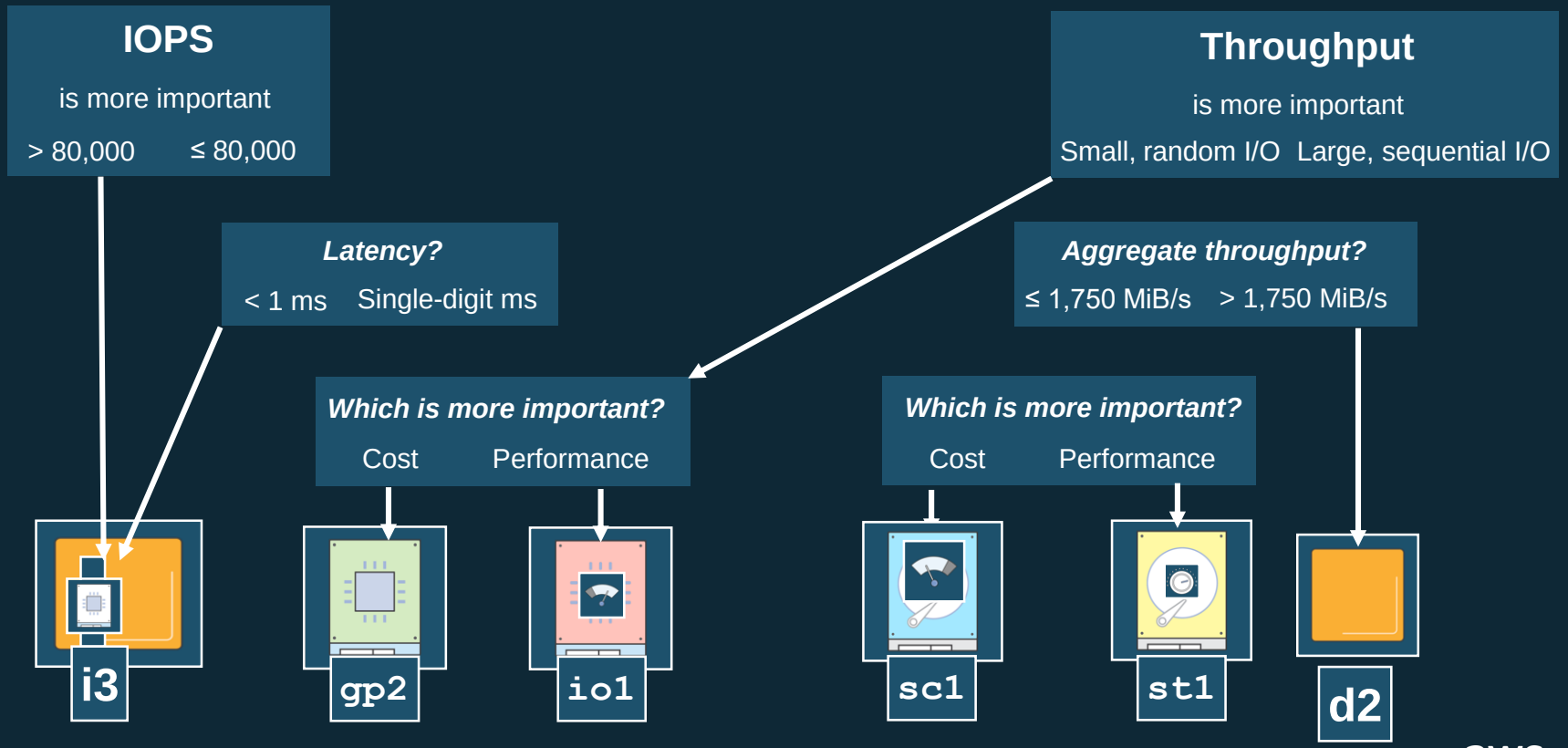


IOPS

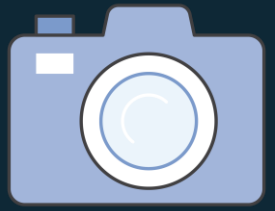
or

Throughput?

Choosing an Amazon EBS volume type

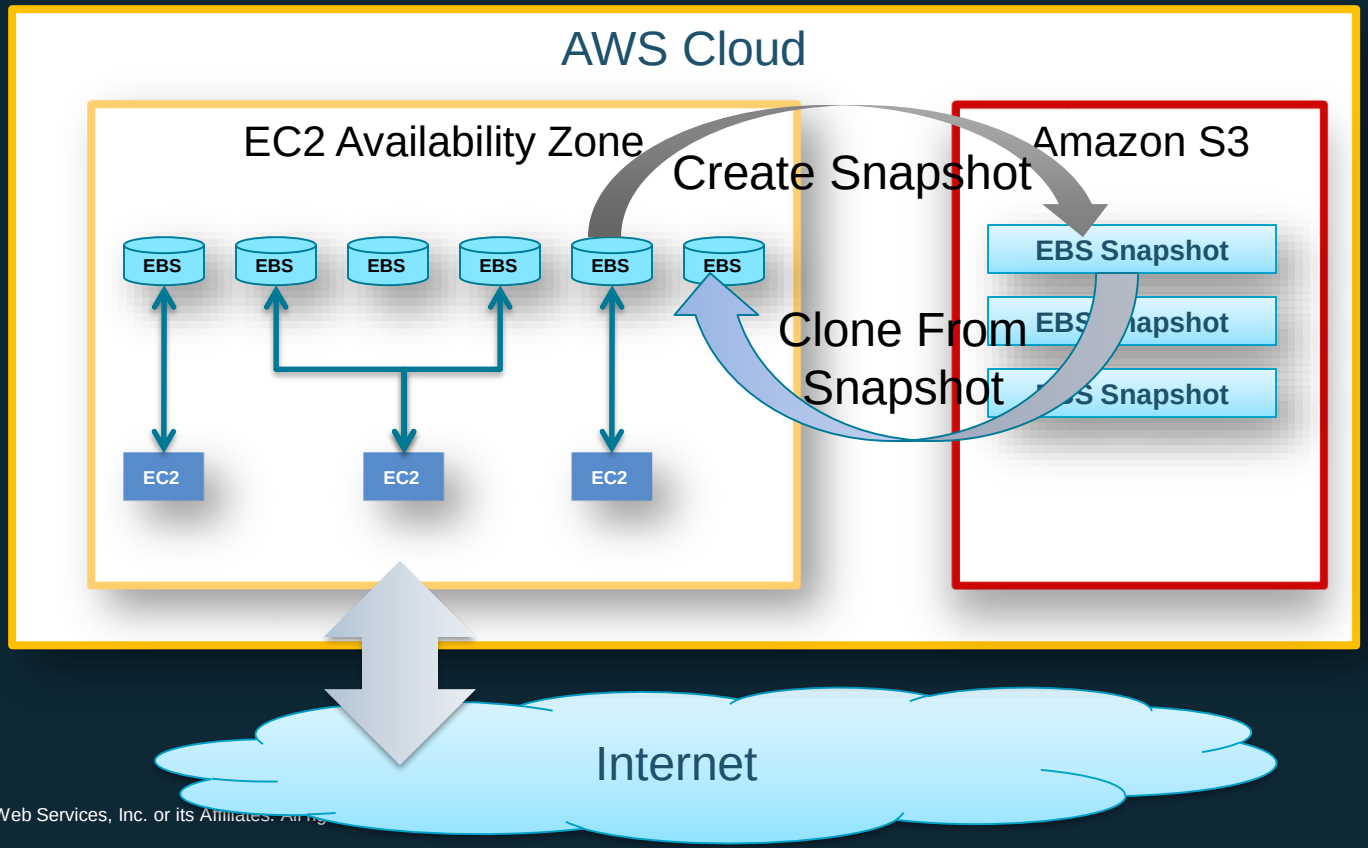


EBS Snapshots



- Point-in-time snapshots of volume blocks
- Stored in Amazon S3 and accessed via EBS APIs
- Key Features:
    - Immediate access to EBS volume data
    - Resizing EBS volumes
    - Sharing EBS Snapshots
    - Copying EBS Snapshots across AWS regions

EBS Snapshots – Delta Block Changes



Let’s add additional EBS disk to our EC2 instance

1. Attach additional EBS volume to EC2
2. Create snapshot of EBS Volume

Command to format and mount EBS (Linux)

$ lsblk
*$ sudo mkfs -t ext4 /dev/**xvdf***
$ sudo mkdir /data-volume
$ sudo mount /dev/xvdf /data-volume
$ df

Object Stores

Amazon S3 (Simple Storage Service)

- Web accessible object store (through API or HTTPS)
- Highly durable (99.999999999% design)
- Limitlessly scalable
- Static Website Hosting
- Secure Access through IAM, Pre-sign URL
- Standard Storage Pricing (us-east-1) - $0.023 per GB

mybucket

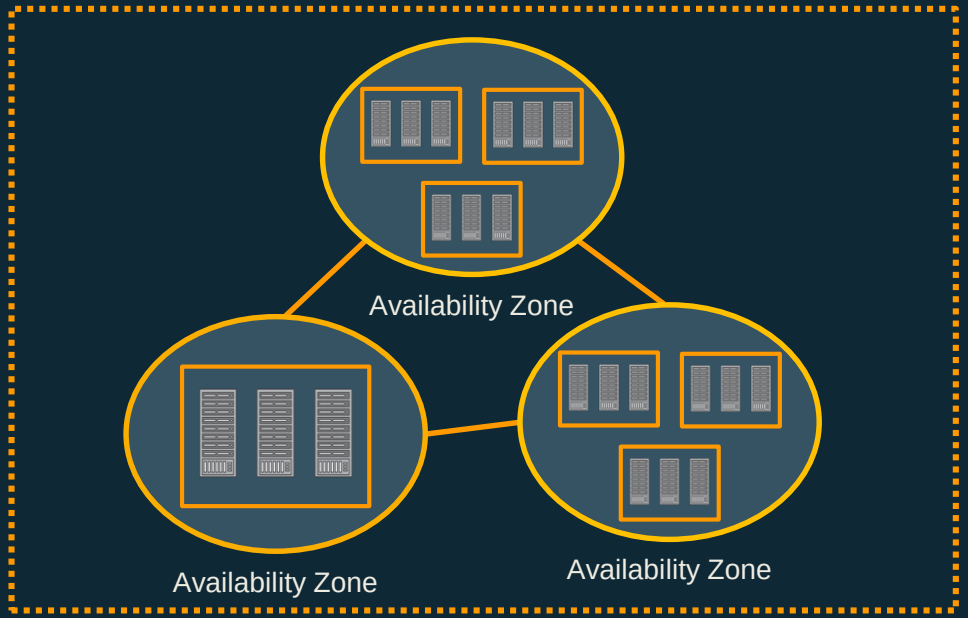
http://mybucket.s3.amazonaws.com/2006-03-01/AmazonS3.html

Bucket

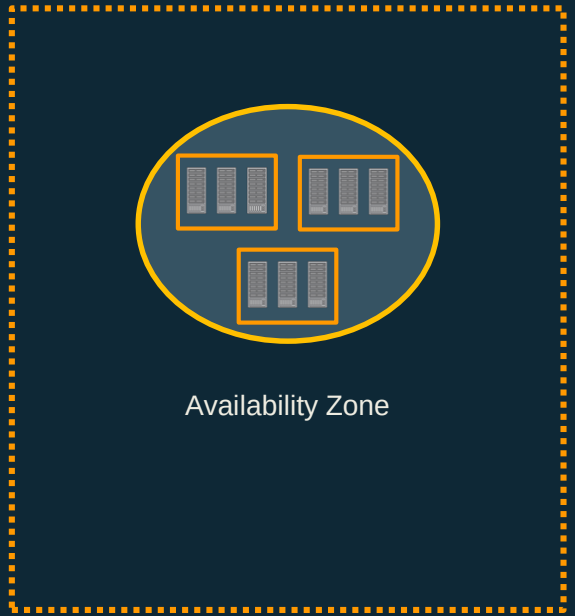
Object/Key

What defines durability and availability numbers?

Copies across Multiple AZs within Region



Single AZ



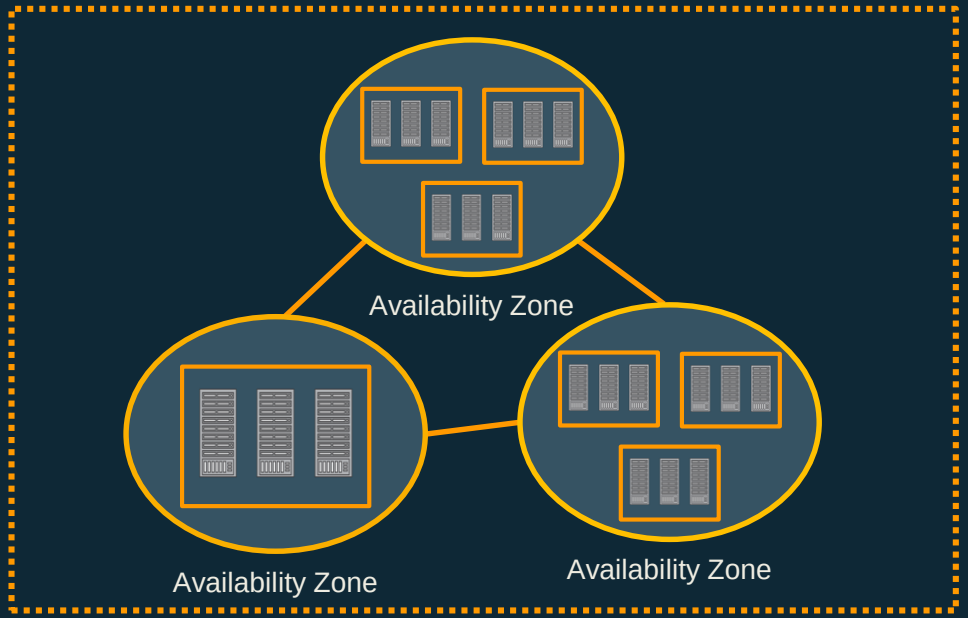
Your choice of Amazon S3 storage classes

S3 Standard	S3 Intelligent-Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier	S3 Glacier Deep Archive
• Active, frequently accessed data • Milliseconds access • ≥ 3 AZ • From: $0.0210/GB	• Data with changing access pattern • Milliseconds access • ≥ 3 AZ • From: $0.0210 to $0.0125/GB • Monitoring fee per obj. • Min storage duration	• Infrequently accessed data • Milliseconds access • ≥ 3 AZ • From: $0.0125/GB • Retrieval fee per GB • Min storage duration • Min object size	• Re-creatable less accessed data • Milliseconds access • 1 AZ • From: $0.0100/GB • Retrieval fee per GB • Min storage duration • Min object size	• Archive data • Minutes to hours access • ≥ 3 AZ • From: $0.0040/GB • Retrieval fee per GB • Min storage duration • Min object size	• Archive data • Hours access • ≥ 3 AZ • From: $0.00099/GB • Retrieval fee per GB • Min storage duration • Min object size

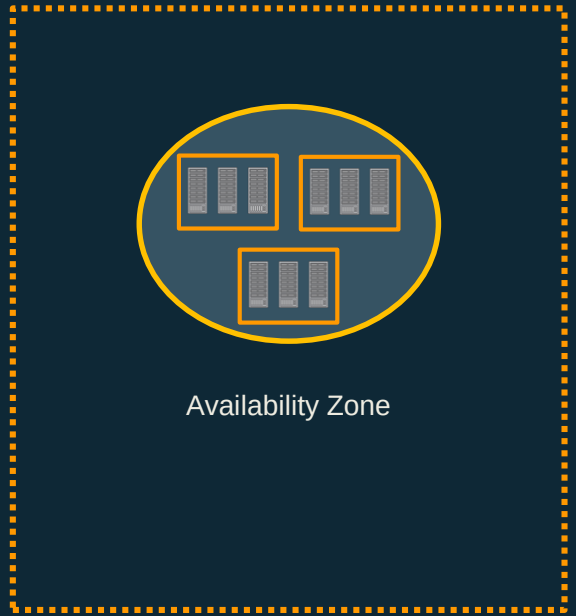
Frequent ← *Access Frequency* → *Infrequent*

How do AWS object storage classes differ in design?

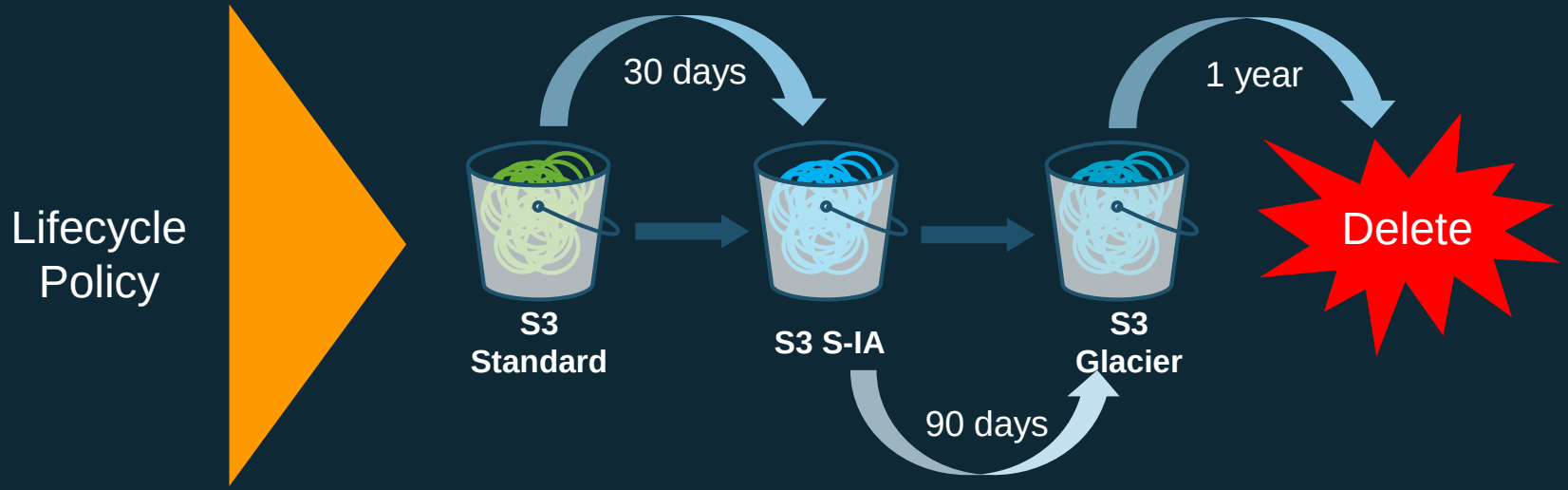
S3 Standard
S3 Standard-IA
Glacier



S3 One Zone-IA



Set S3 Lifecycle Policy to tier and expire storage



S3 Intelligent-Tiering automates cost savings

- Automatically optimizes storage costs for data with changing access patterns
- Moves objects between two storage tiers:
  - Frequent Access Tier
  - Infrequent Access Tier
- Monitors access patterns and auto-tiers on granular object level
- Milliseconds access, ≥ 3 AZ, Monitoring fee per Object, minimum storage duration

Cross-Region Replication

- Flexibility to replicate data
- From any region to any region
- To any storage class
- Across AWS accounts

S3 Security

- AWS Identity and Access Management (IAM)
- S3 bucket policies
- Objects Access Control Lists (ACLs)
- Signed-URL
- S3 supports both server-side & client-side encryption

***S3 Block Public Access** to ensure S3 buckets and objects do not have public access*

Let’s upload something to S3 and try to access

1. Upload new file to S3
2. Try to access over HTTP/S
3. Make the object Public
4. Create pre-sign URL

    $ aws s3 presign "s3://chetan-demo/mustang.jpg" --expires-in 180 --region ap-south-1

S3 High Level CLI operations

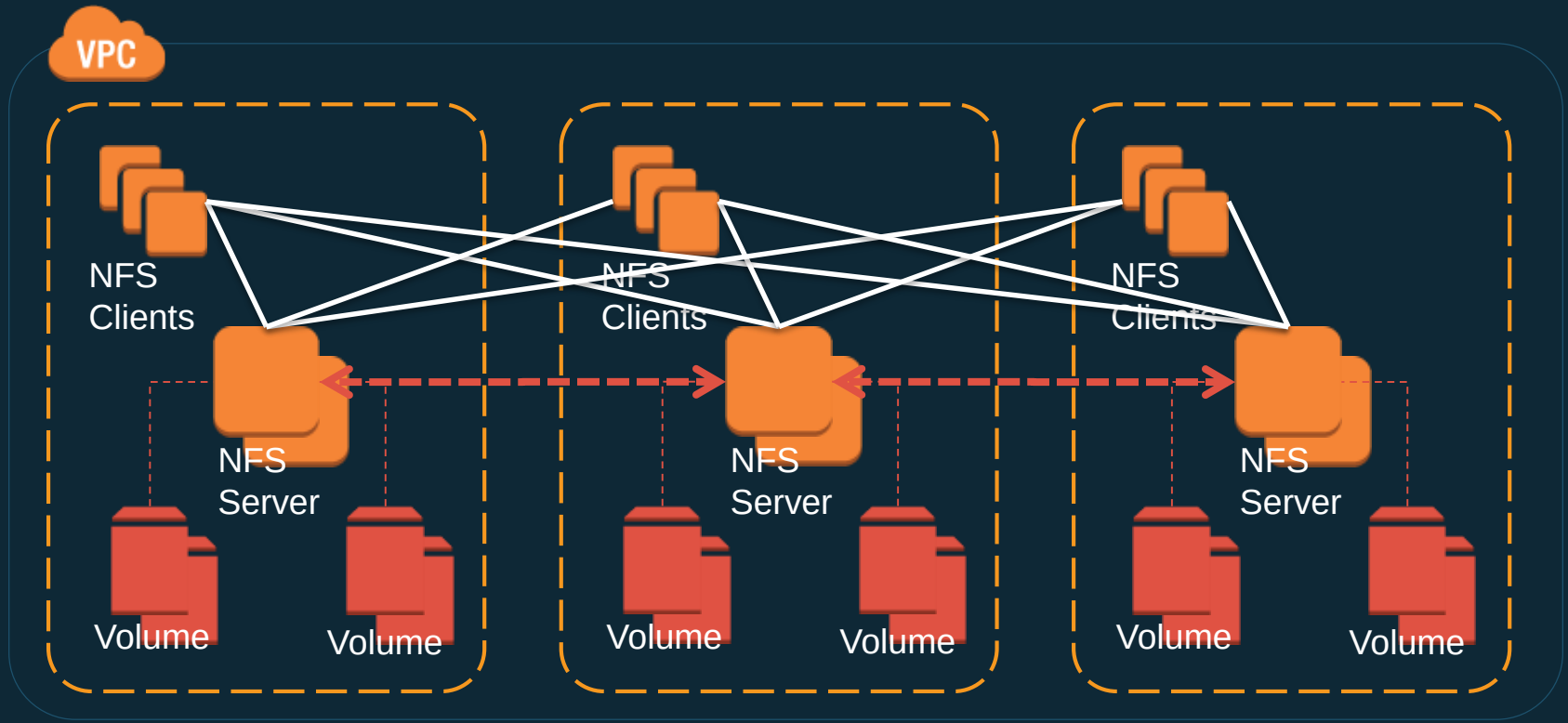
Command	Description
cp	Copies a local file or S3 object to another location locally or in S3.
ls	List S3 objects and common prefixes under a prefix or all S3 buckets.
mb	Creates an S3 bucket.
mv	Moves a local file or S3 object to another location locally or in S3.
presign	Generate a pre-signed URL for an Amazon S3 object. This allows anyone who receives the pre-signed URL to retrieve the S3 object with an HTTP GET request.
rb	Deletes an empty S3 bucket.
rm	Deletes an S3 object.
sync	Syncs directories and S3 prefixes. Recursively copies new and updated files from the source directory to the destination.
website	Set the website configuration for a bucket.

Shared file system

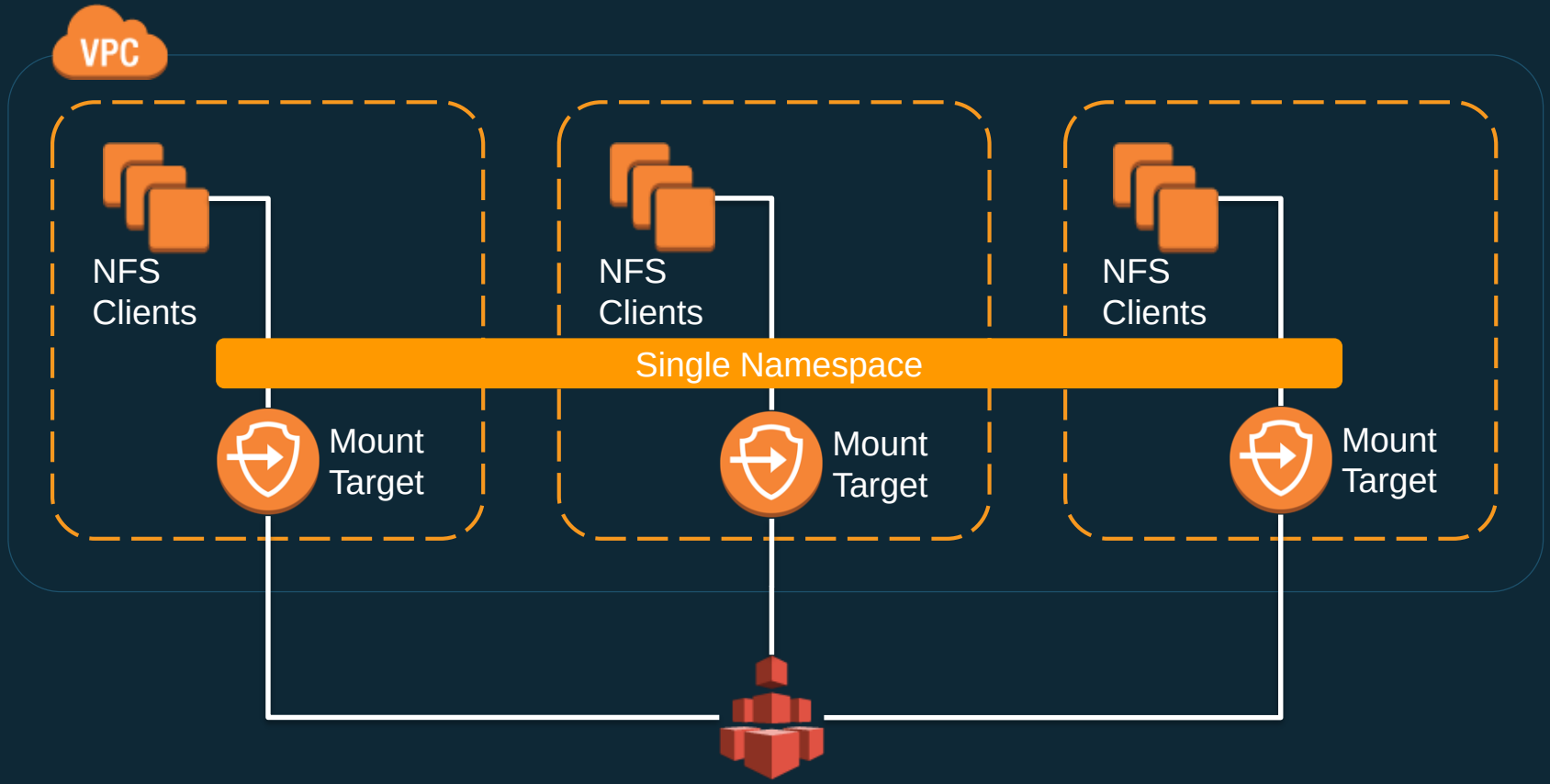
Elastic File System (EFS)

- Provides simple, scalable, highly available, & durable file storage in the cloud
- Petabyte scale file system distributed across an unconstrained number of storage servers in multiple Availability Zones (AZs)
- Elastic capacity, automatically growing & shrinking as you add & remove files
- NFS v4–based
- Accessible from on-premise servers

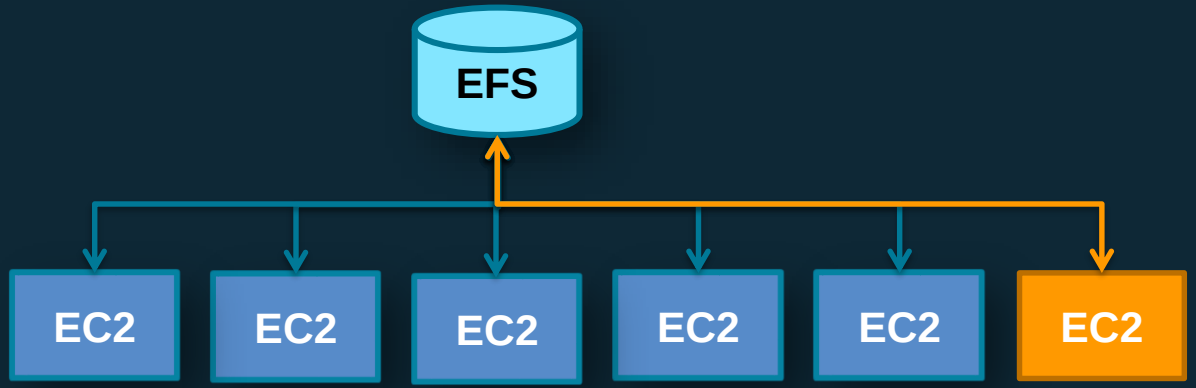
Do it yourself NFS architecture



Amazon EFS architecture



EFS – Mounting



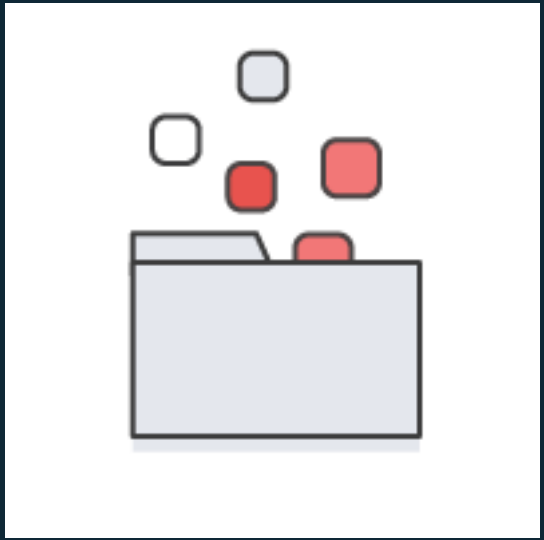
EFS DNS Name

availability-zone.*file-system-id*.efs.*aws-region*.amazonaws.com

Mount on machine

sudo mount -t nfs4 mount-target-DNS:/ ~/efs-mount-point

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

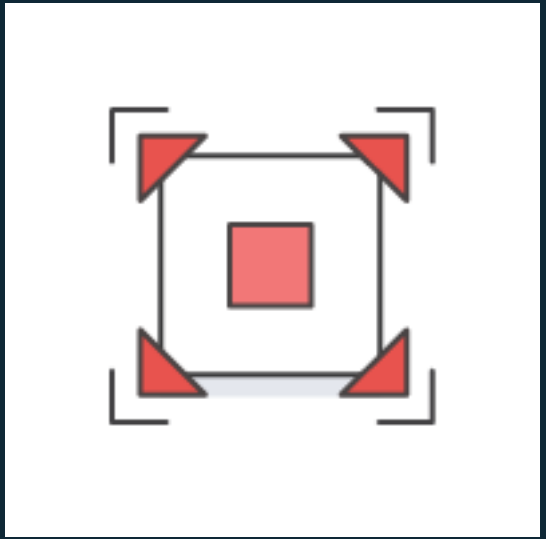
- NFS v4.1—widespread, open
- Standard file system access semantics
- Works with standard OS file system APIs

Simple pricing = simple forecasting

2 Amazon EFS is Elastic

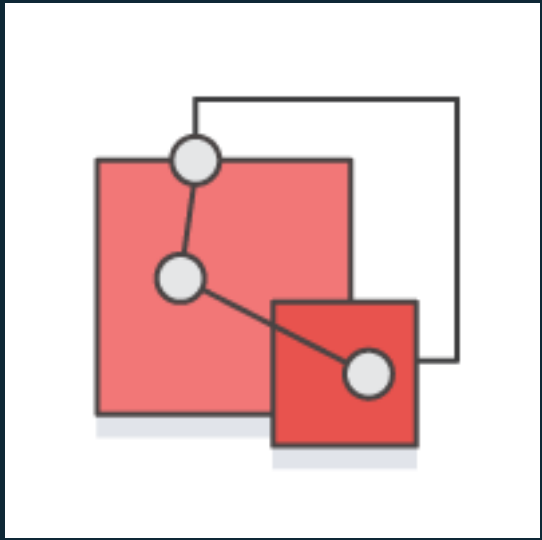
- File systems grow and shrink automatically as you add and remove files
- No need to provision storage capacity or performance
- You pay only for the storage space you use, with no minimum fee

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
- Resources aggregated across multiple AZ’s
- Superior to traditional NAS availability models
- Appropriate for Production / Tier 0 applications

Example use cases

Big Data Analytics

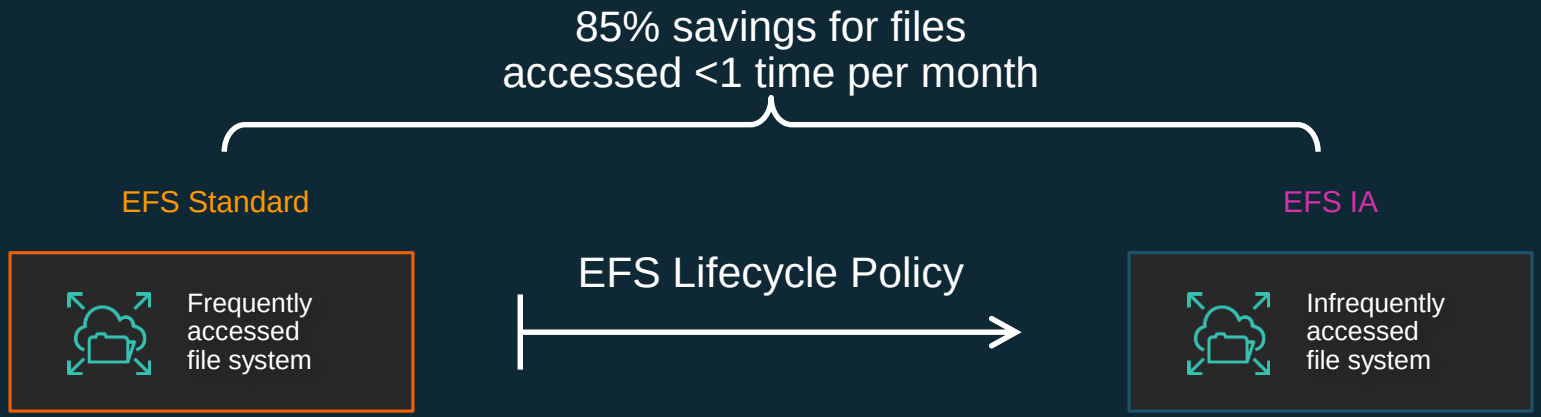
Media Workflow Processing

Web Serving

Content Management

Home Directories

EFS Infrequent Access



Lower cost storage for EFS file data
as it becomes less frequently accessed

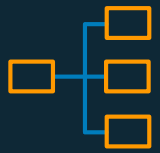
Amazon FSx for Windows File Server

Lift and shift your Windows file storage with fully managed windows file servers

Native Windows compatibility

Fast and flexible performance

Enterprise-ready



Broad accessibility

Fully managed

aws

Amazon FSx for Lustre

Fully managed Lustre file system for compute-intensive workloads

Massively scalable performance

Seamless access to your data repositories

Fully managed

Native file system interface

Cost-optimized for compute-intensive workloads

Secure and compliant

aws

http://lustre.org/

Putting all 3 storage together

- Let’s first have our images on EBS
- Let’s move them to EFS
- Let’s move them to S3

Quiz time ..

joinmyquiz.com

Thank you

Chetan Agrawal – agrcheta@amazon.com

Deven Suri – dsuri@amazon.com