

Copyright © 2013 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.

Errors or corrections? Email us at aws-course-feedback@amazon.com.

Other questions? Email us at aws-training-info@amazon.com.

Architecting with AWS

Data Storage Scaling

Data Storage Scaling | What we'll cover

1

**Data storage
options**

2

Amazon EBS

3

**Instance
storage**

4

**Amazon S3
and Amazon
CloudFront**

Data Storage Scaling | Data Storage Options

1

**Data storage
options**

What we'll cover

1

AWS storage options

2

Best practices for data storage

AWS Storage Options

1. **Block Storage:** Instance Store, Amazon EBS
2. **Object Storage:** Amazon S3, Amazon Glacier, Amazon CloudFront
3. **Sync Volumes:** AWS Storage Gateway
4. **Relational Databases:** Amazon RDS
5. **NoSQL Databases:** Amazon DynamoDB
6. **In-memory Cache:** ElastiCache
7. **Content Cache:** Amazon CloudFront

Best practices on choosing the right data storage solution

- Understand the variety of storage options on AWS
- Access performance, durability, cost, and interface
- POSIX versus Object store—choose where appropriate
- Use multiple cloud storage options—storage hierarchy
- Horizontal versus Vertical scaling
- Be creative. Use storage alternatives like in-memory caches

It's all about performance-oriented and cost-oriented choice.

For review:

- Identify AWS storage options
- Describe best practices on choosing the right data storage solution

Data Storage Scaling | Amazon EBS

2

Amazon EBS



What we'll cover

1

Benefits of using Amazon EBS

2

Standard volumes versus Provisioned IOPS volumes

3

Amazon EBS pricing



Benefits of using Amazon EBS

- POSIX-compliant, network attached, block storage
- Data lifetime independent of Amazon EC2 instance lifetime
- Automatic replication with user-controlled snapshots
- Provisioned performance (up to 4,000 16k IOPS per volume)
- Large data storage capacity (up to 1TB per volume)
- Portability between instances (detach/attach)



POSIX-compliant, network attached

- Each volume is like a hard drive on a physical server
- Attach multiple volumes to an Amazon EC2 instance
- Volumes cannot be shared with multiple Amazon EC2 instances
- Ideal for: OS boot device; file systems; databases; raw block devices



EBS Performance

	Standard Amazon EBS	Provisioned IOPS Amazon EBS
IOPS	100 IOPS steady-state, with best-effort bursts to hundreds	Within 10% of up to 4,000 IOPS, 99.9% of the time, as provisioned
Throughput	Best effort to 10's of MB/sec	16 KB per IO = up to 64 MB/sec. It can burst up to 40 MB/sec on best effort basis.
Latency	Reads typically <20ms writes typically <10ms	Each IO has a service time of provisioned IOPS/s

Use EBS-optimized Amazon EC2 instances when attaching Provisioned IOPS volume for dedicated network bandwidth

Stripe multiple volumes for more IOPS (e.g., (10) x 4,000 IOPS volumes in RAID0 for 40,000 IOPS)



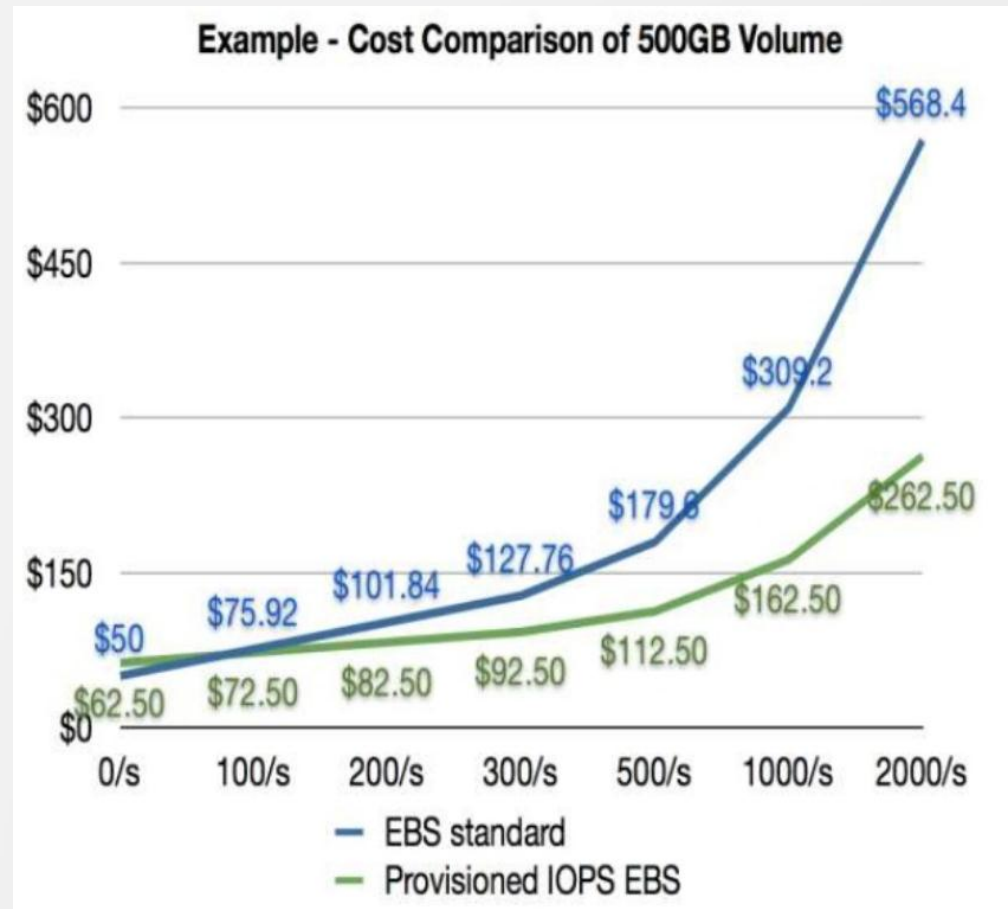
EBS Cost

	Standard	Provisioned IOPS
Storage	\$0.10 per GB-month	\$0.125 per GB-month
IOPS	\$0.10 per 1 million I/O requests	\$0.10 per provisioned IOPS-month
Snapshots	\$0.095 per GB-month of data stored	\$0.095 per GB-month of data stored



Cost Analysis - Standard vs. Provisioned IOPS

- **Steady, predictable IO patterns:** cost effective to use Provisioned IOPS Amazon EBS
- **Bursty IO patterns:** analyze pattern and choose Amazon EBS type.





Native Redundancy; optimized for random IO

- Replicated within single AZ
- .1% - .5% AFR using snapshots
- Optimized for random I/O
- Can be striped using RAID 0 or LVM



Snapshots

- Stored in S3
- May be migrated across regions.
- New volumes can be created from Amazon EBS Snapshots and places in desired Availability Zone.
- AMIs can be created from Amazon EBS Snapshots



EBS Capacity

- **Grow a volume:** snapshot → create new, larger volume from snap
→ detach existing volume → attaching new volume
- Allocate space proactively for root volumes to prevent out-of-space issues.

EBS Capacity

- 1GB to 1TB per volume





EBS Capacity

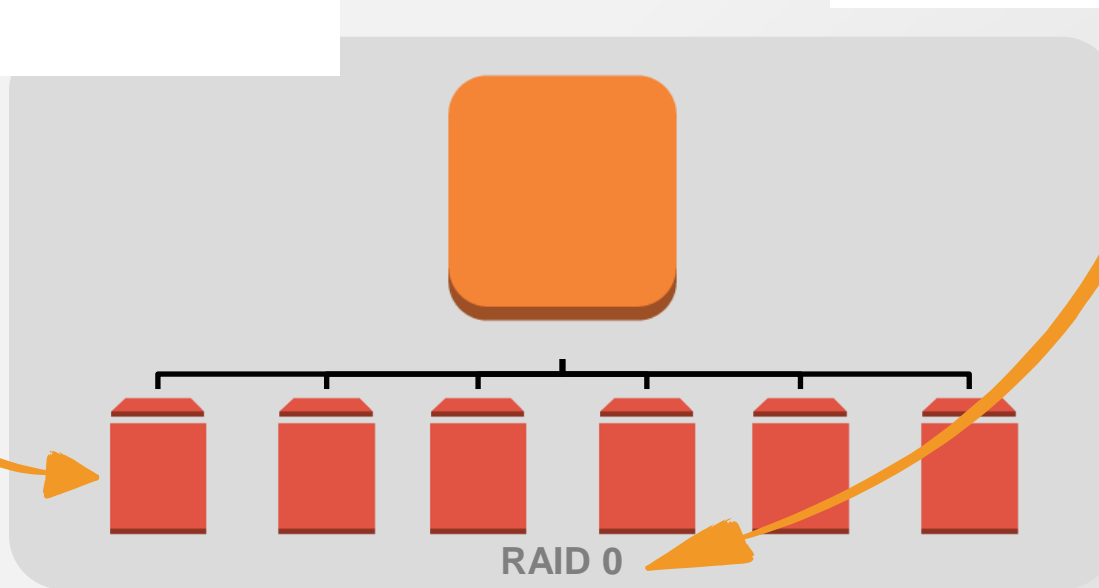
- **Scenario:** 6 TB of block storage at 3,000 IOPS on an EC2 instance

EBS Capacity

- Scenario:** 6 TB of block storage at 3,000 IOPS on an EC2 instance

(6) x 1TB PIOPS EBS Volumes @
500 IOPS/ea

RAID 0 managed at OS level (e.g.,
LVM, mdadm, etc)



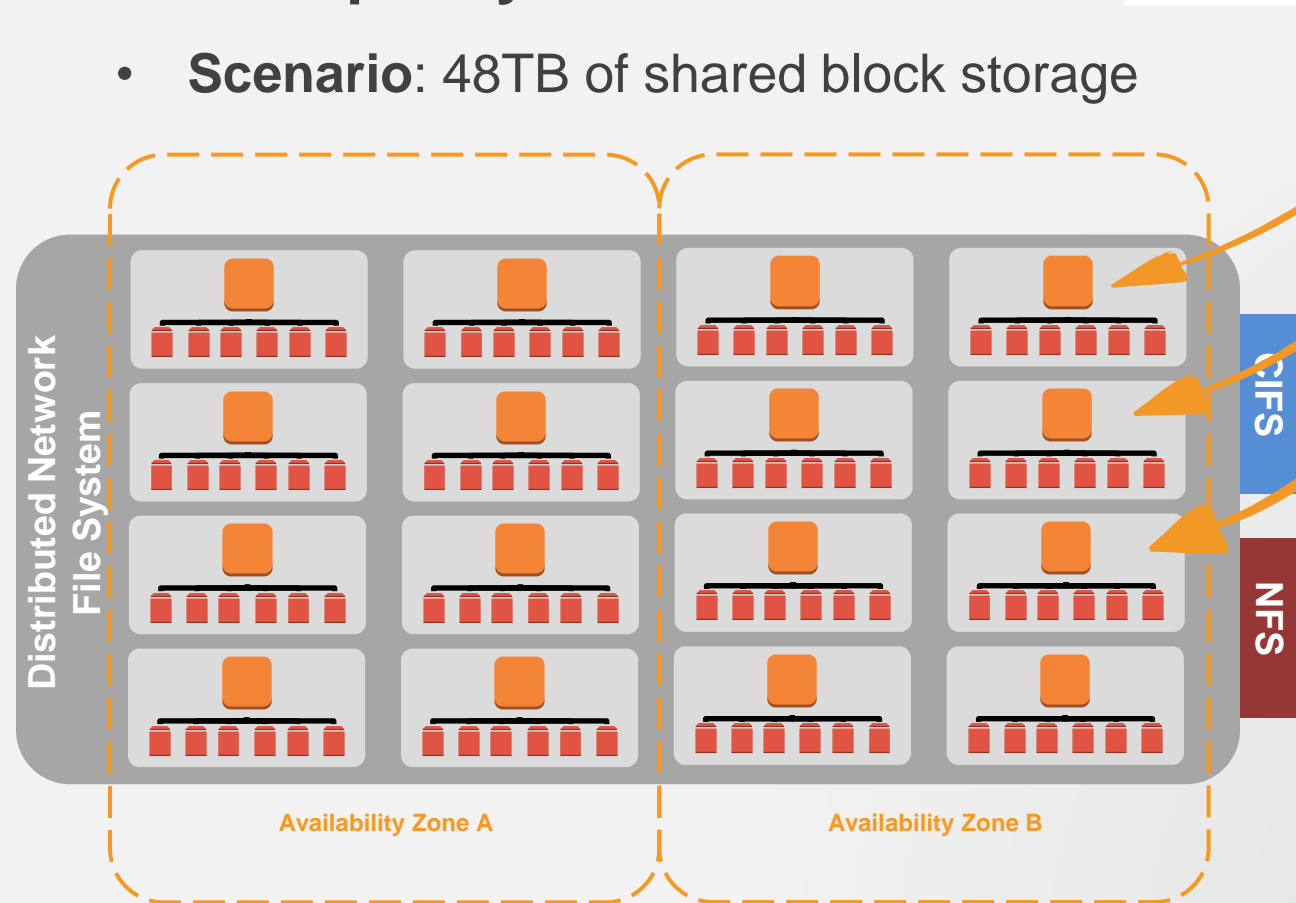
EBS Capacity

- **Scenario:** 48TB of shared block storage

EBS Capacity

- **Scenario:** 48TB of shared block storage

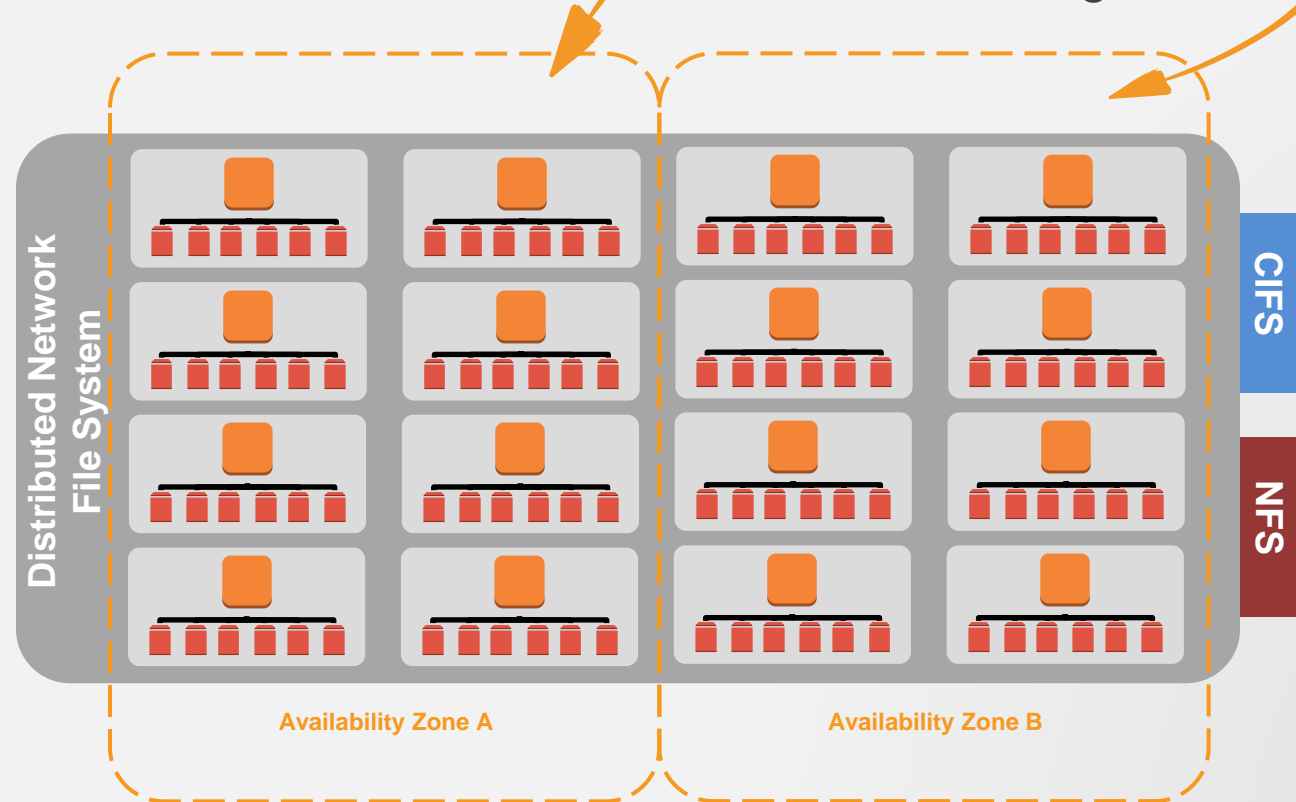
16 EC2 instances - each with (6) x 1TB EBS Volumes – running a DNFS (e.g., GlusterFS, ceph, etc)



EBS Capacity

- **Scenario:** 48TB of shared block storage

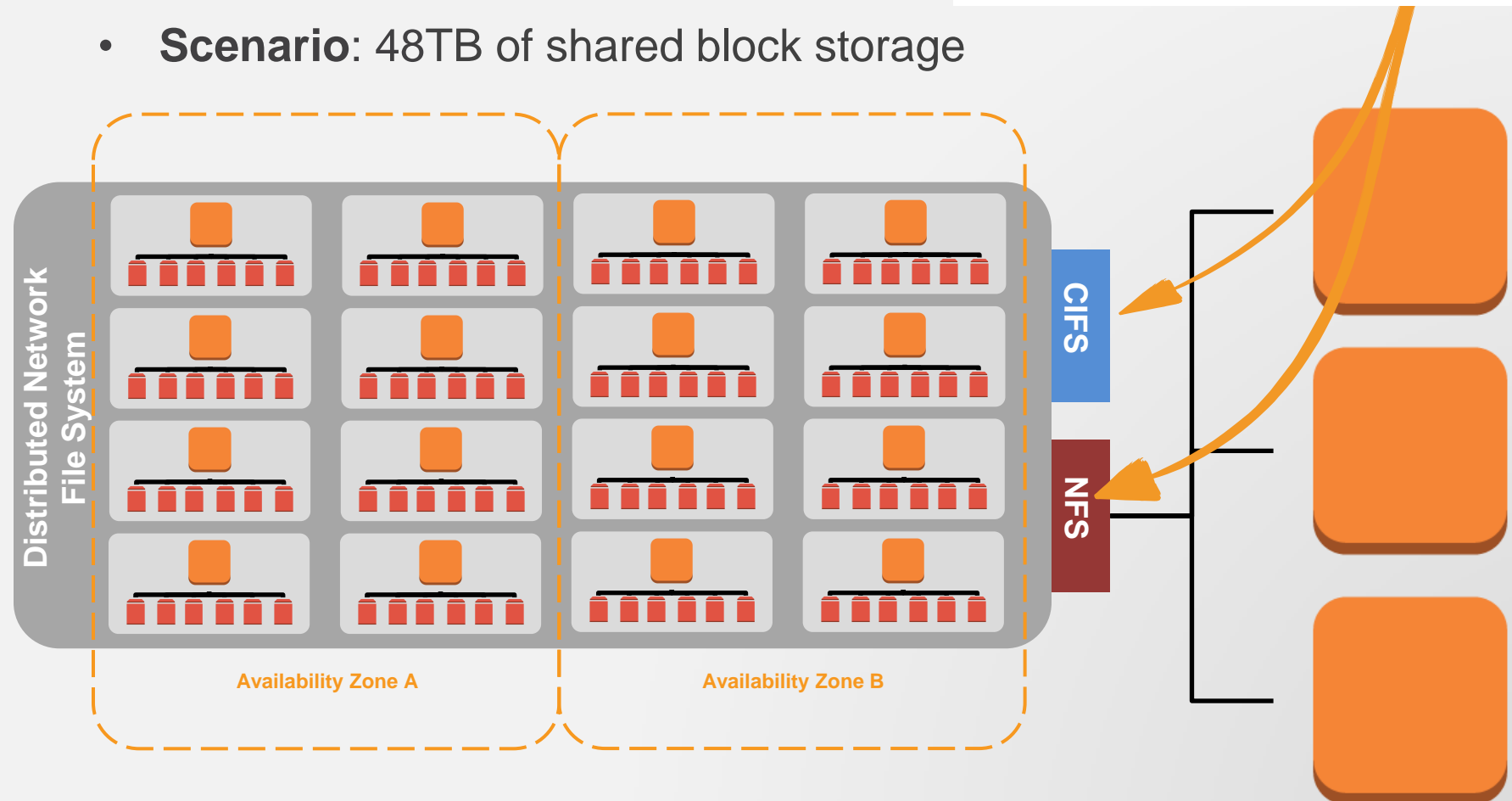
96TB aggregate, 48TB available
after cross-AZ replication



EBS Capacity

- Scenario:** 48TB of shared block storage

48TB exposed over the network
as CIFS, NFS, etc mounts to other
EC2 instances





EBS Anti-Patterns

- Temporary Storage
 - Consider using Amazon EC2 Instance Storage
- Very high durability storage
 - Consider using Amazon S3 or snapshots
- Storing static web content
 - Consider using Amazon S3
- Storing structured data or Key-Value pairs
 - Consider using DynamoDB or Amazon RDS



For review:

- Describe benefits of using Amazon EBS
- Differentiate Amazon EBS Standard volumes from Amazon EBS Provisioned IOPS volume
- Describe Amazon EBS pricing

Data Storage Scaling | Instance Store

3

**Instance
storage**

What we'll cover

1

Benefits of Amazon Instance Store

2

Instance Store Best Practices

Instance Store

- No additional charge beyond your Amazon EC2 Instance
- Number and size of volumes varies by Amazon EC2 instance type
 - Larger instances have larger/more volumes
 - hi1.4xlarge = 2 x 1024GB SSD
 - c1.xlarge = 4 x 450GB
 - t1.micro = none
- Not automatically attached. Must request at instance launch
- Volatile
 - No persistence
 - Data is gone when an Amazon EC2 instance stops, fails or is terminated

Instance Store

- Zero network overhead; local, direct attached resource.
 - No network variability
 - Not optimized for random I/O
 - Generally better for sequential I/O
- Root volume and data volume are lost on physical disk failure, stopping, or terminating of instance
- **Ideal for storing temporary data** like buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

Instance Store

- **High-performance SSD option**
 - hi1.4xlarge EC2 instance type
 - (2) x 1TB SSD local to instance
 - ~120,000 random read IOPS (4 KB blocks)
 - ~10,000-85,000 random write IOPS (4 KB blocks)
- **High-storage**
 - Hs1.8xlarge EC2 instance type
 - (24) x 2TB disks local to instance = 48TiB

Anti Patterns

- Persistent storage
 - Consider Amazon EBS
- Database / Structure Storage
 - Consider Amazon EBS, DynamoDB, etc.
- Shareable storage
 - Local instance storage volumes cannot be shared. Consider Amazon EBS.
- Backups
 - Consider Amazon EBS and Amazon EBS Snapshots

For review:

- Describe benefits of Amazon Instance Store
- Describe best practices of using Instance Store

Data Storage Scaling | Amazon S3 and Amazon CloudFront

4

**Amazon S3
and Amazon
CloudFront**

What we'll cover

- 1 **Amazon S3 and storage classes**
- 2 **Amazon S3 namespaces**
- 3 **Amazon S3 Server-side Encryption**
- 4 **Amazon S3 access controls**
- 5 **Website hosting and Amazon S3**
- 6 **Multi-part upload, object versioning, and server access**
- 7 **Overview of Amazon CloudFront**



Amazon S3 (Simple Storage Service)

- An object store, not a file system
- Write once, read many (WORM)
- Eventually consistent
- 99.999999999% durability
- Unlimited storage capacity. Pay for what you actually use
- Highly scalable and available data storage
- Objects stored in a region never leaves the region unless explicitly transferred
- Storage for backups
- Provides a REST and a SOAP interface



Eventually consistent

- New Objects
 - Synchronously stores your data across multiple facilities before returning SUCCESS
 - Read-after-write consistency*

*except US-STANDARD region



Eventually consistent

- Updates
 - Write then read: could report key does not exist
 - Write then list: might not include key in list
 - Overwrite then read: old data could be returned



Eventually consistent

- Deletes
 - Delete then read: could still get old data
 - Delete then list: deleted key could be included in list



Amazon S3 Storage Classes

- Standard
 - Designed to provide 99.9999999999% durability and 99.99% availability of objects over a given year
 - Designed to sustain the concurrent loss of data in two facilities
 - Objects you want to have high durability e.g. master copy of movie media



Amazon S3 Storage Classes

- Reduced Redundancy Storage (RRS)
 - Reduces costs by storing data at lower levels of redundancy than the Standard storage
 - Designed to provide 99.99% durability and 99.99% availability of objects over a given year
 - Objects you can afford to lose or can recreate e.g. different encodings of movie media



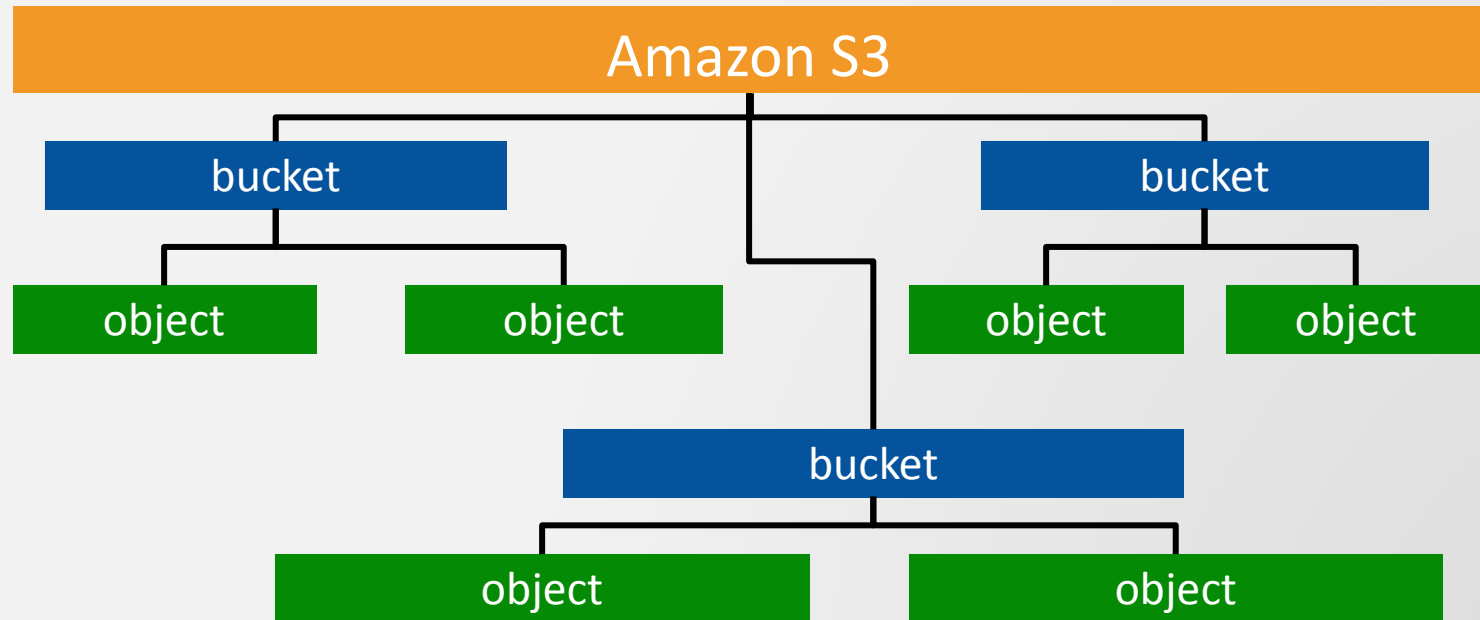
Amazon S3 Storage Classes

- Glacier
 - Suitable for archiving data.
 - Data retrieval time is 3-5 hour
 - Designed for 99.999999999% durability of archives
 - Cost effective - Write-once, read-never.
 - 1c per GB/Month ~ \$120 per TB/Year
 - Pay for accessing data



Buckets, objects and keys

- Bucket name = globally unique
- Maximum of 100 buckets with unlimited object capacity



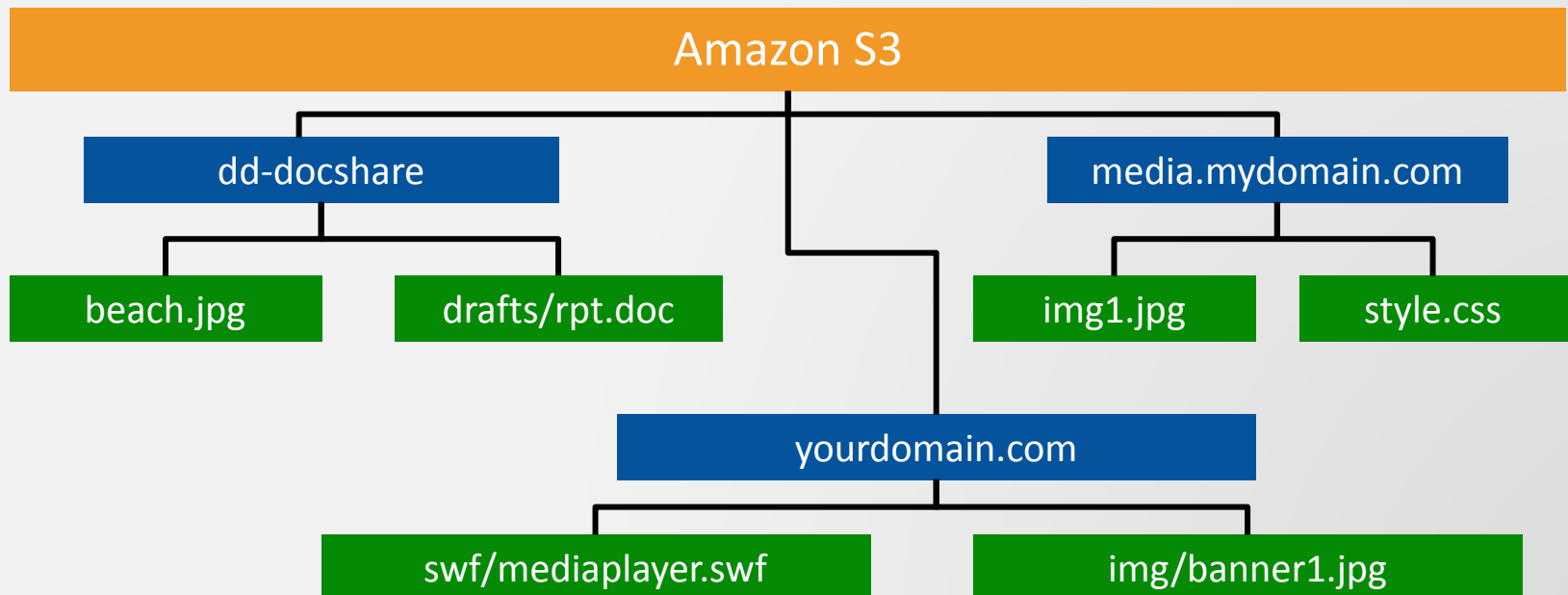


Buckets, objects and keys

- Object key = unique within a bucket
- Bucket name + object name (key) = globally unique
 - Max 1024 bytes UTF-8
 - Including 'path' prefixes

drafts/rpt.doc

this is an object key





Server Side Encryption (SSE)

- Automatic encryption of data at rest
- Durable - Amazon S3 key storage
- Secure - 3-way simultaneous access
- Self managed - No need to manage a key store
- Strong - AES-256
- Simple - Additional PUT header



Access Controls

- Use Amazon S3 policies, ACLs or IAM to define rules
- Apply policies to buckets and objects

IAM

vs

Bucket Policies

vs

ACLs

Fine grained

Administer as part of role
based accessApply policies to Amazon S3
at role, user & group level

Fine grained

Apply policies at the bucket
level in Amazon S3Incorporate user restrictions
without using IAM

Coarse grained

Apply access control rules at
the bucket and/or object
level in Amazon S3

Allow

Actions
PutObjectResource
arn:aws:s3:::mybucket/*

Bob

Jane

Allow

Bob, Jane

Actions
PutObjectResource
arn:aws:s3:::mybucket/*

mybucket

Allow

Everyone, Bob, Jane

Actions
Read

mybucket

myobject



Sample Bucket Policy

```
{ "Statement": [{  
    "Effect": "Allow",  
    "Principal": { "AWS": [ "4649-6425", "5243-0045" ] },  
    "Action": "*",  
    "Resource": "/mybucket/*",  
    "Condition": {  
        "IpAddress": { "AWS:SourceIp": "176.13.0.0/12" }  
    }  
}] }
```

Accounts to allow

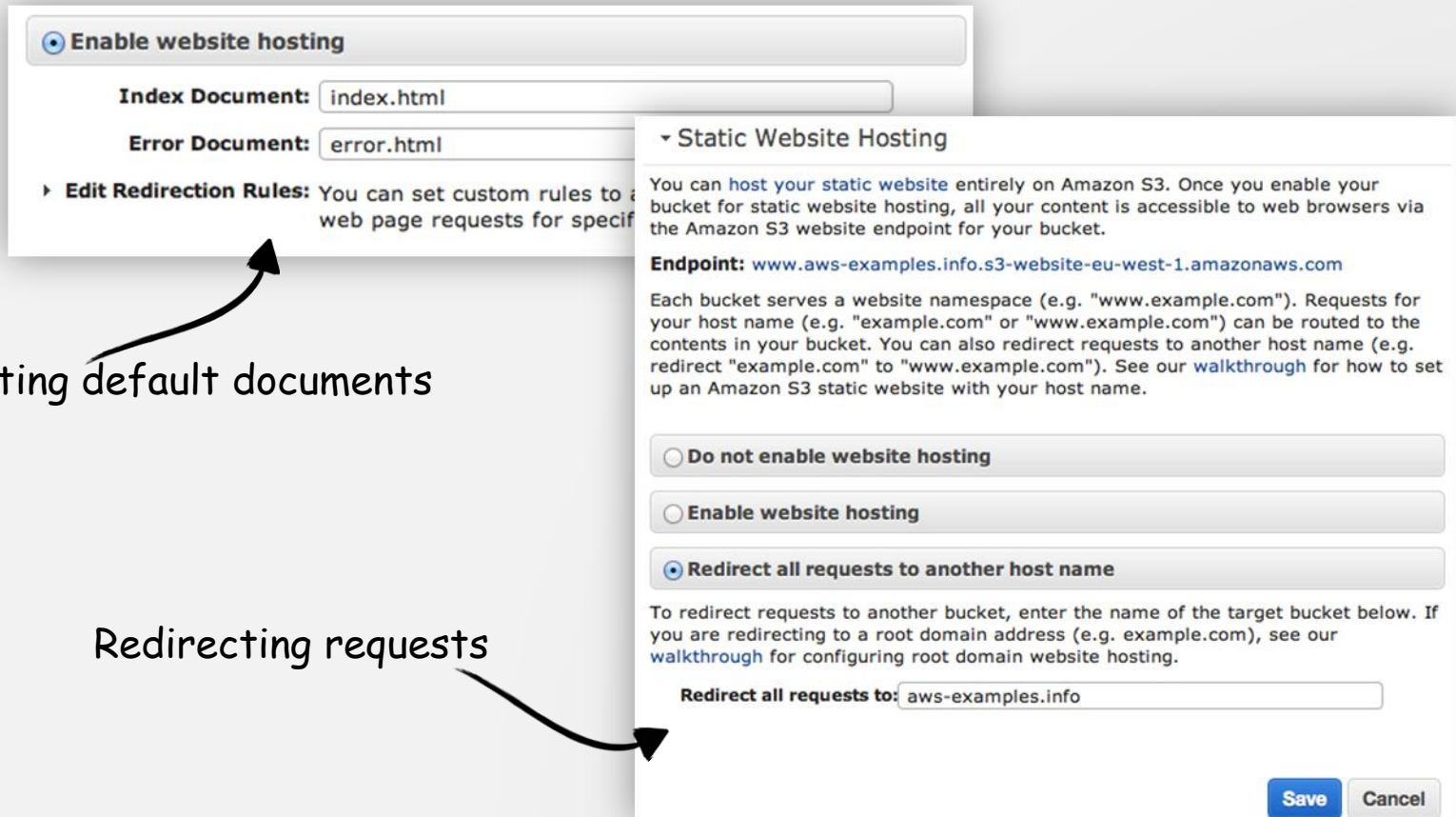
Resource

Source address to allow



Website hosting using Amazon S3

- Static Sites with client-side scripts



Enable website hosting

Index Document:

Error Document:

► **Edit Redirection Rules:** You can set custom rules to redirect web page requests for specific paths.

Static Website Hosting

You can [host your static website](#) entirely on Amazon S3. Once you enable your bucket for static website hosting, all your content is accessible to web browsers via the Amazon S3 website endpoint for your bucket.

Endpoint: www.aws-examples.info.s3-website-eu-west-1.amazonaws.com

Each bucket serves a website namespace (e.g. "www.example.com"). Requests for your host name (e.g. "example.com" or "www.example.com") can be routed to the contents in your bucket. You can also redirect requests to another host name (e.g. redirect "example.com" to "www.example.com"). See our [walkthrough](#) for how to set up an Amazon S3 static website with your host name.

☐ Do not enable website hosting

☐ Enable website hosting

☒ **Redirect all requests to another host name**

To redirect requests to another bucket, enter the name of the target bucket below. If you are redirecting to a root domain address (e.g. example.com), see our [walkthrough](#) for configuring root domain website hosting.

Redirect all requests to:

Save **Cancel**

Setting default documents

Redirecting requests



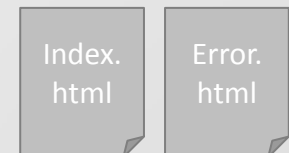
Record set for:
`aws-examples.info`

CNAME for `www.` to:
`www.aws-examples.info.s3-
website-eu-west-
1.amazonaws.com`



A Record 'Alias' to Amazon S3
website:

`aws-examples.info @ s3-website-
eu-west-1.amazonaws.com`



Website redirect to:
`aws-examples.info`

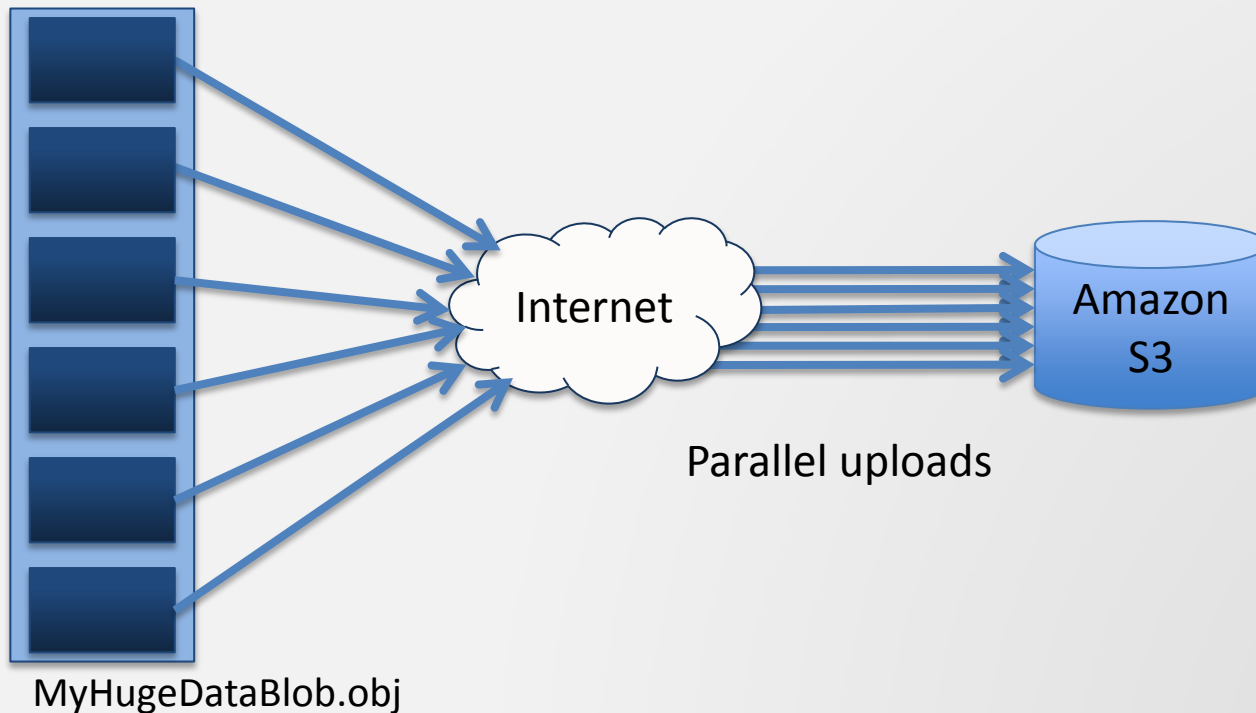
Website bucket name:
`www.aws-examples.info`

Website bucket name:
`aws-examples.info`



Multipart Upload

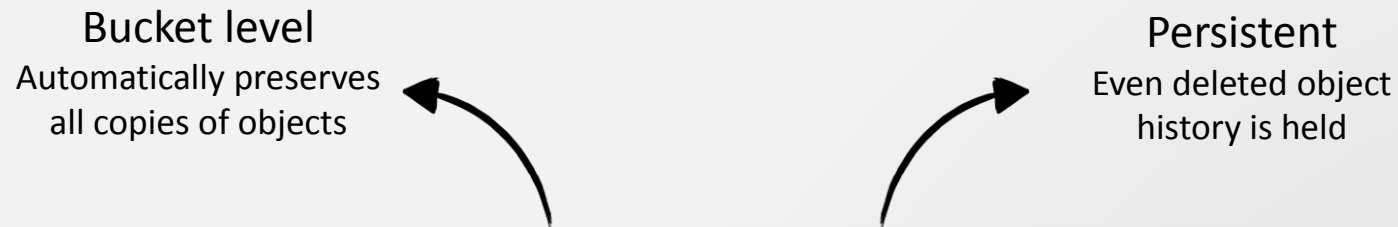
- Upload as 100 MB per part
- Multipart Upload supports a maximum of 10,000 parts





Amazon S3 Object Versioning

- Preserve object histories



▼ Versioning

Versioning allows you to preserve, retrieve, and restore every version of every object stored in this bucket. This provides an additional level of protection by providing a means of recovery for accidental overwrites or deletions.

Once enabled, Versioning cannot be disabled.

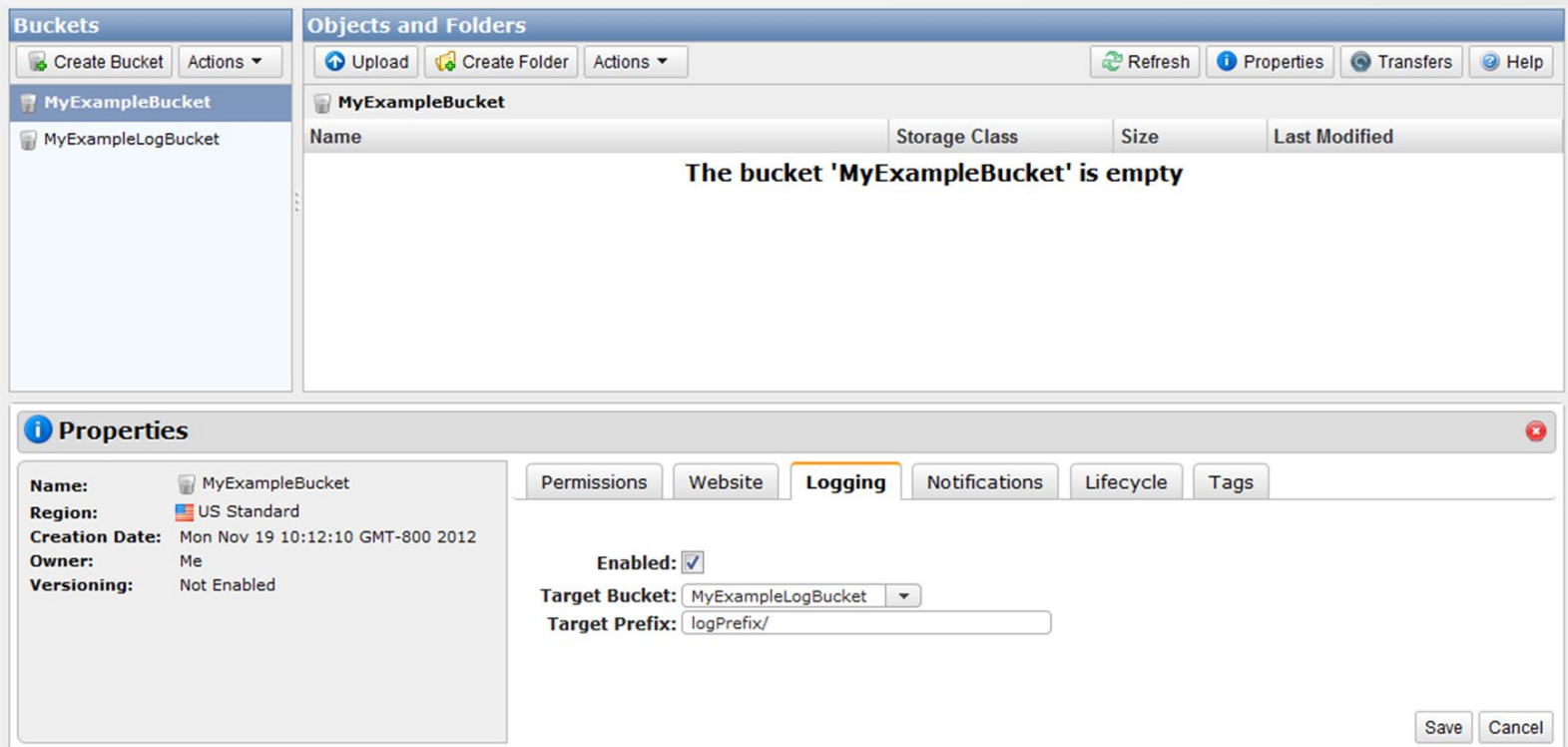
☒ Enabled ☐ Suspended

Save Cancel



Service Access Logging

- Enable logging on the target bucket and choose location where logs are stored.
- Object Lifecycle – Object Archival or Object Expiration



Buckets

Create Bucket Actions

MyExampleBucket

MyExampleLogBucket

Objects and Folders

Upload Create Folder Actions Refresh Properties Transfers Help

MyExampleBucket

Name	Storage Class	Size	Last Modified
The bucket 'MyExampleBucket' is empty			

Properties

Name: MyExampleBucket

Region: US Standard

Creation Date: Mon Nov 19 10:12:10 GMT-800 2012

Owner: Me

Versioning: Not Enabled

Permissions Website **Logging** Notifications Lifecycle Tags

Enabled: ☒

Target Bucket: MyExampleLogBucket

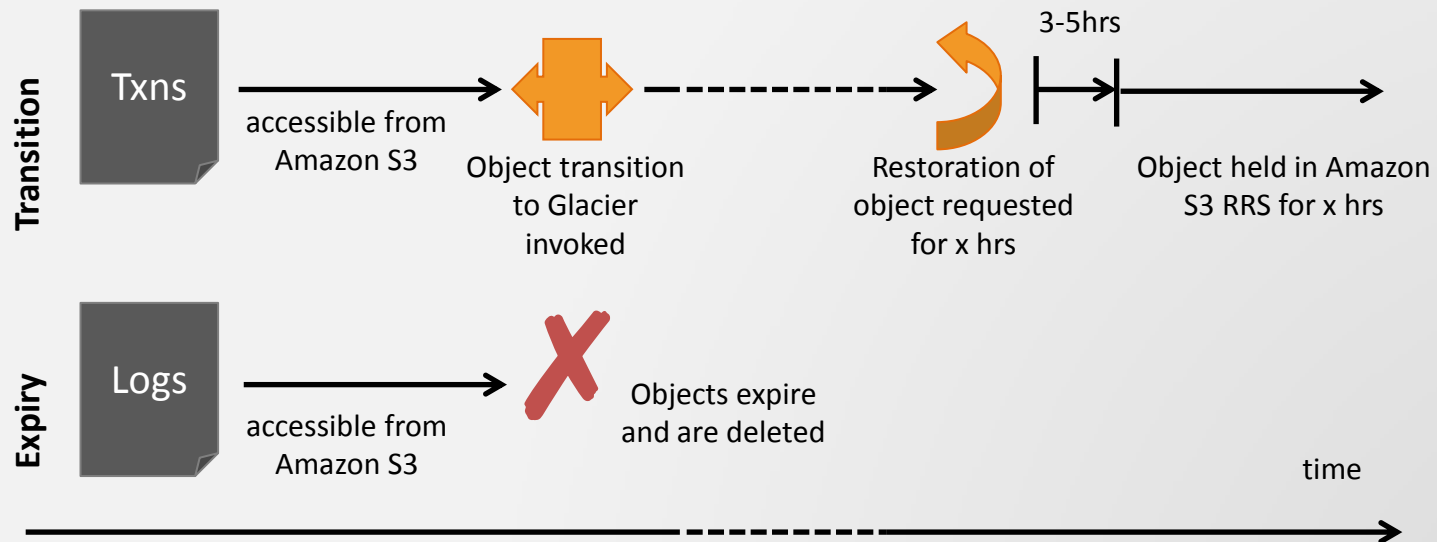
Target Prefix: logPrefix/

Save Cancel



Object Lifecycle Management

- Automated management of objects
- Object Expiration - Permanently delete objects from Amazon S3
- Object Archival - Move objects to Glacier and out of Amazon S3 storage





Object Lifecycle Management

- Considerations before archiving objects
 - Not available in real time
 - Transition action is only one-way
 - Visible and available only through Amazon S3
 - Rule with an empty key prefix
- Before You Decide to Expire Objects
 - Expiration action deletes objects
 - Rule with an empty key prefix

Amazon CloudFront

- Common Use Cases
 - Delivering your entire website or web application by caching static content, proxy requests for dynamic content to the origin server.
 - Distributing software or other large files. Amazon CloudFront also offers lower prices than Amazon S3 at higher usage tiers.
 - Delivering media files using streaming of pre-recorded media and live events.

For review

- Describe Amazon S3 and storage classes
- Describe Amazon S3 namespaces
- Describe Amazon S3 Server Side Encryption
- Define Amazon S3 access controls
- Describe website hosting using Amazon S3
- Describe Multipart Upload, Object Versioning, Server Access Logging, Object Lifecycle Management in Amazon S3
- Describe Amazon CloudFront

Identify the correct statements:

Multiple EBS Volumes may be attached to the same EC2 instance.

EC2 Instance Store (ephemeral volumes) are a local resource directly attached to the host that an EC2 instance runs in.

Because of its scalable and durable nature, S3 is a good candidate for hosting live database transaction log and data files.

CloudFront supports caching of static objects as well as RTMP and RTMPE streaming and HTTP progressive download.

EBS volumes have a maximum size of 2TB.

Sequential IO is a better fit for EBS volumes compared to EC2 Instance Store (ephemeral volumes).

Frequent EBS snapshots increase EBS volume durability.

Multiple EC2 instances with multiple EBS volumes can be pooled to create a multi-petabyte Distributed Network File System (DNFS)