Session 7

1. Finish WordPress Install
2. Demo on DNS Routing
3. Repeat Eventual Consistency vs Strong Consistency

DynamoDB supports *eventually consistent* and *strongly consistent* reads.

**Eventually Consistent Reads**

**When you read data from a DynamoDB table, the response might not reflect the results of a recently completed write operation**. The response might include **some stale data**, for a maximum duration of 20 seconds. If you repeat your read request after a short time, the response should return the latest data.

**Strongly Consistent Reads**

When you request a strongly consistent read, **DynamoDB returns a response with the most up-to-date data,** reflecting the updates from all prior write operations that were successful. A strongly consistent read might not be available if there is a network delay or outage. Strongly consistent reads are not supported on global secondary indexes.

**S3 – Basics Theory-PP**

Create S3 Bucket-Demo

By Default all Buckets Created are Private. Control access using

**Bucket Policies**

**Access Control Lists**

**S3 Buckets🡪 can have access logs that can be created in and saved in your bucket or another bucket**

**Encryption in Transit**🡪 HTTPS Traffic, SSL/TLS-anytime you access using HTTPS

**Encryption at Rest (Server Side)** is achieved by:

AWS provides three ways to protect your data at rest in S3 using server-side encryption:

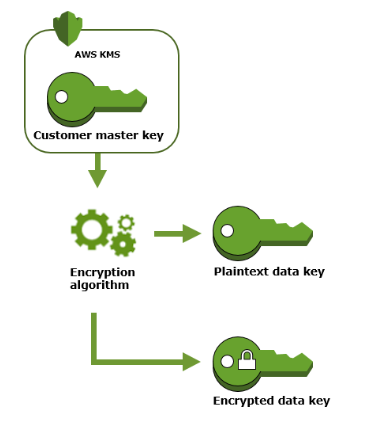
* **SSE-S3 (default) Managed Keys where Amazon Manages Keys for you**
* **SSE with customer provided keys (SSE-C)- Where you give Amazon your keys**
* **SSE with AWS KMS (SSE-KMS)- Both you and Amazon Manage Keys together**

**SSE-S3 encrypts data at rest using 256-bit Advanced Encryption Standard(AES-256**).

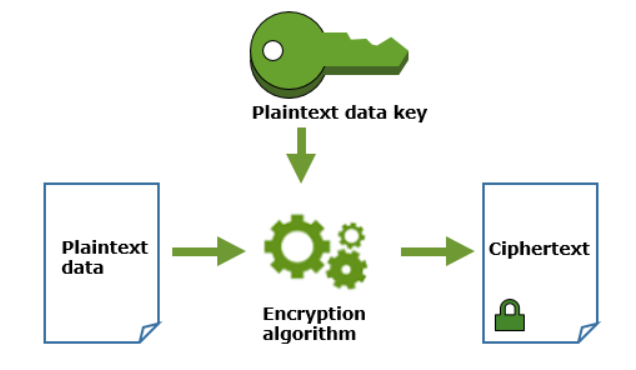
**Each object is encrypted with a unique data/object key and each data/object key is further encrypted using a master key (envelope encryption) which is regularly rotated so as to prevent data getting compromised**.

How does S3 Server-Side Encryption Work?

1. To encrypt the objects you need a data key.
2. Next to generate a data key you can specify a **CMK (Customer Master Key**) that you have already created otherwise S3 will request AWS KMS to create a default CMK which can be used to create a data key.
3. **CMK using the encryption algorithm (AES-256) creates two keys**, one is plaintext data key and the other is encrypted data key.



1. **S3 encrypts the object with plaintext data key and deletes the key from memory**. *The encrypted object along with the encrypted data key is then stored in S3*



1. While retrieving the object S3 sends the encrypted data key to KMS. 3 then KMS matches the correct CMK then it decrypts the encrypted data key and sends the plaintext data key to S3. S3 retrieves the object by decrypting the object with this plaintext data key.

**S3-Version Control- Demo**

Stores all Versions of an Object – all writes and also even if you delete the object, great tool for backing up objects

Versioning once enabled on a bucket cannot be disabled, can only be suspended

Integrates with LifeCycle Rule

Demo- Create a new bucket, Turn on Versioning, Create Text1.txt, Update Versions, Upload

**LifeCycle Management/Glacier -Demo**

**Cross Region Replication-Demo**

**Transfer Acceleration**

Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. Transfer Acceleration takes advantage of Amazon CloudFront’s globally distributed edge locations. As the data arrives at an edge location, data is routed to Amazon S3 over an optimized network path.

When using Transfer Acceleration, additional data transfer charges may apply

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

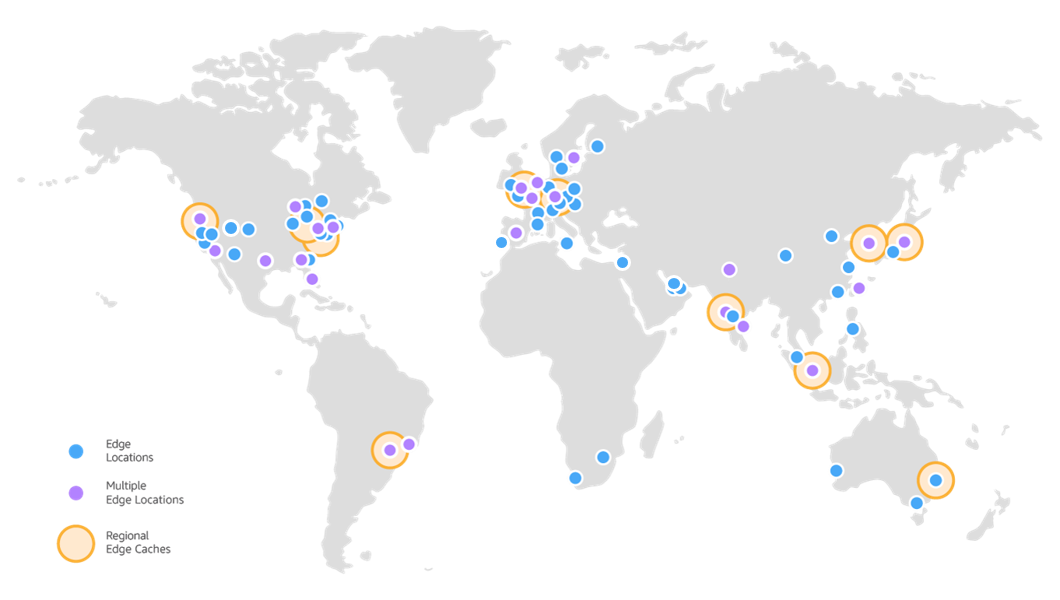
**CloudFront-Theory**

**Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data**, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services

## **Amazon CloudFront Infrastructure**

The Amazon CloudFront Global Edge Network

To deliver content to end users with lower latency, Amazon CloudFront uses a global network of 190 Points of Presence (179 Edge Locations and 11 Regional Edge Caches) in 72 cities across 33 countries. Amazon CloudFront Edge locations are located in:



**Edge Location 🡪 location where the content will be cached**

**Origin🡪 The source or origin of the files that CDN will distribute, can be S3 Bucket, EC2 Instance ,ELB or Route53**

**Distribution🡪 Name of CDN that consists of a collection of Edge Locations**



**Two Kinds of Distribution**

**RTMP- Used for Media Streaming**

**Web Distribution- for Web Location**

Edge Locations- You can write to them and objects live based on TTL.

Cloud Front Demo

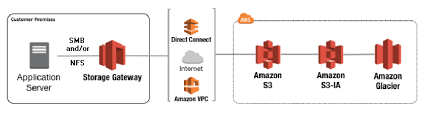
**StorageGateway**

AWS Storage Gateway connects an on-premises software appliance with cloud-based storage to provide seamless integration with data security features between your on-premises IT environment and the AWS storage infrastructure. You can use the service to store data in the AWS Cloud for scalable and cost-effective storage that helps maintain data security.

AWS Storage Gateway offers file-based, volume-based, and tape-based storage solutions:

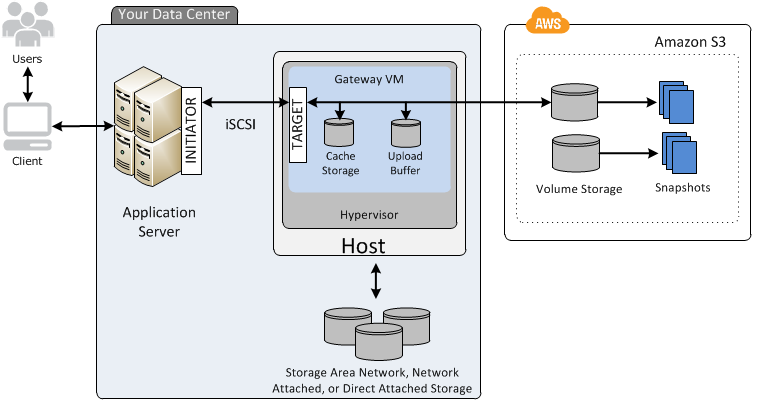
**File Gateway**

A file gateway supports a file interface into Amazon Simple Storage Service (Amazon S3) and combines a service and a virtual software appliance. By using this combination, you can store and retrieve objects in Amazon S3 using industry-standard file protocols such as Network File System (NFS) and Server Message Block (SMB). The software appliance, or gateway, is deployed into your on-premises environment as a virtual machine (VM) running on VMware ESXi or Microsoft Hyper-V hypervisor.

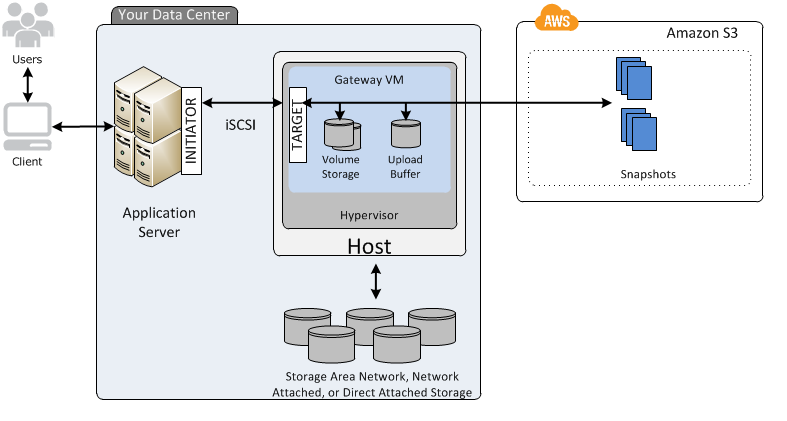


**Volume Gateway** – A volume gateway provides cloud-backed storage volumes that you can mount as Internet Small Computer System Interface (iSCSI) devices from your on-premises application servers. The gateway supports the following volume configurations:

* **Cached volumes** – **You store your data in Amazon Simple Storage Service (Amazon S3) and retain a copy of frequently accessed data subsets locally**. Cached volumes offer a substantial cost savings on primary storage and minimize the need to scale your storage on-premises. You also retain **low-latency access to your frequently accessed data**.



* **Stored volumes** – **If you need low-latency access to your entire dataset, first configure your on-premises gateway to store all your data locally**. Then asynchronously back up point-in-time snapshots of this data to Amazon S3. This configuration provides durable and inexpensive offsite backups that you can recover to your local data center or Amazon EC2. For example, if you need replacement capacity for disaster recovery, you can recover the backups to Amazon EC2.



**Tape Gateway** – With a tape gateway, you can cost-effectively and durably archive backup data in GLACIER or DEEP\_ARCHIVE. A tape gateway provides a virtual tape infrastructure that scales seamlessly with your business needs and eliminates the operational burden of provisioning, scaling, and maintaining a physical tape infrastructure.

**You can run AWS Storage Gateway either on-premises as a VM appliance, as a hardware appliance, or in AWS as an Amazon Elastic Compute Cloud (Amazon EC2) instance.** You deploy your gateway on an EC2 instance to provision iSCSI storage volumes in AWS. You can use gateways hosted on EC2 instances for disaster recovery, data mirroring, and providing storage for applications hosted on Amazon EC2.