AWS S3 Overview

- Amazon S3 allows people to store objects (files) in “buckets” (directories)
- Buckets must have a **globally unique name**
- Objects (files) have a Key. The key is the **FULL** path:
  - <my_bucket>/**my_file.txt**
  - <my_bucket>/**my_folder1/another_folder/my_file.txt**
- This will be interesting when we look at **partitioning**
- Max object size is 5TB
- Object Tags (key / value pair – up to 10) – useful for security / lifecycle

AWS S3 for Machine Learning

- Backbone for many AWS ML services (example: SageMaker)
- Create a “Data Lake”
  - Infinite size, no provisioning
  - 99.999999999% durability
  - Decoupling of storage (S3) to compute (EC2, Amazon Athena, Amazon Redshift Spectrum, Amazon Rekognition, and AWS Glue)
- Centralized Architecture
- Object storage => supports any file format
- Common formats for ML: CSV, JSON, Parquet, ORC, Avro, Protobuf

AWS S3 Data Partitioning

- Pattern for speeding up range queries (ex: AWS Athena)

- By Date: s3://bucket/my-data-set/year/month/day/hour/data_00.csv
- By Product: s3://bucket/my-data-set/product-id/data_32.csv

- You can define whatever partitioning strategy you like!
- Data partitioning will be handled by some tools we use (e.g. AWS Glue)

S3 Storage Tiers

- Amazon S3 Standard - General Purpose
- Amazon S3 Standard-Infrequent Access (IA)
- Amazon S3 One Zone-Infrequent Access
- Amazon S3 Intelligent Tiering
- Amazon Glacier

S3 Storage Tiers Comparison

	Standard	Standard - Infrequent Access	One - Infrequent Access	S3 Intelligent-Tiering	Glacier
Durability	99.999999999%	99.999999999%	99.999999999%	99.999999999%	99.999999999%
Availability	99.99%	99.9%	99.5%	99.90%	NA
AZ	≥3	≥3	1	≥3	≥3
Concurrent facility fault tolerance	2	2	0	1	1

Frequently accessed   Infrequently accessed   Intelligent (new!)   Archives

S3 Lifecycle Rules

- Set of rules to move data between different tiers, to save storage cost
- **Example: General Purpose => Infrequent Access => Glacier**

- **Transition actions**: objects are transitioned to another storage class.
  - Move objects to Standard IA class 60 days after creation
  - And move to **Glacier** for archiving after 6 months

- **Expiration actions:** S3 deletes expired objects on our behalf
  - Access log files can be set to delete after a specified period of time

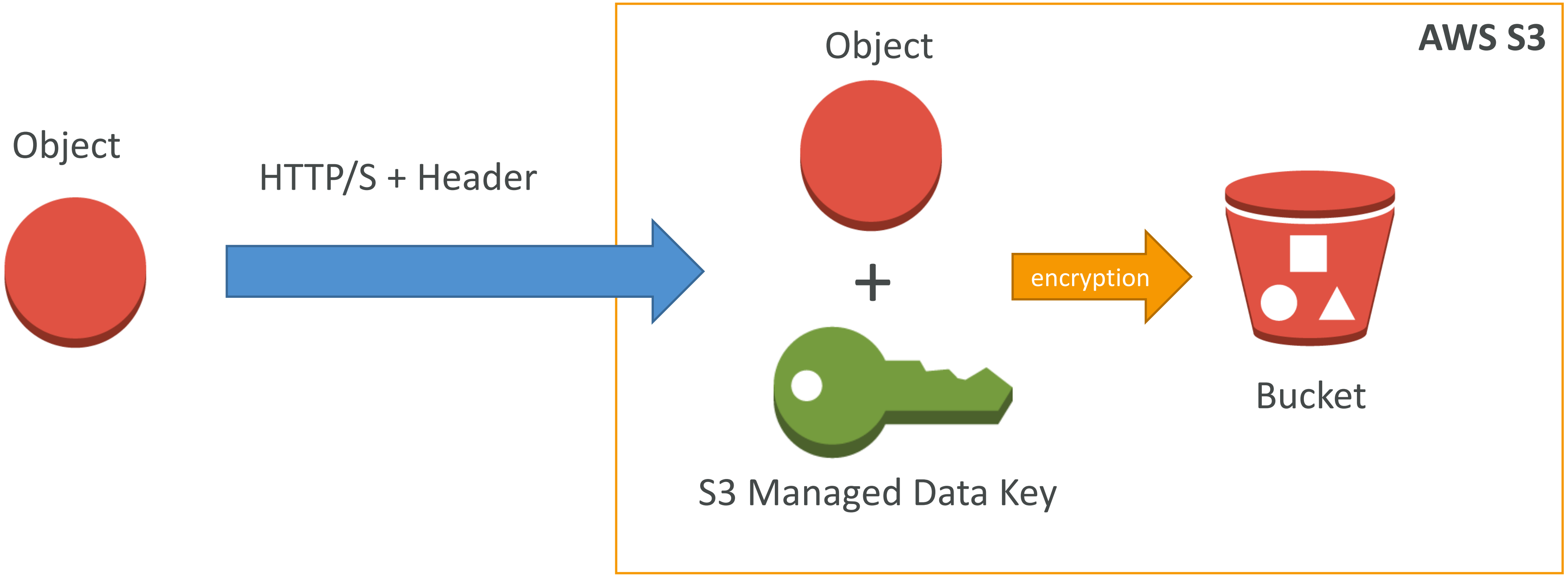
S3 Encryption for Objects

- There are 4 methods of encrypting objects in S3

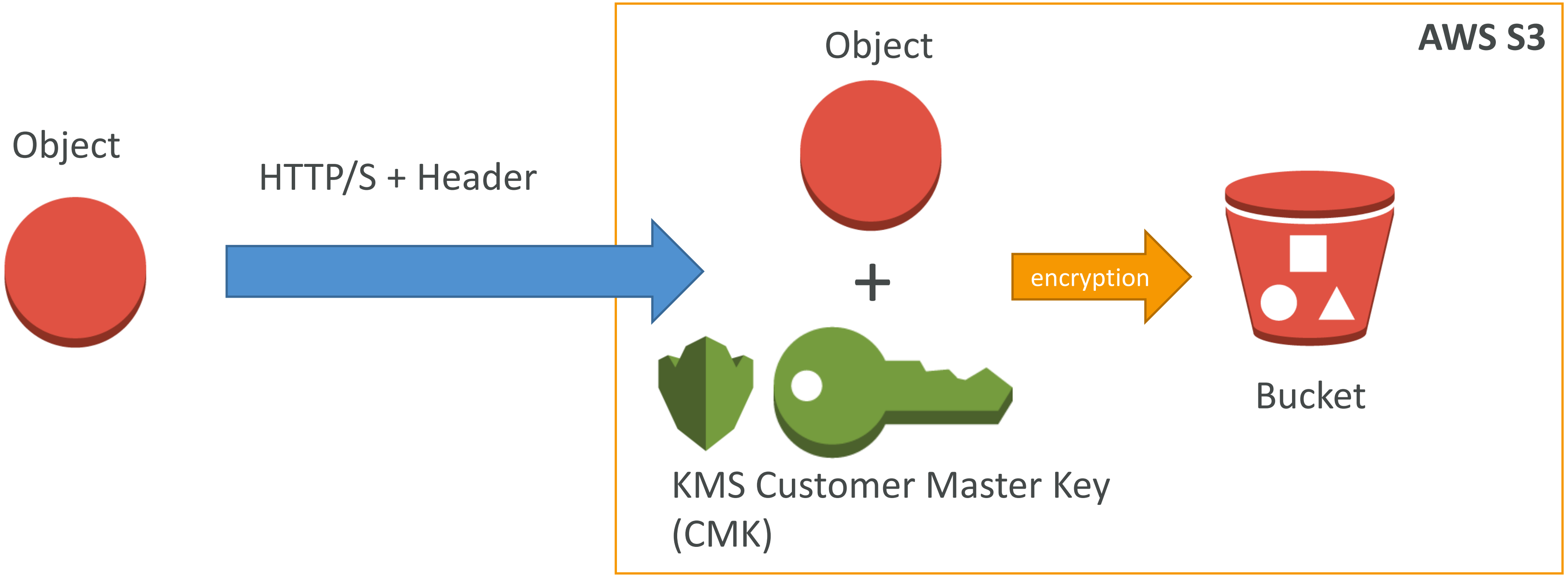
- **SSE-S3:** encrypts S3 objects using keys handled & managed by AWS
- **SSE-KMS:** use AWS Key Management Service to manage encryption keys
  - Additional security (user must have access to KMS key)
  - Audit trail for KMS key usage
- **SSE-C:** when you want to manage your own encryption keys
- **Client Side Encryption**

- **From an ML perspective, SSE-S3 and SSE-KMS will be most likely used**

SSE-S3


Object
HTTP/S + Header
AWS S3
Object
+
S3 Managed Data Key
encryption
Bucket

SSE-KMS


Object
HTTP/S + Header
AWS S3
Object
+
encryption
Bucket
KMS Customer Master Key
(CMK)

S3 Security

- User based
  - IAM policies - which API calls should be allowed for a specific user

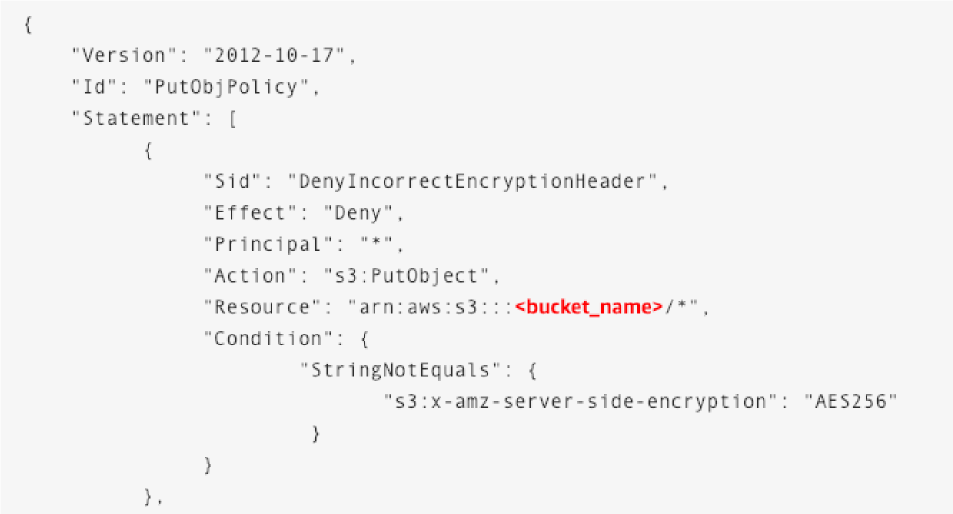
- Resource Based
  - **Bucket Policies -** bucket wide rules from the S3 console - allows cross account
  - Object Access Control List (ACL) – finer grain
  - Bucket Access Control List (ACL) – less common

S3 Bucket Policies

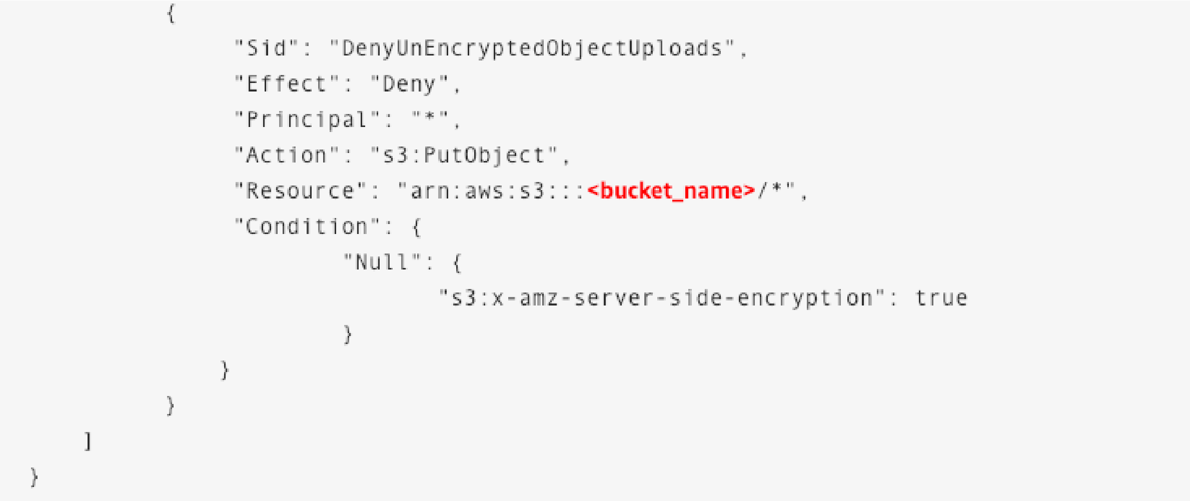
- JSON based policies
  - Resources: buckets and objects
  - Actions: Set of API to Allow or Deny
  - Effect: Allow / Deny
  - Principal: The account or user to apply the policy to
- Use S3 bucket for policy to:
  - Grant public access to the bucket
  - Force objects to be encrypted at upload
  - Grant access to another account (Cross Account)

S3 Default Encryption vs Bucket Policies

- The old way to enable default encryption was to use a bucket policy and refuse any HTTP command without the proper headers:



```
{
    "Version": "2012-10-17",
    "Id": "PutObjPolicy",
    "Statement": [
          {
               "Sid": "DenyIncorrectEncryptionHeader",
               "Effect": "Deny",
               "Principal": "*",
               "Action": "s3:PutObject",
               "Resource": "arn:aws:s3:::<bucket_name>/*",
               "Condition": {
                       "StringNotEquals": {
                              "s3:x-amz-server-side-encryption": "AES256"
                       }
               }
          },
```



```
              {
                   "Sid": "DenyUnEncryptedObjectUploads",
                   "Effect": "Deny",
                   "Principal": "*",
                   "Action": "s3:PutObject",
                   "Resource": "arn:aws:s3:::<bucket_name>/*",
                   "Condition": {
                           "Null": {
                                  "s3:x-amz-server-side-encryption": true
                           }
                   }
              }
        ]
}
```

- The new way is to use the “default encryption” option in S3
- Note: Bucket Policies are evaluated before “default encryption”

S3 Security - Other

- Networking **- VPC Endpoint Gateway:**
  - Allow traffic to stay within your VPC (instead of going through public web)
  - Make sure your private services (AWS SageMaker) can access S3
  - **Very important for AWS ML Exam**

- Logging and Audit:
  - S3 access logs can be stored in other S3 bucket
  - API calls can be logged in AWS CloudTrail

- Tagged Based (combined with IAM policies and bucket policies)
  - Example: Add tag Classification=PHI to your objects