# IAM & S3 & MFA

2019年2月7日 星期四　　下午11:00

## 1.Identity & Access Management

## What is IAM?

- **IAM** (**I**dentity & **A**ccess **M**anagement) is the service where AWS user accounts and their access to various AWS services is managed.

- The common use of **IAM** is to manage:
  - ***Users***
  - ***Groups***
  - ***Access Policies***
  - ***Roles***
  - ***User Credentials***
  - ***User password policies***
  - ***Multi-Factor Authentication (MFA)***
  - ***API keys for programmatic (CLI) access***

- By default, any new user created in an AWS account is created ***without*** access to any AWS services (only the ability to log in).

- For a user to access an AWS service, permission must be granted to that user.  Which is managed in/by IAM.


# IAM Users:

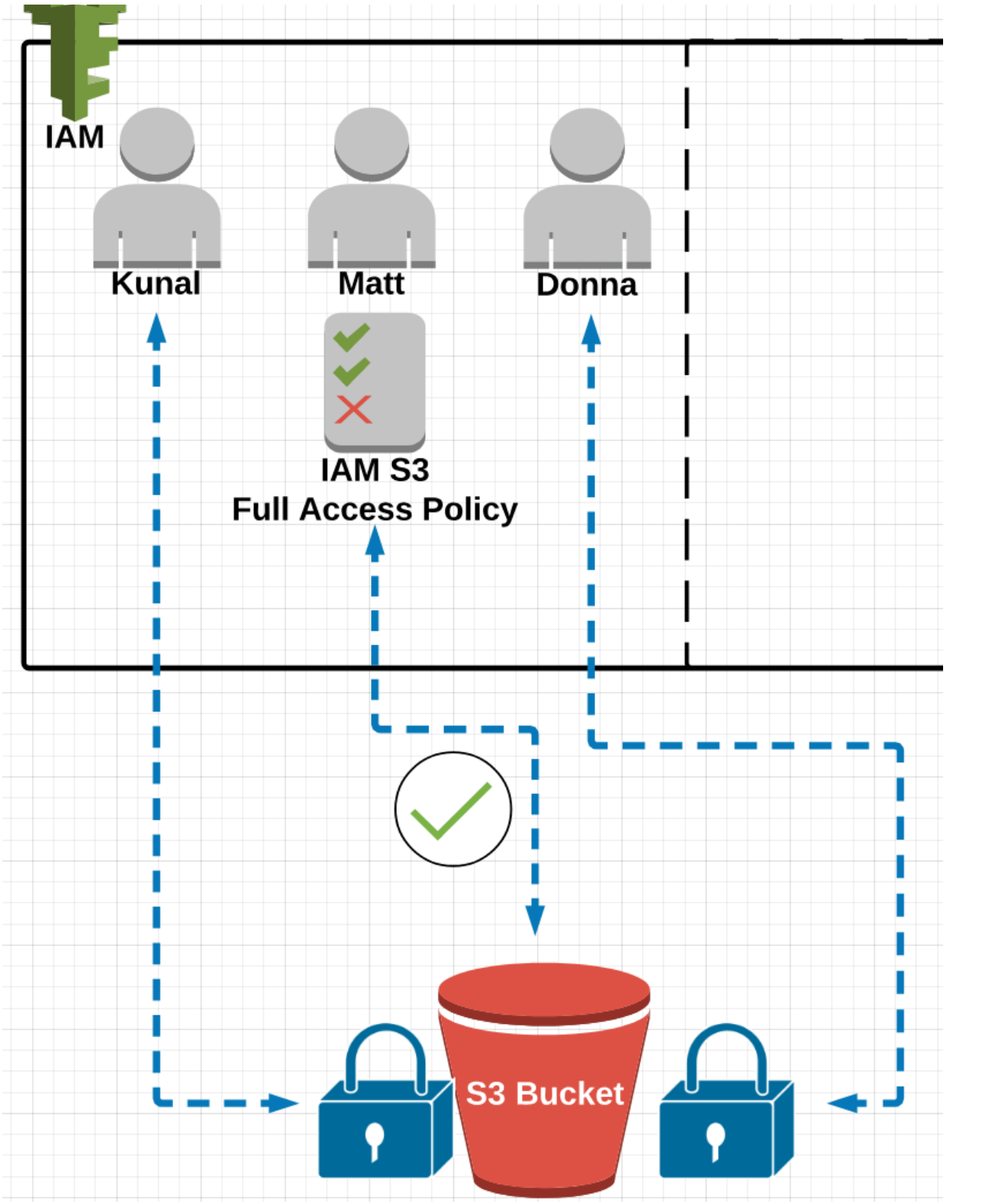
- ***IAM users*** are individuals who have been granted access to an AWS account. For example, if your company gives you access to their AWS account, then you are an IAM user (probably one of many the company has setup).

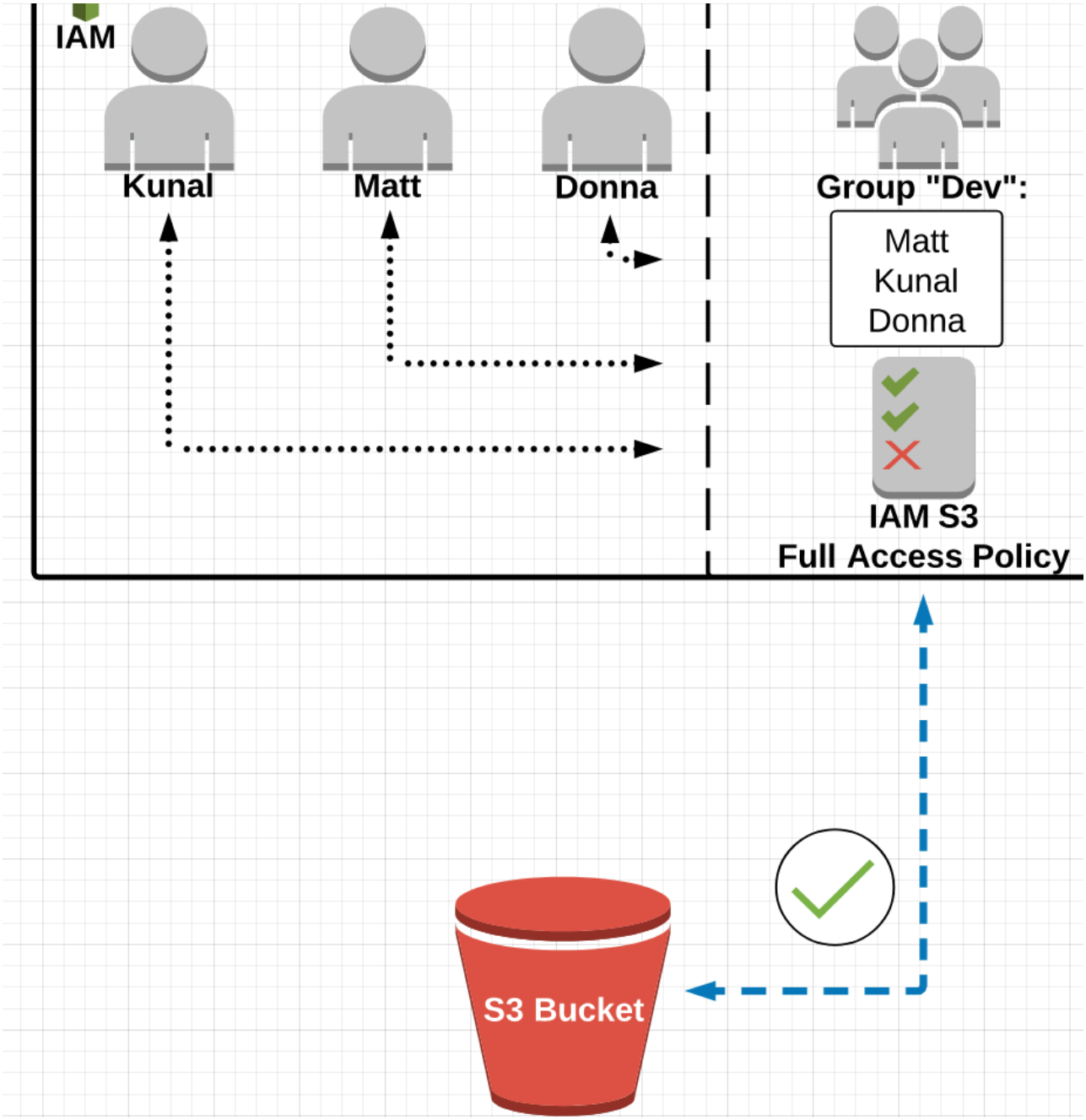
- Each IAM user has three main components:
  - ***A user-name***
  - ***A password***
  - ***Permissions to access various AWS services***

- Without permissions being explicitly granted to an IAM user, that user will not able to access any AWS services.

- Generally, a companies I.T. department will be responsible for "attaching" what are called ***IAM permission policies*** to an IAM user based on what that user needs access to (in order to do their job).

***So what are IAM permission policies and how do they work?***


IAM
Kunal
Matt
Donna
IAM S3
Full Access Policy
S3 Bucket




## Other Uses/Benefits of IAM:

- The common use of **IAM** is to manage:
  - ~~***Users***~~
  - ~~***Groups***~~
  - ~~***Access Policies***~~
  - ***Roles***
  - ***User Credentials***
  - ***User password policies***

- ***Multi-Factor Authentication (MFA)***
- ***API keys for programmatic (CLI) access***

- ***Roles:***
  - How different AWS services, such as EC2 and S3, are granted permission to communicate and share data.

- ***User Credentials:***
  - IAM user's user-name and password for logging into AWS.

- ***User Password Policies:***
  - IAM user password format requirements (i.e. a password must be a minimum of 8 characters and include 1 number).

- ***Multi-Factor Authentication (MFA):***
  - A two-layer form of log-in verification that requires an additional (rotating) code number.

- ***API Keys for Programmatic (CLI) Access:***
  - Special credentials required for accessing AWS resources via the Command Line Interface (CLI).

# 2.Storage Services(S3)

## ***S3 Defined:***

**Simple Definition:**
An online, bulk storage service that you can access from almost any device.

**AWS Definition:**
"Amazon S3 has a simple web services interface that you can use to ***store and retrieve any amount of data, at any time, from anywhere on the web***. It gives any user access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to users."

# AWS S3

## S3 Basics:

***Components and Structure***

**Basics:**

(1) S3 = Simple Storage Service
(2) It is AWS's primary storage service.
(3) You can store any type of file in S3.


**Buckets:**

(1) Root level "Folders" you create in S3 are referred to as ***buckets***.
(2) Any "subfolder" you create in a bucket is referred to as a ***folder***.

**S3 Folder**

**Objects:**

(1) Files stored in a bucket are referred to as ***objects***.

**S3 Object**

**Regions:**

(1) When you create a bucket, you must select a specific region for it to exist in. This means that ***any data you upload to the S3 bucket will be physically***

***located in a data center in that region***.

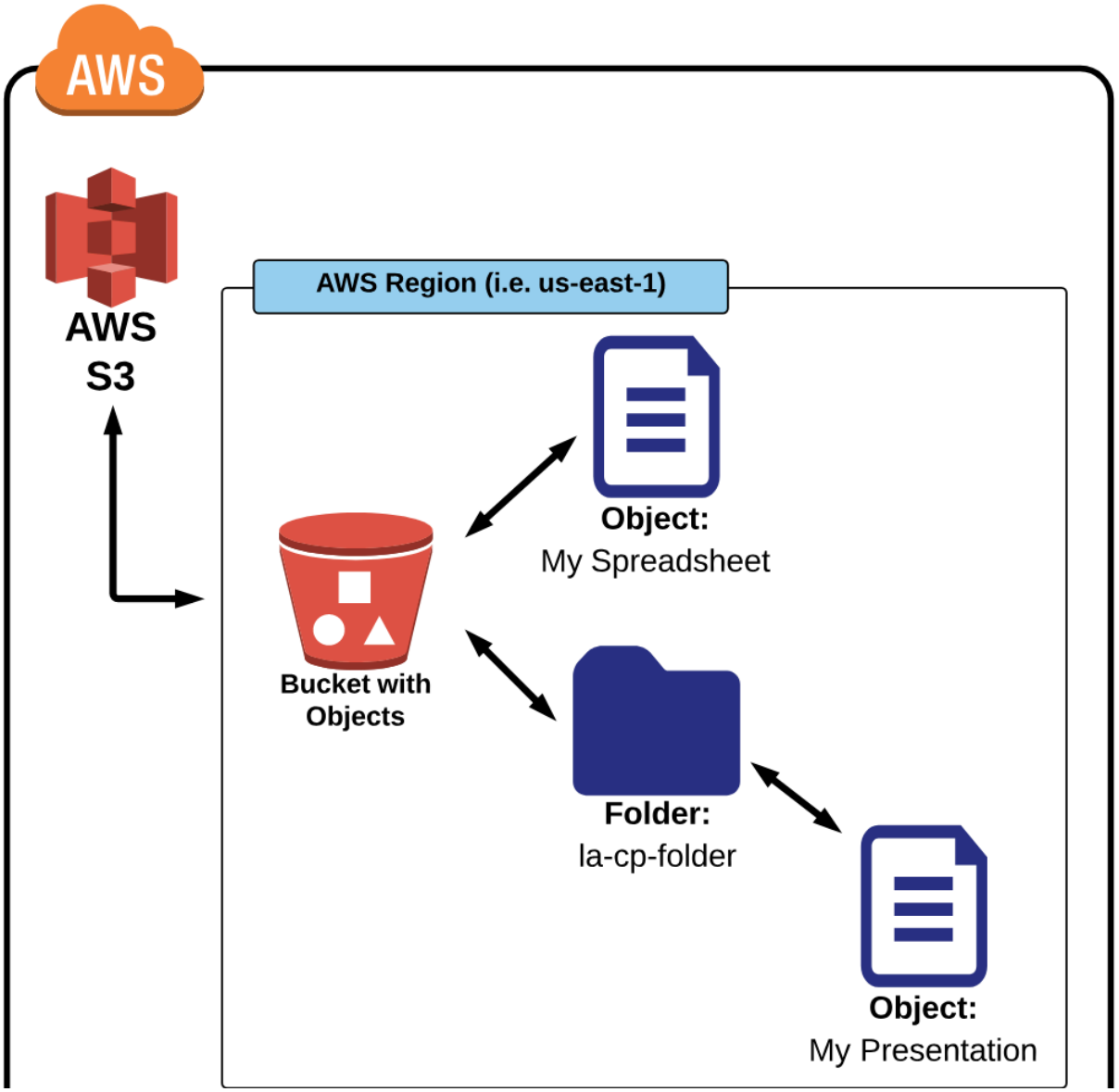
(2) ***Best practice*** is to select the region that is physically ***closest to you*** (to ***reduce transfer latency***).

***OR***

(3) If you are serving files to a ***customer*** based in a certain area of the world, ***create the bucket in a region closest to your customers*** (to reduce latency for your customers).

# S3 Basics:

***Components and Structure (visualized)***



# Buckets and Folders:

## *Creating an S3 Bucket:*

(1) Choose a bucket name:

**Bucket names must follow a set of rules**:
-Bucket names must be unique across ALL of AWS.
-Bucket names must be 3 to 63 characters in length.
-Bucket names can only contain lowercase letters, numbers and hyphens.
-Bucket names must not be formatted as an IP address (e.g., 192.168.5.4).

(2) Select a region

**NOTE: There are more "advanced" rules that allow for some varying formats, which can be found here:**
**http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html**

## *Uploading (Import) an Object to a Bucket:*

(1) Navigate into a bucket
(2) Select "upload"
(3) Select a file to upload
(4) Click "Start Upload"

## *Creating a Folder in a Bucket:*

(1) Navigate into a bucket
(2) Click on "Create Folder"
(3) Give the folder a name

**NOTE: Uploading an object directly into folder is the same process, just navigate into the folder first.**

# S3 Storage Classes:

## What is a storage class?

(1) A ***storage class*** represents the "classification" assigned to each Object in S3.

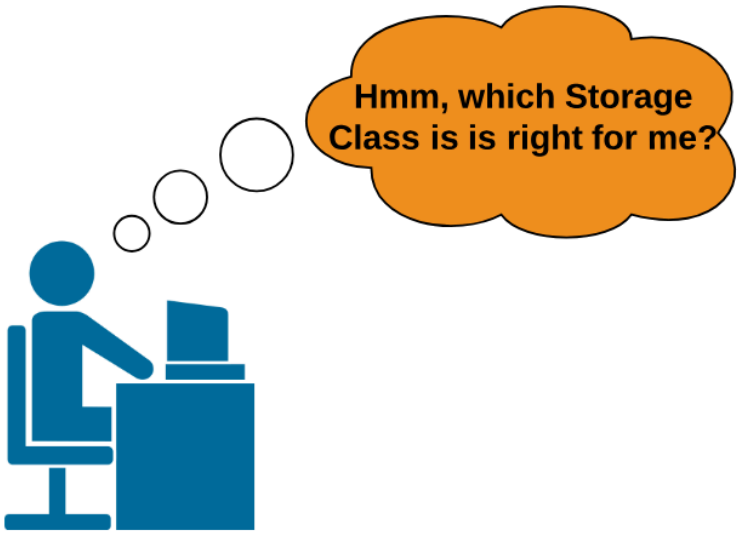
***Available storage classes include:***
-Standard
-Standard-IA (Infrequent Access)
-One Zone-IA (Infrequent Access)
-Intelligent-Tiering
-Glacier

(2) Each ***storage class*** has varying attributes that dictate things like:

-Storage cost
-Object ***availability***
-Object ***durability***
-Frequency of access (to the object)

(3) Each object must be assigned a storage class ("standard" is the default class)

(4) You can change the ***storage class*** of an object at any time (***for the most part***).




# S3 Storage Classes:

***Standard:***
(1) Designed for general, all-purpose storage.
(2) Is the default storage option.
(3) ***99.999999999% object durability*** (“eleven nines”).

(4) ***99.99% object availability.***
(5) Is the most expensive storage class.

***Standard-IA (Infrequent Access):***
(1) Designed for objects that you do not access frequently, but must be immediately available when accessed. (uses multiple Availability Zones)
(2) ***99.999999999% object durability***.
(3) ***99.90% object availability.***
(4) Is less expensive than the standard storage class.

***One Zone-IA (Infrequent Access):***
(1) Designed for objects that you do not access frequently, but must be immediately available when accessed. (only uses 1 Availability Zone)
(2) ***99.99% object durability***.
(3) ***99.50% object availability***.
(4) Is ~20% less expensive than the standard-IA storage class

***Intelligent-Tiering:***
(1) Designed to optimize costs by automatically moving data to the most cost-effective tier based on your usage.
(2) ***99.999999999% object durability*** ("eleven nines").
(3) ***99.90% object availability.***
(4) Pricing depends on the assigned storage class.

***Glacier:***
(1) Designed for long-term archival storage.
(2) May take several hours for objects stored in Glacier to be retrieved.
(3) ***99.999999999% object durability***
(4) Is the cheapest S3 storage class (very low cost)

***Detailed S3 pricing based on storage class:***
https://aws.amazon.com/s3/pricing/

# S3 Storage Classes:

## Setting/changing storage class:

(1) By default, all new objects uploaded to S3 are set to the ***Standard*** storage class

(2) If you want new objects to have a different storage class, then you need to set the proper settings prior to or during the upload process. You can do this by either:

- Selecting another storage class during the upload process ("set properties")

-Selecting another storage class during the upload process ( set properties ).
-Using object ***lifecycle policies***.

(3) For the following storage classes:

-Standard
-Standard-IA (Infrequent Access)
-One Zone-IA (Infrequent Access)
-Intelligent-Tiering

You can manually switch the objects storage class between the four listed above (at any time).

(3) To move an object to the ***Glacier*** storage class:

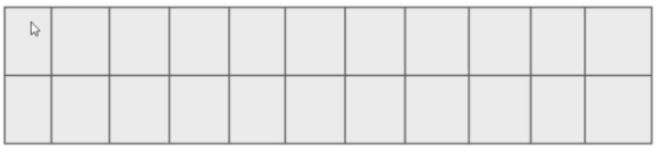
-You can manually change the storage class (new)
-You can use object ***lifecycles.***
-NOTE: Objects cannot be instantly switched from the Glacier storage.

## Block Storage

In block storage, the data is stored in terms of blocks

Data stored in blocks are normally read or written entirely a whole block at the same time

Most of the file systems are based on block devices.



## S3 Object Storage

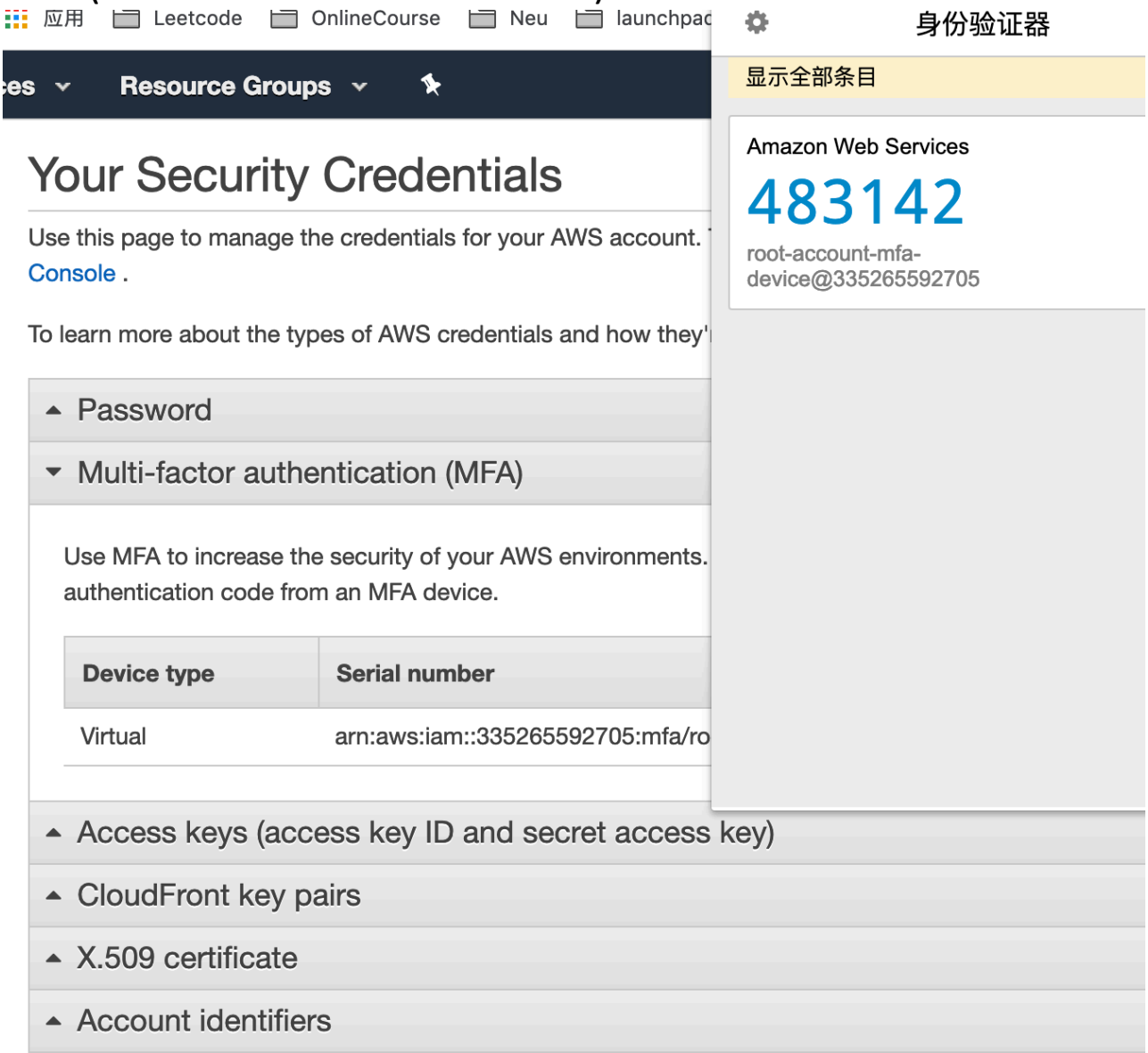
Object storage is a data storage architecture that manages data as objects as opposed to

blocks of storage.

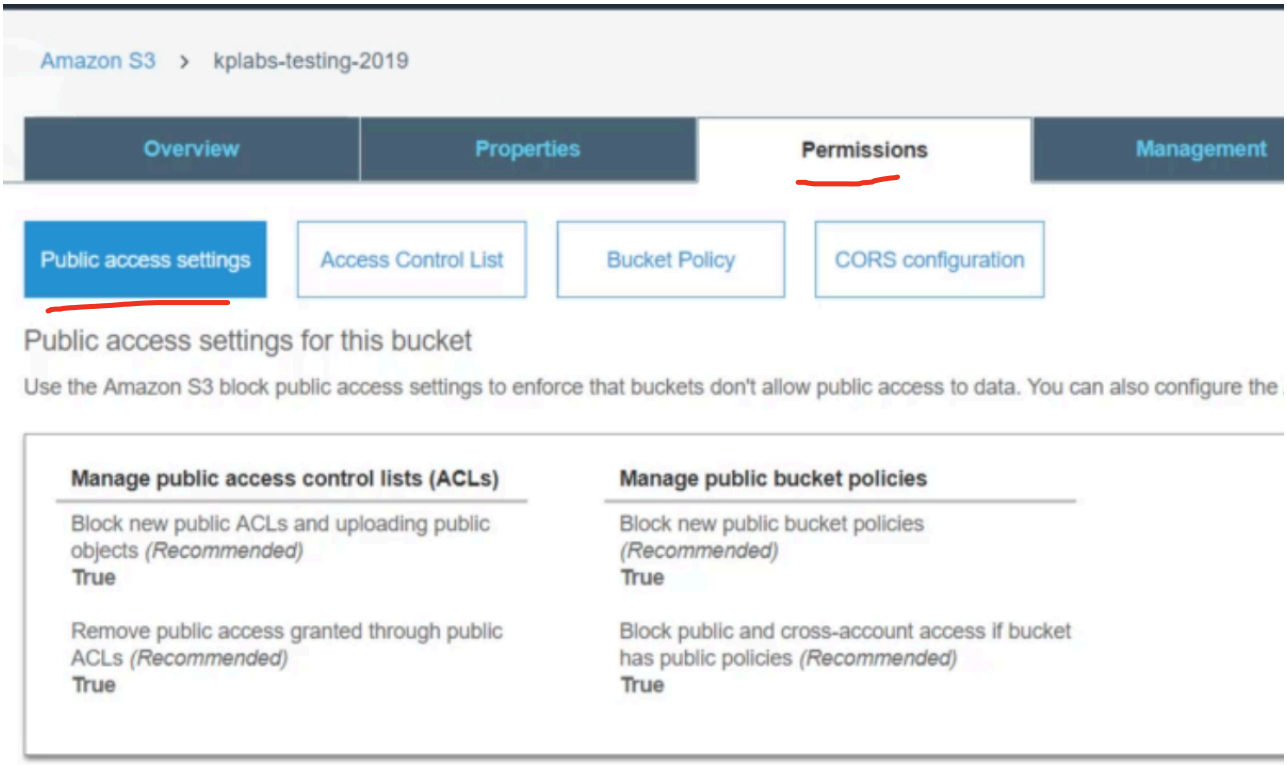
An object is defined as a data (file) along with all it's meta-data which is combined together as an object.

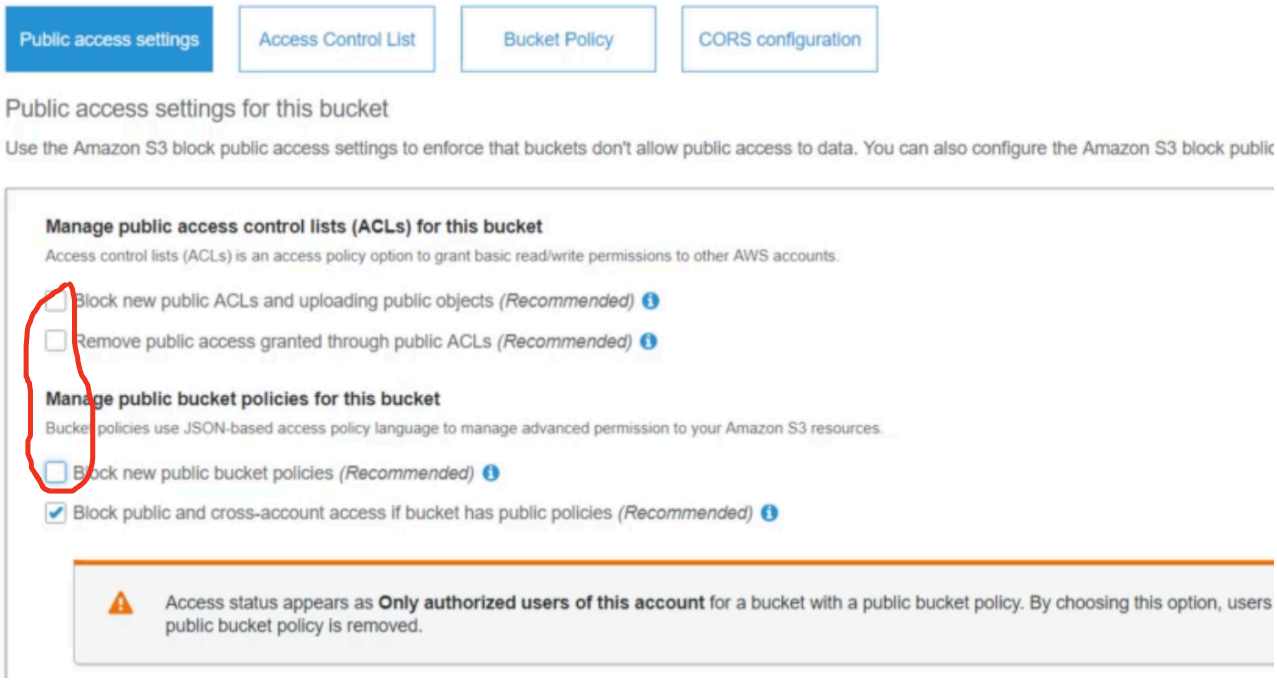
This object is given an ID which is calculated from the content of the object (from the data and metadata ) . Application can then call object with the unique object ID.

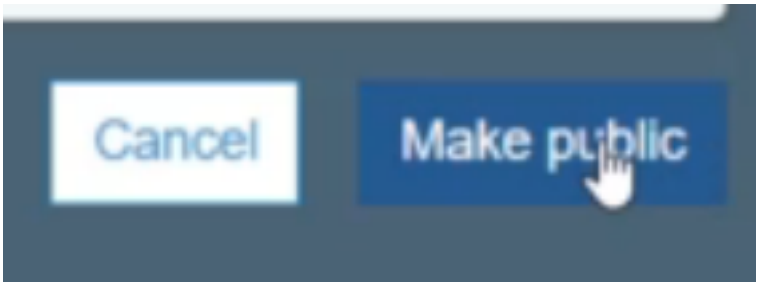
## MFA(Multi-factor authentication)




## Bucket Public:


Amazon S3 › kplabs-testing-2019
Overview
Properties
Permissions
Management
Public access settings
Access Control List
Bucket Policy
CORS configuration
Public access settings for this bucket
Use the Amazon S3 block public access settings to enforce that buckets don't allow public access to data. You can also configure the
Manage public access control lists (ACLs)
Block new public ACLs and uploading public objects (Recommended)
True
Remove public access granted through public ACLs (Recommended)
True
Manage public bucket policies
Block new public bucket policies (Recommended)
True
Block public and cross-account access if bucket has public policies (Recommended)
True


Public access settings
Access Control List
Bucket Policy
CORS configuration
Public access settings for this bucket
Use the Amazon S3 block public access settings to enforce that buckets don't allow public access to data. You can also configure the Amazon S3 block public
Manage public access control lists (ACLs) for this bucket
Access control lists (ACLs) is an access policy option to grant basic read/write permissions to other AWS accounts.
Block new public ACLs and uploading public objects (Recommended)
Remove public access granted through public ACLs (Recommended)
Manage public bucket policies for this bucket
Bucket policies use JSON-based access policy language to manage advanced permission to your Amazon S3 resources.
Block new public bucket policies (Recommended)
Block public and cross-account access if bucket has public policies (Recommended)
Access status appears as Only authorized users of this account for a bucket with a public bucket policy. By choosing this option, users
public bucket policy is removed.


Cancel
Make public

S3 offers various kinds of storage classes for different use cases :-

General Purpose → Standard S3

Infrequent Access → Standard S3 ( Infrequent Access ) [ Standard IA]

RRS → Reduced Redundancy Storage

Archive → Glacier

Standard 99.99..9%
99.99%

IA 99.99..99%
99.90%

RRS 99.99%
99.99%

| | Standard | Standard - Infrequent Access | Reduced Redundancy Storage |
|---|---|---|---|
| Durability | 99.999999999% | 99.999999999% | 99.99% |
| Availability | 99.99% | 99.9% | 99.99% |
| Concurrent facility fault tolerance | 2 | 2 | 1 |

Glacier
99.99...99%

Can't directly to glacier
Upload to standard, then glacier

S3 Life Cycle Use Case:

## Let´s Explore

We now understand that there are various S3 Storage class offered by S3.

We need to make that storage is durable + affordable for long term storage.

- We can store 3 months of logs in Amazon S3 Standard.
- Move the logs older than 3 months to S3 Standard IA
- Move the logs older than 1 year to Glacier