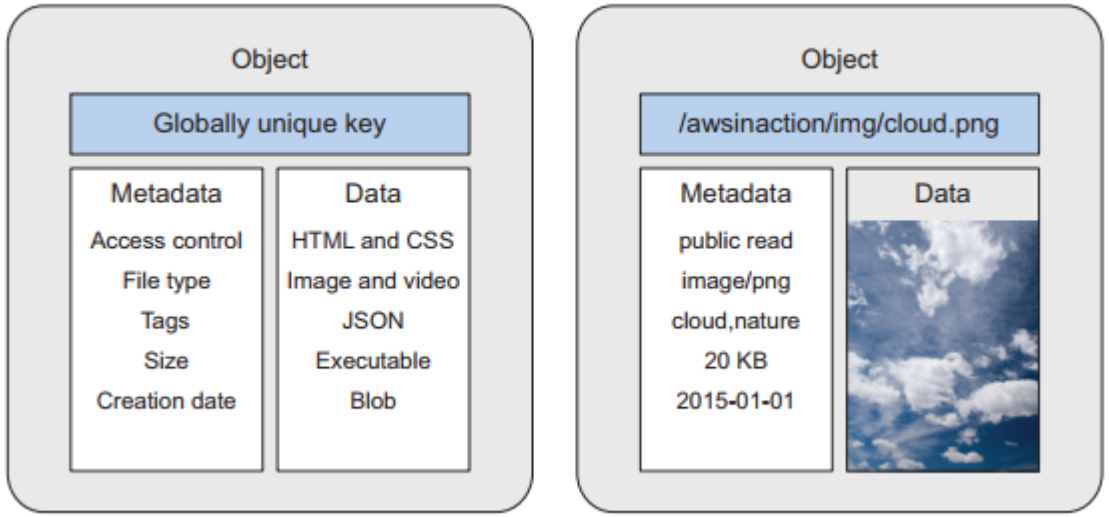
# Lesson 7 S3

Michael Yang

# What is Object Store?

- Back in the old days, data was managed in a hierarchy consisting of folders and files. The file was the representation of the data. In an *object store*, data is stored as objects. Each object consists of a globally unique identifier, some metadata, and the data itself, as figure illustrates. An object's *globally unique identifier* (GUID) is also known as its *key*; you can address the object from different devices and machines in a distributed system using the GUID.



# Amazon S3

- Amazon S3 is a distributed data store, and one of the oldest services provided by AWS. *Amazon S3* is an acronym for *Amazon Simple Storage Service*. It's a typical web service that lets you store and retrieve data organized as objects via an API reachable over HTTPS.

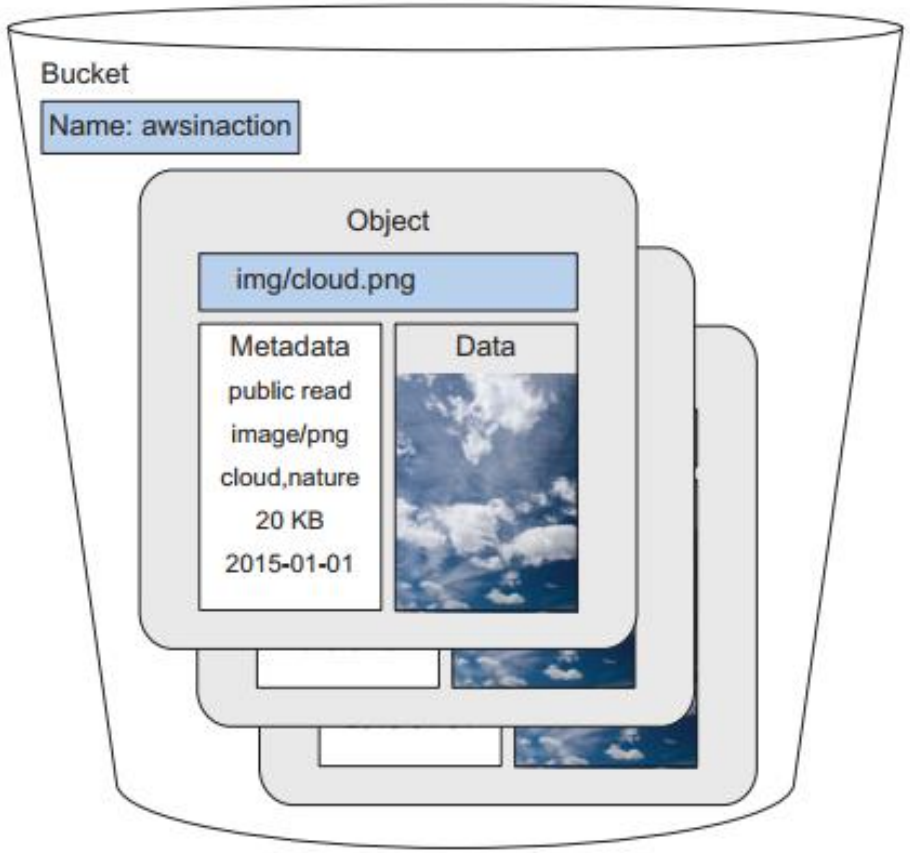
# S3 Use Cases

- *Storing and delivering static website content*
- *Backing up data*
- *Storing structured data for analytics, also called a data lake*
- *Storing and delivering user-generated content*

# Amazon S3


**Uploading and downloading an object to S3 via HTTPS**

# Bucket




**S3 uses buckets with globally unique names to group objects.**

# Backup with AWS CLI

- Use the AWS CLI to upload data to and download data from S3. This approach isn't limited to offsite backups; you can use it in many other scenarios as well, such as the following
  - Sharing files with your coworkers or partners, especially when working from different locations
  - Storing and retrieving artifacts needed to provision your virtual machines (such as application binaries, libraries, or configuration files)
  - Outsourcing storage capacity to lighten the burden on local storage systems—in particular, for data that is accessed infrequently

# AWS CLI for S3

- `$ aws s3 mb s3://$yourbucketname`
- `$ aws s3 sync $path s3://$yourbucketname/backup`
- `$ aws s3 cp --recursive s3://$yourbucketname/backup $path`
- `$ aws s3 rb --force s3://awsinaction-$yourname`

# Versioning

- **$ aws s3api put-bucket-versioning --bucket $yourbucketname --versioning-configuration Status=Enabled**

- **$ aws s3api list-object-versions --bucket  $yourbucketname**

# BucketNotEmpty Error

- If you turn on versioning for your bucket, removing the bucket will cause a `BucketNotEmpty` error. Use the Management Console to delete the bucket in this case as follows:
  1 Open the Management Console with your browser.
  2 Go to the S3 service using the main navigation menu.
  3 Select the bucket you want to delete.
  4 Click the Empty button, and confirm permanently deleting all the objects
  5 Wait until objects and versions have been deleted, and click the Exit button
  6 Select the bucket you want to delete.
  7 Click the Delete button, and confirm deleting the bucket

# Optimize your cost

| | S3 Standard | S3 Glacier Instant Retrieval | S3 Glacier Flexible Retrieval | S3 Glacier Deep Archive |
|---|---|---|---|---|
| Storage costs for 1 GB per month in US East (N. Virginia) | $0.023 | $0.004 | $0.0036 | $0.00099 |
| Costs for 1,000 write requests | $0.005 | $0.02 | $0.03 | $0.05 |
| Costs for retrieving data | Low | High | High | Very High |
| Accessibility | Milliseconds | Milliseconds | 1–5 minutes/<br>3–5 hours/<br>5–12 hours | 12 hours/<br>48 hours |
| Durability objective | 99.999999999% | 99.999999999% | 99.999999999% | 99.999999999% |
| Availability objective | 99.99% | 99.9% | 99.99% | 99.99% |

# S3 Glacier

- `$ aws s3 mb s3://$yourbucketname`
- `$ aws s3 cp --storage-class GLACIER $path s3://$yourbucketname`
- `$ aws s3 cp s3://$yourbucketname/$objectkey $path`
- `$ aws s3api restore-object --bucket $yourbucketname --key $objectkey --restore-request Days=1,,GlacierJobParameters={"Tier"="Expedited"}`

# Download Archives

```
$ aws s3api head-object --bucket $yourbucketname  --key $objectkey
```

```
{
    "AcceptRanges": "bytes",
    "Expiration": "expiry-date=\"Wed, 12 Jul 2023 ...\", rule-id=\"...\"",
    "Restore": "ongoing-request=\"true\"",
    "LastModified": "2022-07-11T09:26:12+00:00",
    "ContentLength": 112,
    "ETag": "\"c25fa1df1968993d8e647c9dcd352d39\"",
    "ContentType": "binary/octet-stream",
    "Metadata": {},
    "StorageClass": "GLACIER"
}
```

**Restoration of the object is still ongoing.**

# Cleanup

- `$ aws s3 rb --force s3://$yourbucketname`

# Storing with SDK

- Listing buckets and their objects
- Creating, removing, updating, and deleting (CRUD) objects and buckets
- Managing access to objects

# S3 Integration examples

- *Allow a user to upload a profile picture.* Store the image in S3, and make it publicly accessible. Integrate the image into your website via HTTPS.
- *Generate monthly reports (such as PDFs), and make them accessible to users.* Create the documents and upload them to S3. If users want to download documents, fetch them from S3.
- *Share data between applications.* You can access documents from different applications. For example, application A can write an object with the latest information about sales, and application B can download the document and analyze the data.