DATA STORAGE, S3

project with the latest technology

Lee Hae Joon

Contents

- **S3**
- HDFS

Contents

- **S3**
  a. Concept
  b. Console
  c. Bucket
  d. Object

S3 Concept

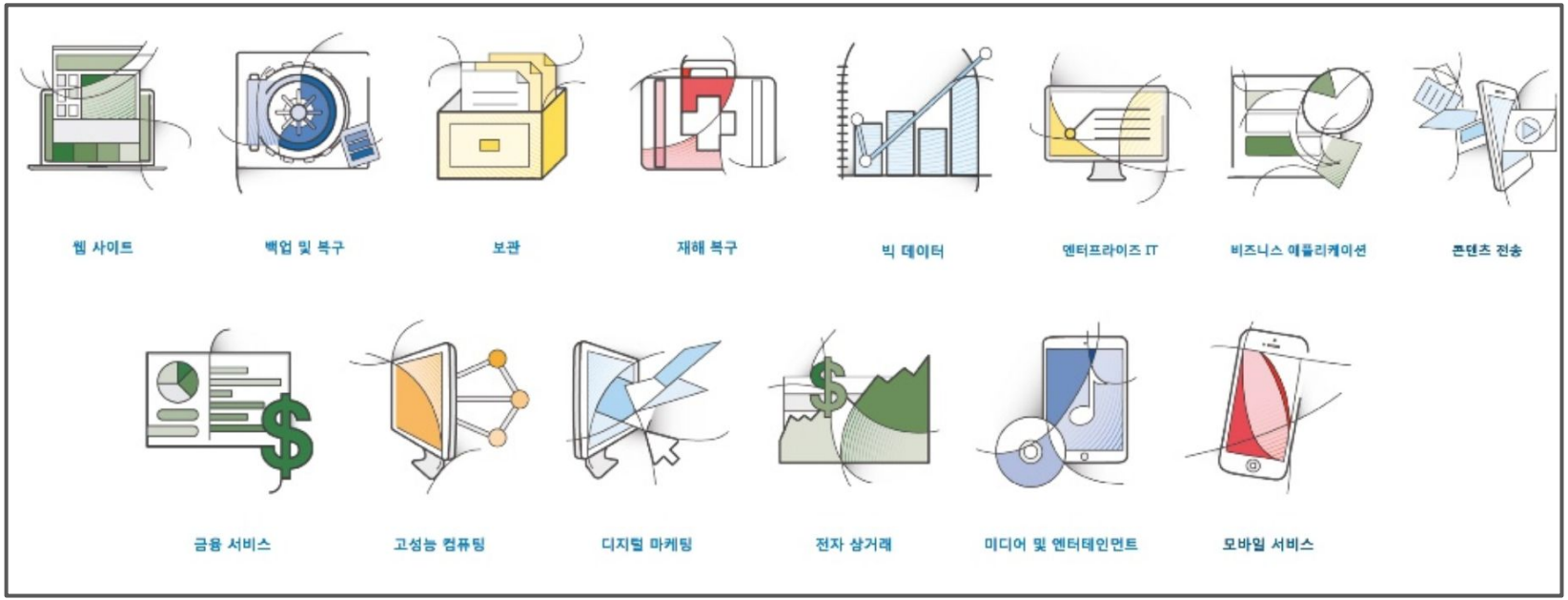
- **S3 - Simple Stroage Service**
  - low price
  - 99.999999999% durability
  - Scalability
  - Flexible Management
  - Reliability
  - Security (Acess)

https://aws.amazon.com/s3/

S3 Concept

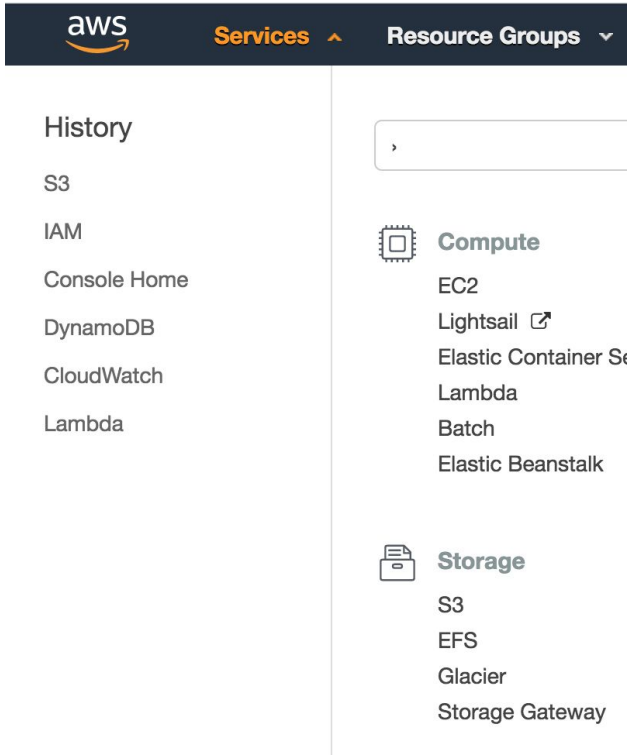
- **Usecase**
  a. Backup & Recovery
  b. Data Archiving
  c. Big Data Analytics
  d. Disaster Recovery

S3


웹 사이트
백업 및 복구
보관
재해 복구
빅 데이터
엔터프라이즈 IT
비즈니스 애플리케이션
콘텐츠 전송
금융 서비스
고성능 컴퓨팅
디지털 마케팅
전자 상거래
미디어 및 엔터테인먼트
모바일 서비스

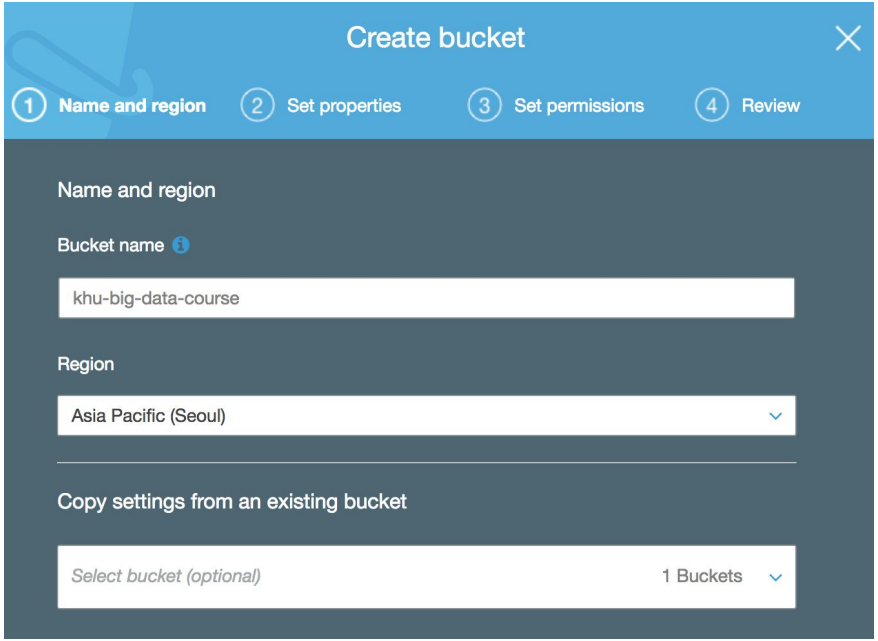
S3 Console

- **Creat a bucket**



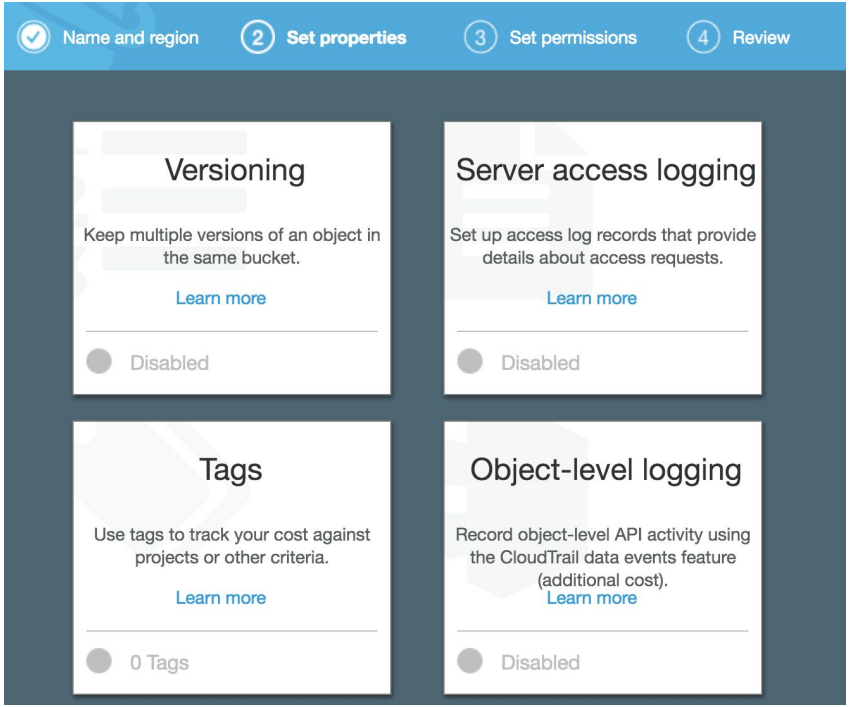
S3 Console

- **Create a bucket**
  - **bucket unique name**
  - **region, e.g, seoul**



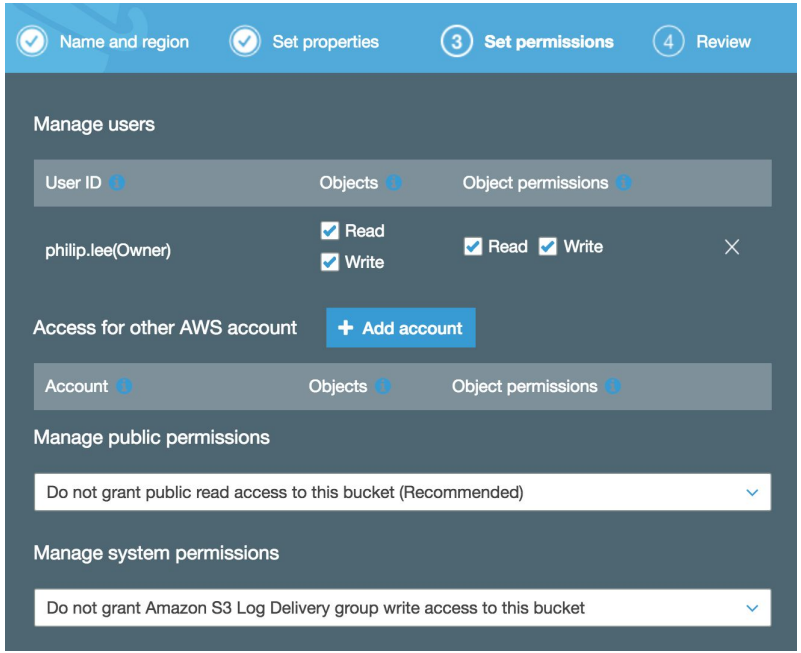
S3 Console

- **Set Properties**



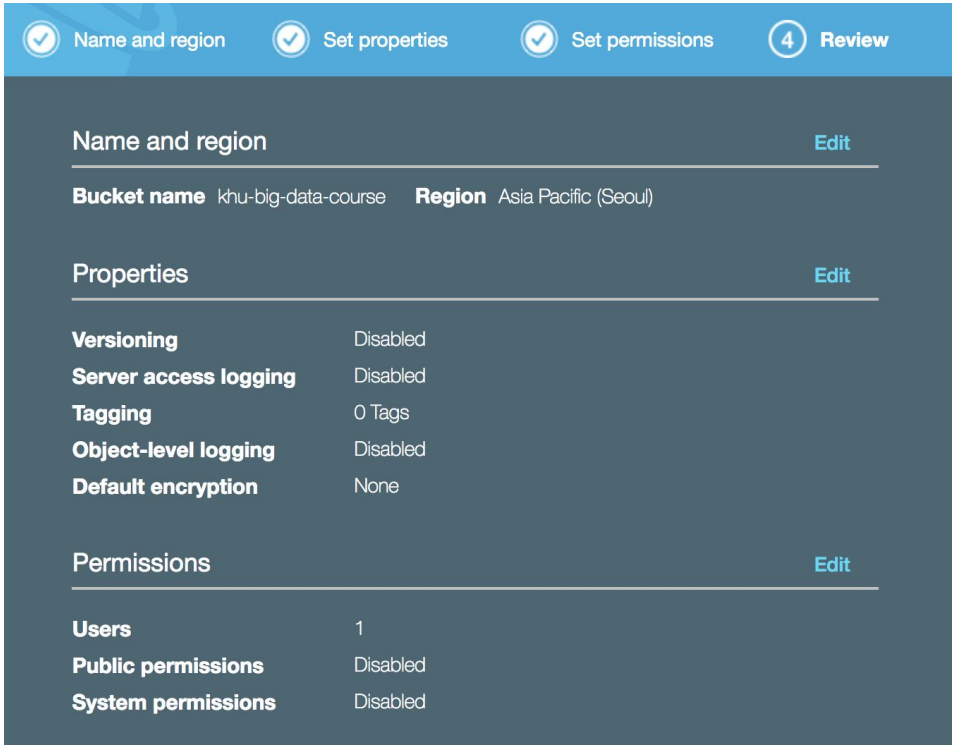
S3 Console

- **Set Permission**
  - **follow recommended opt**

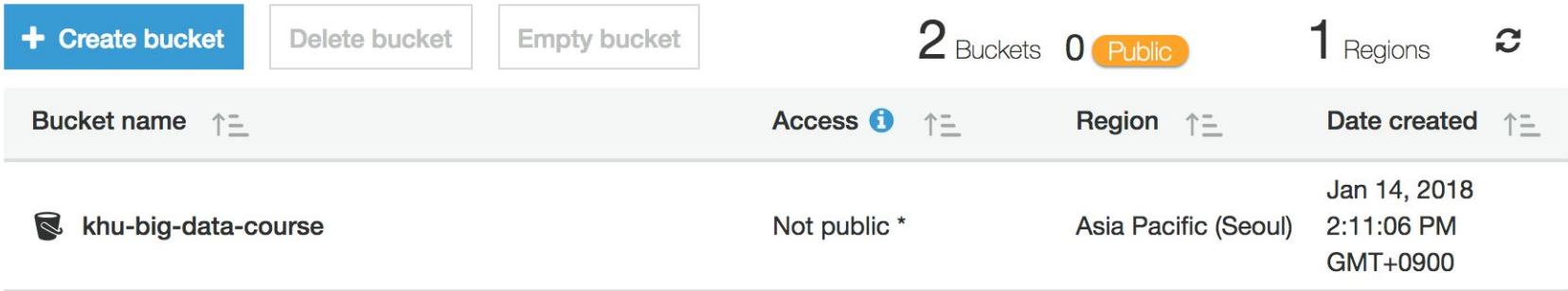


S3 Console

- **Review**

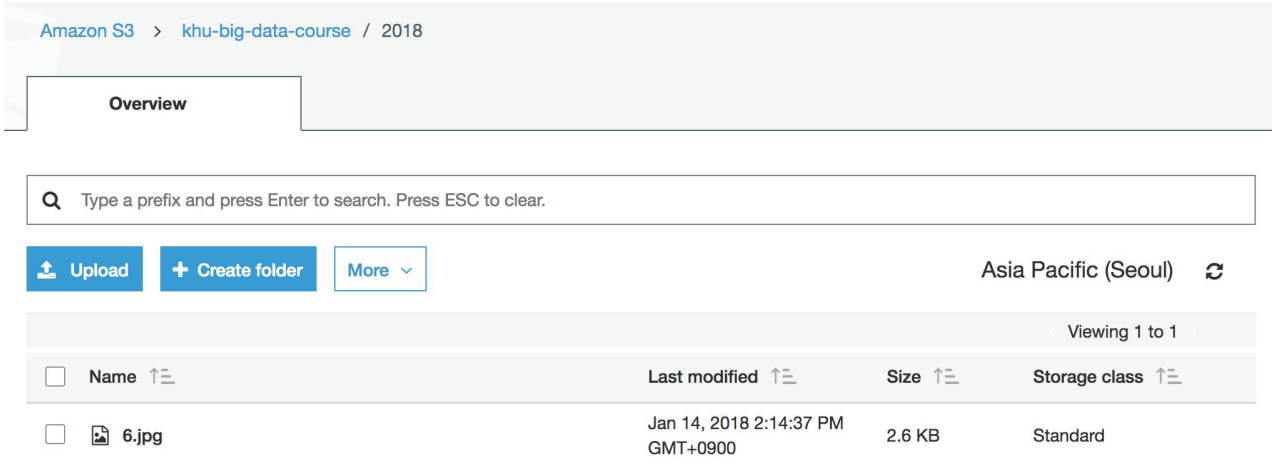


S3 Console

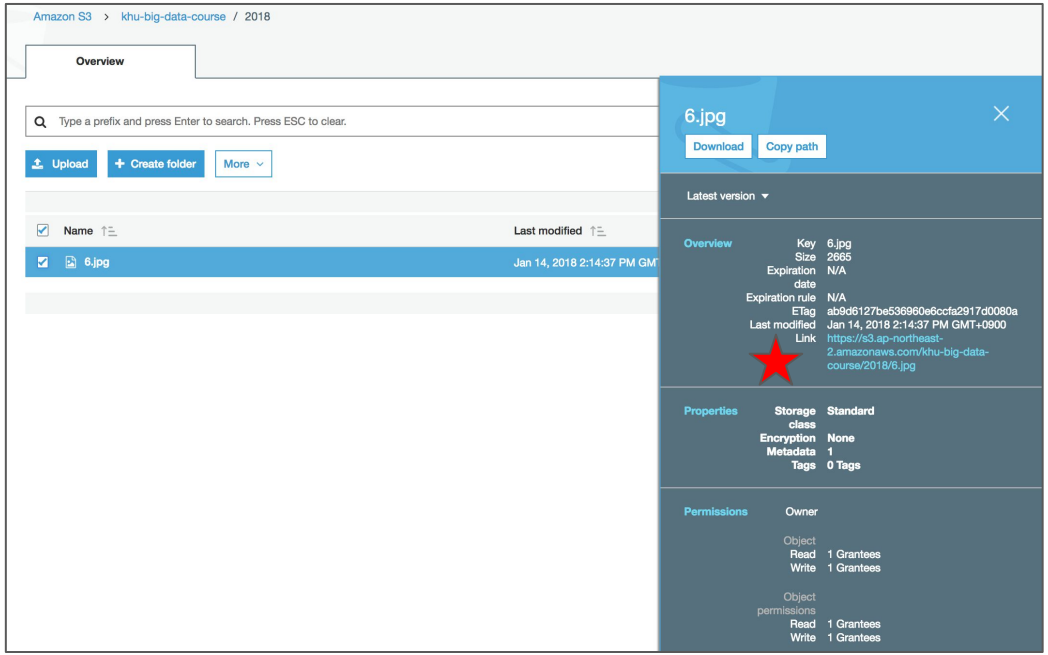

+ Create bucket
Delete bucket
Empty bucket
2 Buckets
0 Public
1 Regions
Bucket name
Access
Region
Date created
khu-big-data-course
Not public *
Asia Pacific (Seoul)
Jan 14, 2018
2:11:06 PM
GMT+0900

S3 Console

- Create a folder : 2018
- Upload an arbitary file in the folder



S3 Console


Amazon S3 > khu-big-data-course / 2018
Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
More
Name
Last modified
6.jpg
Jan 14, 2018 2:14:37 PM GM
6.jpg
Download
Copy path
Latest version
Overview
Key 6.jpg
Size 2665
Expiration date N/A
Expiration rule N/A
ETag ab9d6127be536960e6ccfa2917d0080a
Last modified Jan 14, 2018 2:14:37 PM GMT+0900
Link https://s3.ap-northeast-2.amazonaws.com/khu-big-data-course/2018/6.jpg
Properties
Storage class Standard
Encryption None
Metadata 1
Tags 0 Tags
Permissions
Owner
Object
Read 1 Grantees
Write 1 Grantees
Object permissions
Read 1 Grantees
Write 1 Grantees

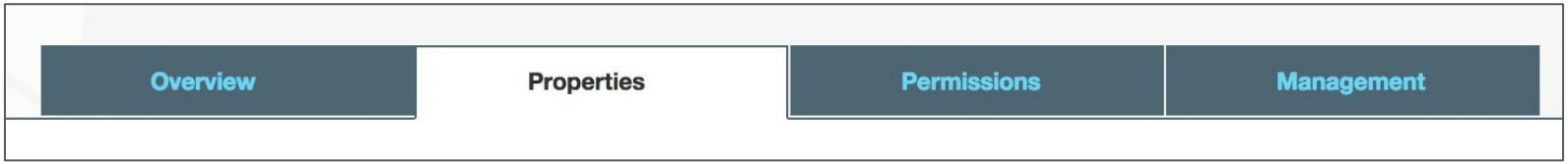
S3 Console

This XML file does not appear to have any style information associated with it. The document tree is shown below.

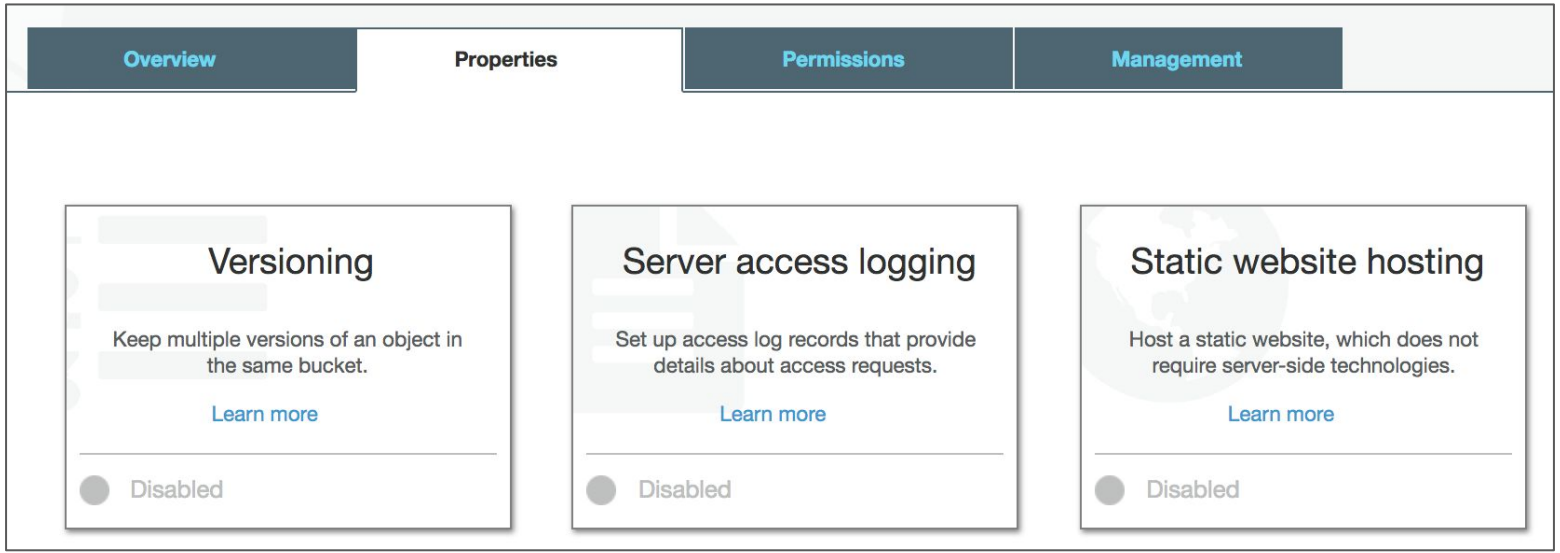
```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>Access Denied</Message>
   <RequestId>35F4769232978AE3</RequestId>
  ▼<HostId>
     s3hOAyob5j7ooEo/9rFZ7DLb0vDvzA/y9U0a9lLLYXudNIwZ+KGPSxbDvSl5Z4BYJM7XvvFYNCc=
   </HostId>
 </Error>
```

Why do you think this happens?

- **Bucket Handling**

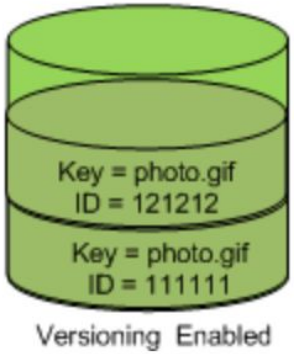


S3 Bucket


Overview
Properties
Permissions
Management
Versioning
Keep multiple versions of an object in the same bucket.
Learn more
Disabled
Server access logging
Set up access log records that provide details about access requests.
Learn more
Disabled
Static website hosting
Host a static website, which does not require server-side technologies.
Learn more
Disabled

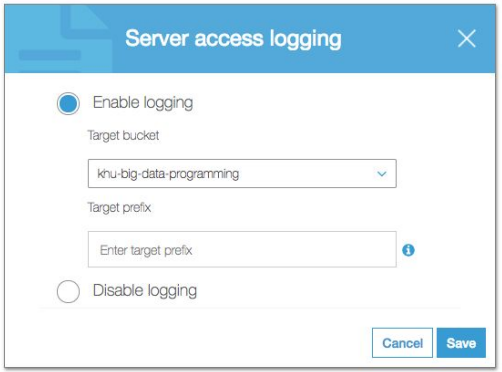
S3 Bucket

- **Versioning**
  - Versioning is a means of keeping multiple variants of an object in the same bucket.
    In one bucket, for example, you can have two objects with the same key, but different version IDs, such as photo.gif (version 111111) and photo.gif (version 121212).



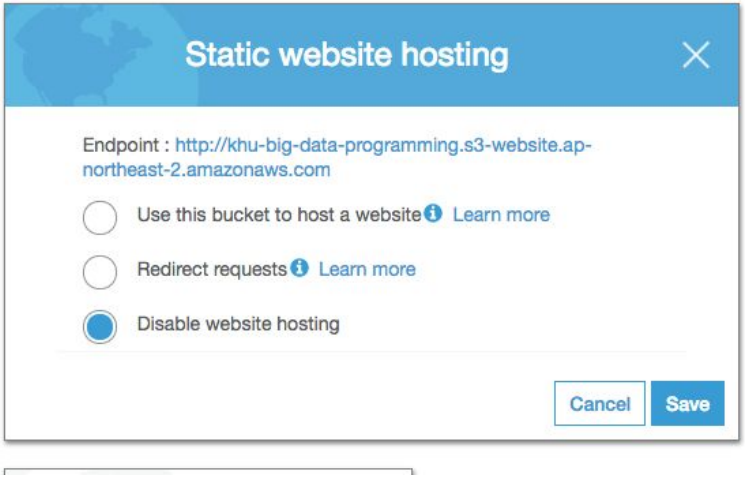
S3 Bucket

- **Server Access Logging**
  - To track requests for access to your bucket, you can enable access logging. Each access log record provides details about a single access request, such as the requester, bucket name, request time, request action, response status, and error code,

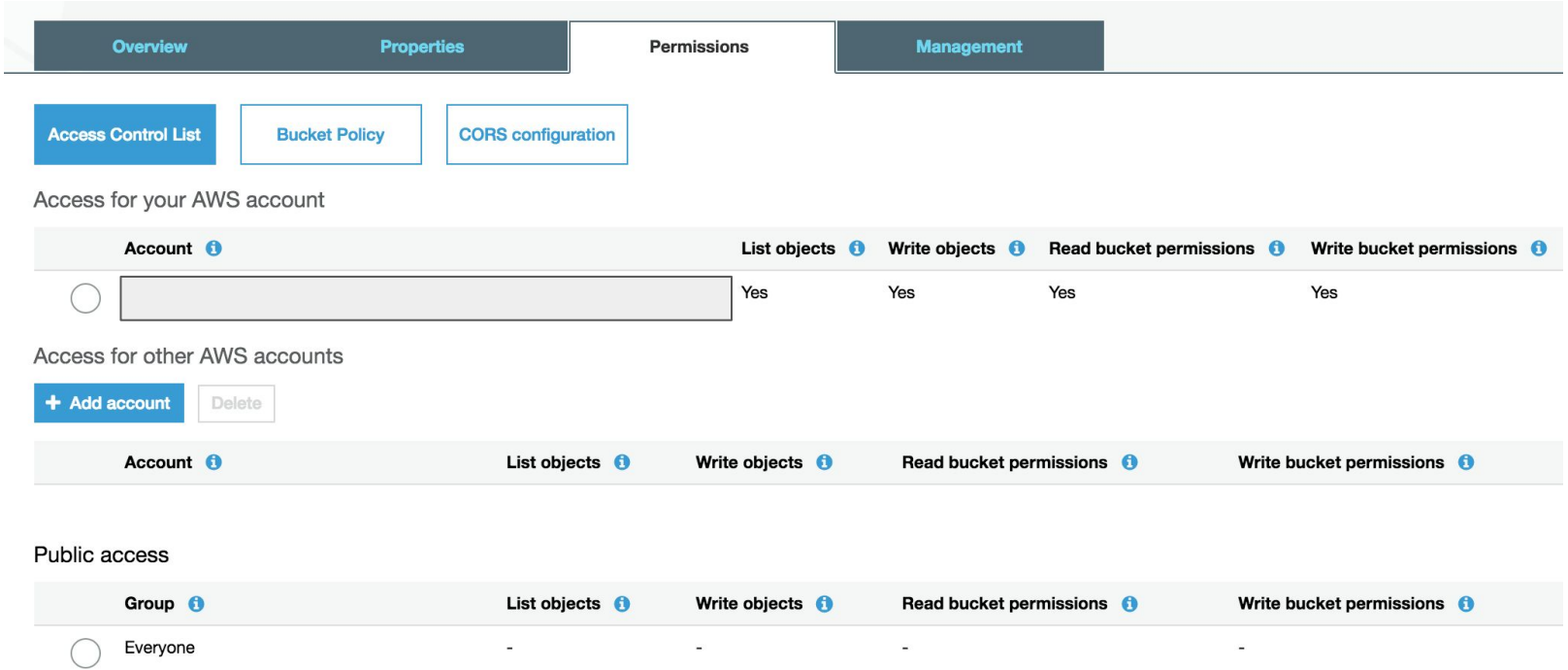


S3 Bucket

- **Static Website Hosting**
  - http://www.smalldatajournalism.com/projects/one-offs/using-amazon-s3/



S3 Bucket

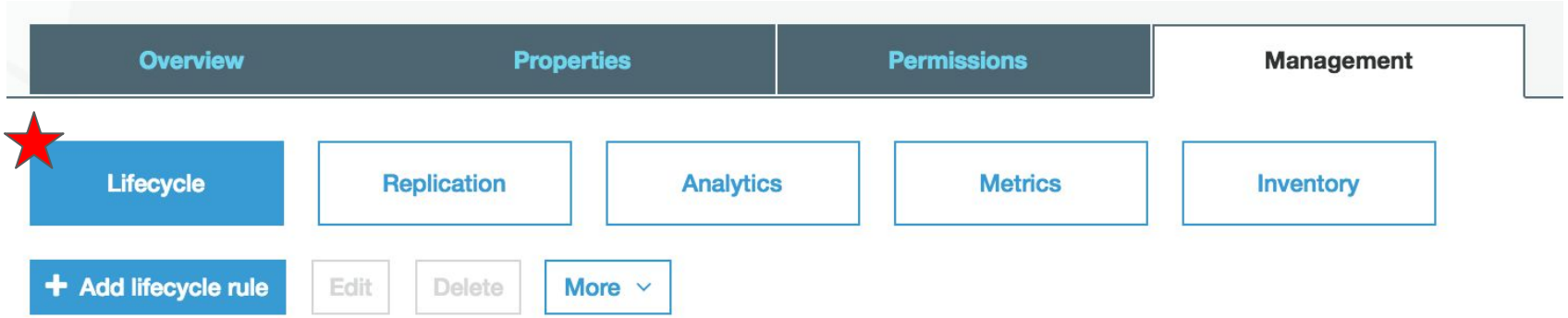

Overview
Properties
Permissions
Management
Access Control List
Bucket Policy
CORS configuration
Access for your AWS account
Account
List objects
Write objects
Read bucket permissions
Write bucket permissions
Yes
Yes
Yes
Yes
Access for other AWS accounts
+ Add account
Delete
Account
List objects
Write objects
Read bucket permissions
Write bucket permissions
Public access
Group
List objects
Write objects
Read bucket permissions
Write bucket permissions
Everyone
-
-
-
-

S3 Bucket

- **Policy**

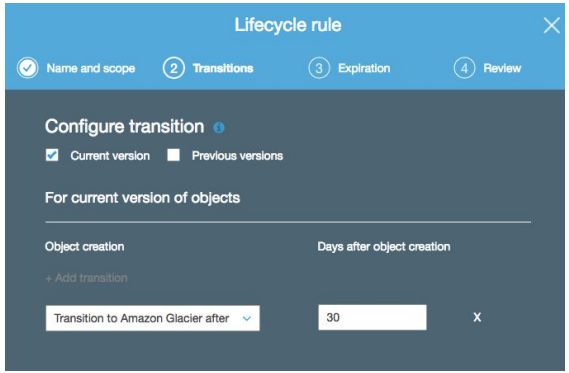
http://awspolicygen.s3.amazonaws.com/policygen.html

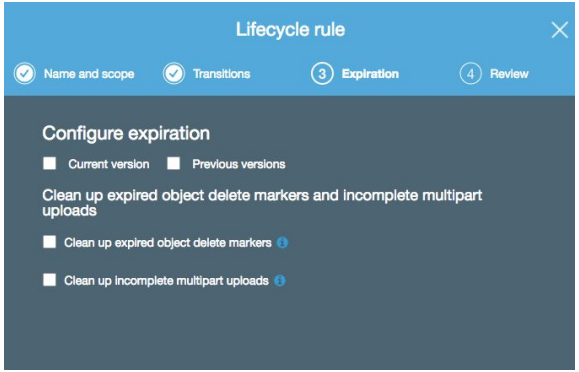
S3 Bucket


Overview
Properties
Permissions
Management
Lifecycle
Replication
Analytics
Metrics
Inventory
Add lifecycle rule
Edit
Delete
More

S3 Bucket

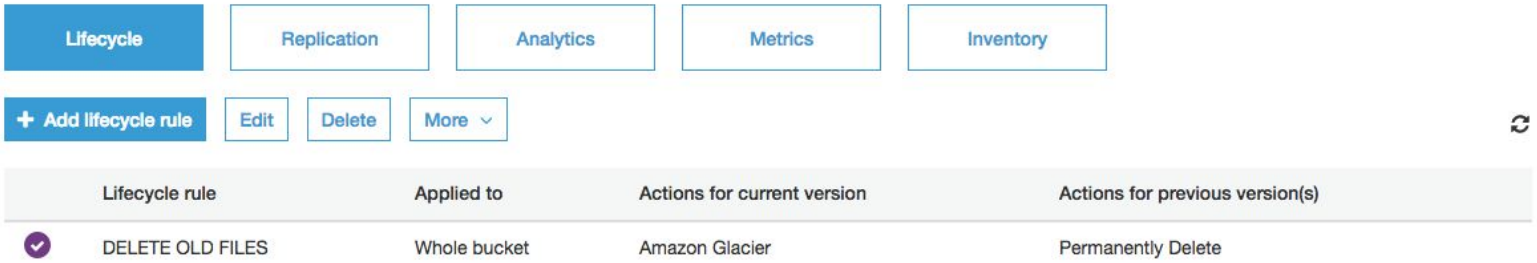
- **Lifecycle, let's explore it**





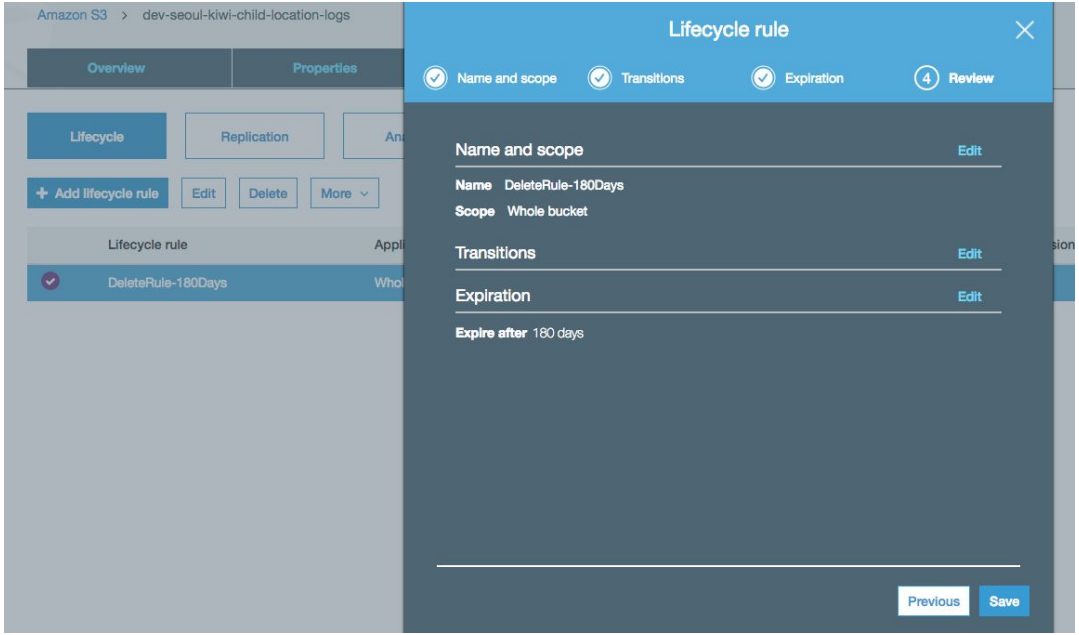
S3 Bucket

- **Lifecycle**

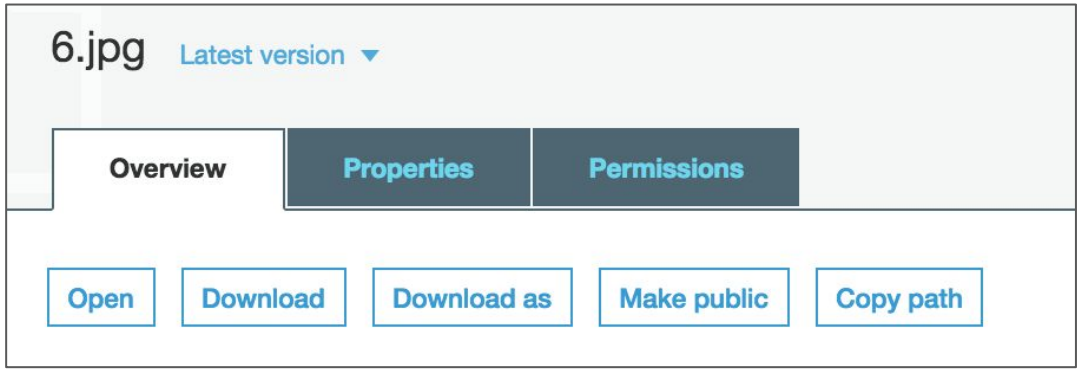


S3 Bucket

- **Lifecycle review**

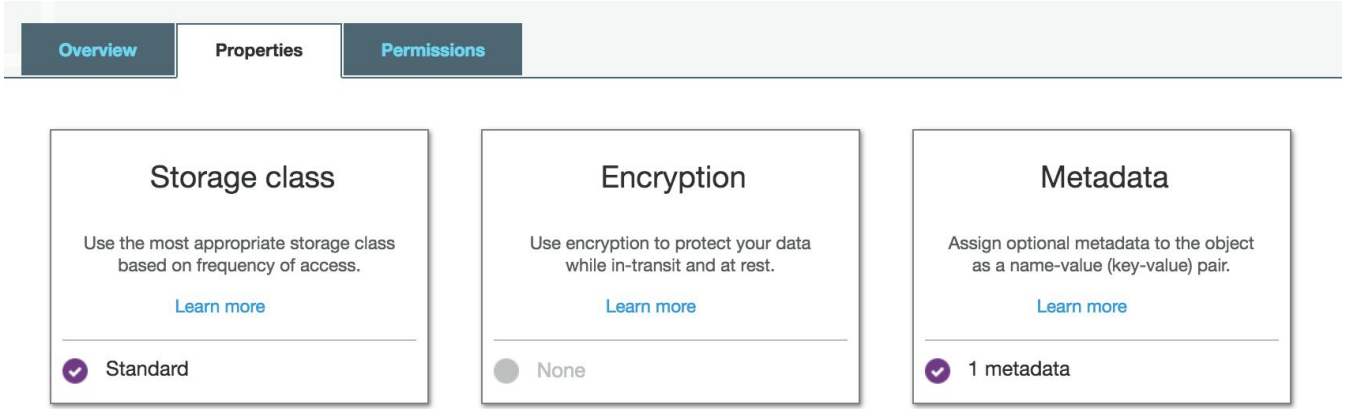


- **Object Handling**



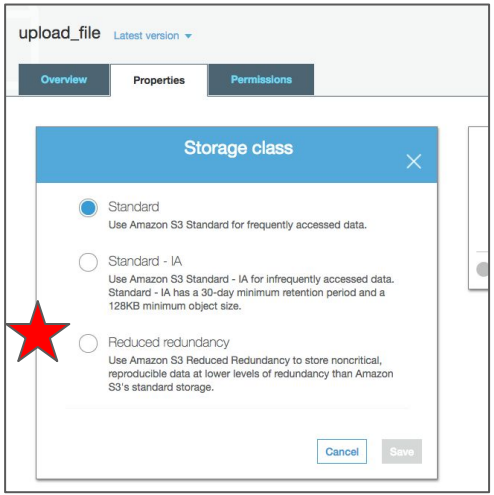
S3 Object

- **Properties**



S3 Object

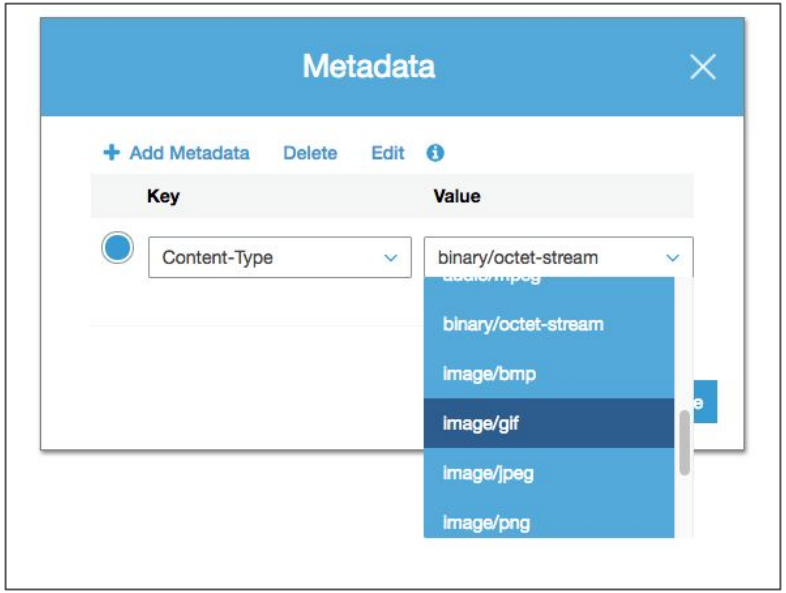
- **Storage Class, Property**



https://docs.aws.amazon.com/AmazonS3/latest/dev/storage-class-intro.html

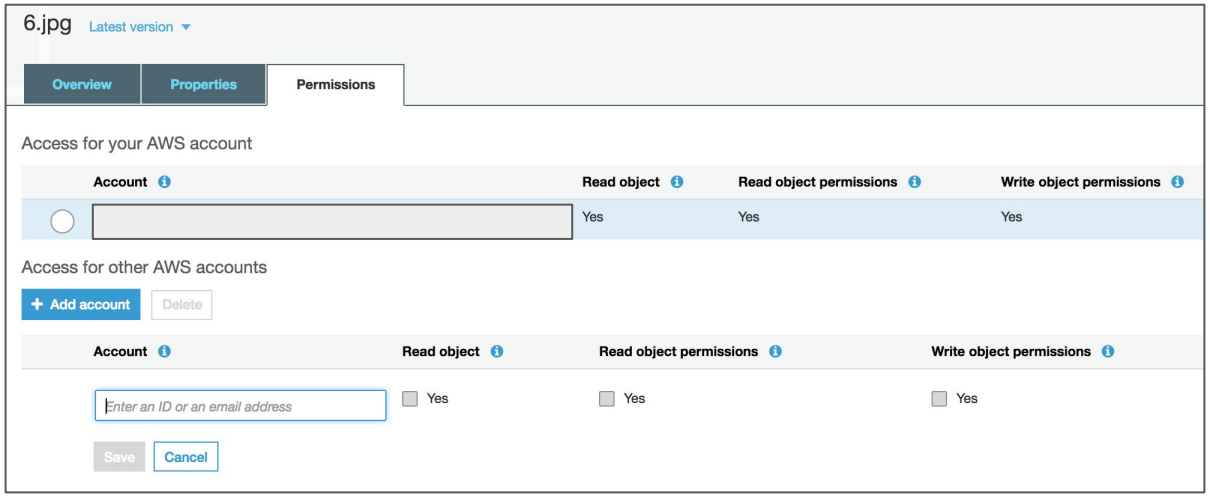
S3 Object

- **Metadata: view in a client side**



S3 Object

- **Permissions or make a public**



AWS 스토리지 선택 옵션

Amazon S3

모든 타입에 대한 내구성 높은 개체 스토리지 서비스

경제적 활용

사용한 만큼만 지불하고, 미리 선타자가 필요 없으며 용량 계산 필요 없음

Amazon Glacier

자주 접근하지 않는 데이터에 대한 백업 서비스

백업 편이성

손쉽게 혼자 관리 가능

데이터 생명 주기에 따라 관리 가능

Amazon EBS

Amazon EC2에 사용할 수 있는 블록 스토리지

활용 용이성

내구성 및 보안성이 높으며, 가상 서버에서 데이터 처리 가능

Amazon EFS

Amazon EC2에 대한 네트워크 스토리지

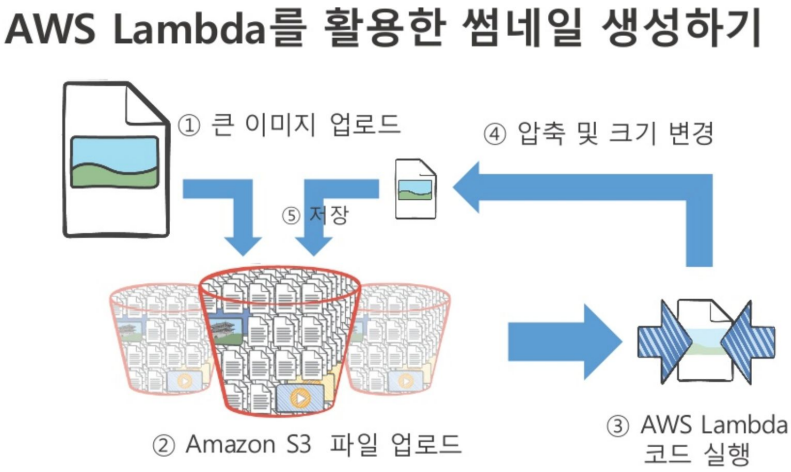
손쉬운 확장성

블록 스토어 관리 비용 절감 및 공유 스토리지 관리 불필요

(현재 Preview로 제공)

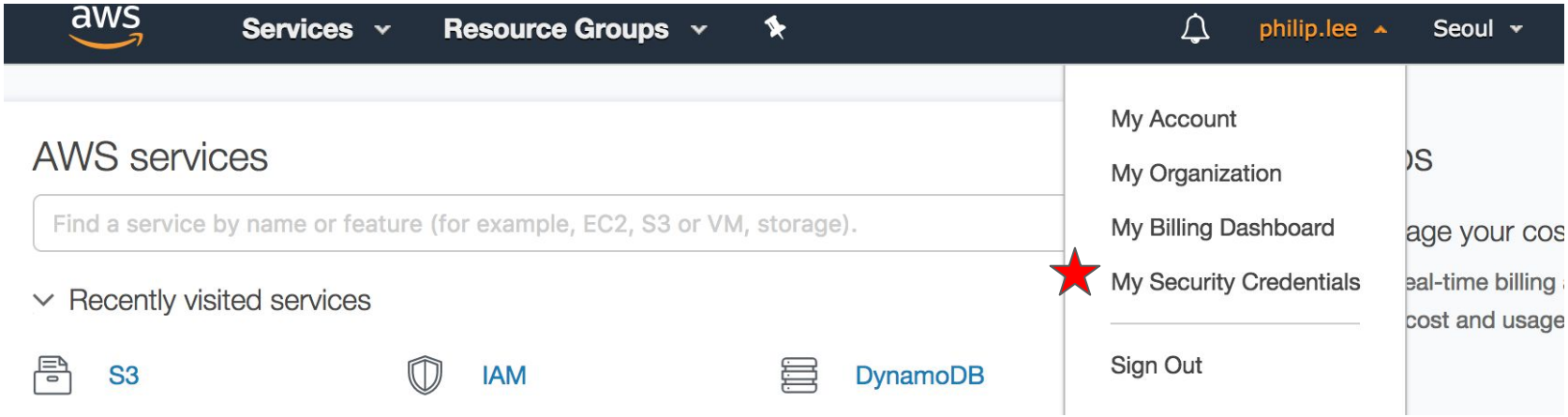
S3 Practical Example

- **Lamba Service**

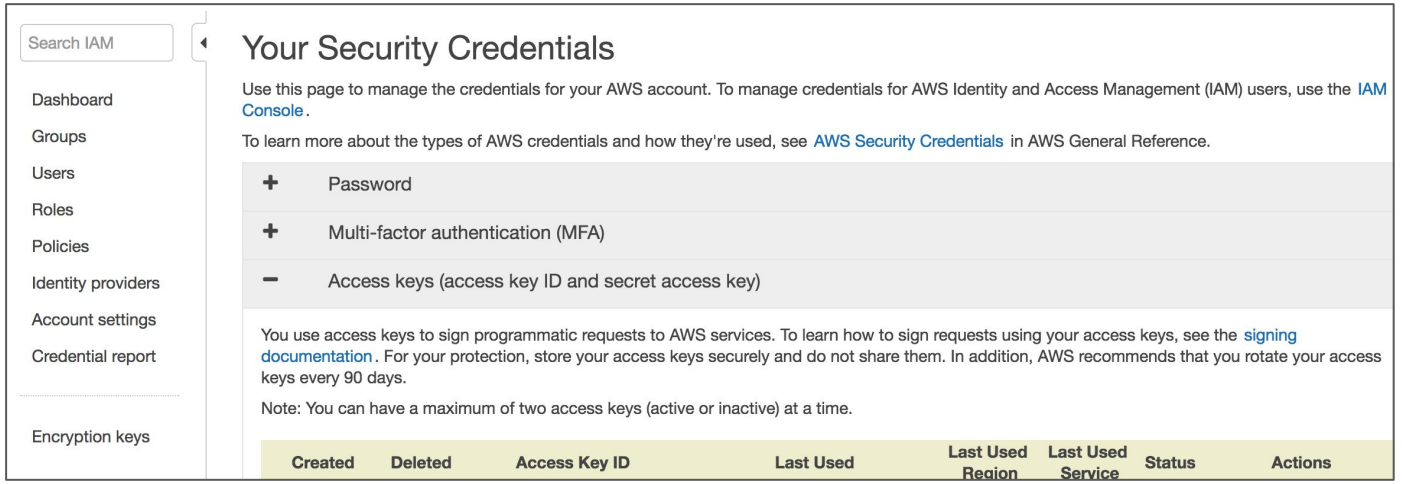


- **S3 practice coding**

S3 Practice


aws
Services
Resource Groups
philip.lee
Seoul
AWS services
Find a service by name or feature (for example, EC2, S3 or VM, storage).
Recently visited services
S3
IAM
DynamoDB
My Account
My Organization
My Billing Dashboard
My Security Credentials
Sign Out

S3 Practice


Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Your Security Credentials
Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.
To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.
+ Password
+ Multi-factor authentication (MFA)
− Access keys (access key ID and secret access key)
You use access keys to sign programmatic requests to AWS services. To learn how to sign requests using your access keys, see the signing documentation. For your protection, store your access keys securely and do not share them. In addition, AWS recommends that you rotate your access keys every 90 days.
Note: You can have a maximum of two access keys (active or inactive) at a time.
Created
Deleted
Access Key ID
Last Used
Last Used Region
Last Used Service
Status
Actions

S3 Practice

IN YOUR MACHINE

>> mkdir ~/.aws
>> vim ~/.aws/config # setting a region

```
1 [default]
2 region = ap-northeast-2
```

>> vim ~/.aws/credentials # setting an authenication key

```
1 [default]
2 aws_access_key_id = AKIA
3 aws_secret_access_key =
```

http://boto3.readthedocs.io/en/latest/guide/configuration.html

S3 Practice

Using BOTO3 in Python to access **S3**

- **pip install boto3**

S3 Practice

- **listing buckets (s3_list.py)**
  >> python s3_list.py

```
import boto3

session = boto3.Session(profile_name ='dev')
s3 = session.resource('s3')
for bucket in s3.buckets.all():
    print(bucket.name)
```

S3 Practice

- **download a file from S3 (s3_run.py)**
  - download ‘2018/file’ you uploaded at first
  - it works ?

```
import os
import boto3
import time

session = boto3.Session(profile_name='dev')
s3 = session.client('s3')

def download(s3, bucket, obj, local_file_path):
    s3.download_file(bucket, obj, local_file_path)
```

S3 Practice

- **upload a file to S3 (s3_run.py)**
  - upload a file to ‘2018/’

```
import os
import boto3
import time

session = boto3.Session(profile_name='dev')
s3 = session.client('s3')

def upload(s3, local_file_path, bucket, obj):
    s3.upload_file(local_file_path, bucket, obj)
```

S3 Practice

- **make your upload file ‘public’**

```
import os
import boto3
import time

session = boto3.Session(profile_name='dev')
s3 = session.client('s3')

def upload(s3, local_file_path, bucket, obj):
    s3.upload_file(local_file_path, bucket, obj)

def make_public_read(s3, bucket, key):
    s3.put_object_acl(ACL='public-read', Bucket=bucket, Key=key)
```

S3 Assignment

make a program in python

- upload files from the certain path to S3 (e.g., argv[1] == '/backup/')
- a upload folder in S3 is $TODAY_YEAR/$TODAY_DATE (e.g., 2018/03-02/)
- make the file public (e.g, argv[2] == 1)
- download the file from S3 (e.g, argv[3] == 1)

S3 Assignment

>> python s3_run.py ‘home/user/phil/’ 1 0

- **upload all files from ‘home/user/phil/’**
- **make those files public**
- **do not download those files**