

AWS S3 버킷 생성 및 DJANGO 연동

Session 14

NEXT X LIKELION 이혁준

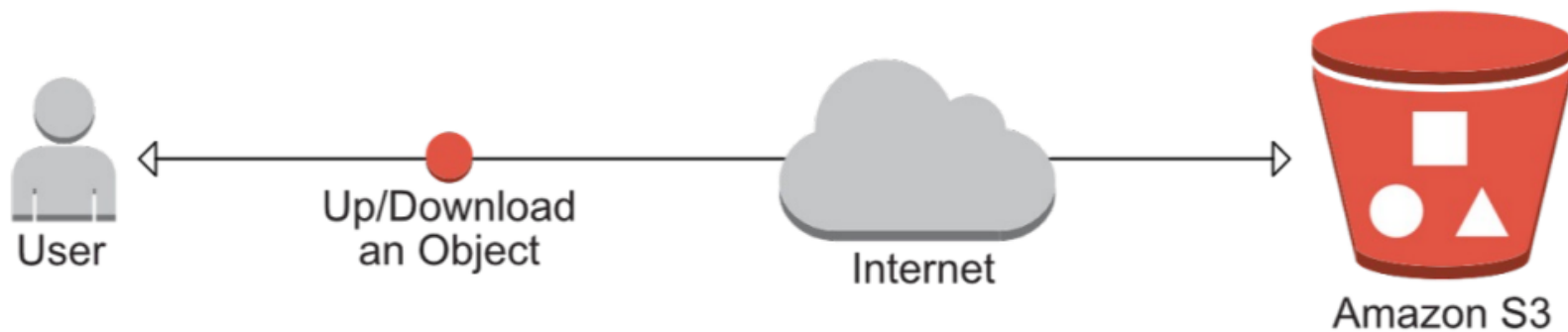
| 목차

1. S3 개념 및 필요성 설명
2. 실습 개괄
3. S3 버킷 생성
4. Django 연동

S3?

S3 = Simple Storage Service

- 사용자가 인터넷 상에 파일을 쉽게 업로드하고, 빠르게 접근할 수 있도록 하는 서비스



| S3?

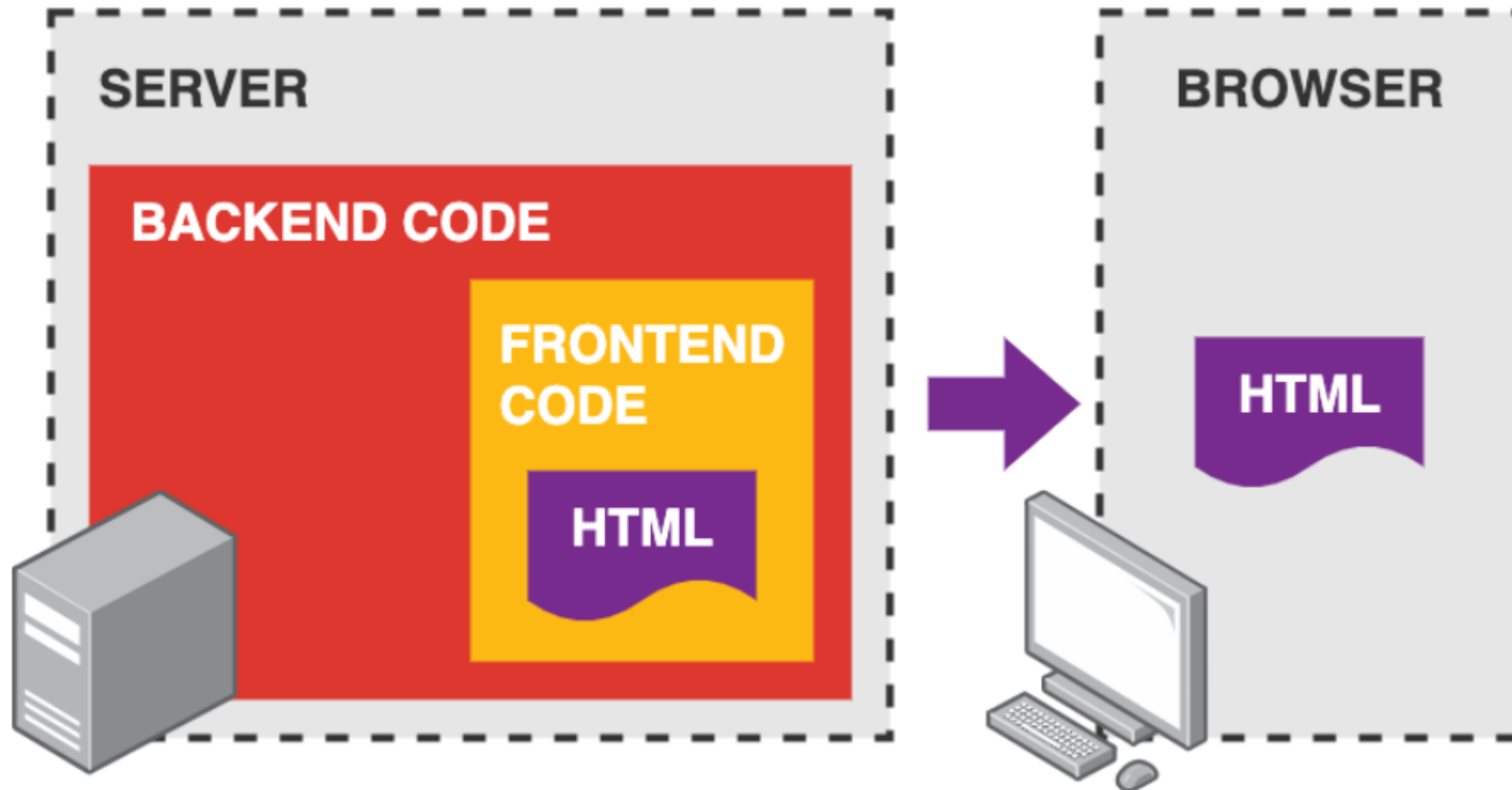
Amazon Simple Storage Service (Amazon S3)는 업계 최고의 확장성과 데이터 가용성 및 보안과 성능을 제공하는 **객체 스토리지 서비스**입니다. 즉, 어떤 규모 어떤 산업의 고객이든 이 서비스를 사용하여 웹 사이트, 모바일 애플리케이션, 백업 및 복원, 아카이브, 엔터프라이즈 애플리케이션, IoT 디바이스, 빅 데이터 분석 등과 같은 다양한 사용 사례에서 원하는 만큼의 데이터를 저장하고 보호할 수 있습니다. Amazon S3는 사용하기 쉬운 관리 기능을 제공하므로 특정 비즈니스, 조직 및 규정 준수 요구 사항에 따라 데이터를 조직화하고 **세부적인 액세스 제어를 구성**할 수 있습니다. Amazon S3는 **99.999999999%의 내구성**을 제공하도록 설계되었으며, 전 세계 기업의 수백만 애플리케이션을 위한 데이터를 저장합니다.

| S3 사용 이유

- S3는 저장 용량이 무한대이고, 파일 입출력에 최적화 되어 있다
- 별도의 클라이언트나 중간 매개체 없이 REST API로도 파일 업로드, 다운로드가 가능하다.
- S3 또한 하나의 서버이기 때문에 S3 자체로 정적인 웹서비스를 제공할 수 있다. 이제 여러분들이 하게 될 REACT는 S3에 배포된다.
- 보통 정적 웹사이트를 S3에, WAS를 EC2에 배포해서 서로 소통하도록 구축한다.

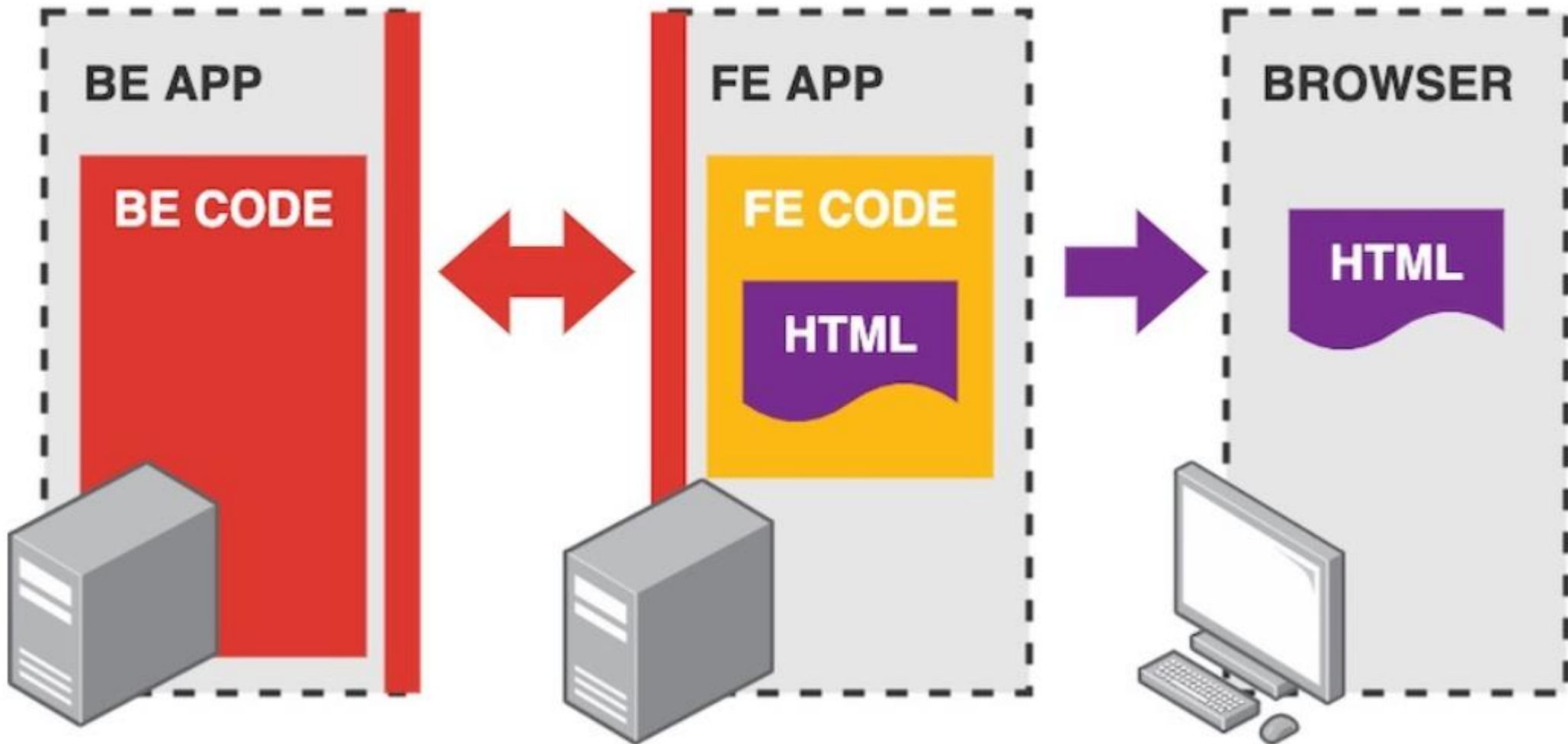
정적 웹사이트, REST..?

기존의 Monolithic Architecture



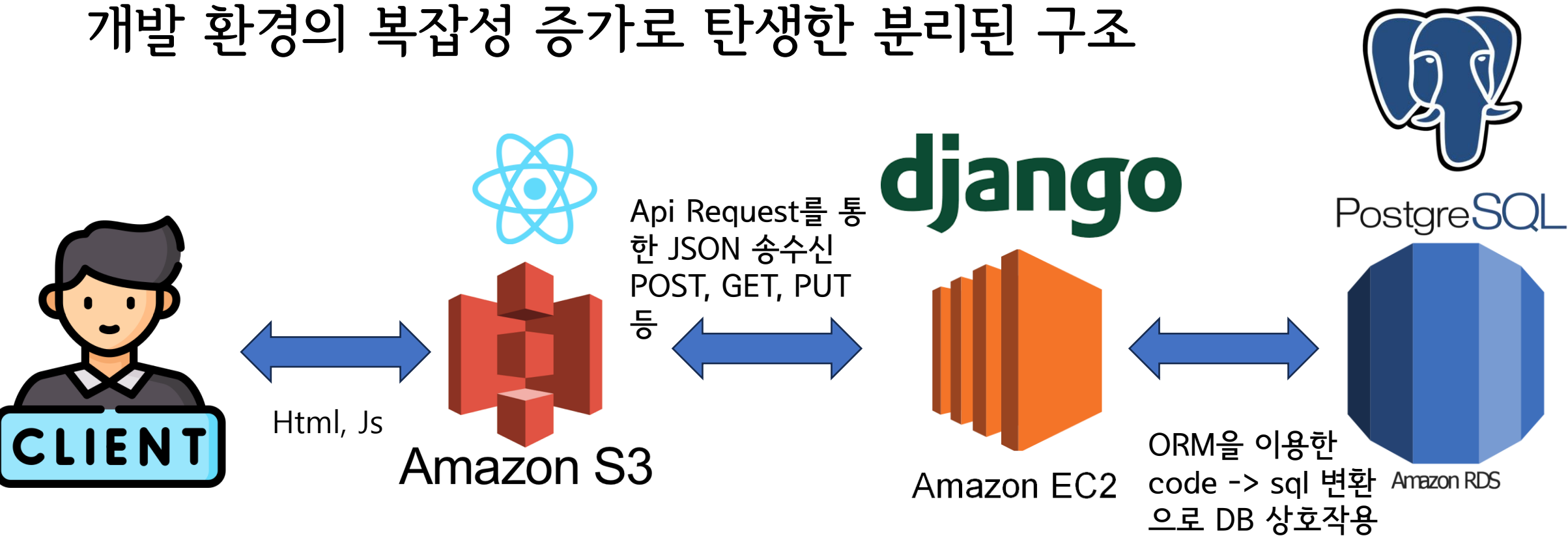
정적 웹사이트, REST..?

개발 환경의 복잡성 증가로 탄생한 분리된 구조



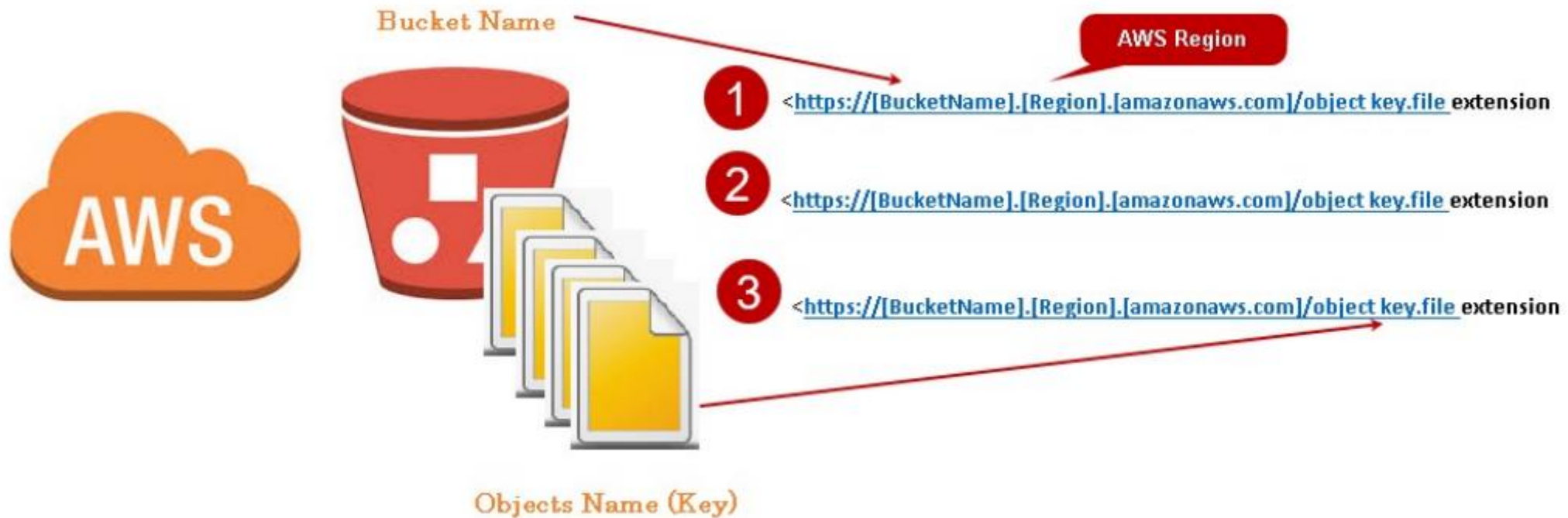
| 정적 웹사이트, REST..?

개발 환경의 복잡성 증가로 탄생한 분리된 구조



S3 구성 : 버킷과 객체

- 버킷(Bucket): 저장된 객체들의 묶음 (하나의 디비로 생각하면 편해요)
- 객체(Object): 데이터를 구성하는 저장 단위



| S3: 실습 개괄

- 오늘 할 일
 - S3 버킷 생성
 - S3 접속 권한을 위한 IAM 생성
 - S3 버킷 권한 정책 설정
 - 미리 만들어둔 깃헙 레포 포크(Fork)
 - Django에 버킷 연결
 - 실제로 이미지 업로드
 - 행복한 마무리...

S3: 버킷 생성

<https://ap-northeast-2.console.aws.amazon.com/s3/get-started?region=ap-northeast-2>

Storage

Amazon S3

Store and retrieve any amount
of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

S3: 버킷 생성: 버킷 이름 정하기(Unique)

<https://ap-northeast-2.console.aws.amazon.com/s3/get-started?region=ap-northeast-2>

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
Asia Pacific (Seoul) ap-northeast-2

Bucket name Info

next-session14-bucket-yourname

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

| S3: 버킷 생성

지금은 이렇게 다 허용하지만, 버킷 생성 이후에 권한 설정 다시 할 것입니다.

모든 퍼블릭 액세스를 허용해주세요

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ Block *all* public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ Block public access to buckets and objects granted through *new* access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ Block public access to buckets and objects granted through *any* access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ Block public access to buckets and objects granted through *new* public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the

S3: 버킷 생성

다 했으면 생성!

Encryption type [Info](#)

☒ Server-side encryption with Amazon S3 managed keys (SSE-S3)

☐ Server-side encryption with AWS Key Management Service keys (SSE-KMS)

☐ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the **Storage** tab of the [Amazon S3 pricing page](#). [↗](#)

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#) [↗](#)

☐ Disable

☒ Enable

▶ Advanced settings

i

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel

Create bucket

S3: IAM 생성

<https://us-east-1.console.aws.amazon.com/iam/home?region=ap-northeast-2#/users>

IAM > Users

▼ Ready to streamline human access to AWS and cloud apps? Dismiss Manage workforce users

Identity Center is enabled. We recommend managing workforce users' access to AWS accounts and cloud applications in Identity Center.

[Learn more](#) | [Watch how it works](#)

Users (0) [Info](#) Refresh Delete Create user

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

< 1 > Settings

	User name	Path	Group	Last activity	MFA	Password age	Console last sign-in
No resources to display							

S3: IAM 생성

Specify user details

User details

User name

next_session14_iam

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

❗ If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

S3: IAM 생성

Django에서 접속할 때 사용하기 위해 관련 정책을 직접 추가해 줄 것입니다.

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☐ Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

S3: IAM 생성

Django에서 접속할 때 사용하기 위해 관련 정책을 직접 추가해 줄 것입니다.

Permissions policies (1/1197)

Choose one or more policies to attach to your new user.

Filter by Type

Q amazons3 X All types 5 matches < 1 > ⚙

	Policy name ↗	Type	Attached entities
<input checked="" type="checkbox"/>	AmazonS3FullAccess	AWS managed	2
<input type="checkbox"/>	AmazonS3ObjectLambdaExecution...	AWS managed	0
<input type="checkbox"/>	AmazonS3OutpostsFullAccess	AWS managed	0
<input type="checkbox"/>	AmazonS3OutpostsReadOnlyAccess	AWS managed	0
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS managed	0

► Set permissions boundary - optional

Cancel Previous Next

S3: IAM 생성


유저 생성!

User details

User name next_session14_iam	Console password type None	Require password reset No
---------------------------------	-------------------------------	------------------------------

Permissions summary

< 1 >

Name 	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create user

S3: IAM 생성


유저 생성!

User details

User name next_session14_iam	Console password type None	Require password reset No
---------------------------------	-------------------------------	------------------------------

Permissions summary

< 1 >

Name 	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create user

S3: IAM 생성

이제 django 서버에서 접속할 수 있도록 Programmatic Call을 허용하러 가봅시다

Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Q Search



Delete

Create user

< 1 > ⚙



User name



Path



Group: ▾

Last activity



MFA



Password age



Console last sign-i



[next_session14_iam](#)

/

0

-

-

-

-


S3: IAM 생성

이제 django 서버에서 접속할 수 있도록 Programmatic Call을 허용하러 가봅시다

next_session14_iam [Info](#)

Summary

ARN

 arn:aws:iam::527222766651:user/next_session14_iam

Created

May 01, 2024, 15:51 (UTC+09:00)

Console access

Disabled

Access key 1

[Create access key](#)

Last console sign-in

-

[Permissions](#)

[Groups](#)

[Tags](#)

[Security credentials](#)


[Access Advisor](#)


S3: IAM Access Key 생성

이제 django 서버에서 접속할 수 있도록 Programmatic Call을 허용하러 가봅시다

Access keys (0)

[Create access key](#)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#) 

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#) 

[Create access key](#)

S3: IAM Access Key 생성

원래는 Application running on an AWS compute service를 하고, 사실 User가 아닌, role을 생성하고 이걸 EC2에 엮어서 사용하는 것이 베스트인데, 지금은 API 사용을 배우는 상황이므로 Credential을 생성합니다.

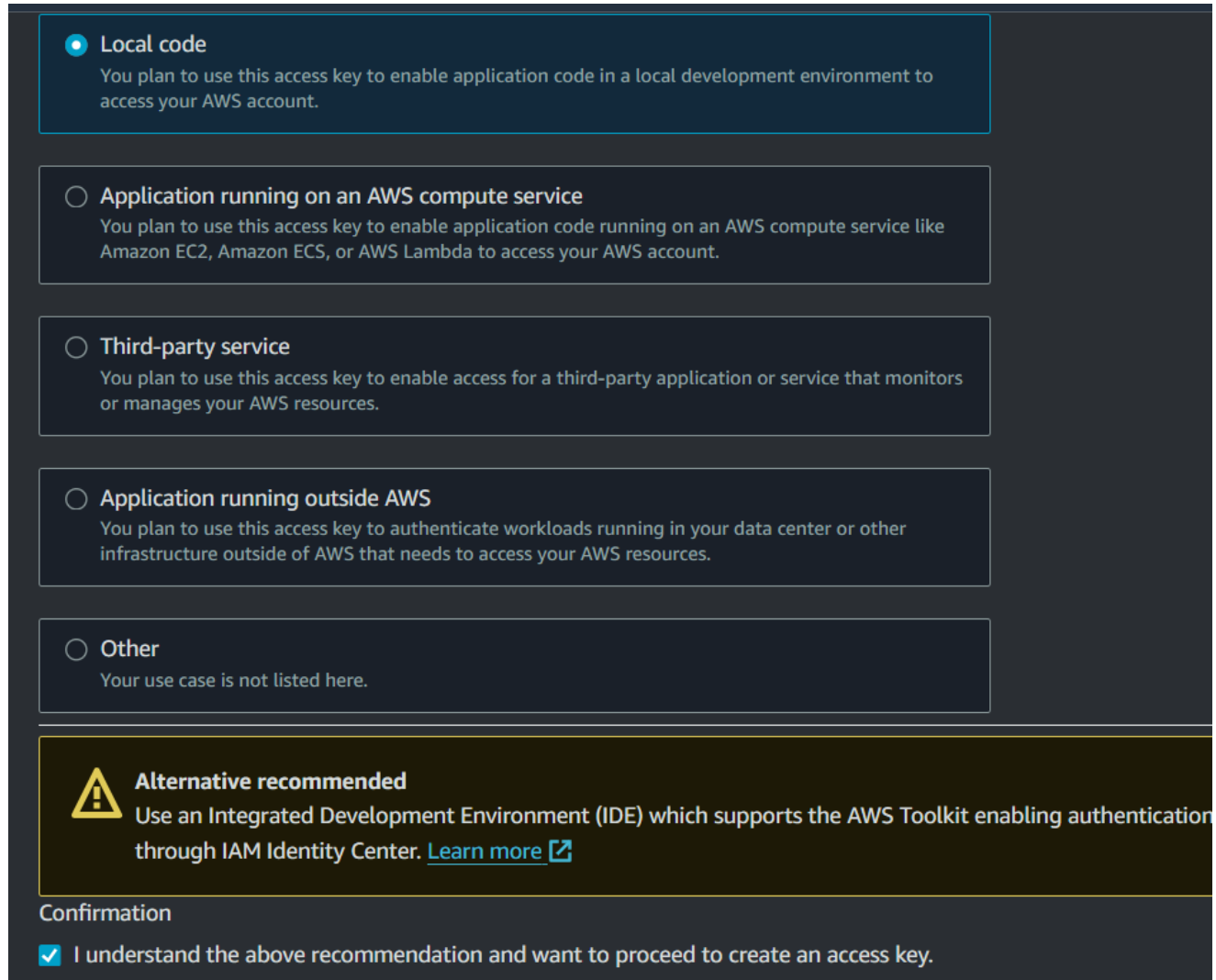
☒ Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

☐ Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

☐ Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

☐ Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

☐ Other
Your use case is not listed here.

 **Alternative recommended**
Use an Integrated Development Environment (IDE) which supports the AWS Toolkit enabling authentication through IAM Identity Center. [Learn more](#)

Confirmation

☒ I understand the above recommendation and want to proceed to create an access key.

S3: IAM Access Key 생성

Set description tag - *optional* Info

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

Cancel

Previous

Create access key

S3: IAM Access Key 생성

✔ Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1

[Access key best practices & alternatives](#)

Step 2 - optional

[Set description tag](#)

Step 3

Retrieve access keys

Retrieve access keys [Info](#)

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key

AKIA XVQM7Q52RF4JU5R

Secret access key

***** [Show](#)

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

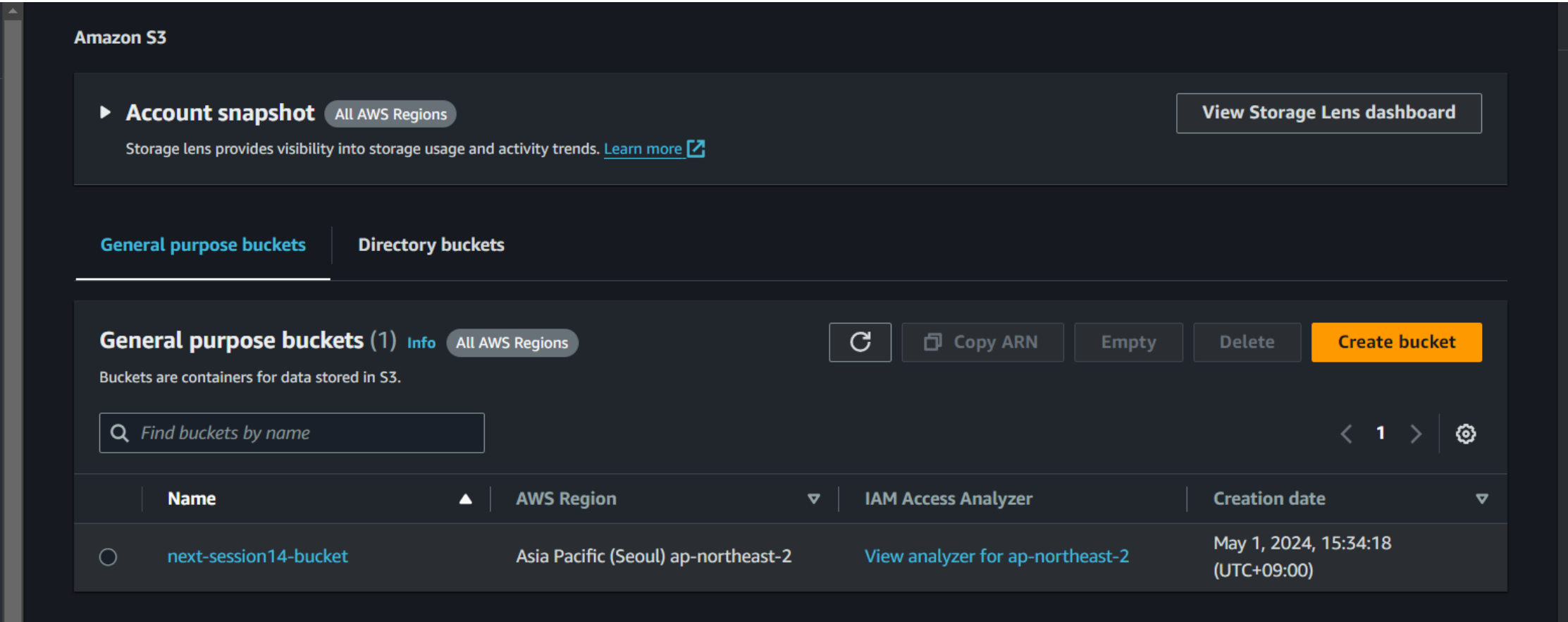
[Download .csv file](#)

[Done](#)

| S3: 버킷 권한 설정

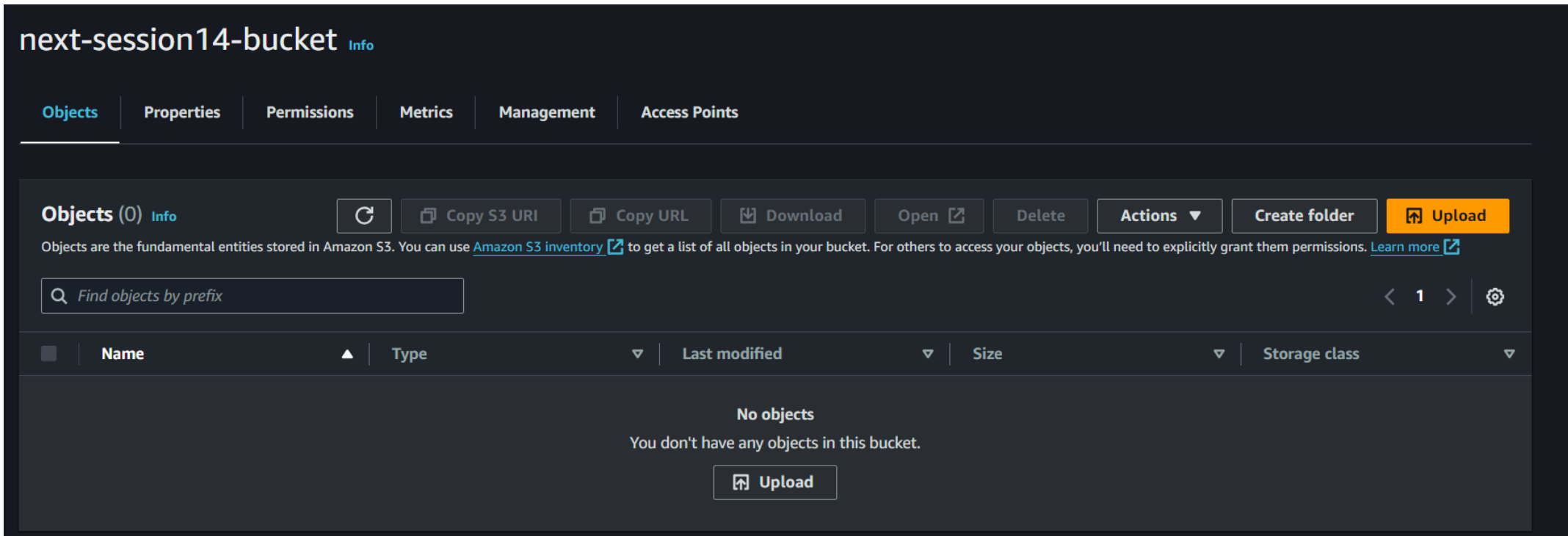
S3로 이동해서 버킷 설정 창으로 들어가기

<https://ap-northeast-2.console.aws.amazon.com/s3/home?region=ap-northeast-2>



S3: 버킷 권한 설정

Permissions 눌러주세요!



S3: 버킷 권한 설정

버킷 정책에서 수정을 눌러서 보기 권한은 익명의 모두에게 제공하지만, 업로드 업데이트 같은 권한은 서버에서만 가능하도록 정할 것입니다.

Bucket policy

[Edit](#)[Delete](#)

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

No policy to display.

[Copy](#)

S3: 버킷 권한 설정

정책 생성기를 눌러주세요!


Edit bucket policy [Info](#)

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

[Policy examples](#) [Policy generator](#)

Bucket ARN

 `arn:aws:s3:::next-session14-bucket`

Policy

1	
---	--

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

[+ Add new statement](#)

S3: 버킷 권한 설정

S3 버킷에 대한 정책 설정임을 명시하고



AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to [Amazon Web Services \(AWS\)](#) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are [sample policies](#).

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an [IAM Policy](#), an [S3 Bucket Policy](#), an [SNS Topic Policy](#), a [VPC Endpoint Policy](#), and an [SQS Queue Policy](#).

Select Type of Policy

SQS Queue Policy

SQS Queue Policy

S3 Bucket Policy

VPC Endpoint Policy

IAM Policy

SNS Topic Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See [a description of elements](#) that you can use in statements.

Effect ☒ Allow ☐ Deny

Principal

Use a comma to separate multiple values.

AWS Service ☐ All Services ('*')

Use multiple statements to add permissions for more than one service.

Actions ☐ All Actions ('*')

Amazon Resource Name (ARN)

ARN should follow the following format: `arn:aws:sqs:${Region}:${Account}:${QueueName}`.
Use a comma to separate multiple values.

S3: 버킷 권한 설정

S3 내의 파일에 대한 읽기 권한을 모두에게 제공하는 것으로 설정합니다.
파일 = Object

Effect ☒ Allow ☐ Deny

Principal

Use a comma to separate multiple values.

AWS Service ☐ All Services ('*')

Use multiple statements to add permissions for more than one service.

Actions ☐ All Actions ('*')

Amazon Resource Name (ARN)

☐ GetMultiRegionAccessPointPolicy
☐ GetMultiRegionAccessPointPolicyStatus
☐ GetMultiRegionAccessPointRoutes
☐ GetObject
☐ GetObjectAcl
☐ GetObjectAttributes
☐ GetObjectLegalHold
☐ GetObjectRetention

must select at least one Action

Step 3: Generate Policy

A policy is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

add one or more statements above to generate a policy.

S3: 버킷 권한 설정

해당 권한이 사용될 리소스를 설정합니다. ARN에 아래 형식에 맞게 작성하고 Add Statement를 써주세요

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See [a description of elements](#) that you can use in statements.

Effect ☒ Allow ☐ Deny

Principal
Use a comma to separate multiple values.

AWS Service ☐ All Services ('*')
Use multiple statements to add permissions for more than one service.

Actions ☐ All Actions ('*')

Amazon Resource Name (ARN)
ARN should follow the following format: arn:aws:s3:::\${BucketName}/\${KeyName}.
Use a comma to separate multiple values.

[Add Conditions \(Optional\)](#)

| S3: 버킷 권한 설정

이제 해당 정책에 맞는 설정을 생성해주세요

You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
• *	Allow	• s3:GetBucketPolicy	arn:aws:s3:::next-session14-bucket/*	None

Step 3: Generate Policy

A *policy* is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

Generate Policy

Start Over

S3: 버킷 권한 설정

정책 JSON을 복사하고,

Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor. Changes made below will **not be reflected in the policy generator tool**.

```
{
  "Id": "Policy1714556348995",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1714556329136",
      "Action": [
        "s3:GetBucketPolicy"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::next-session14-bucket/*",
      "Principal": "*"
    }
  ]
}
```

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether

Close

S3: 버킷 권한 설정

S3 버킷 정책에 붙여넣어주세요. 다했으면 아래에 save!

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

Policy examples [Policy generator](#)

Bucket ARN

[arn:aws:s3::next-session14-bucket](#)

Policy

```
1 {
2   "Id": "Policy1714556137950",
3   "Version": "2012-10-17",
4   "Statement": [
5     {
6       "Sid": "Stmt1714555753689",
7       "Action": [
8         "s3:GetObject"
9       ],
10      "Effect": "Allow",
11      "Resource": "arn:aws:s3::next-session14-bucket",
12      "Principal": "*"
13    }
14  ]
15 }
```

Edit statement

Select a statement

Select an existing statement or add a new statement

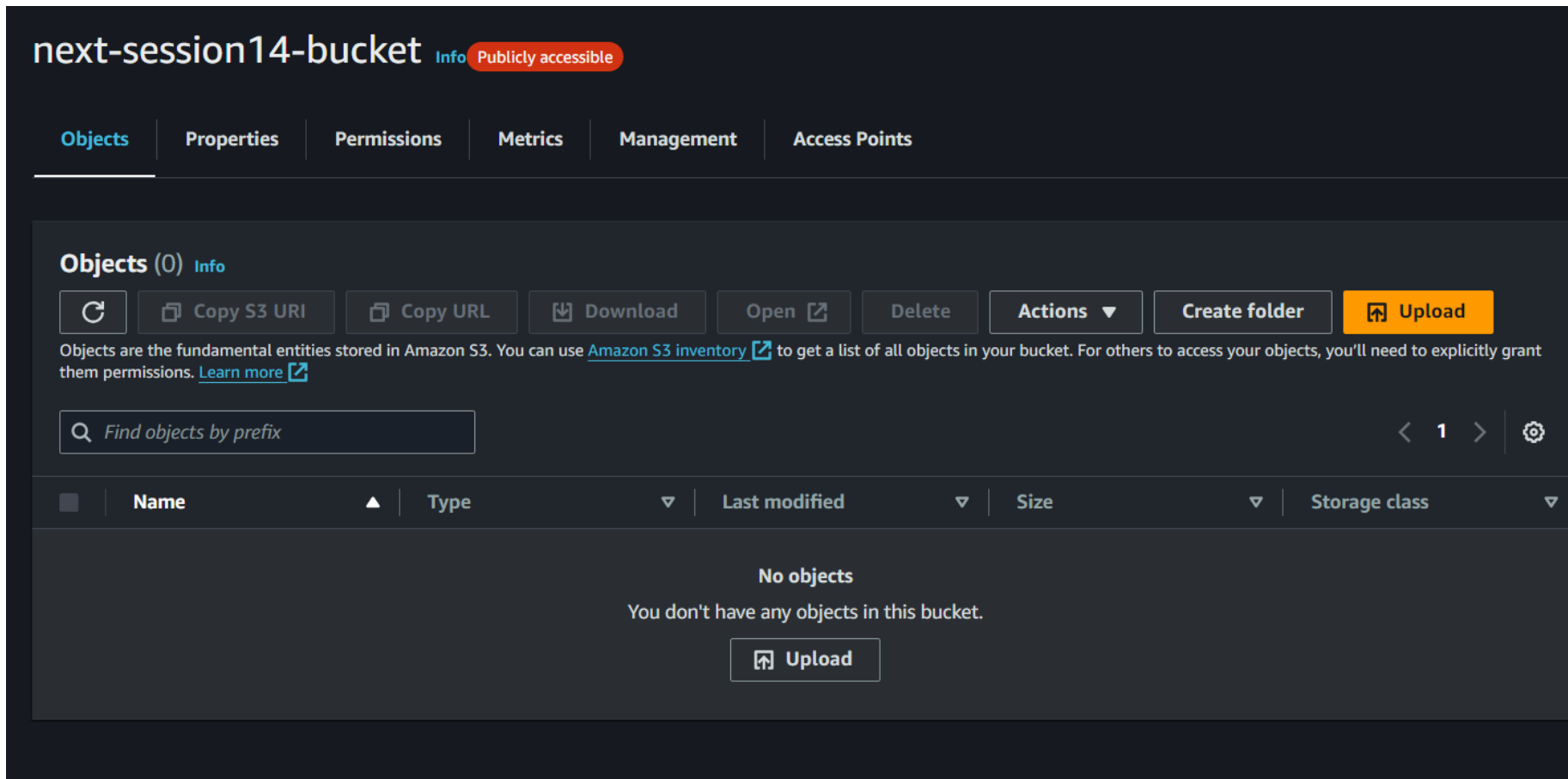
+ Add new statement

Preview external access

Cancel Save changes

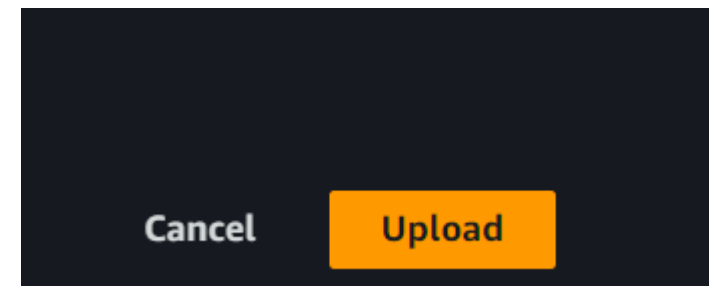
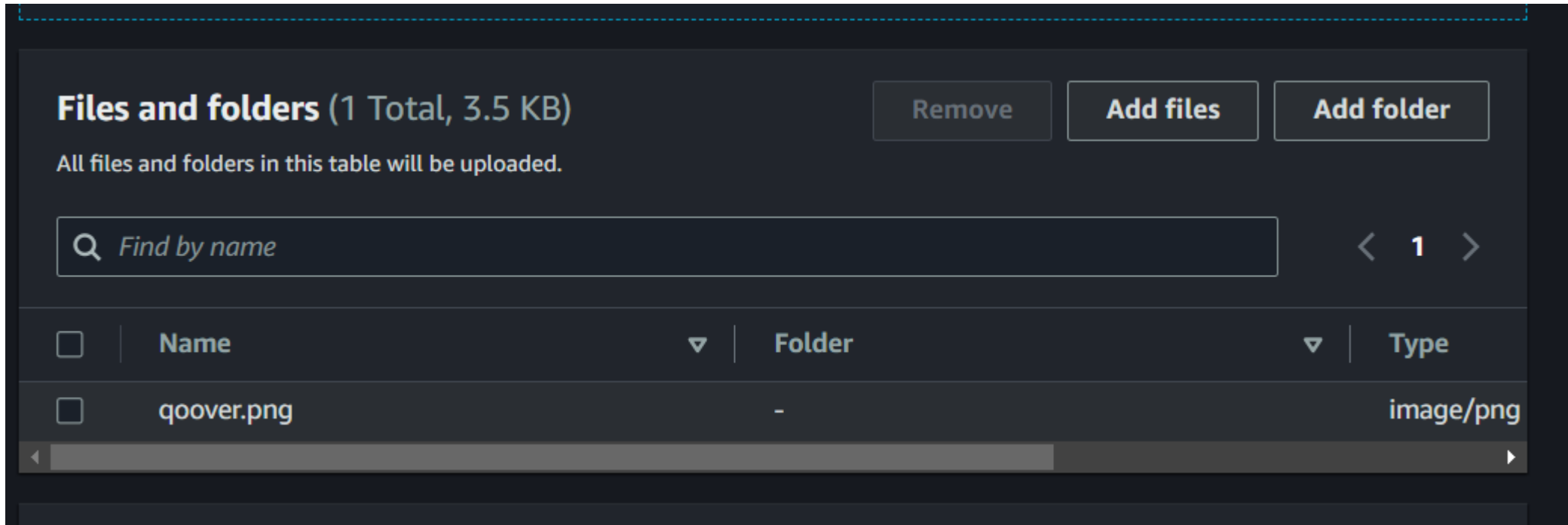
S3: 버킷 테스트

이제 다시 버킷 메인으로 돌아와서, 테스트용 이미지를 하나 올려보죠



S3: 버킷 테스트

이제 다시 버킷 메인으로 돌아와서, 테스트용 이미지를 하나 올려보죠



S3: 버킷 테스트

Summary

Destination

s3://next-session14-bucket

Succeeded

✓ 1 file, 3.5 KB (100.00%)

Failed

⋮ 0 files, 0 B (0%)

Files and folders

Configuration

Files and folders (1 Total, 3.5 KB)

🔍 Find by name

Name	Folder	Type	Size	Status	Error
qoover.png	-	image/png	3.5 KB	✓ Succeeded	-

S3: 버킷 테스트

qoover.png Info

Copy S3 URI

Download

Open

Object actions ▼

Properties

Permissions

Versions

Object overview

Owner

f967e4cc93ea6d806e584e20ce463dc10eb3a6bcadc677aa5eeeba5858b5f7ff

AWS Region

Asia Pacific (Seoul) ap-northeast-2

Last modified

May 1, 2024, 18:41:27 (UTC+09:00)

Size

3.5 KB

Type

png

Key

S3 URI

s3://next-session14-bucket/qoover.png

Amazon Resource Name (ARN)

arn:aws:s3:::next-session14-bucket/qoover.png

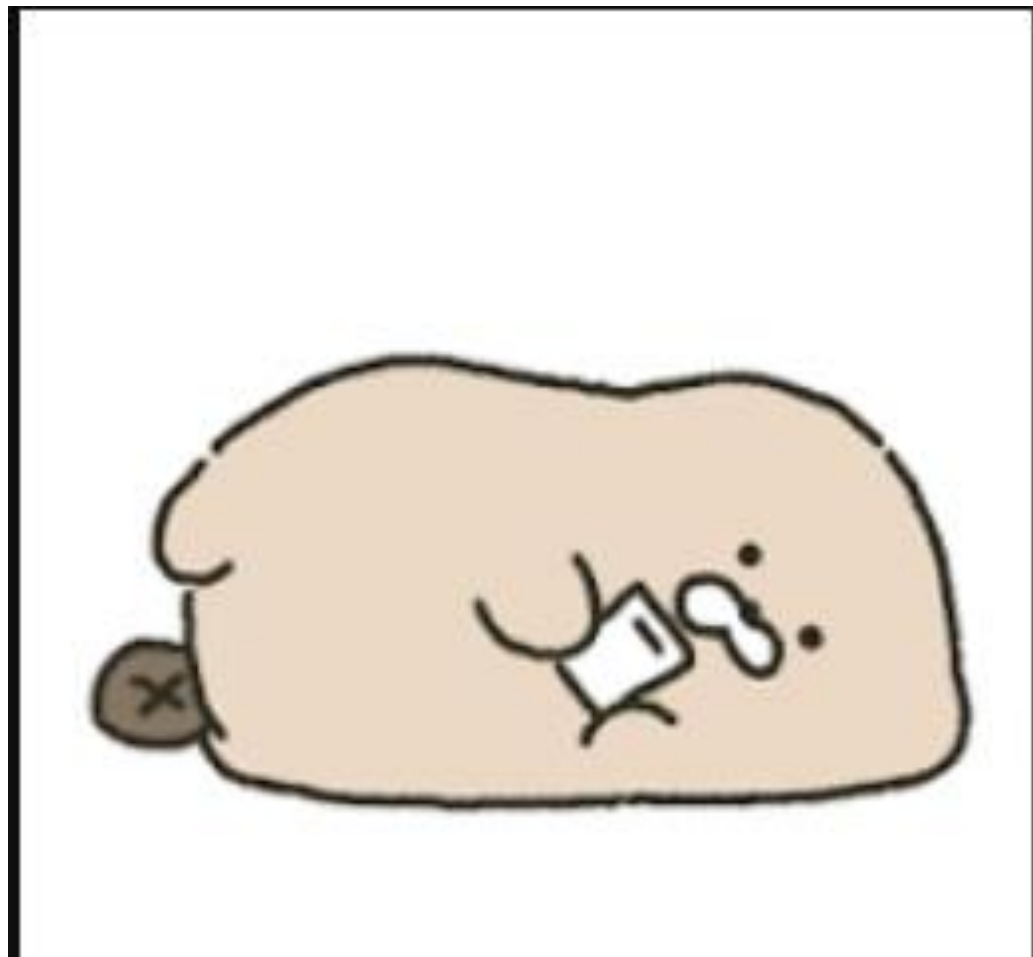
Entity tag (Etag)

8bc9a5f09c81e82a060d578939c0f451

Object URL

https://next-session14-bucket.s3.ap-northeast-2.amazonaws.com/qoover.png

| S3: 버킷 테스트



S3: Django 연동

전에 제가 올려놓은 Django 프로젝트의 조금 수정 버전을 그대로 차용하겠습니다.

다만, 시간을 절약하기 위해서 이번 세션에서 중요하다고 판단되지 않은 내용은 제가 임의로 작성을 완료했고, 여러분들은 필수적인 내용만 함께 실습하면 됩니다.

<https://github.com/newxxson/next-session-11-hw.git>

오늘은 배포까지 하기에는 시간이 부족하므로 로컬에서 진행합니다.

오늘 할 일

- Static storage S3로 이동 및 관련 파일 전체 업로드
- File storage S3로 이동
- 블로그 글 쓸 때 사진도 업로드 가능하도록 수정하고, S3에 업로드하고 Url 제공하기

S3: Static Storage 로컬에서 S3로 변경

- 현재 Static은 다 여러분 각자 app 아래의 static 아래에 저장되어 있고, django에서 알아서 html에 주입해줍니다.
- 이걸 배포할 때 nginx에서 사용할 수 있도록 한 곳에 모으게 된다면 (python manage.py collectstatic) 그때부터는 django.contrib.staticfiles.storage.StaticFilesStorage에서 관리하게 됩니다.
- 문제는 이러면 아직 WAS 서버에서 해당 파일들을 제공하는 것이므로, 비효율적입니다.
- S3로 분리해보겠습니다.
- 이거 좀 해주세요
- `$pip install django, pytz, boto3 django-storages django-environ`

S3: 보안 정보를 담는 곳 .env

접속 키, 아이디, 주소 등의 정보는 서버 코드에 하드 코딩하면 당연히 github에 올렸을 때 심각한 보안 문제를 야기할 수 있습니다. 그래서 보통 이런 중요 정보들은 .env라는 파일을 별도로 만들어 함께 보관하고, 실제 코드에서는 .env 파일에서 변수로 불러와 사용하는 식으로 보안을 강화합니다.

오늘은 그래서 S3를 접속하기 위한 정보들을 .env에 저장할 것입니다!

Base_auth/settings.py에서 S3를 연동하기 위한 보안 설정과 추가 설정을 진행합니다.
전에 다운 받은 Access Key를 찾아주세요!

찾은 다음에 Vscode에서 루트 디렉토리에 .env라는 파일을 작성한 뒤에 다음 코드를 작성해주세요

```
AWS_S3_ACCESS_KEY_ID=  
AWS_S3_SECRET_ACCESS_KEY=  
AWS_STORAGE_BUCKET_NAME=<여러분 버킷 이름 next-session14-~>  
AWS_S3_REGION_NAME=ap-northeast-2  
AWS_S3_SIGNATURE_VERSION=s3v4
```

S3: 보안 정보를 담는 곳 .env

이제 이걸 import하는 코드를 작성합니다. Base_auth/settings.py에 가주세요

```
from pathlib import Path
import os
import environ

# Build paths inside the project like this: BASE_DIR / 'subdir'
BASE_DIR = Path(__file__).resolve().parent.parent

env = environ.Env()
environ.Env.read_env(os.path.join(BASE_DIR, ".env"))

AWS_S3_ACCESS_KEY_ID = env("AWS_S3_ACCESS_KEY_ID")
AWS_S3_SECRET_ACCESS_KEY = env("AWS_S3_SECRET_ACCESS_KEY")
AWS_STORAGE_BUCKET_NAME = env("AWS_STORAGE_BUCKET_NAME")
AWS_S3_REGION_NAME = env("AWS_S3_REGION_NAME")
AWS_S3_SIGNATURE_VERSION = env("AWS_S3_SIGNATURE_VERSION")
```

S3: 보안 정보를 담는 곳 .env

이제 실제 설정을 하는 코드를 작성합니다. Base_auth/settings.py 더 아래를 가세요

```
AWS_S3_CUSTOM_DOMAIN = "%s.s3.%s.amazonaws.com" % (  
    AWS_STORAGE_BUCKET_NAME,  
    AWS_S3_REGION_NAME,  
)  
  
# Static files (CSS, JavaScript, Images)  
# https://docs.djangoproject.com/en/5.0/howto/static-files/  
  
STATIC_URL = "https://%s/static/" % AWS_S3_CUSTOM_DOMAIN  
STATICFILES_STORAGE = "base_auth.storage.StaticStorage"
```

S3: 보안 정보를 담는 곳 .env

Boto default인 S3Boto3Storage를 대체하기 위한 커스텀 StaticStorage를 제작합니다.
base_auth/storage.py를 만들어주세요

```
from storages.backends.s3boto3 import S3Boto3Storage

class StaticStorage(S3Boto3Storage):
    location = "static"
```

S3: 보안 정보를 담는 곳 .env

이제 static 파일을 한 곳에 모아 업로드하는 명령을 수행합니다!

```
(venv) PS C:\Users\duddn\PycharmProjects\testAWS1> python .\manage.py collectstatic

You have requested to collect static files at the destination
location as specified in your settings.

This will overwrite existing files!
Are you sure you want to do this?

Type 'yes' to continue, or 'no' to cancel: yes

130 static files copied.
```


S3: 보안 정보를 담는 곳 .env

잘 올라가는 것을 확인할 수 있습니다!(blog는 안 보이는 것이 정상이에요)

next-session14-bucket

InfoPublicly accessible

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (2) Info

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

< 1 >

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	blog/	Folder	-	-	-
<input type="checkbox"/>	static/	Folder	-	-	-

| S3: css 확인

Python runserver를 하고 css 주소를 확인해도 url이 제대로 설정되어 있는 것을 확인할 수 있습니다.

Name	✕ Headers	Preview	Response	Initiator	>>
blog/	▼ General				
base.css	Request URL: https://next-session14-bucket.s3.ap-northeast-2.amazonaws.com/home.css Request Method: GET Status Code: 200 OK (from memory cache) Remote Address: 52.219.60.127:443 Referrer Policy: same-origin				
home.css					
co3KmW9ljjAjdoyPCM3T3NGs...					
co3KmW9ljjAjdoyPCM3T3NGs...					
co3KmW9ljjAjdoyPCM3T3NGs...					
co3KmW9ljjAjdoyPCM3T3NGs...					
co3KmW9ljjATdOrY.woff2					
AMOQz46as3KIBPemhXo8.wof...					
favicon.png					

| S3:django에서 사진 업로드 하기

이제 블로그 글에 사진을 추가할 수 있도록 파일 입출력을 s3에 연동하겠습니다.

이미 되어있는 것

- Html에서 관련 로직 및 구조
- Views에서 사용할 처리 관련 로직 (utils.py)
- Models.py에서 저장할 attribute

해야하는 것

- settings.py에서 s3 연동하기
- views.py에서 로직 구현하기

S3:django에서 사진 업로드 하기

```
You, 2 hours ago • Uncommitted changes  
DEFAULT_FILE_STORAGE = "storages.backends.s3boto3.S3Boto3Storage"
```

```
@login_required  
def new(request):  
    if request.method == "POST":  
        title = request.POST["title"]  
        content = request.POST["content"]  
  
        new_post = Post(title=title, content=content, creator=request.user)  
  
        image = request.FILES["image"]  
        if image:  
            image_name = get_new_file_name(image, request.user, title)  
  
            saved_path = default_storage.save(image_name, image)  
            image_url = default_storage.url(saved_path)  
  
            new_post.image_url = image_url  
            new_post.save()  
  
            return redirect("detail", new_post.pk)  
  
    return render(request, "new.html")
```

S3:django에서 사진 업로드 하기

```
@login_required
@check_is_creator_or_admin(Post, "post_pk")
def edit(request, post_pk):
    post = Post.objects.get(pk=post_pk)

    if request.method == "POST":
        title = request.POST["title"]
        content = request.POST["content"]
        Post.objects.filter(pk=post_pk).update(title=title, content=content)

        # Image handling
        image = request.FILES["image"]
        if image:
            image_name = get_new_file_name(image, request.user, title)

            saved_path = default_storage.save(image_name, image)
            image_url = default_storage.url(saved_path)

            post.image_url = image_url
            post.save()

        return redirect("detail", post_pk)

    return render(request, "edit.html", {"post": post})
```

| S3:django에서 사진 업로드 하기

Makemigration, migrate 진행하고 서버를 구동해서 새로운 글 쓰기로 가면

NEW!

제목을 입력하세요!

내용을 입력하세요!

Choose File

No file chosen

작성하기

| S3:django에서 사진 업로드 하기

Makemigration, migrate 진행하고 서버를 구동해서 새로운 글을 쓰면

제목: testing

last viewed: May 1, 2024, 2:52 p.m. , by: testtest

내용: 1234

홈 화면

수정하기

삭제하기

댓글 작성



오늘의 과제

1. 오늘 만든 내용 서버에 배포하기 -> 직접 새로 레포 파서 만든 다음 올리고 clone ...
2. 댓글에 사진 첨부할 수 있도록 만들기
3. (선택)사진 삭제할 수 있도록 하기
 - 현재 저희 서버는 글을 작성한 후에 삭제하거나, 다른 사진으로 수정을 해도 서버에 올라간 이미지는 삭제되지 않습니다.
 - 해당 글을 삭제하거나, 사진을 새로운 사진으로 변경할 때 기존 사진을 삭제하는 로직을 구현해 주세요!
 - 현재 저희는 FileField를 사용하지 않고 UrlField를 사용하고 있기 때문에 on delete로직을 새로 구현해야 합니다.
 - <https://docs.djangoproject.com/en/dev/topics/db/models/#overriding-predefined-model-methods>
 - Delete 메서드를 수정해야겠죠? 이외에도, edit에서 온 파일과 기존 디비에서의 파일이 다른지를 검증하는 로직도 필요합니다.