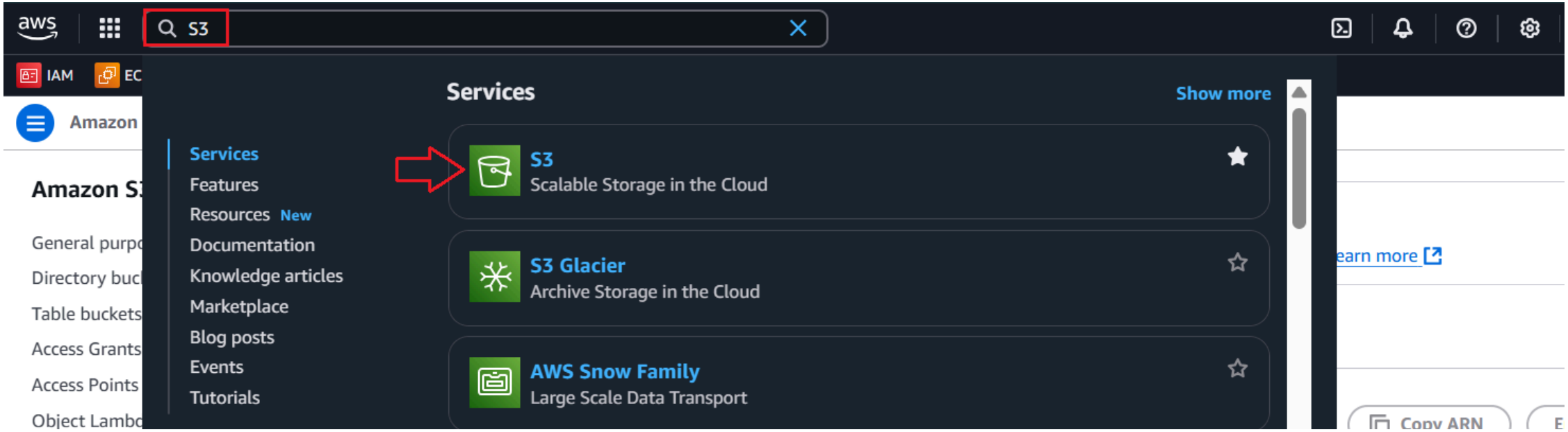
Layer 생성하기

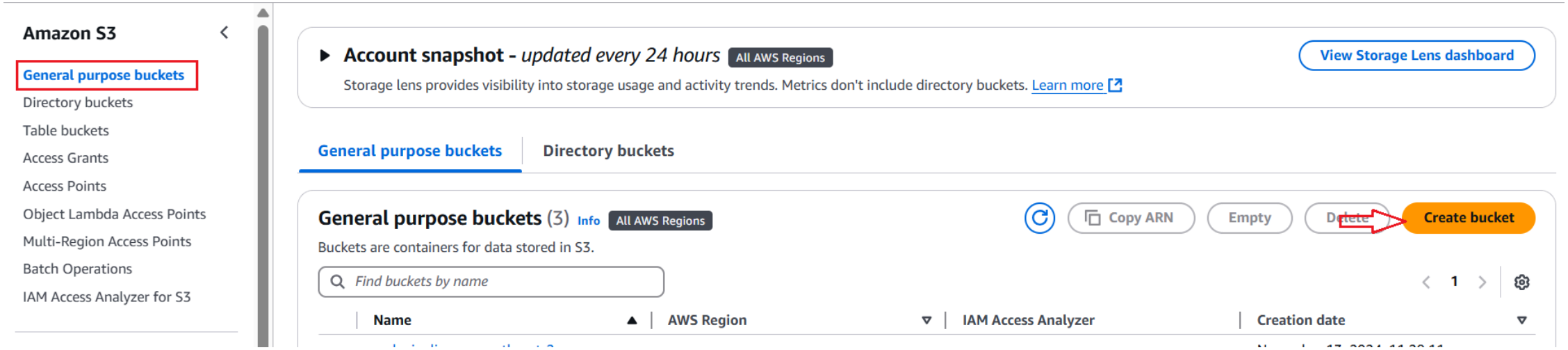
Amazon S3

- Amazon Simple Storage Service(S3)는 업계 최고 수준의 확장성, 데이터 가용성, 보안 및 성능을 제공하는 객체 스토리지 서비스입니다.
- 규모와 업종에 관계없이 수백만 명의 고객이 데이터 레이크, 클라우드 네이티브 애플리케이션, 모바일 앱 등 거의 모든 사용 사례에서 모든 양의 데이터를 저장, 관리, 분석, 보호하고 있습니다.

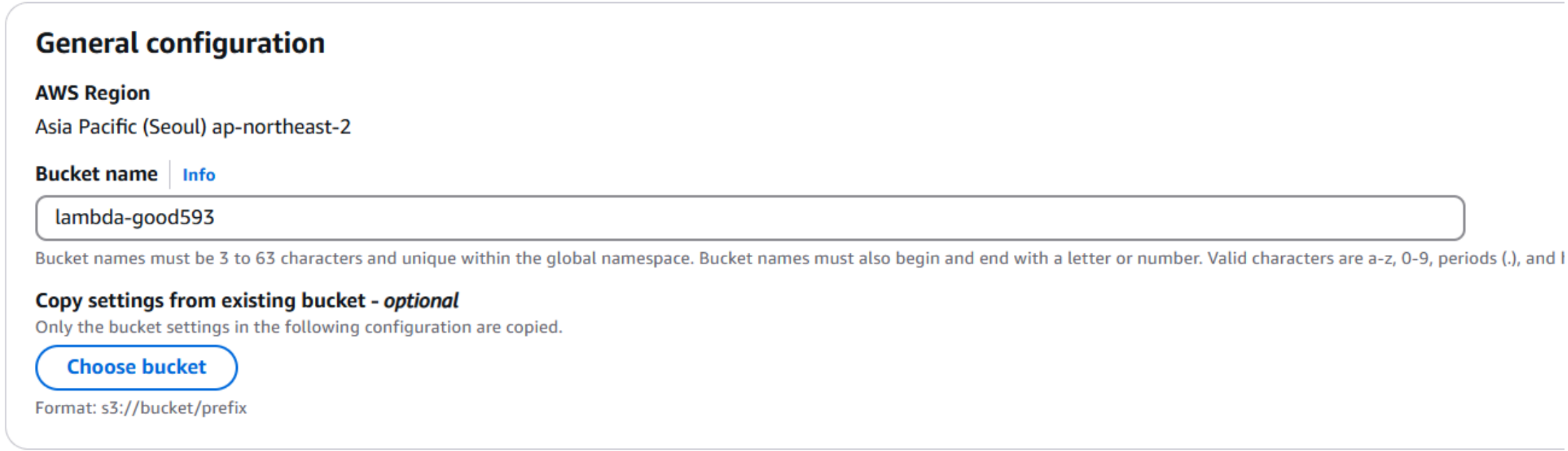
단계1: S3 접속


aws
S3
IAM
EC
Amazon
Amazon S
General purpo
Directory buc
Table buckets
Access Grants
Access Points
Object Lambo
Services
Features
Resources New
Documentation
Knowledge articles
Marketplace
Blog posts
Events
Tutorials
Services
Show more
S3
Scalable Storage in the Cloud
S3 Glacier
Archive Storage in the Cloud
AWS Snow Family
Large Scale Data Transport
earn more
Copy ARN

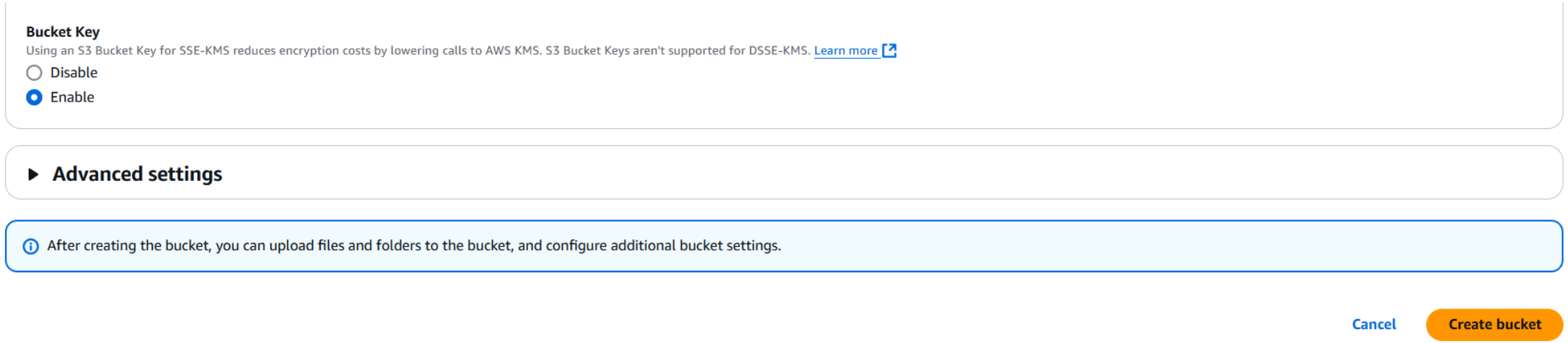
단계2: Create bucket



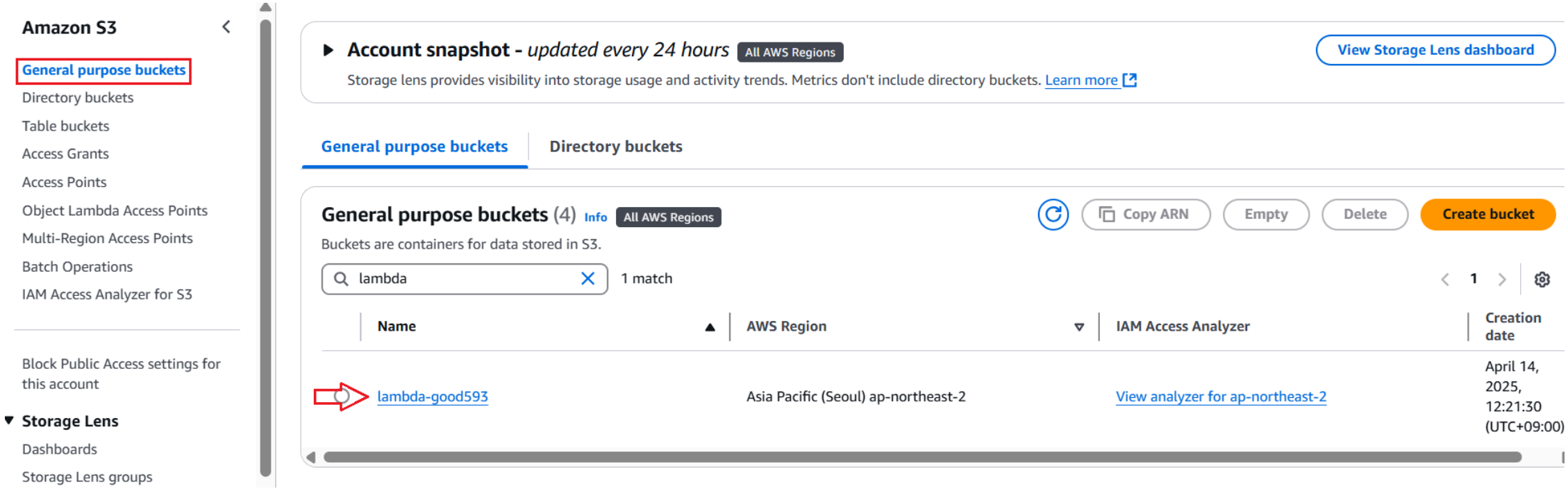
- Bucket name은 Global하게 유일해야함

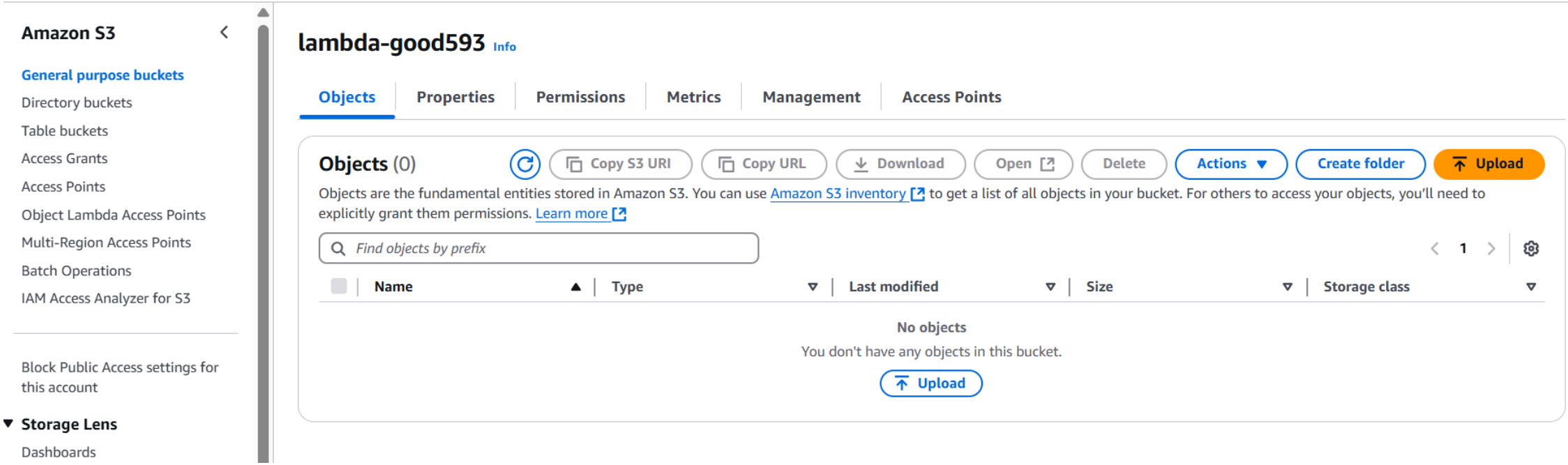


- Bucket 생성하기


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

- 생성 확인


Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
Account snapshot - updated every 24 hours
All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets (4) Info
All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
lambda
1 match
1
Name
AWS Region
IAM Access Analyzer
Creation date
lambda-good593
Asia Pacific (Seoul) ap-northeast-2
View analyzer for ap-northeast-2
April 14, 2025, 12:21:30 (UTC+09:00)


Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
lambda-good593 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (0)
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
Name
Type
Last modified
Size
Storage class
No objects
You don't have any objects in this bucket.
Upload

AWS SDK for Python(Boto3)

- Boto3를 사용하면 Python 애플리케이션, 라이브러리 또는 스크립트를 Amazon S3, Amazon EC2, Amazon DynamoDB 등 AWS 서비스와 쉽게 통합할 수 있습니다.

Boto3 API 문서

Boto3에는 뚜렷이 구별되는 두 가지 수준의 API가 있습니다.

- Client(또는"낮은 수준") API는 기본 HTTP API 작업에 일대일 매핑을 제공합니다.
- Resource(리소스) API 명시적인 네트워크 호출을 숨기지만 대신 속성에 액세스하고 작업을 수행하도록 리소스 객체 및 리소스 모음을 제공합니다.

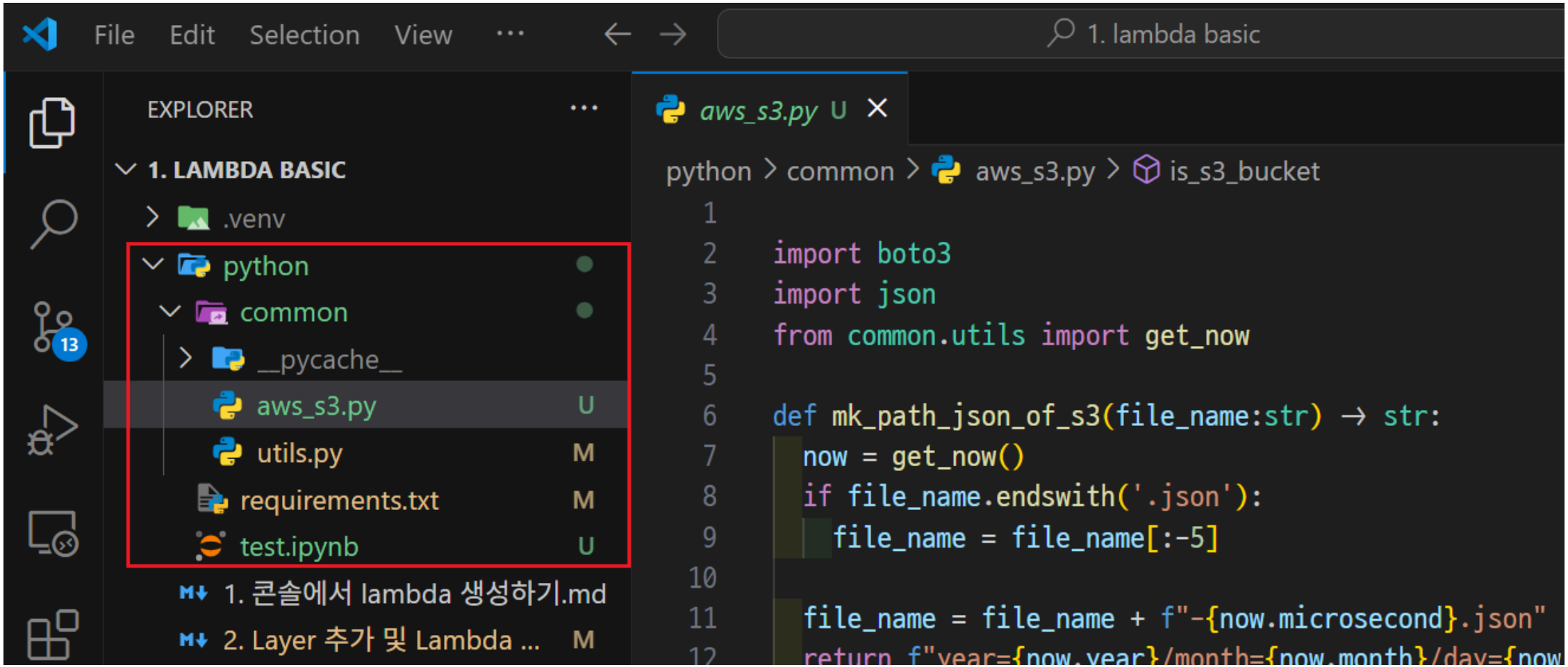
- Client 공식 문서

```
import boto3

client = boto3.client('s3')
```

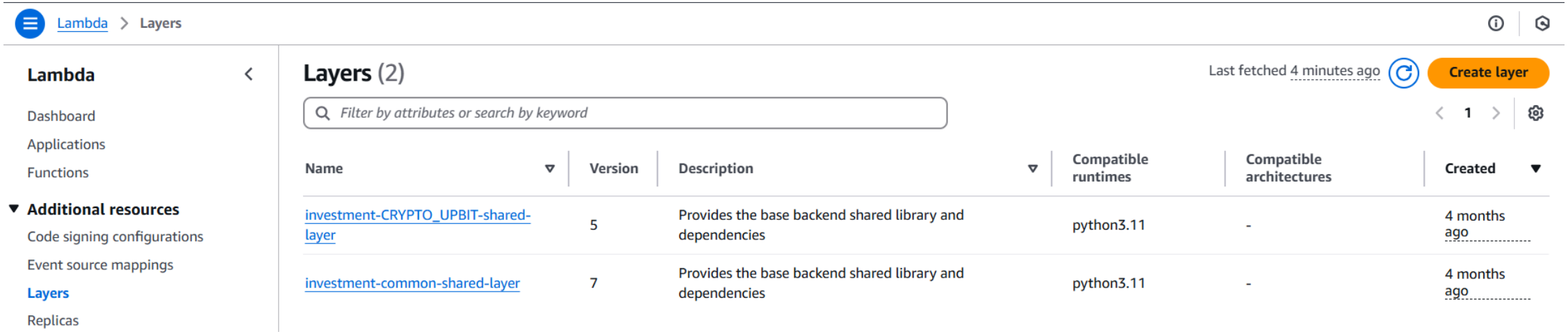
- Resource 공식 문서

```
import boto3

s3 = boto3.resource('s3')
```

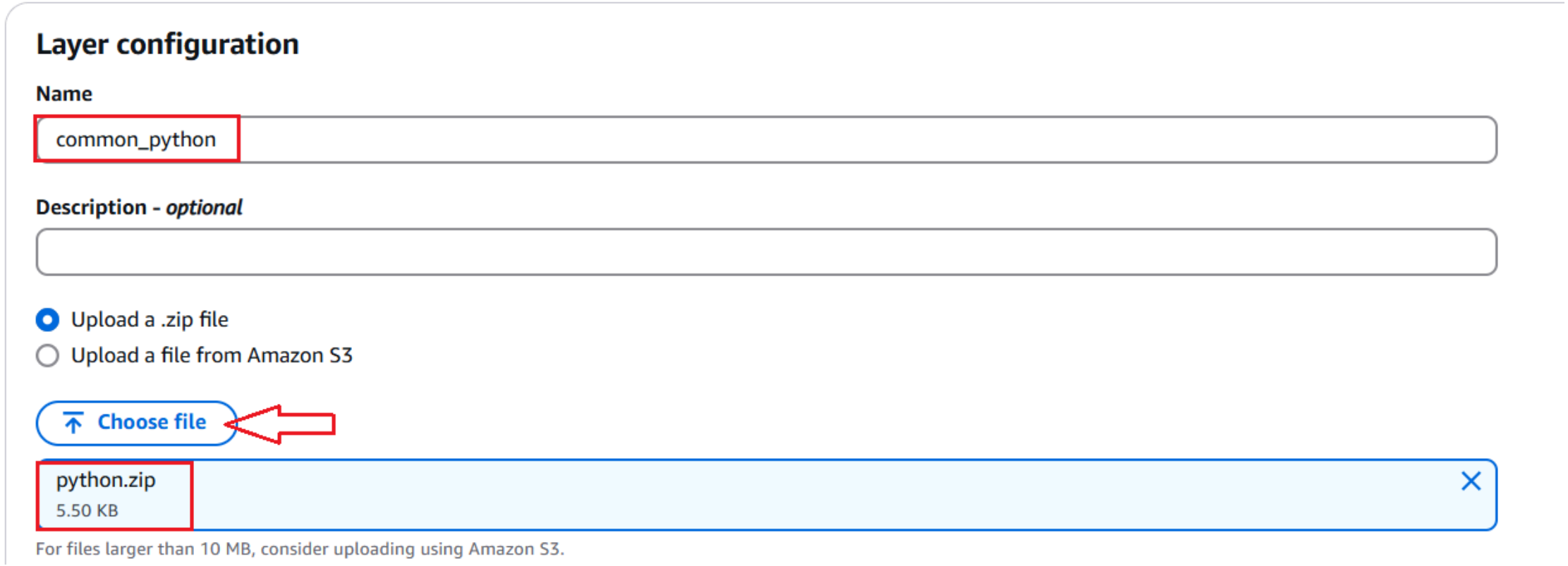
단계1: python 폴더 압축하기



단계2: AWS Lambda Layer 생성



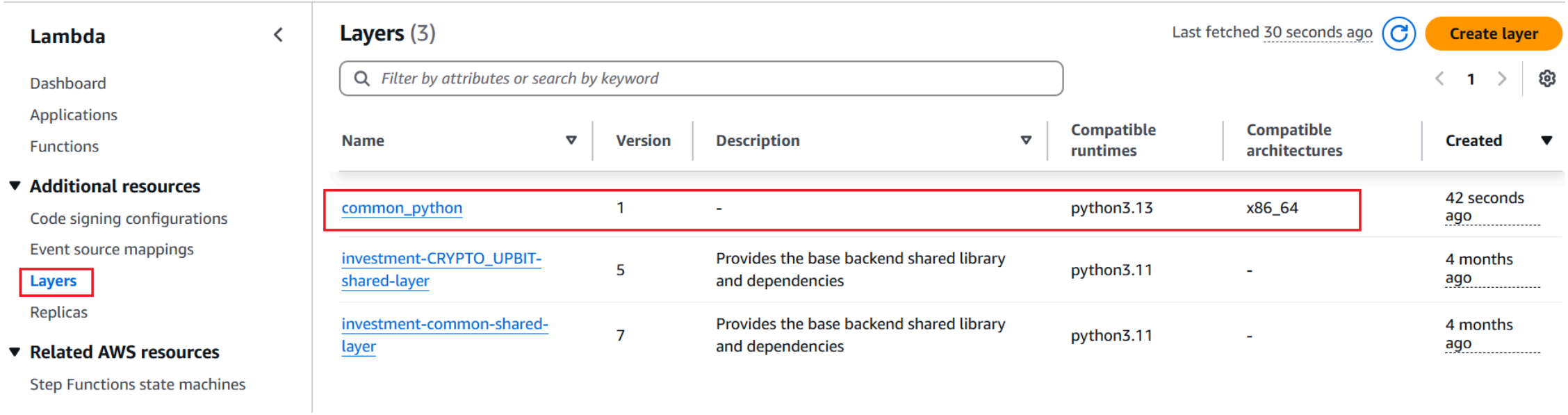
- 압축파일 업로드


Layer configuration
Name
common_python
Description - optional
Upload a .zip file
Upload a file from Amazon S3
Choose file
python.zip
5.50 KB
For files larger than 10 MB, consider uploading using Amazon S3.

- Create

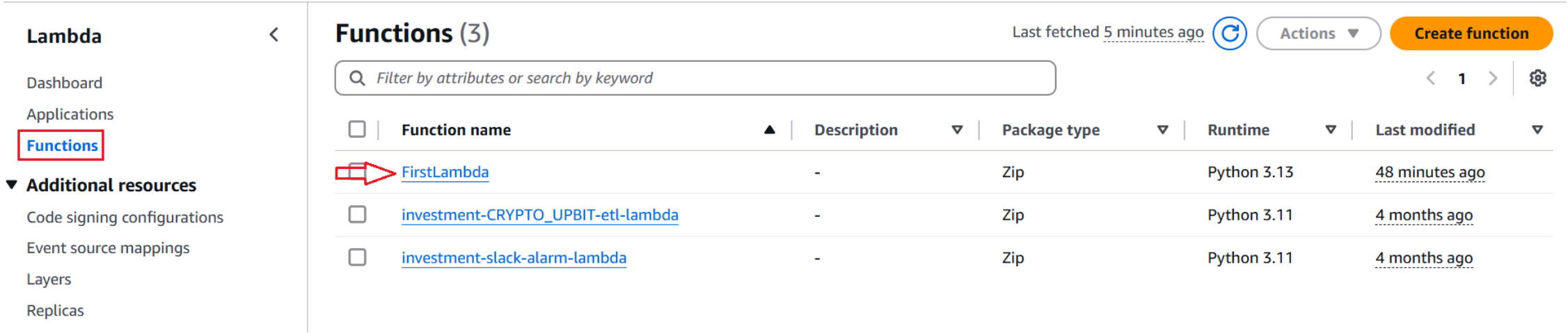

Compatible architectures - optional
Info
Choose the compatible instruction set architectures for your layer.
x86_64
Compatible runtimes - optional
Info
Choose up to 15 runtimes.
Python 3.13
License - optional
Info
Cancel
Create

- 결과 확인

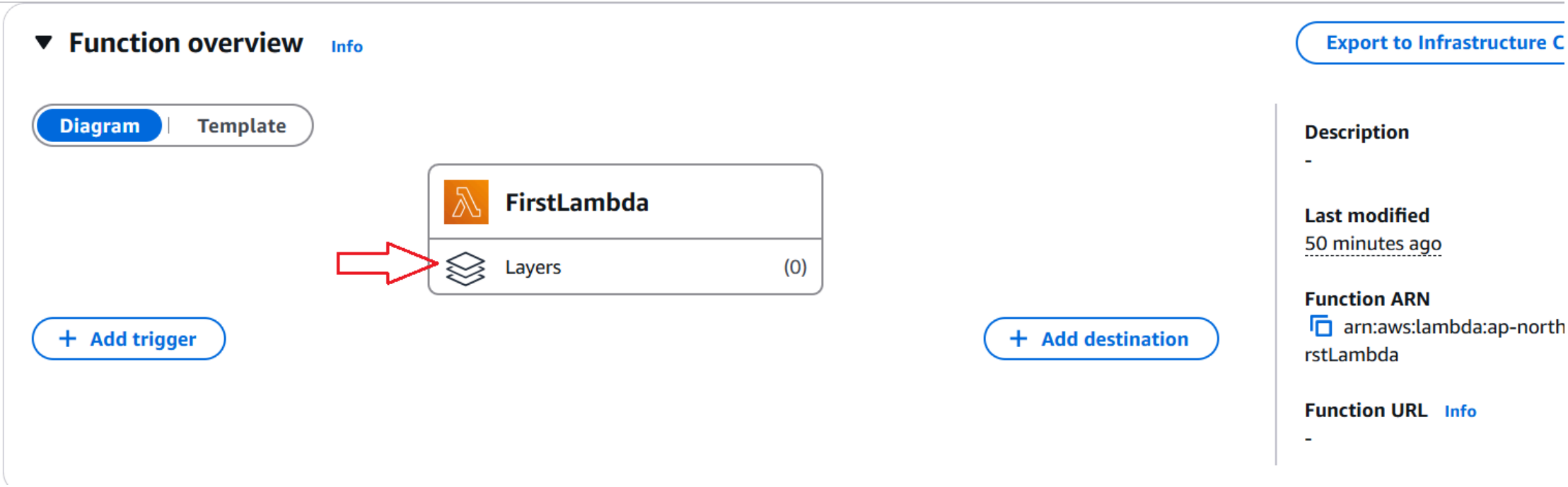

Lambda
Dashboard
Applications
Functions
Additional resources
Code signing configurations
Event source mappings
Layers
Replicas
Related AWS resources
Step Functions state machines
Layers (3)
Last fetched 30 seconds ago
Create layer
Filter by attributes or search by keyword
1
Name
Version
Description
Compatible runtimes
Compatible architectures
Created
common_python
1
-
python3.13
x86_64
42 seconds ago
investment-CRYPTO_UPBIT-shared-layer
5
Provides the base backend shared library and dependencies
python3.11
-
4 months ago
investment-common-shared-layer
7
Provides the base backend shared library and dependencies
python3.11
-
4 months ago

Lambda with Layer

단계1: Lambda 선택



단계2: Add Layder



▶ **Runtime management configuration**

Layers Info

Edit | Add a layer

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN

There is no data to display.

Layer source Info

Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also create a new layer.

- ○ AWS layers
  Choose a layer from a list of layers provided by AWS.
- ◉ Custom layers
  Choose a layer from a list of layers created by your AWS account.
- ○ Specify an ARN
  Specify a layer by providing the ARN.

Custom layers

Layers created by your AWS account that are compatible with your function's runtime.

common_python ▼

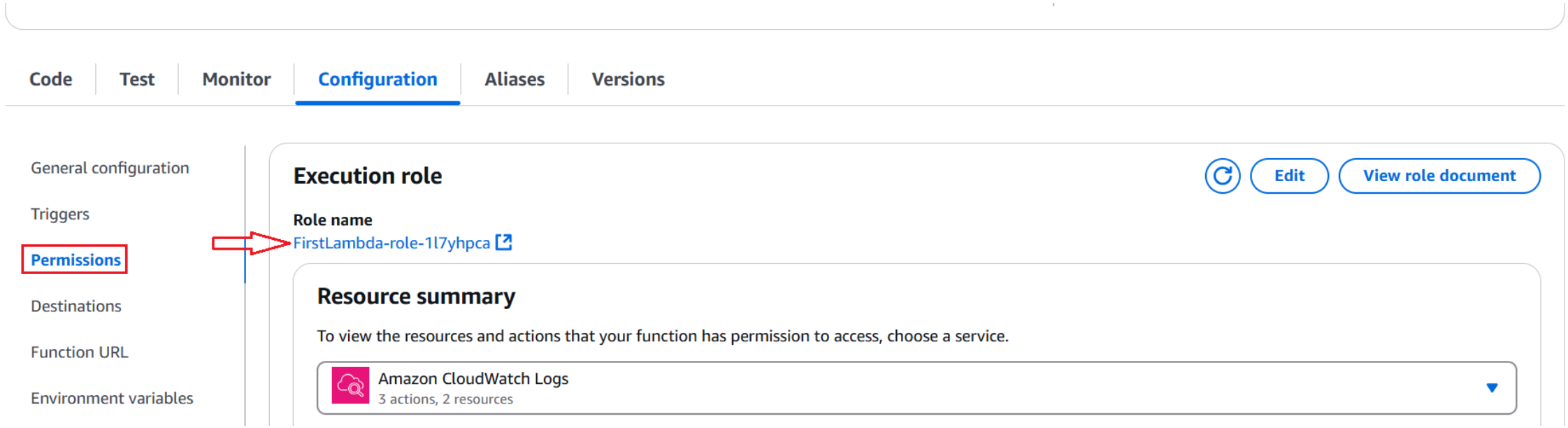
Version

1 ▼

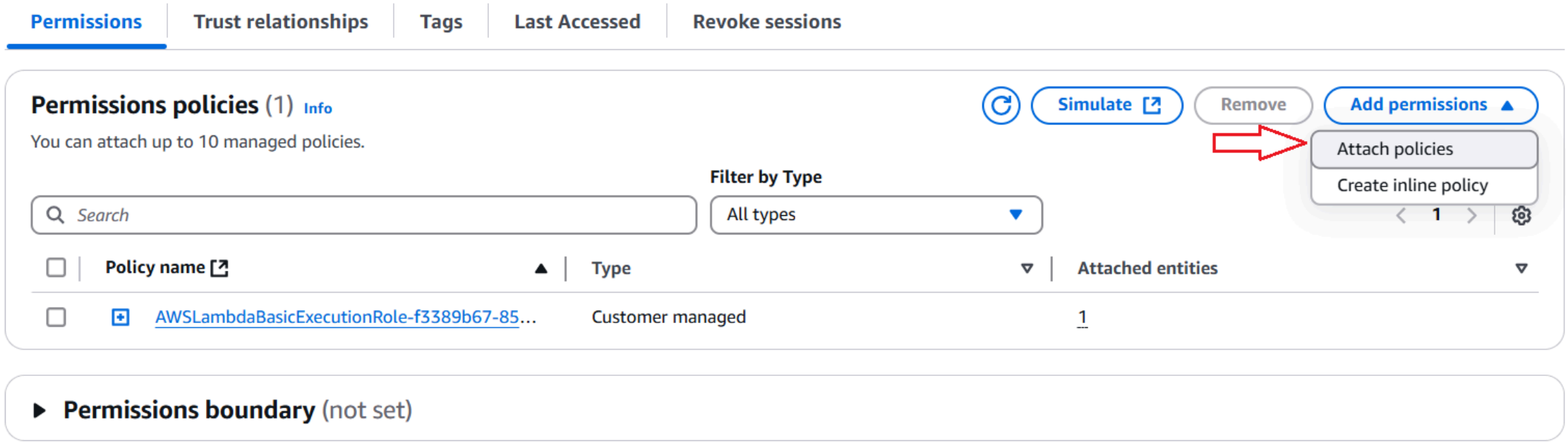
Cancel Add

Permissions 수정

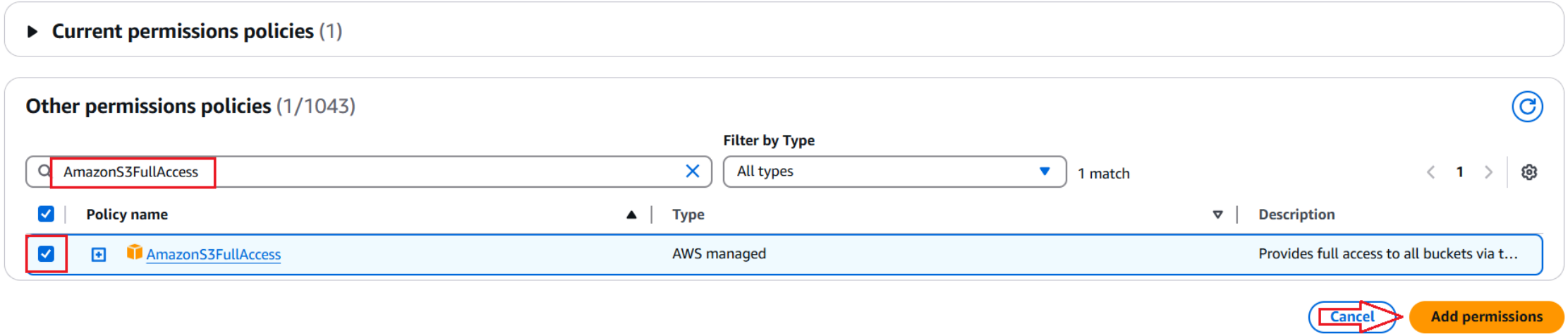
단계1: IAM Role 접속

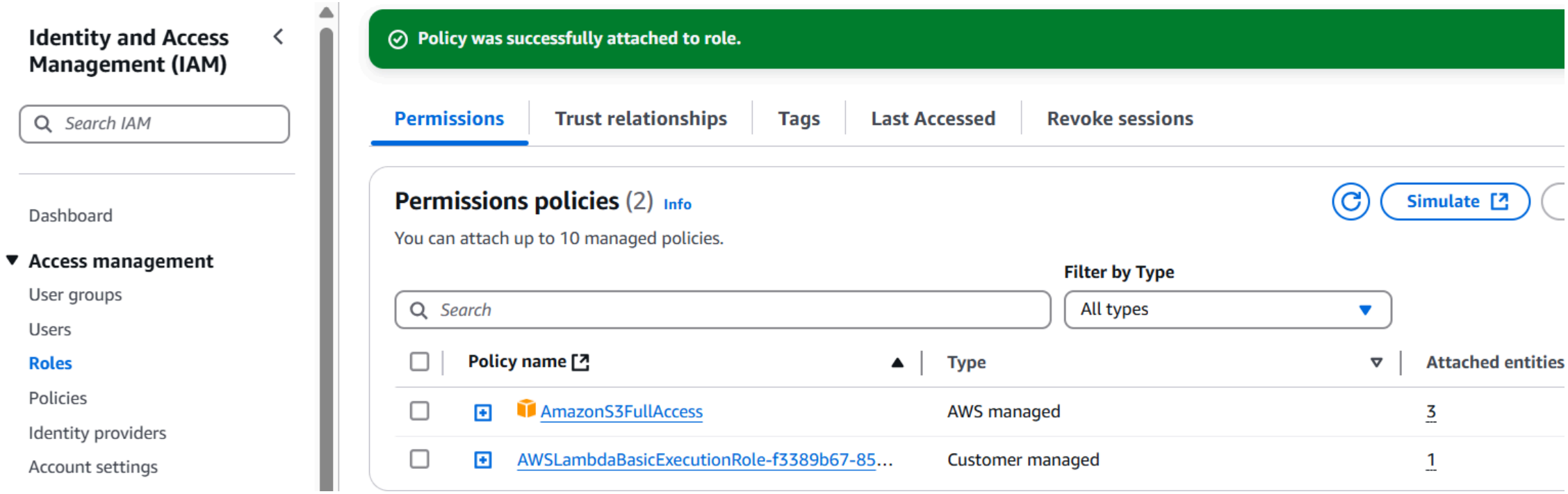


단계2: Attach policies



단계3: add AmazonS3FullAccess



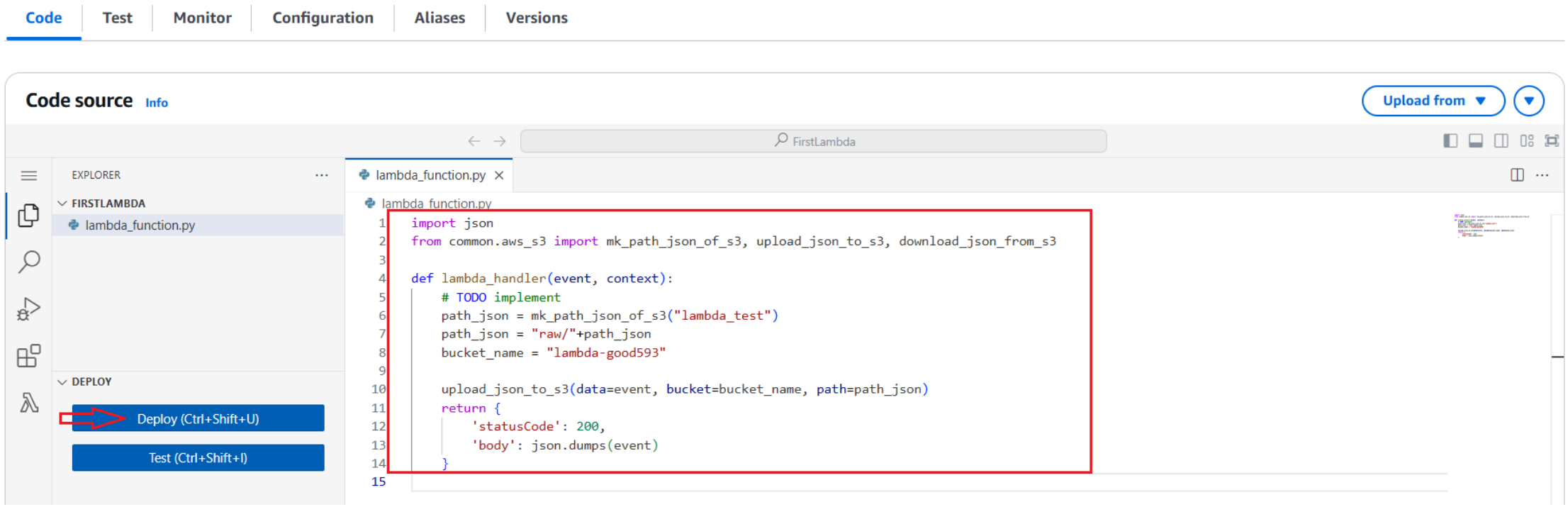

Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Policy was successfully attached to role.
Permissions
Trust relationships
Tags
Last Accessed
Revoke sessions
Permissions policies (2) Info
You can attach up to 10 managed policies.
Simulate
Filter by Type
Search
All types
Policy name
Type
Attached entities
AmazonS3FullAccess
AWS managed
3
AWSLambdaBasicExecutionRole-f3389b67-85...
Customer managed
1

테스트

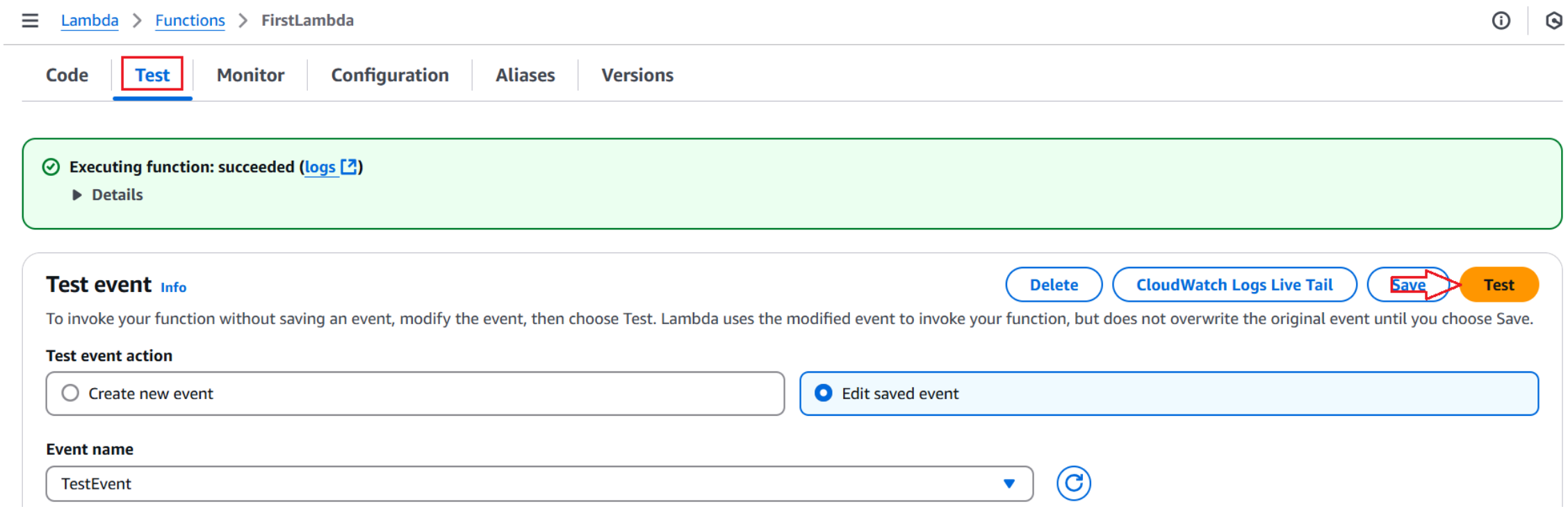
단계1: 코드 수정

```
import json
from common.aws_s3 import mk_path_json_of_s3, upload_json_to_s3, download_json_from_s3

def lambda_handler(event, context):
    # TODO implement
    path_json = mk_path_json_of_s3("lambda_test")
    path_json = "raw/"+path_json
    bucket_name = "생성한 bucket명"

    upload_json_to_s3(data=event, bucket=bucket_name, path=path_json)
    return {
        'statusCode': 200,
        'body': json.dumps(event)
    }
```

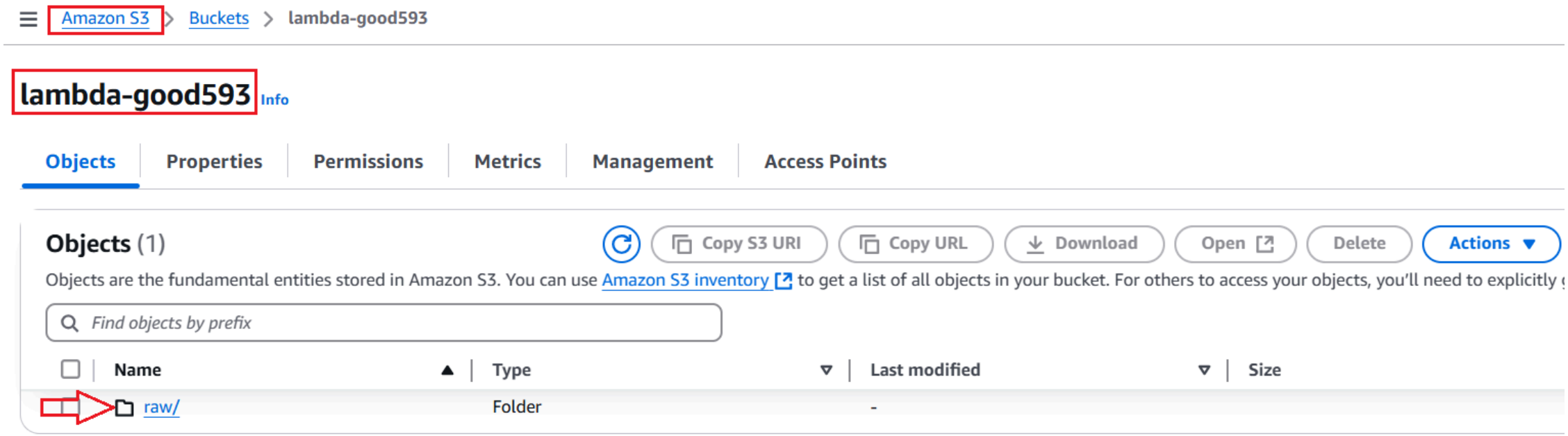
단계2: Deploy

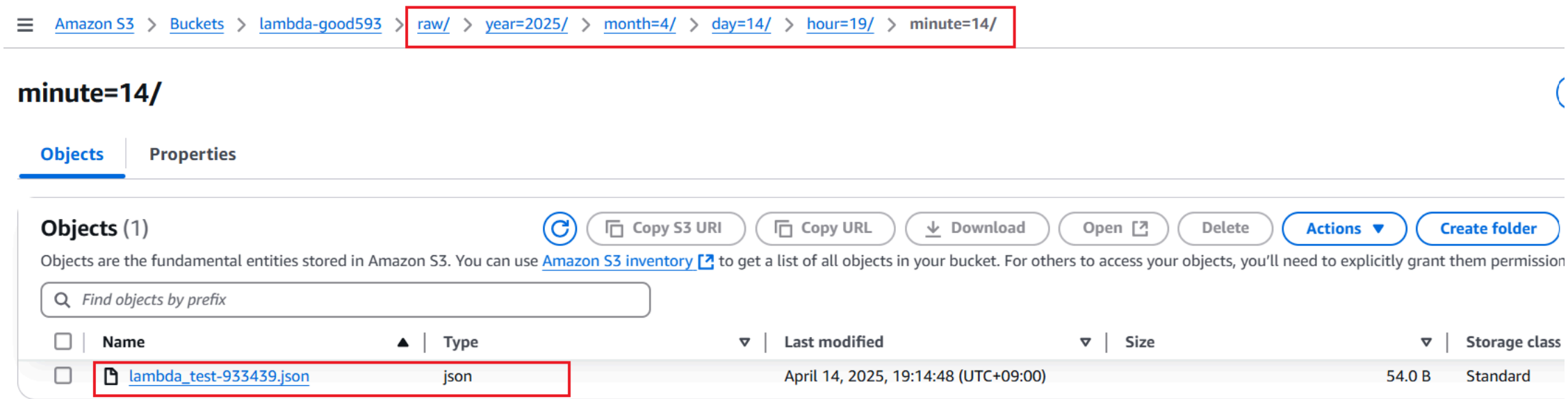

Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Upload from
FirstLambda
EXPLORER
FIRSTLAMBDA
lambda_function.py
DEPLOY
Deploy (Ctrl+Shift+U)
Test (Ctrl+Shift+I)
lambda_function.py
lambda_function.py
import json
from common.aws_s3 import mk_path_json_of_s3, upload_json_to_s3, download_json_from_s3
def lambda_handler(event, context):
# TODO implement
path_json = mk_path_json_of_s3("lambda_test")
path_json = "raw/"+path_json
bucket_name = "lambda-good593"
upload_json_to_s3(data=event, bucket=bucket_name, path=path_json)
return {
'statusCode': 200,
'body': json.dumps(event)
}

단계3: Test


Lambda > Functions > FirstLambda
Code
Test
Monitor
Configuration
Aliases
Versions
Executing function: succeeded (logs)
Details
Test event Info
Delete
CloudWatch Logs Live Tail
Save
Test
To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save.
Test event action
Create new event
Edit saved event
Event name
TestEvent

단계4: 파일 확인 in S3




Amazon S3 > Buckets > lambda-good593 > raw/ > year=2025/ > month=4/ > day=14/ > hour=19/ > minute=14/
minute=14/
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
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 permission
Find objects by prefix
Name
Type
Last modified
Size
Storage class
lambda_test-933439.json
json
April 14, 2025, 19:14:48 (UTC+09:00)
54.0 B
Standard