Chair for Decentralized Information Systems and Data Management
TUM School of Computation, Information and Technology
Technical University of Munich

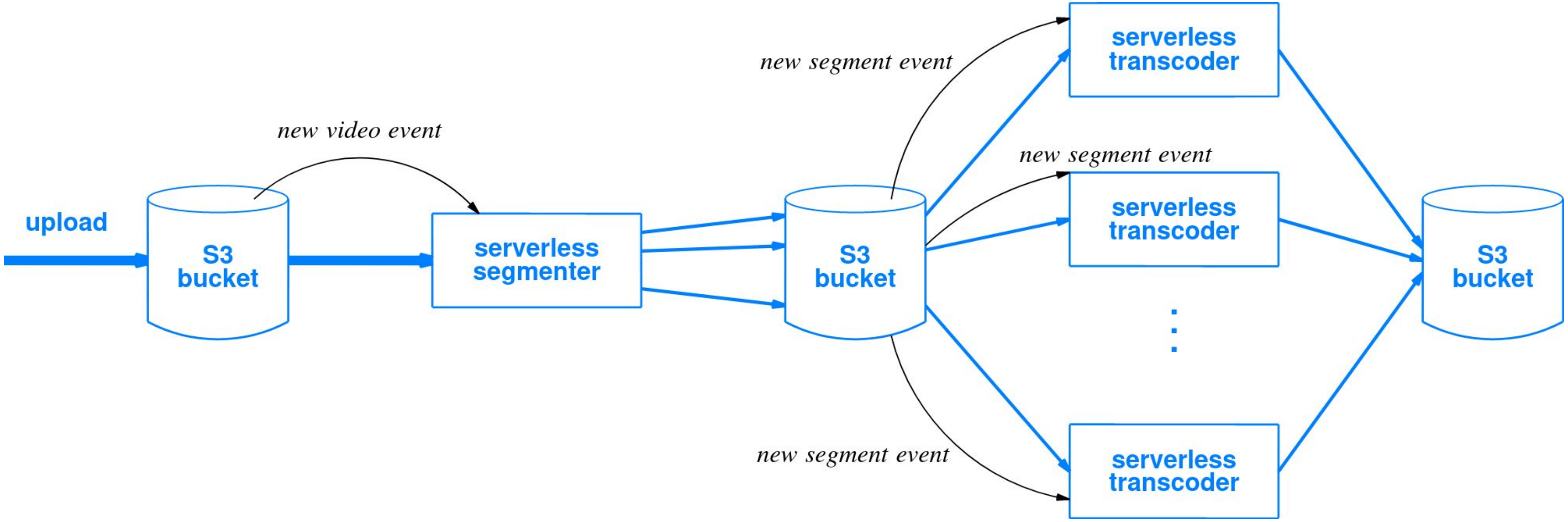
Cloud Information Systems

Exercise 10

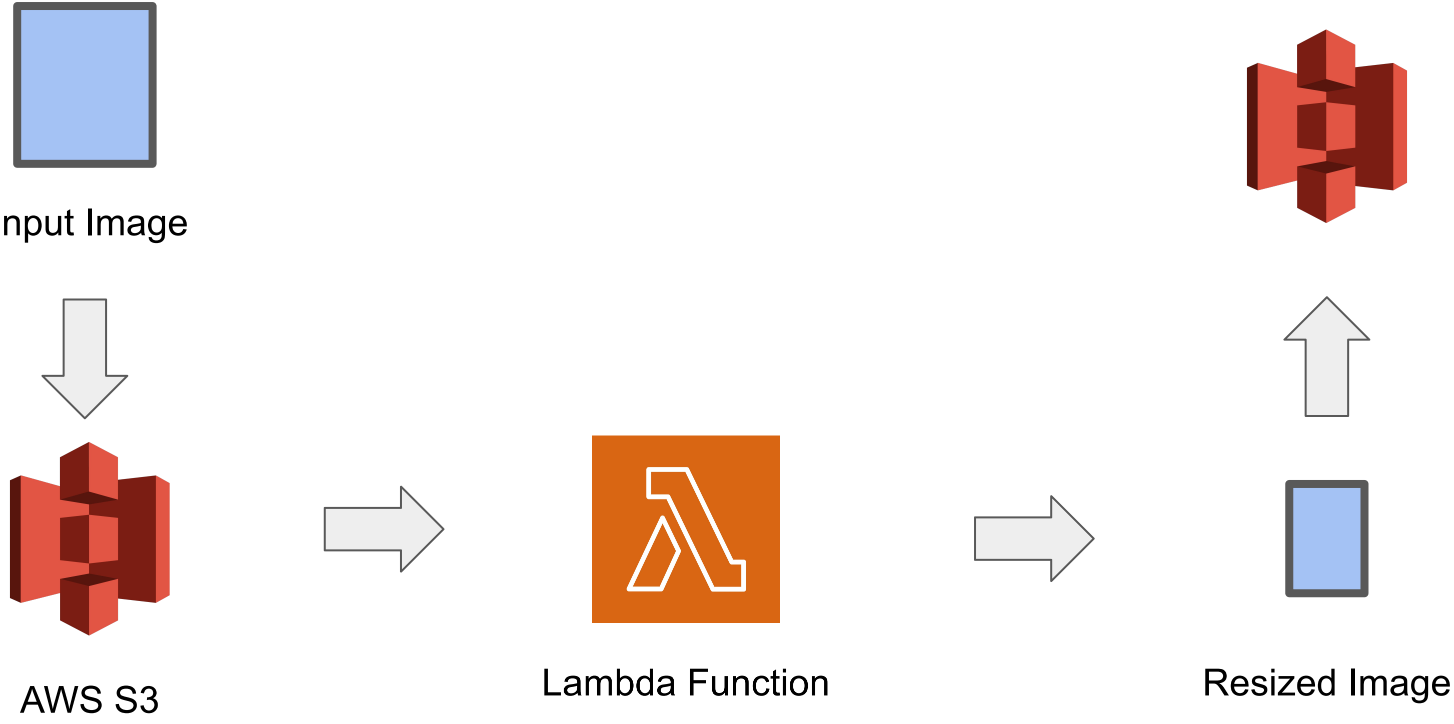
23rd December 2024

Prof. Dr. Viktor Leis, M.Sc. Till Steinert, M.Sc. Jana Vatter

3. Netflix Video Transcoding

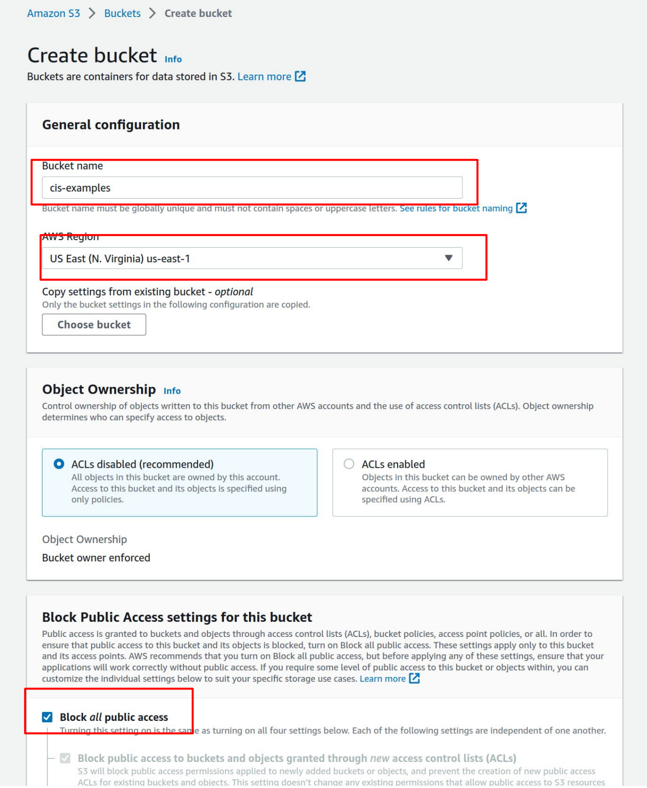


3. Live Demo



Backup

Demo: Creating a new S3 Bucket



Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name: cis-examples

Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region: US East (N. Virginia) us-east-1 ▼

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership: Bucket owner enforced

Block Public Access settings for this bucket

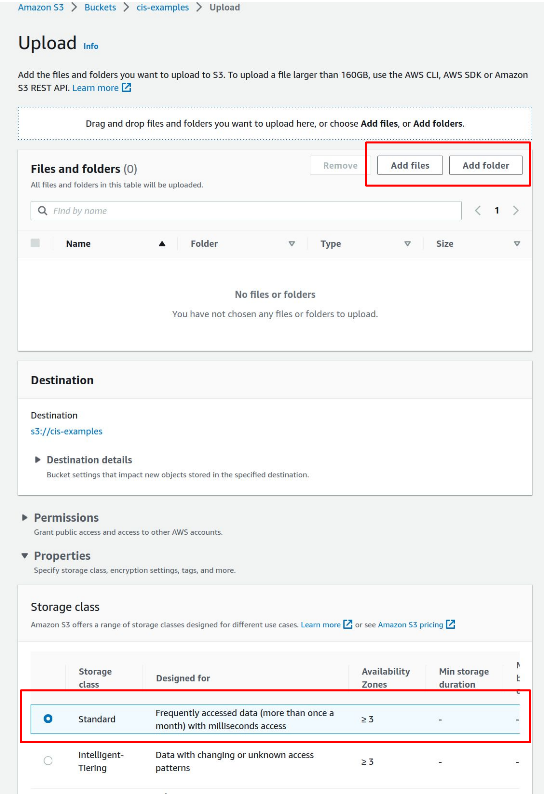
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access 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

- Enter a Name for your Bucket
- Specify your Region
- Make sure to block public Access to your Bucket

Demo: Uploading Files



Amazon S3 > Buckets > cis-examples > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (0)

All files and folders in this table will be uploaded.

□	Name	Folder	Type	Size

No files or folders

You have not chosen any files or folders to upload.

Destination

Destination
s3://cis-examples

▸ Destination details
Bucket settings that impact new objects stored in the specified destination.

▸ Permissions
Grant public access and access to other AWS accounts.

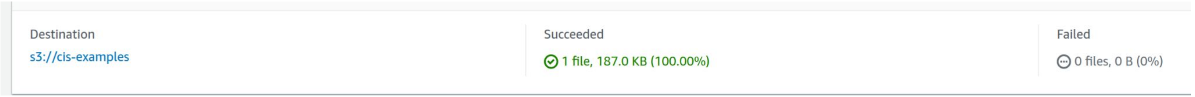
▾ Properties
Specify storage class, encryption settings, tags, and more.

Storage class

Amazon S3 offers a range of storage classes designed for different use cases. Learn more or see Amazon S3 pricing

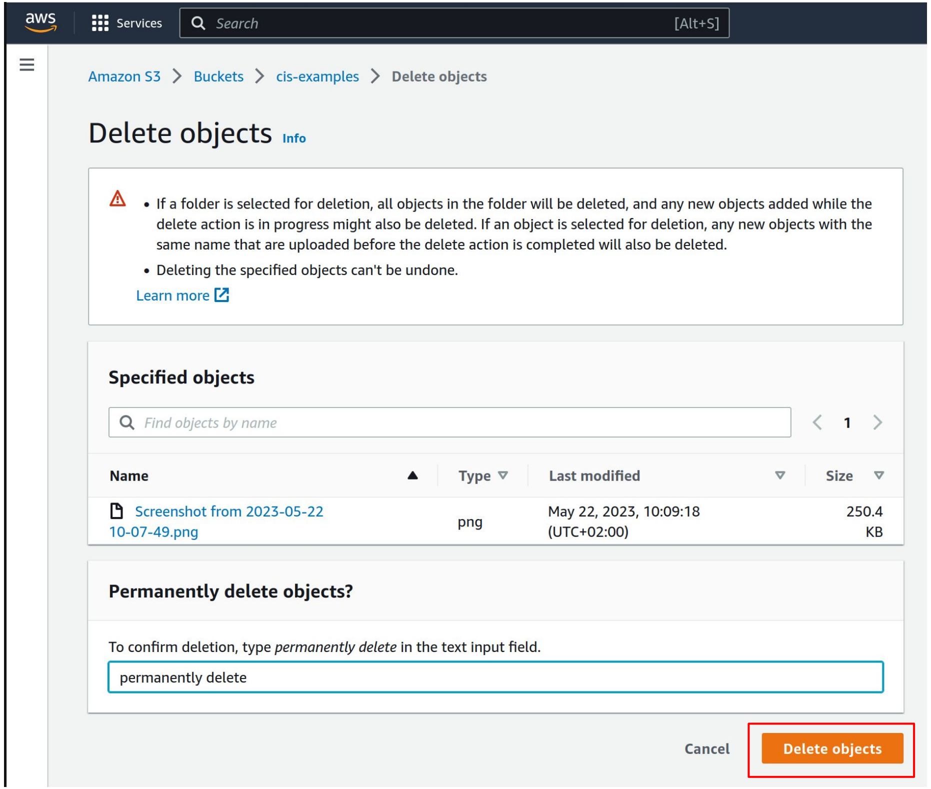
	Storage class	Designed for	Availability Zones	Min storage duration
●	Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-
○	Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-

- You can upload Files either through the “Add Files” Button or via Drag-and-Drop
- For this Exercise, we will be using the “Standard” Storage Class
- After Uploading a File, you should receive a notification such as this:



Destination	Succeeded	Failed
s3://cis-examples	✓ 1 file, 187.0 KB (100.00%)	⊖ 0 files, 0 B (0%)

Demo: Deleting Files



Amazon S3 > Buckets > cis-examples > Delete objects

Delete objects Info

- If a folder is selected for deletion, all objects in the folder will be deleted, and any new objects added while the delete action is in progress might also be deleted. If an object is selected for deletion, any new objects with the same name that are uploaded before the delete action is completed will also be deleted.
- Deleting the specified objects can't be undone.

Learn more

Specified objects

Find objects by name

Name	Type	Last modified	Size
Screenshot from 2023-05-22 10-07-49.png	png	May 22, 2023, 10:09:18 (UTC+02:00)	250.4 KB

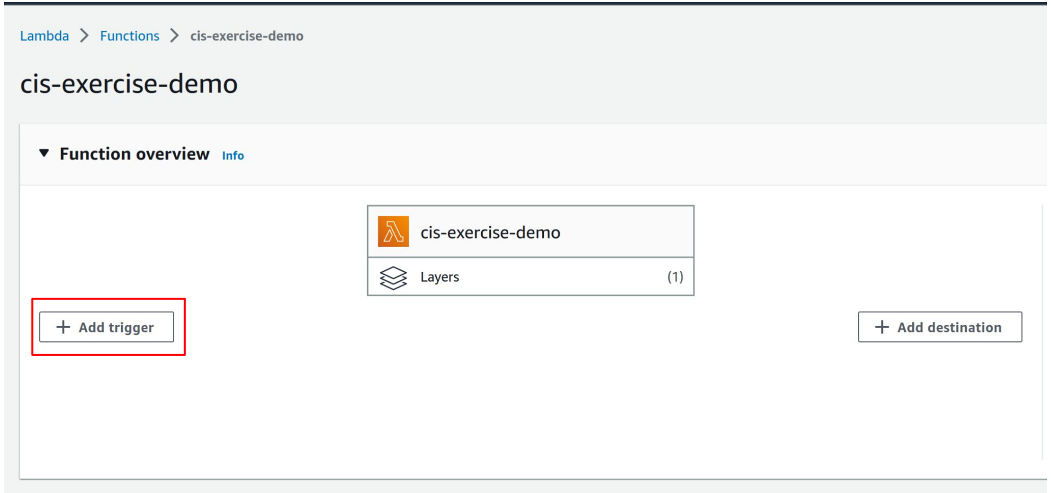
Permanently delete objects?

To confirm deletion, type *permanently delete* in the text input field.

permanently delete

- Select Object and click "Delete"
- Type "permanently delete"

Demo: Add Trigger to Lambda Function



Lambda > Functions > cis-exercise-demo

cis-exercise-demo

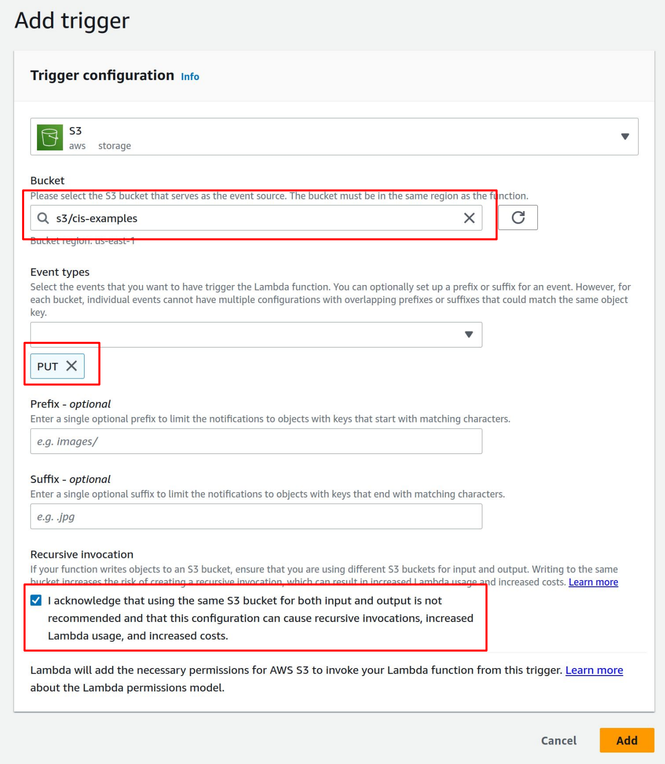
▼ Function overview Info

cis-exercise-demo

Layers (1)

- Navigate to last weeks Lambda Function
- Add Trigger and search for S3

Demo: Add Trigger to Lambda Function



Add trigger

Trigger configuration Info

S3
aws storage

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

s3/cis-examples

Bucket region: us-east-1

Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

PUT ✕

Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

e.g. images/

Suffix - *optional*
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

e.g. .jpg

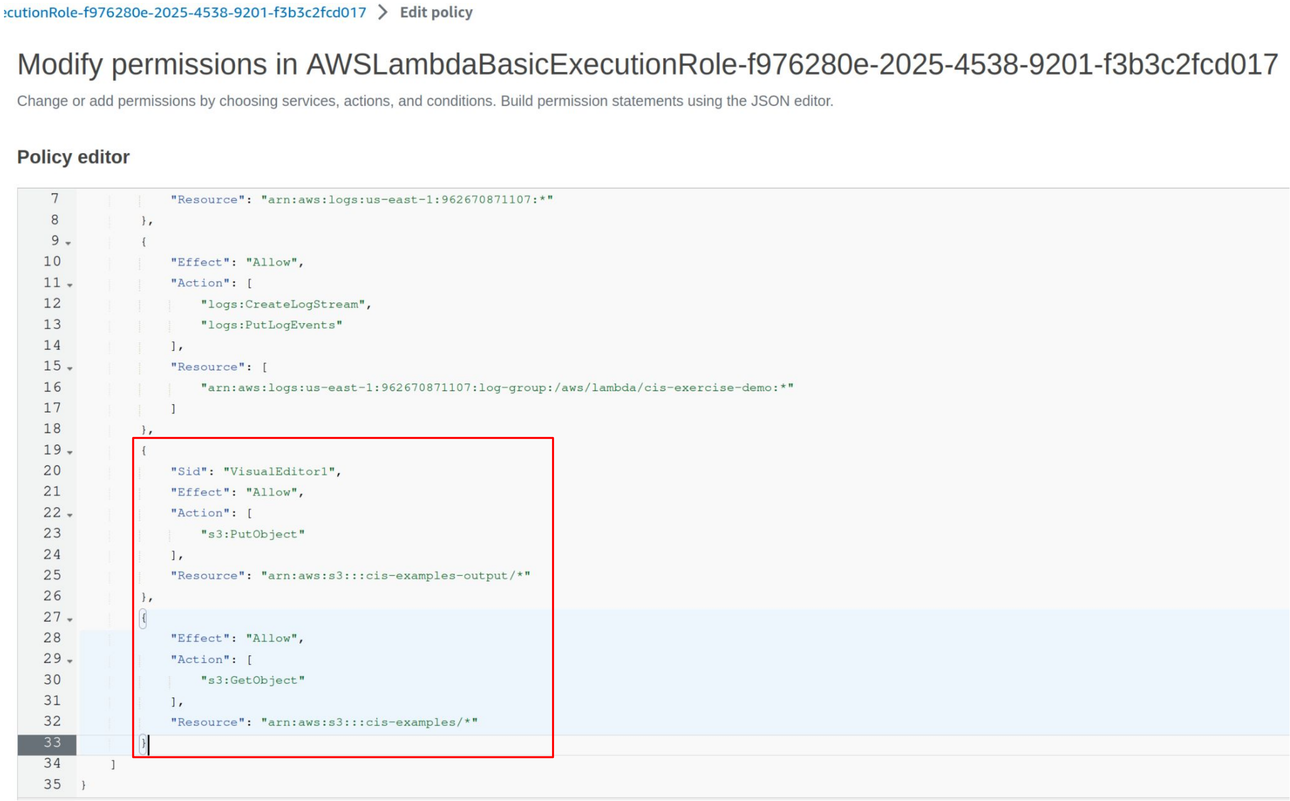
Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more

☑ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.

- Select the Bucket you just created
- Restrict Event Type to PUT
- **Important:** Make sure to write the resulting Images to a ***<u>different</u>*** S3 Bucket (otherwise you might trigger an infinite recursion)

Demo: Adding Permissions to Lambda



ecutionRole-f976280e-2025-4538-9201-f3b3c2fcd017 > Edit policy

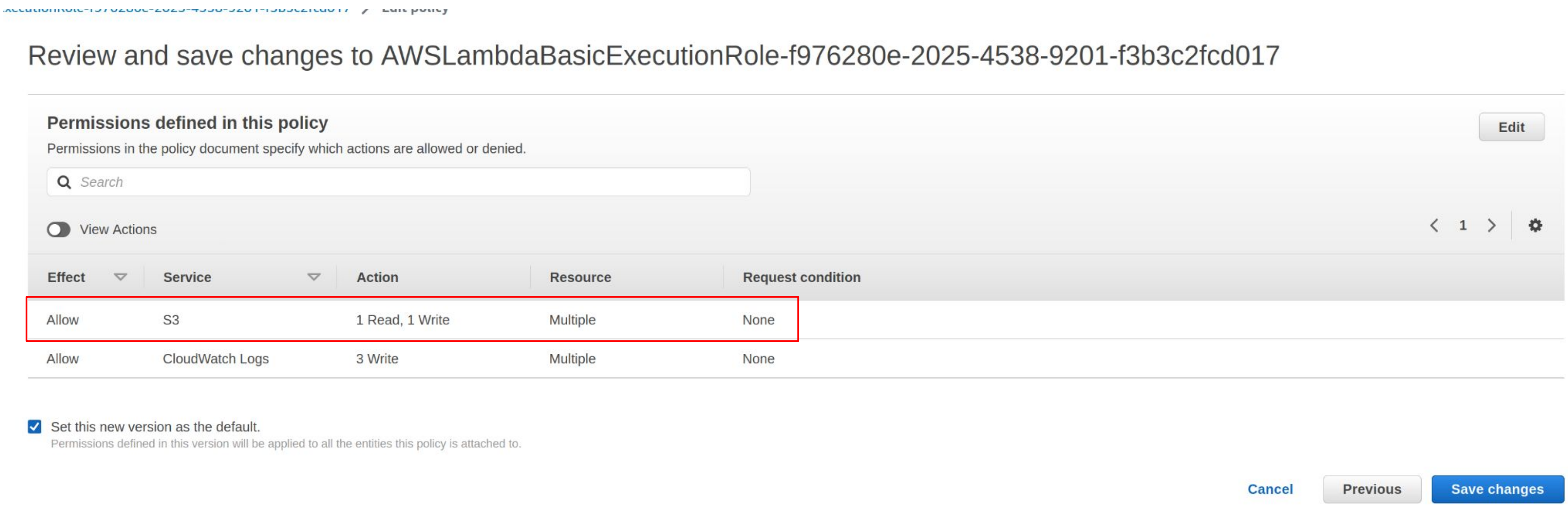
Modify permissions in AWSLambdaBasicExecutionRole-f976280e-2025-4538-9201-f3b3c2fcd017

Change or add permissions by choosing services, actions, and conditions. Build permission statements using the JSON editor.

Policy editor

```
            "Resource": "arn:aws:logs:us-east-1:962670871107:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:962670871107:log-group:/aws/lambda/cis-exercise-demo:*"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::cis-examples-output/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::cis-examples/*"
        }
    ]
}
```

Demo: Adding Permissions to Lambda



xecutionRole-f976280e-2025-4538-9201-f3b3c2fcd017 > Edit policy

Review and save changes to AWSLambdaBasicExecutionRole-f976280e-2025-4538-9201-f3b3c2fcd017

Permissions defined in this policy

Permissions in the policy document specify which actions are allowed or denied.

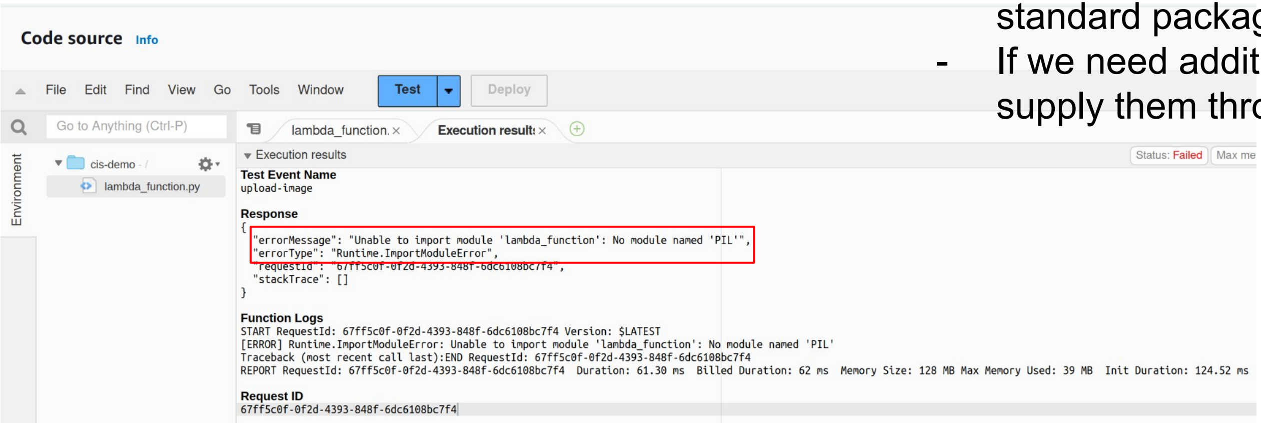
Effect	Service	Action	Resource	Request condition
Allow	S3	1 Read, 1 Write	Multiple	None
Allow	CloudWatch Logs	3 Write	Multiple	None

☑ Set this new version as the default.

Permissions defined in this version will be applied to all the entities this policy is attached to.

3. Live Demo: AWS Lambda

- Lambda natively only supports Python’s standard packages
- If we need additional packages, we must supply them through a Layer



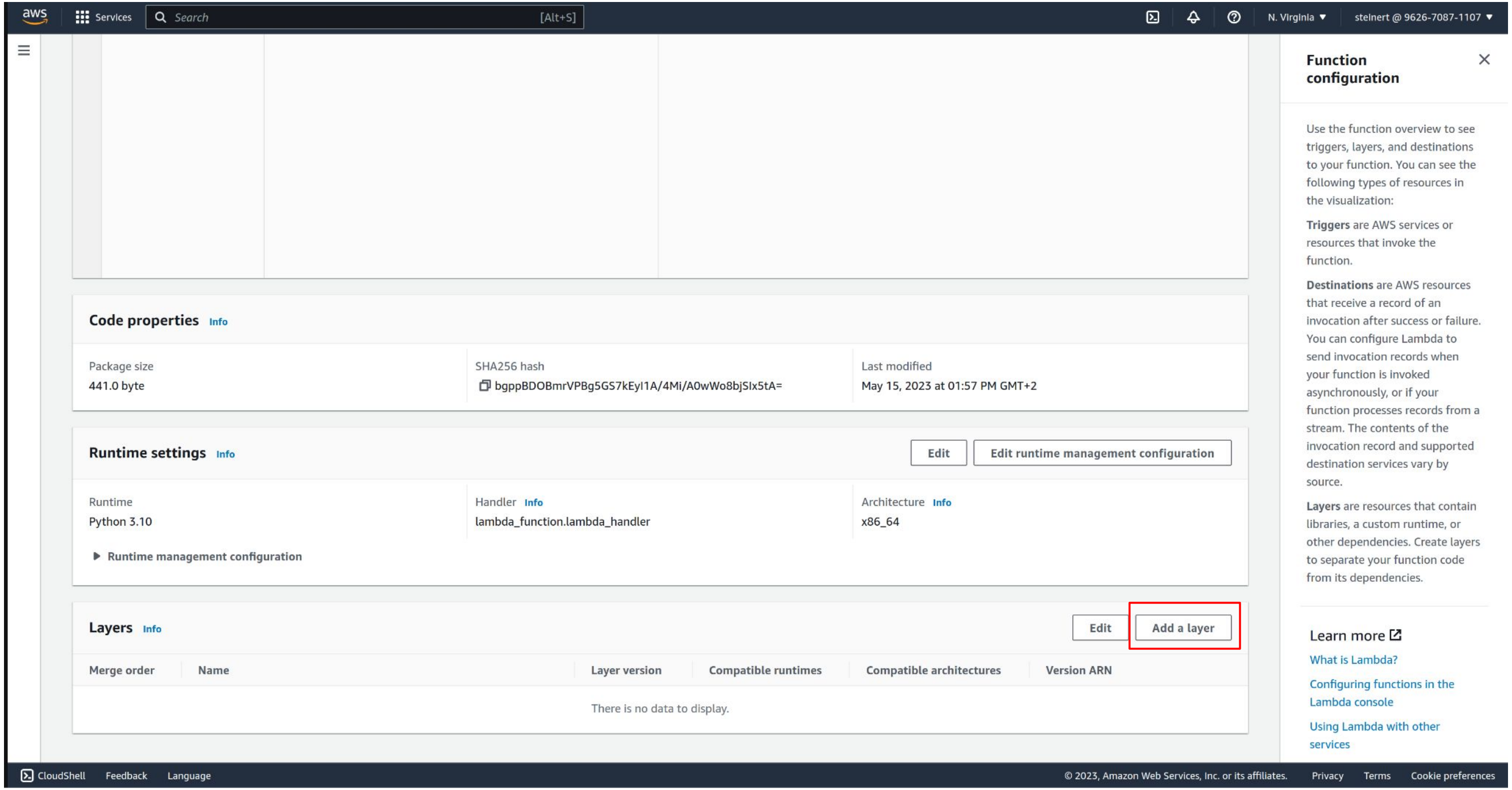
Test Event Name
```
upload-image
```

Response
```
{
  "errorMessage": "Unable to import module 'lambda_function': No module named 'PIL'",
  "errorType": "Runtime.ImportModuleError",
  "requestId": "67ff5c0f-0f2d-4393-848f-6dc6108bc7f4",
  "stackTrace": []
}
```

Function Logs
```
START RequestId: 67ff5c0f-0f2d-4393-848f-6dc6108bc7f4 Version: $LATEST
[ERROR] Runtime.ImportModuleError: Unable to import module 'lambda_function': No module named 'PIL'
Traceback (most recent call last):END RequestId: 67ff5c0f-0f2d-4393-848f-6dc6108bc7f4
REPORT RequestId: 67ff5c0f-0f2d-4393-848f-6dc6108bc7f4  Duration: 61.30 ms  Billed Duration: 62 ms  Memory Size: 128 MB Max Memory Used: 39 MB  Init Duration: 124.52 ms
```

Request ID
```
67ff5c0f-0f2d-4393-848f-6dc6108bc7f4
```

Demo: Third Party Libraries


aws
Services
Search
[Alt+S]
N. Virginia ▼
steinert @ 9626-7087-1107 ▼

Code properties Info

Package size	SHA256 hash	Last modified
441.0 byte	bgppBDOBmrVPBg5GS7kEyl1A/4Mi/A0wWo8bjSlx5tA=	May 15, 2023 at 01:57 PM GMT+2

Runtime settings Info

Edit
Edit runtime management configuration

Runtime	Handler Info	Architecture Info
Python 3.10	lambda_function.lambda_handler	x86_64

▶ Runtime management configuration

Layers Info

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN

There is no data to display.

Function configuration

Use the function overview to see triggers, layers, and destinations to your function. You can see the following types of resources in the visualization:

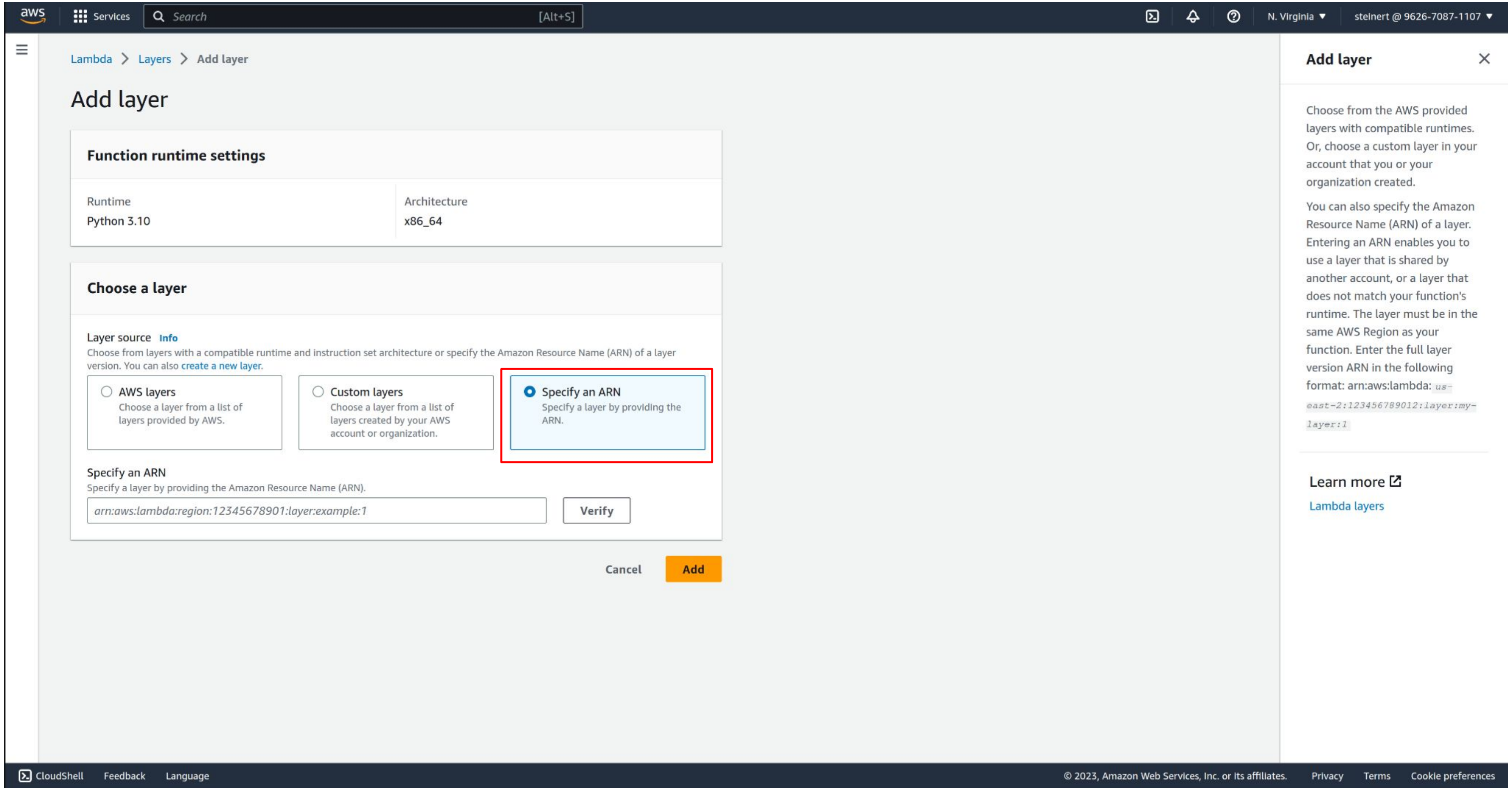
Triggers are AWS services or resources that invoke the function.

Destinations are AWS resources that receive a record of an invocation after success or failure. You can configure Lambda to send invocation records when your function is invoked asynchronously, or if your function processes records from a stream. The contents of the invocation record and supported destination services vary by source.

Layers are resources that contain libraries, a custom runtime, or other dependencies. Create layers to separate your function code from its dependencies.

Learn more
What is Lambda?
Configuring functions in the Lambda console
Using Lambda with other services

Demo: Third Party Libraries



aws
Services
Search
[Alt+S]
N. Virginia ▼
steinert @ 9626-7087-1107 ▼
Lambda > Layers > Add layer
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Add layer

Function runtime settings

Runtime: Python 3.10

Architecture: x86_64

Choose a layer

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 or organization.

● Specify an ARN
Specify a layer by providing the ARN.

Specify an ARN
Specify a layer by providing the Amazon Resource Name (ARN).

arn:aws:lambda:region:12345678901:layer:example:1

Add layer

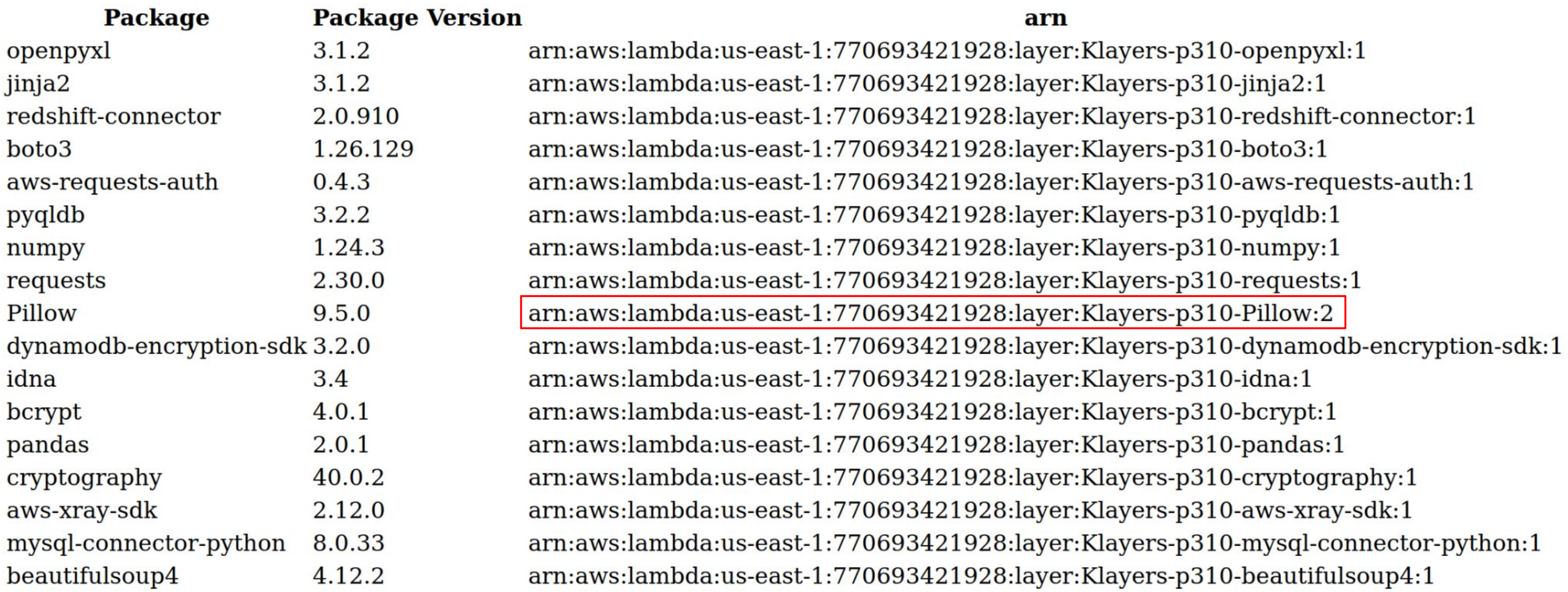
Choose from the AWS provided layers with compatible runtimes. Or, choose a custom layer in your account that you or your organization created.

You can also specify the Amazon Resource Name (ARN) of a layer. Entering an ARN enables you to use a layer that is shared by another account, or a layer that does not match your function's runtime. The layer must be in the same AWS Region as your function. Enter the full layer version ARN in the following format: arn:aws:lambda: `us-east-2:123456789012:layer:my-layer:1`

Learn more

Lambda layers

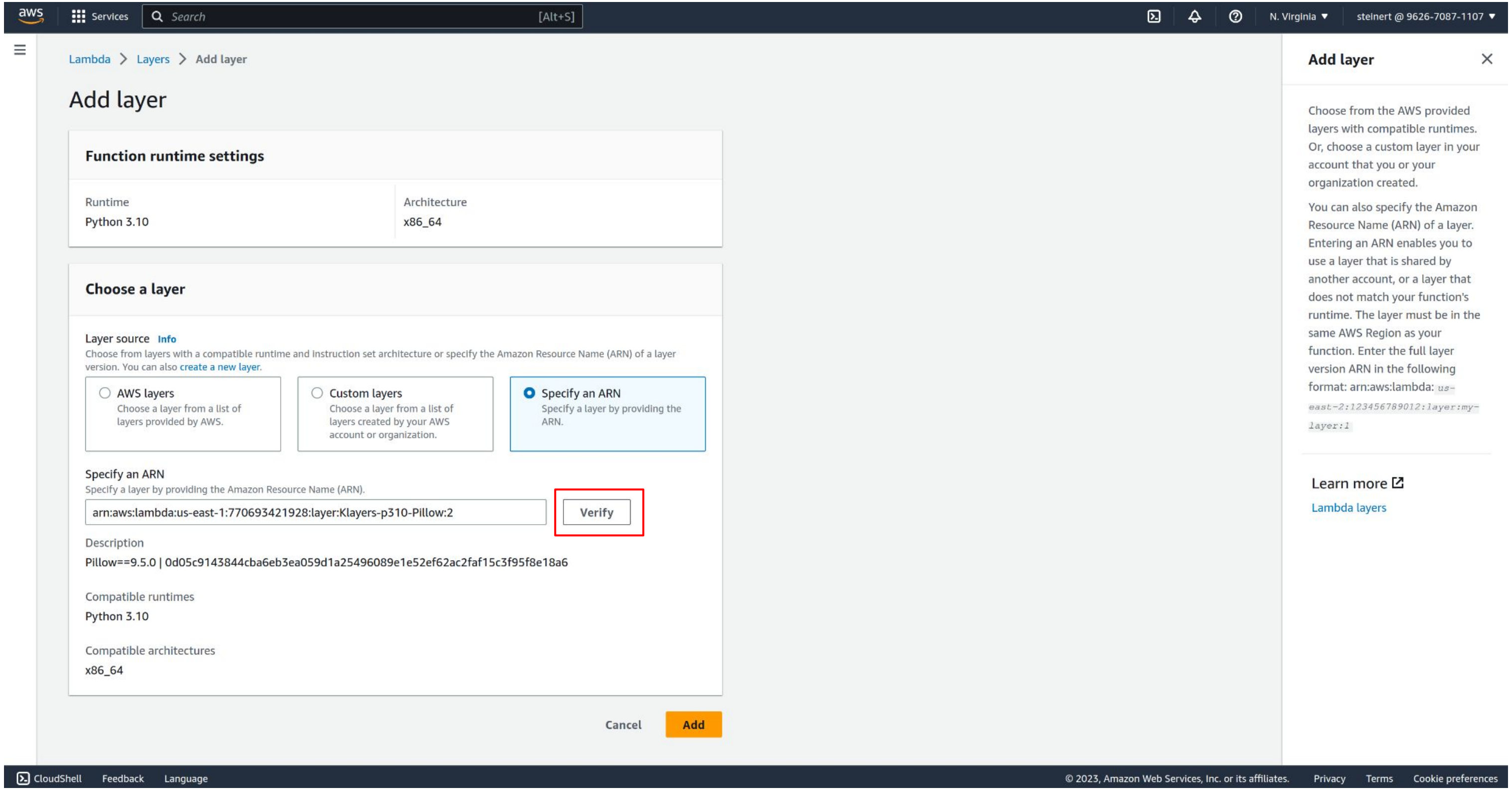
Demo: Third Party Libraries



Package	Package Version	arn
openpyxl	3.1.2	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-openpyxl:1
jinja2	3.1.2	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-jinja2:1
redshift-connector	2.0.910	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-redshift-connector:1
boto3	1.26.129	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-boto3:1
aws-requests-auth	0.4.3	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-aws-requests-auth:1
pyqldb	3.2.2	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-pyqldb:1
numpy	1.24.3	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-numpy:1
requests	2.30.0	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-requests:1
Pillow	9.5.0	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-Pillow:2
dynamodb-encryption-sdk	3.2.0	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-dynamodb-encryption-sdk:1
idna	3.4	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-idna:1
bcrypt	4.0.1	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-bcrypt:1
pandas	2.0.1	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-pandas:1
cryptography	40.0.2	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-cryptography:1
aws-xray-sdk	2.12.0	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-aws-xray-sdk:1
mysql-connector-python	8.0.33	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-mysql-connector-python:1
beautifulsoup4	4.12.2	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-beautifulsoup4:1

Repo Lambda Layers

Demo: Third Party Libraries



Add layer

Function runtime settings

Runtime
Python 3.10

Architecture
x86_64

Choose a layer

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 or organization.

● Specify an ARN
Specify a layer by providing the ARN.

Specify an ARN
Specify a layer by providing the Amazon Resource Name (ARN).

arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-Pillow:2

Description
Pillow==9.5.0 | 0d05c9143844cba6eb3ea059d1a25496089e1e52ef62ac2faf15c3f95f8e18a6

Compatible runtimes
Python 3.10

Compatible architectures
x86_64

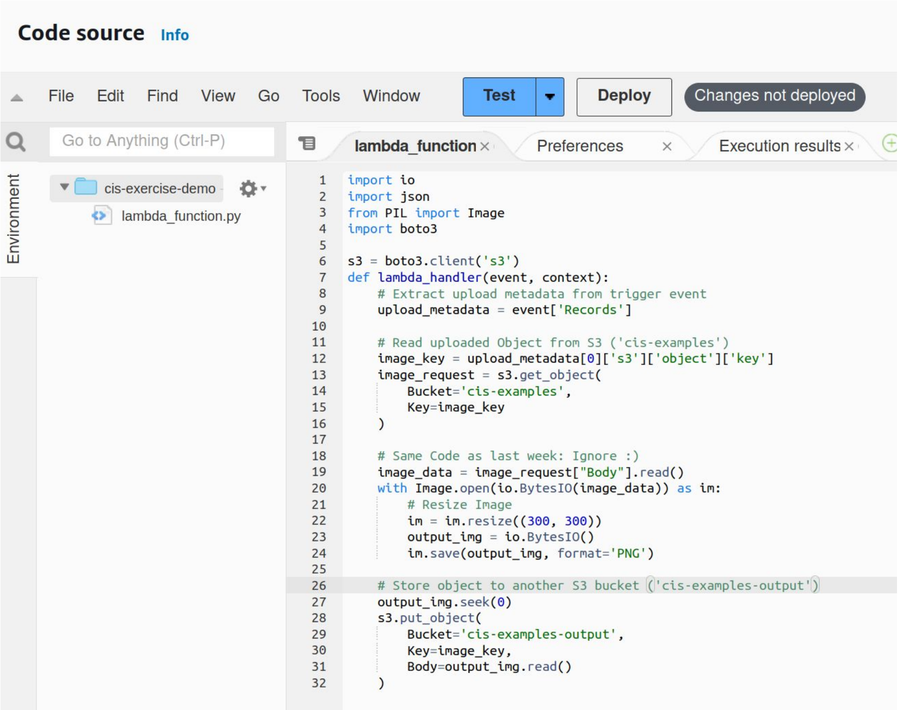
Add layer

Choose from the AWS provided layers with compatible runtimes. Or, choose a custom layer in your account that you or your organization created.

You can also specify the Amazon Resource Name (ARN) of a layer. Entering an ARN enables you to use a layer that is shared by another account, or a layer that does not match your function's runtime. The layer must be in the same AWS Region as your function. Enter the full layer version ARN in the following format: arn:aws:lambda: `us-east-2:123456789012:layer:my-layer:1`

Learn more
Lambda layers

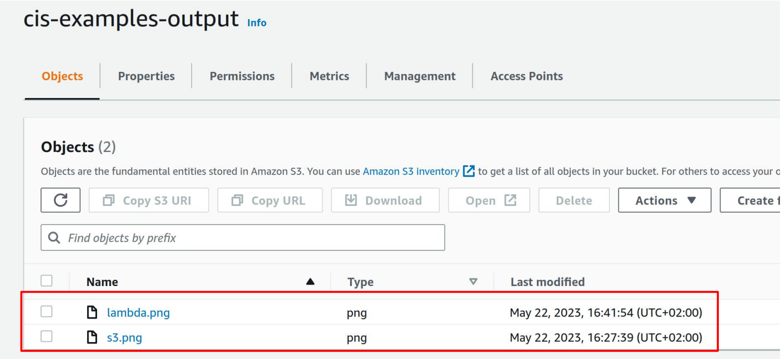
Demo: Writing the Code



```python
import io
import json
from PIL import Image
import boto3

s3 = boto3.client('s3')
def lambda_handler(event, context):
    # Extract upload metadata from trigger event
    upload_metadata = event['Records']

    # Read uploaded Object from S3 ('cis-examples')
    image_key = upload_metadata[0]['s3']['object']['key']
    image_request = s3.get_object(
        Bucket='cis-examples',
        Key=image_key
    )

    # Same Code as last week: Ignore :)
    image_data = image_request["Body"].read()
    with Image.open(io.BytesIO(image_data)) as im:
        # Resize Image
        im = im.resize((300, 300))
        output_img = io.BytesIO()
        im.save(output_img, format='PNG')

    # Store object to another S3 bucket ('cis-examples-output')
    output_img.seek(0)
    s3.put_object(
        Bucket='cis-examples-output',
        Key=image_key,
        Body=output_img.read()
    )
```

- First, we extract the key of the uploaded image (an example Input can be found here)
- Next, we retrieve the Image from S3 and do the processing
- Finally, we store the cropped Image in the output Bucket (cis-examples-output)
- **Important**: Do *not* store the Image to the same (Input) Bucket

Demo: End-2-End Testing

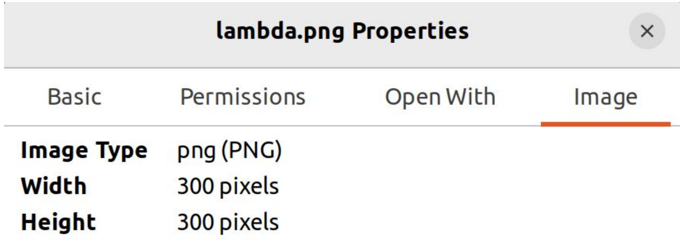


cis-examples-output

Objects (2)

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 o

□	Name ▲	Type ▽	Last modified
□	lambda.png	png	May 22, 2023, 16:41:54 (UTC+02:00)
□	s3.png	png	May 22, 2023, 16:27:39 (UTC+02:00)



lambda.png Properties

Image Type	png (PNG)
Width	300 pixels
Height	300 pixels

- Upload an Image to the Input Bucket (cis-exercises)
- After refreshing, the cis-examples-output Bucket should contain an Image with the same key
- Download the Image and verify that it got processed correctly