Lab 1: Deploy an S3 bucket + Lambda function using Python CDK

1. Prerequisites

- Python 3.9+ installed (`python3 --version`)
- AWS CLI installed & configured (`aws configure list`)
- Node.js 22.x+ installed (`node -v`)
- AWS CDK CLI installed (`npm install -g aws-cdk` → `cdk --version`)

If not bootstrapped yet:

```
cdk bootstrap aws://<ACCOUNT_ID>/<REGION>

```

---

2. Create CDK Project

```
mkdir lab1-s3-lambda && cd lab1-s3-lambda
cdk init app --language python

```

This creates:

- `app.py` (entrypoint)
- `lab1_s3_lambda/` (your stack folder)
- `.venv/` (virtual env)

---

3. Setup Virtual Environment

```
python3 -m venv .venv
source .venv/bin/activate   # (Linux/macOS)
# Windows: .venv\Scripts\activate
pip install -r requirements.txt

```

---

4. Install Dependencies

Add AWS CDK libraries for S3 + Lambda:

```
pip install aws-cdk-lib constructs

```

---

5. Write the Stack Code

Open `lab1_s3_lambda/lab1_s3_lambda_stack.py` and replace with:

```
from aws_cdk import (
    Stack,
    aws_s3 as s3,
    aws_lambda as _lambda,
    aws_s3_notifications as s3n,
)
```

```
from constructs import Construct

class Lab1S3LambdaStack(Stack):

    def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
        super().__init__(scope, construct_id, **kwargs)

        # Create S3 bucket
        bucket = s3.Bucket(self, "MyBucket",
            versioned=True,
            removal_policy=aws_cdk.RemovalPolicy.DESTROY,
            auto_delete_objects=True
        )

        # Create Lambda function
        fn = _lambda.Function(self, "MyLambda",
            runtime=_lambda.Runtime.PYTHON_3_9,
            handler="handler.main",
            code=_lambda.Code.from_asset("lambda")
        )

        # Add event notification: when object created in S3 → trigger Lambda
        bucket.add_event_notification(
            s3.EventType.OBJECT_CREATED,
            s3n.LambdaDestination(fn)
        )

```

---

6. Create Lambda Code

Make a folder `lambda/` in project root:

```
mkdir lambda

```

Inside `lambda/handler.py`:

```
def main(event, context):
    print("Event received:", event)
    return {
        "statusCode": 200,
        "body": "Hello from Lambda, triggered by S3!"
    }

```

---

7. Synthesize the CloudFormation Template

```
cdk synth

```

✅ This should generate a CloudFormation template in `cdk.out/`.

---

8. Deploy the Stack

```
cdk deploy

```

- Type **y** to approve security-related changes.
- CDK will create:
  - An S3 bucket (auto-named like `lab1s3lambda-MyBucket-XYZ`)

- A Lambda function
- Event notification binding

---

9. Test the Setup

Upload a file to the S3 bucket:

```
aws s3 cp test.txt s3://<YOUR_BUCKET_NAME>/
```

Check Lambda logs:

```
aws logs tail /aws/lambda/MyLambda --follow
```

You should see the event details.

---

10. Destroy the Stack (Cleanup)

```
cdk destroy
```

This removes all resources (since we used `auto_delete_objects=True`).

✅ Summary

In this lab, you:

- Created a Python CDK project
- Defined an S3 bucket + Lambda function
- Connected S3 events to Lambda
- Deployed and tested the stack