

Assignment 02

Name : Yash Rahate

Div : D15B

Roll No. : 48

Start the Learner Lab and export the credentials from the CLI.

ALLv2EN-... > Modules > AWS Acad...
> Launch AWS Academy Learner Lab

AWS Used \$0.3 of \$50 03:54 ▶ Start Lab ■ End Lab ⓘ AWS Details ⓘ Readme ↺ Reset ✕

```
eee_w_3428291@runweb148728:~$ export AWS_ACCESS_KEY_ID="ASIA5KXGVK2M5VBZ75"
eee_w_3428291@runweb148728:~$ export AWS_SECRET_ACCESS_KEY="Q+wyYDwy05Sg0bmyrAFzRhDh1anmHIjBwTUMvM"
eee_w_3428291@runweb148728:~$ export AWS_SESSION_TOKEN="Iqo3b3jp22luX2VjEhJ////////wEaCXVzLXdic3Q0tMlJHMEUCID+rHA
2+HnEv72wQUmW56L0d31TD1L1t+0S57VcyNA1EApAqrbK3naG1c31Yudn0b3ezFMc84k3w+Q3rRvst1QqTgIIRABGgw5MTYzODU1MTISMtClDK
10eZ11neyFHCzX5qTAt8nQmaw0RyHkKxKQf5jY02Lcma63ac1Vx0+LASrrSusRTK4j4+1bwj22jd43/YX41kolidcXUKEEjNRkpZjnopv4Hfbue7
TN21Ryq2p0BS5eiekDQ1j8tH511C7r9Fxo0mcBP0R1FcY3jzj6k60pRyH+v4aAobtdgV08PU+CG6Uv/K3/yzHeB+ZYvY8ZQR/2oYmkkYHlC1
gugNu468DwvKTAk9P5PCFUsnzao3I1d0d3b1dapFRk9AMQFQ0kHx+R1HemJ1HwkE9j8asJ4ckVhIdR3EmdCFQV1833so31k0ktsCQCDPpa312y1A9M
TSb01vY1K19p+Q8tM1X2mUB1mSM3e4caRR0YmTvUkM0LnxLgG0p0B+0057HmIV9JRq1asc5j6oFZpq7q484wZn1mP403Hzfndp2wBU001cw7XDA648
h0P6NbYy933hZKKEN14D8Epm1Rf3gGD8o/Ne5g0jj9bv2YGoRDeONFco31UYSF9pnXMMb71oWZwJ3zt/E+1TIXJ1bpumVHs4aAAmWIpj6ZFvkbCW
x/BWZMEfa2jbT2jKkPfeR5+EV5kt1IA=="
eee_w_3428291@runweb148728:~$
```

In the Learner Lab, there is usually a predefined IAM role that you can use. This role should already have the necessary permissions to interact with AWS services (like Lambda and S3).

www.google.com - Search | aws learner lab - Search | Launch AWS Academy Learner Lab | Home | EC2 | us-east-1

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home:

Services Search [Alt+S] N. Virginia voclabs/user3402688=2022.yash.rahate@ves.ac.in @ 3046-4946-3186

Roles (21) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	LabRole	Account: 916385512917	12 hours ago

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

www.google.com - Search | aws learner lab - Search | Launch AWS Academy Learner Lab | Home | EC2 | us-east-1

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home:

Services Search [Alt+S] N. Virginia voclabs/user3402688=2022.yash.rahate@ves.ac.in @ 3046-4946-3186

[IAM](#) > [Roles](#) > [LabRole](#)

LabRole Info

Delete Edit

Summary

Creation date

August 07, 2024, 09:06 (UTC+05:30)

Link to switch roles in console

<https://signin.aws.amazon.com/switchrole?roleName=LabRole&account=916385512917>

Instance profile ARN

[arn:aws:iam::916385512917:instance-profile/LabInstanceProfile](#)

Last activity

12 hours ago

Maximum session duration

1 hour

ARN copied

arn:aws:iam::916385512917:role/LabRole

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

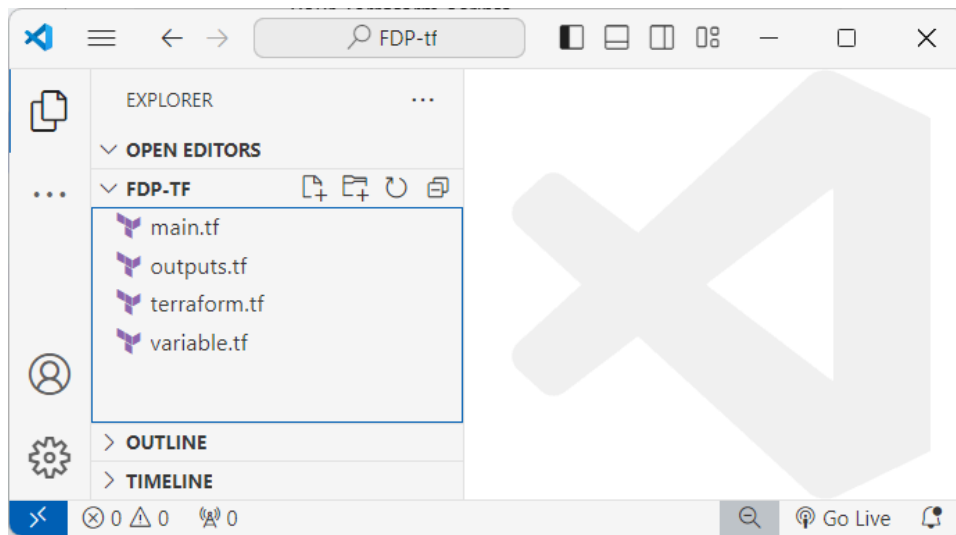
- Variabl
-

Create a Folder for the Project:

- Create a new folder on your local machine (for example: FDP-tf) where you will store your Terraform scripts.

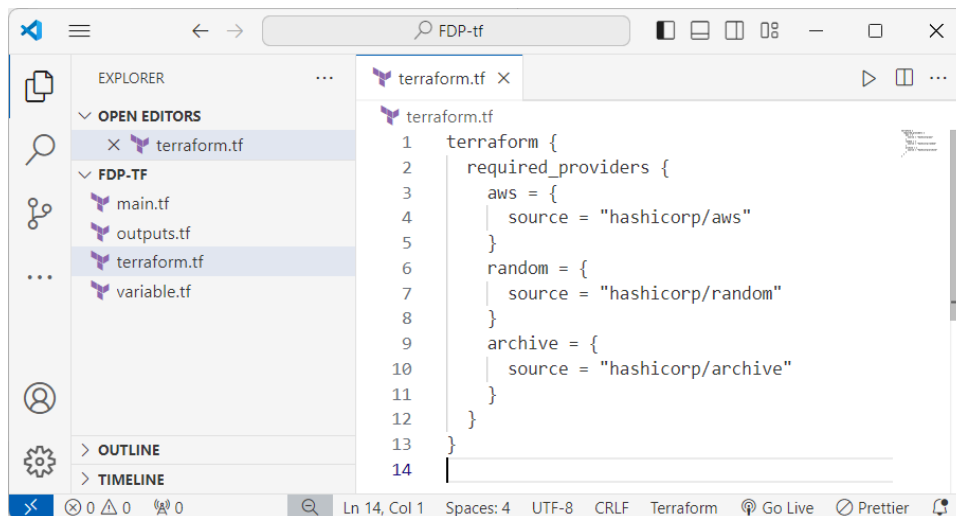
Set Up Terraform Configuration:

- Inside your folder, create four files:
 - terraform.tf
 - main.tf
 - variable.tf
 - outputs.tf



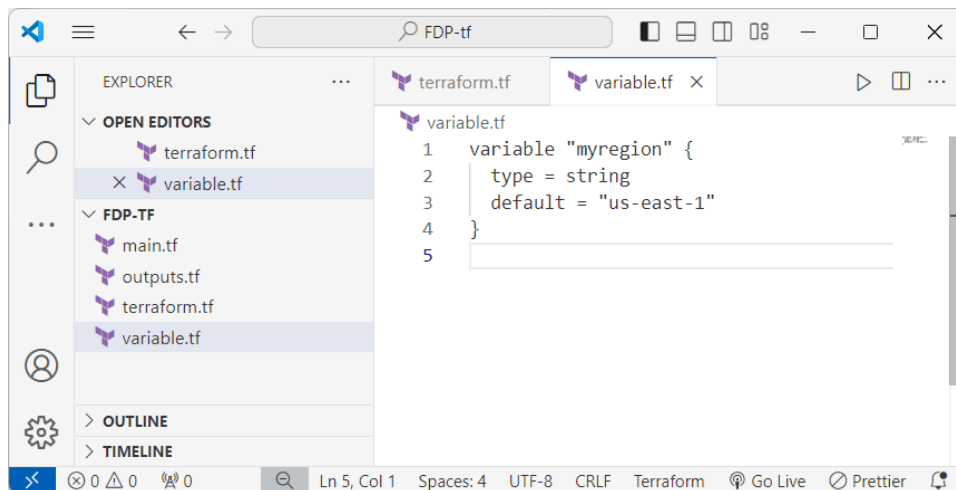
terraform.tf

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
    }
    random = {
      source = "hashicorp/random"
    }
    archive = {
      source = "hashicorp/archive"
    }
  }
}
```



variable.tf

```
variable "myregion" {
  type = string
  default = "us-east-1"
}
```



main.tf

```
provider "aws" {
  access_key = "YOUR_ACCESS_KEY"
  secret_key = "YOUR_SECRET_KEY"
  token      = "YOUR_TOKEN"
  region     = var.myregion
}
```

```
resource "random_pet" "bucketname" {
  length = 3
  prefix = "fdp"
}

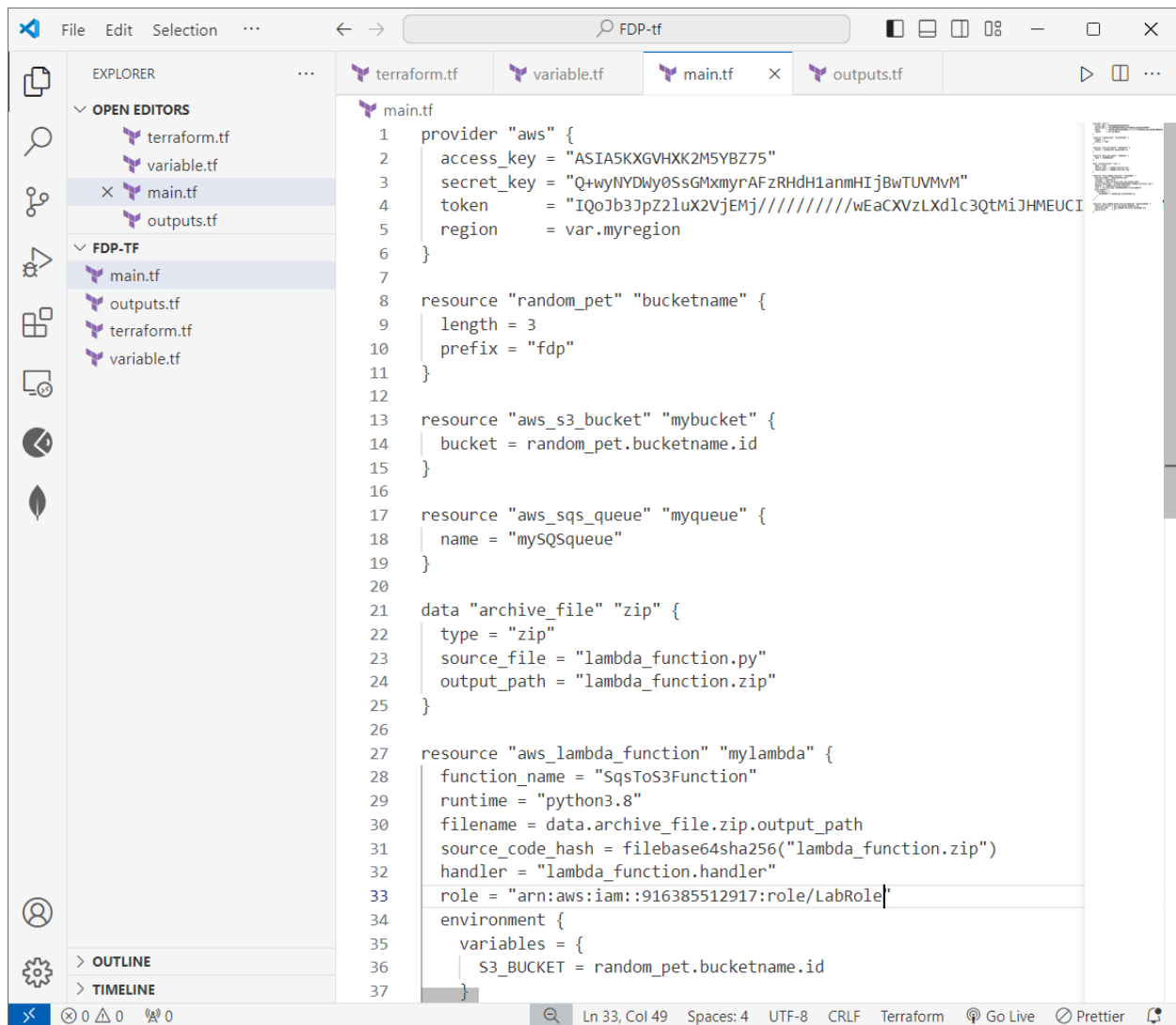
resource "aws_s3_bucket" "mybucket" {
  bucket = random_pet.bucketname.id
}

resource "aws_sqs_queue" "myqueue" {
  name = "mySQSqueue"
}

data "archive_file" "zip" {
  type = "zip"
  source_file = "lambda_function.py"
  output_path = "lambda_function.zip"
}

resource "aws_lambda_function" "mylambda" {
  function_name = "SqsToS3Function"
  runtime = "python3.8"
  filename = data.archive_file.zip.output_path
  source_code_hash = filebase64sha256("lambda_function.zip")
  handler = "lambda_function.handler"
  role = "arn:aws:iam::YOUR_IAM_ROLE"
  environment {
    variables = {
      S3_BUCKET = random_pet.bucketname.id
    }
  }
}

resource "aws_lambda_event_source_mapping" "SqsToLambda" {
  event_source_arn = aws_sqs_queue.myqueue.arn
  function_name = aws_lambda_function.mylambda.arn
  batch_size = 1
}
```

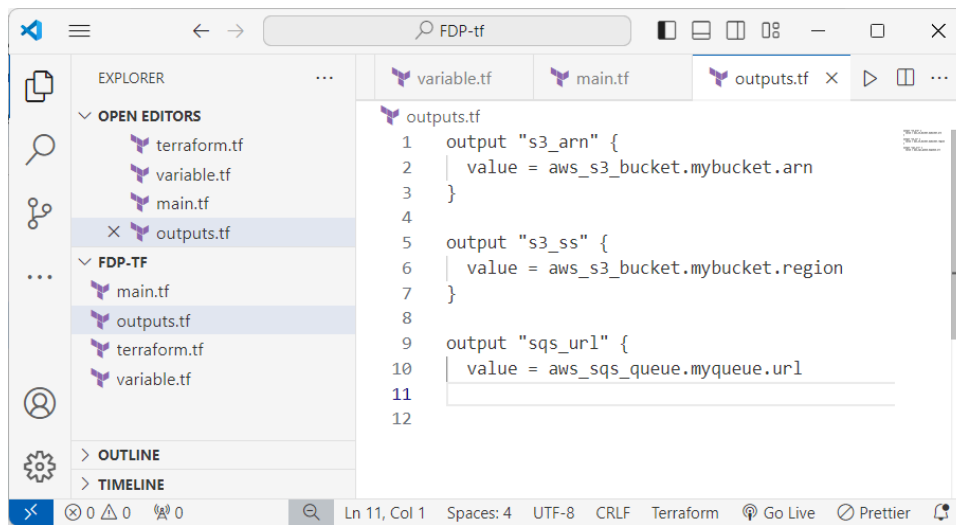


outputs.tf

```
output "s3_arn" {
  value = aws_s3_bucket.mybucket.arn
}

output "s3_ss" {
  value = aws_s3_bucket.mybucket.region
}

output "sqs_url" {
  value = aws_sqs_queue.myqueue.url
}
```



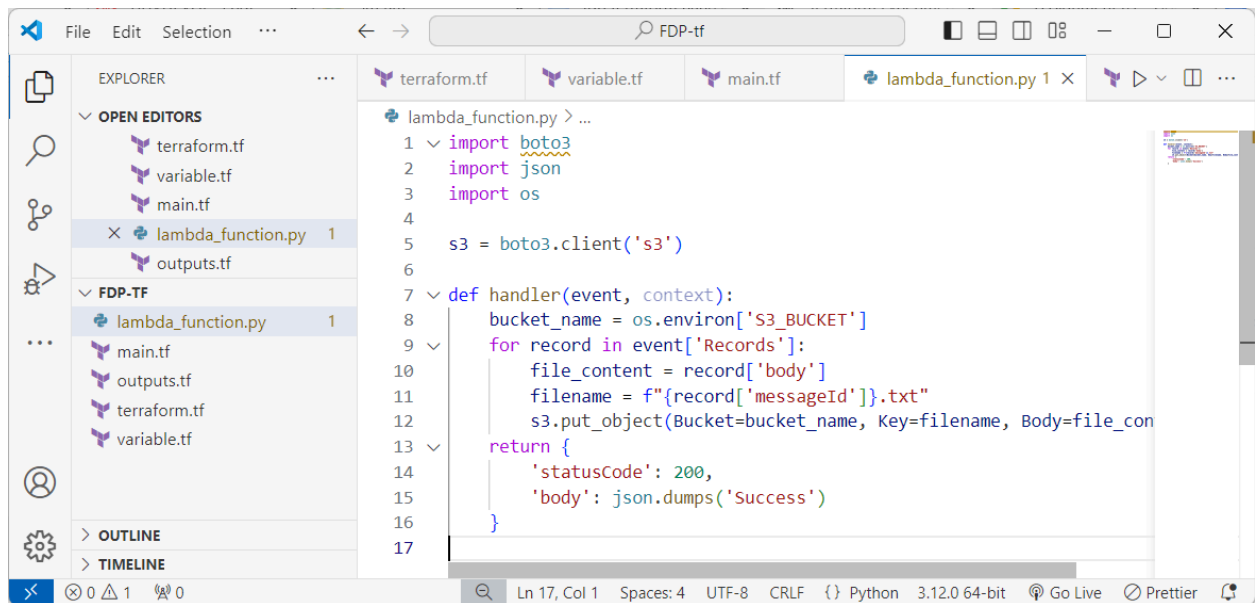
Create Lambda Python File:

- In the same directory, create a file named `lambda_function.py` and paste the following code:

```
import boto3
import json
import os

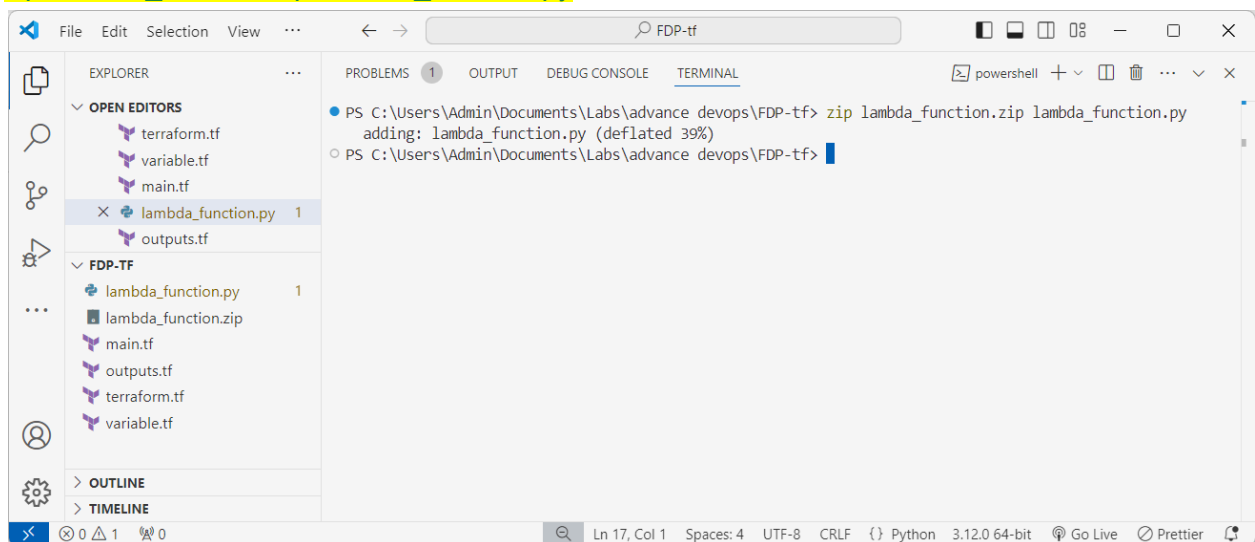
s3 = boto3.client('s3')

def handler(event, context):
    bucket_name = os.environ['S3_BUCKET']
    for record in event['Records']:
        file_content = record['body']
        filename = f"{record['messageId']}.txt"
        s3.put_object(Bucket=bucket_name, Key=filename, Body=file_content)
    return {
        'statusCode': 200,
        'body': json.dumps('Success')
    }
```



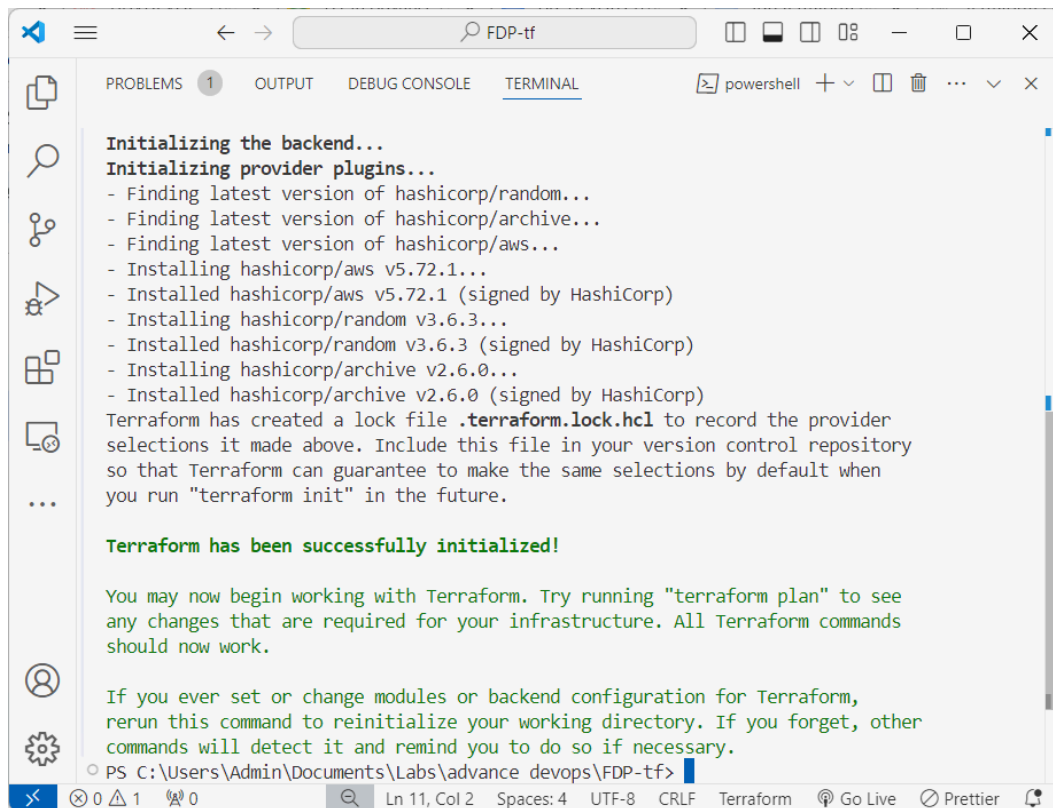
Open the terminal or command prompt in the same directory and run:

`zip lambda_function.zip lambda_function.py`



To Initialize Terraform, run the command:

`terraform init`



The screenshot shows the VS Code interface with the 'TERMINAL' tab active. The terminal output displays the process of initializing Terraform, including finding and installing provider plugins (aws, random, archive) and creating a lock file. The status bar at the bottom indicates the file is 'Terraform' and includes icons for 'Go Live' and 'Prettier'.

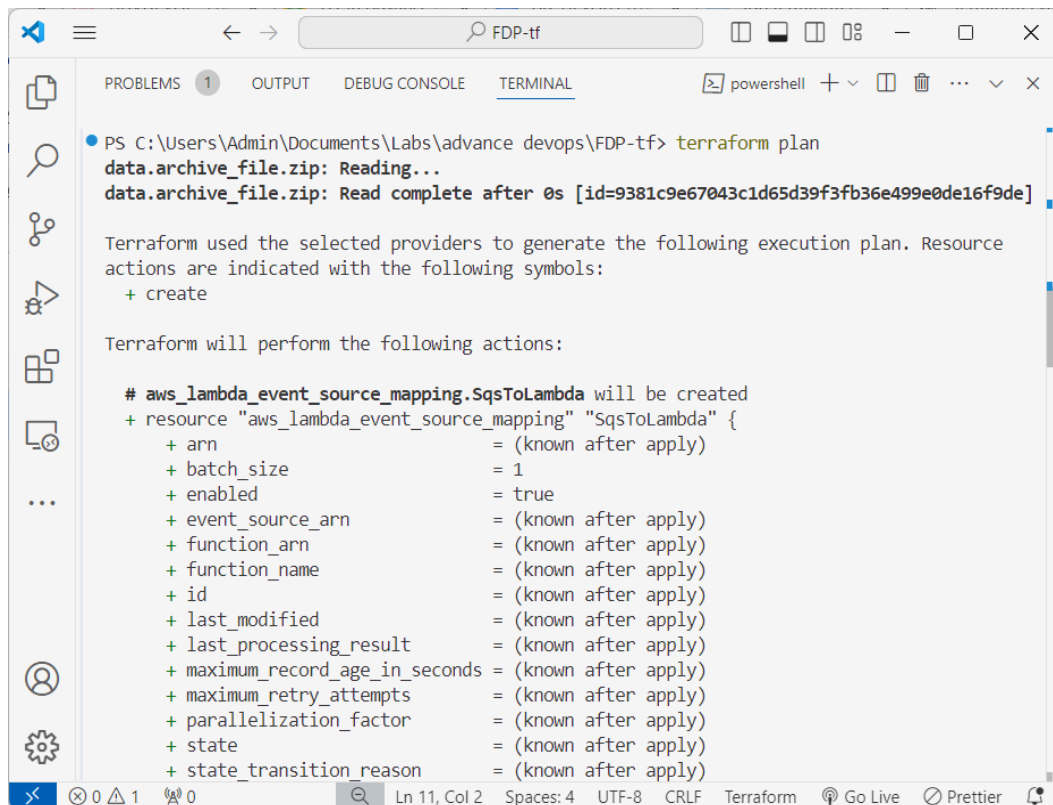
```
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/random...
- Finding latest version of hashicorp/archive...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.72.1...
- Installed hashicorp/aws v5.72.1 (signed by HashiCorp)
- Installing hashicorp/random v3.6.3...
- Installed hashicorp/random v3.6.3 (signed by HashiCorp)
- Installing hashicorp/archive v2.6.0...
- Installed hashicorp/archive v2.6.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\Admin\Documents\Labs\advance devops\FDP-tf>
```

To Plan the Infrastructure, Run the command
terraform plan



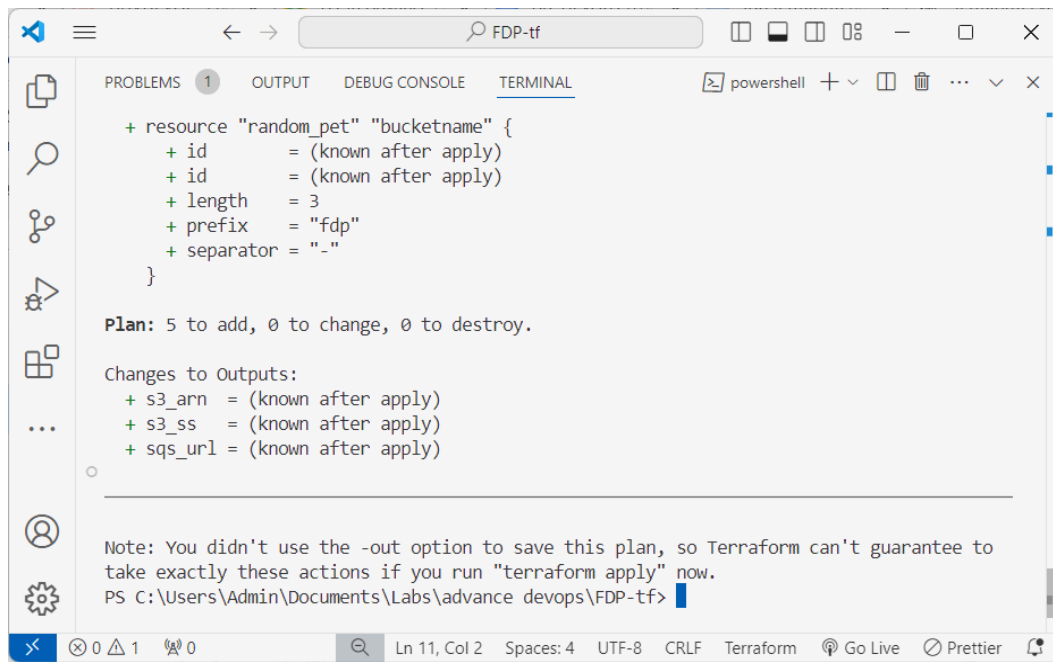
The screenshot shows the VS Code interface with the 'TERMINAL' tab active. The terminal output displays the command 'terraform plan' being executed. It shows the reading of the data.archive_file.zip and the generation of an execution plan. The plan indicates that a resource 'aws_lambda_event_source_mapping' will be created with various attributes. The status bar at the bottom is the same as in the previous screenshot.

```
PS C:\Users\Admin\Documents\Labs\advance devops\FDP-tf> terraform plan
data.archive_file.zip: Reading...
data.archive_file.zip: Read complete after 0s [id=9381c9e67043c1d65d39f3fb36e499e0de16f9de]

Terraform used the selected providers to generate the following execution plan. Resource
actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_lambda_event_source_mapping.SqsToLambda will be created
+ resource "aws_lambda_event_source_mapping" "SqsToLambda" {
+   arn                = (known after apply)
+   batch_size         = 1
+   enabled             = true
+   event_source_arn    = (known after apply)
+   function_arn        = (known after apply)
+   function_name       = (known after apply)
+   id                 = (known after apply)
+   last_modified       = (known after apply)
+   last_processing_result = (known after apply)
+   maximum_record_age_in_seconds = (known after apply)
+   maximum_retry_attempts = (known after apply)
+   parallelization_factor = (known after apply)
+   state              = (known after apply)
+   state_transition_reason = (known after apply)
}
```

The screenshot shows the VS Code interface with the 'FDP-tf' file open. The 'TERMINAL' tab is active, displaying the output of a Terraform plan. The plan shows a resource 'random_pet' with attributes 'id', 'length', 'prefix', and 'separator'. The plan indicates 5 resources to be added, 0 to be changed, and 0 to be destroyed. Below the plan, there are 'Changes to Outputs' for 's3_arn', 's3_ss', and 'sqs_url'. A note at the bottom states: 'Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now. PS C:\Users\Admin\Documents\Labs\advance devops\FDP-tf>'

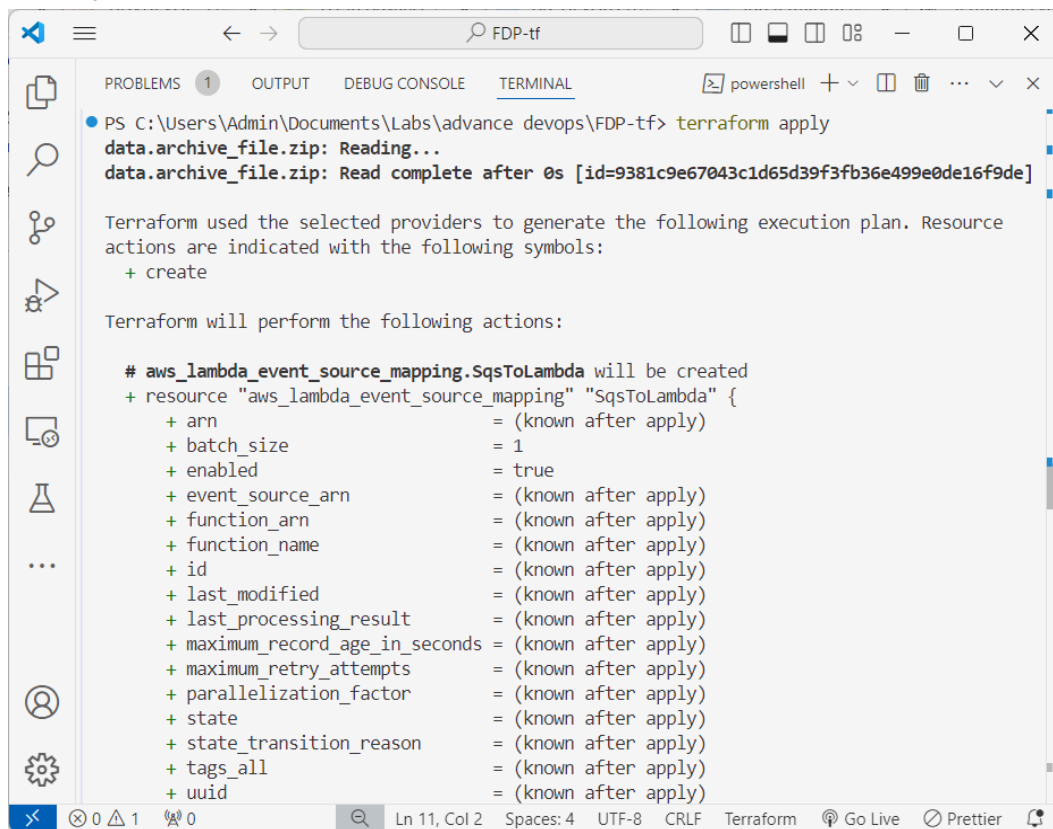
```
+ resource "random_pet" "bucketname" {
+   id      = (known after apply)
+   id      = (known after apply)
+   length  = 3
+   prefix  = "fdp"
+   separator = "-"
+ }

Plan: 5 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ s3_arn  = (known after apply)
+ s3_ss   = (known after apply)
+ sqs_url = (known after apply)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to
take exactly these actions if you run "terraform apply" now.
PS C:\Users\Admin\Documents\Labs\advance devops\FDP-tf>
```

If everything looks good, apply the plan by running **terraform apply**
(Enter **yes** when prompted. This will create the resources.)



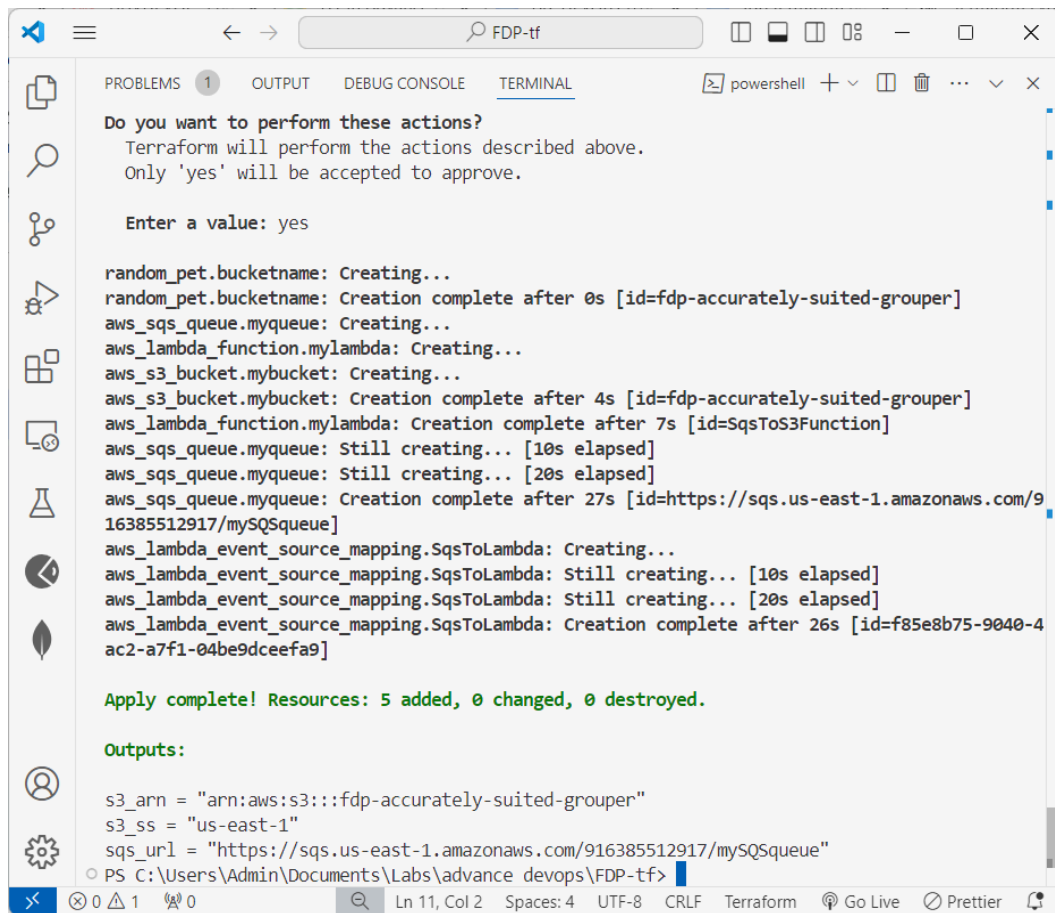
The screenshot shows the VS Code interface with the 'FDP-tf' file open. The 'TERMINAL' tab is active, displaying the output of the 'terraform apply' command. The command has been executed successfully, and the output shows the execution plan for the 'aws_lambda_event_source_mapping' resource. The plan indicates that the resource will be created with various attributes. The output also shows the Terraform version and the provider version.

```
PS C:\Users\Admin\Documents\Labs\advance devops\FDP-tf> terraform apply
data.archive_file.zip: Reading...
data.archive_file.zip: Read complete after 0s [id=9381c9e67043c1d65d39f3fb36e499e0de16f9de]

Terraform used the selected providers to generate the following execution plan. Resource
actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_lambda_event_source_mapping.SqsToLambda will be created
+ resource "aws_lambda_event_source_mapping" "SqsToLambda" {
+   arn              = (known after apply)
+   batch_size       = 1
+   enabled          = true
+   event_source_arn = (known after apply)
+   function_arn     = (known after apply)
+   function_name    = (known after apply)
+   id               = (known after apply)
+   last_modified    = (known after apply)
+   last_processing_result = (known after apply)
+   maximum_record_age_in_seconds = (known after apply)
+   maximum_retry_attempts = (known after apply)
+   parallelization_factor = (known after apply)
+   state            = (known after apply)
+   state_transition_reason = (known after apply)
+   tags_all         = (known after apply)
+   uuid             = (known after apply)
+ }
```



The screenshot shows a Visual Studio Code terminal window with the title bar 'FDP-tf'. The terminal output displays the results of a Terraform apply command. It starts with a confirmation prompt: 'Do you want to perform these actions? Terraform will perform the actions described above. Only 'yes' will be accepted to approve.' The user enters 'yes'. The output then lists the creation of several resources: 'random_pet.bucketname', 'aws_sqs_queue.myqueue', 'aws_lambda_function.mylambda', and 'aws_s3_bucket.mybucket'. It shows the progress of each resource, including creation times and IDs. For example, 'aws_s3_bucket.mybucket' is created after 4s with ID 'fdp-accurately-suited-grouper'. The output concludes with 'Apply complete! Resources: 5 added, 0 changed, 0 destroyed.' and lists the outputs: 's3_arn', 's3_ss', and 'sqs_url'. The terminal window also shows the file explorer on the left and the status bar at the bottom.

```
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

random_pet.bucketname: Creating...
random_pet.bucketname: Creation complete after 0s [id=fdp-accurately-suited-grouper]
aws_sqs_queue.myqueue: Creating...
aws_lambda_function.mylambda: Creating...
aws_s3_bucket.mybucket: Creating...
aws_s3_bucket.mybucket: Creation complete after 4s [id=fdp-accurately-suited-grouper]
aws_lambda_function.mylambda: Creation complete after 7s [id=SqsToS3Function]
aws_sqs_queue.myqueue: Still creating... [10s elapsed]
aws_sqs_queue.myqueue: Still creating... [20s elapsed]
aws_sqs_queue.myqueue: Creation complete after 27s [id=https://sqs.us-east-1.amazonaws.com/916385512917/mySQSqueue]
aws_lambda_event_source_mapping.SqsToLambda: Creating...
aws_lambda_event_source_mapping.SqsToLambda: Still creating... [10s elapsed]
aws_lambda_event_source_mapping.SqsToLambda: Still creating... [20s elapsed]
aws_lambda_event_source_mapping.SqsToLambda: Creation complete after 26s [id=f85e8b75-9040-4ac2-a7f1-04be9dceefa9]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

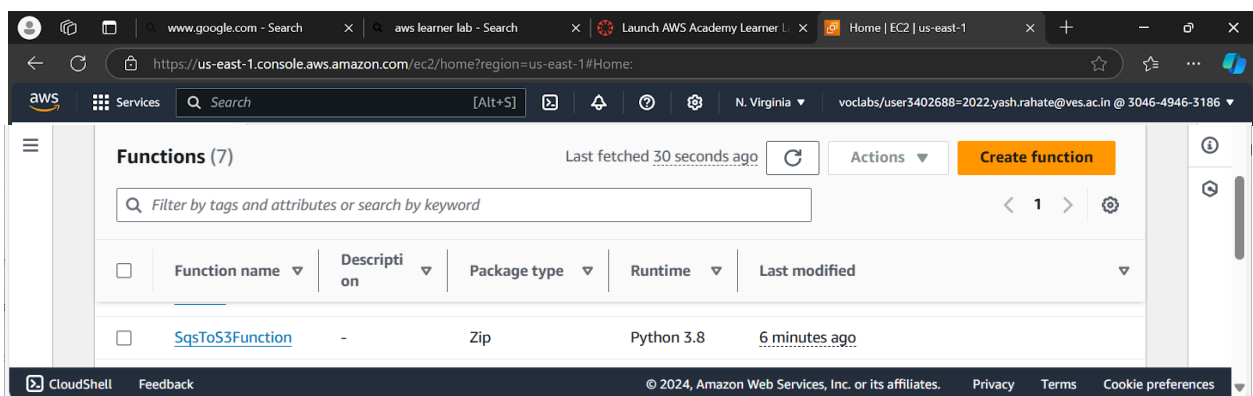
Outputs:

s3_arn = "arn:aws:s3:::fdp-accurately-suited-grouper"
s3_ss = "us-east-1"
sqs_url = "https://sqs.us-east-1.amazonaws.com/916385512917/mySQSqueue"
PS C:\Users\Admin\Documents\Labs\advance devops\FDP-tf>
```

Once the resources are created, you can log into your AWS console and verify that:

- An S3 bucket is created.
- An SQS queue is created.
- A Lambda function is created.

Lambda Function



Browser tabs: www.google.com - Search, aws learner lab - Search, Launch AWS Academy Learner I, Home | EC2 | us-east-1

Address bar: https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home:

Services Search [Alt+S] N. Virginia voclabs/user3402688=2022.yash.rahate@ves.ac.in @ 3046-4946-3186

Lambda > Functions > SqsToS3Function

Throttle Copy ARN Actions

Function overview Info

Export to Application Composer Download

Diagram Template

SqsToS3Function

Layers (0)

SQS

+ Add trigger

+ Add destination

Description

Last modified 10 minutes ago

Function ARN
arn:aws:lambda:us-east-1:9163855129:17:function:SqsToS3Function

Function URL Info

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Browser tabs: www.google.com - Search, aws learner lab - Search, Launch AWS Academy Learner I, Home | EC2 | us-east-1

Address bar: https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home:

Services Search [Alt+S] N. Virginia voclabs/user3402688=2022.yash.rahate@ves.ac.in @ 3046-4946-3186

Code Test Monitor Configuration Aliases Versions

Code source Info

Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

SqsToS3Function

lambda_function.py

```
1 import boto3
2 import json
3 import os
4
5 s3 = boto3.client('s3')
6
7 def handler(event, context):
8     bucket_name = os.environ['S3_BUCKET']
9     for record in event['Records']:
10         file_content = record['body']
11         filename = f"{record['messageId']}.txt"
12         s3.put_object(Bucket=bucket_name, Key=filename, Body=file_content)
13     return {
14         'statusCode': 200,
15         'body': json.dumps('Success')
16     }
17
```

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS Lambda console showing the configuration for the **SqsToS3Function**.

The function is configured with an **SQS** trigger. The configuration page shows the following details:

- Triggers (1)**: One trigger is listed, **SQS: mySQSqueue**, with the ARN `arn:aws:sqs:us-east-1:916385512917:mySQSqueue` and state **Enabled**.
- Function ARN**: `arn:aws:lambda:us-east-1:916385512917:function:SqsToS3Function`
- Function URL**: [Info](#)

The left sidebar shows the configuration tabs: **General configuration**, **Triggers**, **Permissions**, **Destinations**, **Function URL**, and **Environment**.

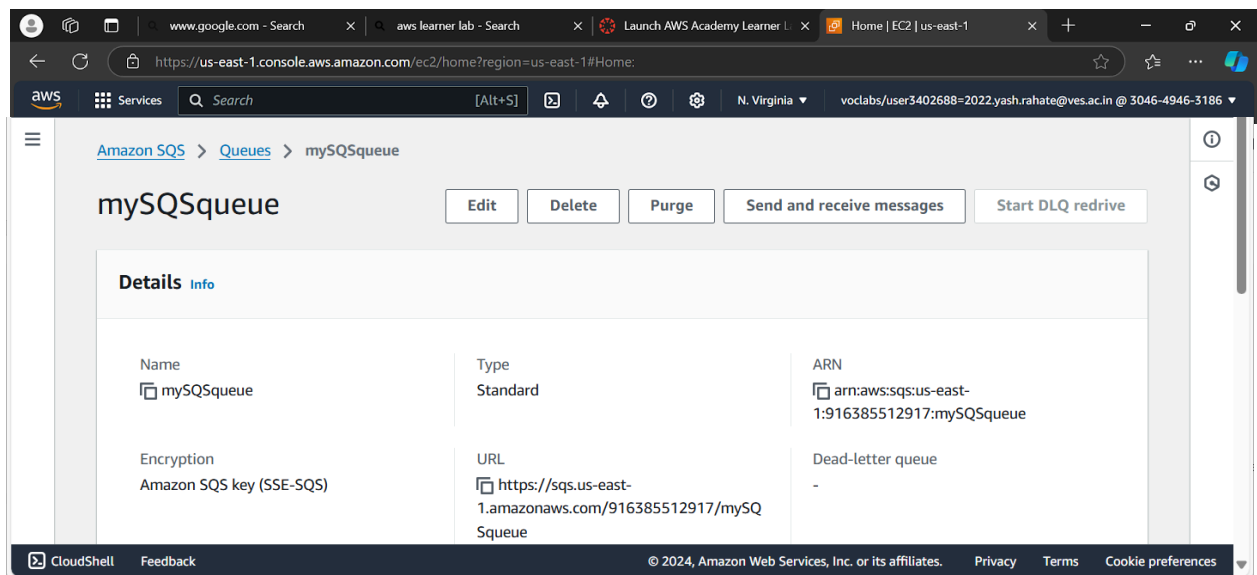
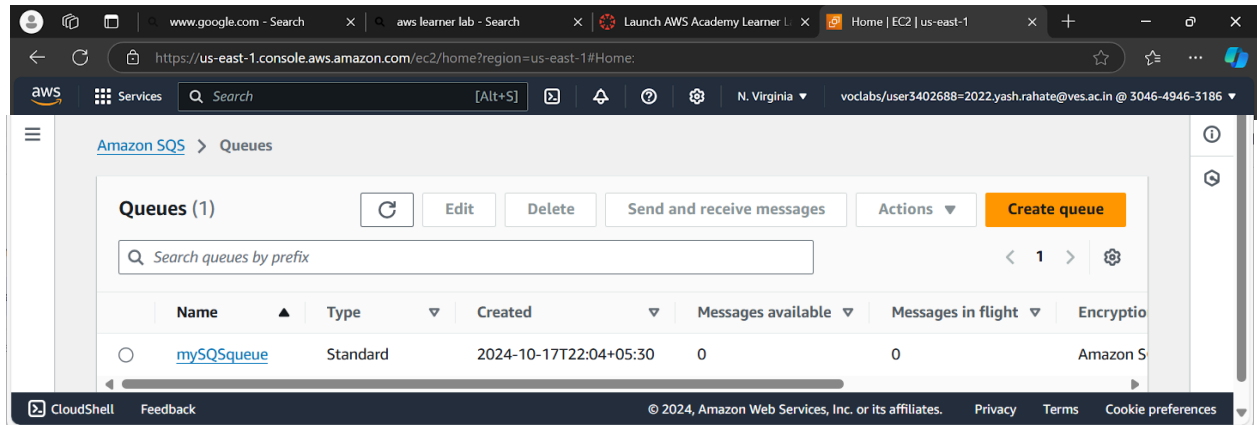
SQS queue

Second screenshot of the AWS Lambda console, showing the **Environment variables** configuration for the **SqsToS3Function**.

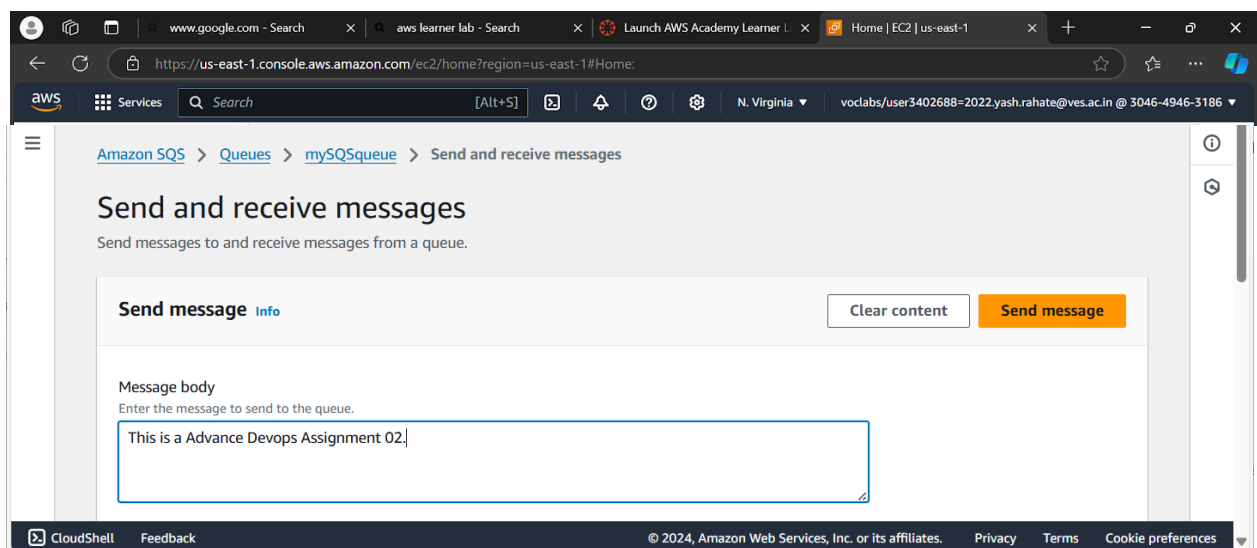
The **Environment variables (1)** section shows one variable:

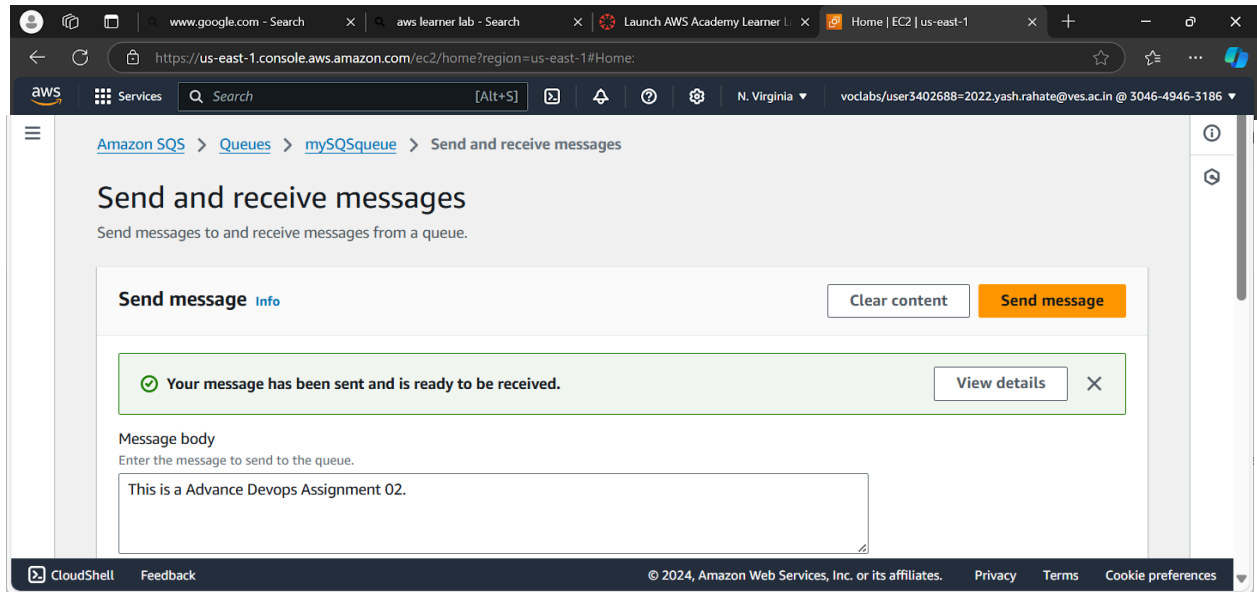
Key	Value
<code>S3_BUCKET</code>	<code>fdp-accurately-suited-grouper</code>

The left sidebar shows the configuration tabs: **General configuration**, **Triggers**, **Permissions**, **Destinations**, **Function URL**, and **Environment variables**.

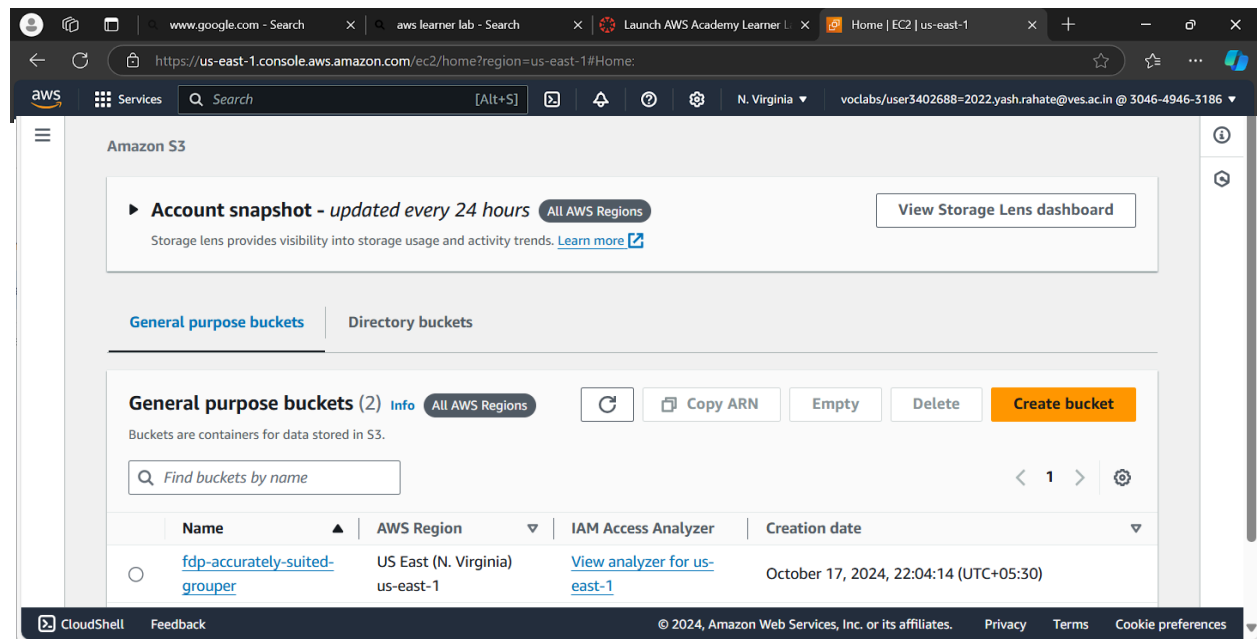


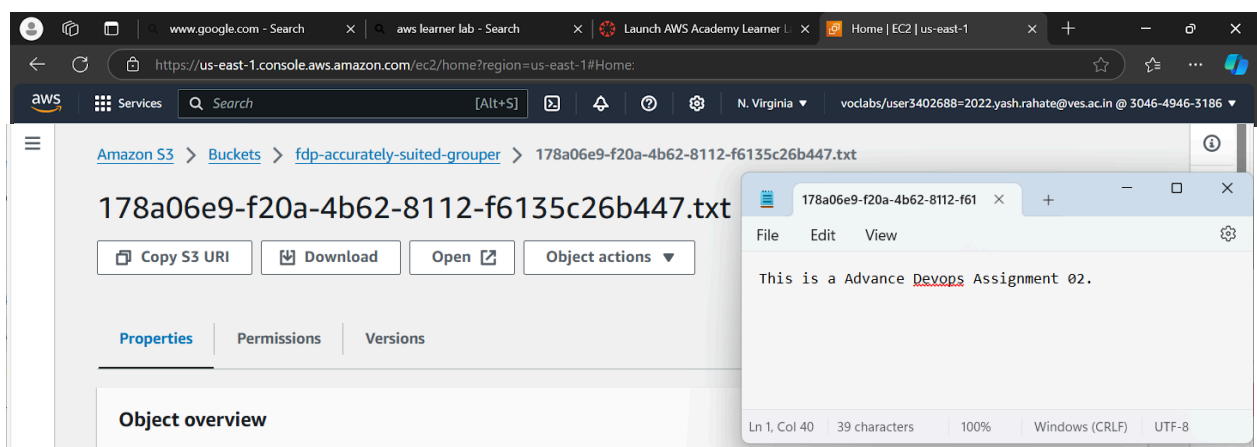
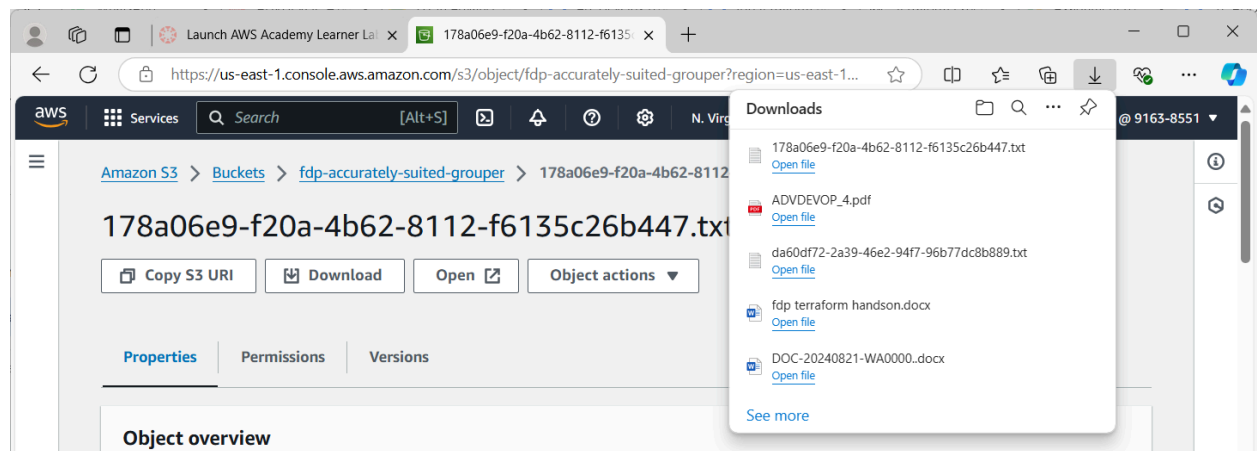
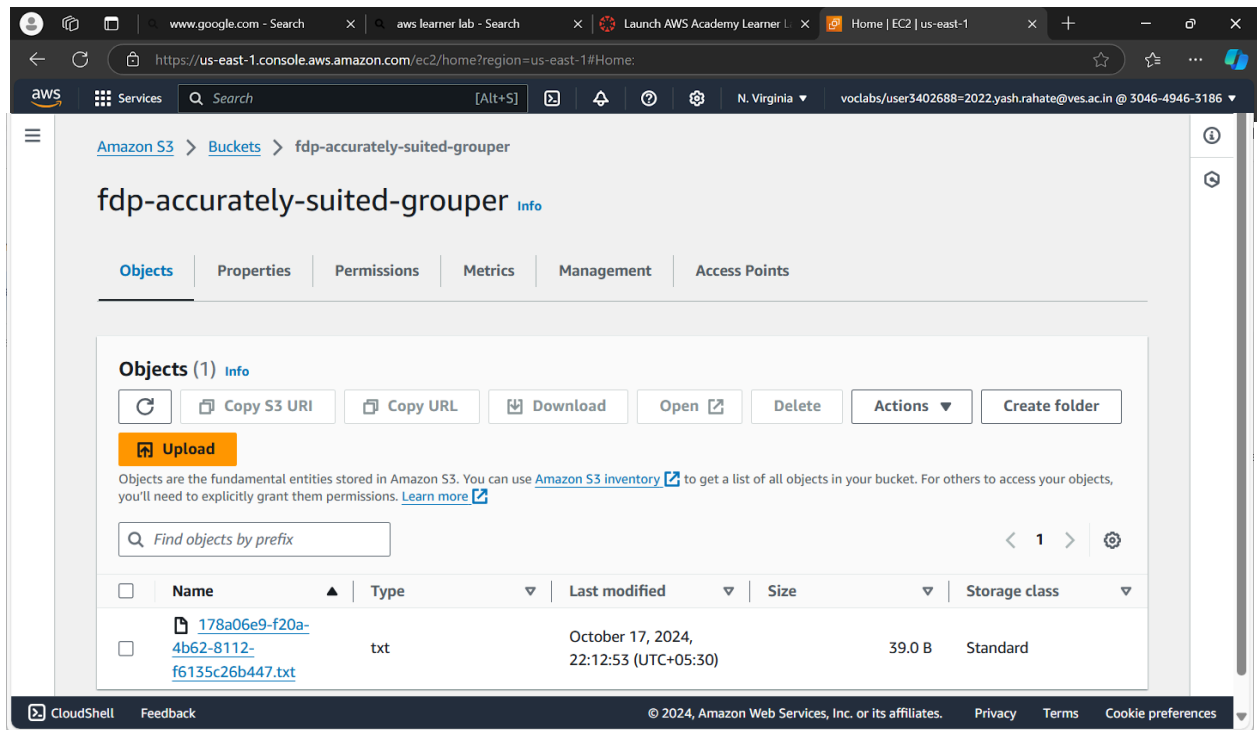
Send the message from the SQS.

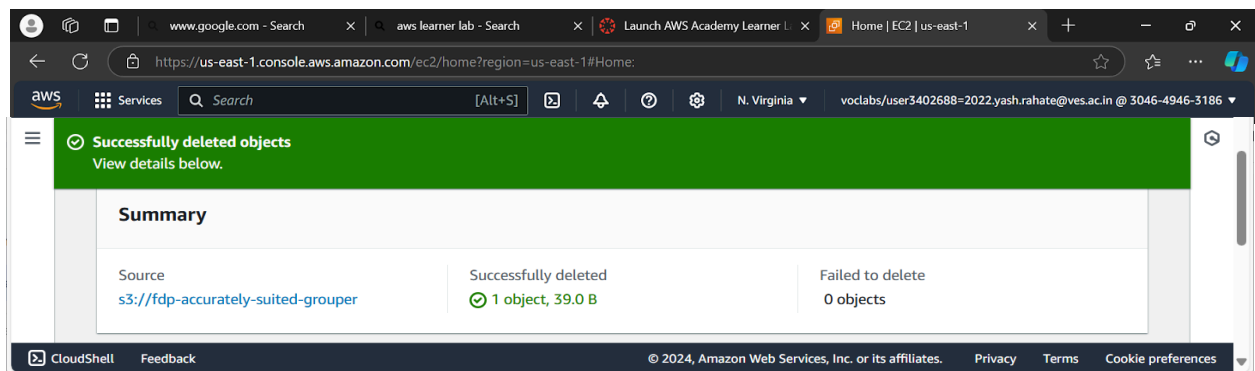
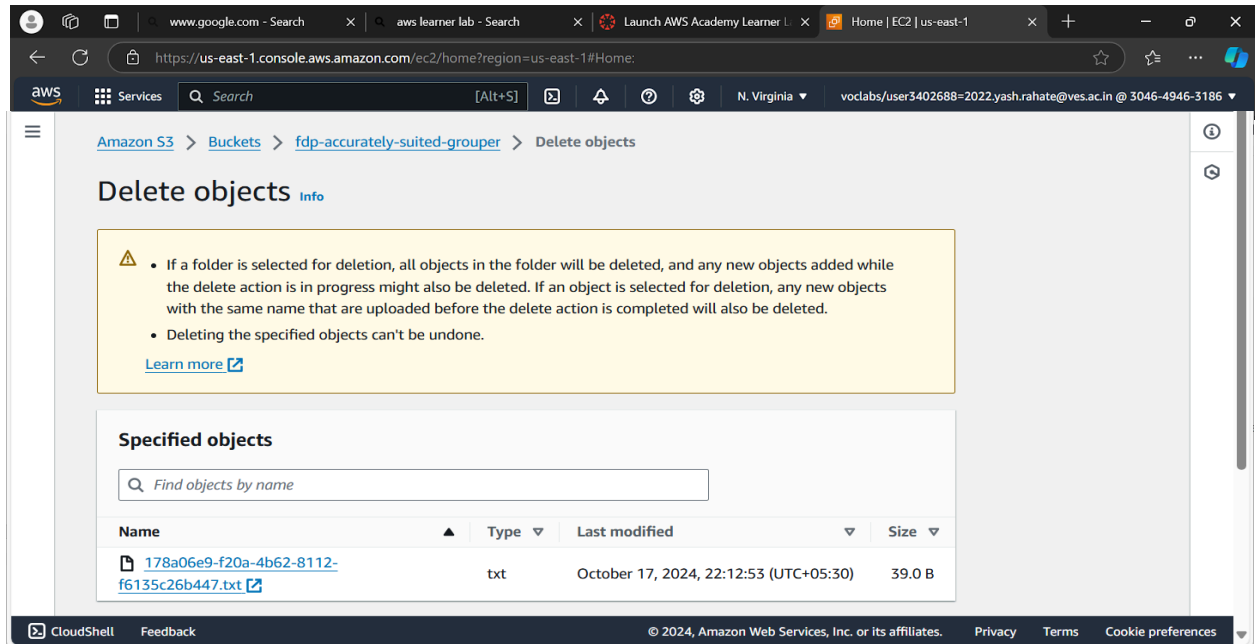




S3 bucket







If you want to clean up the resources after testing, you can destroy them by running:
terraform destroy
(Confirm the destruction by typing **yes**.)


```
PS C:\Users\Admin\Documents\Labs\advance devops\FDP-tf> terraform destroy
data.archive_file.zip: Reading...
random_pet.bucketname: Refreshing state... [id=fdp-accurately-suited-grouper]
data.archive_file.zip: Read complete after 0s [id=9381c9e67043c1d65d39f3fb36e499e0de16f9de]
aws_sqs_queue.myqueue: Refreshing state... [id=https://sqs.us-east-1.amazonaws.com/916385512917/mySQSqueue]
aws_lambda_function.mylambda: Refreshing state... [id=SqsToS3Function]
aws_s3_bucket.mybucket: Refreshing state... [id=fdp-accurately-suited-grouper]
aws_lambda_event_source_mapping.SqsToLambda: Refreshing state... [id=f85e8b75-9040-4ac2-a7f1-04be9dceefa9]

Terraform used the selected providers to generate the following execution plan. Resource
actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_lambda_event_source_mapping.SqsToLambda will be destroyed
- resource "aws_lambda_event_source_mapping" "SqsToLambda" {
  - arn = "arn:aws:lambda:us-east-1:916385512917:event-source-mapping:f85e8b75-9040-4ac2-a7f1-04be9dceefa9" -> null
  - batch_size = 1 -> null
  - bisect_batch_on_function_error = false -> null
  - enabled = true -> null
  - event_source_arn = "arn:aws:sqs:us-east-1:916385512917:mySQSqueue" -> null
  - function_arn = "arn:aws:lambda:us-east-1:916385512917:function:SqsToS3Function" -> null
}
```

```
# random_pet.bucketname will be destroyed
- resource "random_pet" "bucketname" {
  - id = "fdp-accurately-suited-grouper" -> null
  - length = 3 -> null
  - prefix = "fdp" -> null
  - separator = "-" -> null
}

Plan: 0 to add, 0 to change, 2 to destroy.

Changes to Outputs:
- s3_arn = "arn:aws:s3:::fdp-accurately-suited-grouper" -> null
- s3_ss = "us-east-1" -> null

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_s3_bucket.mybucket: Destroying... [id=fdp-accurately-suited-grouper]
aws_s3_bucket.mybucket: Destruction complete after 1s
random_pet.bucketname: Destroying... [id=fdp-accurately-suited-grouper]
random_pet.bucketname: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
PS C:\Users\Admin\Documents\Labs\advance devops\FDP-tf>
```