terraform.tf

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

variable.tf

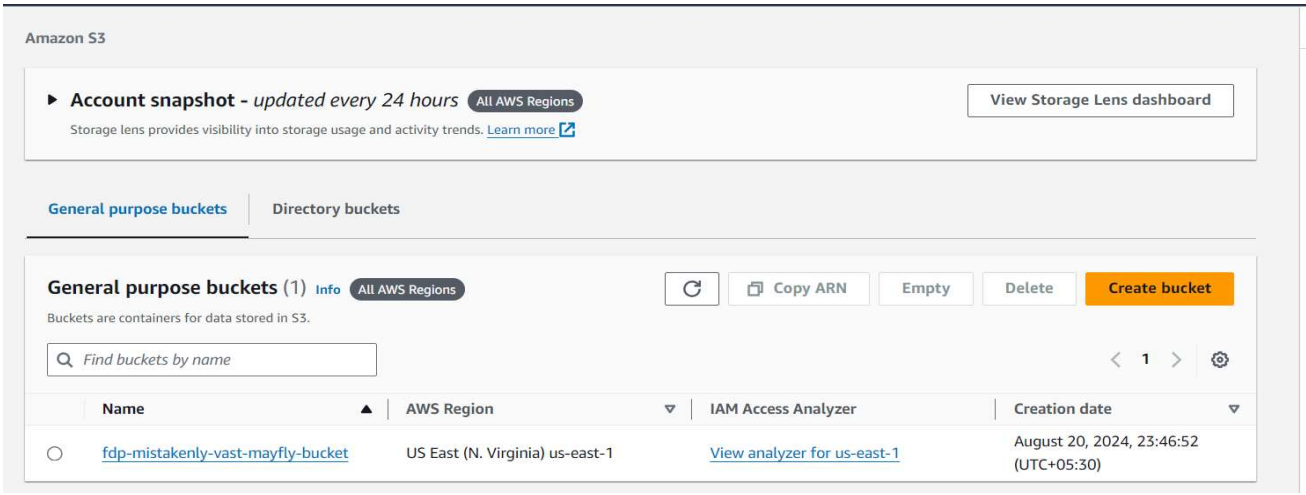
```
variable "myregion" {
  type = string
  default = "us-east-1"
}
```

main.tf

```hcl
provider "aws" {
  access_key = "ASIA4JFIID3KSL3R6QGR"
  secret_key = "pVJBm8BRcsoEwgAt1Nxe1lwkaJ8DYNrouIh+LfHK"
  token      = "IQoJb3JpZ2luX2VjEFsaCXVzLXdlc3QtMiJGMEQCIEkA9LnQh4l/BZuaAj/89eAgGKIiZX2+IjvkavWknzR2AiAE8cNTZX+Vk5ueiKjMfF5DYvvdh4UoIuwywL11WgqBQyq3AghkEA

  region = "us-east-1"
}


resource "random_pet" "romabucket1904" {
  length = 3
  prefix = "fdp"
}

resource "aws_s3_bucket" "romabucket1904" {
  bucket = "${random_pet.romabucket1904.id}-bucket"
}

#resource "aws_s3_bucket_acl" "romabucket1904_acl" {
  #bucket = aws_s3_bucket.romabucket1904.id
  #acl    = "private"
#}

resource "aws_iam_role" "lambda_role" {
  name = "lambda_execution_role"

  assume_role_policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Action = "sts:AssumeRole",
        Effect = "Allow",
        Principal = {
          Service = "lambda.amazonaws.com"
        },
      },
    ]
  })
```

```
C:\Terraform>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/archive...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/random...
- Installing hashicorp/archive v2.5.0...
- Installed hashicorp/archive v2.5.0 (signed by HashiCorp)
- Installing hashicorp/aws v5.63.1...
- Installed hashicorp/aws v5.63.1 (signed by HashiCorp)
- Installing hashicorp/random v3.6.2...
- Installed hashicorp/random v3.6.2 (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.
```

```
C:\Terraform>terraform plan

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_s3_bucket.roma.bucket12 will be created
  + resource "aws_s3_bucket" "roma.bucket12" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
      + bucket                      = (known after apply)
      + bucket_domain_name          = (known after apply)
      + bucket_prefix               = (known after apply)
      + bucket_regional_domain_name = (known after apply)
      + force_destroy               = false
      + hosted_zone_id              = (known after apply)
      + id                          = (known after apply)
      + object_lock_enabled         = (known after apply)
      + policy                      = (known after apply)
      + region                      = (known after apply)
      + request_payer               = (known after apply)
      + tags_all                    = (known after apply)
      + website_domain              = (known after apply)
      + website_endpoint            = (known after apply)

      + cors_rule (known after apply)

      + grant (known after apply)

      + lifecycle_rule (known after apply)

      + logging (known after apply)

      + object_lock_configuration (known after apply)

      + replication_configuration (known after apply)

      + server_side_encryption_configuration (known after apply)

      + versioning (known after apply)

      + website (known after apply)
    }

  # aws_s3_bucket_acl.roma.bucket12_acl will be created
  + resource "aws_s3_bucket_acl" "roma.bucket12_acl" {
      + acl    = "private"
      + bucket = (known after apply)
      + id     = (known after apply)

      + access_control_policy (known after apply)
    }

  # random_pet.roma.bucket12 will be created
  + resource "random_pet" "roma.bucket12" {
      + id        = (known after apply)
      + length    = 3
      + prefix    = "fdp"
      + separator = "-"
    }

Plan: 3 to add, 0 to change, 0 to destroy.
```




```
output "s3_arn" {
  value = aws_s3_bucket.roma bucket12.arn
}
```

Do terraform apply

```
output "s3_region" {
  value = var.myregion
}
```

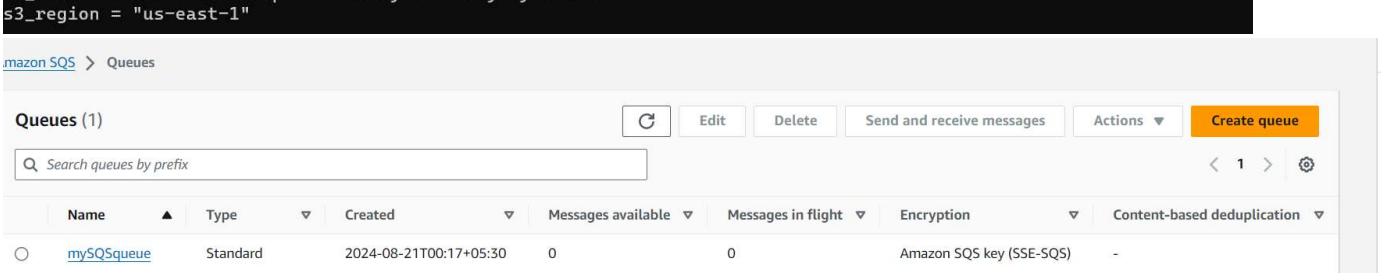
Do terraform apply and then do the following

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

```
C:\Terraform>terraform plan
random_pet.roma bucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly]
aws_s3_bucket.roma bucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly-bucket]

Terraform used the selected providers to generate the following execution plan. Resource actio
following symbols:
  + create

Terraform will perform the following actions:

  # aws_sqs_queue.myqueue will be created
  + resource "aws_sqs_queue" "myqueue" {
      + arn                               = (known after apply)
      + content_based_deduplication       = false
      + deduplication_scope               = (known after apply)
      + delay_seconds                     = 0
      + fifo_queue                        = false
      + fifo_throughput_limit             = (known after apply)
      + id                                = (known after apply)
      + kms_data_key_reuse_period_seconds = (known after apply)
      + max_message_size                  = 262144
      + message_retention_seconds         = 345600
      + name                              = "mySQSqueue"
      + name_prefix                       = (known after apply)
      + policy                            = (known after apply)
      + receive_wait_time_seconds         = 0
      + redrive_allow_policy              = (known after apply)
      + redrive_policy                    = (known after apply)
      + sqs_managed_sse_enabled           = (known after apply)
      + tags_all                          = (known after apply)
      + url                               = (known after apply)
      + visibility_timeout_seconds        = 30
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```

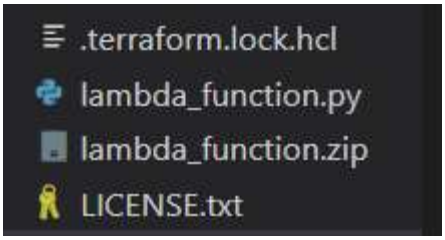
```
C:\Terraform>terraform apply
random_pet.roma_bucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly]
aws_s3_bucket.roma_bucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly-

Terraform used the selected providers to generate the following execution plan.
following symbols:
  + create

Terraform will perform the following actions:

  # aws_sqs_queue.myqueue will be created
  + resource "aws_sqs_queue" "myqueue" {
      + arn                               = (known after apply)
      + content_based_deduplication       = false
      + deduplication_scope               = (known after apply)
      + delay_seconds                     = 0
      + fifo_queue                        = false
      + fifo_throughput_limit             = (known after apply)
      + id                                = (known after apply)
      + kms_data_key_reuse_period_seconds = (known after apply)
      + max_message_size                  = 262144
      + message_retention_seconds         = 345600
      + name                              = "mySQSqueue"
      + name_prefix                       = (known after apply)
      + policy                            = (known after apply)
      + receive_wait_time_seconds         = 0
      + redrive_allow_policy              = (known after apply)
      + redrive_policy                    = (known after apply)
      + sqs_managed_sse_enabled           = (known after apply)
      + tags_all                          = (known after apply)
      + url                               = (known after apply)
      + visibility_timeout_seconds        = 30
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
```

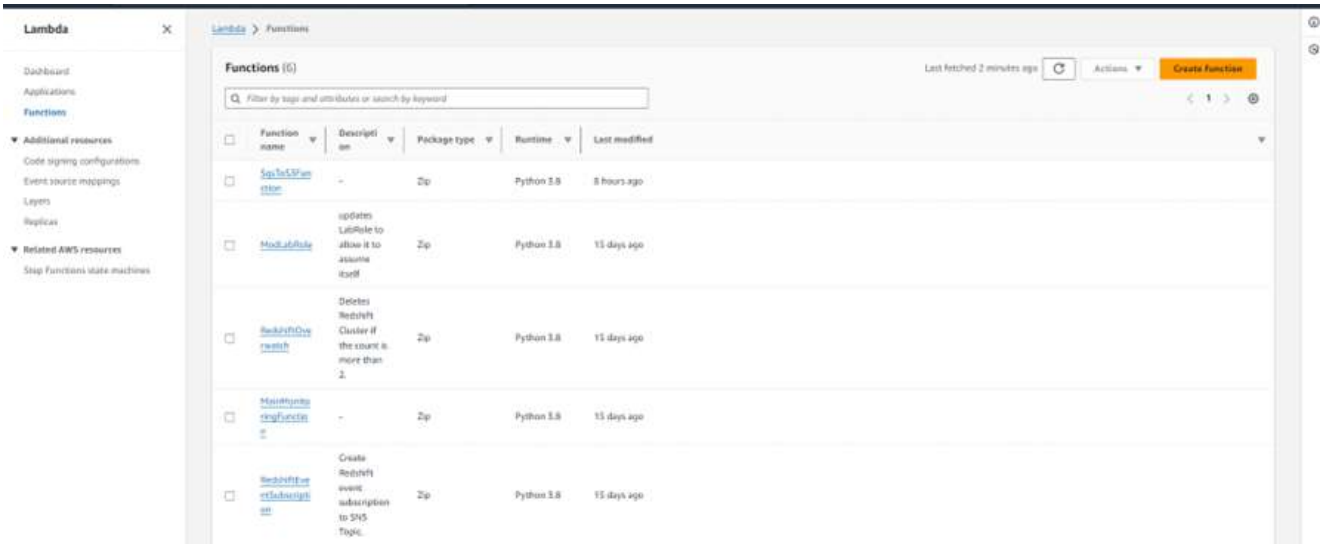
```
Plan: 1 to add, 0 to change, 0 to destroy.

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

aws_sqs_queue.myqueue: Creating...
aws_sqs_queue.myqueue: Still creating... [10s elapsed]
aws_sqs_queue.myqueue: Still creating... [20s elapsed]
aws_sqs_queue.myqueue: Creation complete after 28s [id=https://sqs.us-east-1.amazonaws.com/659038782739/mySQSqueue]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

s3_arn = "arn:aws:s3:::fdp-mistakenly-vast-mayfly-bucket"
s3_region = "us-east-1"
```

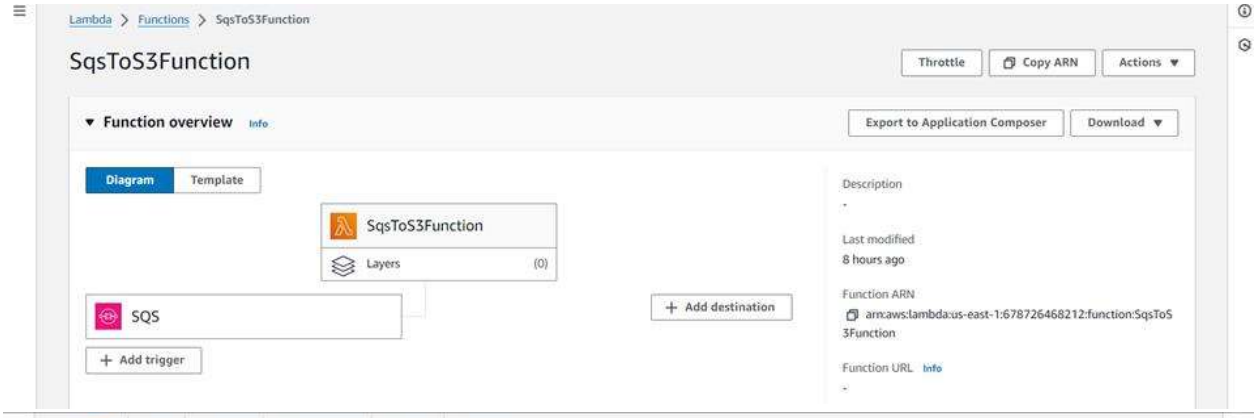


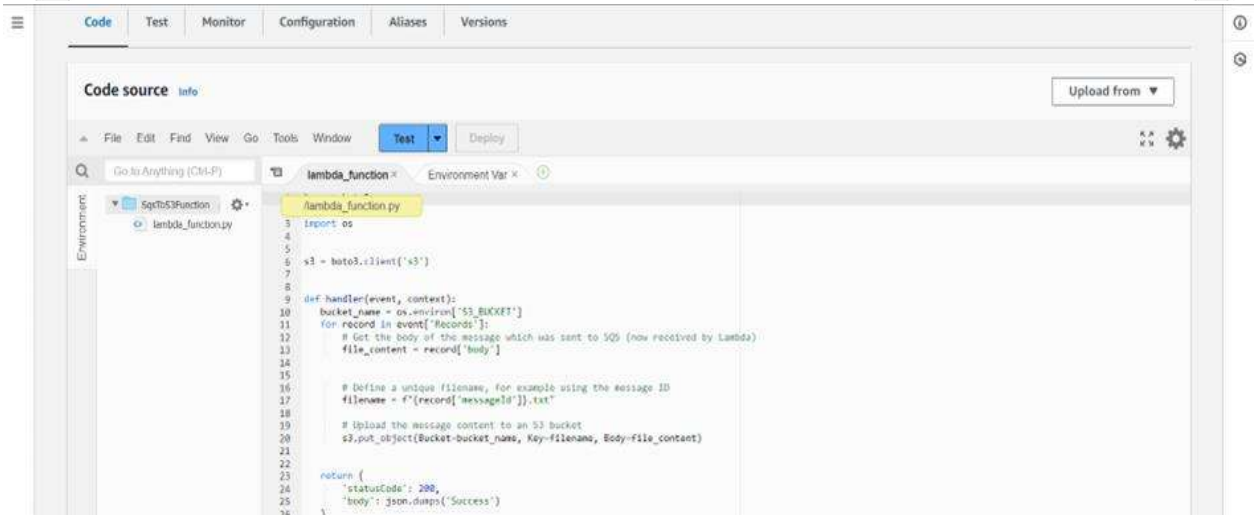
```
data "archive_file" "zip" {
  type        = "zip"
  source_file = "lambda_function.py"
  output_path = "lambda_function.zip"
}
```

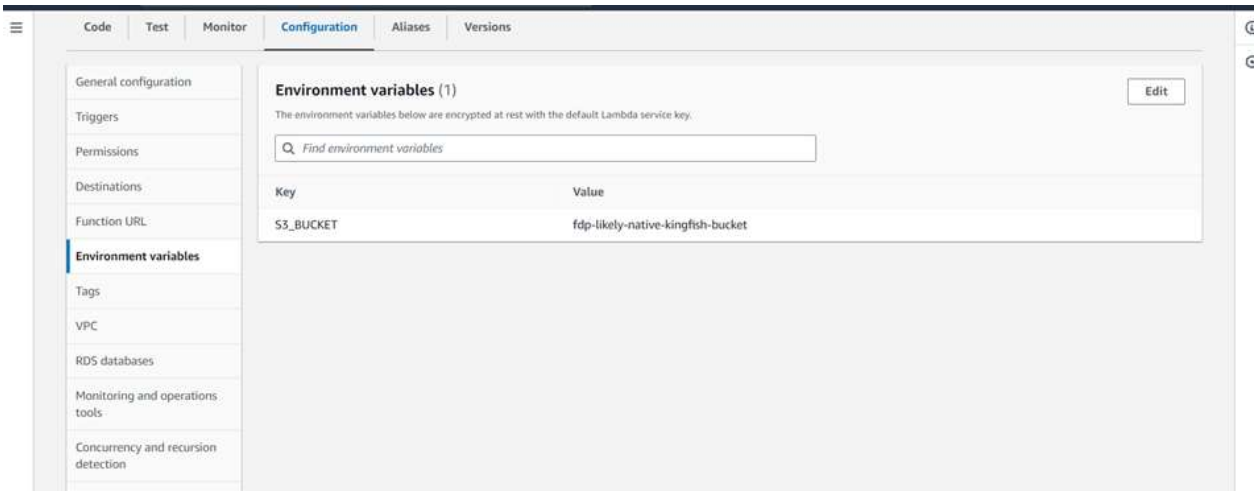
```
C:\Terraform>terraform apply
random_pet.roma bucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly]
data.archive_file.zip: Reading...
data.archive_file.zip: Read complete after 1s [id=8bf3069f9c57671063f5ad7ee3fbe97f62f9460c]
aws_sqs_queue.myqueue: Refreshing state... [id=https://sqs.us-east-1.amazonaws.com/659038782739/mySQSqueue]
aws_s3_bucket.roma bucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly-bucket]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are
needed.

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

Outputs:

s3_arn = "arn:aws:s3:::fdp-mistakenly-vast-mayfly-bucket"
s3_region = "us-east-1"
```

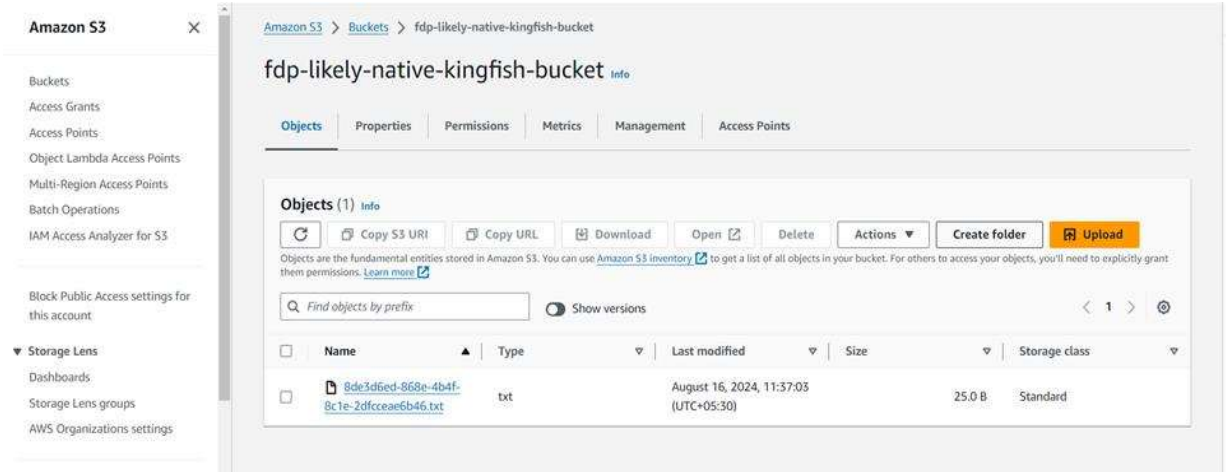


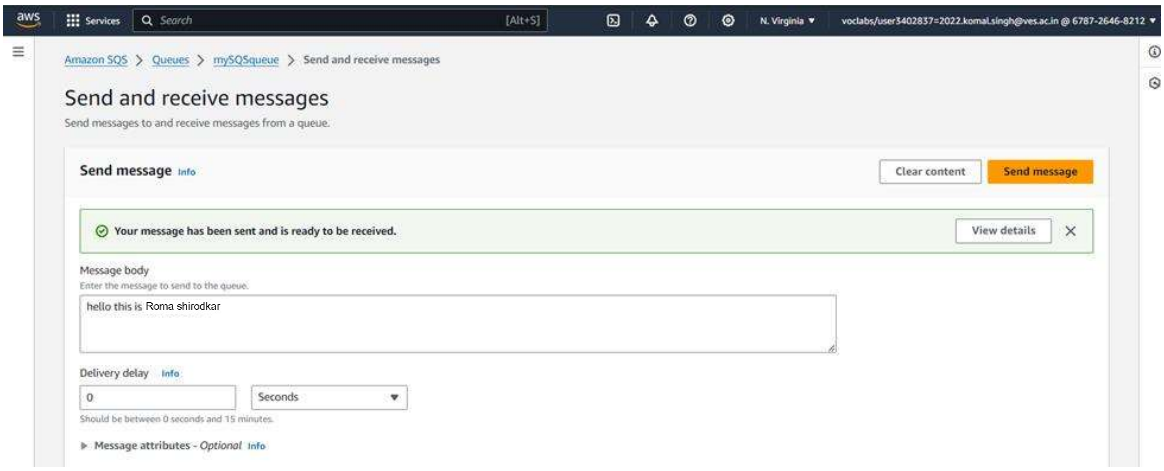



Lambda > Functions > SqsToS3Function
SqsToS3Function
Throttle
Copy ARN
Actions
Function overview Info
Export to Application Composer
Download
Diagram
Template
SqsToS3Function
Layers (0)
SQS
Add trigger
Add destination
Description
-
Last modified
8 hours ago
Function ARN
arn:aws:lambda:us-east-1:678726468212:function:SqsToS3Function
Function URL Info
-


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)
lambda_function
Environment Var
SqsToS3Function
lambda_function.py
/lambda_function.py
import os
s3 = boto3.client('s3')
def handler(event, context):
bucket_name = os.environ['S3_BUCKET']
for record in event['Records']:
# Get the body of the message which was sent to SQS (now received by Lambda)
file_content = record['body']
# Define a unique filename, for example using the message ID
filename = f"{record['messageId']}.txt"
# Upload the message content to an S3 bucket
s3.put_object(Bucket=bucket_name, Key=filename, Body=file_content)
return {
'statusCode': 200,
'body': json.dumps('Success')

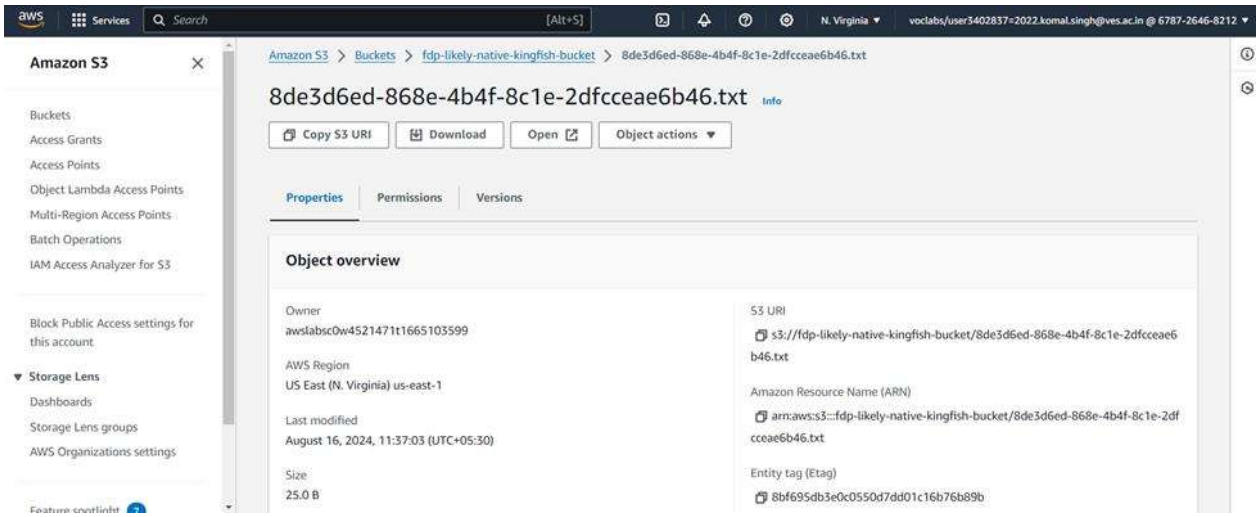


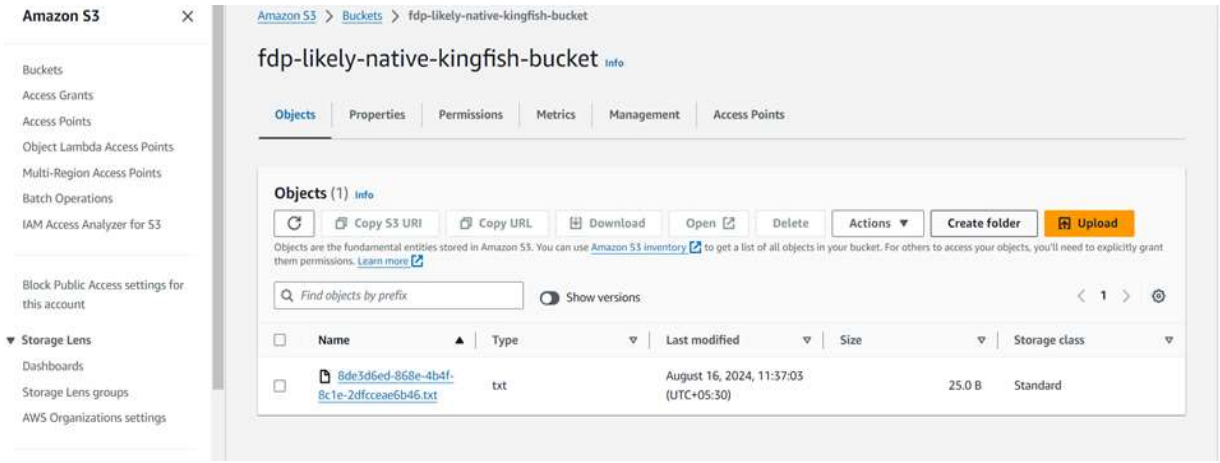


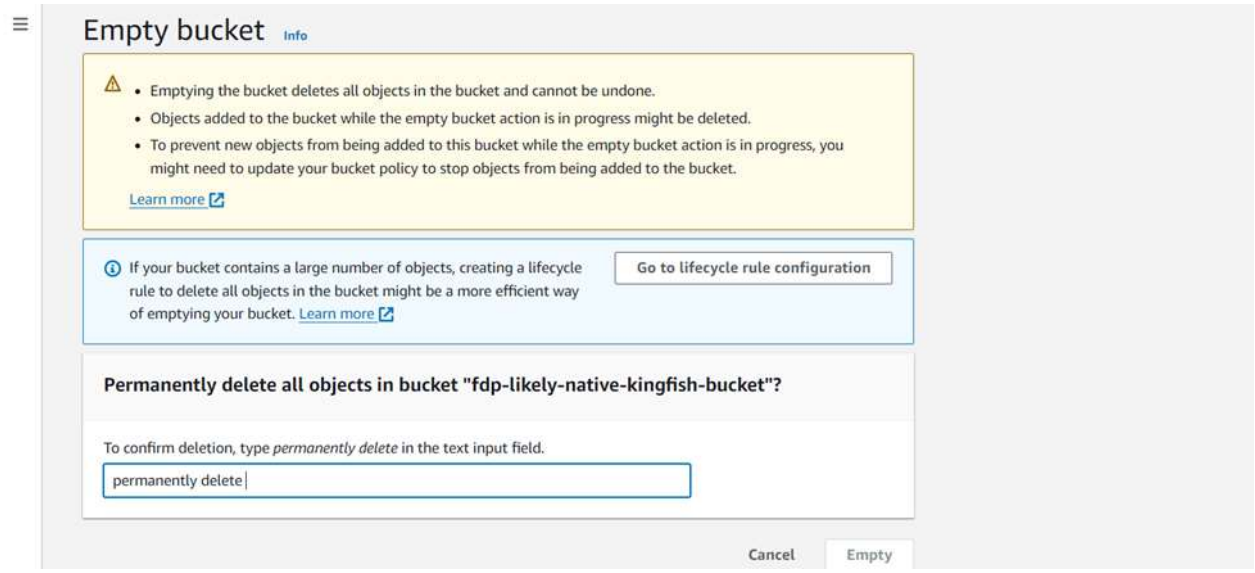


Add S3 Event Notification for Lambda Trigger

- Define a bucket notification to trigger the Lambda function when objects are uploaded to the S3 bucket.


Services
Search
[Alt+S]
N. Virginia
Amazon S3
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
AWS Organizations settings
Amazon S3 > Buckets > fdp-likely-native-kingfish-bucket > 8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt
8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt Info
Copy S3 URI
Download
Open
Object actions
Properties
Permissions
Versions
Object overview
Owner
awslabsc0w4521471t1665103599
AWS Region
US East (N. Virginia) us-east-1
Last modified
August 16, 2024, 11:37:03 (UTC+05:30)
Size
25.0 B
S3 URI
s3://fdp-likely-native-kingfish-bucket/8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt
Amazon Resource Name (ARN)
arn:aws:s3:::fdp-likely-native-kingfish-bucket/8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt
Entity tag (Etag)
8bf695db3e0c0550d7dd01c16b76b89b


Amazon S3
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
AWS Organizations settings
Amazon S3 > Buckets > fdp-likely-native-kingfish-bucket
fdp-likely-native-kingfish-bucket Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (1) 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
Show versions
Name
Type
Last modified
Size
Storage class
8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt
txt
August 16, 2024, 11:37:03 (UTC+05:30)
25.0 B
Standard



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

```
Microsoft Windows [Version 10.0.22000.2057]
(c) Microsoft Corporation. All rights reserved.

C:\Terraform>terraform destroy
random_pet.roma bucket717: Refreshing state... [id=fdp-likely-native-kingfish]
data.archive_file.zip: Reading...
data.archive_file.zip: Read complete after 0s [id=93c92209eafac774599673c33c7e7636e68e60e8]
aws_sqs_queue.myqueue: Refreshing state... [id=https://sqs.us-east-1.amazonaws.com/678726468212/mySQSqueue]
aws_s3_bucket.roma bucket717: Refreshing state... [id=fdp-likely-native-kingfish-bucket]
aws_lambda_event_source_mapping.SqsToLambda: Refreshing state... [id=4582b6aa-1865-4866-86eb-1062635c21a7]
aws_lambda_function.mykomlambda: Refreshing state... [id=SqsToS3Function]

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" {
      - batch_size                       = 1 -> null
      - bisect_batch_on_function_error   = false -> null
      - enabled                          = true -> null
      - event_source_arn                 = "arn:aws:sqs:us-east-1:678726468212:mySQSqueue" -> null
      - function_arn                     = "arn:aws:lambda:us-east-1:678726468212:function:SqsToS3Function" -> null
      - function_name                    = "arn:aws:lambda:us-east-1:678726468212:function:SqsToS3Function" -> null
      - function_response_types          = [] -> null
      - id                               = "4582b6aa-1865-4866-86eb-1062635c21a7" -> null
      - last_modified                    = "2024-08-16T06:50:00Z" -> null
      - maximum_batching_window_in_seconds = 0 -> null
      - maximum_record_age_in_seconds    = 0 -> null
      - maximum_retry_attempts           = 0 -> null
      - parallelization_factor           = 0 -> null
      - queues                           = [] -> null
      - state                            = "Enabled" -> null
      - state_transition_reason          = "USER_INITIATED" -> null
      - topics                           = [] -> null
      - tumbling_window_in_seconds       = 0 -> null
      - uuid                             = "4582b6aa-1865-4866-86eb-1062635c21a7" -> null
```