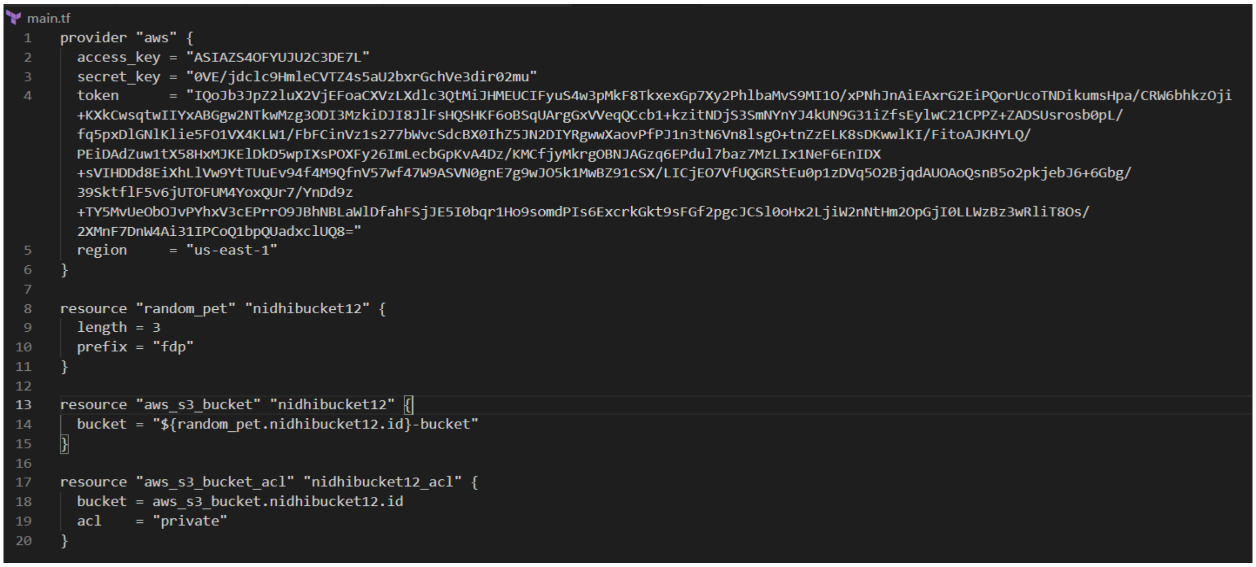
Steps:

Set Up Your Terraform Environment

- **Install Terraform**: If you haven't already installed Terraform, download it from Terraform's official site and follow the installation steps.
- **Configure AWS CLI**: Make sure the AWS CLI is installed and configured with the necessary permissions.

Create a Terraform Project

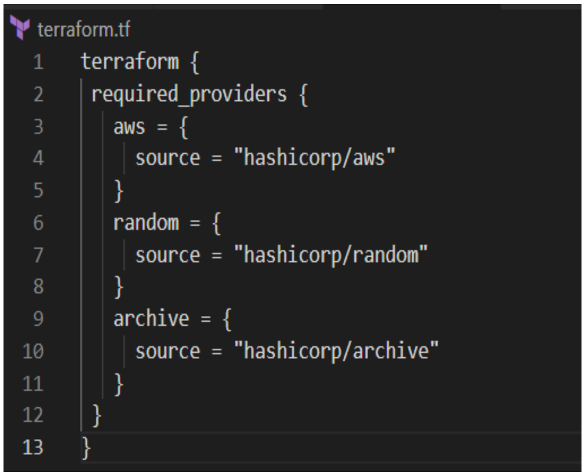
- Create a directory for your project and navigate to it



```
provider "aws" {
  access_key = "ASIAZS4OFYUJU2C3DE7L"
  secret_key = "0VE/jdclc9HmleCVTZ4s5aU2bxrGchVe3dir02mu"
  token      = "IQoJb3JpZ2luX2VjEFoaCXVzLXdlc3QtMiJHMEUCIFyuS4w3pMkF8TkxexGp7Xy2PhlbaMvS9MI1O/xPNhJnAiEAxrG2EiPQorUcoTNDikumsHpa/CRW6bhkzOji
  +KXkCwsqtwIIYxABGgw2NTkwMzg3ODI3MzkiDJI8JlFsHQSHKF6oBSqUArgGxVVeqQCcb1+kzitNDjS3SmNYnYJ4kUN9G31iZfsEylwC21CPPZ+ZADSUsrosb0pL/
  fq5pxDlGNlKlie5FO1VX4KLW1/FbFCinVz1s277bWvcSdcBX0IhZ5JN2DIYRgwwXaovPfPJ1n3tN6Vn8lsgO+tnZzELK8sDKwwlKI/FitoAJKHYLQ/
  PEiDAdZuw1tX58HxMJKElDkD5wpIXsPOXFy26ImLecbGpKvA4Dz/KMCfjyMkrgOBNJAGzq6EPdul7baz7MzLIx1NeF6EnIDX
  +sVIHDDd8EiXhLlVw9YtTUuEv94f4M9QfnV57wf47W9ASVN0gnE7g9wJO5k1MwBZ91cSX/LICjEO7VfUQGRStEu0p1zDVq5O2BjqdAUOAoQsnB5o2pkjebJ6+6Gbg/
  39SktflF5v6jUTOFUM4YoxQUr7/YnDd9z
  +TY5MvUeObOJvPYhxV3cEPrrO9JBhNBLaWlDfahFSjJE5I0bqr1Ho9somdPIs6ExcrkGkt9sFGf2pgcJCSl0oHx2LjiW2nNtHm2OpGjI0LLWzBz3wRliT8Os/
  2XMnF7DnW4Ai31IPCoQ1bpQUadxclUQ8="
  region     = "us-east-1"
}

resource "random_pet" "nidhibucket12" {
  length = 3
  prefix = "fdp"
}

resource "aws_s3_bucket" "nidhibucket12" {
  bucket = "${random_pet.nidhibucket12.id}-bucket"
}

resource "aws_s3_bucket_acl" "nidhibucket12_acl" {
  bucket = aws_s3_bucket.nidhibucket12.id
  acl    = "private"
}
```

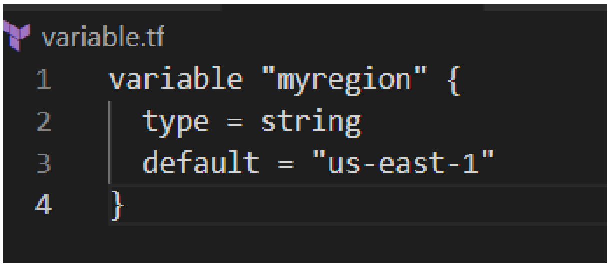
Define the Provider

- In `main.tf`, start by defining the provider (AWS)



terraform.tf

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

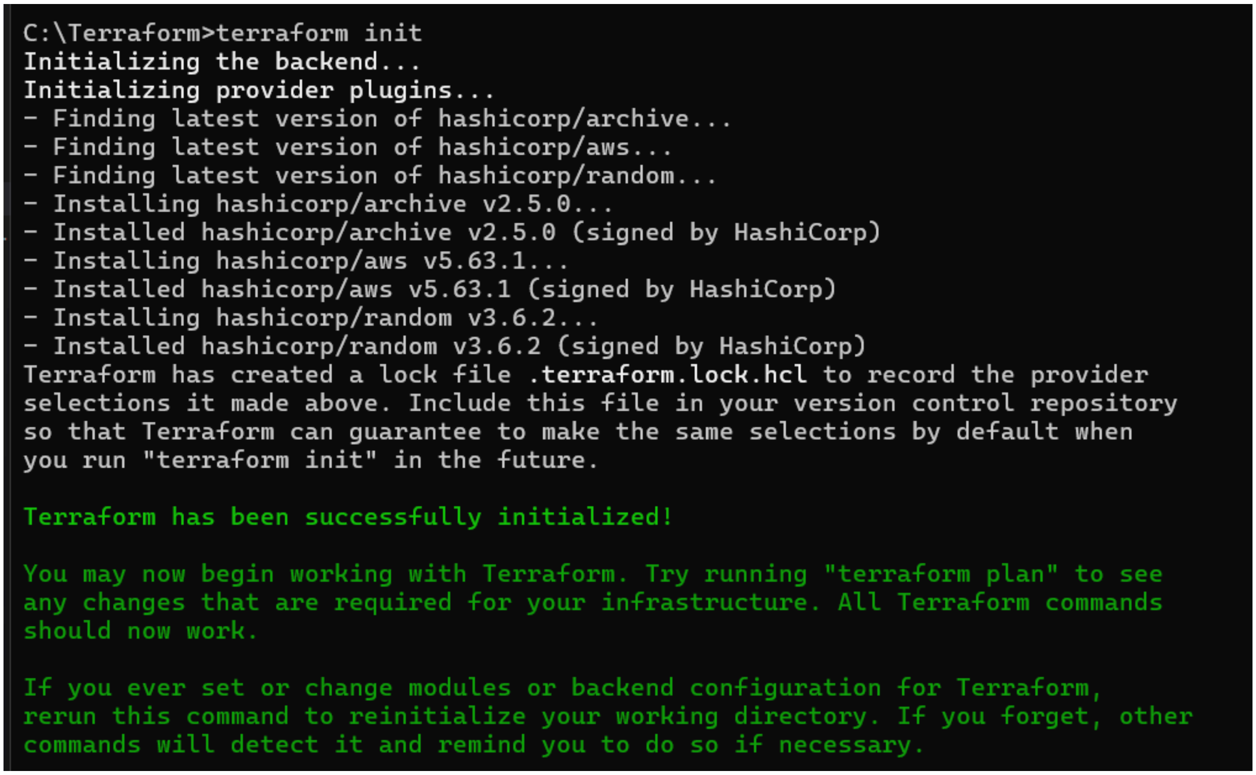


variable.tf

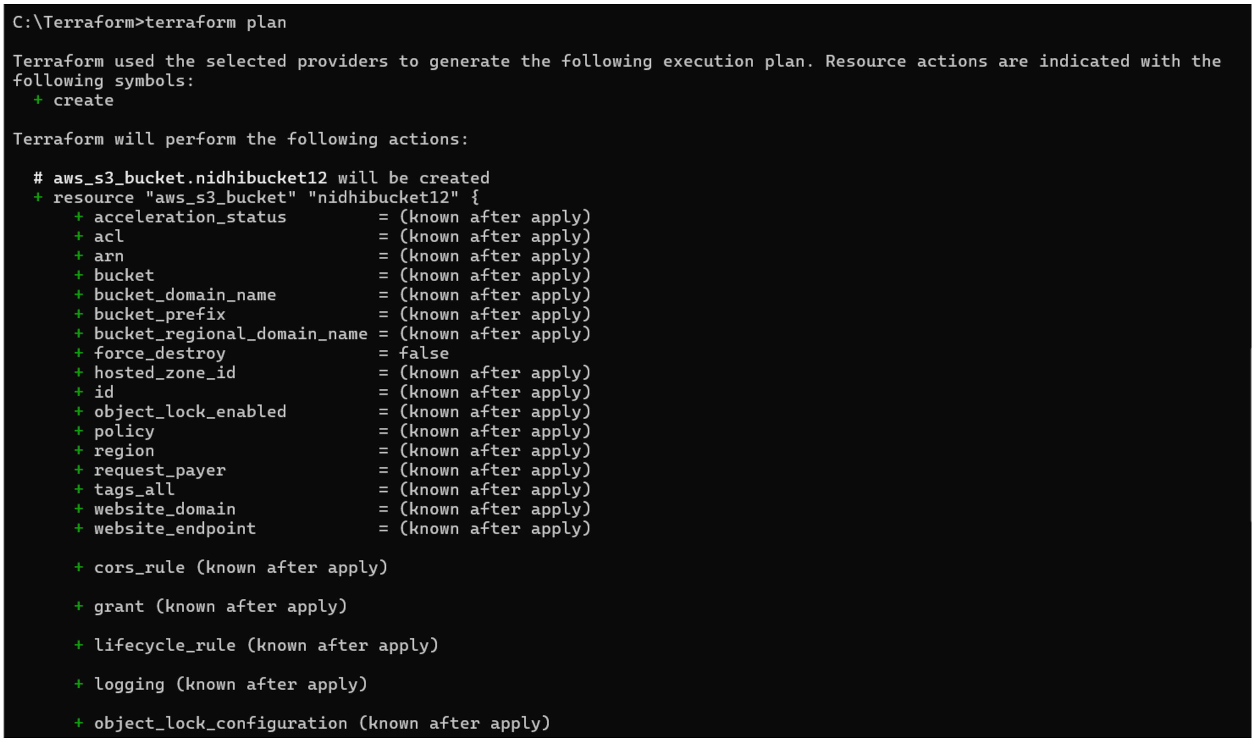
```
variable "myregion" {
  type = string
  default = "us-east-1"
}
```

Create an S3 Bucket

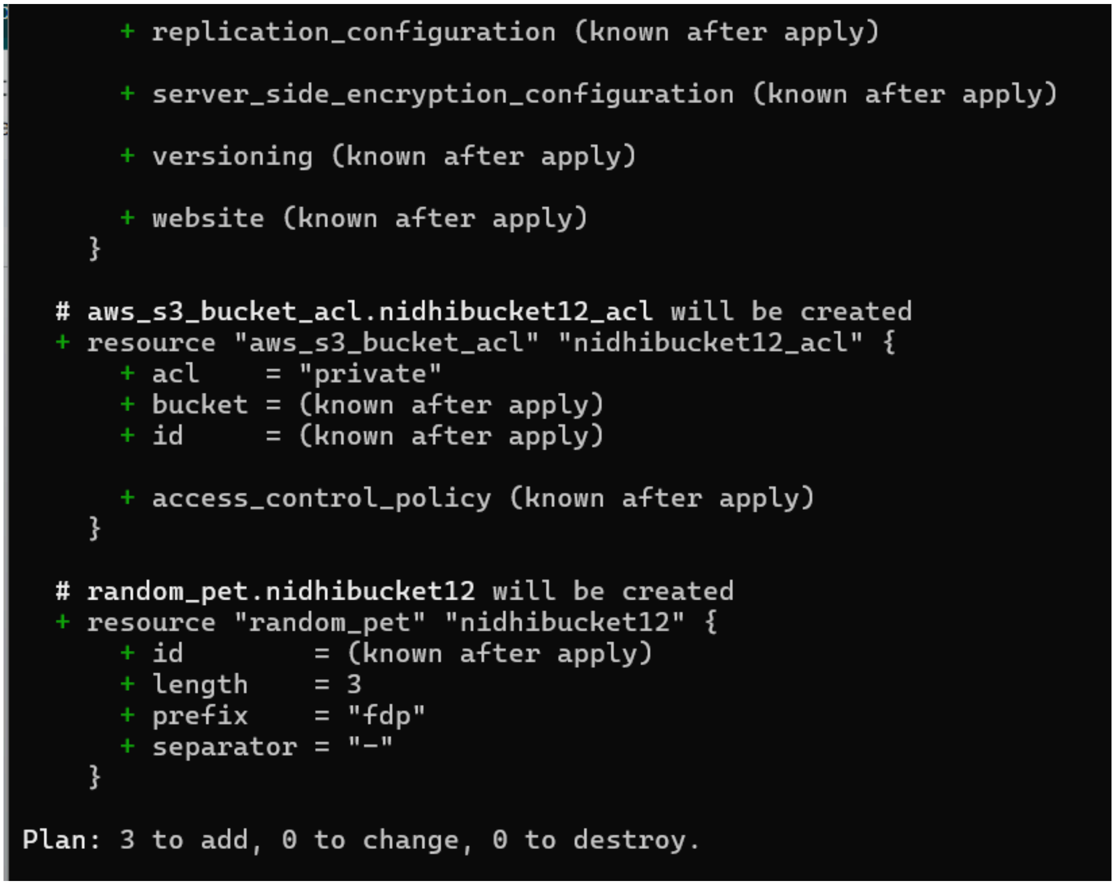
- Next, define an S3 bucket where Lambda can receive file uploads.



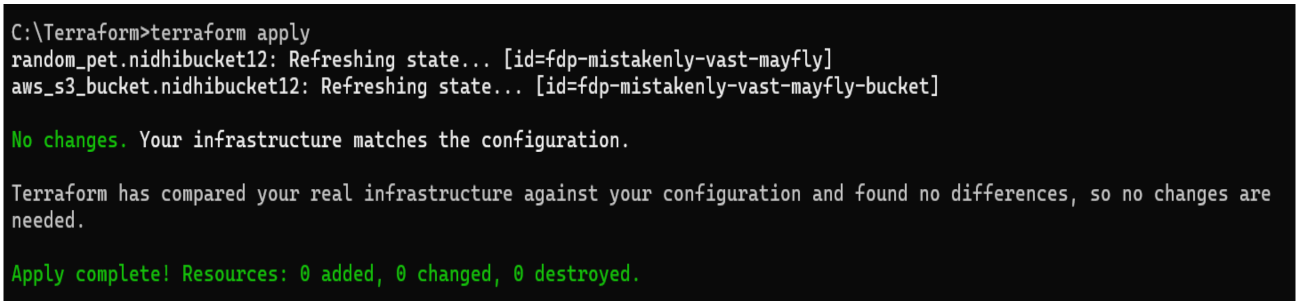
```
C:\Terraform>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/archive...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/random...
- Installing hashicorp/archive v2.5.0...
- Installed hashicorp/archive 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.
```



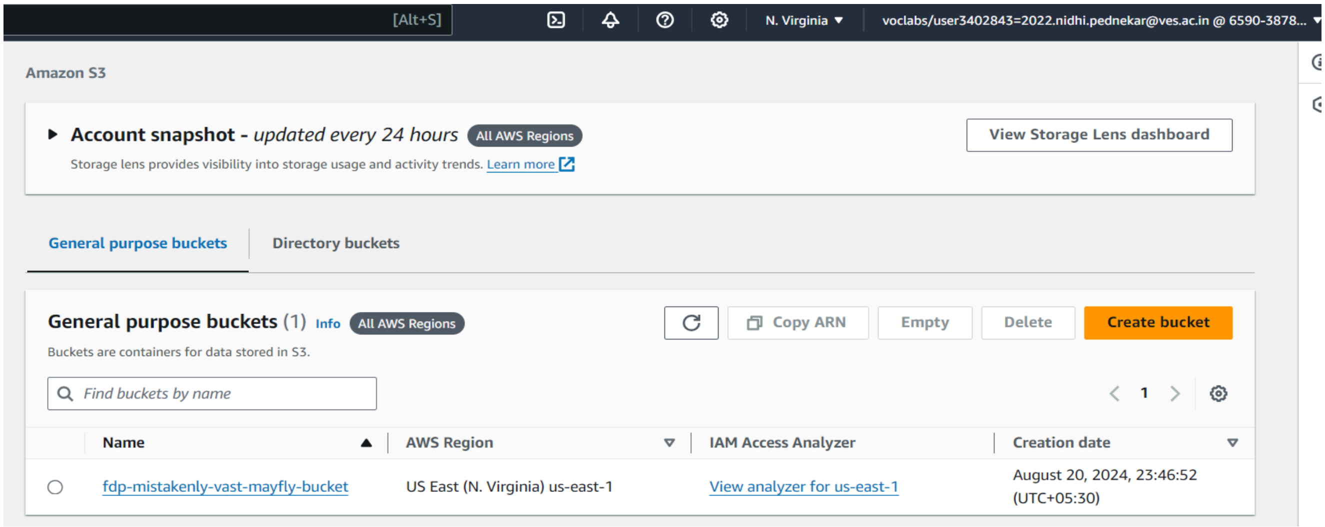
```
C:\Terraform>terraform plan

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.nidhibucket12 will be created
  + resource "aws_s3_bucket" "nidhibucket12" {
      + 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.nidhibucket12_acl will be created
  + resource "aws_s3_bucket_acl" "nidhibucket12_acl" {
      + acl    = "private"
      + bucket = (known after apply)
      + id     = (known after apply)

      + access_control_policy (known after apply)
    }

  # random_pet.nidhibucket12 will be created
  + resource "random_pet" "nidhibucket12" {
      + id        = (known after apply)
      + length    = 3
      + prefix    = "fdp"
      + separator = "-"
    }

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



```
C:\Terraform>terraform apply
random_pet.nidhibucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly]
aws_s3_bucket.nidhibucket12: 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.
```



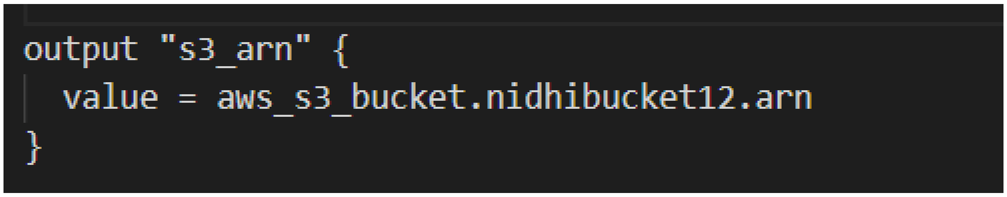
▸ **Account snapshot -** *updated every 24 hours* All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Learn more

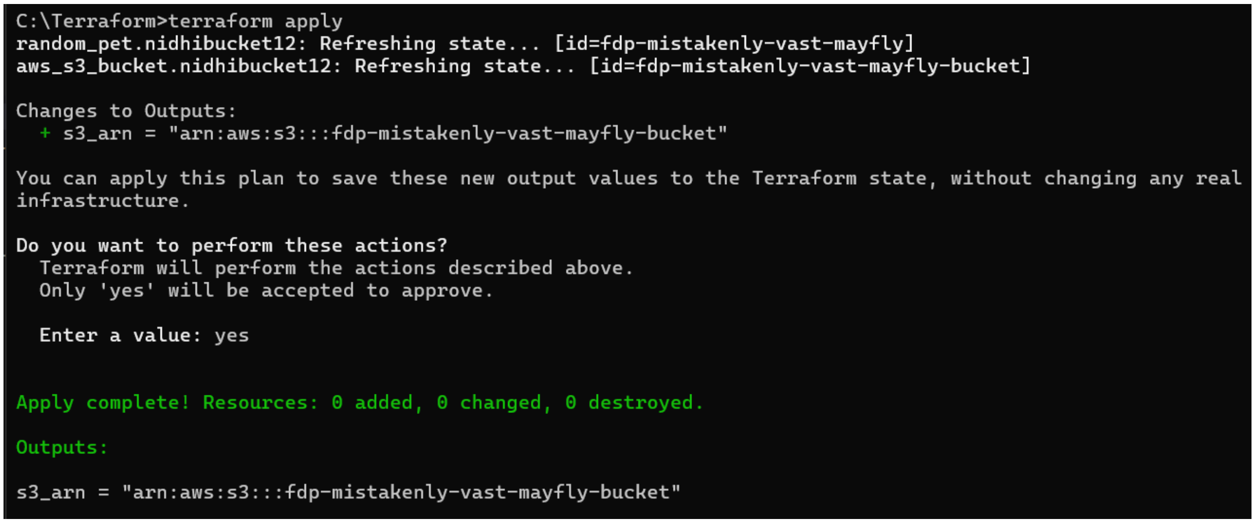
General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

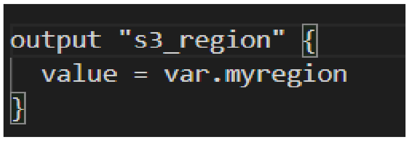
	Name ▲	AWS Region ▽	IAM Access Analyzer	Creation date ▽
○	fdp-mistakenly-vast-mayfly-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 20, 2024, 23:46:52 (UTC+05:30)



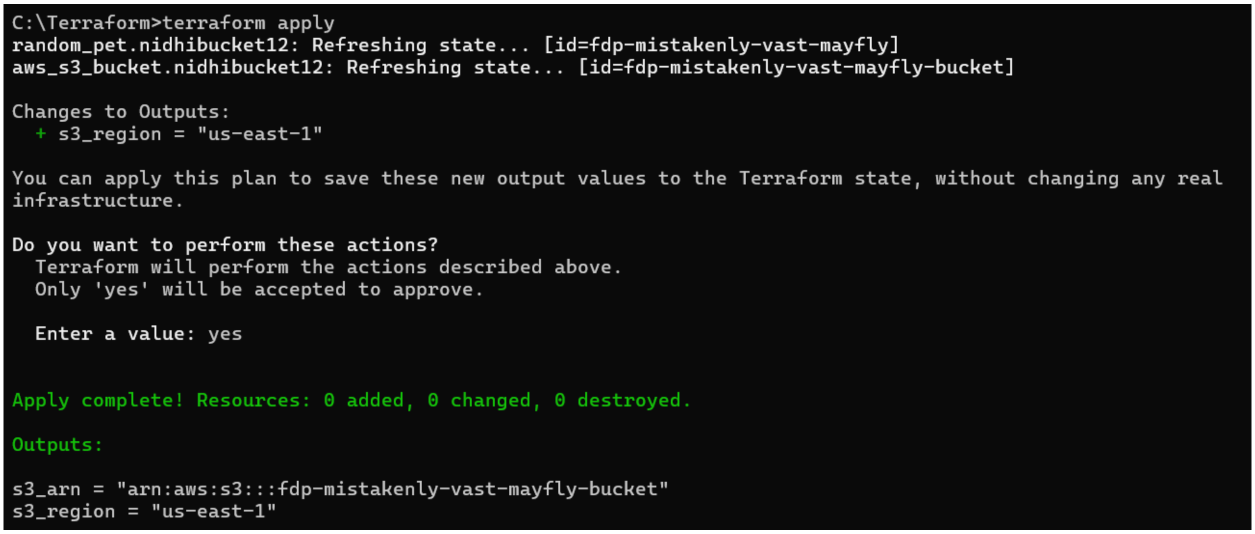
```
output "s3_arn" {
  value = aws_s3_bucket.nidhibucket12.arn
}
```



```
C:\Terraform>terraform apply
random_pet.nidhibucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly]
aws_s3_bucket.nidhibucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly-bucket]

Changes to Outputs:
  + s3_arn = "arn:aws:s3:::fdp-mistakenly-vast-mayfly-bucket"

You can apply this plan to save these new output values to the Terraform state, without changing any real
infrastructure.

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


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

Outputs:

s3_arn = "arn:aws:s3:::fdp-mistakenly-vast-mayfly-bucket"
```



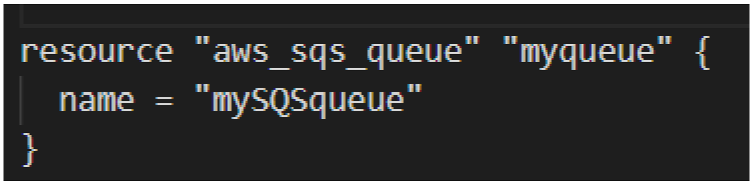
```
output "s3_region" {
  value = var.myregion
}
```



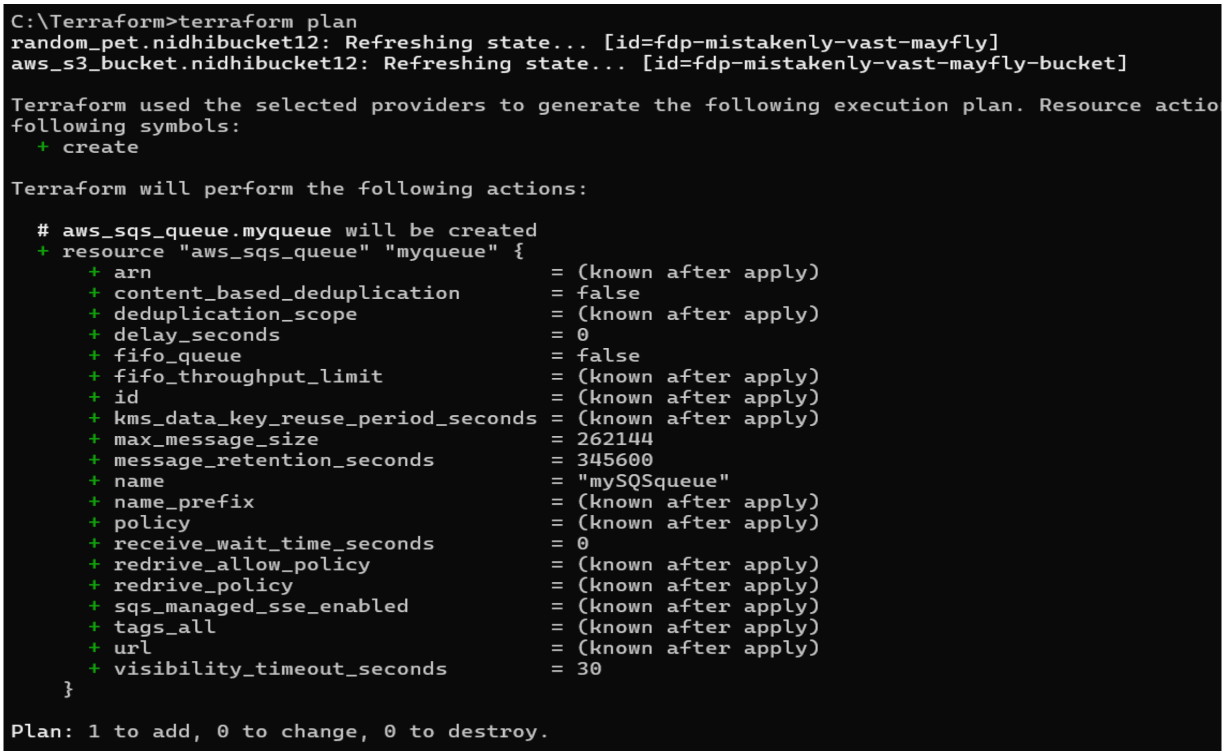
```
C:\Terraform>terraform apply
random_pet.nidhibucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly]
aws_s3_bucket.nidhibucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly-bucket]

Changes to Outputs:
  + s3_region = "us-east-1"

You can apply this plan to save these new output values to the Terraform state, without changing any real
infrastructure.

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


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"
```

Create an SQS Queue

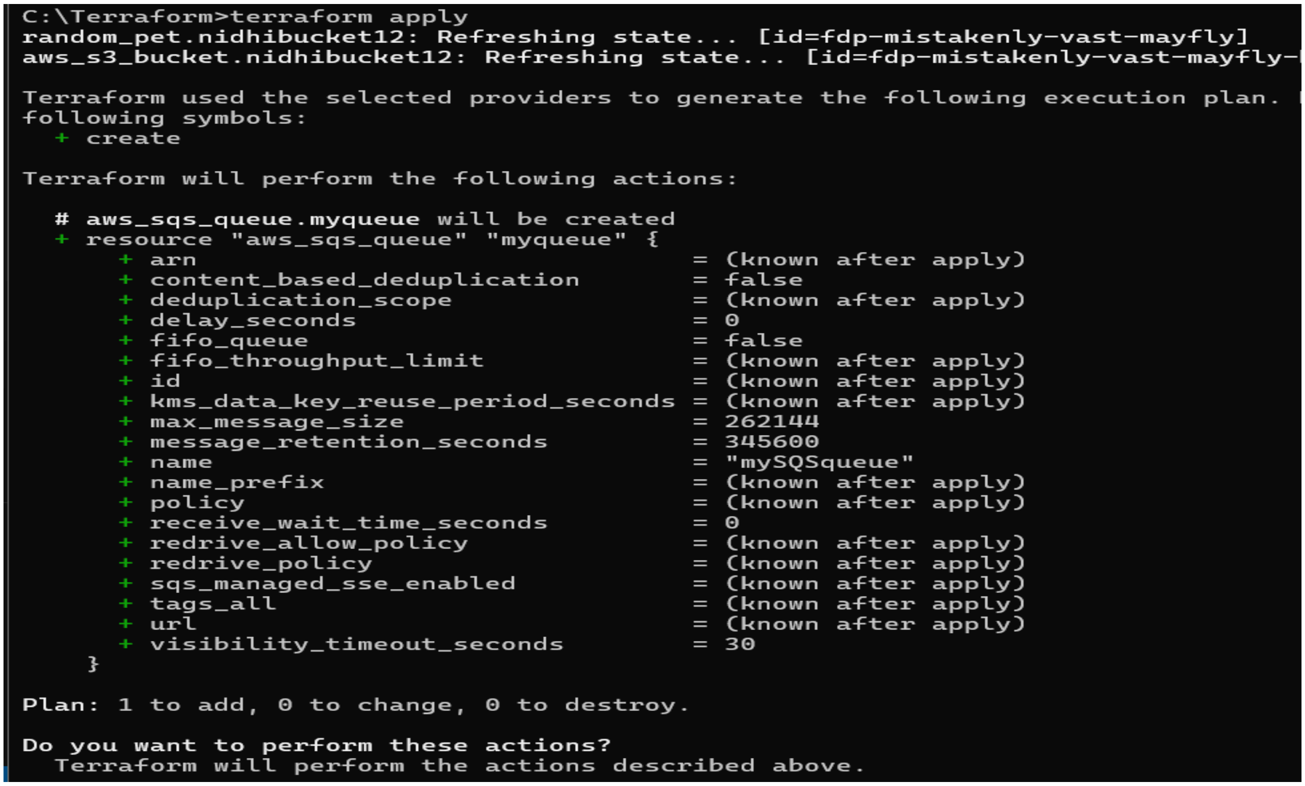
- Add an SQS queue that will receive events from the S3 bucket when a new object is uploaded.



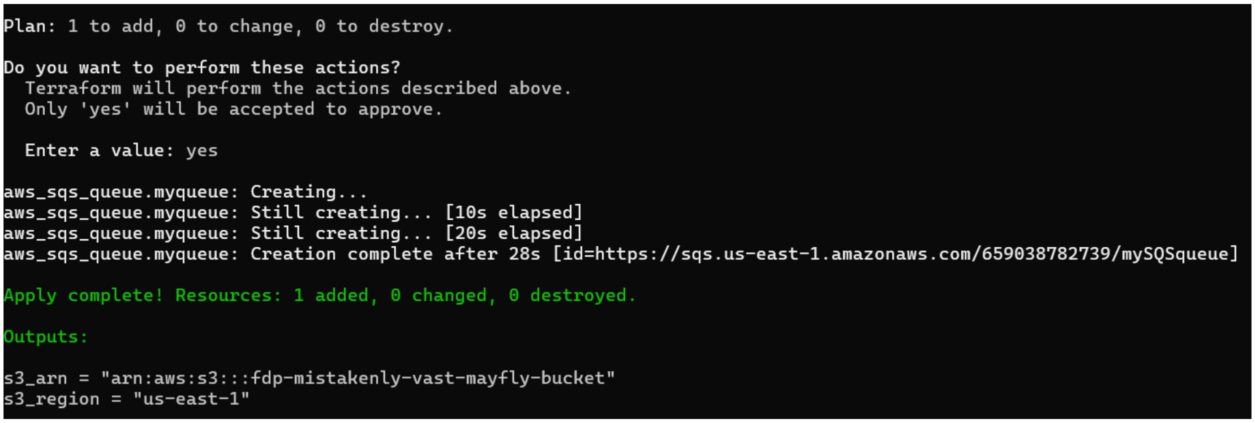
```
resource "aws_sqs_queue" "myqueue" {
  name = "mySQSqueue"
}
```



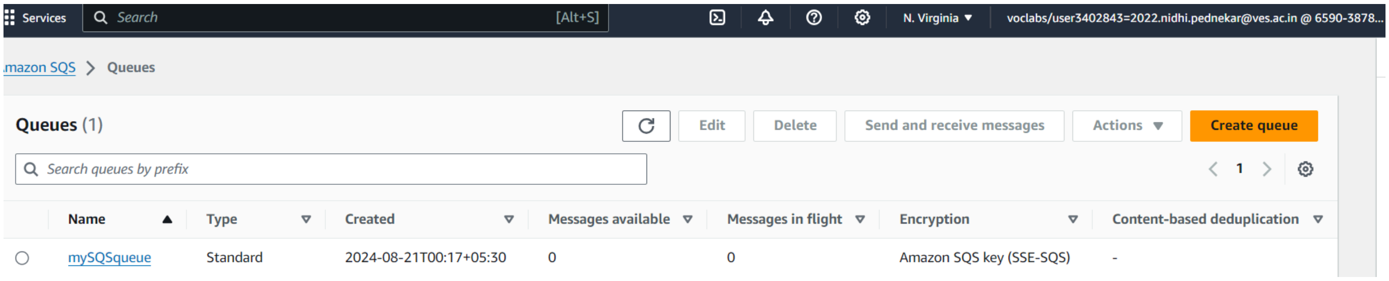
```
C:\Terraform>terraform plan
random_pet.nidhibucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly]
aws_s3_bucket.nidhibucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly-bucket]

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.nidhibucket12: Refreshing state... [id=fdp-mistakenly-vast-mayfly]
aws_s3_bucket.nidhibucket12: 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.
```

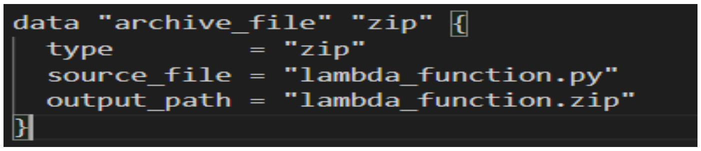


```
Plan: 1 to add, 0 to change, 0 to destroy.

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"
```

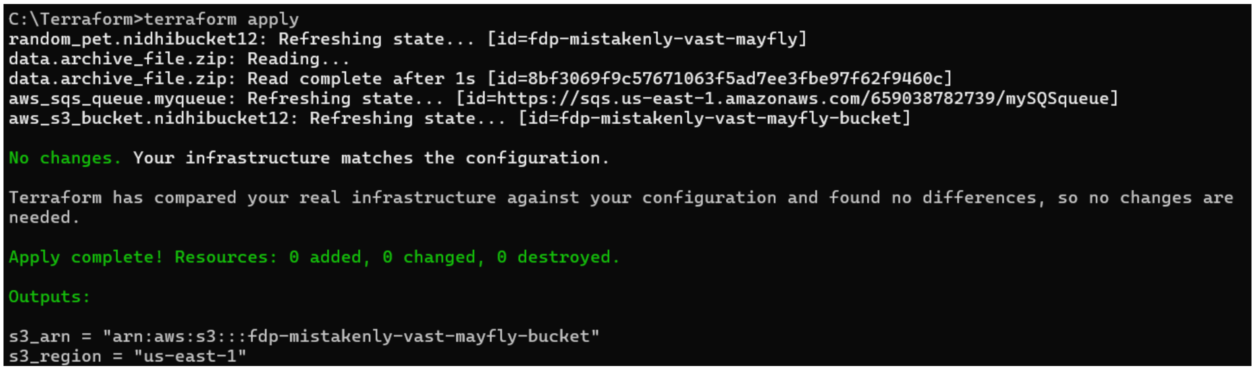


Queues (1)

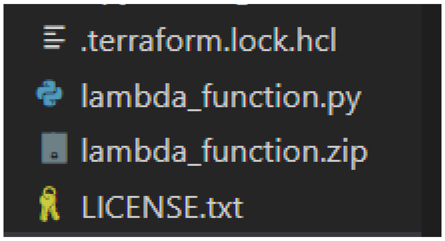
	Name ▲	Type	Created	Messages available	Messages in flight	Encryption	Content-based deduplication
○	mySQSqueue	Standard	2024-08-21T00:17+05:30	0	0	Amazon SQS key (SSE-SQS)	-



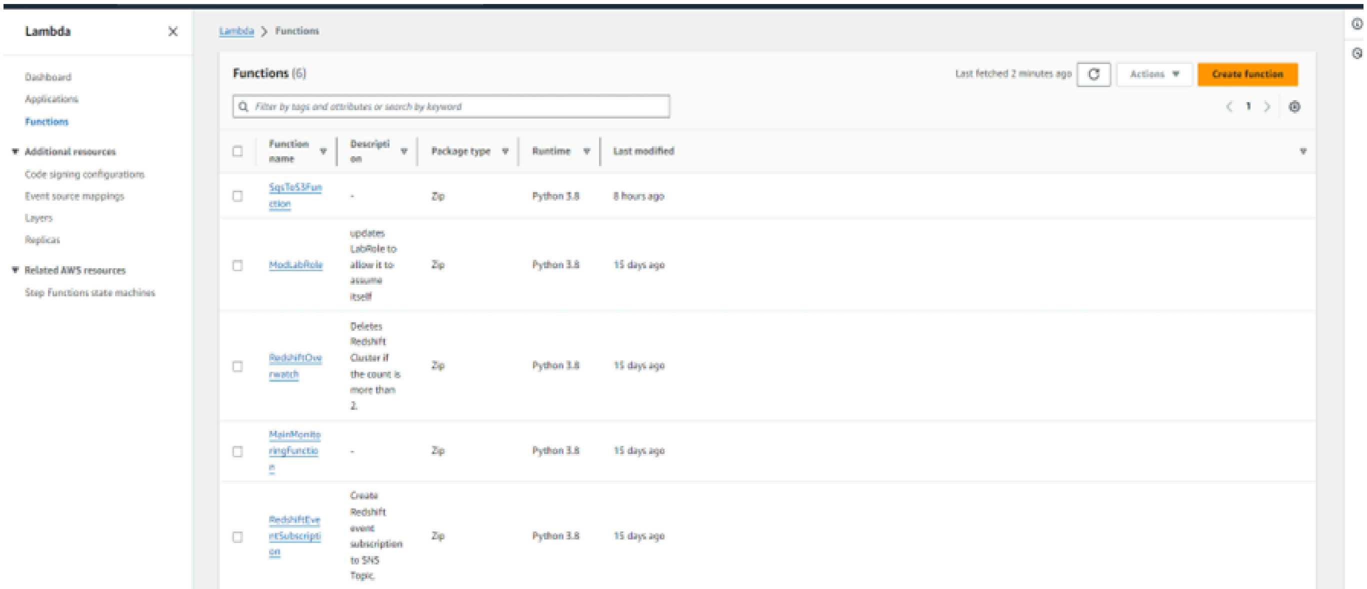
```
data "archive_file" "zip" {
  type        = "zip"
  source_file = "lambda_function.py"
  output_path = "lambda_function.zip"
}
```



```
C:\Terraform>terraform apply
random_pet.nidhibucket12: 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.nidhibucket12: 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"
```



```
.terraform.lock.hcl
lambda_function.py
lambda_function.zip
LICENSE.txt
```



Functions (6)

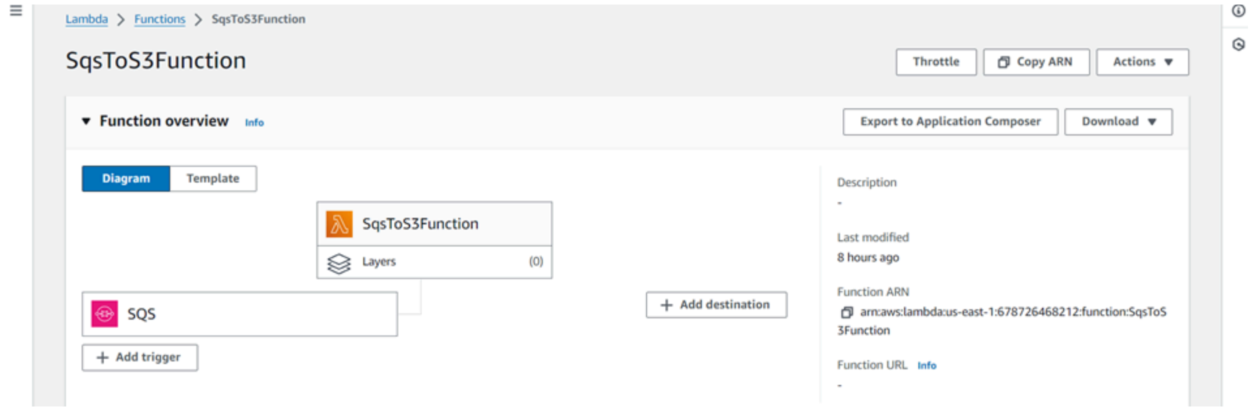
	Function name	Description	Package type	Runtime	Last modified
□	SqsToS3Function	-	Zip	Python 3.8	8 hours ago
□	ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.8	15 days ago
□	RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	15 days ago
□	MainMonitoringFunction	-	Zip	Python 3.8	15 days ago
□	RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.8	15 days ago

Create a Lambda Function

- Define the Lambda function that will be triggered by S3 events and send messages to SQS. You need a ZIP file containing your Lambda function code, which you can upload to the S3 bucket.

Create IAM Role for Lambda

- Lambda needs permissions to read from S3 and send messages to SQS. Define an IAM role and policy for the Lambda function.

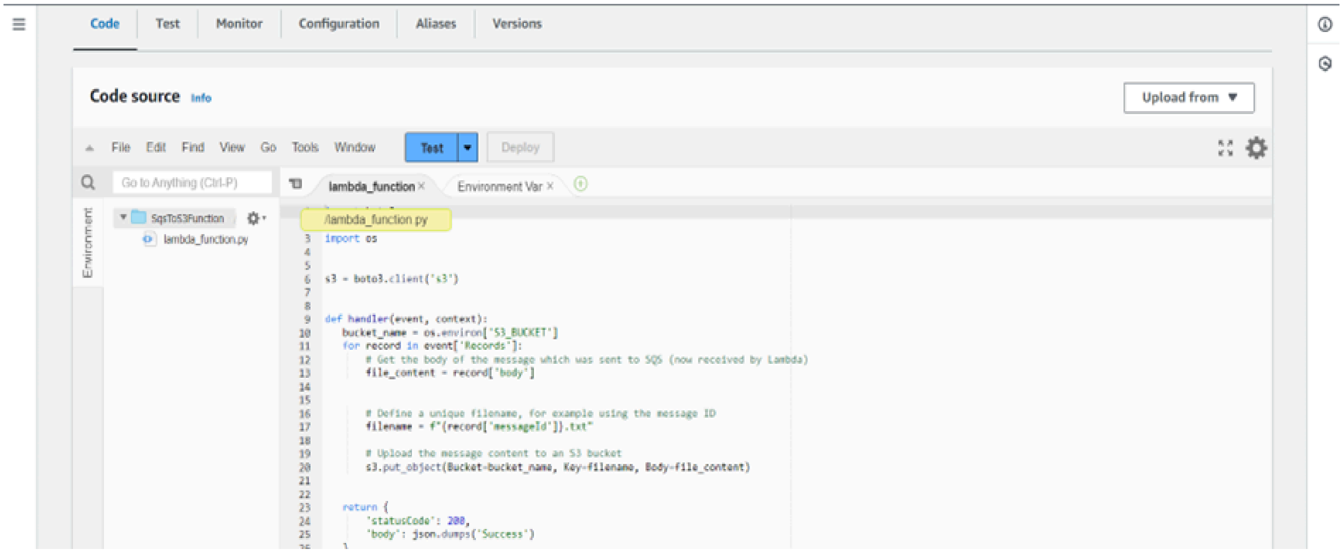


Description
-

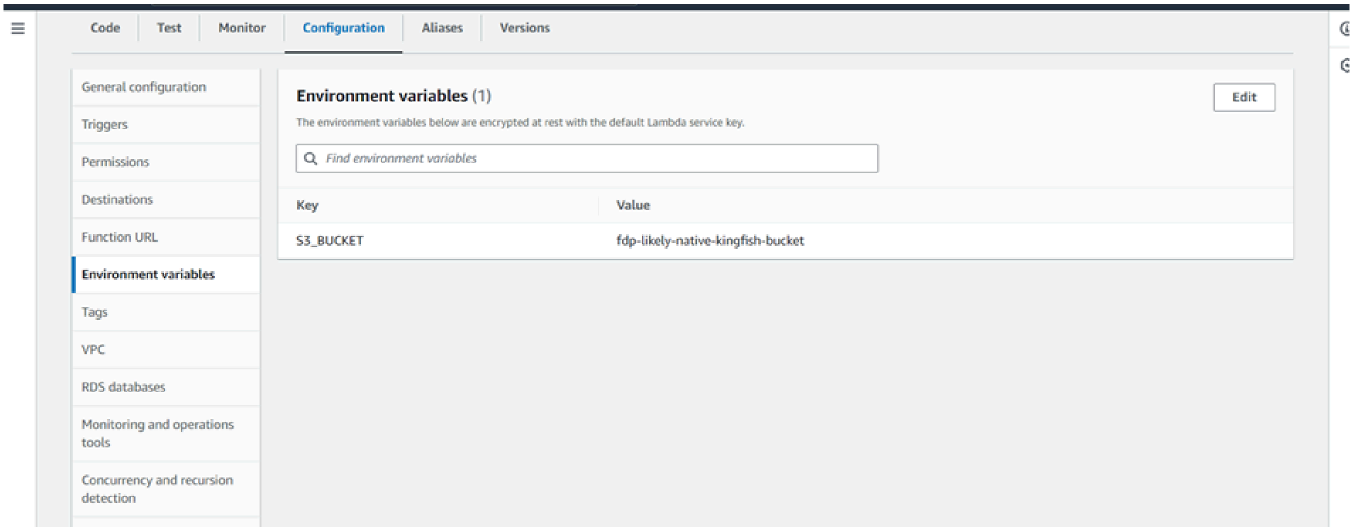
Last modified
8 hours ago

Function ARN
arn:aws:lambda:us-east-1:678726468212:function:SqsToS3Function

Function URL Info
-



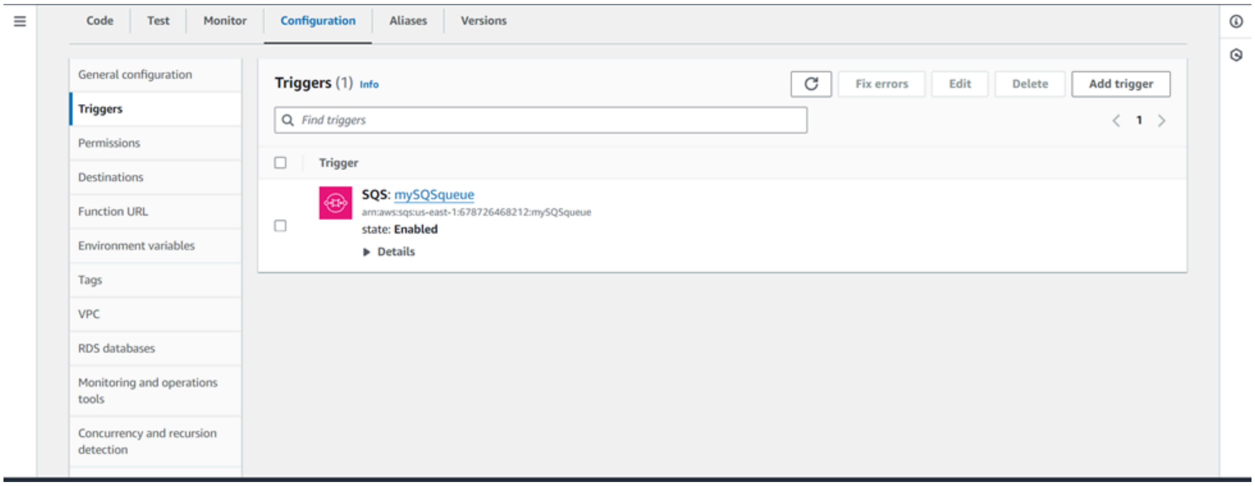
```python
import os


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')
```



Environment variables (1)

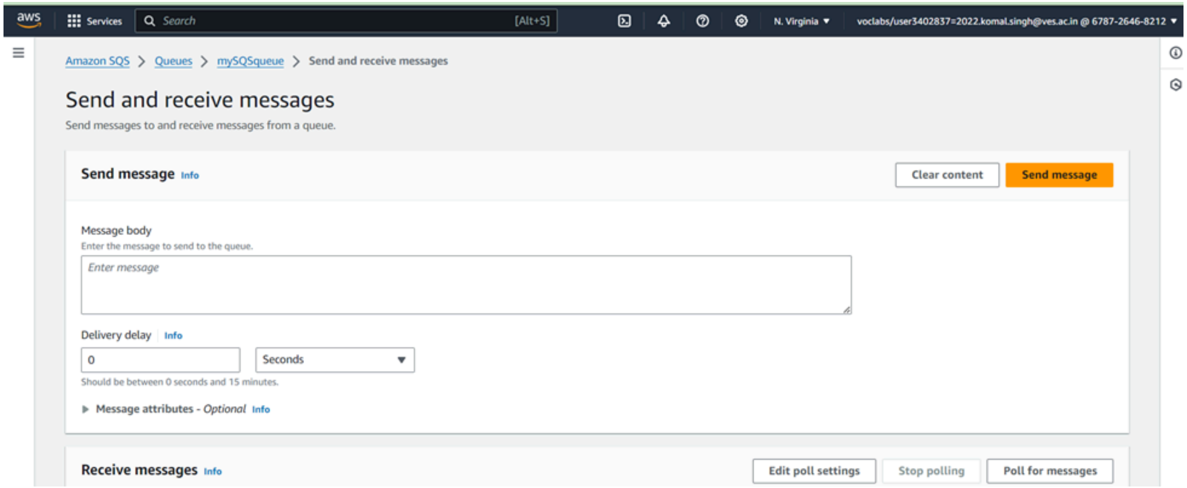
The environment variables below are encrypted at rest with the default Lambda service key.

Key	Value
S3_BUCKET	fdp-likely-native-kingfish-bucket



Triggers (1) Info

□	Trigger
□	SQS: mySQSqueue arn:aws:sqs:us-east-1:678726468212:mySQSqueue state: Enabled ▶ Details


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402837=2022.komal.singh@ves.ac.in @ 6787-2646-8212
Amazon SQS > Queues > mySQSqueue > Send and receive messages
Info
Clear content
Send message
Info
Info
Info
Edit poll settings
Stop polling
Poll for messages

Send and receive messages

Send messages to and receive messages from a queue.

Send message

Message body

Enter the message to send to the queue.

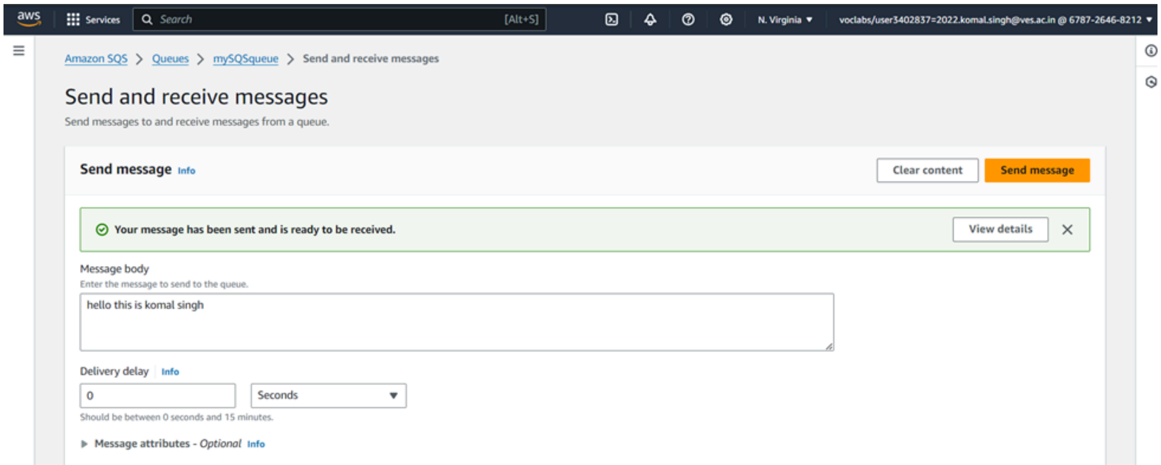
Enter message

Delivery delay: 0 Seconds

Should be between 0 seconds and 15 minutes.

▶ Message attributes - *Optional*

Receive messages


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402837=2022.komal.singh@ves.ac.in @ 6787-2646-8212
Amazon SQS > Queues > mySQSqueue > Send and receive messages
Info
Clear content
Send message
View details
Info
Info

Send and receive messages

Send messages to and receive messages from a queue.

Send message

Your message has been sent and is ready to be received.

Message body

Enter the message to send to the queue.

hello this is komal singh

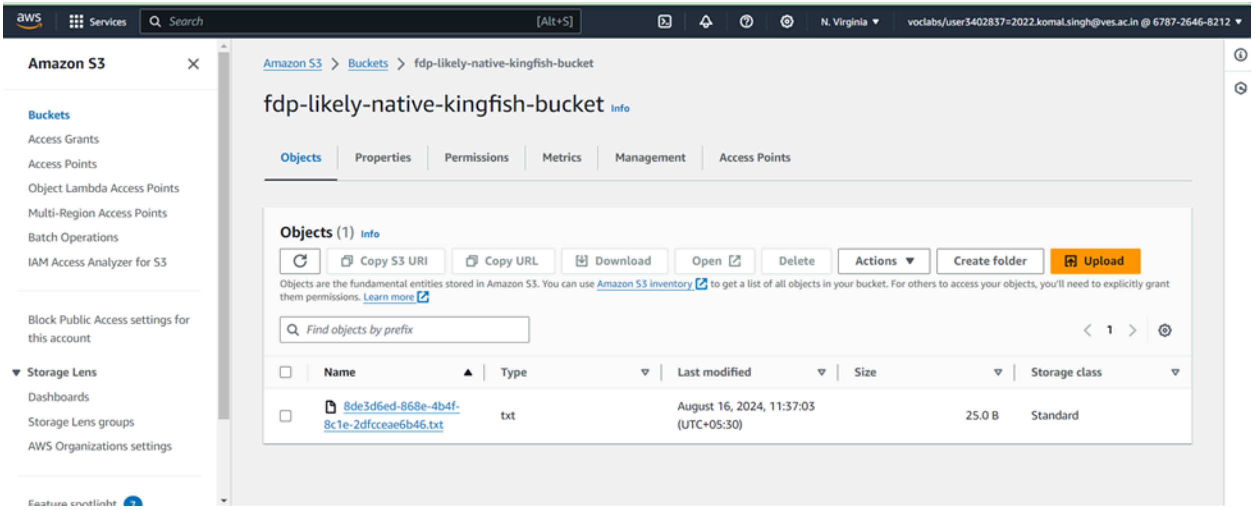
Delivery delay: 0 Seconds

Should be between 0 seconds and 15 minutes.

▶ Message attributes - *Optional*

Add S3 Event Notification for Lambda Trigger

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



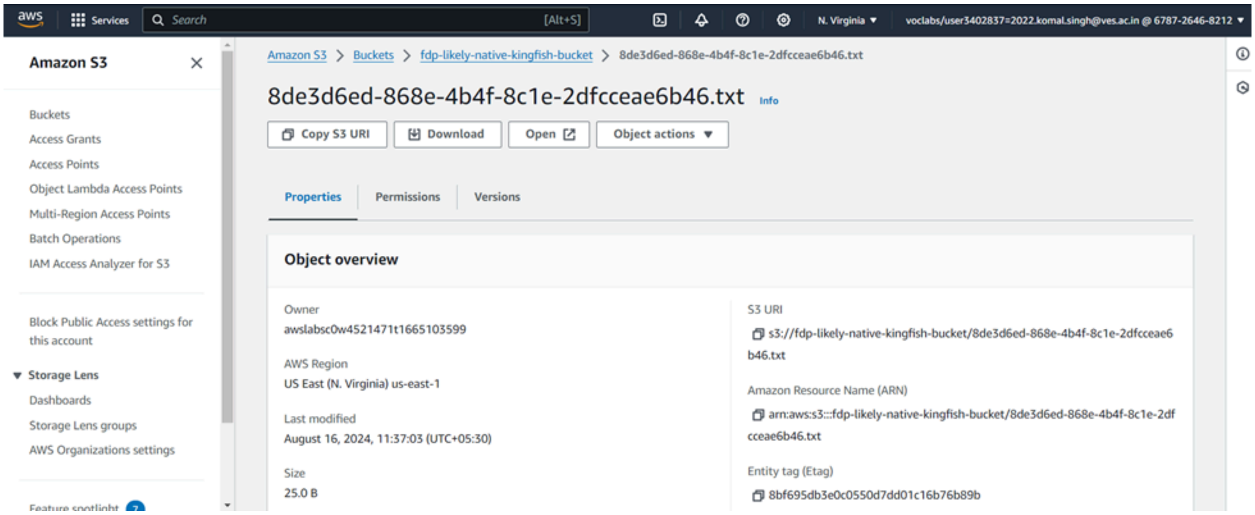
Amazon S3 > Buckets > fdp-likely-native-kingfish-bucket

fdp-likely-native-kingfish-bucket Info

Objects (1) Info

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

□	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



Amazon S3 > Buckets > fdp-likely-native-kingfish-bucket > 8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt

8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt Info

Object overview

Owner
awslabsc0w4521471t1665103599

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

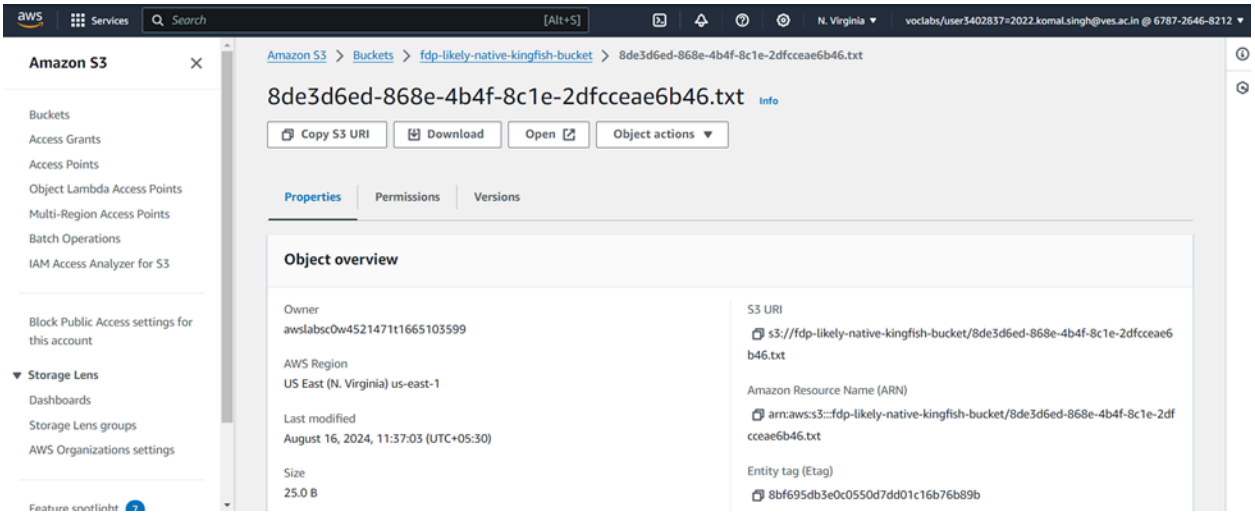
Last modified
August 16, 2024, 11:37:03 (UTC+05:30)

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 > fdp-likely-native-kingfish-bucket > 8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt

8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt Info

Object overview

Owner
awslabsc0w4521471t1665103599

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

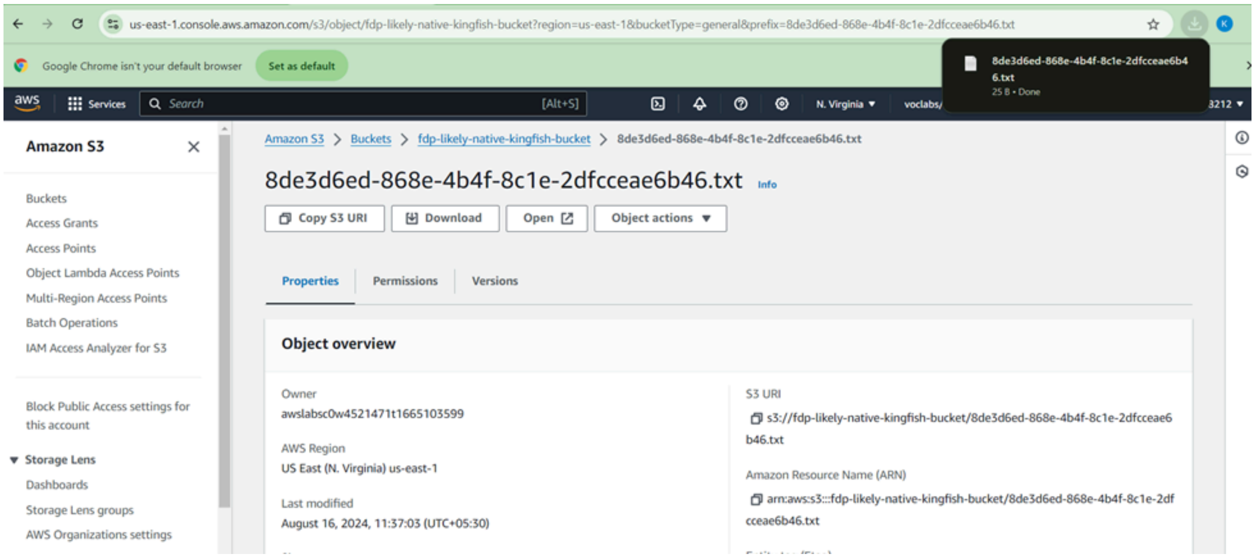
Last modified
August 16, 2024, 11:37:03 (UTC+05:30)

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


us-east-1.console.aws.amazon.com/s3/object/fdp-likely-native-kingfish-bucket?region=us-east-1&bucketType=general&prefix=8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt
Google Chrome isn't your default browser
Set as default
8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt
25 B • Done
aws
Services
Search
[Alt+S]
N. Virginia ▼
voclabs/
3212 ▼
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
Info
Copy S3 URI
Download
Open
Object actions ▼
Properties
Permissions
Versions

8de3d6ed-868e-4b4f-8c1e-2dfcceae6b46.txt

Object overview

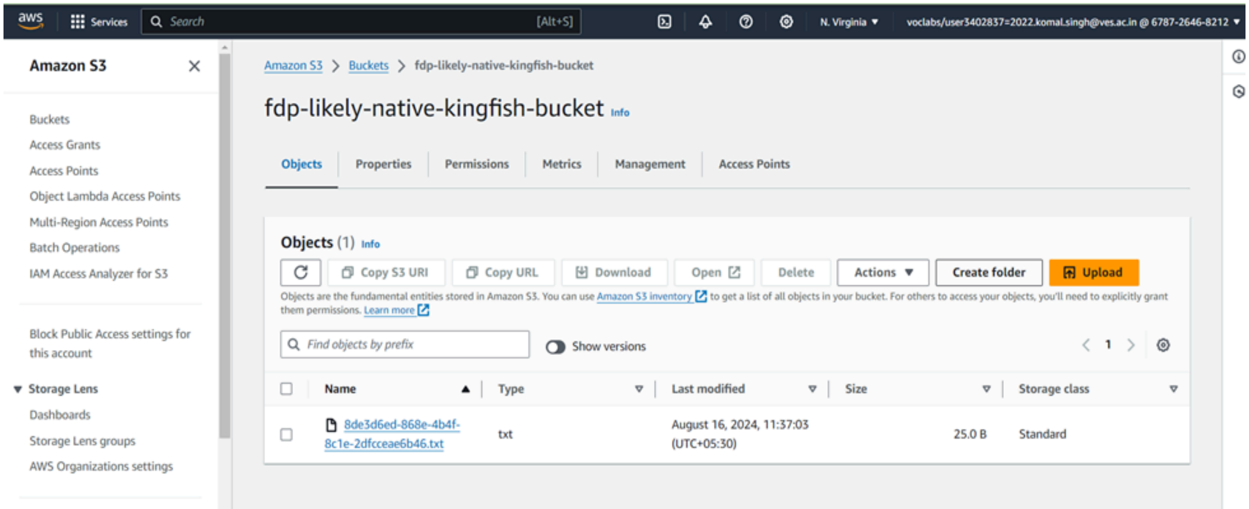
Owner
awslabsc0w4521471t1665103599

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

Last modified
August 16, 2024, 11:37:03 (UTC+05:30)

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

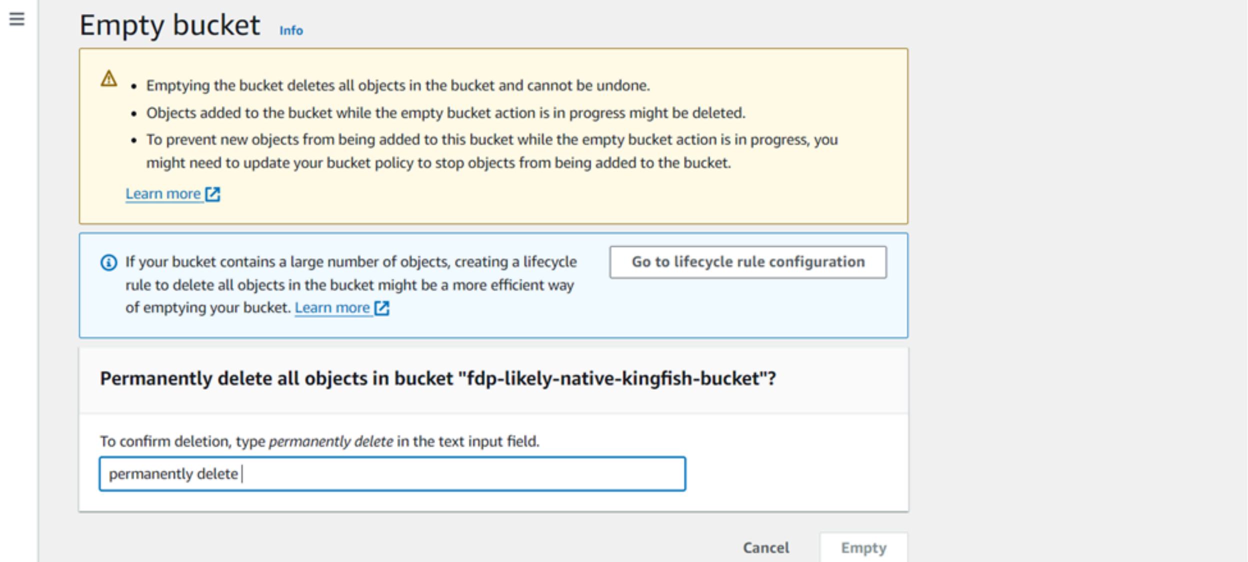


fdp-likely-native-kingfish-bucket

Objects (1)

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

□	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



Empty bucket

- Emptying the bucket deletes all objects in the bucket and cannot be undone.
- Objects added to the bucket while the empty bucket action is in progress might be deleted.
- To prevent new objects from being added to this bucket while the empty bucket action is in progress, you might need to update your bucket policy to stop objects from being added to the bucket.

If your bucket contains a large number of objects, creating a lifecycle rule to delete all objects in the bucket might be a more efficient way of emptying your bucket. Learn more

Permanently delete all objects in bucket "fdp-likely-native-kingfish-bucket"?

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

permanently delete