

Step 1: Create Your Terraform Configuration

1. **Create a directory** for your Terraform configuration if you haven't already.
2. Inside that directory, create a file named `main.tf`.

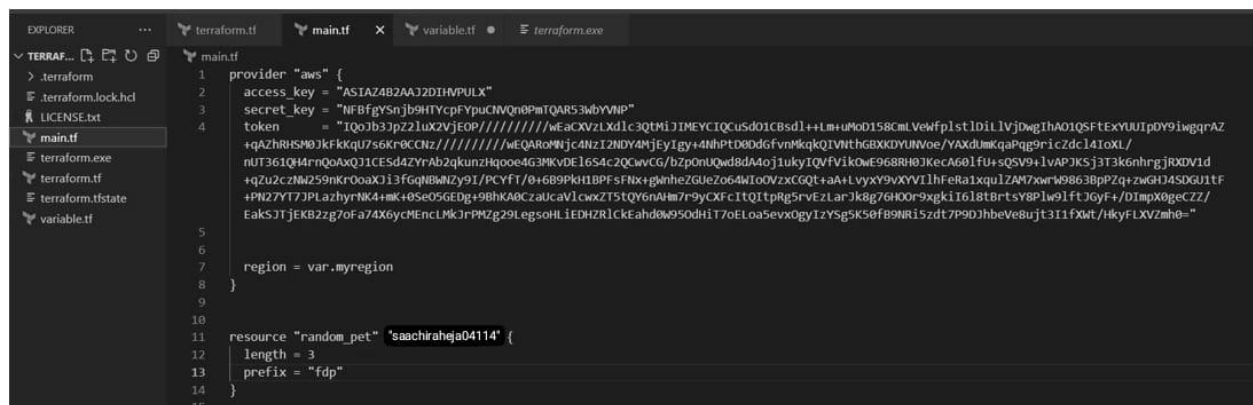
Here's an example configuration that sets up both an S3 bucket and an SQS queue:

```
provider "aws" {
  region = "us-east-1" # Change to your desired region
}

terraform {
  backend "s3" {
    bucket = "your-bucket-name" # Replace with your bucket name
    key    = "terraform/state"   # The path in the bucket for the
state file
    region = "us-east-1"        # Change to the bucket's region
  }
}

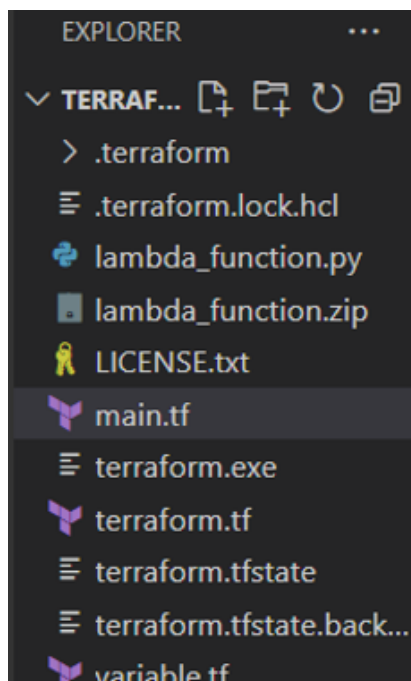
# Create an S3 bucket
resource "aws_s3_bucket" "example_bucket" {
  bucket = "example-bucket-unique-name" # Change to a unique name
  acl    = "private"
}

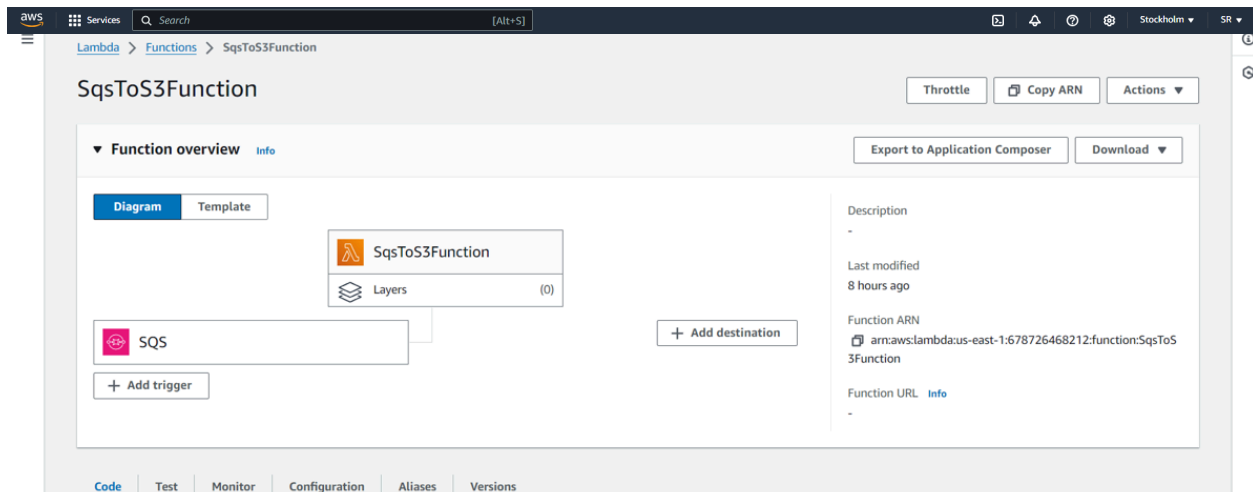
# Create an SQS queue
resource "aws_sqs_queue" "example_queue" {
  name                           = "example-queue"
  visibility_timeout_seconds     = 30
  delay_seconds                  = 0
}
```



```
terraform.tf × main.tf variable.tf ● terraform.exe
terraform.tf
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5     }
6     random = {
7       source = "hashicorp/random"
8     }
9     archive = {
10      source = "hashicorp/archive"
11    }
12  }
13 }
```

```
terraform.tf main.tf variable.tf ● terraform.exe
variable.tf
1 variable "myregion" {
2   type = string
3   default = "us-east-1"
4 }
5
```





Step 2: Initialize Terraform

Run the following command in the directory where your `main.tf` file is located:

```
terraform init
```

This command initializes the Terraform working directory and sets up the backend configuration to use S3 for state storage.

```
C:\Terraform>terraform -v
Terraform v1.9.4
on windows_amd64

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 v5.62.0...
- Installed hashicorp/aws v5.62.0 (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!
```

Step 3: Plan the Deployment

```
terraform plan
```

```
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:

# random_pet.saachiraheja041114 will be created
+ resource "random_pet" "saachiraheja041114" {
+   id       = (known after apply)
+   length   = 3
+   prefix   = "fdp"
+   separator = "-"
}

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

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.

Step 4: Apply the Configuration

To create the S3 bucket and the SQS queue, execute:

`terraform apply`

```
C:\Terraform>terraform apply

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:

# random_pet.saachiraheja041114 will be created
+ resource "random_pet" "saachiraheja041114" {
+   id       = (known after apply)
+   length   = 3
+   prefix   = "fdp"
+   separator = "-"
}

Plan: 1 to add, 0 to change, 0 to destroy.
Terraform will perform the following actions:

# random_pet.saachiraheja041114 will be created
+ resource "random_pet" "saachiraheja041114" {
+   id       = (known after apply)
+   length   = 3
+   prefix   = "fdp"
+   separator = "-"
}

Plan: 1 to add, 0 to change, 0 to destroy.
+ resource "random_pet" "saachiraheja041114" {
+   id       = (known after apply)
+   length   = 3
+   prefix   = "fdp"
+   separator = "-"
}
```

```

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

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

Only 'yes' will be accepted to approve.

Enter a value: yes

aws_sqs_queue.myqueue: Creating...

Enter a value: yes

aws_sqs_queue.myqueue: Creating...

aws_sqs_queue.myqueue: Creating...
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/678726468212/mySQSqueue]

aws_sqs_queue.myqueue: Still creating... [20s elapsed]
aws_sqs_queue.myqueue: Creation complete after 28s [id=https://sqs.us-east-1.amazonaws.com/678726468212/mySQSqueue]

aws_sqs_queue.myqueue: Creation complete after 28s [id=https://sqs.us-east-1.amazonaws.com/678726468212/mySQSqueue]

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

Outputs:

s3_arn = "arn:aws:s3:::fdp-likely-native-kingfish-bucket"

```

You will be prompted to confirm the changes. Type **yes** to proceed.

Step 5: Verify the Deployment

After the apply command completes, check the AWS S3 console to see your new bucket and the AWS SQS console to see your new queue.

The screenshot displays two screenshots from the AWS Management Console. The top screenshot shows the Amazon SQS console with a table of queues. The bottom screenshot shows the Amazon S3 console for a specific bucket.

Amazon SQS Console:

- Page title: Amazon SQS > Queues
- Buttons: Create queue, Edit, Delete, Send and receive messages, Actions
- Search: Search queues by prefix
- Table:

Name	Type	Created	Messages available	Messages in flight	Encryption	Content-based deduplication
mySQSqueue	Standard	2024-08-16T01:43:05:30	0	0	Amazon SQS key (SSE-SQS)	-

Amazon S3 Console:

- Page title: Amazon S3 > Buckets > fdp-likely-native-kingfish-bucket
- Bucket name: fdp-likely-native-kingfish-bucket
- Buttons: Objects, Properties, Permissions, Metrics, Management, Access Points
- Buttons: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, Upload
- Search: Find objects by prefix
- Table:

Name	Type	Last modified	Size	Storage class
8de3d6ed-868e-4b4f-8c1e-2dfccea6b46.txt	txt	August 16, 2024, 11:37:03 (UTC+05:30)	25.0 B	Standard

Step 6: Clean Up

To remove the resources you created, you can run:

`terraform destroy`

Empty bucket Info

⚠

- 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.

[Learn more](#)

📘 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](#)

Go to lifecycle rule configuration

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

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

permanently delete

Cancel

Empty

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:mysqlQueue" -> 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
```