Lab5a. Migrating a local Statefile to an AWS S3 bucket

Overview

This lab will use terraform to deploy a network stack into AWS. This will create a local backend (Statefile) on the client device used for the deployment. This Statefile will then be migrated to a centrally controlled secure backend, an S3 bucket, as is more typical in a production environment.

Solution

The solution to this lab can be found in awslabs/solutions/05a. Try to use this only as a last resort if you are struggling to complete the step-by-step processes.

Setup

1. Ensure you have completed Lab0 before attempting this lab.
2. In the IDE terminal pane, enter the following command…

   cd ~/environment/awslabs/05

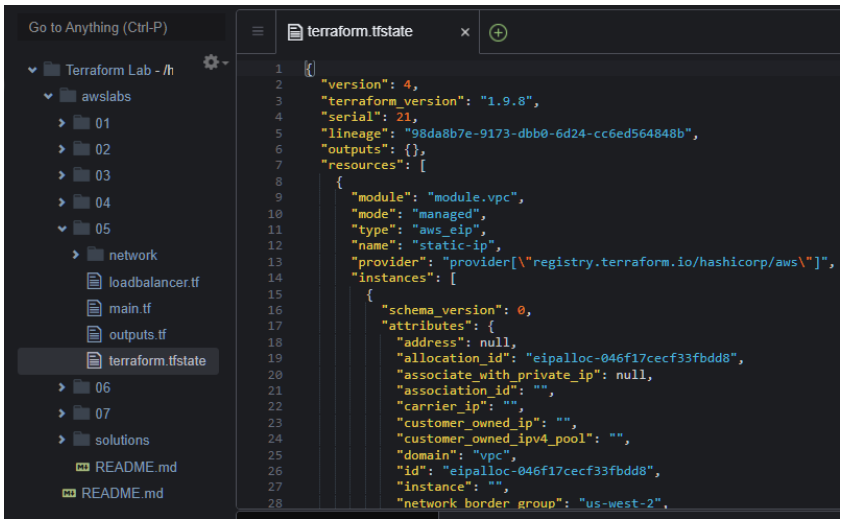
3. This shifts your current working directory to awslabs/labs/05. Ensure all commands are executed in this directory
4. Close any open files and use the Explorer pane to navigate to and open the awslabs/05 folder.
5. Run **terraform init** and then **terraform plan** commands
6. Verify there are no errors and then run **terraform apply**
7. Switch the console to review the deployment

You have just deployed a VPC into the us-west-2 region (you may currently be focussed on N. California from the previous lab so select the Oregon region). There is a public and private subnet in each of the 3 availability-zones. There

is a public routing table which uses an internet gateway and a private routing table which uses a NAT Gateway. The subnets have been associated with these routing tables. There are 2 security groups, one for ec2 instances, the other for load-balancers.

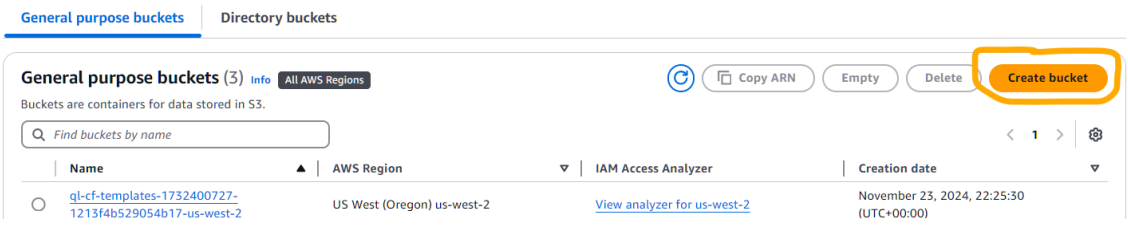
Task1: Review the Local Terraform State

1. Open the terminal and navigate to the Terraform project directory.
2. Using Explorer, expand the awslabs\05 folder and select the newly created terraform.tfstate file…



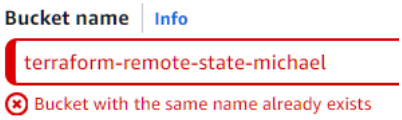
Task2: Create an S3 Bucket for Remote State

1. Switch to the AWS Console.
2. Search for and the navigate to the **S3** service and click on Create bucket …



3. Ensure you are focussed on the **Oregon** (us-west-2) region…

4. Name your bucket **terraform-remote-state-<your-name>** Every bucket name must be globally unique; therefore you may get a message indicating that a bucket already exists with your chosen name. If so, then simply append a random number after your name. Make a note of the bucket name that is accepted…



5. Leaving all settings at their default values, scroll down and select Create bucket…

Task3: Update the Terraform Configuration

1. Open awslabs/05/main.tf Terraform configuration file.
2. Add the following backend configuration for S3 code to the end of the file …

```
terraform {
 backend "s3" {
  bucket       = "terraform-remote-state-<your-name>"
  key          = " terraform.tfstate"
  region       = "us-west-2"
 }
}
```

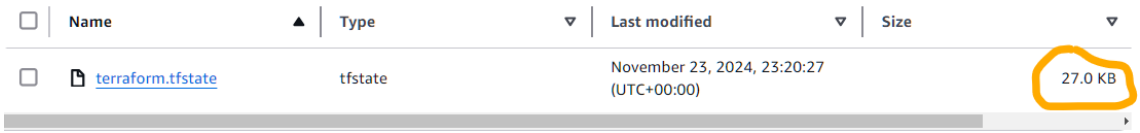
3. Save the file.

Task4: Initialize the S3 Backend and verify migration

1. Run the following command to migrate the state: **terraform init**

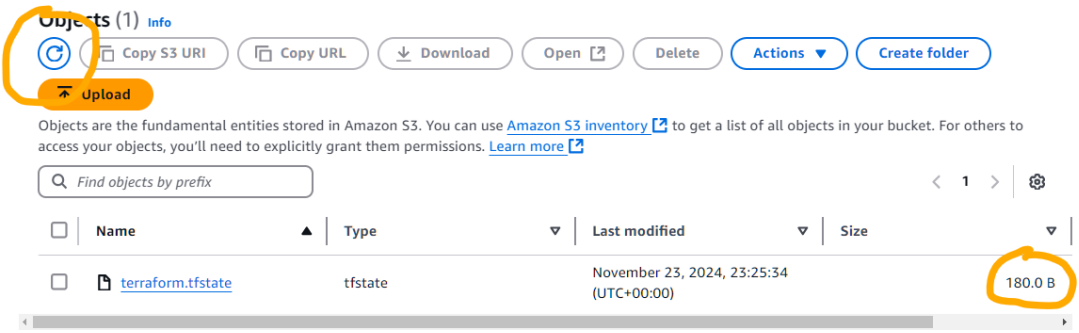
2. During the initialization, Terraform will prompt to confirm the migration of the local state to S3. Type **yes** when prompted.

3. Confirm that the state has been migrated by checking the S3 bucket for the terraform.tfstate file. Make a note of its size…



4. Switch to the IDE and destroy all your deployed resources using **terraform destroy**

5. In the Console, re-examine your remote terraform.tfstate file to verify it shrinks in size as a result of the destroy operation (you may need to refresh your view of the buckets contents)…



***** Congratulations, you have completed this lab *****