Table of Contents

[Description 3](#_Toc158731015)

[Preparation 3](#_Toc158731016)

[Observation 3](#_Toc158731017)

[Screenshots 4](#_Toc158731018)

[Appendix 16](#_Toc158731020)

[References 16](#_Toc158731021)

### Description

Terraform allows you to describe your complete infrastructure in the form of code. Even if your servers come from different providers such as AWS or Azure, terraform helps you build and manage these resources in parallel across.

This Lab consists of following tasks:

* Create AWS Account, and install Terraform.
* Create an EC2 instance using Terraform (Build Infrastructure)
* Destroy the existing instance and then create a new one (Change Infrastructure)
* Destroy the resources you created (Destroy Infrastructure)
* Include variables to make your configuration more dynamic and flexible (Define Input variables)
* Use output values to present useful information to the Terraform user (Query Data with Outputs)
* Migrate your state to Terraform Cloud (Store Remote State)

### Preparation

Terraform needs to be installed on the machine. This assignment requires an AWS account, I had an account which was created earlier. Here I implemented it in AWS using Amazon Linux Instance, installed all the relevant dependencies required by the Terraform. The console has my unique student ID: 8797297 and all the screenshots have my unique student ID, my AWS account: 7heKn1ght and the private and public address of my machine as I used AWS EC2 connect.

### Observation

This Lab was fairly straight forward, I didn’t have any issues or hiccups, I was able to follow the necessary steps which was required to finish this assignment.

### Screenshots

A screen shot of a computer

Description automatically generatedFig 1: Terraform Installation

Made sure the Amazon Linux instance was up to date and installed Terraform.  
  
A computer screen with a black screen

Description automatically generated Fig 2: Verify the installation  
To verify if Terraform was installed properly, I used Terraform -help which will list all the functions.

### A computer screen with a black screen Description automatically generatedFig 3: Created an AWS instance using Terraform

I had to make changes in main.tf as the region was us-east-1. After the file was created, I performed Terraform apply to create the instance.

A computer screen shot of a computer screen

Description automatically generated Fig 4: AWS instance created

Aninstance ‘ExampleAppServerInstance’ was created.

A computer screen shot of a black screen

Description automatically generatedfig 5: Terraform show command

The show command lists out all the specific details about the instance which was made.

A computer screen shot of a computer screen

Description automatically generatedFig 6: Change Infrastructure

Edited the main.tf file to make changes, I changed the AMI field, and created an ubuntu image to change the infrastructure.

A computer screen with a black screen

Description automatically generatedFig 7: Completion of Infrastructure change

The terraform apply will change the infrastructure, removing the existing instance and creates a new instance.

A screenshot of a computer

Description automatically generatedFig 8:New Ubuntu Instance deployed using Terraform

A computer screen shot of a black screen

Description automatically generatedFig 9: Terraform destroy

Delete the instance which was created using destroy command

A computer screen with a black screen

Description automatically generatedFig 10: After successful deletion of Infrastructure

A screen shot of a computer

Description automatically generatedFig 11: Defining variables and creation of instance

Created a .tf file to create customized name for the instances, when terraform apply is performed it loads all the .tf files.

A computer screen shot of a black screen

Description automatically generatedFig 12: Query data with Output

Initialize the main.tf and applied the initial configuration

A computer screen with a black screen

Description automatically generatedFig 13: Inspect the output values

Here I created an output.tf file which contains the two functions aws\_instance.app\_server.id, aws\_instance.app\_server.public\_ip. These will list out the instance ID and its public IP.

A computer screen shot of a black screen

Description automatically generatedFig 14.1: Destroyed the infrastructure.

A computer screen with a black screen

Description automatically generatedFig 14.2: Destroyed the infrastructure.

A computer screen with a black screen

Description automatically generatedFig 15: Created an account in Terraform Cloud to input the token

Created an account linked it and generated a token which was used to login in Terraform CLI.

A screenshot of a computer

Description automatically generatedFig 16: Created an organization and initialized the Terraform.

A screenshot of a computer

Description automatically generatedFig 17: Created the environment variables

Used the AWS IAM access keys to create Workspace Variables for the organization Conestog4College.

A computer screen shot of a black screen

Description automatically generated Fig 18: Performing remote state to check if there are any changes in existing infrastructure

A computer screen shot of a black screen

Description automatically generated Fig 19: Destroyed the infrastructure to release resources.

### Appendix

#### AWS account information

A screenshot of a computer

Description automatically generated

### References

* (Harrington D. , April 6, 2023)

What is Terraform: Everything You Need to Know

<https://www.varonis.com/blog/what-is-terraform>

* (Terraform)

What is Infrastructure as Code with Terraform?

<https://developer.hashicorp.com/terraform/tutorials/aws-get-started/infrastructure-as-code>