**Ontario Skill Cloud Computing 2025**

Camila Alejandra Esguerra Casas

Sheridan College

May 6th of 2025

**Index**

Table of Contents

[Introduction 3](#_Toc197431970)

[What is Terraform? 4](#_Toc197431971)

[Terraform important commands 5](#_Toc197431972)

[Connecting to AWS through Terraform 6](#_Toc197431973)

[What is a VPC 7](#_Toc197431974)

[Creating a VPC with Terraform 8](#_Toc197431975)

[Conclusion 10](#_Toc197431976)

[References 11](#_Toc197431977)

# Introduction

This document outlines the journey and methodologies used complete the documentation in regarding Amazon Web Services (AWS). This document will walk you through the process, steps, and detailed documentation of creating the best implementation of Terraform using Cloud computing techniques and more to create the same resources with the advantage of the mentioned tool. The documentation will show a variety of visual aids and enough information for the user get a better grasp of the subject and learn in the way how to do the procedures itself.

# What is Terraform?

Terraform is a infrastructure software and tool developed by the HashiCorp company. It allows users to connect with many platforms and providers that are compatible and use its tools as a better alternative to cloud computing by using an easy to use and manageable code language.

There is many way to code and deploy resources, especially in AWS. However, Terraform is a considerable useful tool for cloud related tasks, especially when paired up with AWS. Not only the syntax is easy to understand and stays consistent, but its straightforward and intuitive, making it an ideal tool.

A black screen with text and words

Description automatically generated with medium confidence

Screenshot 1

For a better understanding, the main point of the syntax can be divided into three main parts.

* ***“ Blocks:*** *They represent an object or resource. In blocks, there has to be a label and a body. Within the body, there can be more arguments nested within that specify other criteria (just like in other languages).*
* ***Arguments:*** *typically assign a value to a name and appear inside of the block.*
* ***Expressions:***  *They represent a value, either literal or using references and combining other values withing the body.”* Terraform Language documentation. *About the Terraform Language*. <https://developer.hashicorp.com/terraform/language>

# Terraform important commands

* **Terraform init:** Looks at all the configurations and looks for all the providers and then it will download all the necessary plugins to interact with it.
* **Terraform plan:** Will take a look at the current state of the infrastructure and compare it to the desired state (written code), after it will show the user a plan of what it will do to get to the desired state.
* **Terraform apply:** Will apply the changes to the infrastructure and create the necessary resources as specified in the code. If the user sends a terraform apply with a copy of the same instance within the code, it will not create a new instance and only update the existing one.
* **Terraform destroy:** will destroy the whole infrastructure in the account.

# Connecting to AWS through Terraform

The first steps into starting to manage or do any actions overall is specifying the provider, which in our case is clearly AWS. Terraform, as mentioned before, is a tool used for creating or generating any needed resources by coding it using the terraform provided language. This way, it is more accessible to the programmer to manage and do actions overall, since we won’t be needed to interact with the interface directly step by step, but instead will be able to create and plan everything all at once.

In the next example, we can see the syntax used in a terraform code to connect to the AWS server. In the next image we can visualize an example of the code used to connect to the server. The programmer specifies the provider (aws in this case), the region and it connects with its own access keys that are needed for terraform to access everything.

**A screenshot of a computer code

Description automatically generated**

Following up with the providers. Give the access keys to terraform and specify the desired availability zone

Specify the providers and version of the program

Screenshot 2

# What is a VPC

A Virtual Cloud Provider is a Virtual Network isolated inside of a cloud provider. It resembles a traditional network with the difference that can be contained online and replicated. In addition to this, it allows for more objects and extra resource or components to be added inside of it, complementing its function.

The main point of having a VPC is to be able to host your multiple instances and resources (websites, etc) inside of a Virtual Cloud Network rather than a fully physical one.

A diagram of a network

Description automatically generated

Screenshot 3. Resource: Amazon documentation. What is an amazon VPC? <https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>

# Creating a VPC with Terraform

Usually, a user can deploy one by either manually going to the GUI of AWS and performing a series of steps, or hard coding it with other tools (AWS CLI, etc). Following the Terraform concept and given the case scenario, a user should be able to deploy a VPC by using Terraform.

The following image contains a mentioned label as per stated in the Terraform syntax. The label specifies the type of resource as an “aws\_vpc” and next to it an assigned name. Additionally, nested inside, we can find the CIDR block which we will usually select in the GUI. In pair with the latter, the user is able to select any CIDR block as long as it doesn’t conflict with rules or creation of other resources.

A close-up of a number

Description automatically generated

Screenshot 4. Creating a VPC with terraform

# Conclusion

During this document, we learned about the different AWS services and resources that a person is able to create to fulfill a specific request and how to facilitate the deployment by using tools like Terraform. It delved into syntax, visualization of resources and meaning of important AWS concepts needed in the process of making a Service with the mentioned resources.

# References

* Terraform Language documentation. *About the Terraform Language*. <https://developer.hashicorp.com/terraform/language>
* Amazon documentation. What is an amazon VPC? <https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>