DevOps Tools You Should Know: Terraform

DJ. KONE

In this short demo, we'll create a simple VPC with CIDR Range: 192.168.0.0/24 using Terraform.

Background

Interested in automating infrastructure across any cloud platform? Terraform is your go-to tool.

Terraform is a tool that helps you build, manage, and organize your digital infrastructure in a smooth and automated way. It's like a wizard for your computer systems, allowing you to create and control your servers, databases, networks, and other components by writing code.

- `terraform init`: Prepares your workspace for applying configurations.
- `terraform plan`: Allows you to preview changes before applying them.
- `terraform apply`: Implements changes as defined by your plan to create, update, or destroy resources.
- `terraform destroy`: Destroys resources created by Terraform if they're no longer needed.

What's a VPC? Read this article I wrote to learn more: find it here.

DevOps Project Outline

Using Terraform To Deploy a VPC with CIDR range: 192.168.0.0/24.

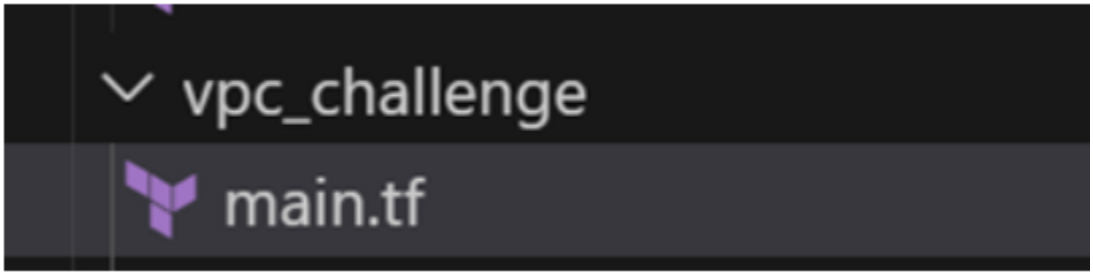
Prerequisite

- Install Visual Studio Code.
- Install Terraform.
- Access to AWS account.

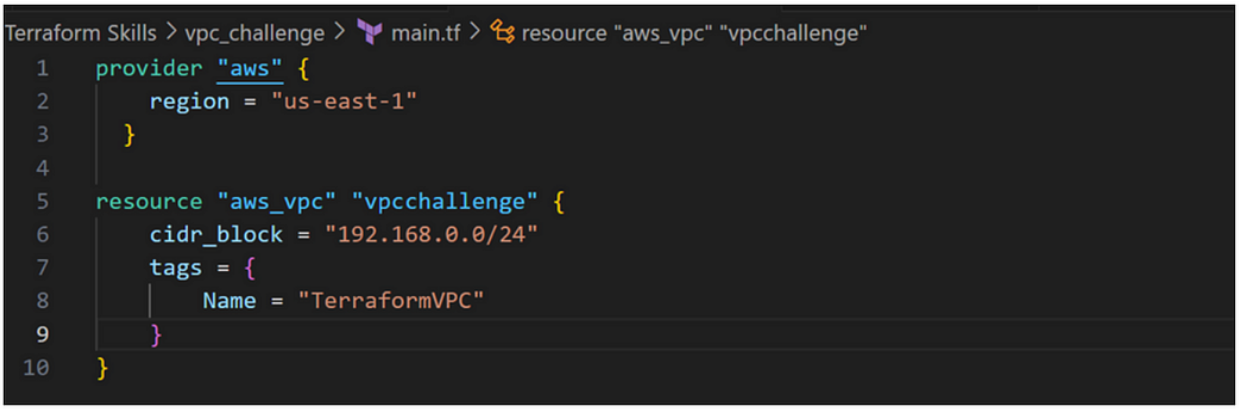
Let's have fun.

1- Launch Visual Studio Code

- Under ***Explorer,*** create a new folder named ***vpc_challenge***.
- Within that folder, add a new file named ***main.tf.***



- Add the script required to create the VPC.



Cheat Sheet: In the script above, 'provider' represents the cloud provider, such as AWS. It informs Terraform which services it needs

to interact with. 'Resource' is one of the fundamental elements in the Terraform language. Each 'resource' block describes one or more infrastructure objects, such as a VPC (Virtual Private Cloud). 'TerraformVPC' is the name assigned to our VPC that we intend to create. It's important to remember this name for future reference.

- Then, Save all.

2. Launch the terminal

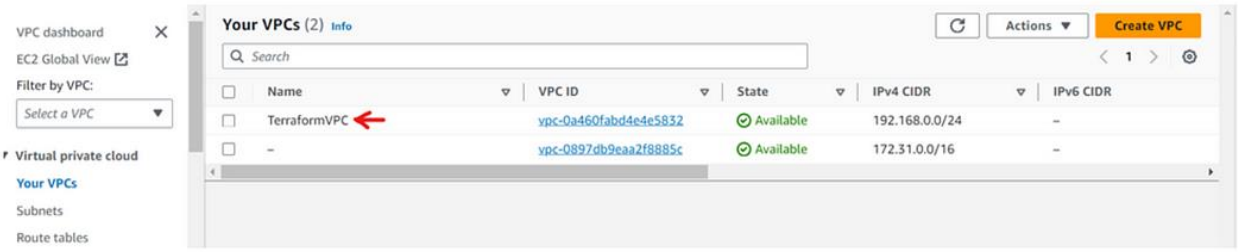
- Open the terminal in VS code.
- Navigate to the ***vpc_challenge*** folder.
- Execute this command in the terminal `terraform init`:

```
C:\Users\djkon\OneDrive\Documents\GitHub\Terraform\Terraform Skills\vpc_challenge>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.38.0...
- Installed hashicorp/aws v5.38.0 (signed by HashiCorp)
```

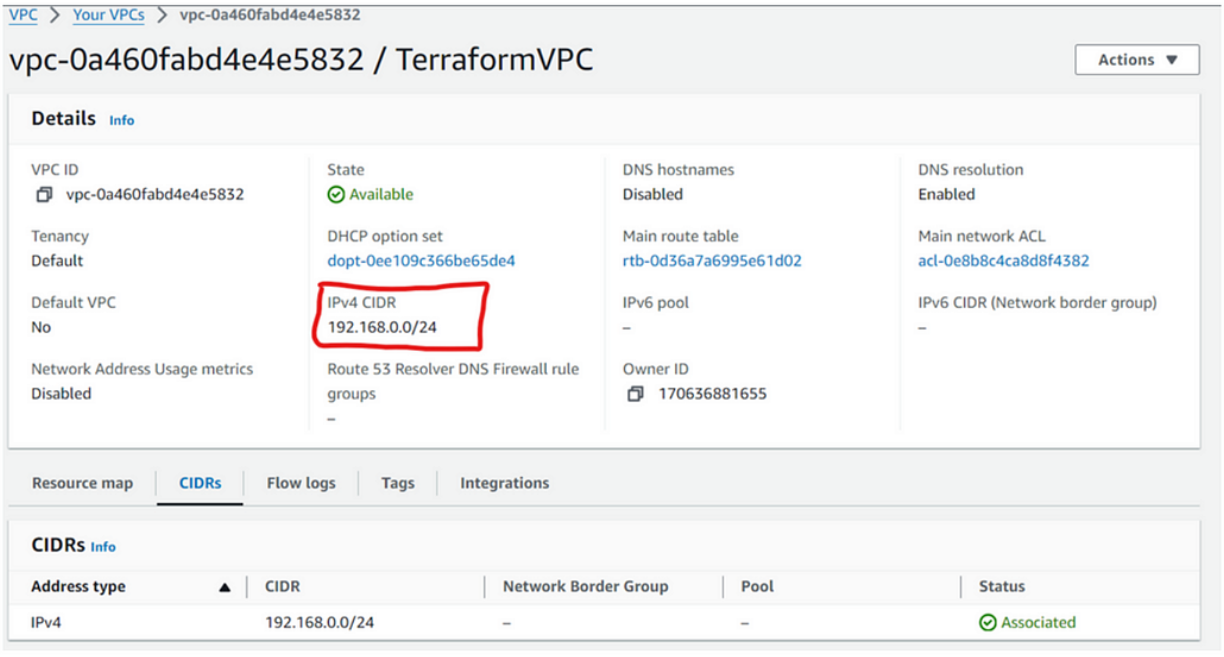
- Verify if the `cidr_block` is correct and if the resource has all the specifications as intended. Type "Yes" to proceed.

```
Terraform has been successfully initialized!
```

- Next, enter the command `terraform apply` .
- Navigate back to the AWS VPC console and confirm that the VPC has been successfully created.



- Verify if resource in the VPC console has all the specifications.



Voilà! You now know how to use terraform. This was a short tutorial on how to create a simple vpc with CIDR Block using Terraform.

Thank you for reading and/or following along! Leave us a clap or a comment. Please stay tuned for all my upcoming projects.

➞ Don’t forget to catch my latest medium Cloud projects.

➞ If you haven’t read my earlier articles on AWS Services, be sure to visit my daily blog post.

➞ Seeking lab exercises to replicate? Discover my comprehensive, step-by-step tutorials on GitHub.

➞ Or Contact me directly on LinkedIn or Email.