**Name**: Hiren Shah

**Date:** 12/25/2021

**Purpose**: To help complete beginners learn to use Terraform to automate AWS infrastructure.

The first thing you need to do is install terraform at the CLI using **terraform -v.**

Next configure the environment and region as well as access keys

I got most of the coding information from the Terraform docs here: <https://registry.terraform.io/providers/hashicorp/aws/latest/docs>

**Spin up EC2 Instances and Destroy Them:**

1. First thing we want to do is open up your favorite ide and set up the environment and region. Make sure to configure the access keys.

Text

Description automatically generated

Text

Description automatically generated

1. Run the **terraform init** command: This lets Terraform look at config that starts with .tf and looks at all the providers provided in the code and then it will download the plugins to interact with AWS API. (see screenshot below)

Text

Description automatically generated

1. Next use **terraform plan** command: As a sanity check to gloss over any changes made so you don’t break anything (see screenshot)

Text

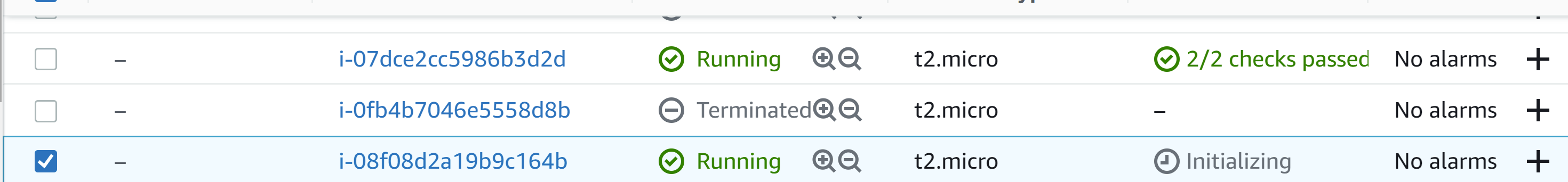
Description automatically generated

1. Next use the **terraform apply** command: To run the code and then enter ‘**yes**’ to continue to spin up instance. (see screenshot below of success)

Text

Description automatically generated

To verify that the instance lauched, I checked the AWS console:



**Note:** If you run **terraform apply** again, it will not spin up another instance because what is actually happening is that we are telling tf what we want our infrastructure should look like in the end. And it states that we should have one aws instance in the code (my-first-server).

**Modify Resources:**

Lets add a tag to make some changes and see what TF does:

I added a tag name “ubuntu” in the code and then ran the **terraform apply** command:

Text

Description automatically generated

To verify it added the tag to the instance in the console:

Graphical user interface, text, application

Description automatically generated

**Destroy Instances:**

Run the **terraform destroy** command to terminate instances:

Text

Description automatically generated

It took a while to destroy but this is the success you want to see:

Text

Description automatically generated

Note: Another way to destroy the instances is to delete the code and run **terraform apply**

**Create a VPC and Subnet via Terraform:**

The first thing you want to do is define the resources and the CIDR block for the vpc and subnet.

Text

Description automatically generated

For the vpc-id, you want to take the name of the aws\_vpc and paste it into the subnet’s vpc\_id section. This is how to reference other resources in your code. Run **terraform apply**.

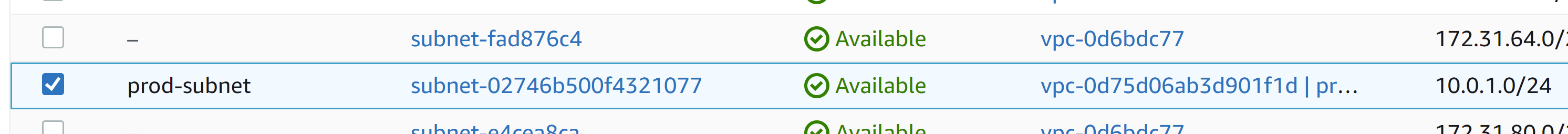
Text

Description automatically generated

Text

Description automatically generated

To verify that it was created in the console:



**Note:** It does not matter if you create subnet in you code before the vpc in terraform. TF can read your code asynchronously. Also if you are tired of typing “yes” to apply changes, you can use **terraform apply –auto-approve**. Go get some coffe and something to eat and come back when it’s deployed.