

Introduction to Terraform- Creating a simple cloud resource

Contents

Overview	1
Objectives	1
Lab Setup	1
Setup Terraform and Cloud Shell.....	2
Task 1: Review the documentation and create a configuration file to deploy a GCE instance.	3
Task 2: Run Terraform & Test.....	6
Solution.....	7

Overview

This lab will create an GCE instance on Google Cloud using Terraform.

Objectives

In this lab, you will:

1. Review the Terraform registry documentation
2. Deploy a GCE instance
3. Test the deployment
4. Modify the deployment
5. Test the modified deployment
6. Destroy the deployment

There is a Solution section at the very end of these instructions. Try to use this only as a last resort if you are struggling to complete the step-by-step processes.




Lab Setup

For each lab, you are provided with a new Google Cloud project and a set of resources for a fixed time at no cost.

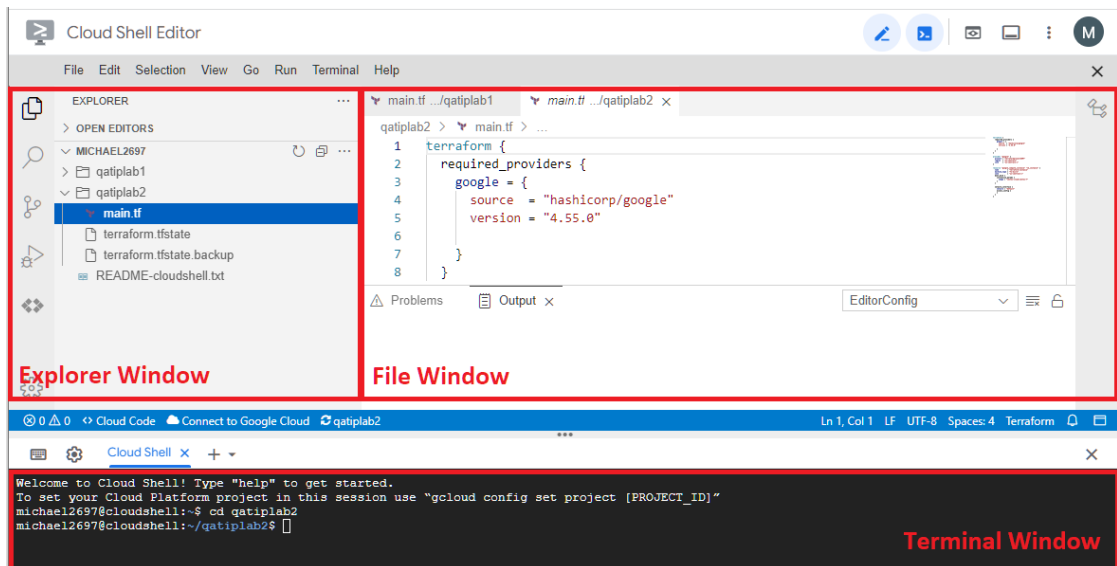
1. Note the lab's access time (for example, `1:15:00`), and make sure you can finish within that time. There is no pause feature. You can restart if needed, but you have to start at the beginning.
2. When ready, click **Start lab**.
3. Note your lab credentials (Username and Password). You will use them to sign in to the Google Cloud Console.
4. Click **Open Google Console**.
5. Click **Use another account** and copy/paste credentials for this lab into the prompts. If you use other credentials, you'll receive errors or incur charges.
6. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.

Terraform and Cloud Shell Setup

1. In the Cloud Console, click Activate Cloud Shell 
2. If prompted, click Continue.
3. Select the **Open in a new window** icon 
4. Select the **Open Editor** icon 
5. Select **Open Home Workspace** if explorer pane does not display.
6. Create a directory for your Terraform configuration and initial files by running the following commands in the terminal:

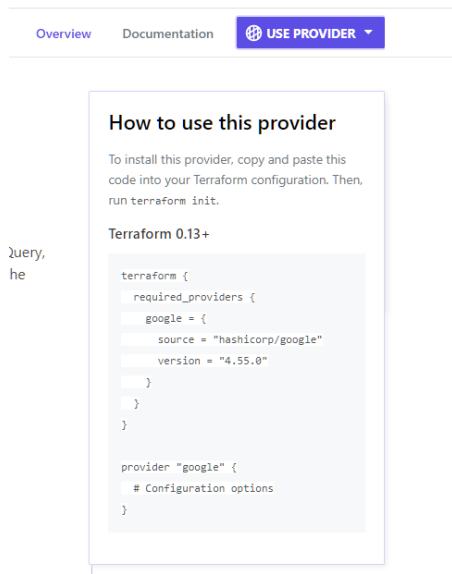
```
cd ~  
mkdir ./lab  
cd ./lab  
touch main.tf
```
7. In Explorer, navigate to and click on main.tf to open the empty file for editing...



8. You now have 2 browser tabs open. The current tab will be referred to as the `IDE` throughout these instructions.
9. Switch to the Google console tab and close the cloud shell. This tab will be referred to as the `Console` throughout these instructions.
10. Switch back to the `IDE`
11. Execute all terraform commands from within the lab directory

Task 1: Review the documentation and create a configuration file to deploy a GCE instance.

1. Review Terraform AWS Provider documentation:
<https://registry.terraform.io/providers/hashicorp/google/4.55.0/docs>
2. Click **Use Provider**



3. Copy the code block into main.tf in the IDE....
4. For convenience the code is listed below:

```
terraform {
  required_providers {
    google = {
      source = "hashicorp/google"
      version = "4.55.0"
    }
  }
}
```

```
provider "google" {
  # Configuration options
}
```

5. From within the documentation, we see that the provider configuration options typically include the project and region..

A typical provider configuration will look something like:

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}
```

6. Update your provider configuration options to include your `project id` and `region` which can both be found on the qwiklabs instruction screen.

```
provider "google" {
  project = "your project id here"
  region = "your region here"
```

```
}
```

7. Within the Terraform Google Cloud Provider documentation; https://registry.terraform.io/providers/hashicorp/google/4.55.0/docs/resources/compute_instance, navigate to the resource type **google_compute_instance**. Copy the sample code within the **Example Usage** section, and paste it into main.tf. For convenience, the code is listed below:

```
resource "google_service_account" "default" {
  account_id = "service_account_id"
  display_name = "Service Account"
}

resource "google_compute_instance" "default" {
  name          = "test"
  machine_type  = "e2-medium"
  zone          = "us-central1-a"

  tags = ["foo", "bar"]

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-11"
      labels = {
        my_label = "value"
      }
    }
  }

  // Local SSD disk
  scratch_disk {
    interface = "SCSI"
  }

  network_interface {
    network = "default"

    access_config {
      // Ephemeral public IP
    }
  }

  metadata = {
    foo = "bar"
  }

  metadata_startup_script = "echo hi > /test.txt"
```

```

    service_account {
      # Google recommends custom service accounts that have cloud-platform scope and
      # permissions granted via IAM Roles.
      email = google_service_account.default.email
      scopes = ["cloud-platform"]
    }
  }
}

```

8. Remove the `google_service_account` resource block in its entirety
9. Within the `google_compute_instance` resource block, remove the **tags**, the **labels** portion of the **boot disk**, all of the **scratch_disk**, metadata, and service account details.
10. Your modified block should now be as below...

```

resource "google_compute_instance" "default" {
  name          = "test"
  machine_type  = "e2-medium"
  zone          = "us-central1-a"

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-11"
    }
  }
  network_interface {
    network = "default"

    access_config {
      // Ephemeral public IP
    }
  }
}

```

11. Change the zone to reflect your zone, as displayed on the qwiklabs instructions screen.
12. Change the machine_type to **e2-small**
13. Save main.tf

Task 2: Run Terraform & Test

1. Run **terraform init** ensuring you are in the correct working directory
2. If there are any errors, correct them before continuing. (Use the solution guide if needed - but try first!)
3. Run **terraform plan** -- review what will be created

4. Run **terraform apply** typing **yes** when prompted. Review output in the CLI. Notice now that a persistent terraform.tfstate file now exists.
5. Switch to the console and confirm the instance creation. Click on the instance name to drill into the details and note the instance ID.
6. Within the resource block for the aws_instance, change the machine_type size from **e2-small** to **e2-medium** to ascertain if this performs a modification or a deletion/re-creation.
7. Save the main.tf file
8. Run **terraform plan** Review the output and note that this is an 'update in-place'
9. Run **terraform apply** typing **yes** when prompted.
10. If your instance is currently running then you will receive an error message regarding the change. You can terminate - shut down, the instance manually ahead of the change, but the instance will then have to be restarted manually following the update. The alternative is to set the argument **allow_stopping_for_update = true** We will use this option, so add the following to the instance resource block...

allow_stopping_for_update = true
11. **Save, plan, and apply** to implement the change and use the Console to verify.
12. Check if the instance_id changed ?
13. Within the boot_disk section of the instance resource block, change the image from **debian-cloud/debian-11** to **ubuntu-2204-jammy-v20230302**
14. Run **terraform plan** Review the output and note that this is a replacement. A new instance must be created when changing the instance image.
15. Run **terraform apply** typing **yes** when prompted.
16. Watch as the old instance is first destroyed and a new one is created.
17. Run **terraform destroy** and type **yes** to destroy all lab resources

Congratulations, you have completed this lab

Solution

The completed main.tf for this lab is...

```
terraform {  
  required_providers {  
    google = {  
      source = "hashicorp/google"    }  
  }  
}
```

```
    version = "4.55.0"
  }
}

provider "google" {
  project = "your project id here"
  region = "your region here"
}

resource "google_compute_instance" "default" {
  name          = "test"
  machine_type  = "e2-medium"
  #machine_type = "e2-small"
  zone          = "your zone here"
  allow_stopping_for_update = true

  boot_disk {
    initialize_params {
      #image = "ubuntu-2204-jammy-v20230302"
      image = "debian-cloud/debian-11"
    }
  }

  network_interface {
    network = "default"

    access_config {
      // Ephemeral public IP
    }
  }
}
```