01:54:10 Exit Lab ✓ Complete Lab

Deploying Infrastructure to GCP with Terraform Cloud

⏱ 2 hours duration Practitioner Rate this lab

VIDEOS GUIDE

Deploying Infrastructure to GCP with Terraform Cloud

Introduction

You are a DevOps engineer working with an organization that has embraced Terraform to manage their infrastructure as code. Right now, they use cloud storage buckets as remote backends for managing state, but they are interested in the enhanced features offered by Terraform Cloud. You will need to configure a Terraform Cloud account with access to a test GCP project and deploy a set of resources in order to better understand how Terraform Cloud works as a backend, as well as the security implications of that access.

Solution

The Google service account credentials needed to set up the environment variable for your Terraform Cloud workspace are provided on the lab instructions page, so it is recommended that you keep the lab page open in a browser window and open all other required pages in additional windows or tabs.

In an incognito or private browser window, navigate to the Terraform Cloud homepage (https://app.terraform.io/session) in order to create a free account. In another window, navigate to your email account to access the confirmation email used to verify your new Terraform Cloud account.

In an additional incognito or private browser window, log in to the Google Cloud Platform using the credentials provided on the lab instructions page. On the **Welcome, Cloud Student** screen, review the text and click **Accept**. In the **Welcome, Cloud Student!** pop-up, choose your country of residence, check to agree to the Terms of Service, and click **AGREE AND CONTINUE**.

Create a Free Terraform Cloud Account and Configure Your Workspace

Create Your Free Terraform Cloud Account

1. Navigate to the Terraform Cloud webpage at **https://app.terraform.io/session**.
2. On the Terraform Cloud homepage, click **Create your free account**.
3. Provide a username, an email address, and a password.
4. Check the **I agree to the Terms of Use** and **I acknowledge the Privacy Policy** checkboxes.
5. Click **Create account**.
6. Navigate to the inbox for the email address used to create the Terraform Cloud account.
7. In the Terraform Cloud confirmation email you received, copy the URL provided.

> **Note:** If you did not receive the confirmation email from Terraform Cloud, click Resend Confirmation Link on the User Settings page that displays in Terraform cloud.

8. Paste the URL into the browser window and hit **Enter** to complete account creation.

Configure a New Organization and Workspace in Terraform Cloud

1. On the **Welcome to Terraform Cloud!** page, click **Start from scratch**.
2. In the **Organization name** field, enter a unique name for your organization.

> **Note:** If you get an error when creating your organization name, try adding some unique characters at the end until the name is accepted.

3. The **Email address** field will be auto-populated with the email address used for account creation.
4. Click **Create organization**.
5. On the **Choose your workflow** screen, click **CLI-driven workflow**.
6. In the **Workspace Name** field, enter *my-first-workspace*.
7. Click **Create workspace**.

Set the Service Account Credentials as an Environment Variable

Add the GCP Service Account Credentials as an Environment Variable for the Terraform Cloud Workspace

1. In your workspace in Terraform Cloud, click the **Variables** tab.
2. Click **+ Add variable**.
3. For the variable category, select **Environment variable**.
4. In the **Key** field, enter *GOOGLE_CREDENTIALS*.
5. Navigate to the instructions page for this lab and copy the **Service Account Credentials** provided in the **Credentials** section.
6. Paste the service account information you copied into the **Value** field.
7. Check the **Sensitive** checkbox.
8. Click **Save variable**.

In the Lab VM, Deploy Resources with Terraform

Connect to the Lab VM and Authenticate with Terraform Cloud

1. Navigate to the Google Cloud Platform.
2. In the menu on the left, scroll down and click **Compute Engine**.
3. For the pre-configured **lab-vm**, under **Connect**, click **SSH**.
4. In the pop-up, click **Connect**.
5. When the SSH session connects, log in to Terraform from the terminal:

   ```
   terraform login
   ```

6. At the `Enter a value:` prompt, type `yes` and press **Enter**.
7. Copy the URL that is returned, and navigate to it in a separate incognito or private window.
8. In the **Create API token** pop-up, enter *lab vm* in the **Description** field.
9. Click **Create API token**.
10. Copy the API token string that is displayed.
11. Navigate back to the terminal in the SSH session for **lab-vm**.
12. At the `Enter a value:` prompt, paste in the API token string that you copied and press **Enter**.

Set Up the Terraform Configuration

1. From the terminal, create a `main.tf` file:

```
touch main.tf
```

2. Open the `main.tf` file in the GNU nano editor:

   ```
   nano main.tf
   ```

3. Navigate to the Terraform Cloud workspace and click the **Overview** tab, and
4. Under **Example code**, copy the `terraform` configuration block provided.
5. Navigate back to the `main.tf` file and paste in the `terraform` block.
6. Add in a `provider` block with Google as the provider:

   ```
   provider "google" {
     project = ""
   }
   ```

7. From the address bar, copy the GCP project ID as it appears between `/projects/` and `/zones/` in the URL for the SSH session.
8. Paste the project ID in as the `project` value in the `provider` block.
9. Add a generic `google_compute_instance` resource for a VM named `created-with-terraform-cloud`:

   ```
   resource "google_compute_instance" "vm" {
     name         = "created-with-terraform-cloud"
     machine_type = "f1-micro"
     zone         = "us-central1-a"
     boot_disk {
       initialize_params {
         image = "debian-cloud/debian-11"
       }
     }
     network_interface {
       network = "default"
     }
   }
   ```

10. Press **Ctrl + X** to exit the GNU nano editor.
11. When prompted, press **Y** and then **Enter** to save the changes to the `main.tf` file.

Apply the Terraform Configuration From Both the CLI and Terraform Cloud

1. In the terminal, run `terraform init`.
2. Then, run `terraform apply`.

3. Navigate back to the Terraform Cloud workspace and, in the **Latest Run** section, click the **See details** button for the run that was just triggered in the CLI.
4. In the Terraform Cloud console, scroll through the resources that will be created when the configuration is applied.
5. Scroll down and click **Confirm & Apply.**
6. Add a comment and click **Confirm Plan**.
7. Once the resources have been created successfully, navigate to the Google Cloud Platform and refresh the **VM instances** page.
8. Verify that the `created-with-terraform-cloud` VM now appears in the list of VMs.
9. Navigate back to the Terraform Cloud workspace and, under **State versions created:**, click the link to the state file.
10. Scroll through the state file to view its contents.

Conclusion

Congratulations — you've completed this hands-on lab!

Tools

Lab Diagram Instant Terminal

Credentials

How do I connect?

google labs Account

Username

cloud_user_p_c59e5daf@linuxacademygclabs.com

Password

lPP6bXNQ

Open Link in Incognito Window

How do I connect?

Service Account

Service Account Credentials

```
{  "type": "service_account",  "project_id": "deploying-in-269-df20d012",  "p...
```

Additional Resources

Lab Resources

- **Terraform Cloud**: A HashiCorp offering that provides enhanced workflow support for Terraform projects.
- **Remote State**: Information about infrastructure can be stored remotely for repeatability and collaboration.

Learning Objectives

0 of 3 completed

□ Create a Free Terraform Cloud Account

- Go to **https://app.terraform.io/session** to sign up for a free Terraform Cloud account.
- Click the confirmation email link.
- Create a new organization and workspace.

□ Set Up Your GCP Service Account Credentials in Terraform Cloud

- Copy the lab service account string.
- Create an environment variable in Terraform Cloud called *GOOGLE_CREDENTIALS*, set the value to the service account information provided with the lab credentials, and mark it sensitive.

□ In the Lab VM, Deploy Resources with Terraform

- Log into the lab VM.
- Run `terraform login` and copy the link to your clipboard.

- Go to the link and name the access key.
- Copy the access key and paste it back into the lab terminal.
- Create a `main.tf` file.
- Add a `terraform` block, a `google` provider, and a VM resource.
- Execute `terraform init`.
- Execute `terraform apply`.
- Examine the output in Terraform Cloud.