Deploying a VM in AWS Using the Terraform Workflow

00:12:12 Exit Lab Complete Lab

45 minutes duration Apprentice Rate this lab

VIDEOS GUIDE

Deploying a VM in AWS Using the Terraform Workflow

Introduction

In this hands-on lab, we will be following the Terraform workflow — Write > Plan > Apply — to deploy a virtual machine (VM) in AWS. After a successful deployment, we will then clean up our infrastructure and destroy the resource we created.

Solution

1. Log in to the lab server using the credentials provided:

   ```
   ssh cloud_user@<Terraform-Controller>
   ```

2. In a web browser, log in to the AWS Management Console using the credentials provided.

Create a Directory and Write Your Terraform Code (Write)

1. In the CLI, create a new directory in the cloud_user's home directory called `terraform_code` to house your Terraform code:

   ```
   mkdir terraform_code
   ```

2. Switch to the new directory:

```
cd terraform_code
```

3. Using vi, create a new file called `main.tf` where you will write your code:

```
vi main.tf
```

4. In the file, paste the provided code that will be used to create the required VM (EC2 instance) in AWS:

```
provider "aws" {
  region = "us-east-1"
}
resource "aws_instance" "vm" {
  ami           = "DUMMY_VALUE_AMI_ID"
  subnet_id     = "DUMMY_VALUE_SUBNET_ID"
  instance_type = "t3.micro"
  tags = {
    Name = "my-first-tf-node"
  }
}
```

5. Press **Escape** and enter `:wq` to save and exit the file.

Plug the Provided AMI and Subnet ID Values Into Your Code

1. View the contents of the `resource_ids.txt` file that has been saved on the lab server:

```
cat /home/cloud_user/resource_ids.txt
```

The `ami` and `subnet_id` values that have been saved in this file will be displayed.

2. Copy the `ami` value.

3. Open the `main.tf` file that houses your code:

```
vi main.tf
```

4. Paste the `ami` value into your code for the `ami` parameter, replacing the

`DUMMY_VALUE_AMI_ID` placeholder text. (Note: You can add the subnet_id value at this step as well if you wish).

5. Press **Escape** and enter `:wq` to save and exit the file.
6. Copy the `subnet_id` value.
7. Open the file with your code again:

   ```
   vi main.tf
   ```

8. Paste the `subnet_id` value into your code for the `subnet_id` parameter, replacing the `DUMMY_VALUE_SUBNET_ID` placeholder text.
9. Press **Escape** and enter `:wq` to save and exit the file.

Initialize and Review Your Terraform Code (Plan)

1. Initialize the Terraform configuration and download the required providers:

   ```
   terraform init
   ```

2. Review the actions that will be performed when you deploy your code:

   ```
   terraform plan
   ```

   In this case, it will create `1` resource: the EC2 instance you configured in your code.

   If you scroll up, you will notice that only the `ami`, `instance_type`, `subnet_id`, and `tags` properties are configured, as that was included in your code.

   Everything else, denoted with a `+` sign, will be created from scratch or will be populated when Terraform creates the resource upon deployment of your code.

Deploy Your Terraform Code (Apply), Verify Your Resources, and Clean Up

1. Deploy the code:

   ```
   terraform apply
   ```

2. When prompted, type *yes* and press **Enter**.

3. Once the code has executed successfully, note in the output that `1` resource has been created.

   Note: You could also use the `terraform output` command at any time in the CLI to view the output on demand.

4. Verify that the resource was created correctly in the AWS Management Console:

   - Navigate to the AWS Management Console in your browser.
   - Type *EC2* in the search bar and select **EC2** from the contextual menu.
   - On the *Resources* page, click **Instances (running)**.
   - Verify that the instance, named `my-first-tf-node` (as configured in your code), appears in the list.

5. Back in the CLI, remove the infrastructure you just created:

   ```
   terraform destroy
   ```

6. In the plan output, notice that it will destroy `1` resource: the EC2 instance you just created.

   Note: You can also scroll through the rest of the plan output and view the properties of the resource that will be destroyed, if desired.

7. When prompted, type *yes* and press **Enter**.

8. In the notifications displayed in the CLI, note that the `aws_instance.vm` resource you created is now being destroyed.

9. In the AWS Management Console, click the refresh button inside the *Instances* page and verify that the `my-first-tf-node` instance no longer appears in the list.

Conclusion

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

Tools

Lab Diagram | Instant Terminal

Credentials

How do I connect?

AWS Account

Username

cloud_user

Password

%1Qzfosata

Copy Console URL

How do I connect?

Cloud Server

Username

cloud_user

Password

H&9rF_%Y

Terraform-Controller

3.92.69.26

Launch Instant Terminal

How do I connect?

Additional Resources

To create your EC2 instance (VM) in AWS, use the code provided below:

```
provider "aws" {
  region = "us-east-1"
}
resource "aws_instance" "vm" {
  ami           = "DUMMY_VALUE_AMI_ID"
  subnet_id     = "DUMMY_VALUE_SUBNET_ID"
  instance_type = "t3.micro"
  tags = {
    Name = "my-first-tf-node"
  }
}
```

Note: Please ensure that all resources are deployed in the AWS region **us-east-1**.

The Amazon Machine Image (AMI) ID and subnet ID has been placed in a file on the lab server called `resource_ids.txt`. You will access that file and copy/paste these values into your code to create your VM.

To make sure the lab is fully provisioned, please wait an extra minute or two before connecting via ssh to the lab provided server.

```
ssh cloud_user@<Terraform-Controller>
```

And, in a web browser, log in to the AWS Management Console using the credentials provided.

Learning Objectives

4 of 4 completed

- [ ] **Create a Directory and Write Your Terraform Code (Write)**
- [ ] **Initialize and Review Your Terraform Code (Plan)**
- [ ] **Deploy Your Terraform Code (Apply), Verify Your Resources, and Clean Up**
- [ ] **Plug the Provided AMI and Subnet ID Values Into Your Code**

    1. Copy the AMI and subnet ID for the VM that have been saved in the `resource_ids.txt` file on the lab server.
    2. Paste these values into your code in the `main.tf` file.