

**Installing Terraform and Working With Terraform Providers**

**Introduction**

In this hands-on lab, we'll go through installing and configuring Terraform version 13 on a Linux OS. We will also explore how to select and use a Terraform provider from among the many providers available publicly.

**Solution**

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

ssh cloud\_user@<PublicIP>

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

**Download And Manually Install the Terraform Binary**

1. Download the appropriate Terraform binary package for the provided lab server VM (Linux 64-bit) using the wget command:

wget -c https://releases.hashicorp.com/terraform/0.13.4/terraform\_0.13.4\_linux\_amd64.zip

1. Unzip the downloaded file:

unzip terraform\_0.13.4\_linux\_amd64.zip

1. Place the Terraform binary in the PATH of the VM operating system so the binary is accessible system-wide to all users:

sudo mv terraform /usr/sbin/

**Note:** If prompted, enter the username and password provided for the lab server.

1. Check the Terraform version information:

terraform version

Since the Terraform version is returned, you have validated that the Terraform binary is installed and working properly.

**Clone Over Code for Terraform Providers**

1. Create a providers directory:

mkdir providers

1. Move into the providers directory:

cd providers/

1. Create the file main.tf:

vim main.tf

1. Paste in the following code from the provided [GitHub repo](https://github.com/linuxacademy/content-hashicorp-certified-terraform-associate-foundations/blob/master/section3-hol1/main.tf):
2. provider "aws" {
3. alias = "us-east-1"
4. region = "us-east-1"
5. }
6. provider "aws" {
7. alias = "us-west-2"
8. region = "us-west-2"

}

resource "aws\_sns\_topic" "topic-us-east" {

provider = aws.us-east-1

name = "topic-us-east"

}

resource "aws\_sns\_topic" "topic-us-west" {

provider = aws.us-west-2

name = "topic-us-west"

}

```

1. To save and exit the file, press **Escape** and enter :wq.

**Deploy the Code with Terraform Apply**

1. Enable verbose output logging for Terraform commands using TF\_LOG=TRACE:

export TF\_LOG=TRACE

**Note:** You can turn off verbose logging at any time using the export TF\_LOG= command.

1. Initialize the working directory where the code is located:

terraform init

1. Review the actions performed when you deploy the Terraform code:

terraform plan

**Note:** Two resources will be created, consistent with the providers that were configured in the provided code snippet.

1. Deploy the code:

terraform apply

1. When prompted, type yes and press **Enter**.
2. Verify that two resources were created with their corresponding Amazon Resource Name (ARN) IDs in the region in which they were spun up.
3. Optionally, verify that the resources were created in their respective regions within the AWS Management Console:
   * Navigate to the AWS Management Console in your browser.
   * Verify that you are logged in to the *us-east-1* region upon signing in.
   * Click **Services**.
   * Type *SNS* in the search bar and select **Simple Notification Service** from the contextual menu.
   * In the menu on the left, click **Topics**.
   * Verify that the *topic-us-east* resource appears in the list.
   * At the top-right, click **N. Virginia** and select **us-west-2**.
   * Verify that the *topic-us-west* resource appears in the list.
4. Tear down the infrastructure you just created before moving on:

terraform destroy --auto-approve

**Conclusion**

Congratulations