

-: Terraform :-

Terraform is an open-source infrastructure as code software tool that provides a consistent CLI workflow to manage hundreds of cloud services. Terraform codifies cloud APIs into declarative configuration files.

Deliver Infrastructure as Code

- **Write**

Write infrastructure as code using declarative configuration files. HashiCorp Configuration Language (HCL) allows for concise descriptions of resources using blocks, arguments, and expressions.

- **Plan**

Run terraform plan to check whether the execution plan for a configuration matches your expectations before provisioning or changing infrastructure.

- **Apply**

Apply changes to hundreds of cloud providers with terraform apply to reach the desired state of the configuration.

Installation of Terraform: Linux – Ubuntu

- Install HashiCorp's Debian package repository.
`sudo apt-get update && sudo apt-get install -y gnupg software-properties-common curl`
- Add the HashiCorp GPG Key
`curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -`
- Add the official HashiCorp Linux repository.
`sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs) main"`
- Update to add the repository, and install the Terraform CLI.
`sudo apt-get update && sudo apt-get install terraform`

Terraform sections required in configuration file (main.tf)

1. Terraform Block
2. Providers
3. Resources

Deploying NGNIX Server using Terraform

- Ensure Docker engine is installed in your box.
- Create a directory named learn-terraform-docker-container
 - `mkdir learn-terraform-docker-container`
- Navigate to it
 - `cd learn-terraform-docker-container`
- Paste the following Terraform configuration into a file and name it main.tf

```
terraform {
  required_providers {
    docker = {
      source = "kreuzwerker/docker"
      version = "~> 2.13.0"
    }
  }
}

provider "docker" {}

resource "docker_image" "nginx" {
  name      = "nginx:latest"
  keep_locally = false
}

resource "docker_container" "nginx" {
  image = docker_image.nginx.latest
  name  = "tutorial"
  ports {
    internal = 80
    external = 8000
  }
}
```
- Initialize the project, which downloads a plugin that allows Terraform to interact with Docker.
 - `terraform init`
- Provision the NGINX server container with apply. When Terraform asks you to confirm type yes and press ENTER.
 - `terraform apply`
- Verify the existence of the NGINX container by visiting **localhost:8000** in your web browser or running `docker ps` to see the container.
- To make sure your configuration is syntactically valid and internally consistent by using the terraform validate command
 - `terraform validate`
Success! The configuration is valid.
- Inspect the current state using `terraform show`
- To stop the container, run terraform destroy.
 - `terraform destroy`

Terraform in AWS: Requirements

- 1) The Terraform CLI (1.14.9+) Installed
- 2) The AWS CLI Installed
- 3) An AWS Account
- 4) AWS Credentials with access key

VBR HDV TRAINING INC.