

Terraform Tutorial

1. Download and unzip Terraform

wget

https://releases.hashicorp.com/terraform/1.3.5/terraform_1.3.5_linux_amd64.zip -O terraform.zip

unzip terraform.zip

2. Move the binary to users bin

sudo mv terraform /usr/local/bin/

3. Verify Terraform is installed

terraform -v

```
liox@DESKTOP-SGIFBT8:~/jalasoft/jalasoft-bootcamp/devOps_1/JalaUniversity_DevOps$ terraform -v
Terraform v1.3.5
on linux_amd64
+ provider registry.terraform.io/kreuzwerker/docker v2.23.1
```

4. Create tf config file to pull & create a docker container

```
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "2.23.1"
    }
  }
}

provider "docker" {
  host = "unix:///var/run/docker.sock"
}

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

resource "docker_container" "nginx_Terraform_erase" {
  image = docker_image.nginx.image_id
  name  = "nginx_Terraform_erase"
  ports {
    internal = 80
  }
}
```

```

    external = 8080
  }
}

```

5. Initialize Terraform so that it can scan for resources needed and prepare everything

terraform init

```

liox@DESKTOP-SG1FBT8:~/jalasoft/jalasoft-bootcamp/devOps_1/JalaUniversity_DevOps$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of kreuzwerker/docker from the dependency lock file
- Using previously-installed kreuzwerker/docker v2.23.1

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

```

6. Check the plan of Terraform, which shows what resources will be created.

terraform plan

```

liox@DESKTOP-SG1FBT8:~/jalasoft/jalasoft-bootcamp/devOps_1/JalaUniversity_DevOps$ terraform plan
docker_image.nginx: Refreshing state... [id=sha256:88736fe827391462a4db99252117f136b2b25d1d31719806326a437bb48cb12d]
docker_container.nginx_Terraform: Refreshing state... [id=1859b975d5f6c22a458b39df39ec237af8fac93322d1c358d8c340dd594276e4]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

# docker_container.nginx_Terraform will be created
+ resource "docker_container" "nginx_Terraform" {
  + attach                = false
  + bridge                = (known after apply)
  + command               = (known after apply)
  + container_logs        = (known after apply)
  + container_read_refresh_timeout_milliseconds = 15000
  + entrypoint            = (known after apply)
  + env                  = (known after apply)
  + exit_code             = (known after apply)
  + gateway               = (known after apply)
  + hostname              = (known after apply)
  + id                    = (known after apply)
  + image                 = "sha256:88736fe827391462a4db99252117f136b2b25d1d31719806326a437bb48cb12d"
  + init                  = (known after apply)
  + ip_address            = (known after apply)
  + ip_prefix_length      = (known after apply)
  + ipc_mode              = (known after apply)
  + log_driver            = (known after apply)
  + logs                  = false
  + must_run              = true
  + name                  = "nginx_Terraform"
  + network_data          = (known after apply)
  + read_only             = false
  + remove_volumes        = true
  + restart               = "no"
  + rm                    = false
  + runtime                = (known after apply)
}

```

7. Apply all the configurations and create resources specified in the main.tf file

terraform apply

```
lisa@lisa-54700-SU18781:~/jalesoft/jalesoft-boutcamp/devops_17/JaleUniversity/DevOps$ terraform apply
docker_image.nginx: Refreshing state... [id=sha256:88736fe827391462a4db99252117f136b2b25d1d31719006326a437bb48cb12dnginx:latest]
docker_container.nginx_Terraform: Refreshing state... [id=1809b975d5f6c22a456b39d739ec237af8fac93322d1c388d8c340dd594276e4]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

# docker_container.nginx_Terraform will be created
+ resource "docker_container" "nginx_Terraform" {
  + attach      = false
  + bridge      = (known after apply)
  + command     = (known after apply)
  + container_logs = (known after apply)
}
```

8. Answer **yes** to run and complete the actions.

```
Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

docker_image.nginx: Creating...
docker_image.nginx: Still creating... [10s elapsed]
docker_image.nginx: Creation complete after 11s [id=sha256:88736fe827391462a4db99252117f136b2b25d1d31719006326a437bb48cb12dnginx:latest]
docker_container.nginx_Terraform: Creating...
docker_container.nginx_Terraform: Creation complete after 1s [id=8001a261118b2ede7ccd5139dbff4c1fd7bb7e1903cf75ed8c08bd01bf3a50ef]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

9. Test if the project is running at **localhost:8080**

