**3.Adding Maps and Lookups in your Terraform files**

**The lab files can be found @ https://github.com/satyensingh/terraform-assignment-resources.git**

**In the repo you will find main.tf, outputs.tf, and variables.tf.**

**Add a new variable called env. Set a description to “env: dev or prod”.**

**Convert the type from image\_name to map.**

**Change the default to use key/value pairs. Set dev to ghost:latest and prod to ghost:alpine.**

**Convert container\_name to a map. Change the default to use key/value pairs. Set dev to blog\_dev and prod to blog\_prod.**

**Convert ext\_port to a map. Change the default to use key/value pairs. Set dev to 8080 and prod to 80.**

**Now initialize Terraform.**

**Setup the Development environment**

**Create a workspace called dev.**

**Generate a Terraform plan. Output the plan and call it tfdev\_plan. Pass in a variable called env and set it to dev.**

**Apply tfdev\_plan.**

**Setup the Production environment**

**Create a workspace called prod.**

**Generate a Terraform plan. Output the plan and call it tfprod\_plan. Pass in a variable called env and set it to prod.**

**Apply tfprod\_plan.**

**Verify both environments work**

**Open a browser and navigate to the public IP. This should pull up the production environment.**

**Open a browser tab and navigate to the public IP on port 8080. This should pull up the development environment.**

**🡪**

terraform init

Plan the dev deploy:

terraform plan -out=tfdev\_plan -var env=dev

Apply the dev plan:

terraform apply tfdev\_plan

Destroy prod deployment:

terraform destroy -var env=dev -auto-approve

Plan the prod deploy:

terraform plan -out=tfprod\_plan -var env=prod

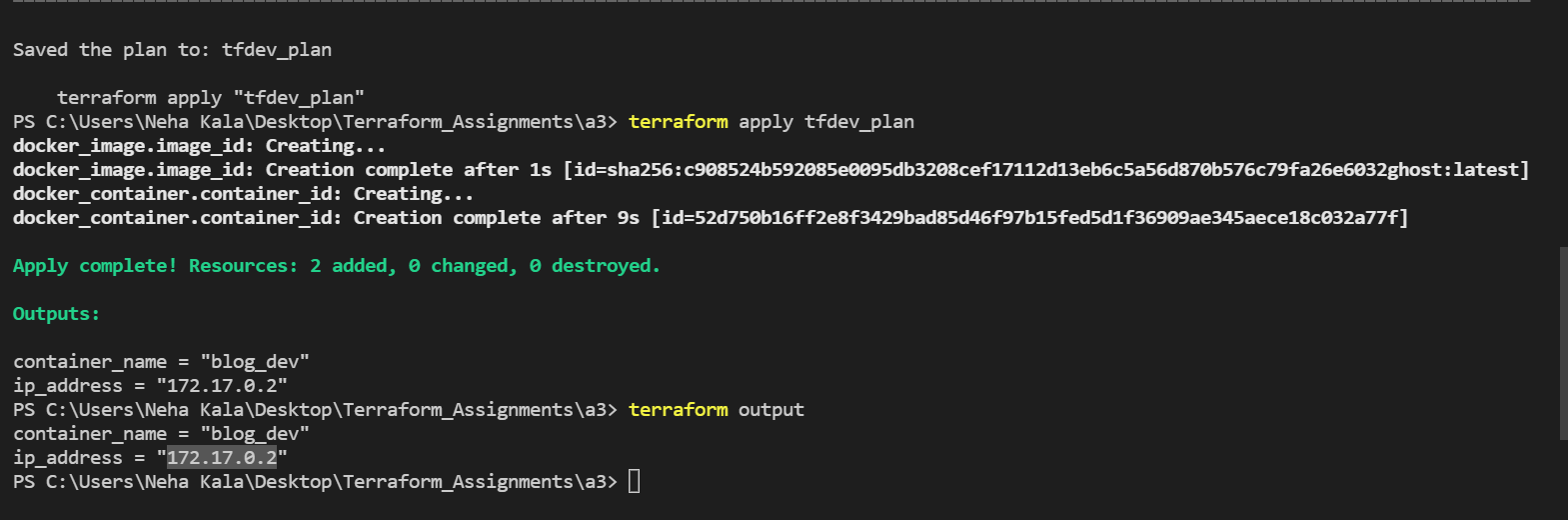
Apply the prod plan:

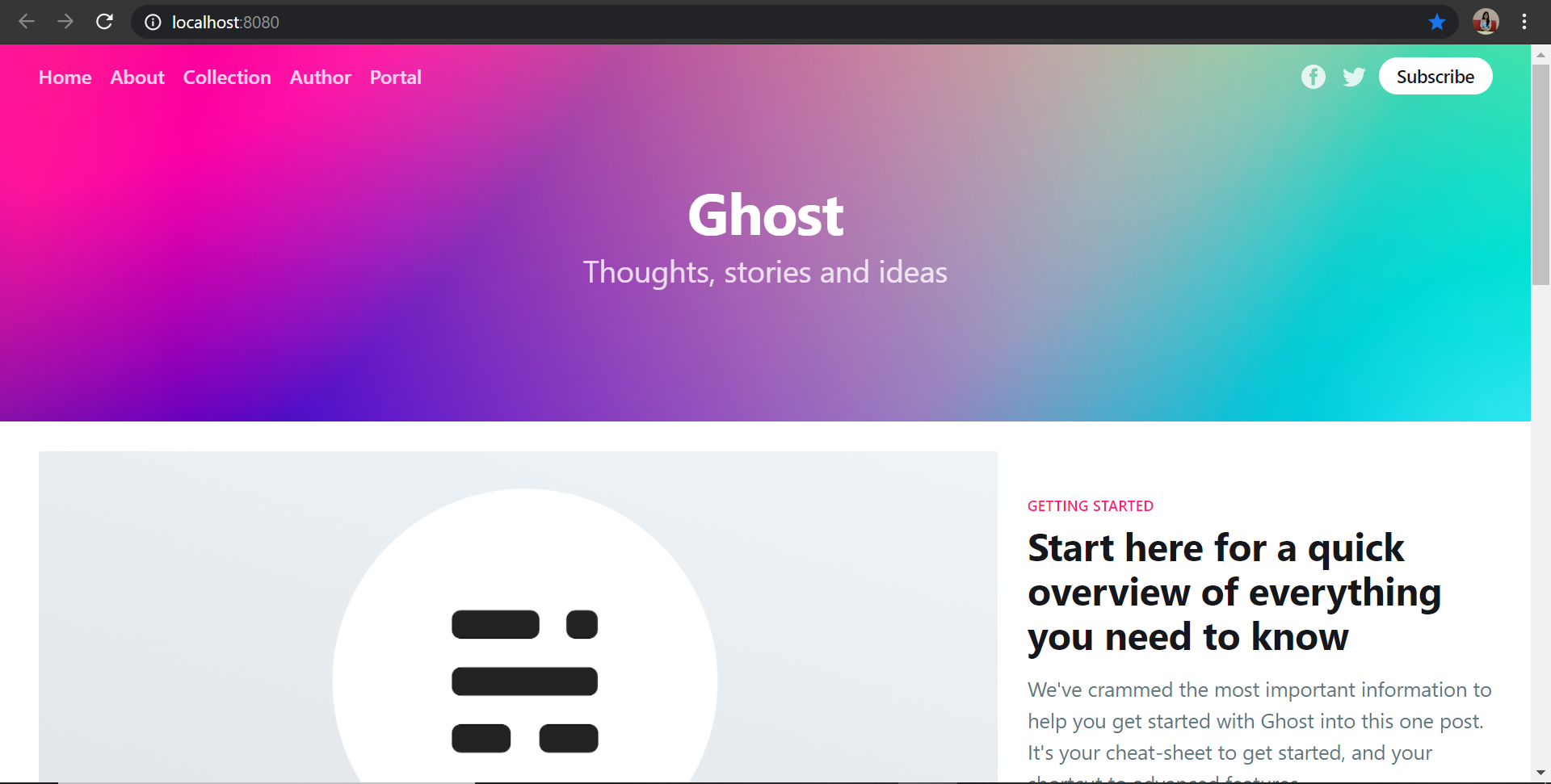
terraform apply tfprod\_plan

Destroy prod deployment:

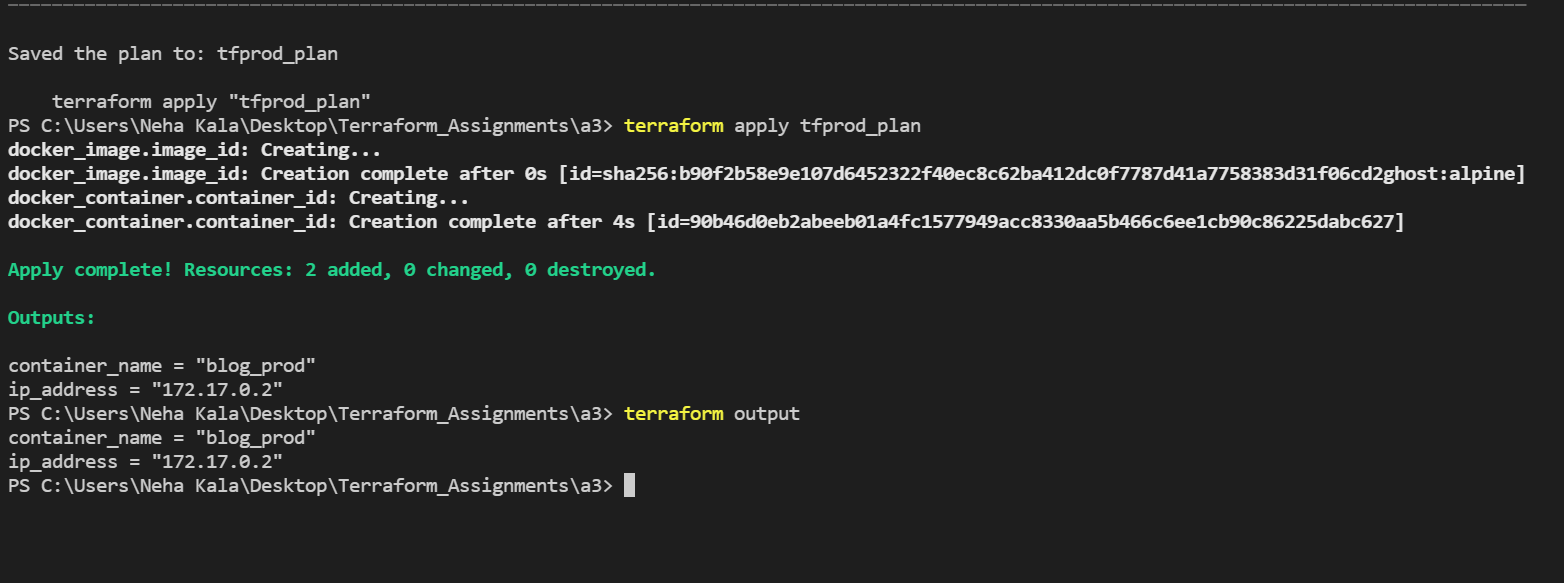
terraform destroy -var env=prod -auto-approve

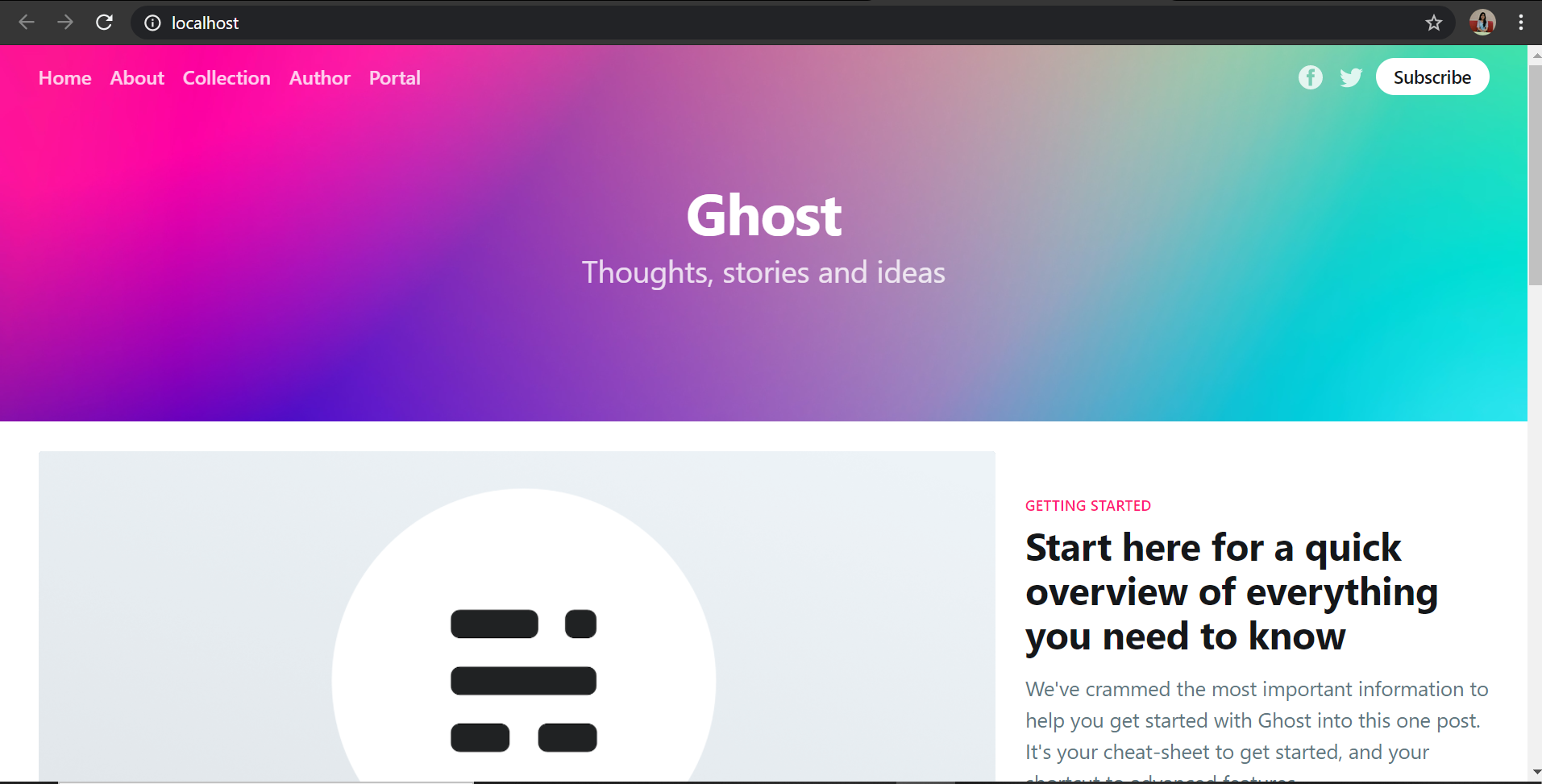
**Dev Environment-**

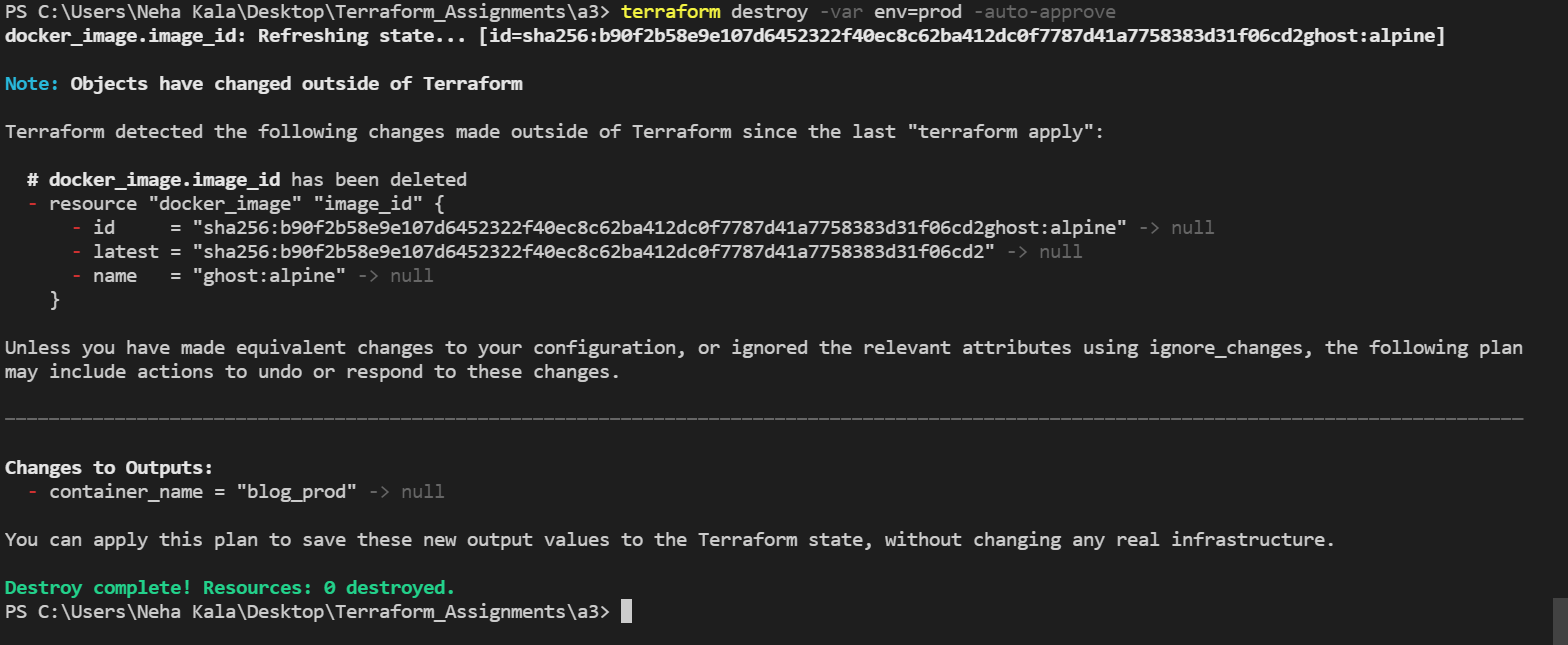




**Prod Environment-**







**Main.tf**

terraform {

  required\_providers {

    docker = {

      source = "kreuzwerker/docker"

    }

  }

}

provider "docker" {

  host    = "npipe:////.//pipe//docker\_engine"

}

# Download the latest Ghost Image

resource "docker\_image" "image\_id" {

  name = "${lookup(var.image\_name, var.env)}"

}

# Start the Container

resource "docker\_container" "container\_id" {

  name  = "${lookup(var.container\_name, var.env)}"

  image = "${docker\_image.image\_id.latest}"

  ports {

    internal = "2368"

    external = "${lookup(var.ext\_port, var.env)}"

  }

}

**Variables.tf**

variable "env" {

  description = "env: dev or prod"

  default = "dev"

}

variable "image\_name" {

  description = "Image for container."

  type = map(string)

  default = {

      dev : "ghost:latest"

      prod : "ghost:alpine"

  }

}

variable "container\_name" {

  description = "Name of the container."

  type = map(string)

  default = {

    dev : "blog\_dev"

    prod : "blog\_prod"

  }

}

variable "ext\_port" {

  description = "External port for container."

  type = map(string)

  default = {

    dev : 8080

    prod : 80

  }

}

**Output.tf**

#Output the IP Address of the Container

output "ip\_address" {

  value       = "${docker\_container.container\_id.ip\_address}"

  description = "The IP for the container."

}

output "container\_name" {

  value       = "${docker\_container.container\_id.name}"

  description = "The name of the container."

}