Assignment 1

Rishi (8972657)

Scripting and Automation

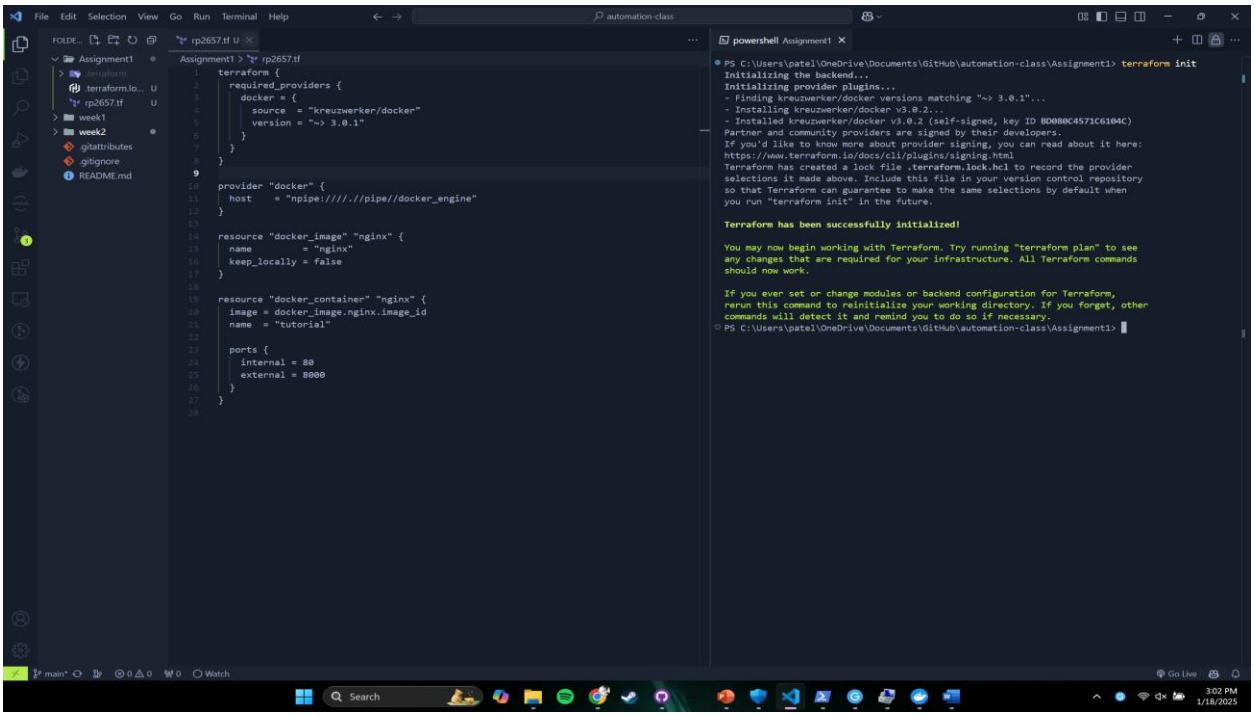
PROG8830 - Winter 2025 - Section 1

Prof. Mike Dabydeen

Jan 18, 2025

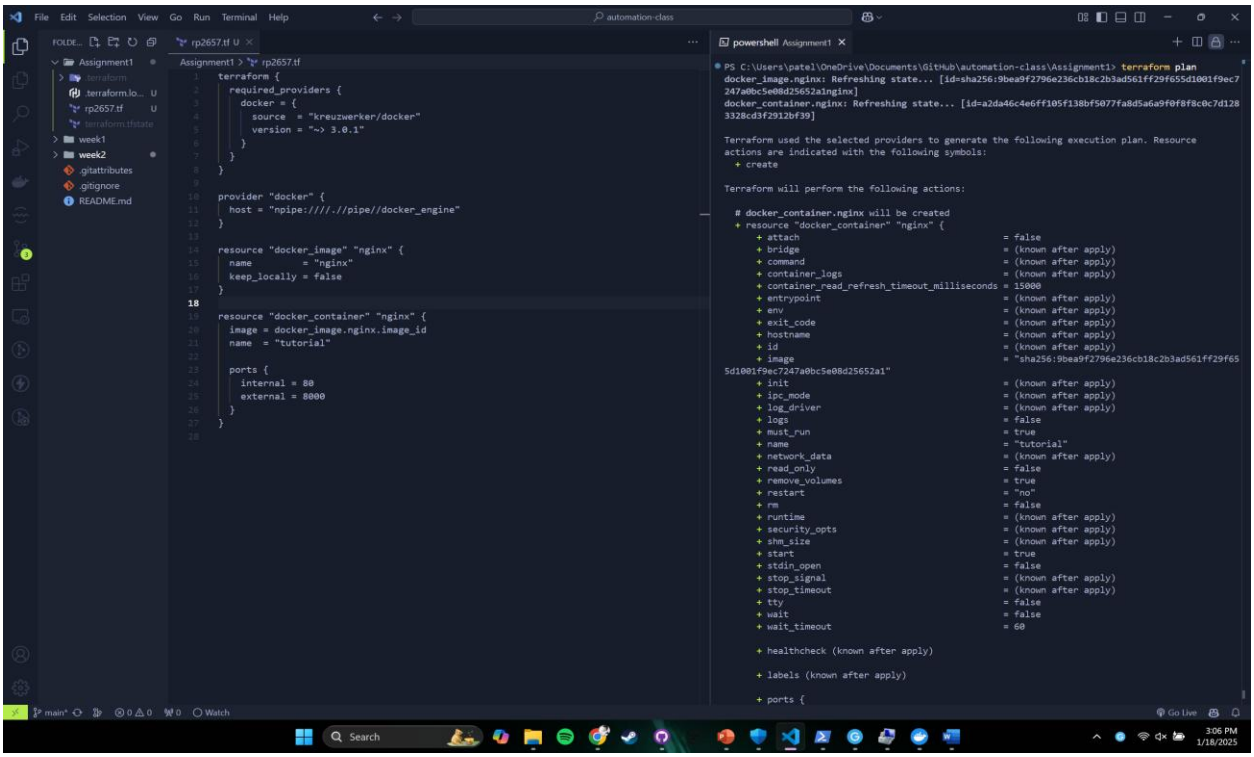
Initialize Terraform

Initialize Terraform in your working directory using this command:
`terraform init`



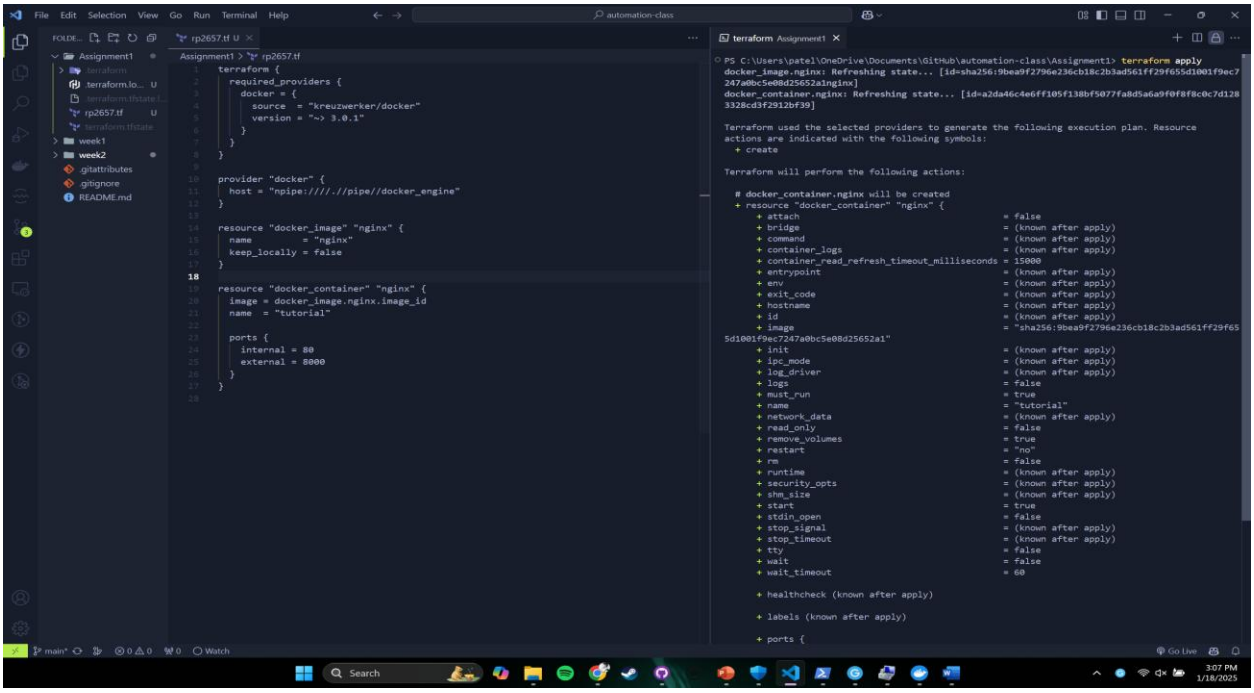
Review and Apply Configuration

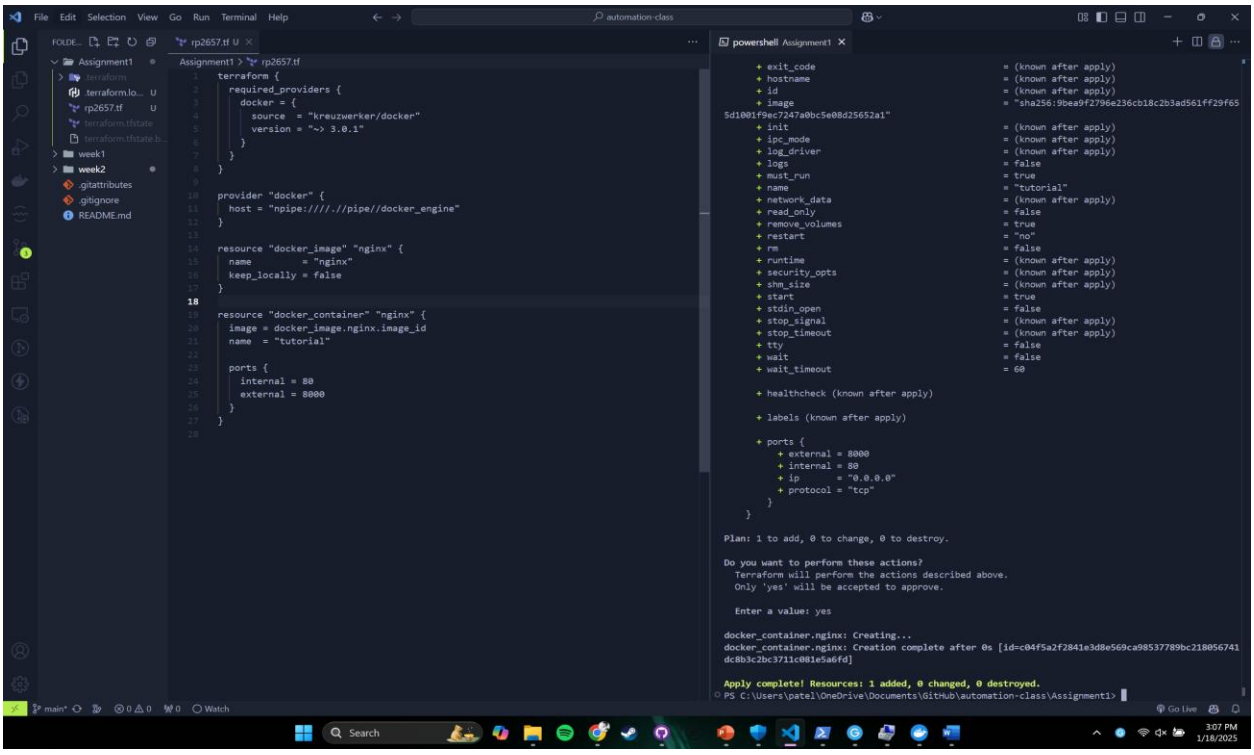
- Review the Terraform execution plan using the command `terraform plan`
- Ensure that the plan includes the creation of the desired Docker container running Nginx



Apply the Terraform configuration to provision the resources using the command:

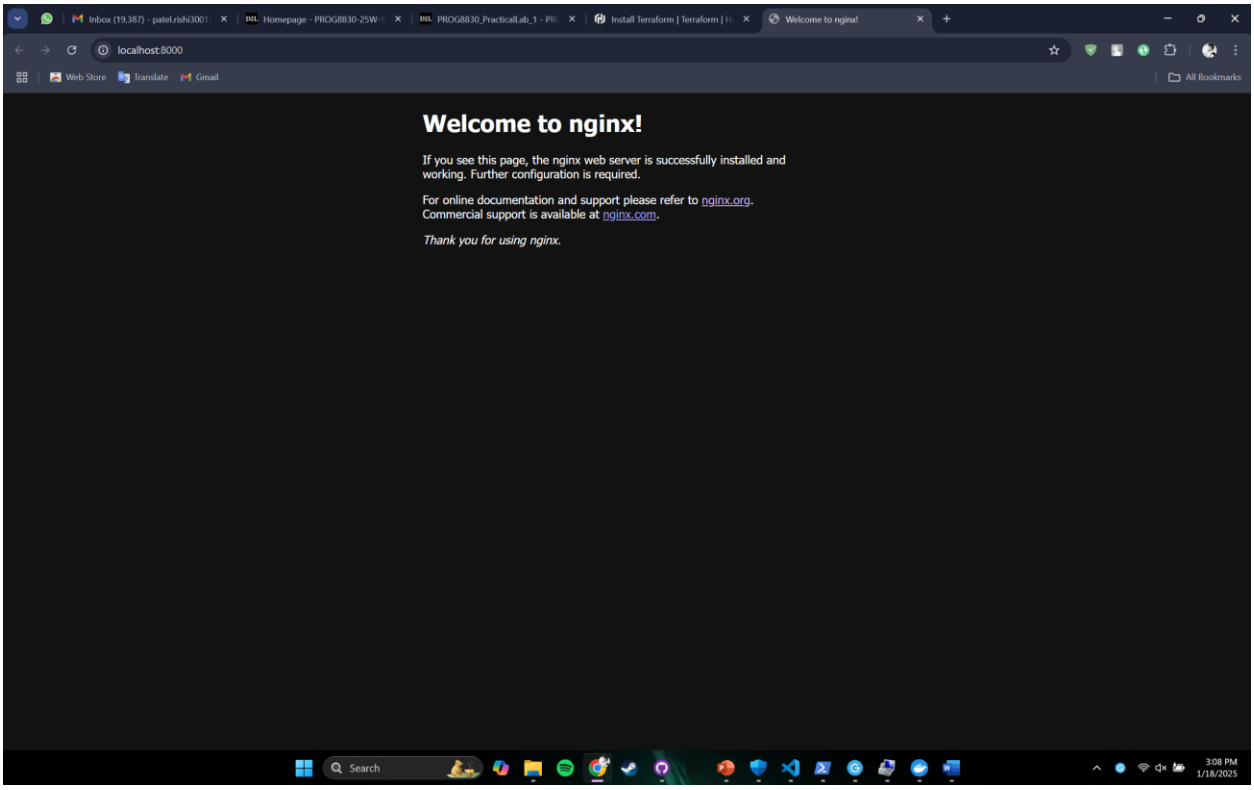
```
terraform apply
```





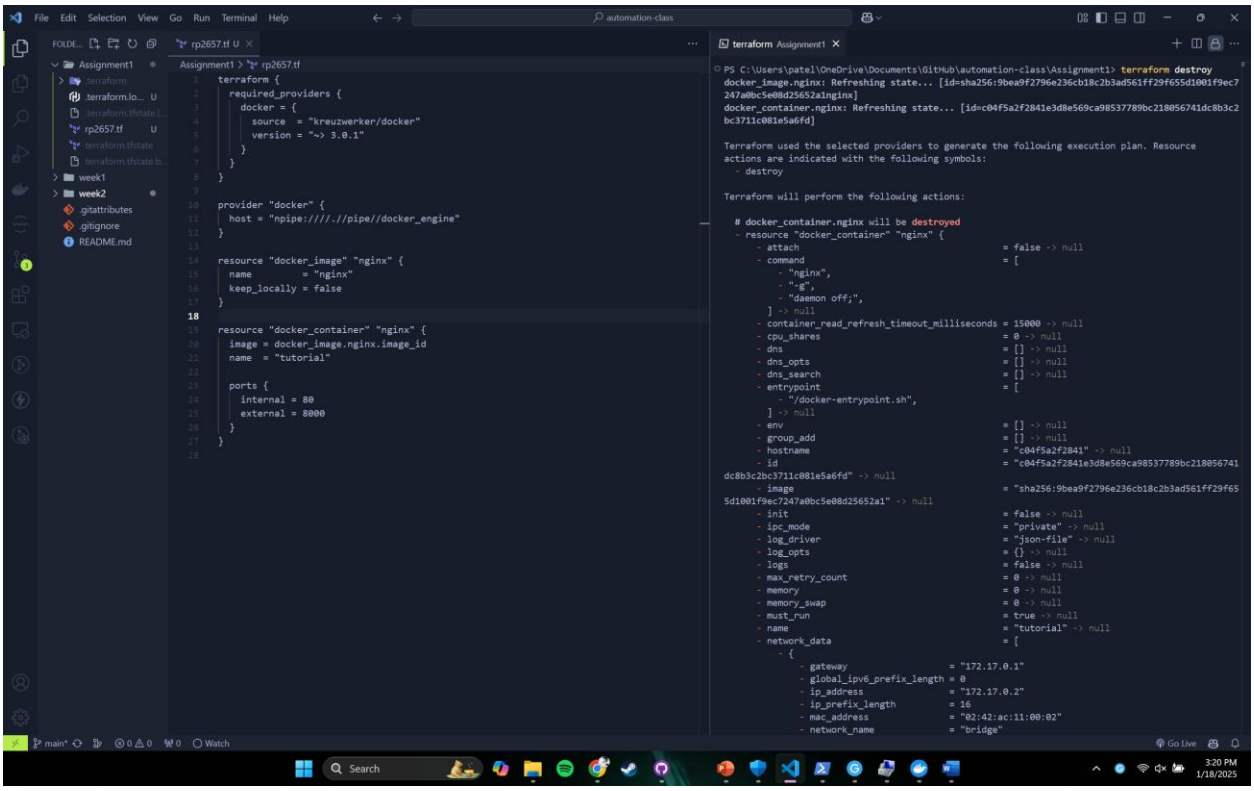
Verify Nginx Server

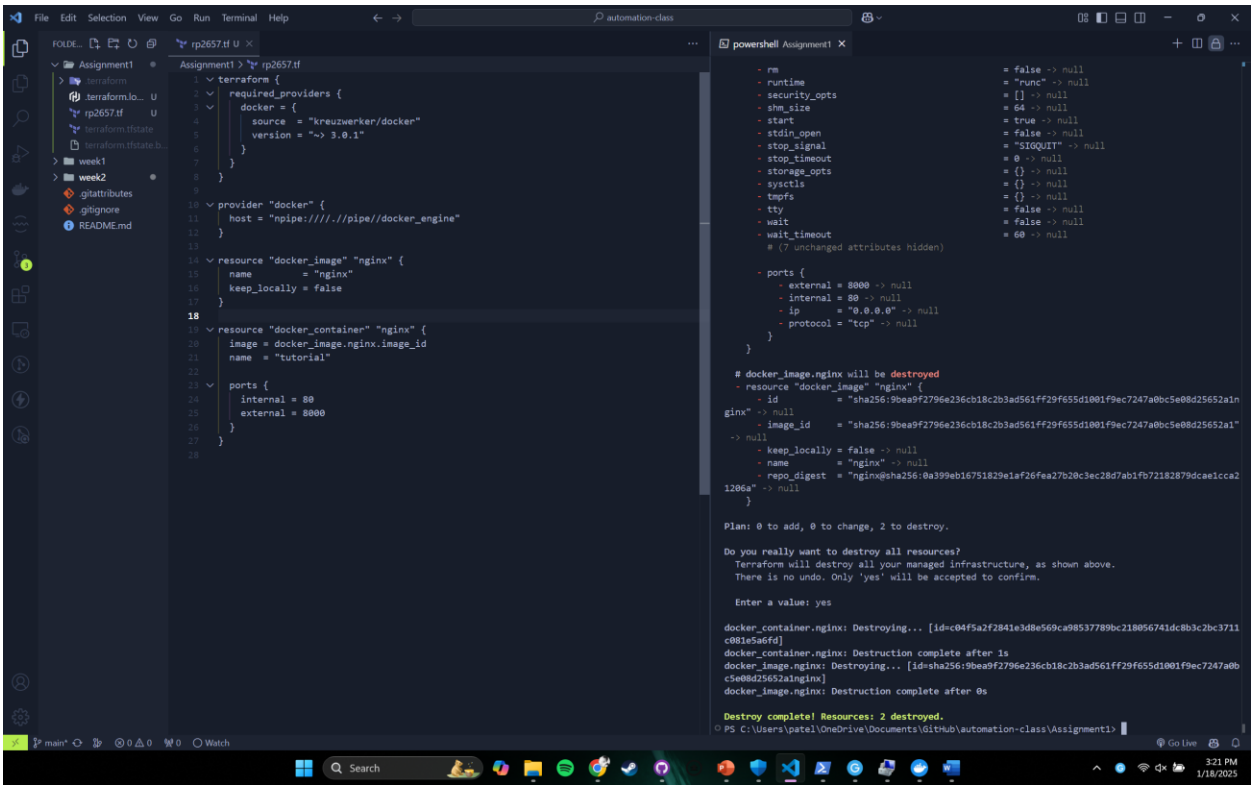
- Once the Terraform apply process is completed successfully, verify that the Nginx server is running
- Use a web browser or tools like curl to access the Nginx server running inside the Docker container by navigating to http://localhost:8000/



Cleanup

- After completing the verification, remove the resources provisioned by Terraform to avoid unnecessary costs and resource consumption
  `terraform destroy`
- Confirm the destruction of resources when prompted.




File Edit Selection View Go Run Terminal Help
automation-class
FOLDE...
Assignment1
.terraform
.terraform.lo... U
rp2657.tf U
terraform.tfstate
terraform.tfstate.b...
week1
week2
.gitattributes
.gitignore
README.md
rp2657.tf U
Assignment1 > rp2657.tf
terraform {
required_providers {
docker = {
source = "kreuzwerker/docker"
version = "~> 3.0.1"
}
}
}
provider "docker" {
host = "npipe:////.//pipe//docker_engine"
}
resource "docker_image" "nginx" {
name = "nginx"
keep_locally = false
}
resource "docker_container" "nginx" {
image = docker_image.nginx.image_id
name = "tutorial"
ports {
internal = 80
external = 8000
}
}
powershell Assignment1
- rm = false -> null
- runtime = "runc" -> null
- security_opts = [] -> null
- shm_size = 64 -> null
- start = true -> null
- stdin_open = false -> null
- stop_signal = "SIGQUIT" -> null
- stop_timeout = 0 -> null
- storage_opts = {} -> null
- sysctls = {} -> null
- tmpfs = {} -> null
- tty = false -> null
- wait = false -> null
- wait_timeout = 60 -> null
# (7 unchanged attributes hidden)
- ports {
- external = 8000 -> null
- internal = 80 -> null
- ip = "0.0.0.0" -> null
- protocol = "tcp" -> null
}
}
# docker_image.nginx will be destroyed
- resource "docker_image" "nginx" {
- id = "sha256:9bea9f2796e236cb18c2b3ad561ff29f655d1001f9ec7247a0bc5e08d25652a1nginx" -> null
- image_id = "sha256:9bea9f2796e236cb18c2b3ad561ff29f655d1001f9ec7247a0bc5e08d25652a1" -> null
- keep_locally = false -> null
- name = "nginx" -> null
- repo_digest = "nginx@sha256:0a399eb16751829e1af26fea27b20c3ec28d7ab1fb72182879dcae1cca21206a" -> null
}
Plan: 0 to add, 0 to change, 2 to destroy.
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.
Enter a value: yes
docker_container.nginx: Destroying... [id=c04f5a2f2841e3d8e569ca98537789bc218056741dc8b3c2bc3711c081e5a6fd]
docker_container.nginx: Destruction complete after 1s
docker_image.nginx: Destroying... [id=sha256:9bea9f2796e236cb18c2b3ad561ff29f655d1001f9ec7247a0bc5e08d25652a1nginx]
docker_image.nginx: Destruction complete after 0s
Destroy complete! Resources: 2 destroyed.
PS C:\Users\patel\OneDrive\Documents\GitHub\automation-class\Assignment1>
main*
Watch
Go Live
Search
3:21 PM
1/18/2025