

Assignment 1

Rishi (8972657)

Scripting and Automation

PROG8830 - Winter 2025 - Section 1

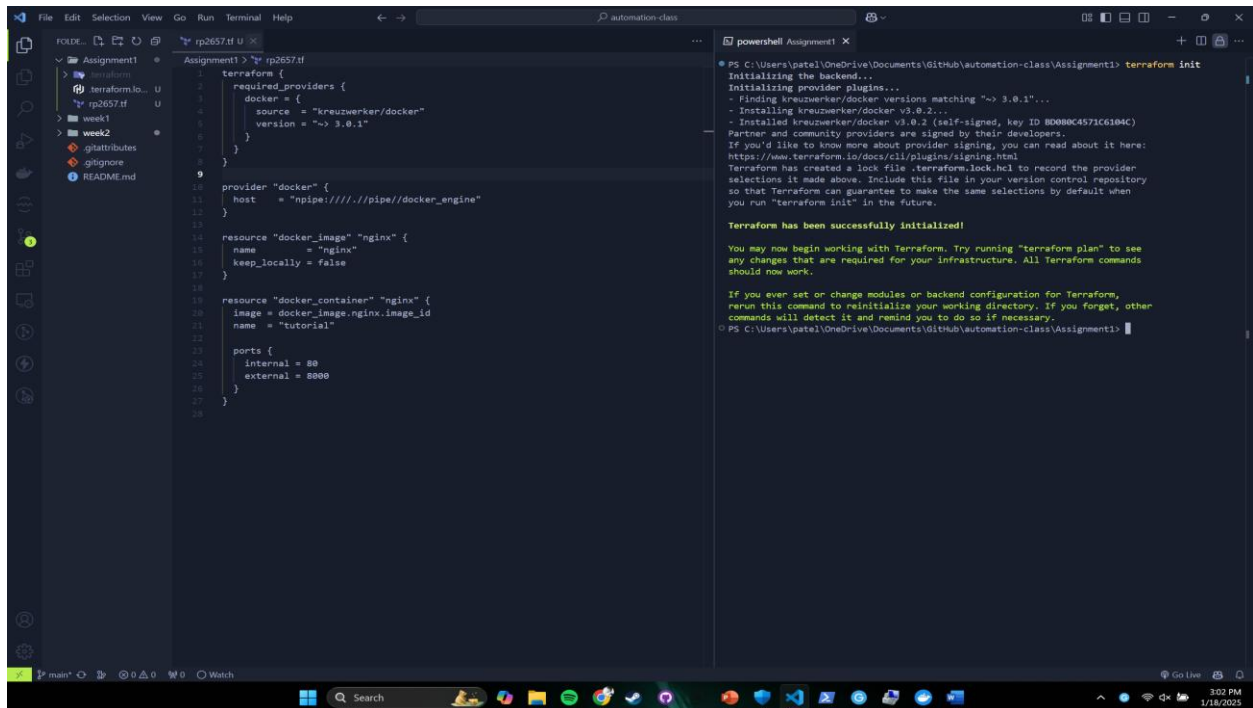
Prof. Mike Dabydeen

Jan 18, 2025

Initialize Terraform

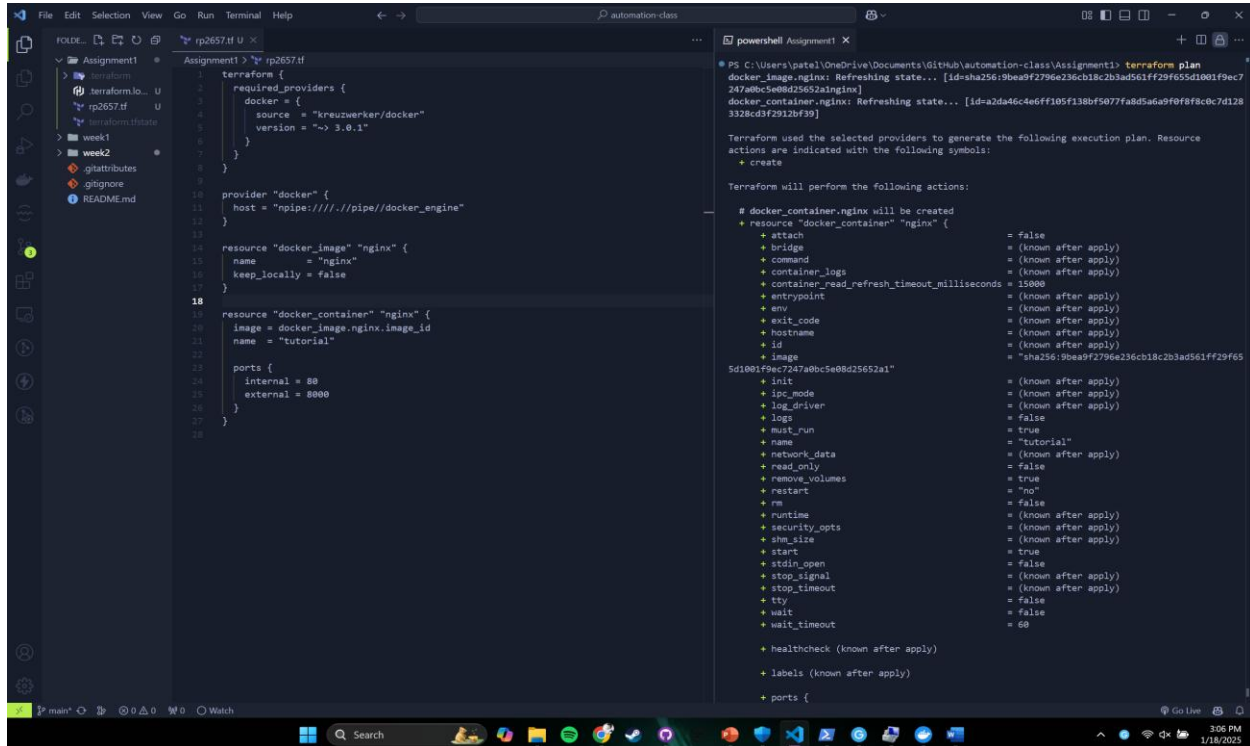
Initialize Terraform in your working directory using this command:

```
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



The screenshot shows a VS Code editor with two panels. The left panel displays a Terraform configuration file named `rp2657.tf` with the following content:

```
1 terraform {
2   required_providers {
3     docker = {
4       source = "kreuzwerker/docker"
5       version = "~> 3.0.1"
6     }
7   }
8 }
9
10 provider "docker" {
11   host = "npipe:////./pipe/docker_engine"
12 }
13
14 resource "docker_image" "nginx" {
15   name = "nginx"
16   keep_locally = false
17 }
18
19 resource "docker_container" "nginx" {
20   image = docker_image.nginx.image_id
21   name = "tutorial"
22
23   ports {
24     internal = 80
25     external = 8080
26   }
27 }
28
```

The right panel shows the output of the `terraform plan` command. It indicates that the `docker_image.nginx` and `docker_container.nginx` resources are being refreshed. The plan shows that the `docker_container.nginx` will be created with the following configuration:

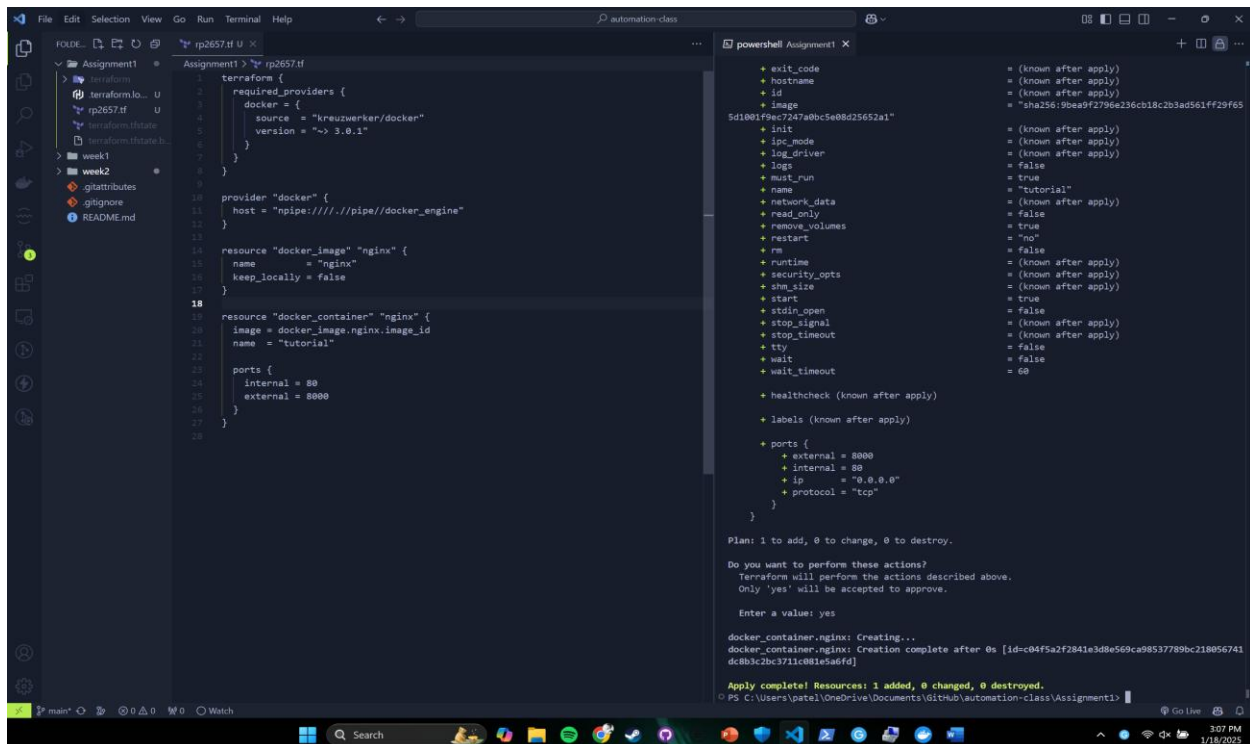
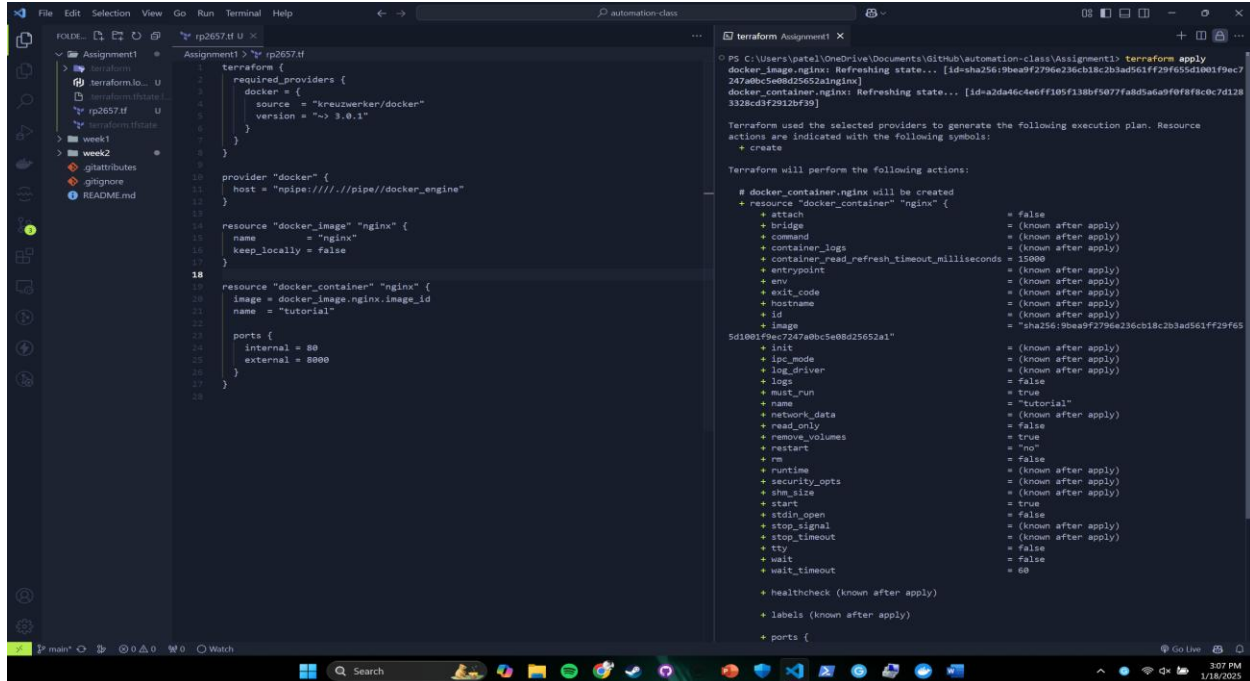
```
# docker_container.nginx will be created
+ resource "docker_container" "nginx" {
  + attach                = false
  + bridge                = (known after apply)
  + command               = (known after apply)
  + container_logs        = (known after apply)
  + container_read_refresh_timeout_milliseconds = 15000
  + entrypoint             = (known after apply)
  + env                  = (known after apply)
  + exit_code             = (known after apply)
  + hostname              = (known after apply)
  + id                    = (known after apply)
  + image                 = "sha256:9bea9f2796e236cb18c2b3ad561ff29f65
5d1001f9c7247a0bc5e08d25652a1"
  + init                  = (known after apply)
  + ipc_mode              = (known after apply)
  + log_driver            = (known after apply)
  + logs                  = false
  + name                  = "tutorial"
  + network_data          = (known after apply)
  + read_only             = false
  + remove_volumes        = true
  + restart               = "no"
  + runtime               = false
  + security_opts         = (known after apply)
  + shm_size              = (known after apply)
  + start                 = true
  + stdin_open            = false
  + stop_signal           = (known after apply)
  + stop_timeout          = (known after apply)
  + tty                   = false
  + wait                  = false
  + wait_timeout          = 60
  + healthcheck (known after apply)
  + labels (known after apply)
  + ports {

```

The status bar at the bottom indicates the file is named `rp2657.tf` and the workspace is `automation-class`. The system clock shows 3:06 PM on 1/18/2025.

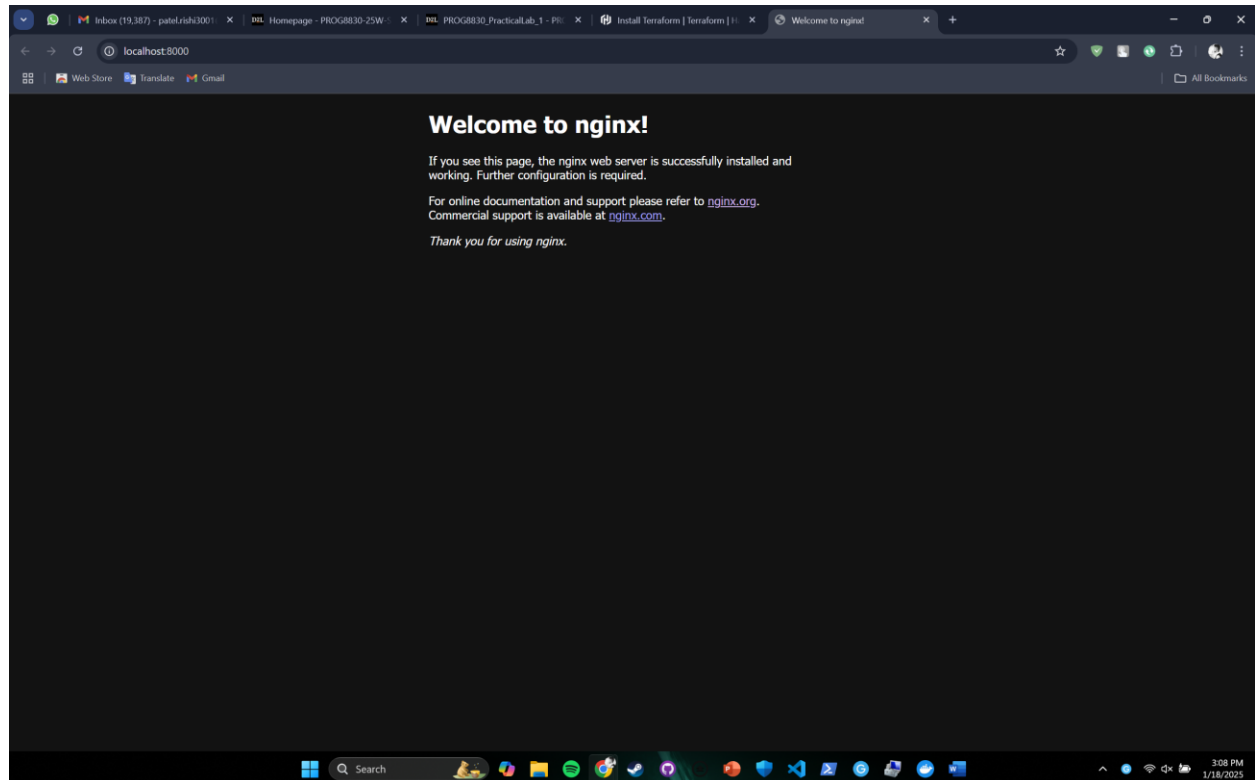
Apply the Terraform configuration to provision the resources using the command:

```
terraform apply
```



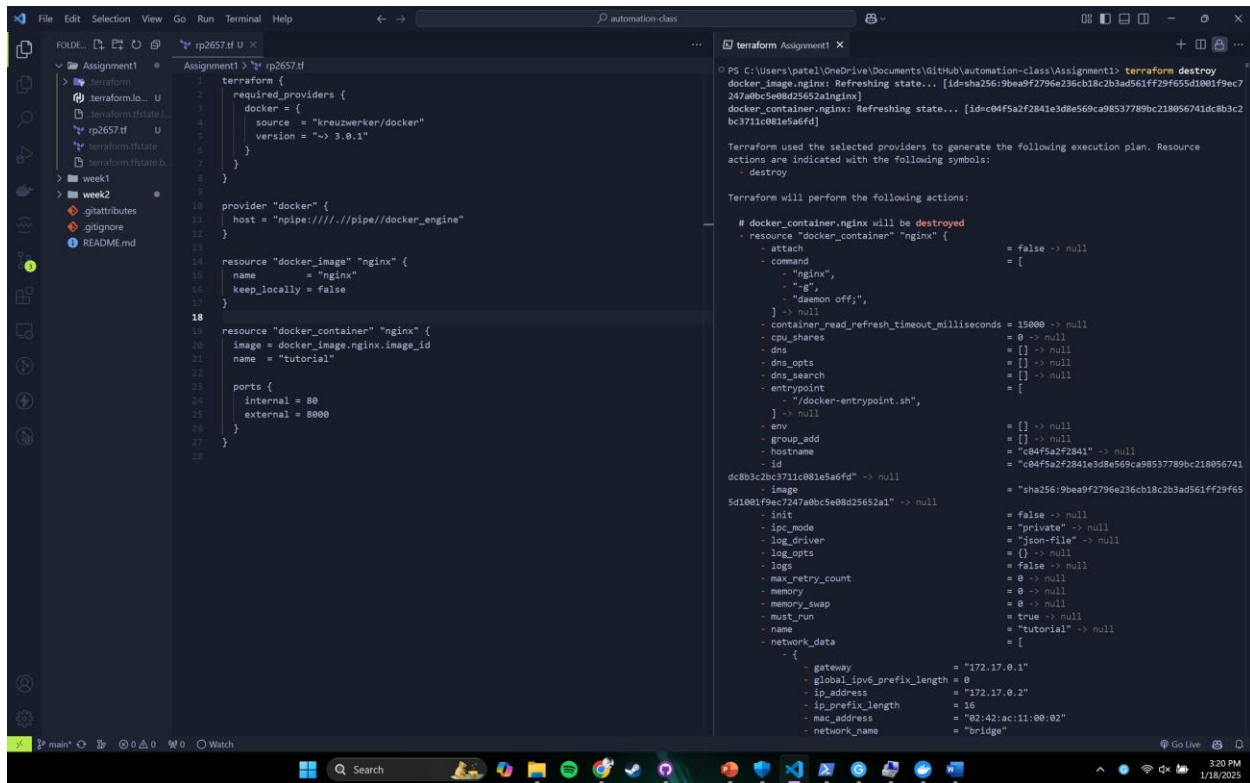
Verify Nginx Server

- 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.



The screenshot shows a VS Code editor with two files open. The left file, `rp2657.tf`, contains Terraform configuration for a Docker container named 'nginx'. The right file, `terraform Assignment1 X`, shows the output of the `terraform destroy` command, including the execution plan and the list of resources to be destroyed.

```
1 terraform {
2   required_providers {
3     docker = {
4       source = "kreuzwerker/docker"
5       version = "~> 3.0.1"
6     }
7   }
8 }
9
10 provider "docker" {
11   host = "npipe:////./pipe/docker_engine"
12 }
13
14 resource "docker_image" "nginx" {
15   name = "nginx"
16   keep_locally = false
17 }
18
19 resource "docker_container" "nginx" {
20   image = docker_image.nginx.image_id
21   name = "tutorial"
22
23   ports {
24     internal = 80
25     external = 8080
26   }
27 }
28
```

```
PS C:\Users\pate\OneDrive\Documents\Github\automation-class\Assignment1> terraform destroy
docker_image.nginx: Refreshing state... [id=sha256:9bea9f2796e236cb18c2b3ad561ff29f655d1001f9ec7
24f9ebc5e8d25652a1nginx]
docker_container.nginx: Refreshing state... [id=c04f5a2f2841e3d8e569ca98537789bc218056741dc8b3c2
bc3711c081e5a6fd]

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

Terraform will perform the following actions:

# docker_container.nginx will be destroyed
- resource "docker_container" "nginx" {
  - attach = false -> null
  - command = [
    - "nginx",
    - "g",
    - "daemon off;",
  ] -> null
  - container_read_refresh_timeout_milliseconds = 15000 -> null
  - cpu_shares = 0 -> null
  - dns = [] -> null
  - dns_opts = [] -> null
  - dns_search = [] -> null
  - endpoint = [
    - "/docker-entrypoint.sh",
  ] -> null
  - env = [] -> null
  - group_add = [] -> null
  - hostname = "c04f5a2f2841" -> null
  - id = "c04f5a2f2841e3d8e569ca98537789bc218056741dc8b3c2bc3711c081e5a6fd" -> null
  - image = "sha256:9bea9f2796e236cb18c2b3ad561ff29f655d1001f9ec724f9ebc5e8d25652a1" -> null
  - init = false -> null
  - ipc_mode = "private" -> null
  - log_driver = "json-file" -> null
  - log_opts = {} -> null
  - logs = false -> null
  - max_retry_count = 0 -> null
  - memory = 0 -> null
  - memory_swap = 0 -> null
  - must_run = true -> null
  - name = "tutorial" -> null
  - network_data = [
    - {
      - gateway = "172.17.0.1"
      - global_ipv6_prefix_length = 0
      - ip_address = "172.17.0.2"
      - ip_prefix_length = 16
      - mac_address = "92:42:ac:11:00:02"
      - network_name = "bridge"
    }
  ]
}
```

