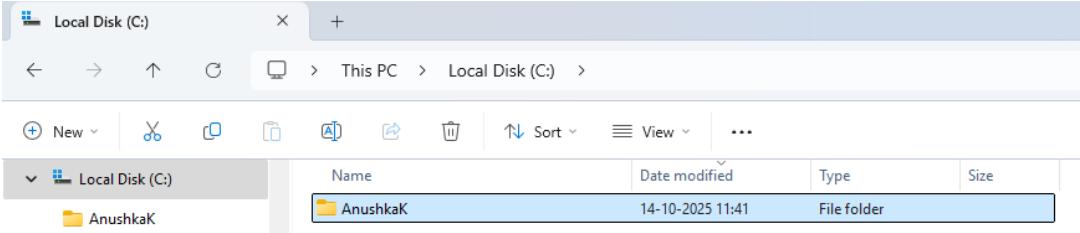


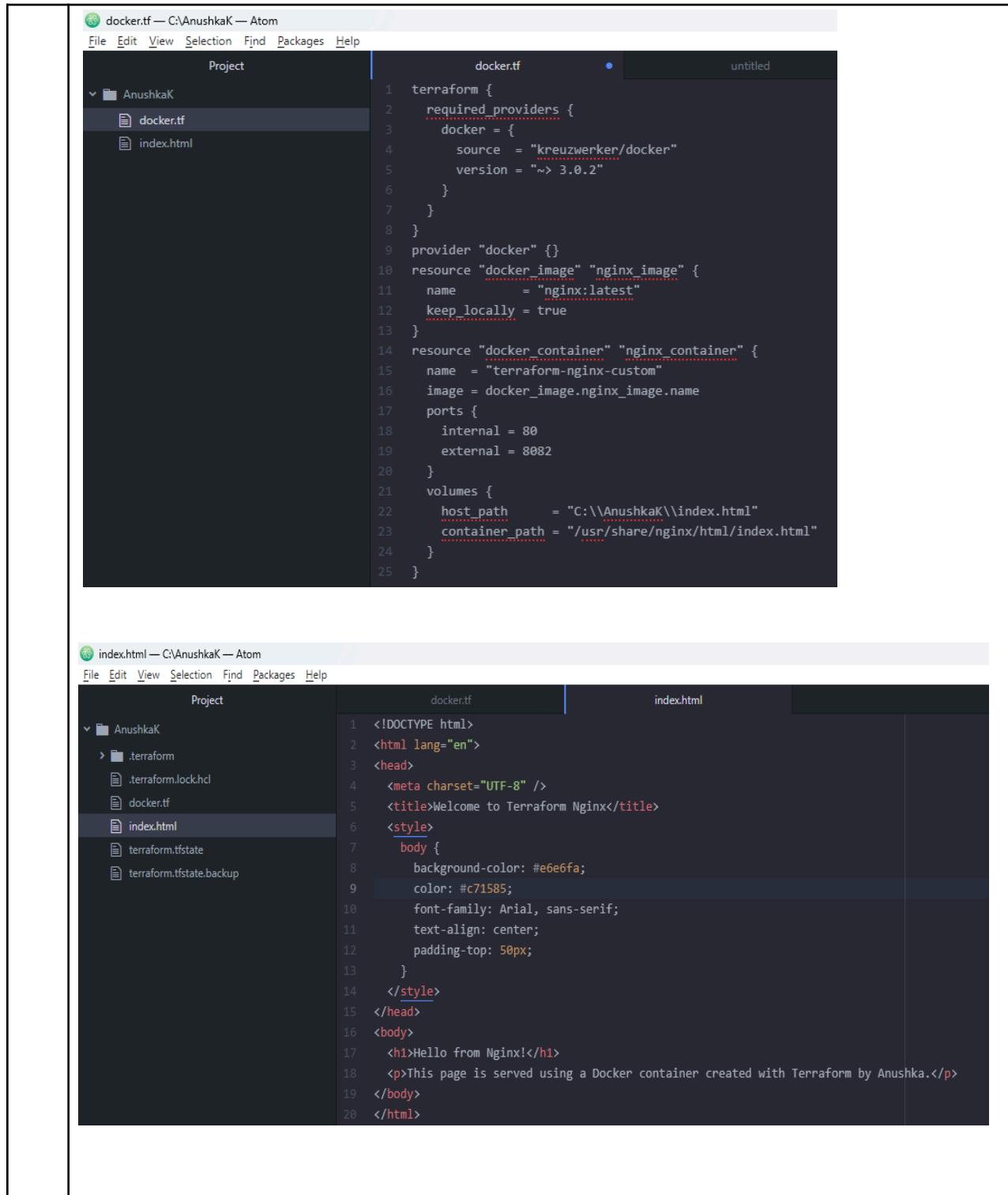
St. Francis Institute of Technology
Borivali (West), Mumbai-400103
Department of Information Technology
ADL Practical Exam

Name: Anushka Kotal	Date: 14\10\25
Class: TE INFT A	Exam seat no:
PID: 231055	Time: 11:15 to 1:15
Roll No: 52	No. of Pages:8
Duration: One Hour	

Sr No	Problem Statement : To deploy and manage and NGINX web server container using Terraform with Docker as the provider
1.	<p>Step 1 (creation/configuration/initial step(s)):</p> <p>Observation: Here we created folder name AnushkaK in local disk.</p> <p>Screenshot 1:</p>  <p>The screenshot shows a Windows File Explorer window. The address bar says 'Local Disk (C:)'. The left sidebar shows 'This PC' and 'Local Disk (C:)'. The main area shows a folder named 'AnushkaK' under 'Local Disk (C:)'. The file list table has columns: Name, Date modified, Type, and Size. The 'AnushkaK' folder is listed with a date of 14-10-2025 11:41 and type 'File folder'.</p>
2.	<p>Step 2 (internal step 1):</p> <p>Observation: Here opened Atom in that i opened my folder name AnushkaK. In that i created two files ie. docker.tf and index.html. In docker.tf we wrote the code to create docker image_nginx image. And in index.html we wrote the code to show the text in localhost:8082 in nginx server and put the css.</p> <p>Screenshot 2:</p>

St. Francis Institute of Technology
Borivali (West), Mumbai-400103
Department of Information Technology
ADL Practical Exam

Name: Anushka Kotal	Date: 14\10\25
Class: TE INFT A	Exam seat no:
PID: 231055	Time: 11:15 to 1:15
Roll No: 52	No. of Pages:8
Duration: One Hour	



The screenshot shows the Atom code editor interface with two tabs open: "docker.tf" and "index.html".

docker.tf (Active Tab):

```

1 terraform {
2   required_providers {
3     docker = {
4       source  = "kreuzwerker/docker"
5       version = "~> 3.0.2"
6     }
7   }
8 }
provider "docker" {}
resource "docker_image" "nginx_image" {
11   name      = "nginx:latest"
12   keep_locally = true
}
resource "docker_container" "nginx_container" {
15   name      = "terraform-nginx-custom"
16   image     = docker_image.nginx_image.name
17   ports {
18     internal = 80
19     external = 8082
20   }
21   volumes {
22     host_path    = "C:\\AnushkaK\\index.html"
23     container_path = "/usr/share/nginx/html/index.html"
24   }
25 }

```

index.html:

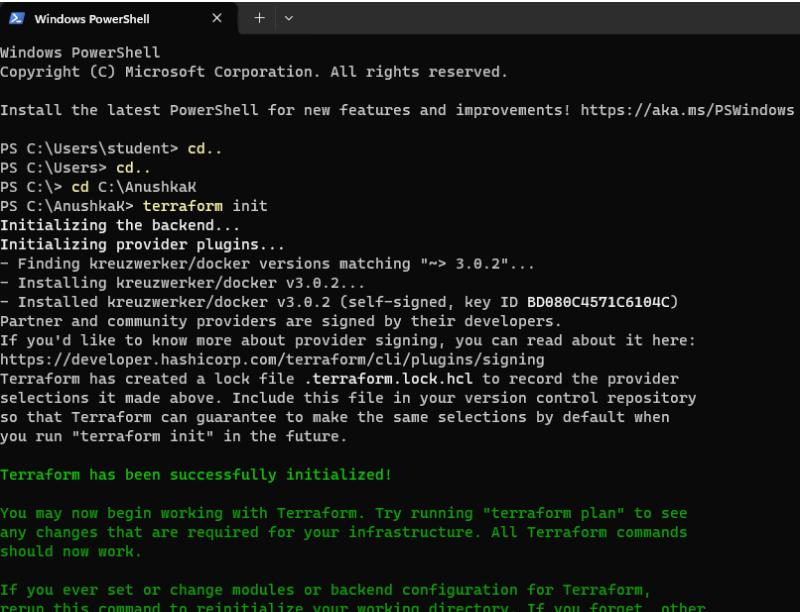
```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <title>Welcome to Terraform Nginx</title>
6   <style>
7     body {
8       background-color: #e6e6fa;
9       color: #c71585;
10      font-family: Arial, sans-serif;
11      text-align: center;
12      padding-top: 50px;
13    }
14   </style>
15 </head>
16 <body>
17   <h1>Hello from Nginx!</h1>
18   <p>This page is served using a Docker container created with Terraform by Anushka.</p>
19 </body>
20 </html>

```

St. Francis Institute of Technology
Borivali (West), Mumbai-400103
Department of Information Technology
ADL Practical Exam

Name: Anushka Kotal	Date: 14\10\25
Class: TE INFT A	Exam seat no:
PID: 231055	Time: 11:15 to 1:15
Roll No: 52	No. of Pages:8
Duration: One Hour	

3.	<p>Step 3 (internal step 2):</p> <p>Observation: Here we opened Windows powershell in that i put my directory of my folder ie. C:\AnushkaK then we run the terraform commands ie. terraform init, terraform plan, terraform apply</p> <p>Screenshot 3:</p>  <pre> Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved. Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows PS C:\Users\student> cd.. PS C:\Users> cd.. PS C:\> cd C:\AnushkaK PS C:\AnushkaK> terraform init Initializing the backend... Initializing provider plugins... - Finding kreuzwerker/docker versions matching "> 3.0.2"... - Installing kreuzwerker/docker v3.0.2... - Installed kreuzwerker/docker v3.0.2 (self-signed, key ID BD080C4571C6104C) Partner and community providers are signed by their developers. If you'd like to know more about provider signing, you can read about it here: https://developer.hashicorp.com/terraform/cli/plugins/signing Terraform has created a lock file .terraform.lock.hcl to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future. 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 </pre>
----	--

St. Francis Institute of Technology
Borivali (West), Mumbai-400103
Department of Information Technology
ADL Practical Exam

Name: Anushka Kotal	Date: 14\10\25
Class: TE INFT A	Exam seat no:
PID: 231055	Time: 11:15 to 1:15
Roll No: 52	No. of Pages:8
Duration: One Hour	

```
PS C:\AnushkaK> terraform plan

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_container will be created
+ resource "docker_container" "nginx_container" {
    + 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                           = "nginx:latest"
    + init                            = (known after apply)
    + ipc_mode                         = (known after apply)
    + log_driver                      = (known after apply)
    + logs                            = false
    + must_run                        = true
    + name                            = "terraform-nginx-custom"
    + network_data                    = (known after apply)
    + read_only                        = false
    + remove_volumes                 = true
    + restart                          = "no"
    + rm                             = false
    + runtime                          = (known after apply)
    + 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

    + ip      = "0.0.0.0"
    + protocol = "tcp"
}

+ volumes {
    + container_path = "/usr/share/nginx/html/index.html"
    + host_path     = "C:\\\\AnushkaK\\\\index.html"
    # (2 unchanged attributes hidden)
}
}

# docker_image.nginx_image will be created
+ resource "docker_image" "nginx_image" {
    + id      = (known after apply)
    + image_id = (known after apply)
    + keep_locally = true
    + name    = "nginx:latest"
    + repo_digest = (known after apply)
}

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

St. Francis Institute of Technology
Borivali (West), Mumbai-400103
Department of Information Technology
ADL Practical Exam

Name: Anushka Kotal	Date: 14\10\25
Class: TE INFT A	Exam seat no:
PID: 231055	Time: 11:15 to 1:15
Roll No: 52	No. of Pages:8
Duration: One Hour	

```
PS C:\AnushkaK> terraform apply

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

Terraform will perform the following actions:

# docker_container.nginx_container will be created
+ resource "docker_container" "nginx_container" {
    + 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           = "nginx:latest"
    + init            = (known after apply)
    + ipc_mode        = (known after apply)
    + log_driver      = (known after apply)
    + logs            = false
    + must_run        = true
    + name            = "terraform-nginx-custom"
    + network_data   = (known after apply)
    + read_only       = false
    + remove_volumes = true
    + restart         = "no"
    + rm              = false
    + runtime         = (known after apply)
    + 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

    + volumes {
        + container_path = "/usr/share/nginx/html/index.html"
        + host_path      = "C:\\\\AnushkaK\\\\index.html"
        # (2 unchanged attributes hidden)
    }
}

# docker_image.nginx_image will be created
+ resource "docker_image" "nginx_image" {
    + id          = (known after apply)
    + image_id    = (known after apply)
    + keep_locally = true
    + name        = "nginx:latest"
    + repo_digest = (known after apply)
}

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.

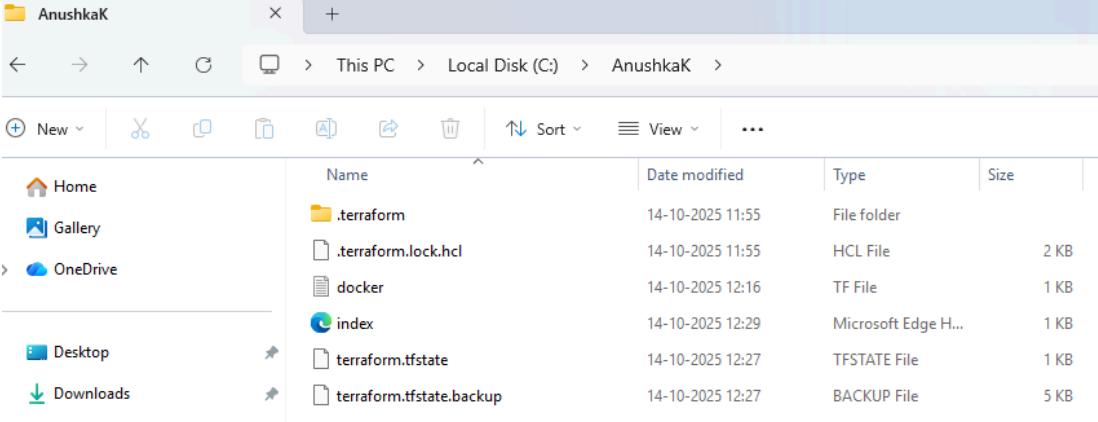
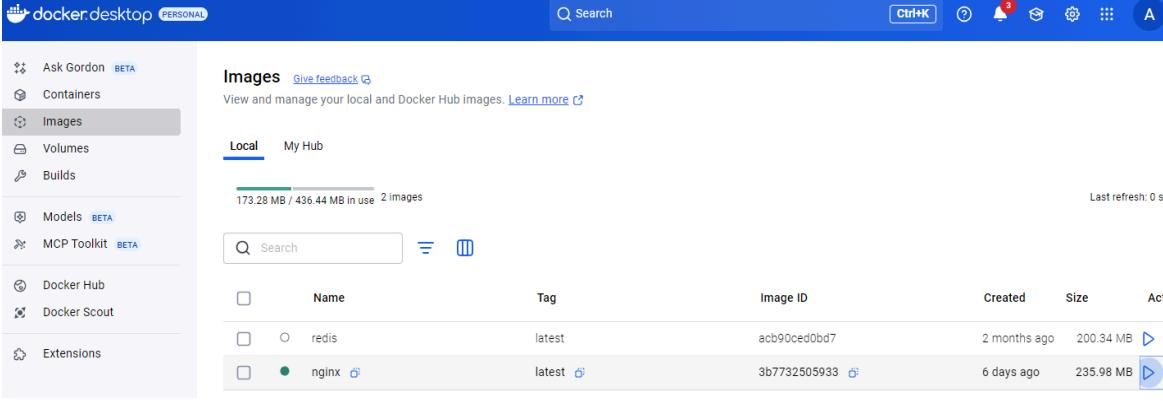
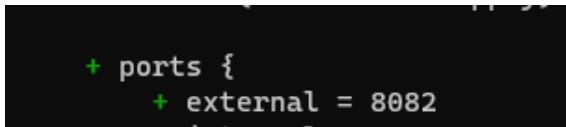
Enter a value: yes

docker_image.nginx_image: Creating...
docker_image.nginx_image: Creation complete after 7s [id=sha256:3b7732505933ca591ce4a6d860cb713ad96
3486a0d6nginx:latest]
docker_container.nginx_container: Creating...
docker_container.nginx_container: Creation complete after 1s [id=21f6df30b79e761fc3c798624cf4055ec
fcbe4741a]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\AnushkaK> |
```

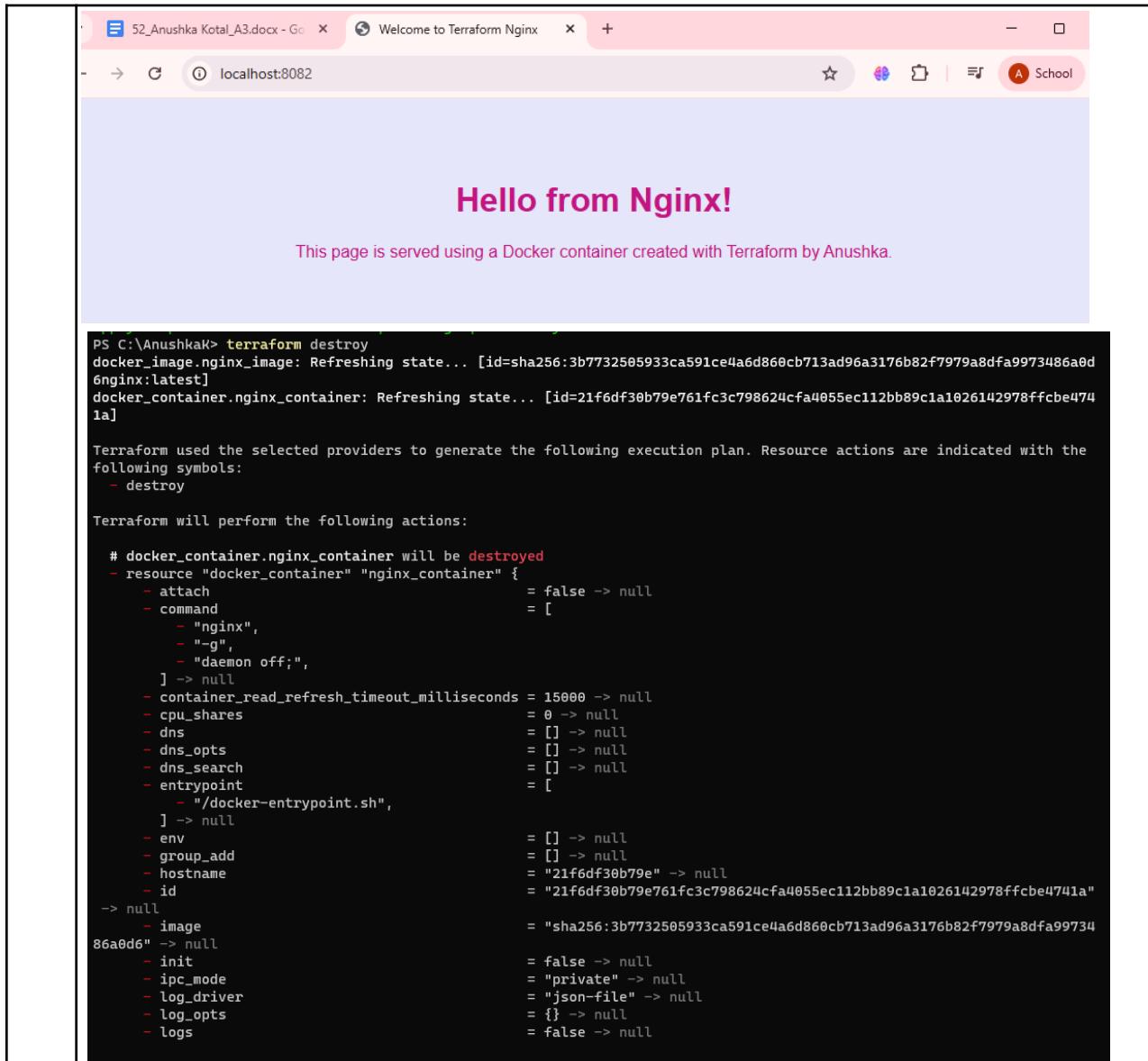
St. Francis Institute of Technology
Borivali (West), Mumbai-400103
Department of Information Technology
ADL Practical Exam

Name: Anushka Kotal	Date: 14\10\25
Class: TE INFT A	Exam seat no:
PID: 231055	Time: 11:15 to 1:15
Roll No: 52	No. of Pages:8
Duration: One Hour	

	
4.	<p>Step 4 (internal step 3):</p> <p>Observation: Here we opened Docker Desktop and the terraform commands we applied to create docker_image.nginx_image.</p> <p>Screenshot 4:</p> 
5.	<p>Step 5 (stop/delete step) :</p> <p>Observation: Here in the browser we opened the localhost which was given in windows powershell while executing terraform apply command. In the browser we typed localhost:8082 and we go the output showing Hello from Nginx!. Then we applied terraform destroy to terminate the docker image_ngnix image.</p> <p>Screenshot 5:</p> 

St. Francis Institute of Technology
Borivali (West), Mumbai-400103
Department of Information Technology
ADL Practical Exam

Name: Anushka Kotal	Date: 14\10\25
Class: TE INFT A	Exam seat no:
PID: 231055	Time: 11:15 to 1:15
Roll No: 52	No. of Pages: 8
Duration: One Hour	



The screenshot shows a web browser window titled "Welcome to Terraform Nginx" at "localhost:8082". The page content includes the text "Hello from Nginx!" and "This page is served using a Docker container created with Terraform by Anushka." Below the browser is a terminal window displaying the output of a Terraform destroy command. The terminal output shows the state of resources like docker_image.nginx_image and docker_container.nginx_container, followed by the execution plan and actions that will be performed.

```
PS C:\AnushkaK> terraform destroy
docker_image.nginx_image: Refreshing state... [id=sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6nginx:latest]
docker_container.nginx_container: Refreshing state... [id=21f6df30b79e761fc3c798624cfa4055ec112bb89c1a1026142978ffcbef4741a]

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_container will be destroyed
- resource "docker_container" "nginx_container" {
    - 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
    - entrypoint
        = [
            - "/docker-entrypoint.sh",
        ] -> null
    - env
        = [] -> null
    - group_add
        = [] -> null
    - hostname
        = "21f6df30b79e761fc3c798624cfa4055ec112bb89c1a1026142978ffcbef4741a"
    - id
        -> null
    - image
        = "sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6nginx:latest" -> null
    - init
        = false -> null
    - ipc_mode
        = "private" -> null
    - log_driver
        = "json-file" -> null
    - log_opts
        = {} -> null
    - logs
        = false -> null
}
```

St. Francis Institute of Technology
Borivali (West), Mumbai-400103
Department of Information Technology
ADL Practical Exam

Name: Anushka Kotal	Date: 14\10\25
Class: TE INFT A	Exam seat no:
PID: 231055	Time: 11:15 to 1:15
Roll No: 52	No. of Pages:8
Duration: One Hour	

<pre> # docker_image.nginx_image will be destroyed resource "docker_image" "nginx_image" { - id = "sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6nginx:latest" -> null - image_id = "sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6" -> null - keep_locally = true -> null - name = "nginx:latest" -> null - repo_digest = "nginx@sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6" -> 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_container: Destroying... [id=21f6df30b79e761fc3c798624cfa4055ec112bb89c1a1026142978ffcbef4741a] docker_container.nginx_container: Destruction complete after 0s docker_image.nginx_image: Destroying... [id=sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6nginx:latest] docker_image.nginx_image: Destruction complete after 0s Destroy complete! Resources: 2 destroyed. PS C:\AnushkaK> </pre>
Note: Adjust SS to fit in the table row. SS should include username and timestamp. Add rows if needed