

**BCDV 4033**

**DevOps & Blockchain Admin**

**Lab-4 - Terraform**

**Name - Aratrika Mukherjee**

**Student ID - 101413594**

# 1. Install Terraform on local system

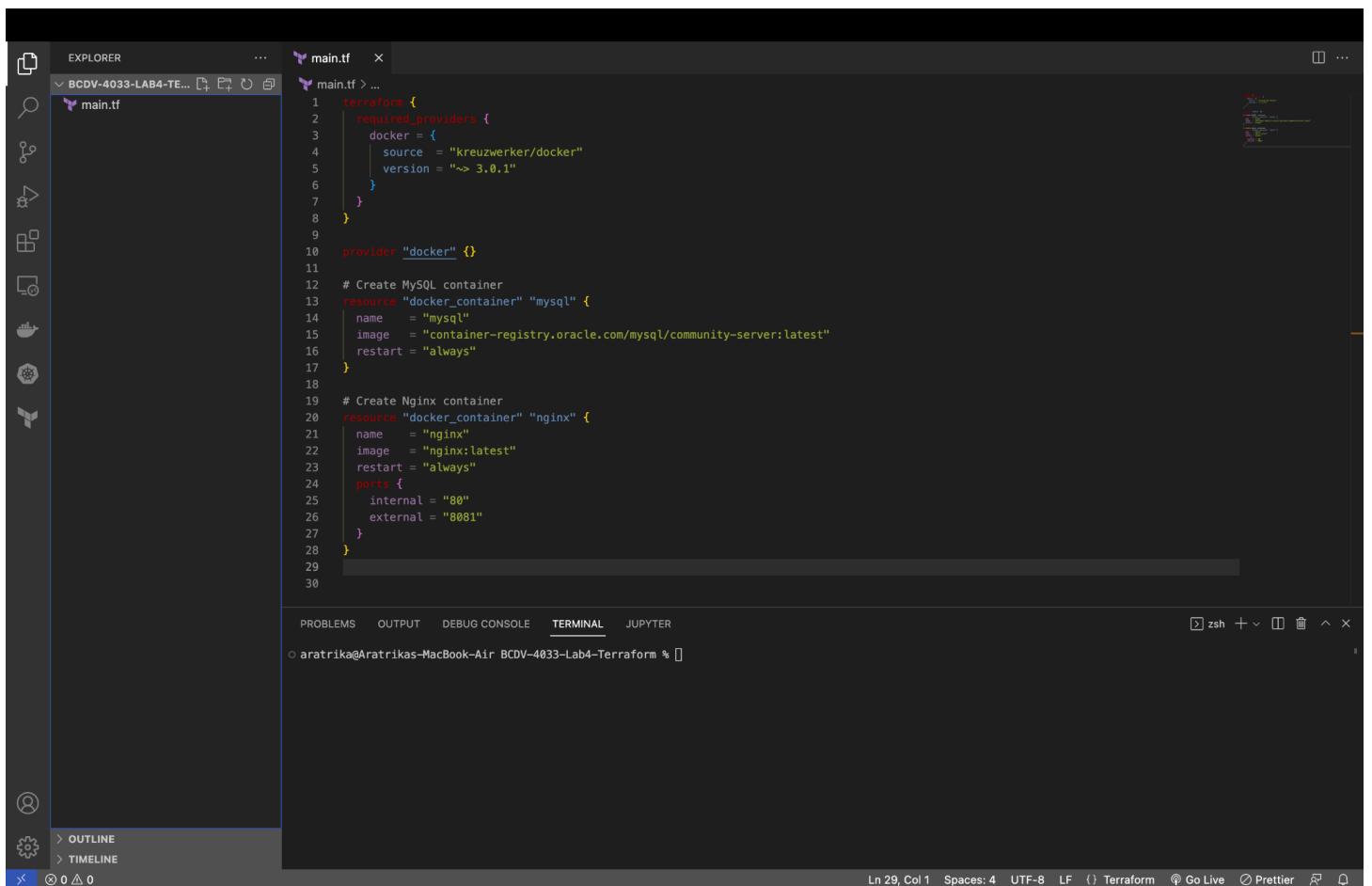
```
[aratrika@Aratrikas-MacBook-Air ~ % brew tap hashicorp/tap
Running `brew update --auto-update'...
=> Auto-updated Homebrew!
Updated 3 taps (homebrew/services, homebrew/core and homebrew/cask).
=> New Formulae
dhall-toml      gimmecert      mlx          sqlboiler
edbrowse        glasskube      neosync      sshpass
envio           mariadb@11.2   podman-tui  vulkan-profiles
=> New Casks
acrem          dosbox-staging  kit          notesollama
amnezia vpn    endless-sky-high-dpi  overlayed
deelay         fixkey        segger-ozone
dnsmonitor     jordanbaird-ice

You have 20 outdated formulae and 1 outdated cask installed.

=> Tapping hashicorp/tap
Cloning into '/opt/homebrew/Library/Taps/hashicorp/homebrew-tap'...
remote: Enumerating objects: 4160, done.
remote: Counting objects: 100% (4160/4160), done.
remote: Compressing objects: 100% (1474/1474), done.
remote: Total 4160 (delta 2728), reused 4834 (delta 2651), pack-reused 0
Receiving objects: 100% (4160/4160), 753.35 KiB | 4.92 MiB/s, done.
Resolving deltas: 100% (2728/2728), done.
Tapped 10 casks and 28 formulae (87 files, 1MB).
[aratrika@Aratrikas-MacBook-Air ~ % brew install hashicorp/tap/terraform
  Fetching hashicorp/tap/terraform
=> Downloading https://releases.hashicorp.com/terraform/1.7.4/terraform_1.7.4.d
#####
#-----#
##### 100.0%
=> Installing terraform from hashicorp/tap
Warning: Your Xcode (15.0.1) is outdated.
Please update to Xcode 15.1 (or delete it).
Xcode can be updated from the App Store.

Warning: Cask terraform was renamed to homebrew/core/terraform.
  /opt/homebrew/Cellar/terraform/1.7.4: 3 files, 88.6MB, built in 4 seconds
=> Running 'brew cleanup terraform'...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew').
[aratrika@Aratrikas-MacBook-Air ~ % terraform -version
Terraform v1.7.4
on darwin_arm64
[aratrika@Aratrikas-MacBook-Air ~ % ]
```

## 2. Create a tf configuration for running docker resource



```
1  terraform {
2      required_providers {
3          docker = {
4              source  = "kreuzwerker/docker"
5              version = ">= 3.0.1"
6          }
7      }
8  }
9
10 provider "docker" {}
11
12 # Create MySQL container
13 resource "docker_container" "mysql" {
14     name    = "mysql"
15     image   = "container-registry.oracle.com/mysql/community-server:latest"
16     restart = "always"
17 }
18
19 # Create Nginx container
20 resource "docker_container" "nginx" {
21     name    = "nginx"
22     image   = "nginx:latest"
23     restart = "always"
24     ports {
25         internal = "80"
26         external = "8081"
27     }
28 }
29
30
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

aratrika@Aratrikas-MacBook-Air BCDV-4033-Lab4-Terraform % ]

Ln 29, Col 1 Spaces: 4 UTF-8 LF ⚙ Terraform ⚙ Go Live ⚙ Prettier ⚙ ⚙

### 3. Apply the initial resources

The screenshot shows the VS Code interface with the Terraform extension. The Explorer sidebar shows a project named "BCDV-4033-LAB4-TERRAFORM" containing ".terraform", ".terraform.lock.hcl", and "main.tf". The main editor tab is "main.tf", displaying the following code:

```
1 terraform {  
2   required_providers {  
3     docker = {  
4       source  = "kreuzwerker/docker"  
5       version = "~> 3.0.1"  
6     }  
7   }  
8 }  
9  
10 provider "docker" {}  
11  
12 # Create MySQL container  
13 resource "docker_container" "mysql" {  
14   name    = "mysql"  
15   image   = "container-registry.oracle.com/mysql/community-server:latest"  
16   restart = "always"  
17 }  
18  
19 # Create Nginx container  
20 resource "docker_container" "nginx" {  
21   name    = "nginx"  
22   image   = "nginx:latest"
```

The terminal tab shows the command "terraform init" being run:

```
● aratrika@Aratrikas-MacBook-Air BCDV-4033-Lab4-Terraform % terraform init  
Initializing the backend...  
Initializing provider plugins...  
- Finding kreuzwerker/docker versions matching "~> 3.0.1"...  
- 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://www.terraform.io/docs/cli/plugins/signing.html  
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!
```

The status bar at the bottom indicates "Ln 29, Col 1 Spaces: 4 UTF-8 LF {} Terraform ⚡ Go Live ⚡ Prettier ⚡".

The screenshot shows the VS Code interface with the Terraform extension. The Explorer sidebar shows a project named "BCDV-4033-LAB4-TE..." containing ".terraform", ".terraform.lock.hcl", "main.tf", and "terraform.tfstate". The main editor tab is "main.tf", displaying the same code as the previous screenshot.

The terminal tab shows the command "terraform apply" being run:

```
● aratrika@Aratrikas-MacBook-Air BCDV-4033-Lab4-Terraform % terraform apply  
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.mysql will be created  
+ resource "docker_container" "mysql" {  
  + attach  
  + bridge  
  + command  
  + container_logs  
  + container_read_refresh_timeout_milliseconds = 15000  
  + entrypoint  
  + env  
  + exit_code  
  + hostname  
  + id  
  + image  
  + init  
  + ipc_mode  
  + log_driver  
  + logs  
  + mount_run  
  + name  
  + network_data  
  + read_only
```

The status bar at the bottom indicates "Ln 29, Col 1 Spaces: 4 UTF-8 LF {} Terraform ⚡ Go Live ⚡ Prettier ⚡".

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows files: .terraform, .terraform.lock.hcl, main.tf, and terraform.tfstate.
- main.tf** (selected):

```
1 terraform {
2   required_providers {
3     docker = {
4       source  = "kreuzwerker/docker"
5       version = ">= 3.0.1"
6     }
7   }
8 }
9 provider "docker" {}
10
11 # Create MySQL container
12 resource "docker_container" "mysql" {
13   name    = "mysql"
14   image   = "container-registry.oracle.com/mysql/community-server:latest"
15   restart = "always"
16 }
17
18 # Create Nginx container
19 resource "docker_container" "nginx" {
20   name    = "nginx"
21   image   = "nginx:latest"
22 }
```
- TERMINAL**:

```
+ tty                         = false
+ wait                        = false
+ wait_timeout                = 60

+ ports {
+   + external = 8081
+   + internal = 80
+   + ip      = "0.0.0.0"
+   + protocol = "tcp"
}
}

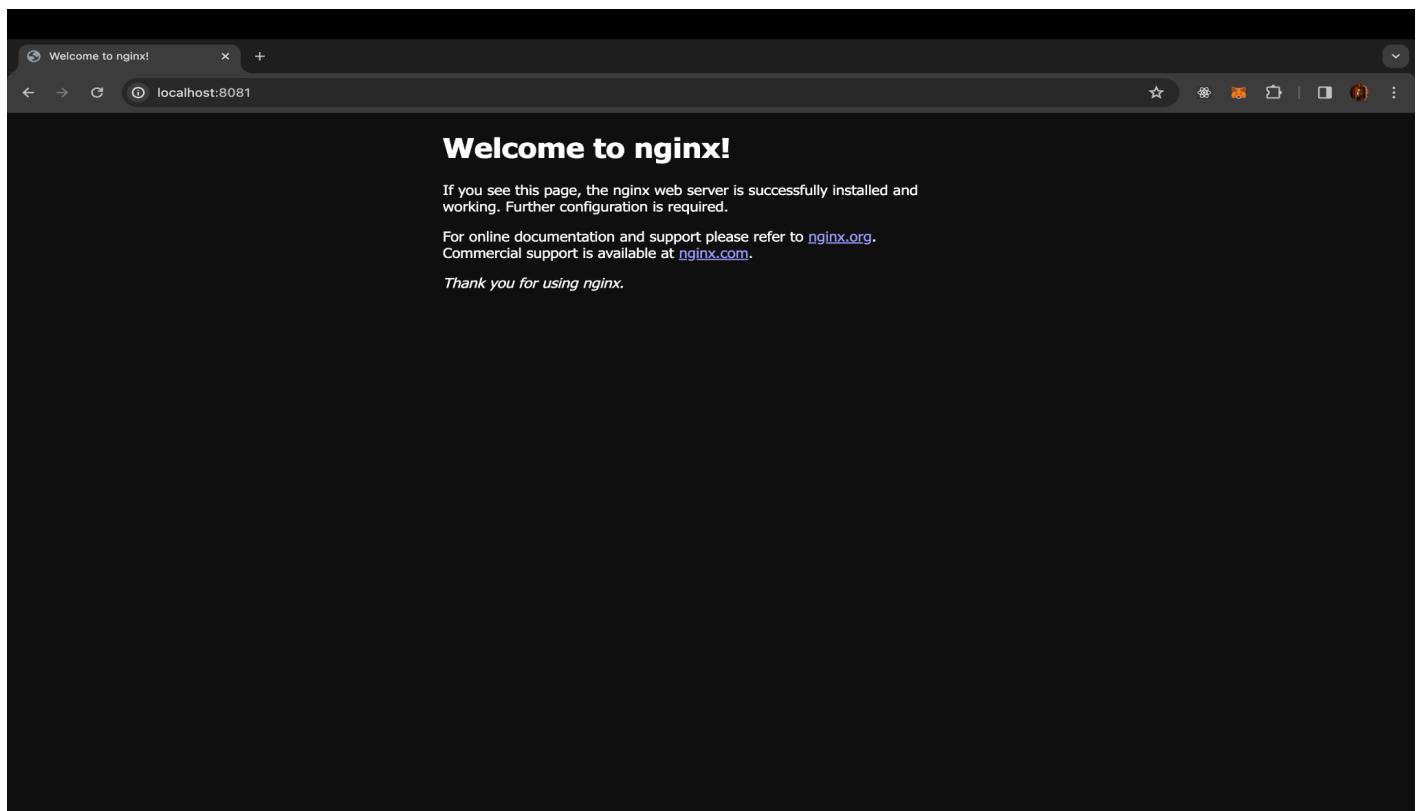
Plan: 2 to add, 0 to change, 0 to destroy.

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

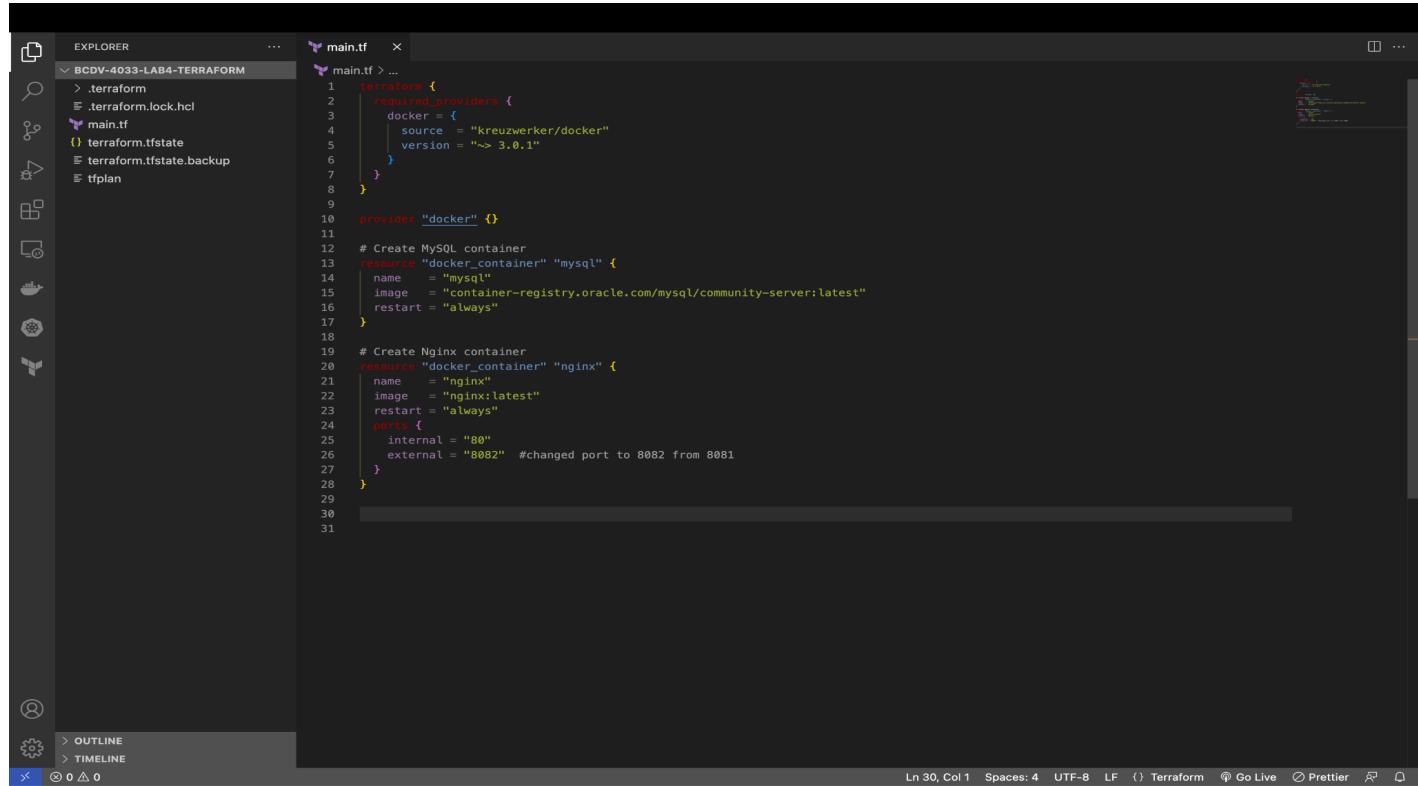
docker_container.mysql: Creating...
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 4s [id=60f6bc2defe55a819204c88f3723dd3b307e46954713fd070b4170f5eb324be4]
docker_container.mysql: Still creating... [10s elapsed]
docker_container.mysql: Creation complete after 14s [id=8d05d07f9fb7c5d97ac8a02ca84ef62068b2431826d0fc9777c10055d9adcfb3]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
aratrika@Aratrikas-MacBook-Air:~/BCDV-4033-Lab4-Terraform%
```
- STATUS BAR**: Ln 29, Col 1 | Spaces: 4 | UTF-8 | LF | Terraform | Go Live | Prettier



## 4. Change infrastructure

Changed the nginx port to 8082 from 8081



The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER**: Shows a folder named "BCDV-4033-LAB4-TERRAFORM" containing files: .terraform, .terraform.lock.hcl, main.tf, terraform.tfstate, terraform.tfstate.backup, and tfplan.
- main.tf**: The active code editor tab contains the following Terraform configuration:

```
1 terraform {
2   required_providers {
3     docker = {
4       source = "kreuzwerker/docker"
5       version = "~> 3.0.1"
6     }
7   }
8 }
9
10 provider "docker" {}
11
12 # Create MySQL container
13 resource "docker_container" "mysql" {
14   name    = "mysql"
15   image   = "container-registry.oracle.com/mysql/community-server:latest"
16   restart = "always"
17 }
18
19 # Create Nginx container
20 resource "docker_container" "nginx" {
21   name    = "nginx"
22   image   = "nginx:latest"
23   restart = "always"
24   ports {
25     internal = "80"
26     external = "8082" #changed port to 8082 from 8081
27   }
28 }
29
30
31 }
```

At the bottom of the interface, there are status indicators: Ln 30, Col 1, Spaces: 4, UTF-8, LF, Terraform, Go Live, Prettier, and a refresh icon.

## 5. Create a plan and use the plan to make changes to the resource

The screenshot shows the VS Code interface with the Terraform extension. The Explorer sidebar shows files .terraform, .terraform.lock.hcl, main.tf, terraform.tfstate, and tfplan. The main editor shows the Terraform configuration file main.tf:

```
12 # Create MySQL container
13 resource "docker_container" "mysql" {
14   name     = "mysql"
15   image    = "container-registry.oracle.com/mysql/community-server:latest"
16   restart  = "always"
17 }
18
19 # Create Nginx container
20 resource "docker_container" "nginx" {
21   name     = "nginx"
22   image    = "nginx:latest"
23   restart  = "always"
24   ports {
25     internal = "80"
26     external = "8082" #changed port to 8082 from 8081
27   }
28 }
29
30
```

The terminal below shows the output of the `terraform plan` command:

```
● aratrika@Aratrikas-MacBook-Air BC DV-4033-Lab4-Terraform % terraform plan -out=tfplan
docker_container.mysql: Refreshing state... [id=8d05d07f9fb7c5d07ac8a2ca84ef62068b2431826d0fc9777c10055d9adcfb3]
docker_container.nginx: Refreshing state... [id=60f6bc2defe55a819204c88f3723dd3b307e46954713fd870b4170f5eb324be4]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement
Terraform will perform the following actions:

# docker_container.mysql must be replaced
-/+ resource "docker_container" "mysql" {
  + bridge
  ^ command
  ~ container_id
  + container_ip
  + container_logs
  - cpu_shares
  - dns
  - dns_opts
  - dns_search
  + environment
  + entrypoint
  + env
  + exit_code
  + group_add
  ~ hostname
  ~ id
  ~ image
  + network_mode
  - privileged
  + publish_all_ports
  ~ runtime
  ~ security_opts
  ~ shm_size
  ~ stdio_signal
  ~ stop_timeout
  ~ storage_opts
  ~ sysctls
  ~ tmpfs
  # (13 unchanged attributes hidden)
  ~ ports {
    ~ external = 8081 -> 8082 # forces replacement
    # (3 unchanged attributes hidden)
  }
}
```

The terminal also shows the command to apply the plan: `terraform apply "tfplan"`.

The screenshot shows the VS Code interface with the Terraform extension. The Explorer sidebar shows files .terraform, .terraform.lock.hcl, main.tf, terraform.tfstate, and tfplan. The main editor shows the Terraform configuration file main.tf:

```
12 # Create MySQL container
13 resource "docker_container" "mysql" {
14   name     = "mysql"
15   image    = "container-registry.oracle.com/mysql/community-server:latest"
16   restart  = "always"
17 }
18
19 # Create Nginx container
20 resource "docker_container" "nginx" {
21   name     = "nginx"
22   image    = "nginx:latest"
23   restart  = "always"
24   ports {
25     internal = "80"
26     external = "8082" #changed port to 8082 from 8081
27   }
28 }
29
30
```

The terminal below shows the output of the `terraform apply "tfplan"` command:

```
● aratrika@Aratrikas-MacBook-Air BC DV-4033-Lab4-Terraform % terraform apply "tfplan"
Plan: 2 to add, 0 to change, 2 to destroy.

Saved the plan to: tfplan
To perform exactly these actions, run the following command to apply:
  terraform apply "tfplan"
aratrika@Aratrikas-MacBook-Air BC DV-4033-Lab4-Terraform %
```

The terminal shows the command to apply the plan: `terraform apply "tfplan"`.

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER**: Shows files in the current workspace: .terraform, .terraform.lock.hcl, main.tf, terraform.tfstate, terraform.tfstate.backup, and tfplan.
- main.tf**: The Terraform configuration file. It defines two Docker containers: "mysql" and "nginx". The "mysql" container uses the "mysql" image from the Oracle MySQL registry and has its port 3306 mapped to 8081. The "nginx" container uses the "nginx:latest" image and has its port 80 mapped to 8082.
- TERMINAL**: Shows the command-line output of the Terraform plan. It indicates 2 resources to add, 0 to change, and 2 to destroy. The output also lists temporary files and ports being managed.
- BROWSER**: A preview window showing the "Welcome to nginx!" page at [localhost:8082](http://localhost:8082).

```
12 # Create MySQL container
13 resource "docker_container" "mysql" {
14   name  = "mysql"
15   image = "container-registry.oracle.com/mysql/community-server:latest"
16   restart = "always"
17 }
18
19 # Create Nginx container
20 resource "docker_container" "nginx" {
21   name  = "nginx"
22   image = "nginx:latest"
23   restart = "always"
24   ports {
25     internal = "80"
26     external = "8082" #changed port to 8082 from 8081
27   }
28 }
29
30
```

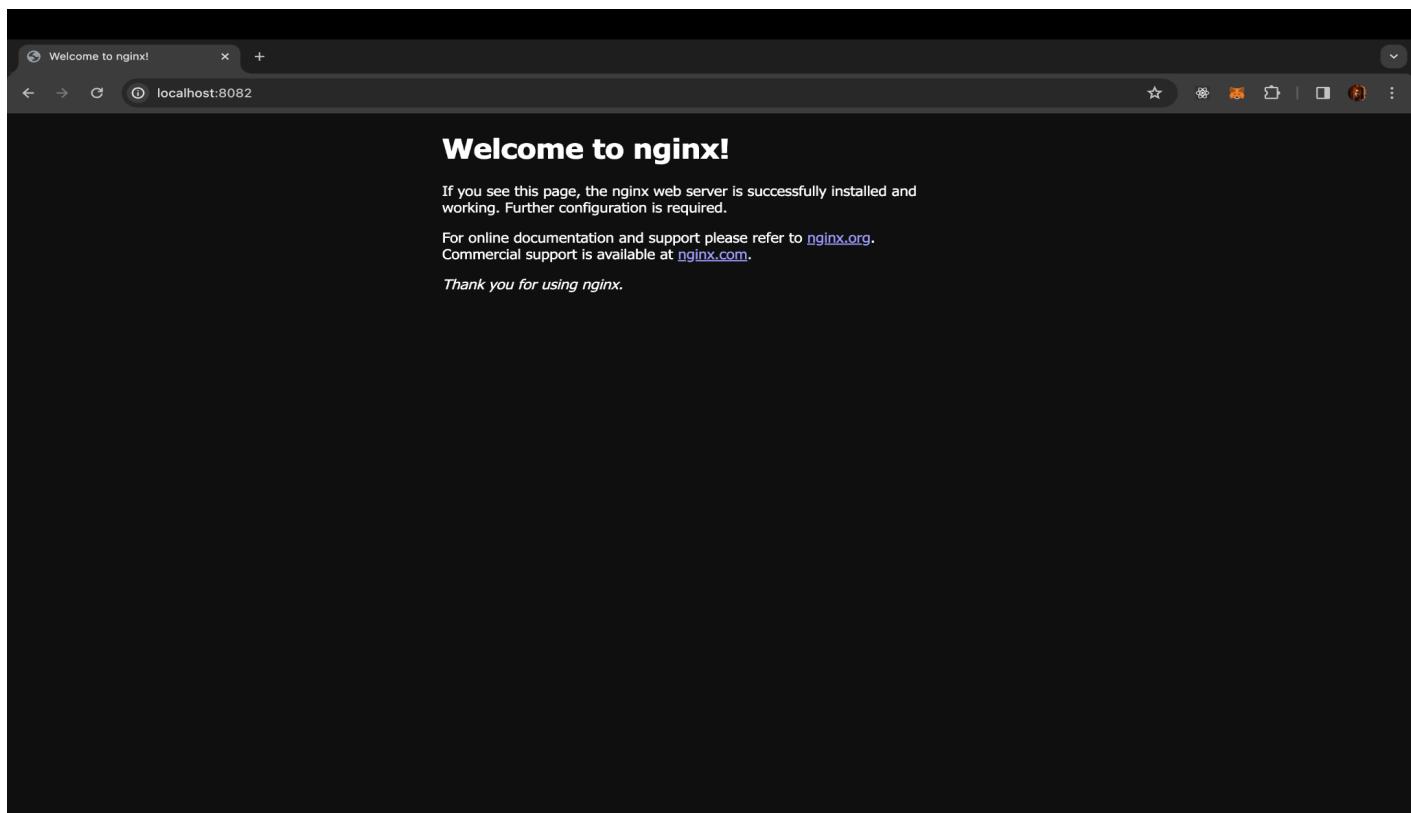
```
- tmpfs
# (13 unchanged attributes hidden)
= {} -> null
~ ports
  ~ external = 8081 -> 8082 # forces replacement
    # (3 unchanged attributes hidden)
  }
)

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

Saved the plan to: tfplan

To perform exactly these actions, run the following command to apply:
  terraform apply "tfplan"
● aratrika@Aratrikas-MacBook-Air BCDV-4033-Lab4-Terraform % terraform apply tfplan
docker_container.mysql: Destroying... [id=8d05d07f9fb7c5d97ac8a02ca84ef626bb2431826d0fc9777c1005d9adcfb3]
docker_container.nginx: Destroying... [id=60f6bc2cdefe55a19204c88f3723dd3b307e46954713fd670b4170f5eb324be4]
docker_container.nginx: Destruction complete after 0s
docker_container.nginx: Creating...
docker_container.mysql: Creation complete after 0s
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 0s [id=03fba8a5f620f74e991dfbd0fc32e99bfff5d677b3829d4db562adc182a9cae23]
docker_container.mysql: Creation complete after 0s [id=b61afb7dfc72b391d41c80a3d5ff4e8a4fa92cb97e1e55097121bf597fc5acfcd]
```

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.  
● aratrika@Aratrikas-MacBook-Air BCDV-4033-Lab4-Terraform %



## 6. Make destructive changes

Removed mysql container

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER**: Shows the project structure: BCDV-4033-LAB4-TERRAFORM, .terraform, terraform.lock.hcl, main.tf (selected), terraform.tfstate, and tfplan.
- MAIN PANE**: The code editor displays the `main.tf` file. The provider `provider "docker"` is highlighted. The code defines two Docker containers: `mysql` and `nginx`. The `mysql` container uses the `mysql:latest` image and restarts always. The `nginx` container uses the `nginx:latest` image and maps port 8082 to 8081.
- PROBLEMS**: Shows a warning: `# docker_container.mysql will be destroyed` because `resource "docker_container" "mysql"` is not in configuration.
- OUTPUT**, **DEBUG CONSOLE**, **TERMINAL**, **JUPYTER**: Tabs for running commands and viewing logs.
- STATUS BAR**: Shows the current file is `main.tf`, Col 4 (162 selected), Spaces: 4, UTF-8, LF, Terraform, Go Live, and Prettier.

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows the project structure with files: `.BCDV-4033-LAB4-TERRAFORM`, `.terraform`, `.terraform.lock.hcl`, `main.tf` (selected), `terraform.tfstate`, and `tfplan`.
- Terminal View:** Displays the Terraform configuration (`main.tf`) and its execution results:

```
main.tf > ...
8 }
9
10 provider "docker" {}
11
12 # Create MySQL container
13 # resource "docker_container" "mysql" {
14 #   name     = "mysql"
15 #   image    = "container-registry.oracle.com/mysql-community-server:latest"
16 #   restart  = "always"
17 # }
18
19 # Create Nginx container
20 resource "docker_container" "nginx" {
21   name     = "nginx"
22   image    = "nginx:latest"
23   restart  = "always"
24   ports {
25     internal = "80"
26     external = "8082" #changed port to 8082 from 8081
27   }
28 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
~ security_opts          = [] -> (known after apply)
~ shm_size                = 64 -> (known after apply)
~ stop_signal              = "SIGQUIT" -> (known after apply)
~ stop_timeout             = 0 -> (known after apply)
~ storage_opts            = {} -> null
- sysctls                 = {} -> null
- tmpfs                   = {} -> null

# (13 unchanged attributes hidden)

}
# (1 unchanged block hidden)
```

Plan: 1 to add, 0 to change, 2 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_container.mysql: Destroying... [id=b61afbf7dfc72b39d41c80a3d5f4e8a4fa92cb97e1e55097121bf597fc5acfcd]
docker_container.nginx: Destroying... [id=03fb8a5f620f74e991dfbd0fc32e99bf5fd677b3829d4db562adc182a9caa23]
docker_container.nginx: Destruction complete after 0s
docker_container.nginx: Creating...
docker_container.mysql: Destruction complete after 0s
docker_container.nginx: Creation complete after [id=012655e619ee355931984681d10a41085e9783abda1b01b336617785fb7c2fb]
```

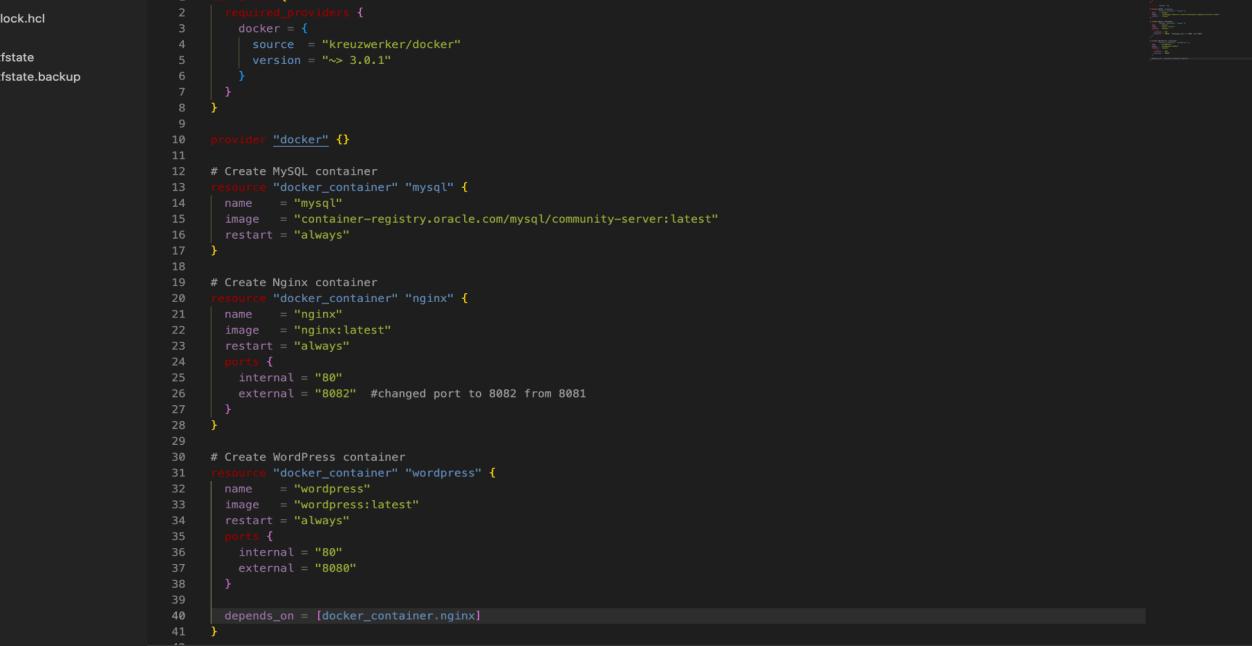
## 7. Destroy the complete resource

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER**: Shows a tree view with the current workspace: BCDV-4033-LAB4-TERRAFORM.
- main.tf**: The active code editor window contains the Terraform configuration file. The code defines two Docker containers: one for MySQL and one for Nginx, both using the latest Docker images and configured to always restart.
- TERMINAL**: The terminal tab shows the output of the `terraform destroy` command. It indicates that the `docker_container.nginx` resource will be destroyed. Terraform also lists other resources and their actions, such as `destroy` for the `docker_container.nginx` resource.
- OUTPUT**: Shows the output of the `terraform destroy` command, including the execution plan and resource actions.
- PROBLEMS**: Shows no problems or errors in the current workspace.
- JUPYTER**: Shows no Jupyter notebooks.
- STATUS BAR**: Shows the current terminal profile (zsh), window controls, and file navigation icons.

## **8. Create resource with dependencies (implicit and explicit)**

In the below example, the wordpress docker container implicitly depends on the mysql container and explicitly depends on the nginx container because of the “depends\_on” attribute.



BCDV-4033-LAB4-TERRAFORM

```
main.tf
1 terraform {
2   required_providers {
3     docker = {
4       source  = "kreuzwerker/docker"
5       version = "~> 3.0.1"
6     }
7   }
8 }
9
10 provider "docker" {}
11
12 # Create MySQL container
13 resource "docker_container" "mysql" {
14   name    = "mysql"
15   image   = "container-registry.oracle.com/mysql-community-server:latest"
16   restart = "always"
17 }
18
19 # Create Nginx container
20 resource "docker_container" "nginx" {
21   name    = "nginx"
22   image   = "nginx:latest"
23   restart = "always"
24   ports {
25     internal = "80"
26     external = "8082" #changed port to 8082 from 8081
27   }
28 }
29
30 # Create WordPress container
31 resource "docker_container" "wordpress" {
32   name    = "wordpress"
33   image   = "wordpress:latest"
34   restart = "always"
35   ports {
36     internal = "80"
37     external = "8080"
38   }
39
40   depends_on = [docker_container.nginx]
41 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

docker\_container.wordpress: Creation complete after 0s [id=698ce04988052931225c9408e37bce5603eab0c42cc5aeaeda1e2397f994b33]

Apply complete! Resources: 3 added, 0 changed, 3 destroyed.

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** sidebar: Shows files in the current workspace, including `.terraform`, `.terraform.lock.hcl`, `main.tf`, `terraform.tfstate`, `terraform.tfstate.backup`, and `tfplan`.
- main.tf** editor tab: Displays Terraform configuration code for a WordPress Docker container. The code defines a `resource "docker_container" "wordpress"` block with attributes like `restart = "always"`, `ports { internal = "80"; external = "8082" }`, and `depends_on = [docker_container.mysql, docker_container.nginx]`.
- TERMINAL**: Shows the command `terraform apply` being run, and the resulting execution plan which includes creating a MySQL Docker container.
- OUTPUT**: Shows the results of the `terraform apply` command, indicating the creation of the MySQL Docker container.
- PROBLEMS**: Shows no errors or warnings.
- STATUS BAR**: Shows the terminal output, including the command `zsh +` and file paths.

EXPLORER    main.tf

```

BCDV-4033-LAB4-TERRAFORM
> .terraform
≡ .terraform.lock.hcl
main.tf
() terraform.tfstate
≡ terraform.state.backup
≡ tfplan

main.tf
  resource "docker_container" "wordpress"
    restart = "always"
    ports {
      internal = "80"
      external = "8082" #changed port to 8082 from 8081
    }
    # Create WordPress container
    resource "docker_container" "wordpress" {
      name   = "wordpress"
      image  = "wordpress:latest"
      restart = "always"
      ports {
        internal = "80"
        external = "8080"
      }
      depends_on = [docker_container.mysql, docker_container.nginx]
    }
  
```

TERMINAL

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER
+ tty
+ wait
+ wait_timeout
= false
= false
= 60

+ ports {
  + external = 8080
  + internal = 80
  + ip       = "0.0.0.0"
  + protocol = "tcp"
}
}

Plan: 3 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_container.mysql: Creating...
docker_container.nginx: Creating...
docker_container.mysql: Creation complete after 0s [id=c53a890f4ac57c7613196a32915a98944abe89e5a82e28fffc9357fabf6d8799c]
docker_container.nginx: Creation complete after 0s [id=0749eb51b3ee1bd62b471933c41f51ed80e3f3f529025bb388b6d435dac4]
docker_container.wordpress: Creating...
docker_container.wordpress: Creation complete after 10s [id=8dc08305f5b55917880e69e0f5ddfe450e4de910e83befef9dba7f0ef8190054]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
○ aratrika@Aratrikas-MacBook-Air BCDV-4033-Lab4-Terraform %
```

Ln 41, Col 2    Spaces: 4    UTF-8    LF    () Terraform    ⚡ Go Live    ⚡ Prettier

