LAB 12

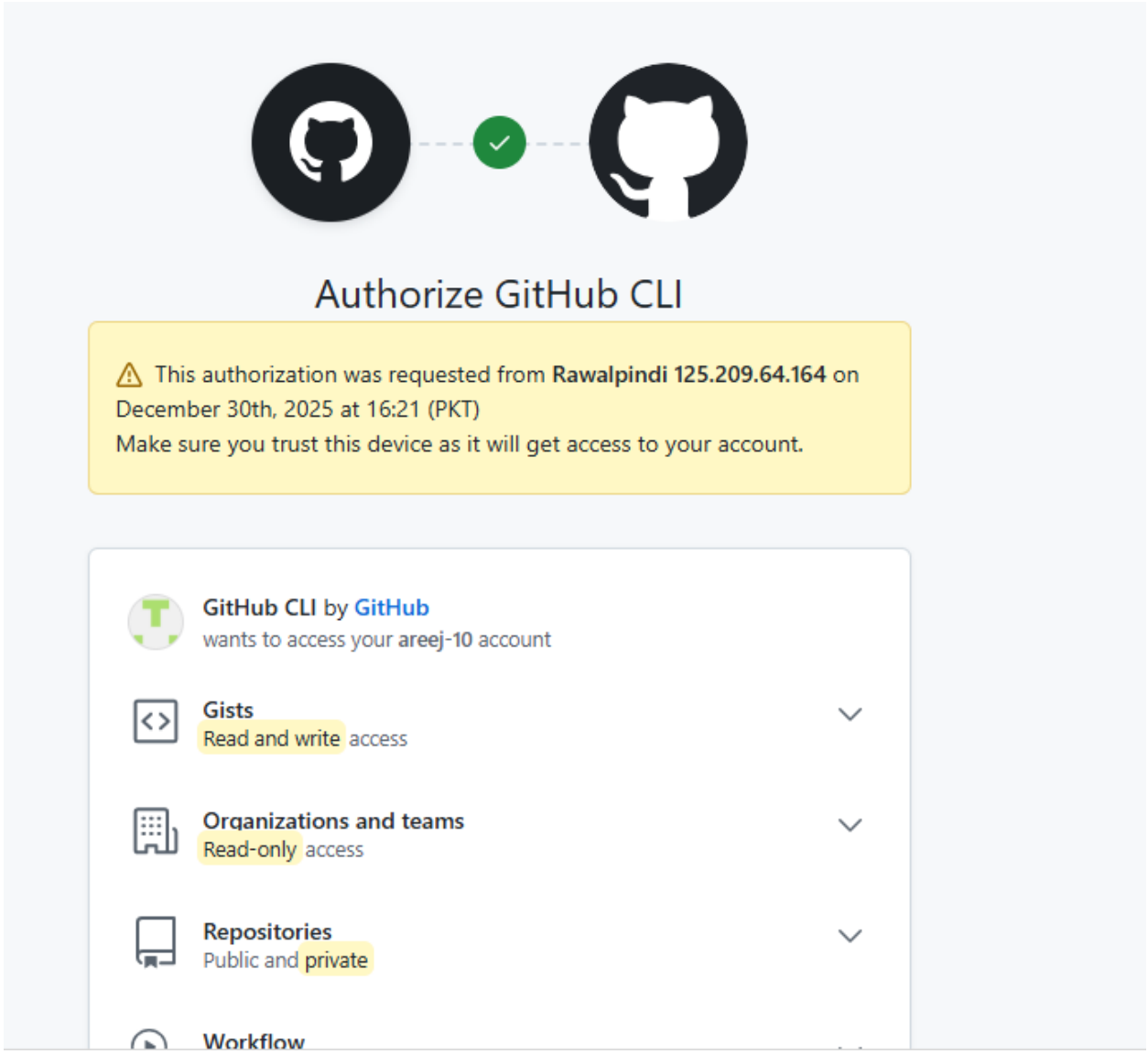
CLOUD COMPUTING

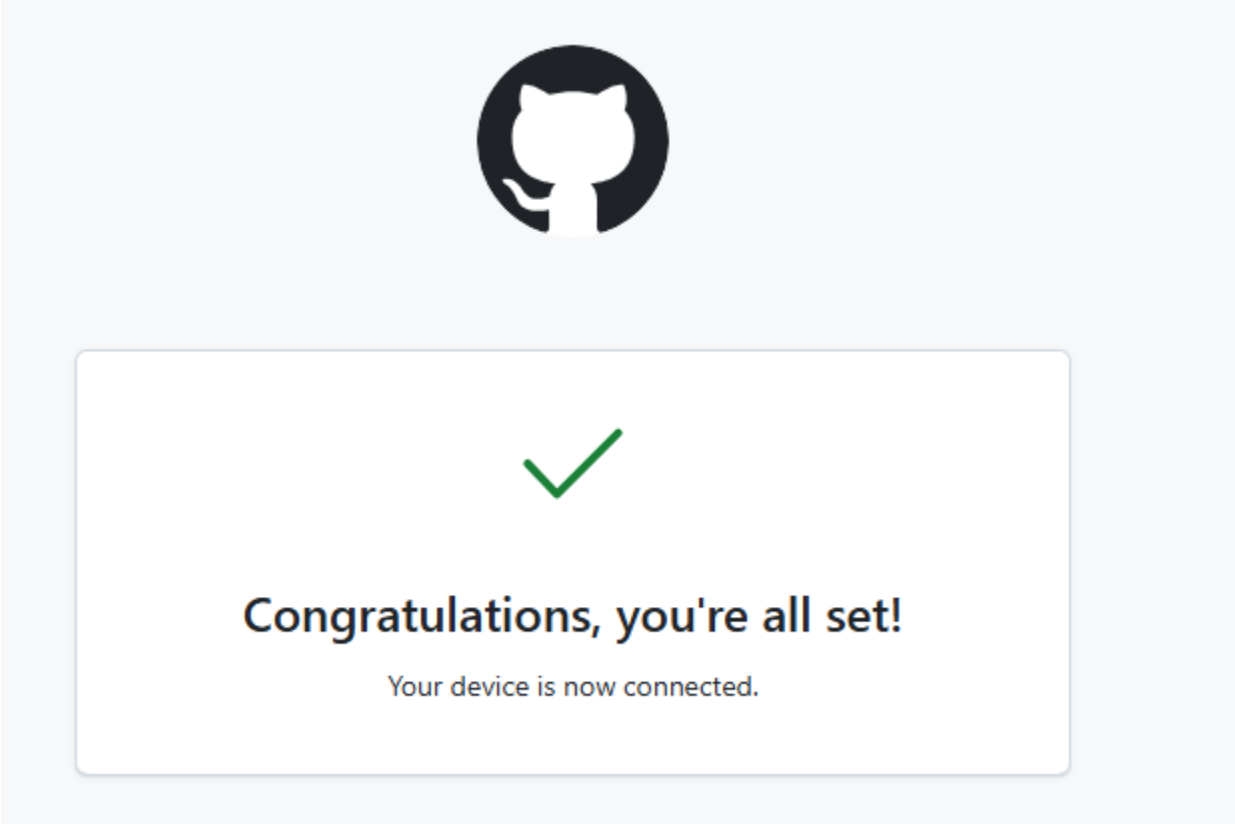
SUBMITTED BY:

AREEJ FATIMA

BSE-2023-010

SEMESTER VA


Authorize GitHub CLI
This authorization was requested from Rawalpindi 125.209.64.164 on December 30th, 2025 at 16:21 (PKT)
Make sure you trust this device as it will get access to your account.
GitHub CLI by GitHub
wants to access your areej-10 account
Gists
Read and write access
Organizations and teams
Read-only access
Repositories
Public and private
Workflow




```
Dell@DESKTOP-JNCVEJH MINGW64 ~ (main)
$ gh repo create areej-10/CC_areej_010_Lab12 --public
✓ Created repository areej-10/CC_areej_010_Lab12 on github.com
  https://github.com/areej-10/CC_areej_010_Lab12

Dell@DESKTOP-JNCVEJH MINGW64 ~ (main)
$ gh repo create areej-10/CC_areej_010_Lab12 --public
GraphQL: Name already exists on this account (createRepository)

Dell@DESKTOP-JNCVEJH MINGW64 ~ (main)
$
```

task0_codespace_create_and_list.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~ (main)
$ gh codespace list
NAME                DISPLAY NAME    REPOSITORY       BRANCH  STATE     CREATED AT
urban-carnival-...  urban carnival  areej-10/are...  main    Shutdown  about 19 hou...
```

task0_codespace_ssh_connected.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~ (main)
$  gh codespace ssh -c urban-carnival-r45r7xvvwwxvh565g
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

@areej-10 → /workspaces/areej-10-lab6 (main) $
```

task1_project_directory.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~ (main)
$ # Go to your desired location (e.g., Documents)
   cd ~/Documents

   # Create project folder
   mkdir -p CC_areej_10/Assignment2
   cd CC_areej_10/Assignment2

Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$
```

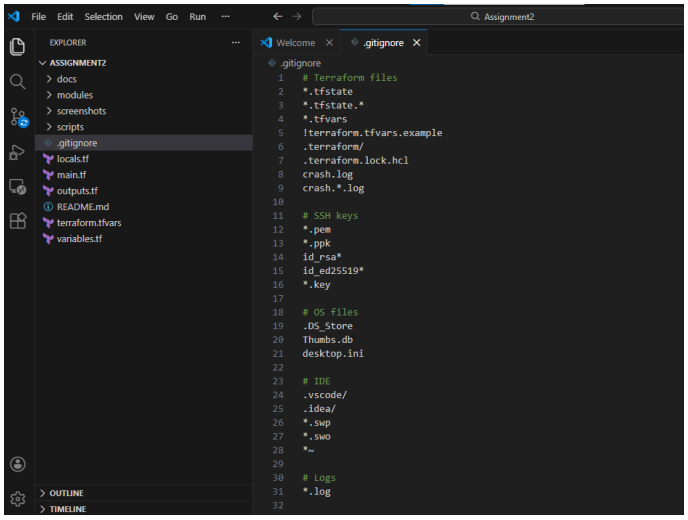
task1_ssh_keygen.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ # Generate ED25519 key (more secure than RSA)
   ssh-keygen -t ed25519 -C "areejfatimaa101@gmail.com" -f ~/.ssh/id_ed25519
Generating public/private ed25519 key pair.
/c/Users/Dell/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase for "/c/Users/Dell/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Dell/.ssh/id_ed25519
Your public key has been saved in /c/Users/Dell/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:8nRrIL1lJnuuPOBPMpFQVnL2t2kn7qKd7M4tw9eWC5s areejfatimaa101@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|      .o+o+      |
|       .o+ .     |
|         .  . .  |
|       .+ .  . o |
|      o.S==   = .|
|       =.X.. o o |
|        +.+. ....|
|         +.+=+=.o|
|        .o+=BEoo.|
+----[SHA256]-----+
```

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ ls -la ~/.ssh/
total 26
drwxr-xr-x 1 Dell 197121    0 Oct 24 12:38 ./
drwxr-xr-x 1 Dell 197121    0 Dec 28 16:09 ../
-rw-r--r-- 1 Dell 197121  419 Dec 28 16:13 id_ed25519
-rw-r--r-- 1 Dell 197121  107 Dec 28 16:13 id_ed25519.pub
-rw-r--r-- 1 Dell 197121 1674 Oct 24 12:38 known_hosts
-rw-r--r-- 1 Dell 197121  925 Oct 24 12:38 known_hosts.old

Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILcxgqdS2YYYrtrFoZAnsYRnRk9xJODcgNim9hvWmJj7 areejfatimaa101@gmail
.com
```

task1_files_created.png

```
$ ls -R
.:
README.md  docs/  locals.tf  main.tf  modules/  outputs.tf  screenshots/  scripts/  terraform.tfvars  variables.tf

./docs:
architecture.md  troubleshooting.md

./modules:
networking/  security/  webserver/

./modules/networking:
main.tf  outputs.tf  variables.tf

./modules/security:
main.tf  outputs.tf  variables.tf

./modules/webserver:
main.tf  outputs.tf  variables.tf

./screenshots:
bonus/  part1/  part2/  part3/  part4/  part5/  part6/

./screenshots/bonus:

./screenshots/part1:

./screenshots/part2:

./screenshots/part3:

./screenshots/part4:

./screenshots/part5:

./screenshots/part6:

./scripts:
apache-setup.sh  nginx-setup.sh
```



task1_variables_tf.png

variables.tf

```
# VPC Configuration
variable "vpc_cidr_block" {
  description = "CIDR block for VPC"
  type        = string
  default     = "10.0.0.0/16"

  validation {
    condition     = can(cidrhost(var.vpc_cidr_block, 0))
    error_message = "Must be a valid IPv4 CIDR block."
  }
}

variable "subnet_cidr_block" {
  description = "CIDR block for subnet"
  type        = string
  default     = "10.0.10.0/24"

  validation {
    condition     = can(cidrhost(var.subnet_cidr_block, 0))
    error_message = "Must be a valid IPv4 CIDR block."
  }
}

variable "availability_zone" {
  description = "AWS availability zone"
  type        = string
  default     = "me-central-1a"
}

# Environment Configuration
variable "env_prefix" {
  description = "Environment prefix for resource naming"
```

The git repository a
changes, only a sul
Source: Git

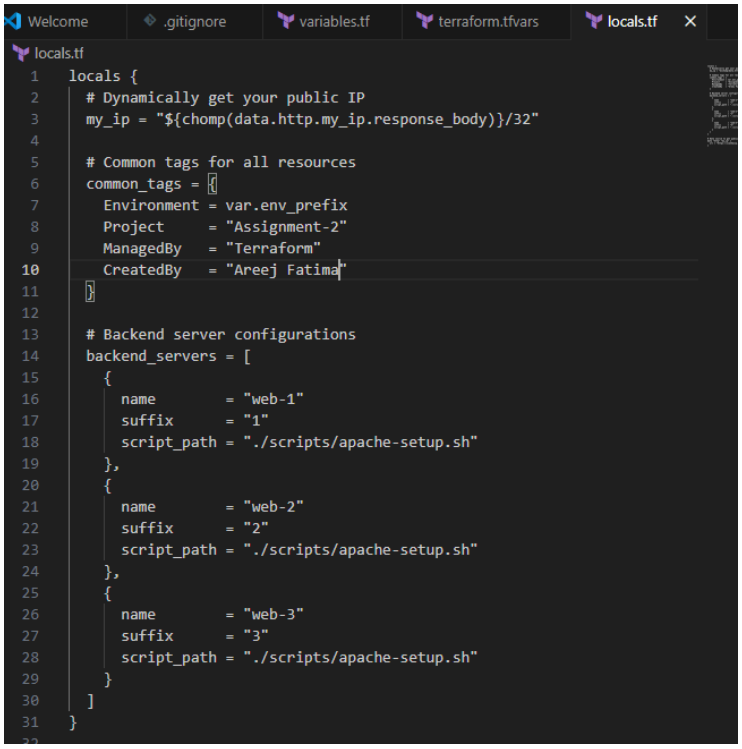
task1_terraform_tfvars.png

Welcome | .gitignore | variables.tf | terraform.tfvars

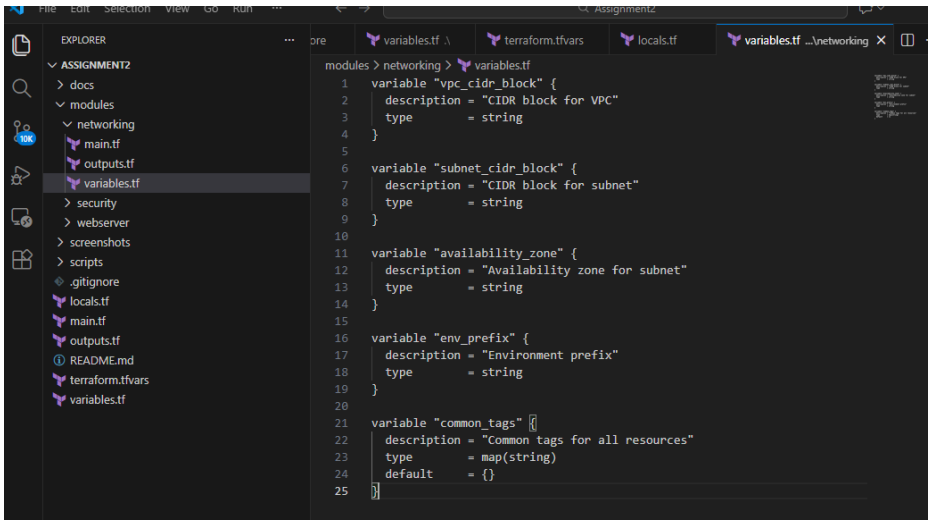
terraform.tfvars

```
vpc_cidr_block     = "10.0.0.0/16"
subnet_cidr_block  = "10.0.10.0/24"
availability_zone  = "me-central-1a"
env_prefix         = "prod"
instance_type      = "t3.micro"
public_key         = "~/.ssh/id_ed25519.pub"
private_key        = "~/.ssh/id_ed25519"
```

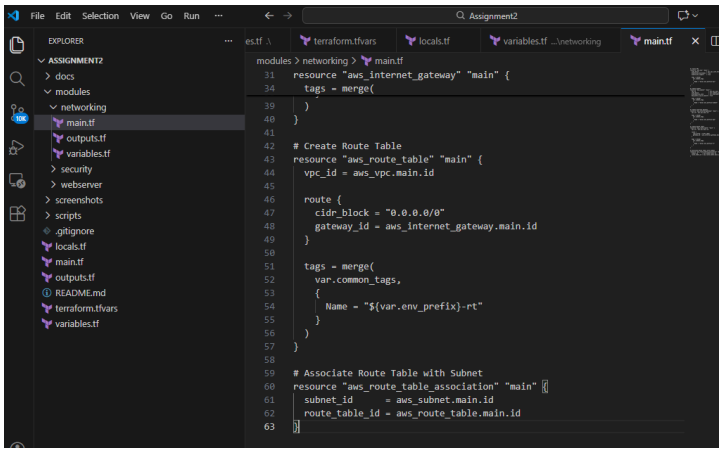
task1_locals_tf.png



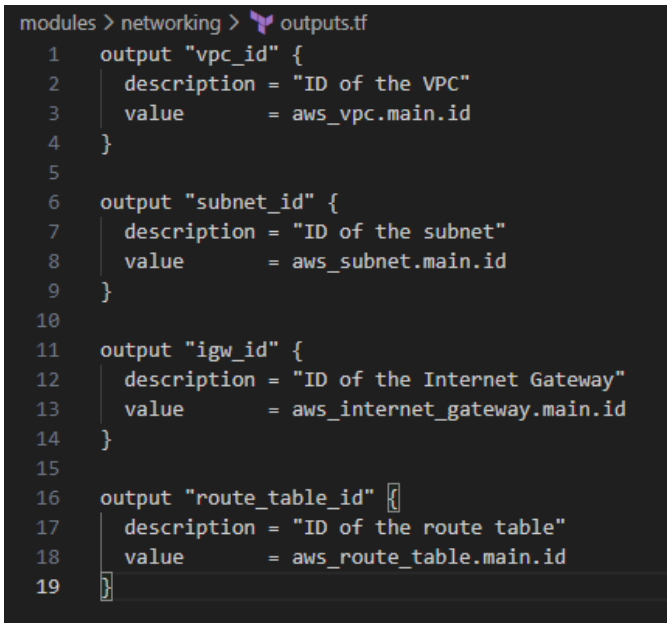
task1_variables_tf.png



task1_main_tf.png



task1_outputs_tf.png



task1_entry_script.png

scripts > $ apache-setup.sh

```
cat > /var/www/html/index.html <<EOF
<!DOCTYPE html>
<html>
<head>
    <title>Backend Web Server - $HOSTNAME</title>
    <style>
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif
            margin: 0;
            padding: 0;
            background: linear-gradient(135deg, #667eea 0%, #764ba2 100%
            min-height: 100vh;
            display: flex;
            justify-content: center;
            align-items: center;
        }
        .container {
            background: rgba(255, 255, 255, 0.95);
            padding: 40px;
            border-radius: 20px;
            box-shadow: 0 20px 60px rgba(0, 0, 0, 0.3);
            max-width: 800px;
            width: 90%;
        }
        h1 {
            color: #667eea;
            text-align: center;
            margin-bottom: 30px;
            font-size: 2.5em;
            text-shadow: 2px 2px 4px rgba(0,0,0,0.1);
        }
        .info-grid {
```

task1_terraform_init.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ # Initialize Terraform
terraform init
Initializing the backend...
Initializing modules...
- backend_servers in modules\webserver
- networking in modules\networking
- nginx_server in modules\webserver
- security in modules\security
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 5.0"...
- Finding latest version of hashicorp/http...
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
- Installing hashicorp/aws v5.100.0...
- Installed hashicorp/aws v5.100.0 (signed by HashiCorp)
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
commands will detect it and remind you to do so if necessary.
```

task1_terraform_apply.png

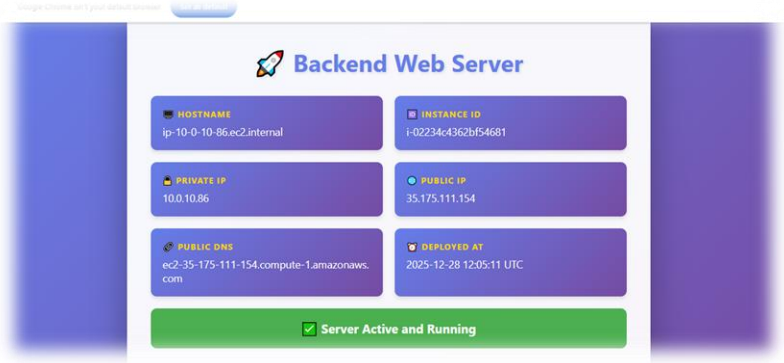
MINGW64:/c/Users/Dell/Documents/CC_areej_10/Assignment2

```
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
   - web-2: 35.175.111.154 (private: 10.0.10.86)
   - web-3: 44.204.96.43 (private: 10.0.10.180)

========================================

EOT
nginx_instance_id = "i-0e668eb55f575d3c4"
nginx_public_ip = "3.230.127.48"
subnet_id = "subnet-0d8278da93817bc6b"
vpc_id = "vpc-0df4a962ae2d7806a"
```

task1_terraform_output.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ terraform output
backend_servers_info = {
  "web-1" = {
    "instance_id" = "i-049a54d483fd72337"
    "private_ip" = "10.0.10.67"
    "public_ip" = "98.92.216.122"
  }
  "web-2" = {
    "instance_id" = "i-02234c4362bf54681"
    "private_ip" = "10.0.10.86"
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
   - web-2: 35.175.111.154 (private: 10.0.10.86)
```

task1_nginx_browser.png



task1_terraform_destroy.png

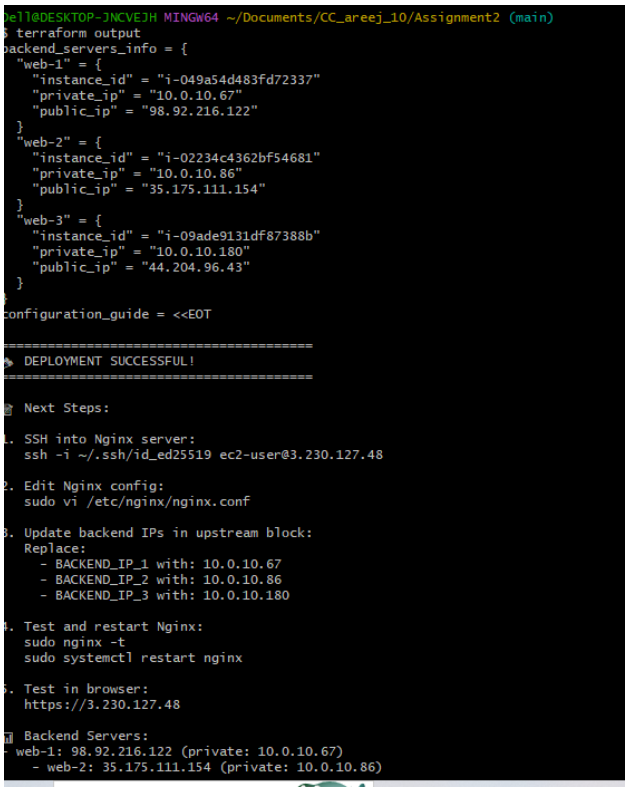
```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ terraform destroy
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
module.nginx_server.data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-1"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-2"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-3"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-2"].aws_key_pair.server_key: Refreshing state... [id=p
module.networking.aws_vpc.main: Refreshing state... [id=vpc-0df4a962ae2d7806a]
```

task2_terraform_apply.png

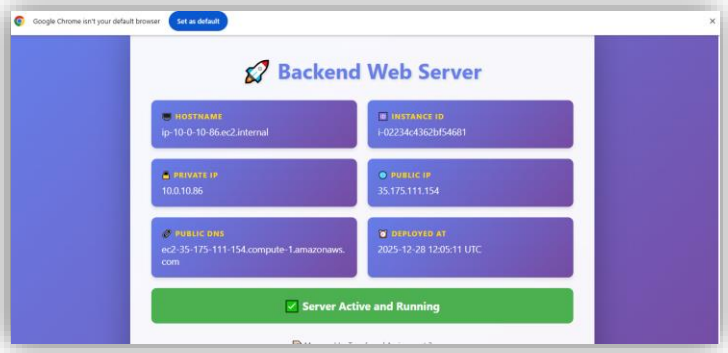
MINGW64:/c/Users/Dell/Documents/CC_areej_10/Assignment2

```
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
 web-1: 98.92.216.122 (private: 10.0.10.67)
    - web-2: 35.175.111.154 (private: 10.0.10.86)
    - web-3: 44.204.96.43 (private: 10.0.10.180)

========================================

EOT
nginx_instance_id = "i-0e668eb55f575d3c4"
nginx_public_ip = "3.230.127.48"
subnet_id = "subnet-0d8278da93817bc6b"
vpc_id = "vpc-0df4a962ae2d7806a"
```

task2_terraform_output.png



task2_nginx_browser.png



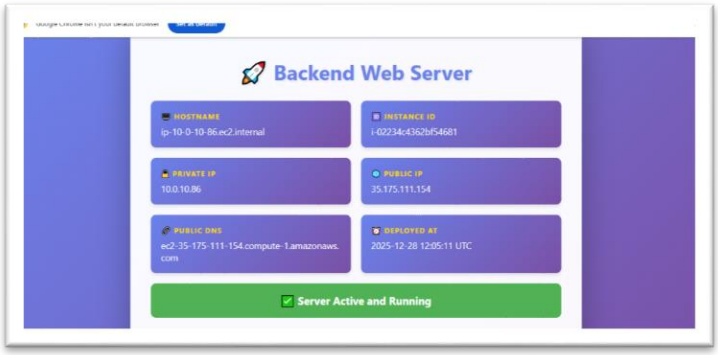
Task3_terraform_apply.png

```
MINGW64:/c/Users/Dell/Documents/CC_areej_10/Assignment2
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
   - web-2: 35.175.111.154 (private: 10.0.10.86)
   - web-3: 44.204.96.43 (private: 10.0.10.180)

========================================

EOT
nginx_instance_id = "i-0e668eb55f575d3c4"
nginx_public_ip = "3.230.127.48"
subnet_id = "subnet-0d8278da93817bc6b"
vpc_id = "vpc-0df4a962ae2d7806a"
```

Task3_terraform_output.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ terraform output
backend_servers_info = {
  "web-1" = {
    "instance_id" = "i-049a54d483fd72337"
    "private_ip" = "10.0.10.67"
    "public_ip" = "98.92.216.122"
  }
  "web-2" = {
    "instance_id" = "i-02234c4362bf54681"
    "private_ip" = "10.0.10.86"
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
   - web-2: 35.175.111.154 (private: 10.0.10.86)
```

Task3_nginx_browser.png



Task3_terraform_destroy.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ terraform destroy
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
module.nginx_server.data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-1"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-2"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-3"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-2"].aws_key_pair.server_key: Refreshing state... [id=p
module.networking.aws_vpc.main: Refreshing state... [id=vpc-0df4a962ae2d7806a]
```

task3_main_tf_restored.png

```
main.tf
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
  required_version = ">= 1.0"
}

provider "aws" {
  region = "me-central-1"
}

# Networking Module
module "networking" {
  source = "./modules/networking"

  vpc_cidr_block    = var.vpc_cidr_block
  subnet_cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  env_prefix        = var.env_prefix
  common_tags       = local.common_tags
}

# Security Module
module "security" {
  source = "./modules/security"

  vpc_id      = module.networking.vpc_id
  env_prefix  = var.env_prefix
  my_ip       = local.my_ip
```

TASK 4 - Subnet Module

task4_module_structure.png

```
$ ls -R
.:
README.md  docs/  locals.tf  main.tf  modules/  outputs.tf  screenshots/  scripts/  terraform.tfvars  variables.tf

./docs:
architecture.md  troubleshooting.md

./modules:
networking/  security/  webserver/

./modules/networking:
main.tf  outputs.tf  variables.tf

./modules/security:
main.tf  outputs.tf  variables.tf

./modules/webserver:
main.tf  outputs.tf  variables.tf

./screenshots:
bonus/  part1/  part2/  part3/  part4/  part5/  part6/

./screenshots/bonus:

./screenshots/part1:

./screenshots/part2:

./screenshots/part3:

./screenshots/part4:

./screenshots/part5:

./screenshots/part6:

./scripts:
apache-setup.sh  nginx-setup.sh
```

task4_subnet_variables.png

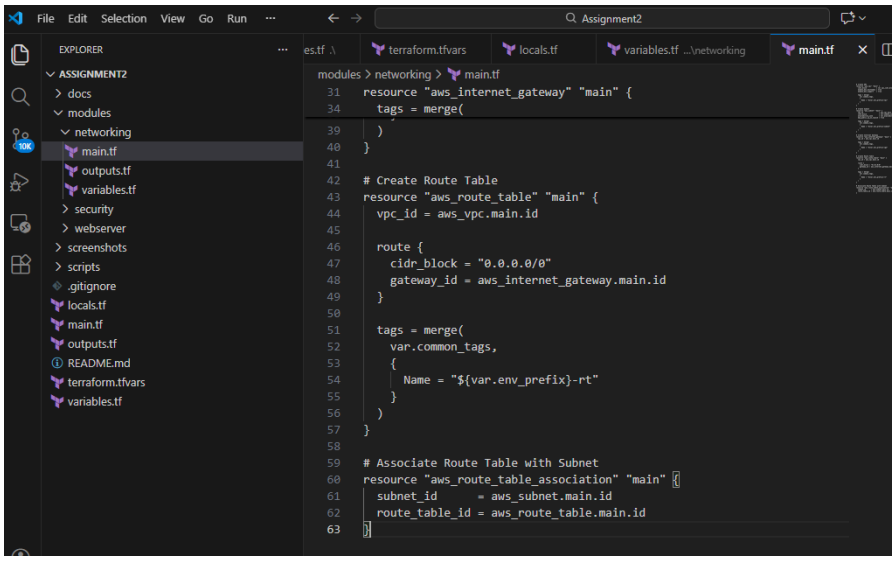
variables.tf

```
# VPC Configuration
variable "vpc_cidr_block" {
  description = "CIDR block for VPC"
  type        = string
  default     = "10.0.0.0/16"

  validation {
    condition     = can(cidrhost(var.vpc_cidr_block, 0))
    error_message = "Must be a valid IPv4 CIDR block."
  }
}

variable "subnet_cidr_block" {
  description = "CIDR block for subnet"
  type        = string
  default     = "10.0.10.0/24"

  validation {
    condition     = can(cidrhost(var.subnet_cidr_block, 0))
    error_message = "Must be a valid IPv4 CIDR block."
  }
}

variable "availability_zone" {
  description = "AWS availability zone"
  type        = string
  default     = "me-central-1a"
}

# Environment Configuration
variable "env_prefix" {
  description = "Environment prefix for resource naming"
```

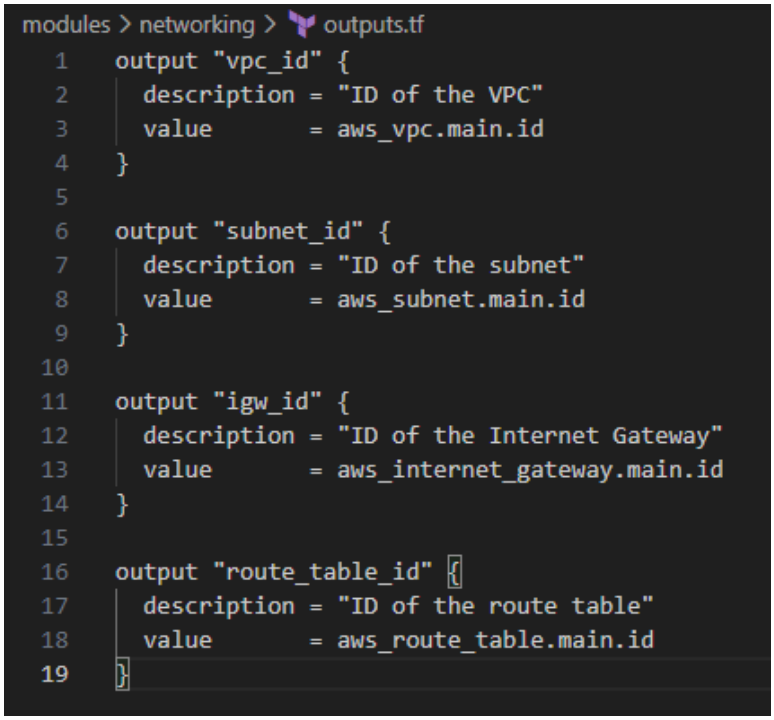
The git repository a
changes, only a sub

Source: Git

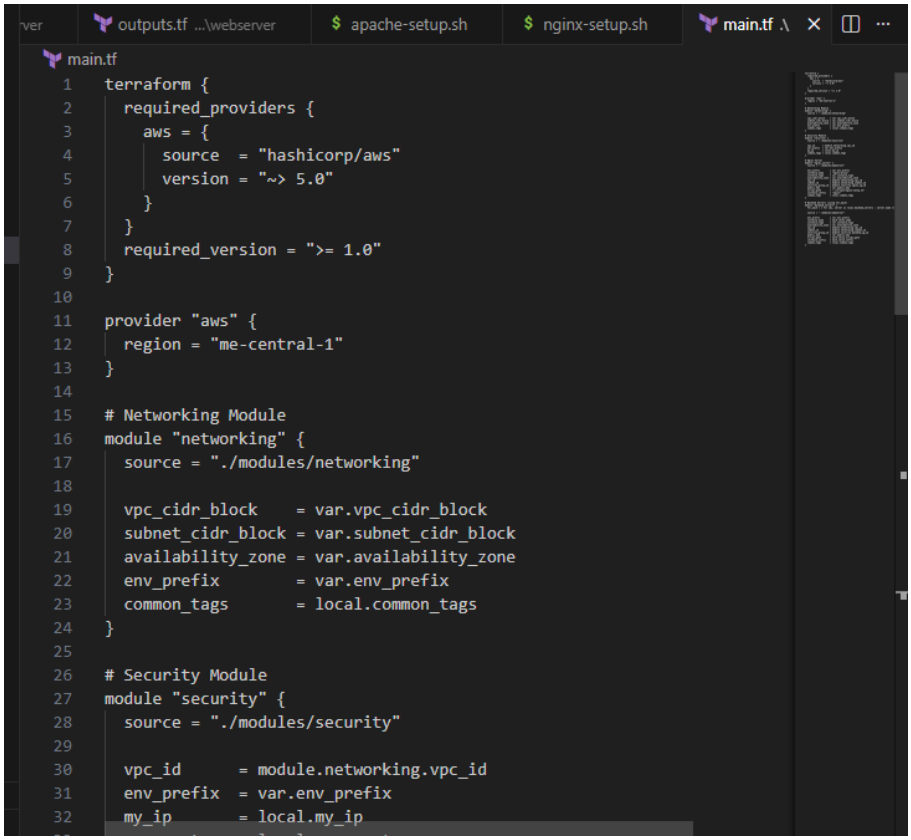
task4_subnet_main.png



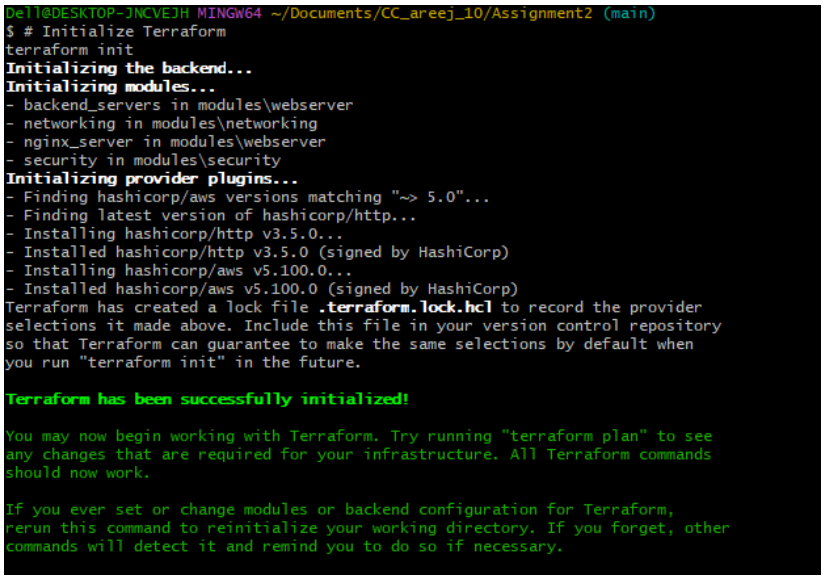
task4_subnet_outputs.png



task4_main_tf_with_module.png



Task4_terraform_init.png



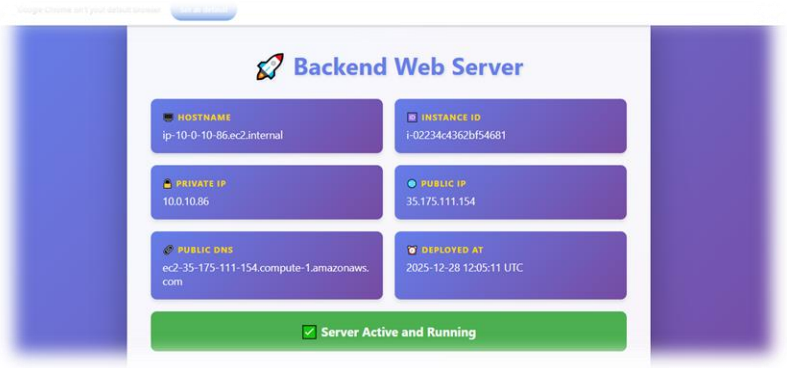
Task4_terraform_apply.png

```
MINGW64:/c/Users/Dell/Documents/CC_areej_10/Assignment2
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
   - web-2: 35.175.111.154 (private: 10.0.10.86)
   - web-3: 44.204.96.43 (private: 10.0.10.180)

========================================

EOT
nginx_instance_id = "i-0e668eb55f575d3c4"
nginx_public_ip = "3.230.127.48"
subnet_id = "subnet-0d8278da93817bc6b"
vpc_id = "vpc-0df4a962ae2d7806a"
```

Task4_terraform_output.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ terraform output
backend_servers_info = {
  "web-1" = {
    "instance_id" = "i-049a54d483fd72337"
    "private_ip" = "10.0.10.67"
    "public_ip" = "98.92.216.122"
  }
  "web-2" = {
    "instance_id" = "i-02234c4362bf54681"
    "private_ip" = "10.0.10.86"
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
   - web-2: 35.175.111.154 (private: 10.0.10.86)
```

Task4_nginx_browser.png



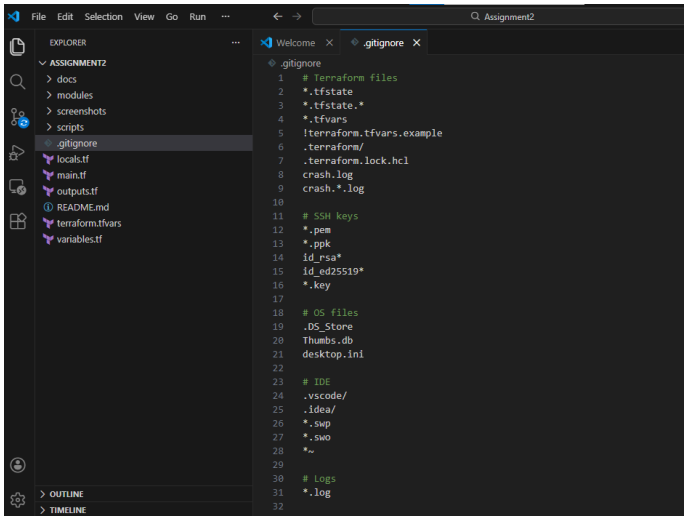
Task4_terraform_destroy.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ terraform destroy
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
module.nginx_server.data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-1"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-2"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-3"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-2"].aws_key_pair.server_key: Refreshing state... [id=p
module.networking.aws_vpc.main: Refreshing state... [id=vpc-0df4a962ae2d7806a]
```

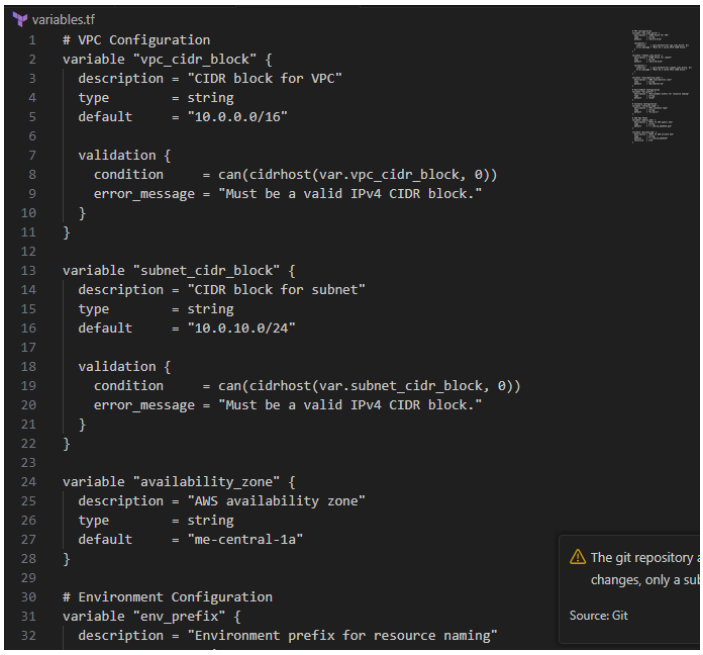
TASK 5 - Webserver Module

task5_webserver_module_structure.png

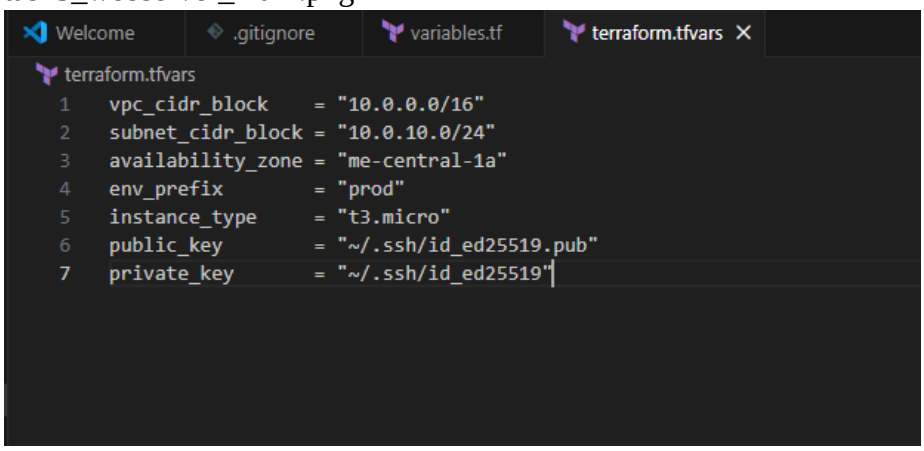
```
Dell@DESKTOP-JNCVEJH MINGW64 ~ (main)
$ # Go to your desired location (e.g., Documents)
  cd ~/Documents

  # Create project folder
  mkdir -p CC_areej_10/Assignment2
  cd CC_areej_10/Assignment2

Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$
```



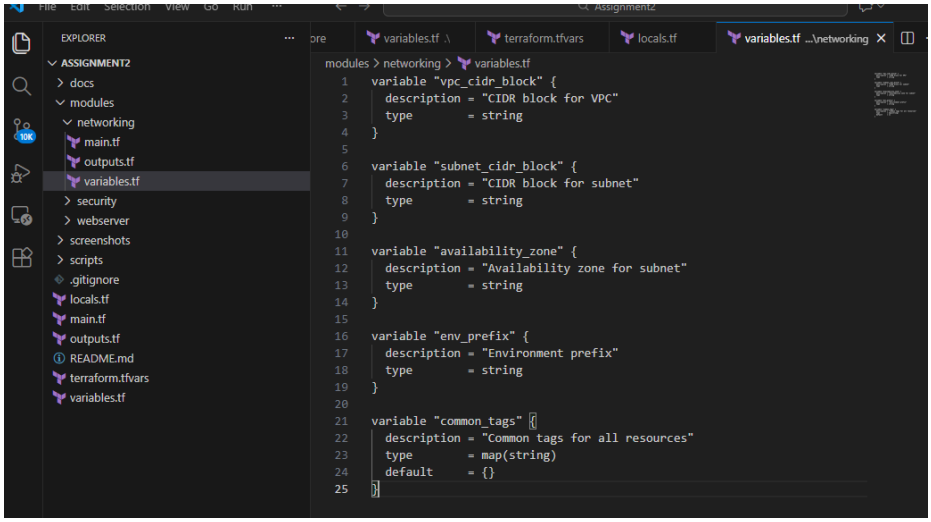
task5_webserver_variables.png



task5_webserver_main.png



task1_variables_tf.png

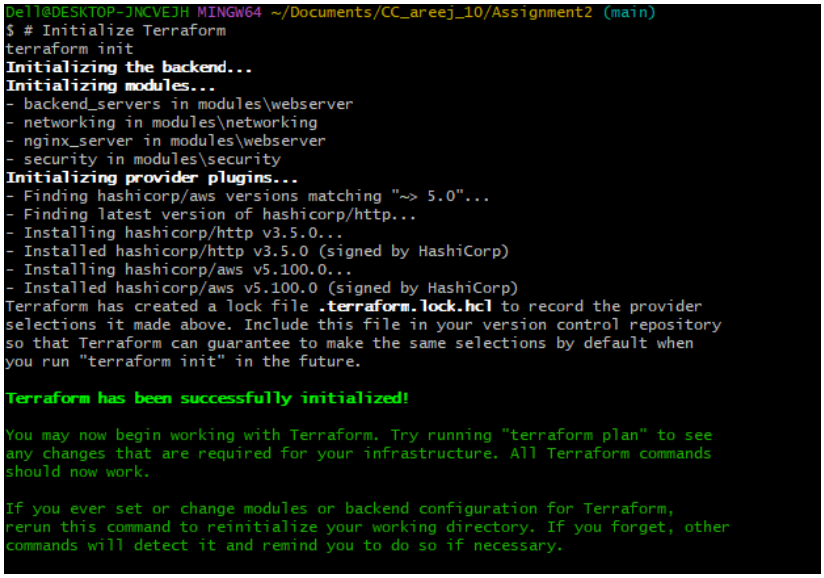


task5_outputs_updated.png

modules > networking > outputs.tf

```hcl
output "vpc_id" {
  description = "ID of the VPC"
  value       = aws_vpc.main.id
}

output "subnet_id" {
  description = "ID of the subnet"
  value       = aws_subnet.main.id
}

output "igw_id" {
  description = "ID of the Internet Gateway"
  value       = aws_internet_gateway.main.id
}

output "route_table_id" {
  description = "ID of the route table"
  value       = aws_route_table.main.id
}
```

assignment_part4_terraform_init.png



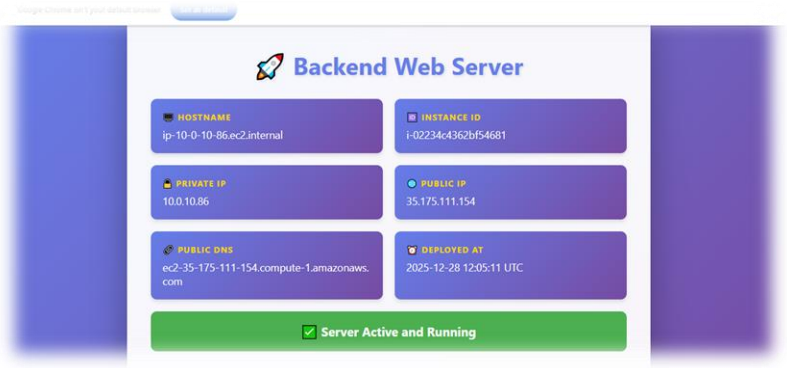
Task5_terraform_apply.png

```
MINGW64:/c/Users/Dell/Documents/CC_areej_10/Assignment2
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
   - web-2: 35.175.111.154 (private: 10.0.10.86)
   - web-3: 44.204.96.43 (private: 10.0.10.180)

========================================

EOT
nginx_instance_id = "i-0e668eb55f575d3c4"
nginx_public_ip = "3.230.127.48"
subnet_id = "subnet-0d8278da93817bc6b"
vpc_id = "vpc-0df4a962ae2d7806a"
```

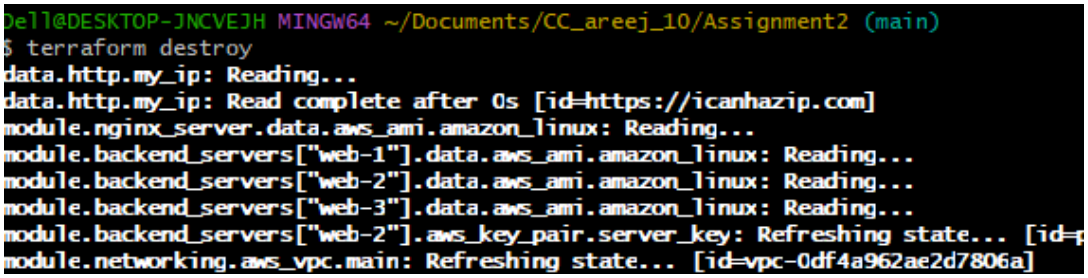
Task5_terraform_output.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ terraform output
backend_servers_info = {
  "web-1" = {
    "instance_id" = "i-049a54d483fd72337"
    "private_ip" = "10.0.10.67"
    "public_ip" = "98.92.216.122"
  }
  "web-2" = {
    "instance_id" = "i-02234c4362bf54681"
    "private_ip" = "10.0.10.86"
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
   - web-2: 35.175.111.154 (private: 10.0.10.86)
```

Task5_nginx_browser.png



Task5_terraform_destroy.png

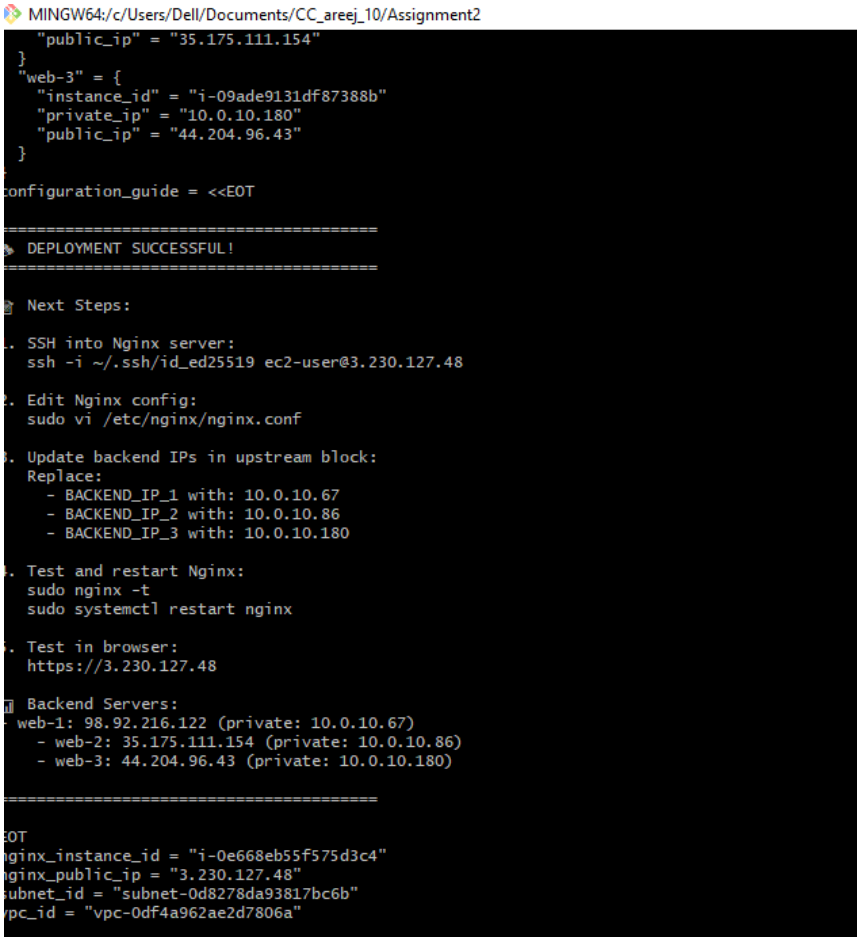


task6_entry_script_https.png

scripts > $ nginx-setup.sh

```
#!/bin/bash
set -e

# Update and install Nginx
yum update -y
yum install -y nginx openssl
systemctl start nginx
systemctl enable nginx

# Create SSL directories
mkdir -p /etc/ssl/private
mkdir -p /etc/ssl/certs

# Get metadata token
TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" \
  -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

# Get public IP
PUBLIC_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
  http://169.254.169.254/latest/meta-data/public-ipv4)

# Generate self-signed certificate
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/ssl/private/selfsigned.key \
  -out /etc/ssl/certs/selfsigned.crt \
  -subj "/C=US/ST=State/L=City/O=Organization/CN=$PUBLIC_IP" \
  -addext "subjectAltName=IP:$PUBLIC_IP" \
  -addext "basicConstraints=CA:FALSE" \
  -addext "keyUsage=digitalSignature,keyEncipherment" \
  -addext "extendedKeyUsage=serverAuth"

echo "Self-signed certificate created for IP: $PUBLIC_IP"
```

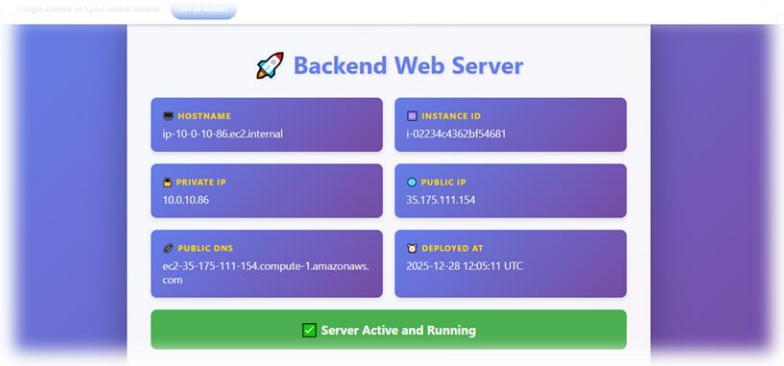
task6_terraform_apply.png



task6_terraform_output.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ terraform output
backend_servers_info = {
  "web-1" = {
    "instance_id" = "i-049a54d483fd72337"
    "private_ip" = "10.0.10.67"
    "public_ip" = "98.92.216.122"
  }
  "web-2" = {
    "instance_id" = "i-02234c4362bf54681"
    "private_ip" = "10.0.10.86"
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
🎉 DEPLOYMENT SUCCESSFUL!
========================================

📋 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

📊 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
  - web-2: 35.175.111.154 (private: 10.0.10.86)
```

task6_nginx_https_browser.png

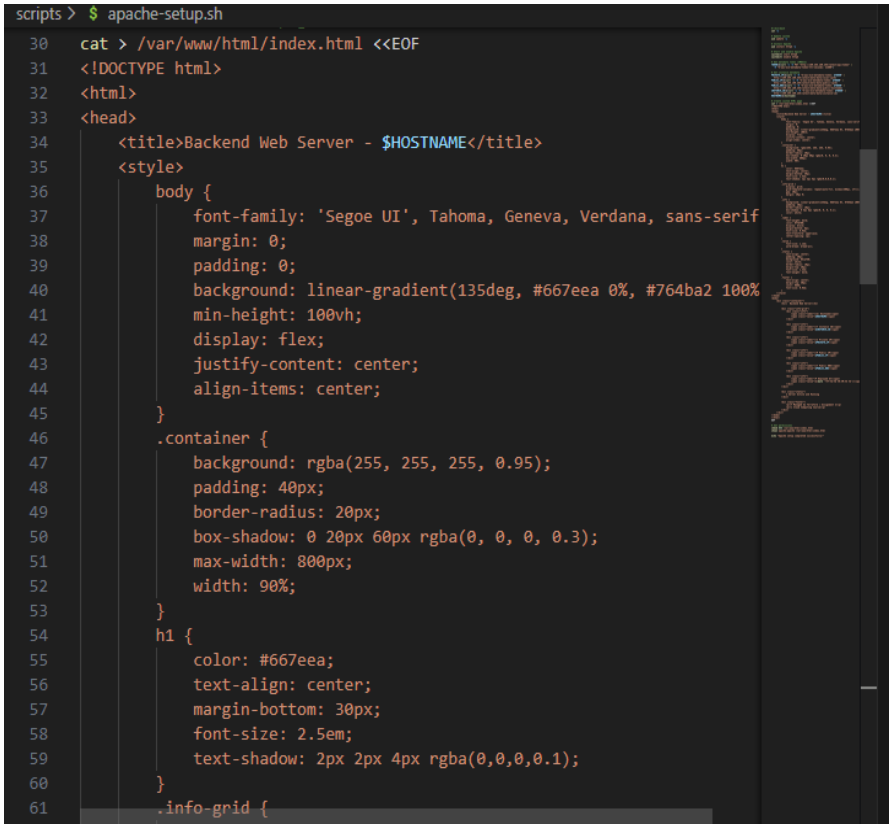


task6_http_redirect.png

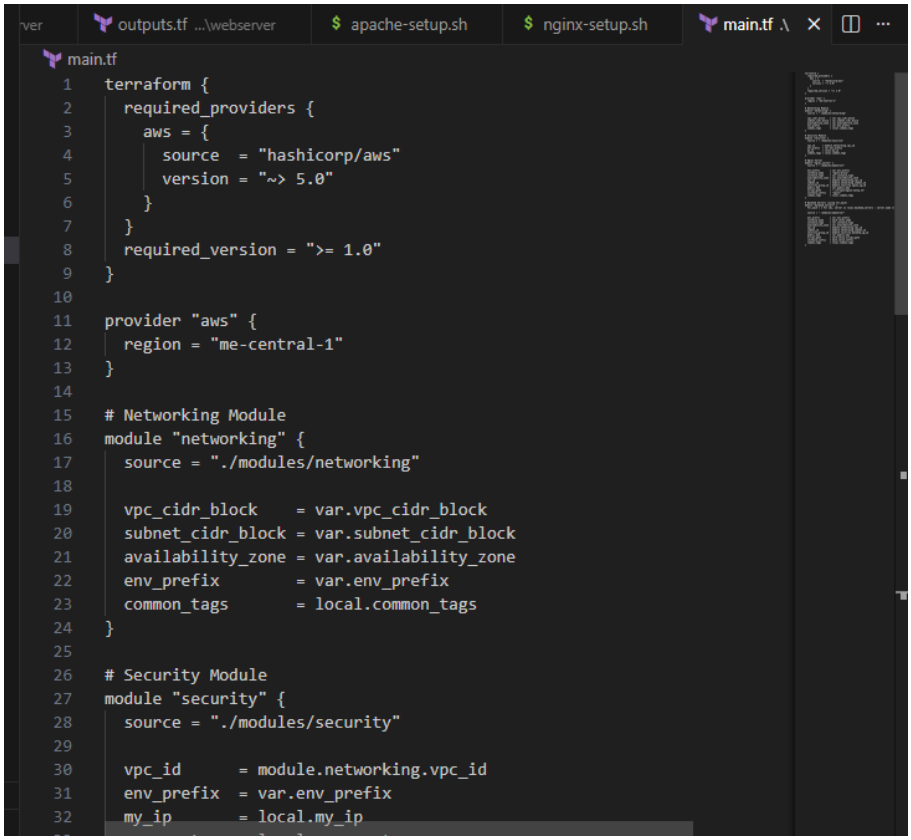
```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ curl -I http:///3.230.127.48
HTTP/1.1 301 Moved Permanently
Server: nginx/1.28.0
Date: Sun, 28 Dec 2025 15:17:18 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Location: https://3.230.127.48/
```

TASK 7 - Reverse Proxy

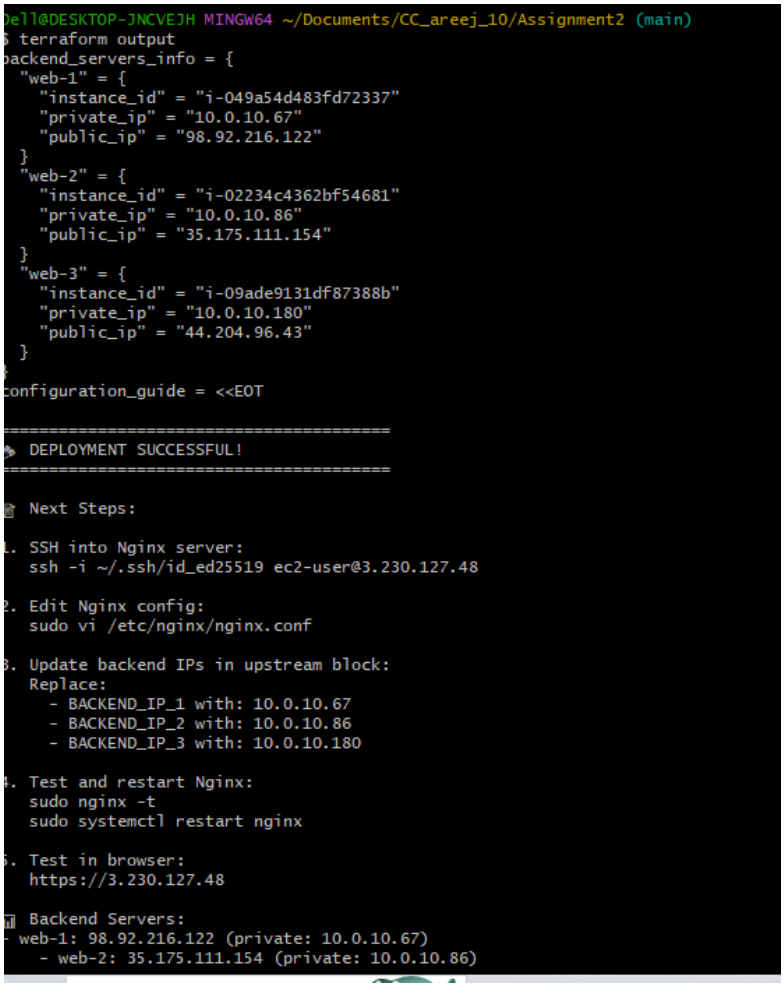
task7_apache_script.png



task7_main_tf_web1.png



task7_outputs_web1.png



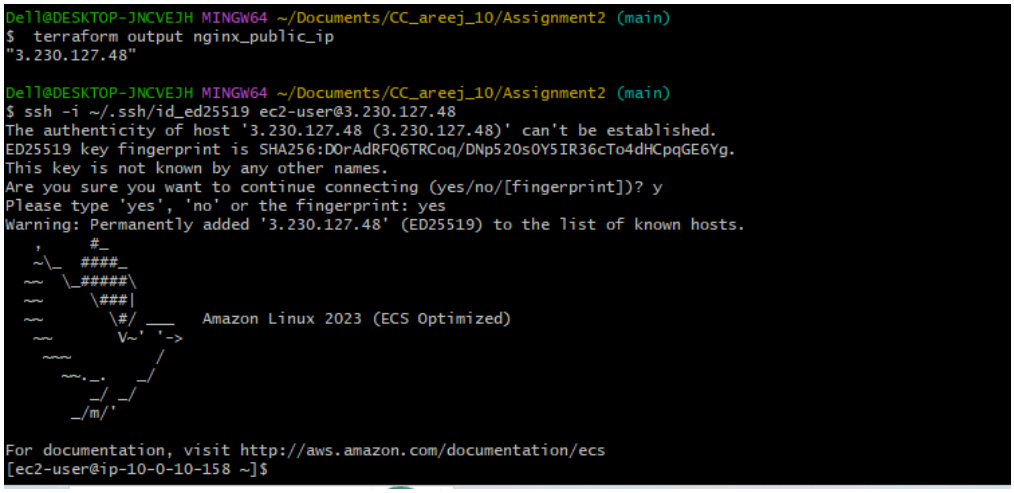
task7_terraform_apply.png

```
MINGW64:/c/Users/Dell/Documents/CC_areej_10/Assignment2
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
    - web-2: 35.175.111.154 (private: 10.0.10.86)
    - web-3: 44.204.96.43 (private: 10.0.10.180)

========================================

EOT
nginx_instance_id = "i-0e668eb55f575d3c4"
nginx_public_ip = "3.230.127.48"
subnet_id = "subnet-0d8278da93817bc6b"
vpc_id = "vpc-0df4a962ae2d7806a"
```

task7_terraform_output.png

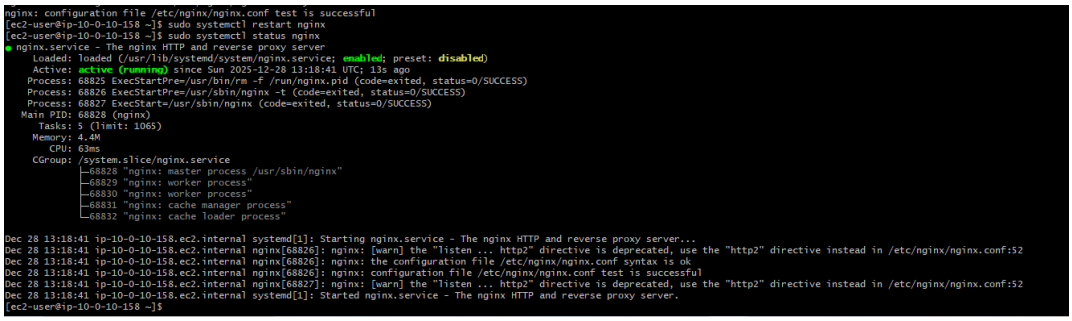
```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ terraform output
backend_servers_info = {
  "web-1" = {
    "instance_id" = "i-049a54d483fd72337"
    "private_ip" = "10.0.10.67"
    "public_ip" = "98.92.216.122"
  }
  "web-2" = {
    "instance_id" = "i-02234c4362bf54681"
    "private_ip" = "10.0.10.86"
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}
configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
    - web-2: 35.175.111.154 (private: 10.0.10.86)
```

task7_ssh_webserver.png



task7_nginx_conf_reverse_proxy.png

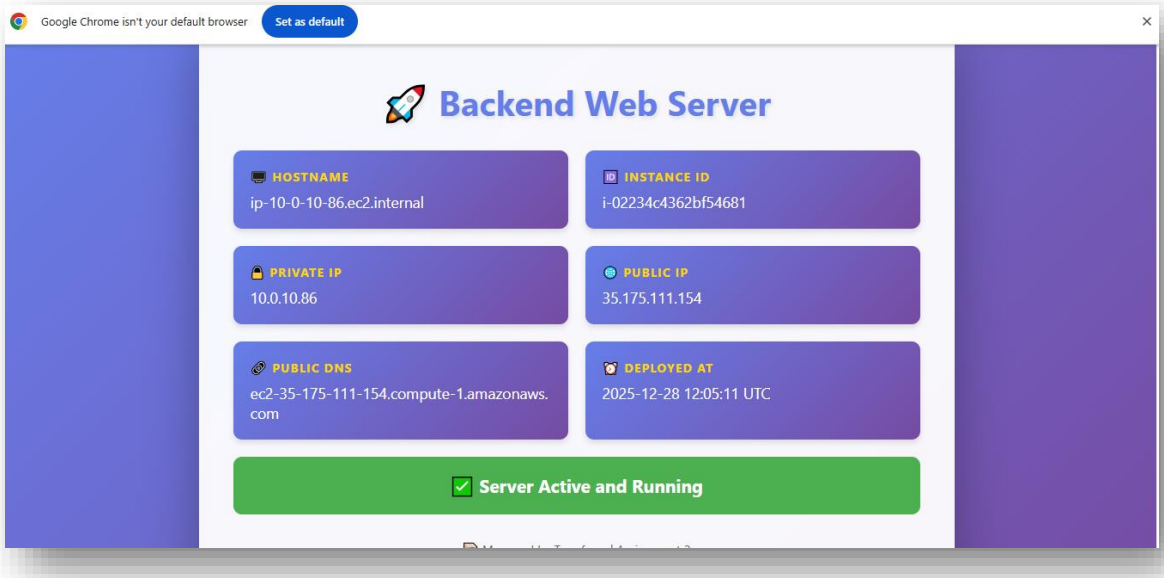
task7_nginx_restart.png



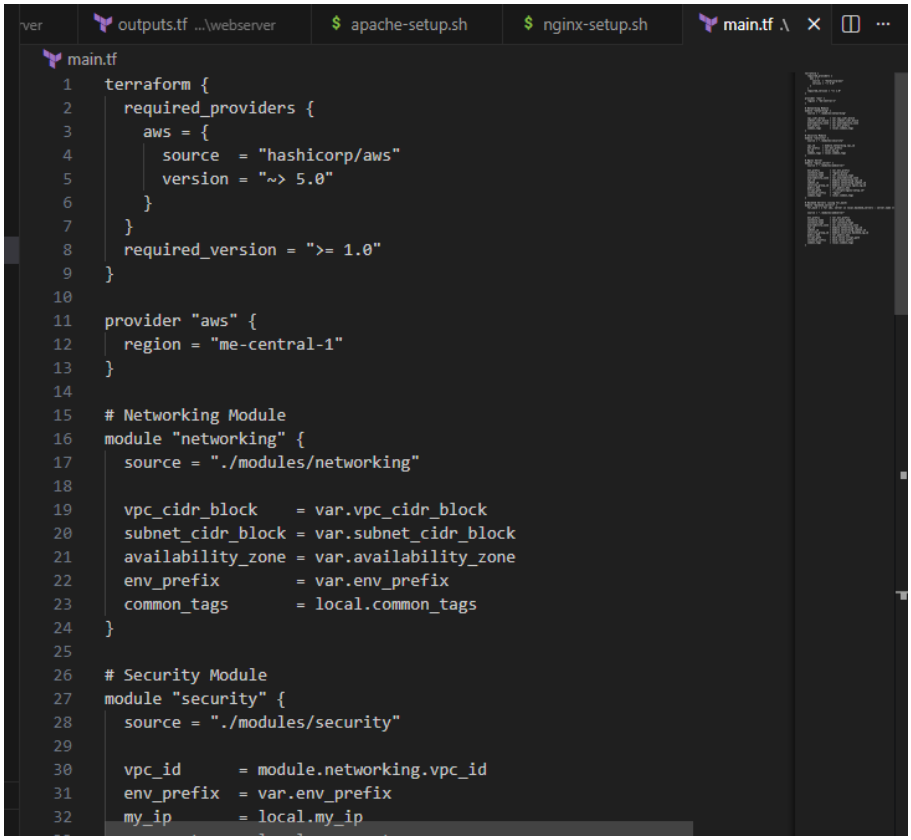
task7_ssl_cert.png

```
For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Sun Dec 28 14:56:47 2025 from 203.215.174.31
[ec2-user@ip-10-0-10-158 ~]$ sudo openssl x509 -in /etc/ssl/certs/selfsigned.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            46:d2:69:60:96:0b:85:aa:f1:9f:fa:81:bc:f9:6e:a3:79:30:1c:17
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=State, L=City, O=Organization, CN=3.230.127.48
        Validity
            Not Before: Dec 28 12:05:22 2025 GMT
            Not After : Dec 28 12:05:22 2026 GMT
        Subject: C=US, ST=State, L=City, O=Organization, CN=3.230.127.48
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ca:06:aa:d5:a8:08:2c:1d:7c:97:e8:ba:7a:e4:
                    16:f8:29:f3:6e:d1:36:69:c8:33:18:f0:3d:ad:82:
                    6c:19:76:bf:81:07:23:7d:9f:95:5a:2b:32:2d:a3:
                    5e:26:38:ae:bf:e7:fd:e7:5f:9f:74:e6:55:0f:9d:
                    f4:11:8a:72:5b:04:c6:6d:83:50:69:f7:0c:de:20:
                    ad:cb:9b:25:f8:fb:d0:4b:47:a7:d3:db:34:2e:6e:
                    96:07:86:86:bd:77:91:06:ea:5a:58:d6:71:25:57:
                    ae:f7:33:a1:48:e2:58:83:f7:a9:60:77:97:15:2f:
                    4a:15:97:bd:71:a5:85:9b:59:2d:f8:f2:3b:a9:50:
                    fe:2e:7f:2c:7b:19:40:d1:e1:e4:4a:90:72:14:a2:
                    51:50:90:f9:ab:15:b2:a2:ef:31:aa:2b:24:d5:f8:
                    3a:87:03:10:19:04:e3:67:66:7e:9a:07:70:d4:a3:
                    2c:fd:d5:ce:d7:51:0a:6e:8e:a0:a1:62:a3:7d:2c:
                    7c:c3:12:e0:8b:65:d5:11:a4:46:b2:32:e1:97:ac:
                    e5:ae:cd:0e:e2:bc:e0:1a:92:29:ab:3e:1f:f1:46:
                    45:3b:61:61:be:79:8b:68:48:5b:fc:fd:80:ab:78:
                    92:85:58:c5:35:9b:fa:35:91:e7:7b:88:e5:dc:74:
                    c1:df
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                28:D9:40:D9:86:08:D3:35:69:25:31:FB:C9:E8:07:3D:74:A9:F6:56
            X509v3 Authority Key Identifier:
                28:D9:40:D9:86:08:D3:35:69:25:31:FB:C9:E8:07:3D:74:A9:F6:56
            X509v3 Subject Alternative Name:
                IP Address:3.230.127.48
            X509v3 Basic Constraints:
```

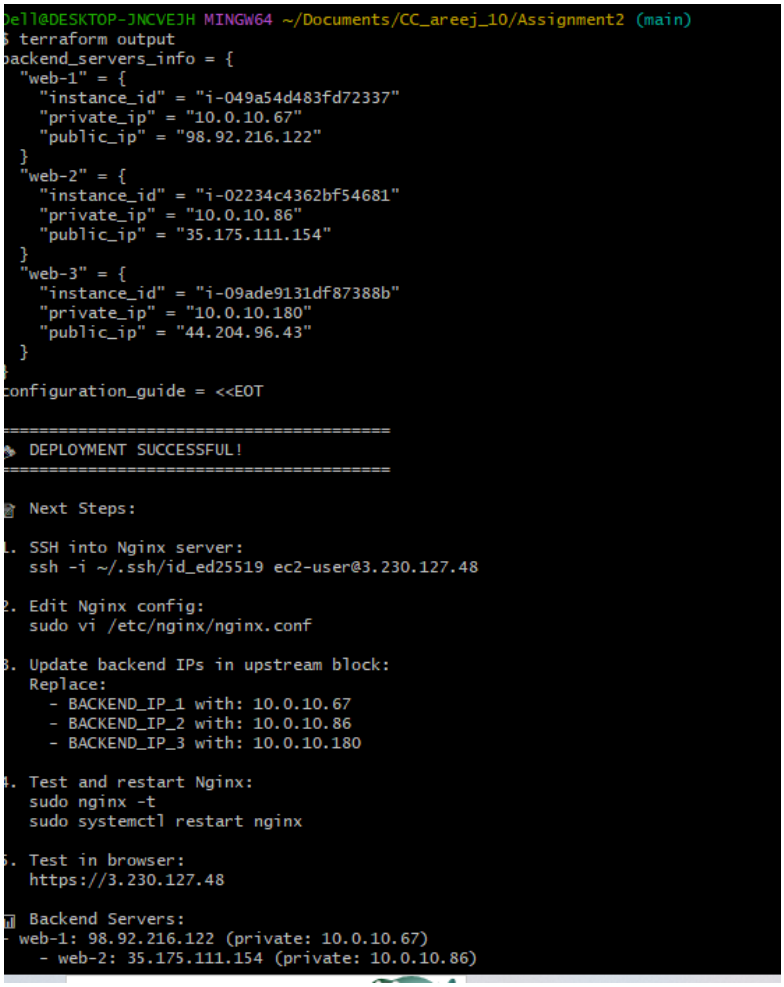
task7_reverse_proxy_browser.png



Task8_main_tf_web2.png



Task_outputs_we21.png



Task8_terraform_apply.png

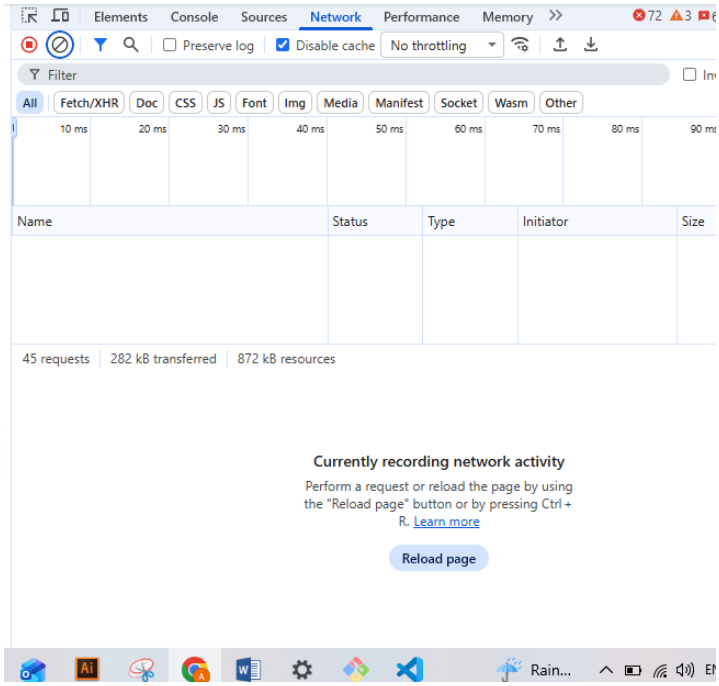
MINGW64:/c/Users/Dell/Documents/CC_areej_10/Assignment2

```
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}

configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
    - web-2: 35.175.111.154 (private: 10.0.10.86)
    - web-3: 44.204.96.43 (private: 10.0.10.180)

========================================

EOT
nginx_instance_id = "i-0e668eb55f575d3c4"
nginx_public_ip = "3.230.127.48"
subnet_id = "subnet-0d8278da93817bc6b"
vpc_id = "vpc-0df4a962ae2d7806a"
```

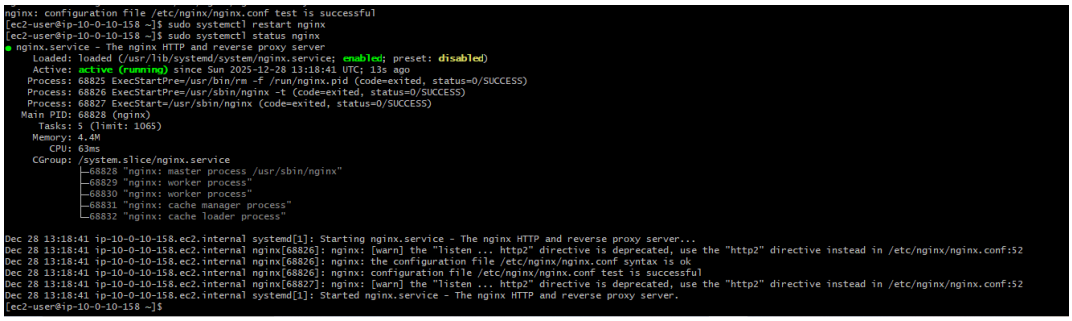
Task8_terraform_output.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ terraform output
backend_servers_info = {
  "web-1" = {
    "instance_id" = "i-049a54d483fd72337"
    "private_ip" = "10.0.10.67"
    "public_ip" = "98.92.216.122"
  }
  "web-2" = {
    "instance_id" = "i-02234c4362bf54681"
    "private_ip" = "10.0.10.86"
    "public_ip" = "35.175.111.154"
  }
  "web-3" = {
    "instance_id" = "i-09ade9131df87388b"
    "private_ip" = "10.0.10.180"
    "public_ip" = "44.204.96.43"
  }
}

configuration_guide = <<EOT

========================================
 DEPLOYMENT SUCCESSFUL!
========================================

 Next Steps:

1. SSH into Nginx server:
   ssh -i ~/.ssh/id_ed25519 ec2-user@3.230.127.48

2. Edit Nginx config:
   sudo vi /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   Replace:
     - BACKEND_IP_1 with: 10.0.10.67
     - BACKEND_IP_2 with: 10.0.10.86
     - BACKEND_IP_3 with: 10.0.10.180

4. Test and restart Nginx:
   sudo nginx -t
   sudo systemctl restart nginx

5. Test in browser:
   https://3.230.127.48

 Backend Servers:
- web-1: 98.92.216.122 (private: 10.0.10.67)
    - web-2: 35.175.111.154 (private: 10.0.10.86)
```

task8_nginx_conf_load_balancer.png

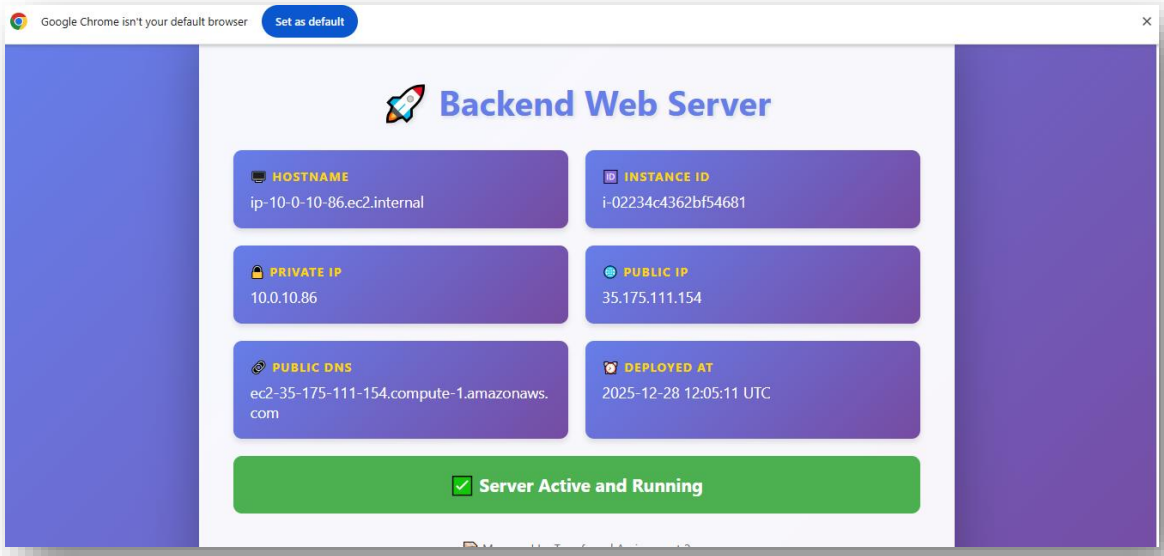


Task8_nginx_restart.png

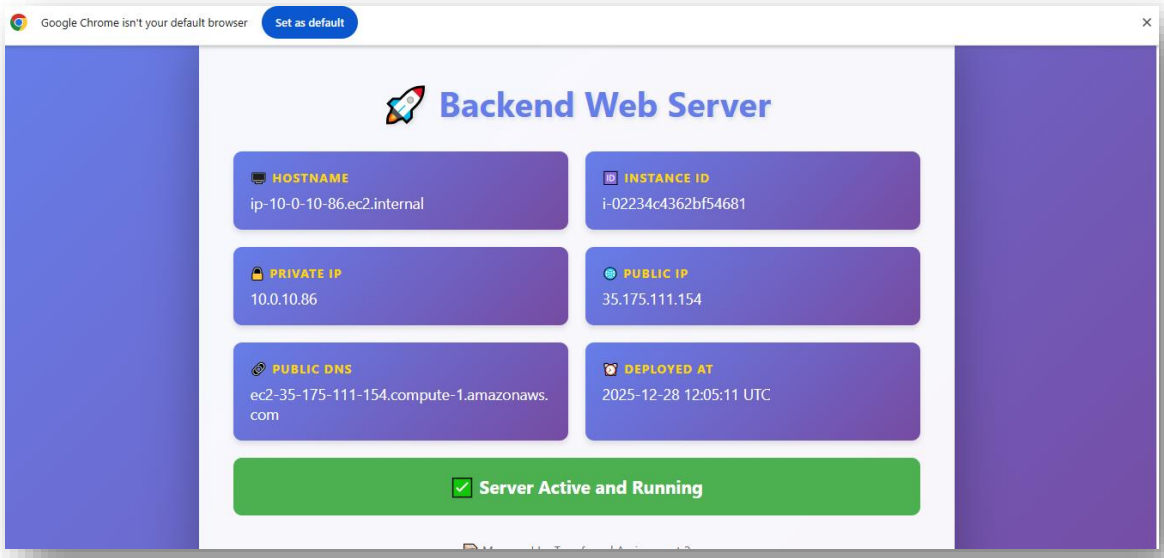


TASK 9 - High Availability

task9_ha_web1_only.png

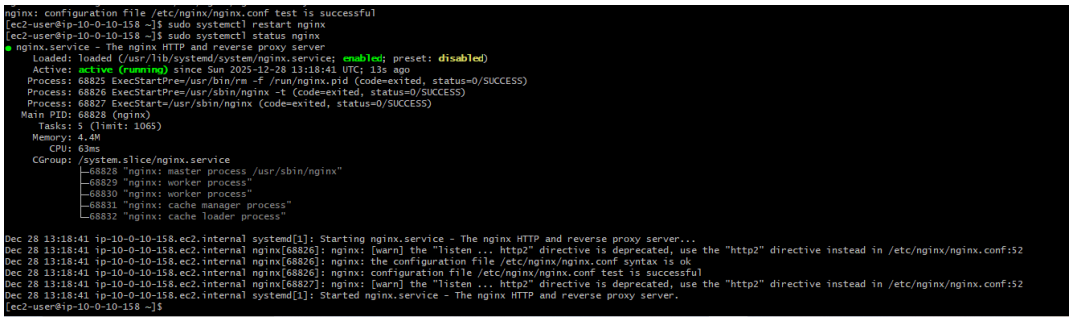


task9_ha_web2_only.png



TASK 10 – Caching

task10_nginx_restart.png



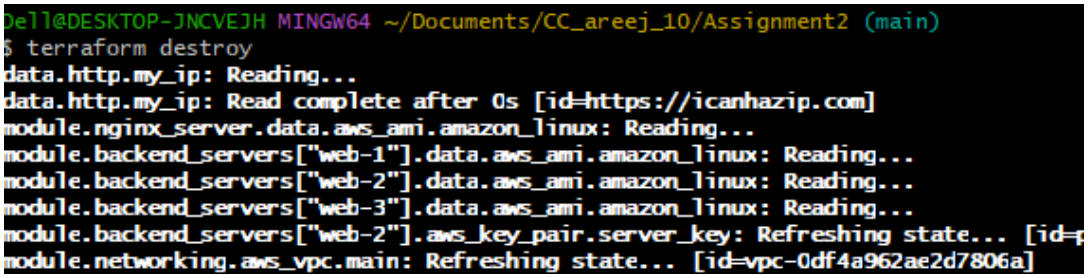
task10_cache_miss.png,task10_cache_hit.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ # Get Nginx IP
NGINX_IP=$(terraform output -raw nginx_public_ip)

# First request - should show MISS
curl -k -I https://$NGINX_IP 2>&1 | grep -i "x-cache"

# Wait 2 seconds, then second request - should show HIT
sleep 2
curl -k -I https://$NGINX_IP 2>&1 | grep -i "x-cache"

# Third request - should show HIT
curl -k -I https://$NGINX_IP 2>&1 | grep -i "x-cache"
X-Cache-Status: MISS
X-Cache-Status: HIT
X-Cache-Status: HIT
```

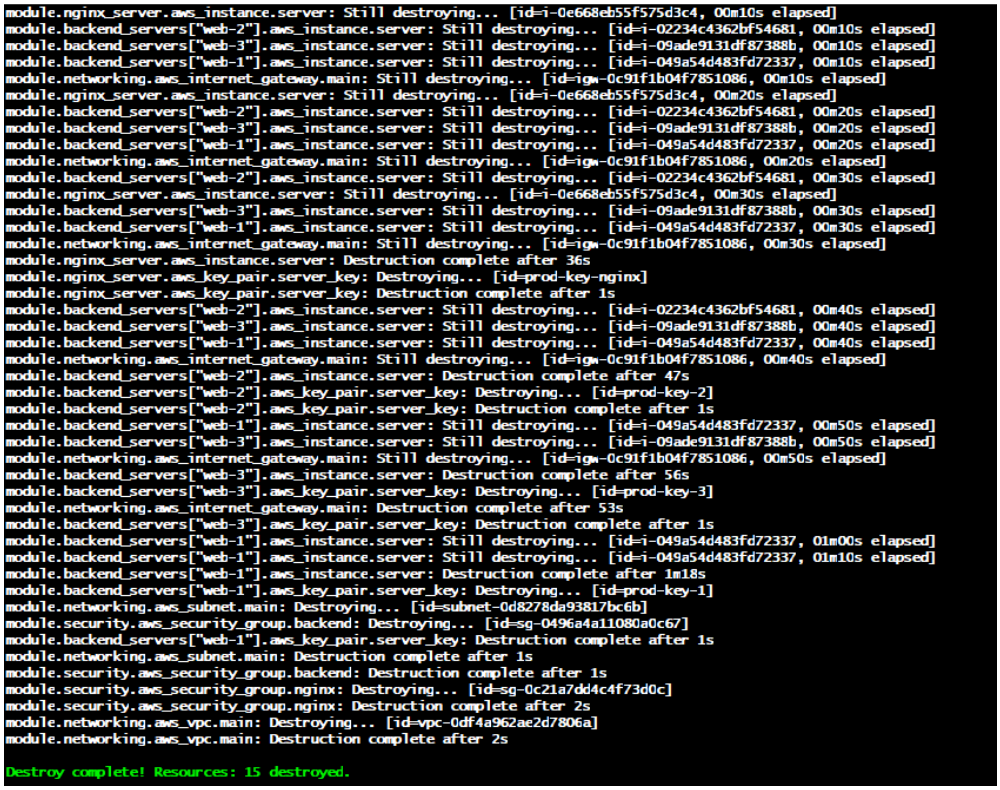
task10_cache_directory.png

```
Dell@DESKTOP-JNCVEJH MINGW64 ~/Documents/CC_areej_10/Assignment2 (main)
$ # Get Nginx IP
NGINX_IP=$(terraform output -raw nginx_public_ip)

# First request - should show MISS
curl -k -I https://$NGINX_IP 2>&1 | grep -i "x-cache"

# Wait 2 seconds, then second request - should show HIT
sleep 2
curl -k -I https://$NGINX_IP 2>&1 | grep -i "x-cache"

# Third request - should show HIT
curl -k -I https://$NGINX_IP 2>&1 | grep -i "x-cache"
X-Cache-Status: MISS
X-Cache-Status: HIT
X-Cache-Status: HIT
```

cleanup_destroy_prompt.png



cleanup_destroy_complete.png



cleanup_state_empty.png