Name: Zuha Irfan

Roll No: 2023-BSE-073

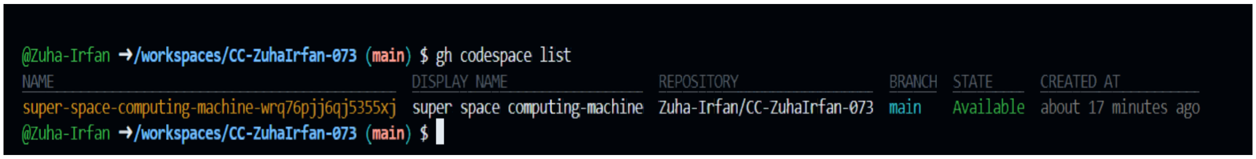
Subject: Cloud Computing

Semester & Section: V-B

Lab: 11

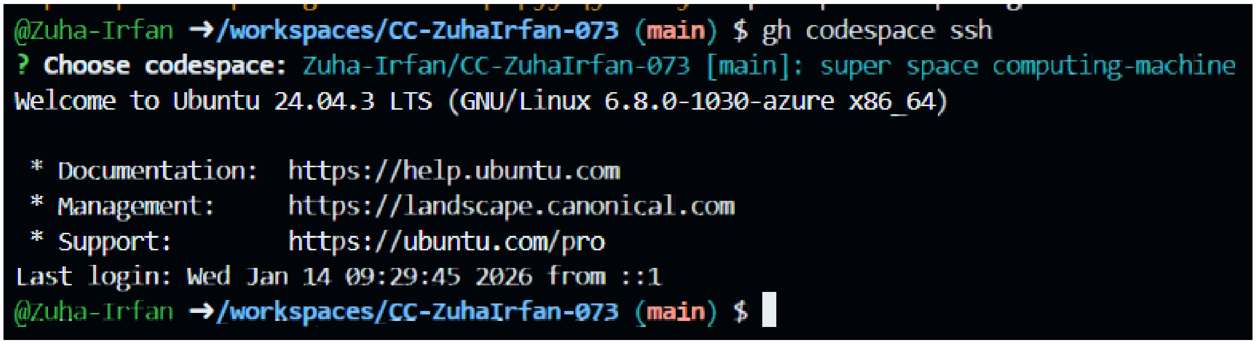
Task 0 Lab Setup (Codespace & GH CLI)

- task0_codespace_create_and_list. Png



```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ gh codespace list
NAME                                          DISPLAY NAME                   REPOSITORY                    BRANCH  STATE      CREATED AT
super-space-computing-machine-wrq76pjj6qj5355xj  super space computing-machine  Zuha-Irfan/CC-ZuhaIrfan-073  main    Available  about 17 minutes ago
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $
```

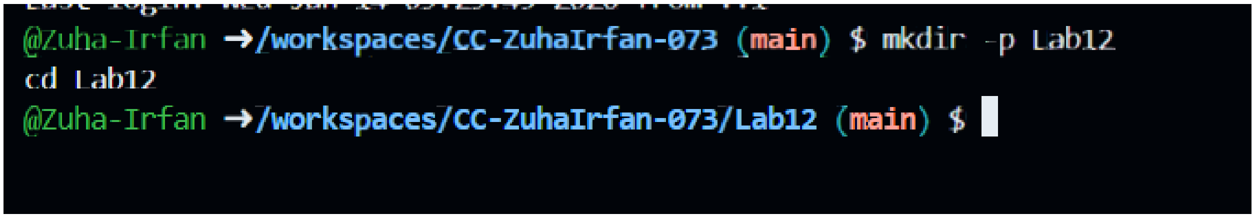
- task0_codespace_ssh_connected.png



```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ gh codespace ssh
? Choose codespace: Zuha-Irfan/CC-ZuhaIrfan-073 [main]: super space computing-machine
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
Last login: Wed Jan 14 09:29:45 2026 from ::1
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $
```

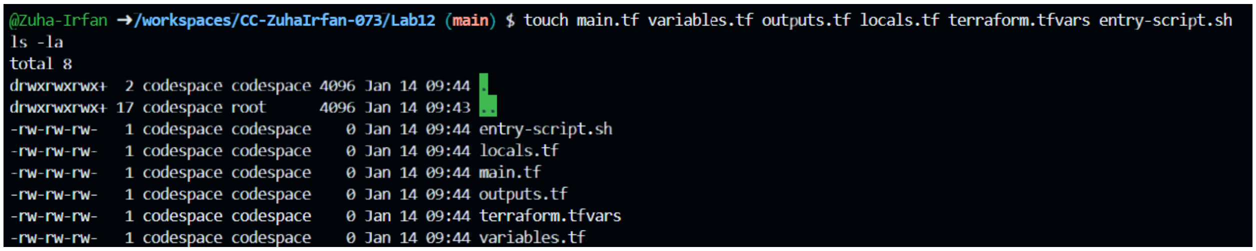
Task 1 — Organize Terraform code into separate files

- task1_project_directory.png



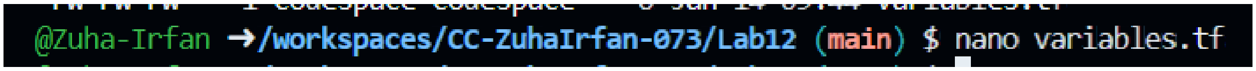
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ mkdir -p Lab12
cd Lab12
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $
```

- task1_files_created.png

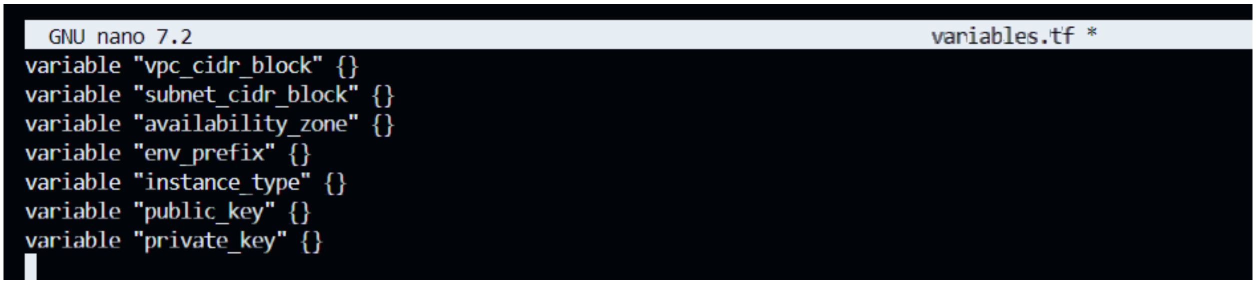


```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ touch main.tf variables.tf outputs.tf locals.tf terraform.tfvars entry-script.sh
ls -la
total 8
drwxrwxrwx+  2 codespace codespace 4096 Jan 14 09:44 .
drwxrwxrwx+ 17 codespace root      4096 Jan 14 09:43 ..
-rw-rw-rw-   1 codespace codespace    0 Jan 14 09:44 entry-script.sh
-rw-rw-rw-   1 codespace codespace    0 Jan 14 09:44 locals.tf
-rw-rw-rw-   1 codespace codespace    0 Jan 14 09:44 main.tf
-rw-rw-rw-   1 codespace codespace    0 Jan 14 09:44 outputs.tf
-rw-rw-rw-   1 codespace codespace    0 Jan 14 09:44 terraform.tfvars
-rw-rw-rw-   1 codespace codespace    0 Jan 14 09:44 variables.tf
```

- task1_variables_tf.png

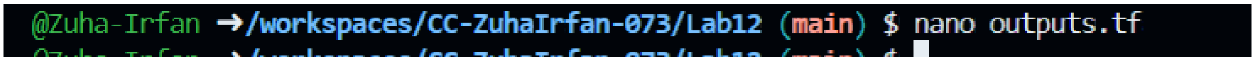


```
-rw-rw-rw-   1 codespace codespace    0 Jan 14 09:44 variables.tf
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano variables.tf
```

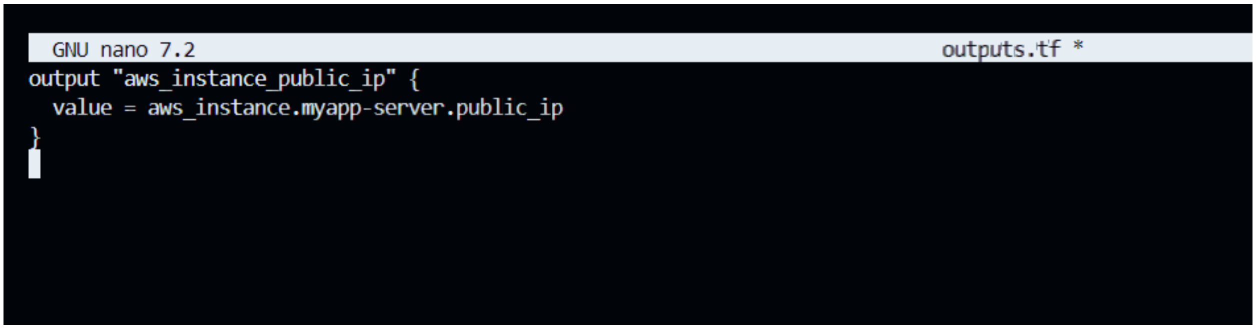


```
  GNU nano 7.2                                                          variables.tf *
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "instance_type" {}
variable "public_key" {}
variable "private_key" {}
```

- task1_outputs_tf.png

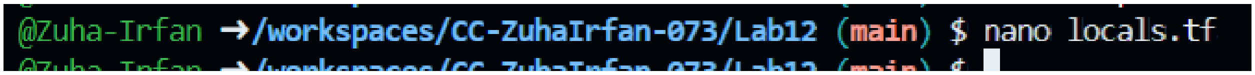


```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano outputs.tf
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $
```

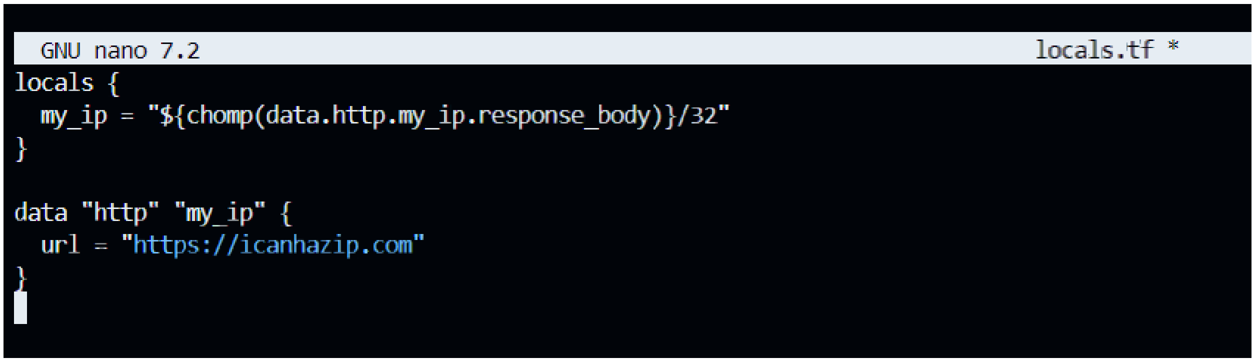


```
  GNU nano 7.2                                                          outputs.tf *
output "aws_instance_public_ip" {
  value = aws_instance.myapp-server.public_ip
}
```

- task1_locals_tf.png

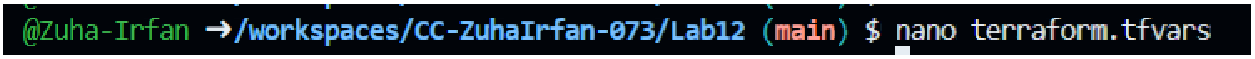


```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano locals.tf
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $
```

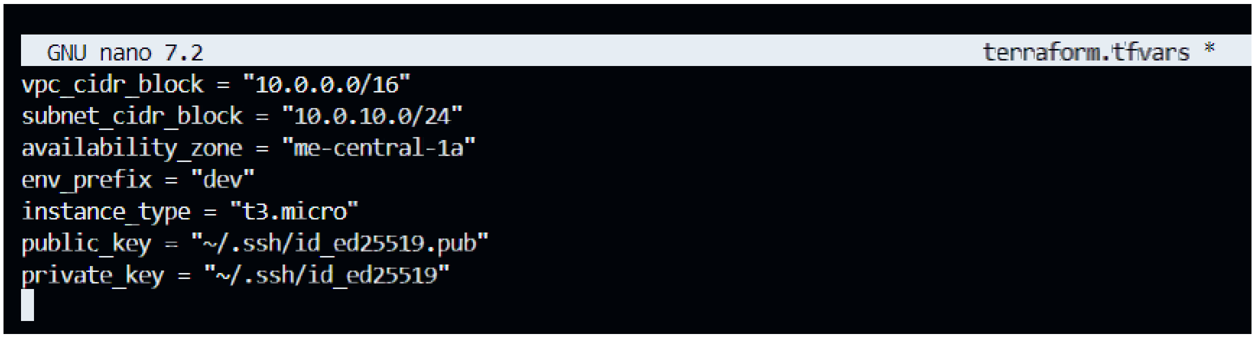


```
  GNU nano 7.2                                                      locals.tf *
locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"
}

data "http" "my_ip" {
  url = "https://icanhazip.com"
}
```

- task1_terraform_tfvars.png

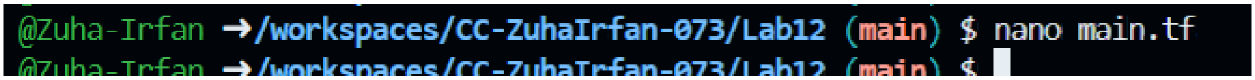


```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano terraform.tfvars
```

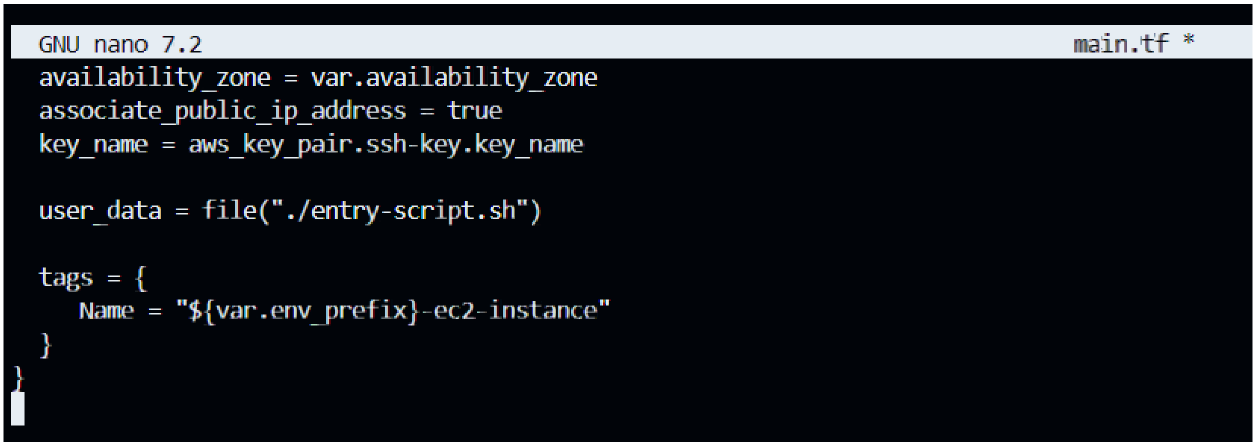


```
  GNU nano 7.2                                               terraform.tfvars *
vpc_cidr_block = "10.0.0.0/16"
subnet_cidr_block = "10.0.10.0/24"
availability_zone = "me-central-1a"
env_prefix = "dev"
instance_type = "t3.micro"
public_key = "~/.ssh/id_ed25519.pub"
private_key = "~/.ssh/id_ed25519"
```

- task1_main_tf.png

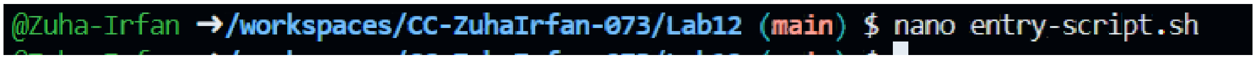


```
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano main.tf
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $
```

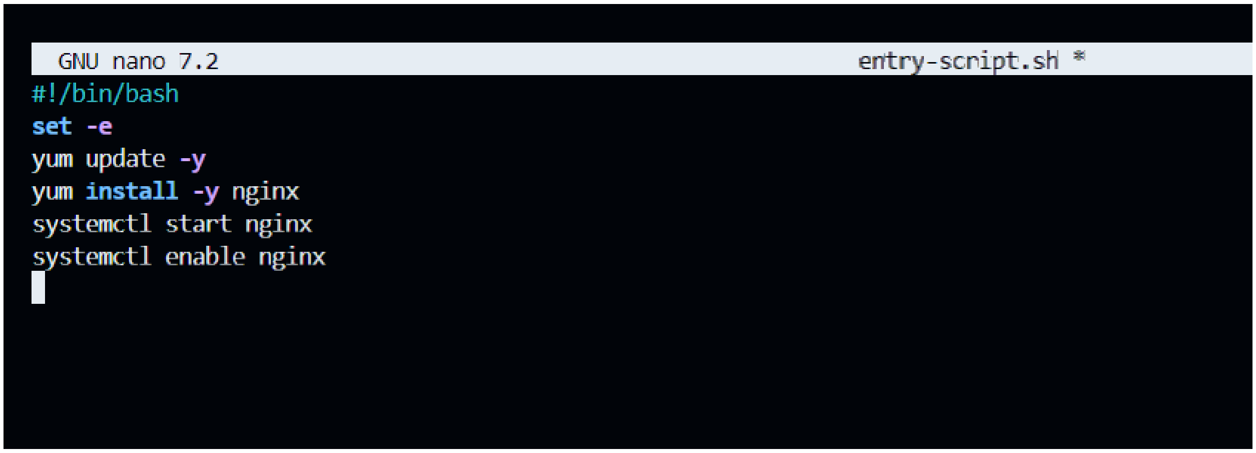


```
  availability_zone = var.availability_zone
  associate_public_ip_address = true
  key_name = aws_key_pair.ssh-key.key_name

  user_data = file("./entry-script.sh")

  tags = {
     Name = "${var.env_prefix}-ec2-instance"
  }
}
```

- task1_entry_script.png

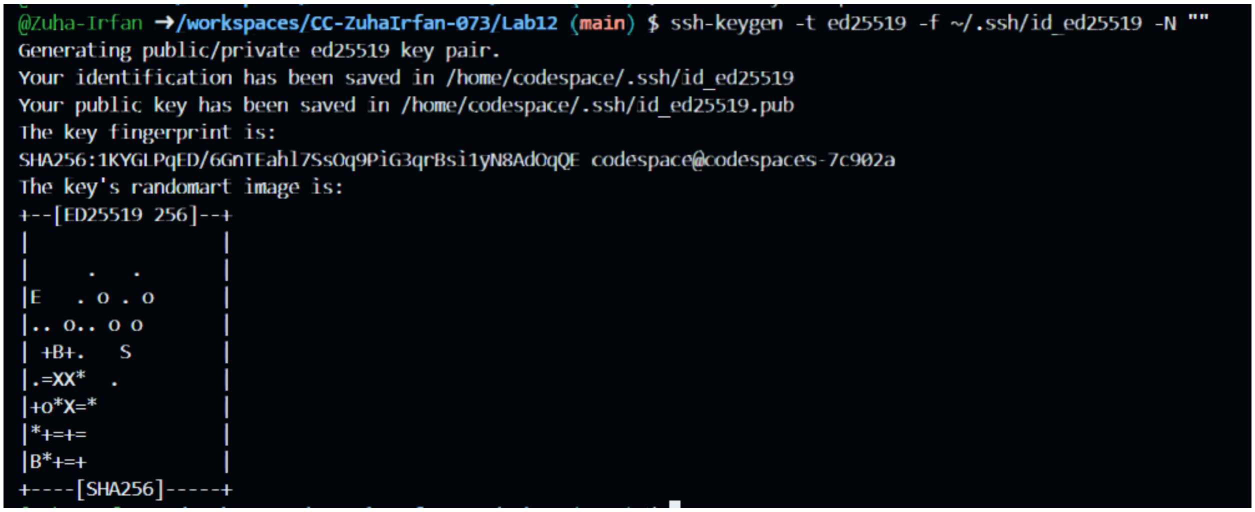


```
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano entry-script.sh
```



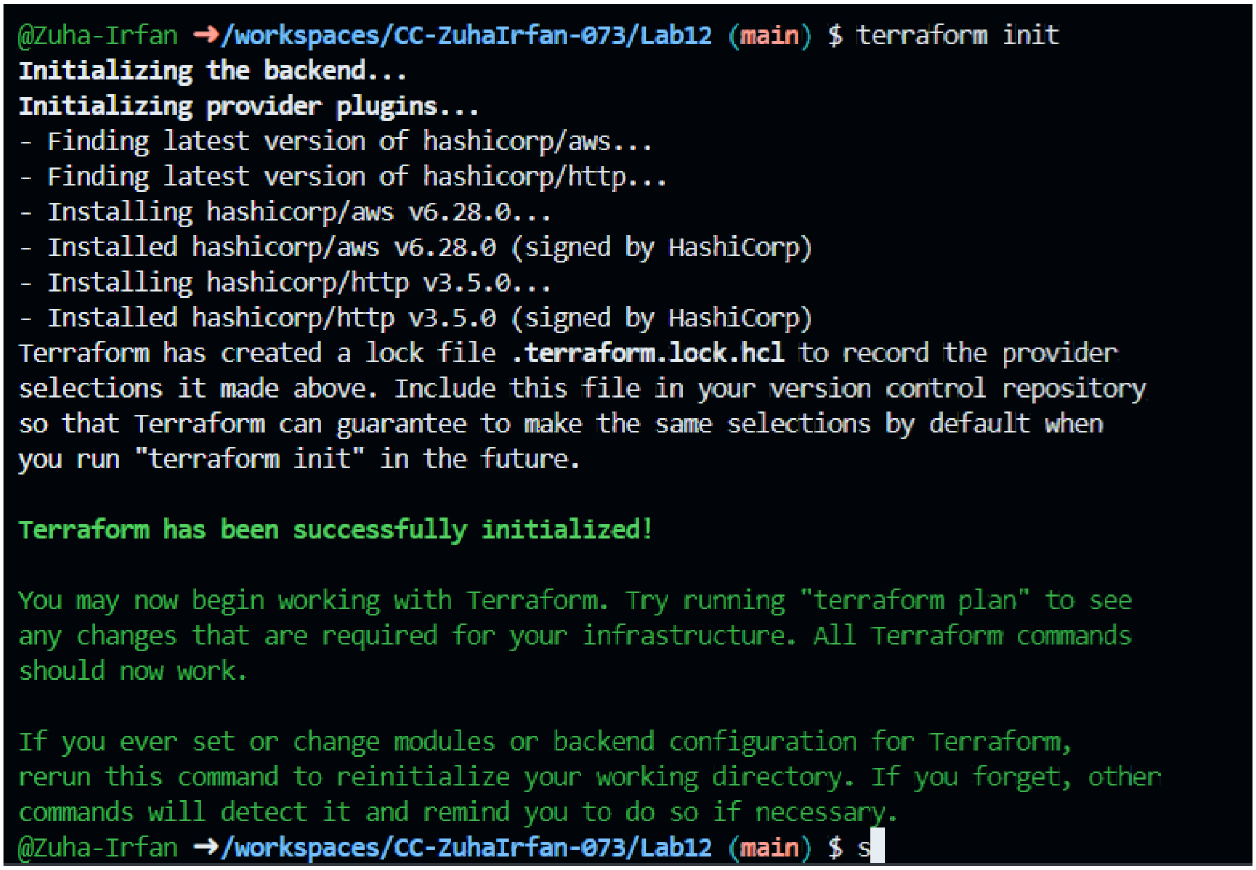
```
#!/bin/bash
set -e
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx
```

- task1_ssh_keygen.png



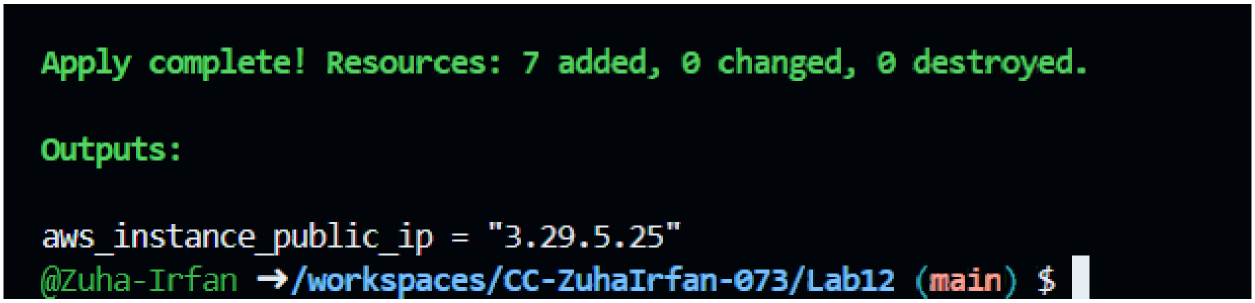
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
Your identification has been saved in /home/codespace/.ssh/id_ed25519
Your public key has been saved in /home/codespace/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:1KYGLPqED/6GnTEahl7SsOq9PiG3qrBsi1yN8AdOqQE codespace@codespaces-7c902a
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|     .   .       |
|E   . o . o      |
|.. o.. o o       |
| +B+.   S        |
|.=XX*  .         |
|+o*X=*           |
|*+=+=            |
|B*+=+            |
+----[SHA256]-----+
```

- task1_terraform_init.png



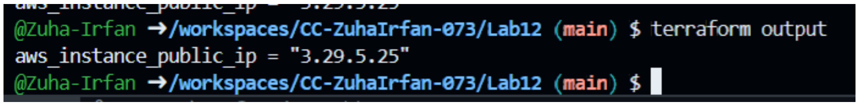
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/http...
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (signed by HashiCorp)
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (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.
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ s
```

- task1_terraform_apply.png



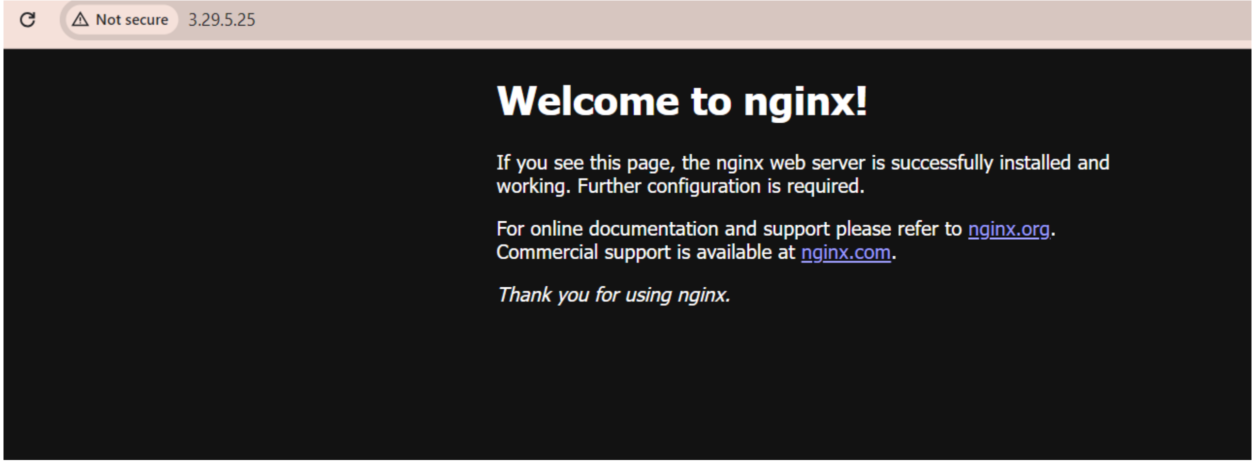
```
Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

aws_instance_public_ip = "3.29.5.25"
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ 
```

- task1_terraform_output.png



```
aws_instance_public_ip = "3.29.5.25"
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform output
aws_instance_public_ip = "3.29.5.25"
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ 
```

- task1_nginx_browser.png



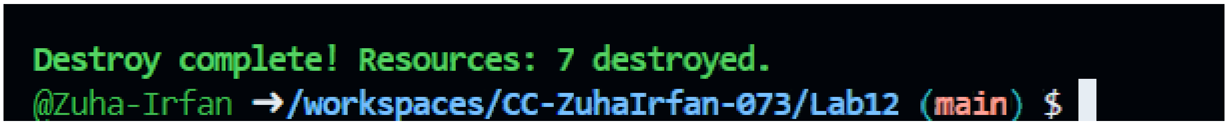
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

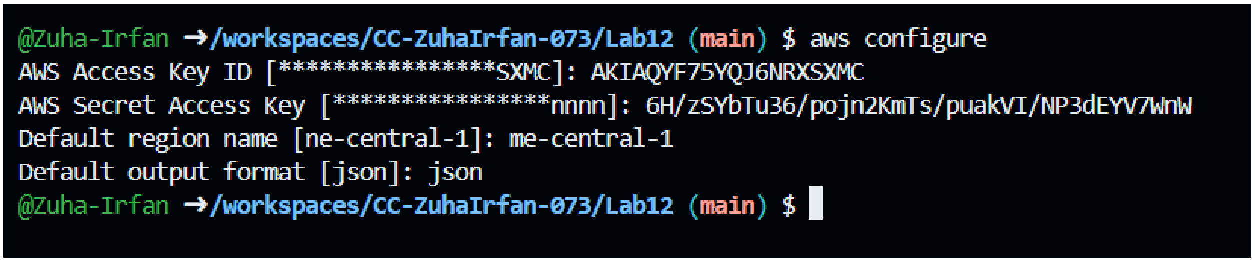
- task1_terraform_destroy.png



```
Destroy complete! Resources: 7 destroyed.
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ 
```

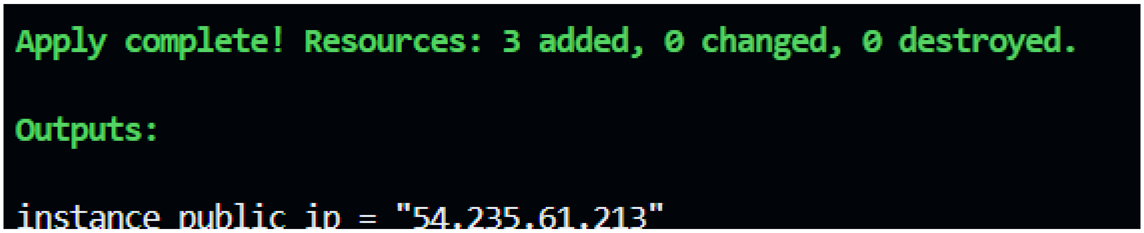
Task 2 — Use remote-exec provisioner

- task2_main_tf_remote_exec.png



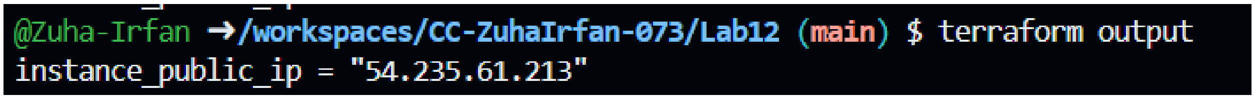
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ aws configure
AWS Access Key ID [****************SXMC]: AKIAQYF75YQJ6NRXSXMC
AWS Secret Access Key [****************nnnn]: 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
Default region name [ne-central-1]: me-central-1
Default output format [json]: json
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $
```

- task2_terraform_apply.png



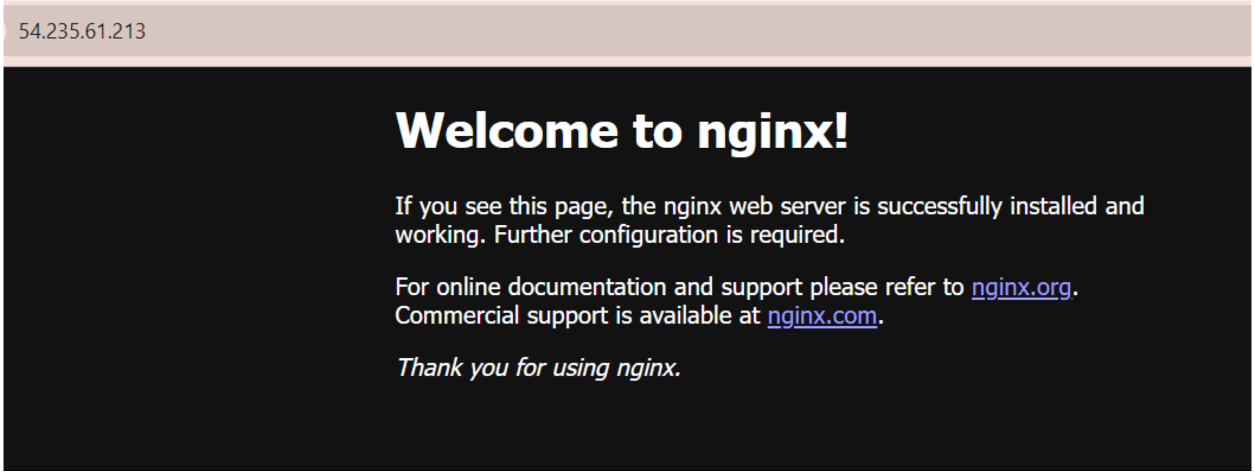
```
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

instance_public_ip = "54.235.61.213"
```

- task2_terraform_output.png



```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform output
instance_public_ip = "54.235.61.213"
```

- task2_nginx_browser.png



Welcome to nginx!

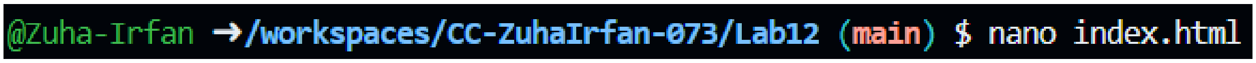
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

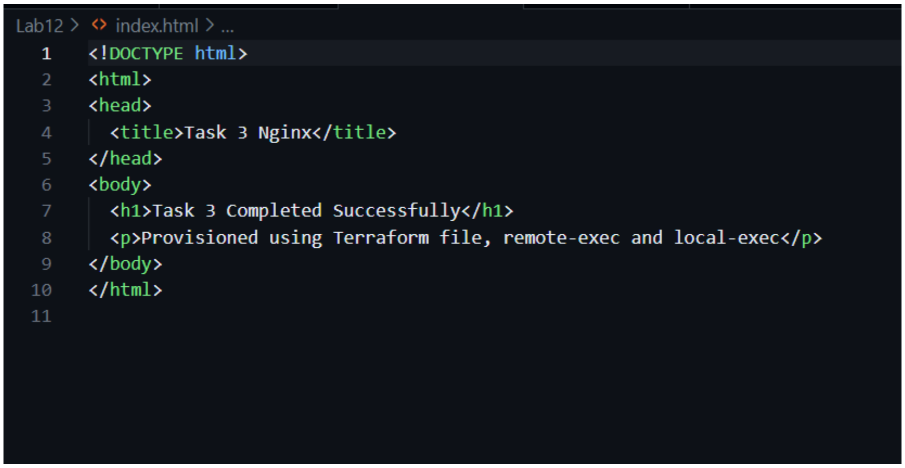
Thank you for using nginx.

Task 3 — Use file and local-exec provisioners

- task3_main_tf_all_provisioners.png

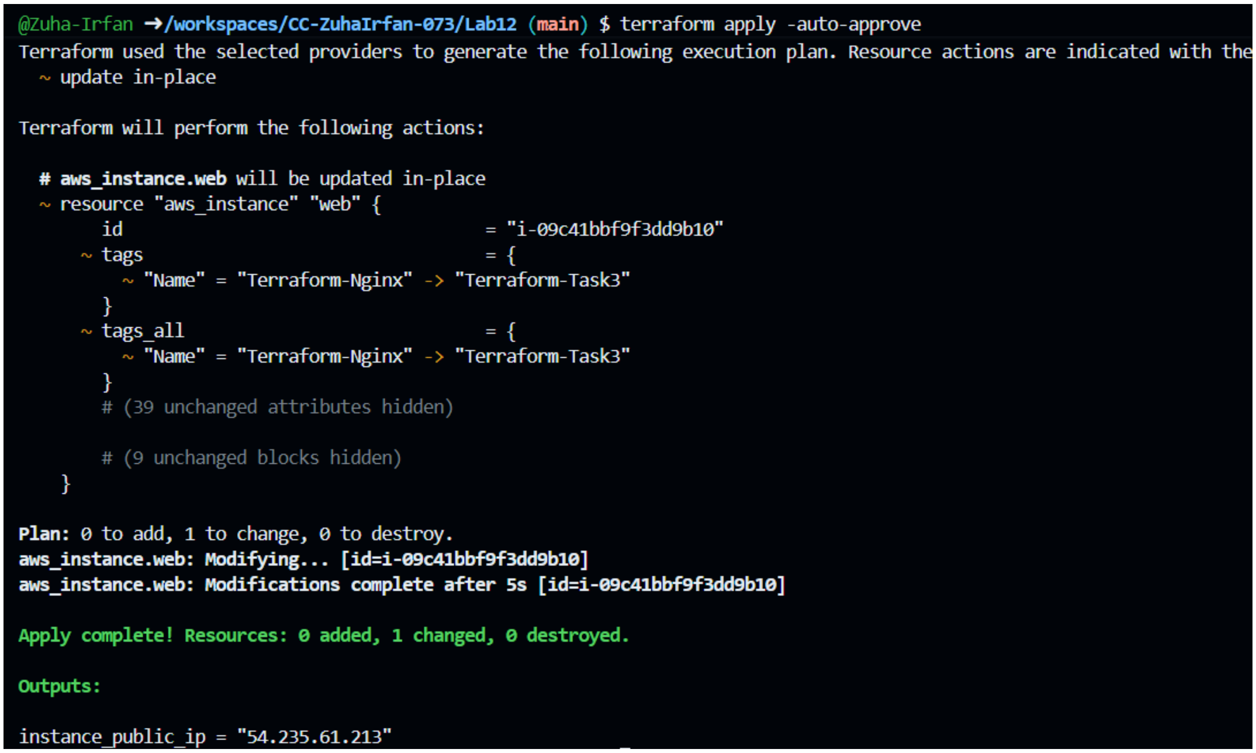


```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano index.html
```



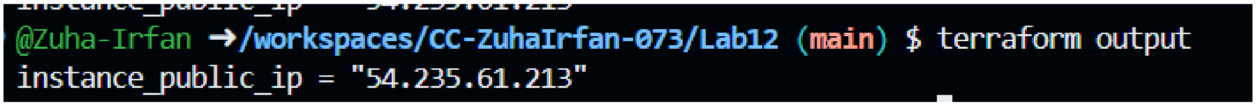
```html
<!DOCTYPE html>
<html>
<head>
  <title>Task 3 Nginx</title>
</head>
<body>
  <h1>Task 3 Completed Successfully</h1>
  <p>Provisioned using Terraform file, remote-exec and local-exec</p>
</body>
</html>
```

- task3_terraform_apply.png



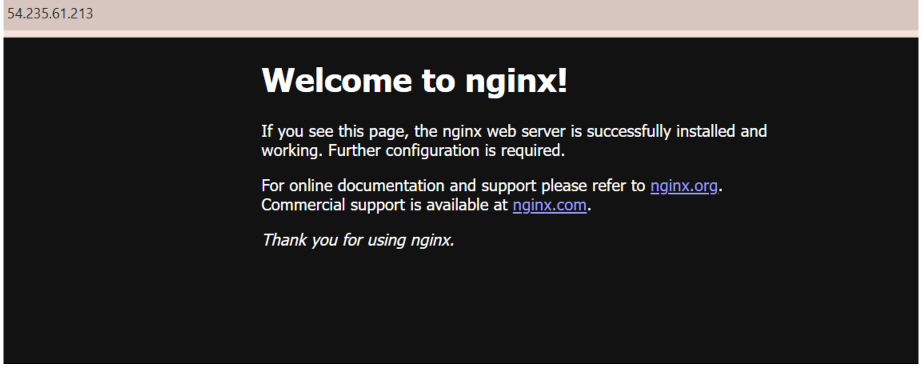
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform apply -auto-approve
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
  ~ update in-place

Terraform will perform the following actions:

  # aws_instance.web will be updated in-place
  ~ resource "aws_instance" "web" {
        id                                   = "i-09c41bbf9f3dd9b10"
      ~ tags                                 = {
          ~ "Name" = "Terraform-Nginx" -> "Terraform-Task3"
        }
      ~ tags_all                             = {
          ~ "Name" = "Terraform-Nginx" -> "Terraform-Task3"
        }
        # (39 unchanged attributes hidden)

        # (9 unchanged blocks hidden)
    }

Plan: 0 to add, 1 to change, 0 to destroy.
aws_instance.web: Modifying... [id=i-09c41bbf9f3dd9b10]
aws_instance.web: Modifications complete after 5s [id=i-09c41bbf9f3dd9b10]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.

Outputs:

instance_public_ip = "54.235.61.213"
```

- task3_terraform_output.png



```
instance_public_ip = "54.235.61.213"
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform output
instance_public_ip = "54.235.61.213"
```

- task3_nginx_browser.png



54.235.61.213

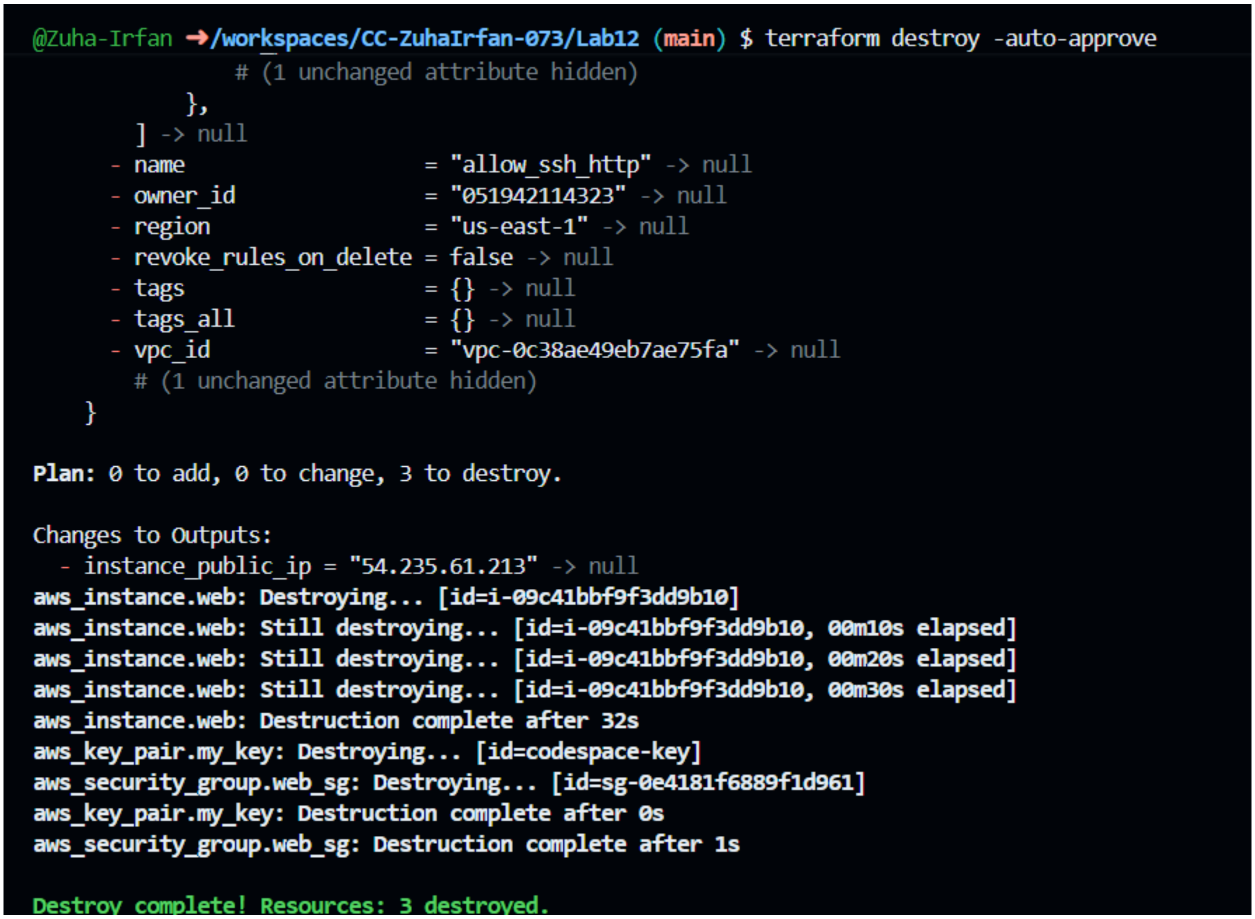
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

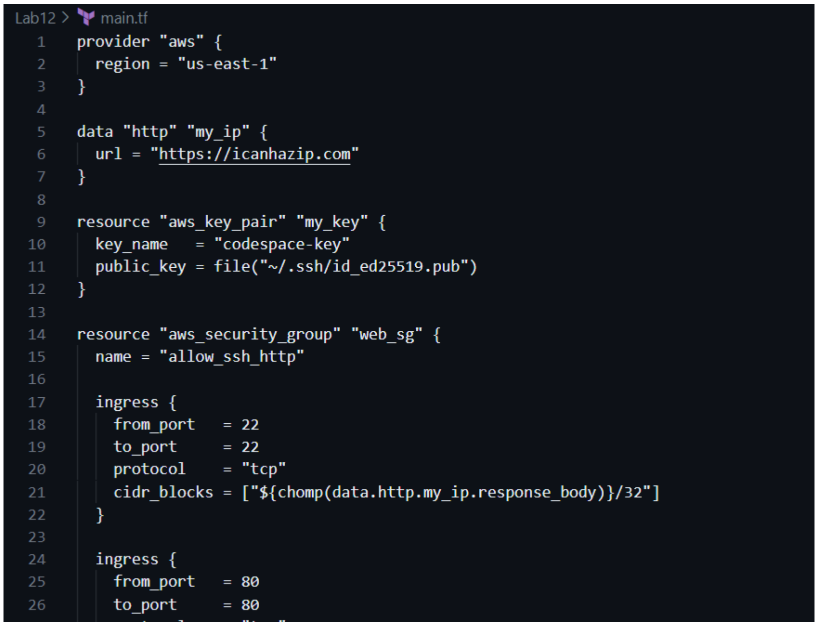
Thank you for using nginx.

- task3_terraform_destroy.png



```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform destroy -auto-approve
                # (1 unchanged attribute hidden)
            },
        ] -> null
      - name                   = "allow_ssh_http" -> null
      - owner_id               = "051942114323" -> null
      - region                 = "us-east-1" -> null
      - revoke_rules_on_delete = false -> null
      - tags                   = {} -> null
      - tags_all               = {} -> null
      - vpc_id                 = "vpc-0c38ae49eb7ae75fa" -> null
        # (1 unchanged attribute hidden)
    }

Plan: 0 to add, 0 to change, 3 to destroy.

Changes to Outputs:
  - instance_public_ip = "54.235.61.213" -> null
aws_instance.web: Destroying... [id=i-09c41bbf9f3dd9b10]
aws_instance.web: Still destroying... [id=i-09c41bbf9f3dd9b10, 00m10s elapsed]
aws_instance.web: Still destroying... [id=i-09c41bbf9f3dd9b10, 00m20s elapsed]
aws_instance.web: Still destroying... [id=i-09c41bbf9f3dd9b10, 00m30s elapsed]
aws_instance.web: Destruction complete after 32s
aws_key_pair.my_key: Destroying... [id=codespace-key]
aws_security_group.web_sg: Destroying... [id=sg-0e4181f6889f1d961]
aws_key_pair.my_key: Destruction complete after 0s
aws_security_group.web_sg: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
```

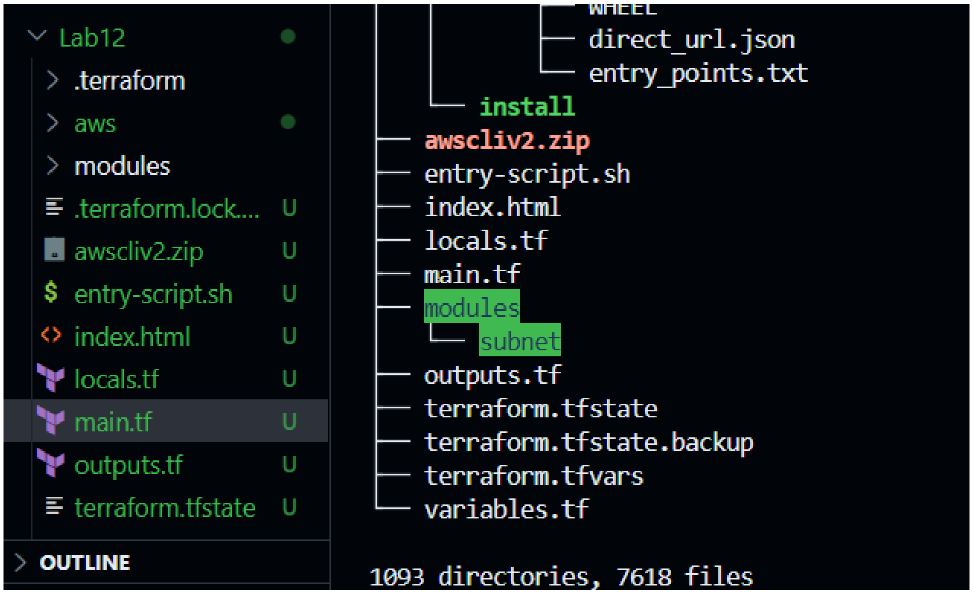
- task3_main_tf_restored.png



```
provider "aws" {
  region = "us-east-1"
}

data "http" "my_ip" {
  url = "https://icanhazip.com"
}

resource "aws_key_pair" "my_key" {
  key_name   = "codespace-key"
  public_key = file("~/.ssh/id_ed25519.pub")
}

resource "aws_security_group" "web_sg" {
  name = "allow_ssh_http"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["${chomp(data.http.my_ip.response_body)}/32"]
  }

  ingress {
    from_port   = 80
    to_port     = 80
```

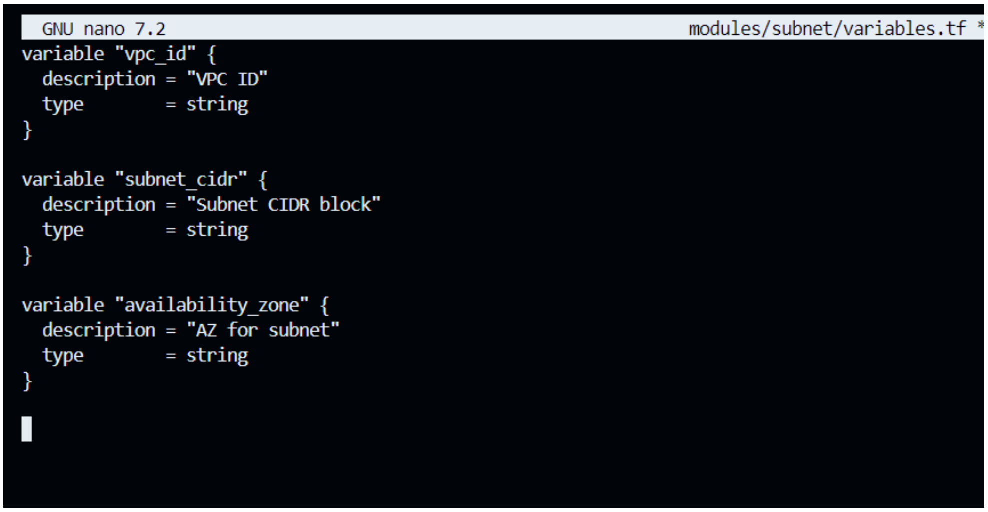
Task 4 — Terraform Create modules (subnet module)

- task4_module_structure.png



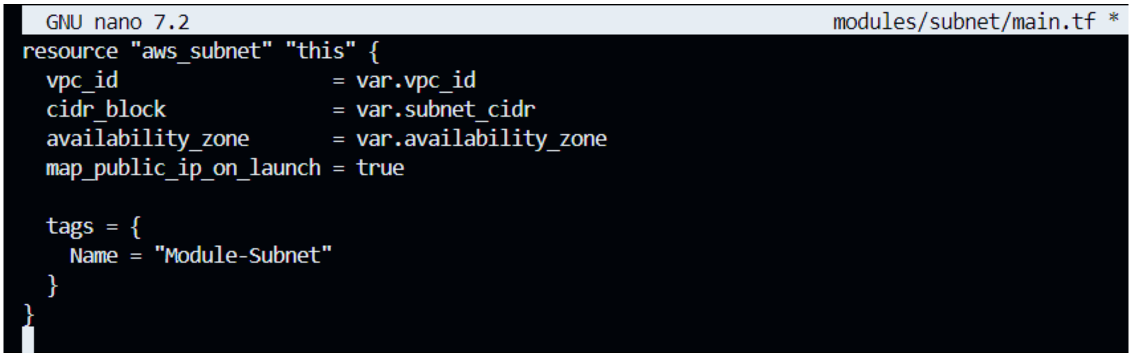
```
│       │           ├── WHEEL
│       │           ├── direct_url.json
│       │           └── entry_points.txt
│       └── install
├── awscliv2.zip
├── entry-script.sh
├── index.html
├── locals.tf
├── main.tf
├── modules
│   └── subnet
├── outputs.tf
├── terraform.tfstate
├── terraform.tfstate.backup
├── terraform.tfvars
└── variables.tf

1093 directories, 7618 files
```

- task4_subnet_variables.png



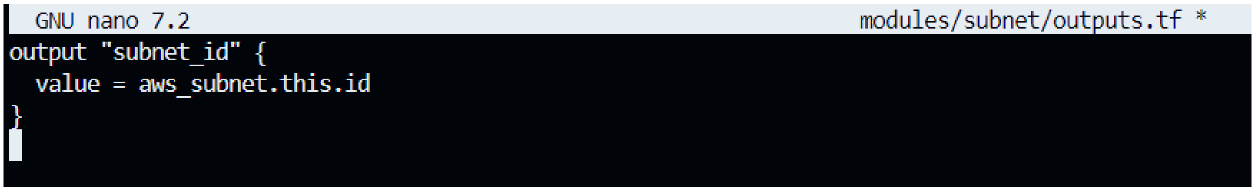
```
variable "vpc_id" {
  description = "VPC ID"
  type        = string
}

variable "subnet_cidr" {
  description = "Subnet CIDR block"
  type        = string
}

variable "availability_zone" {
  description = "AZ for subnet"
  type        = string
}
```

- task4_subnet_main.png



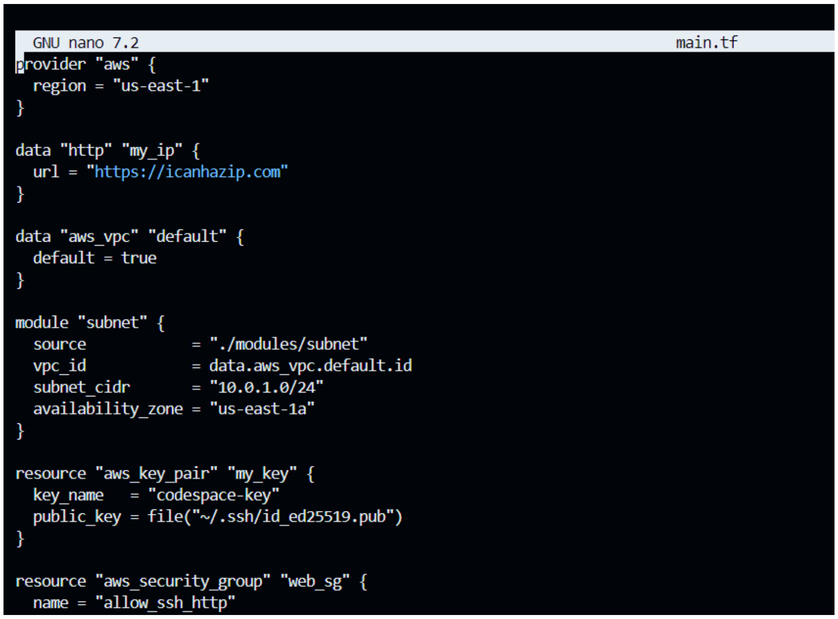
```
resource "aws_subnet" "this" {
  vpc_id                  = var.vpc_id
  cidr_block              = var.subnet_cidr
  availability_zone       = var.availability_zone
  map_public_ip_on_launch = true

  tags = {
    Name = "Module-Subnet"
  }
}
```

- task4_subnet_outputs.png



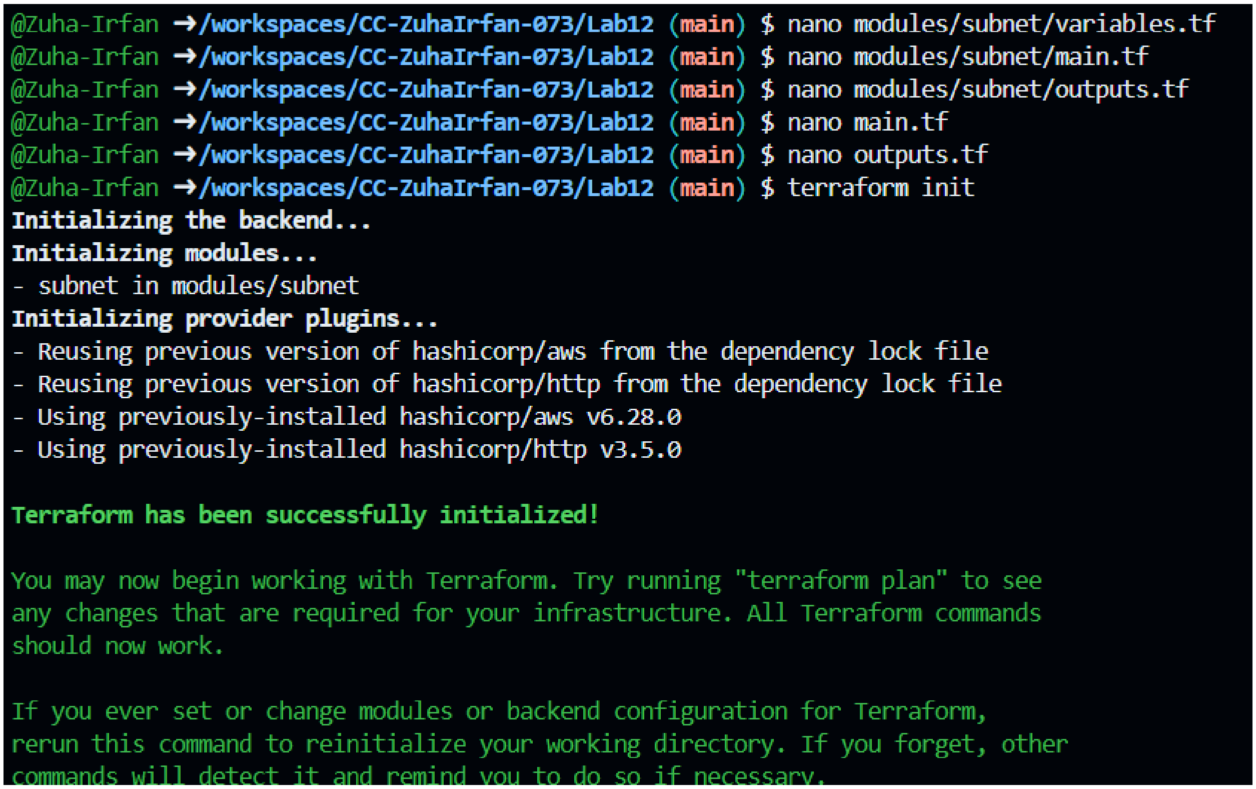
```
output "subnet_id" {
  value = aws_subnet.this.id
}
```

- task4_main_tf_with_module.png



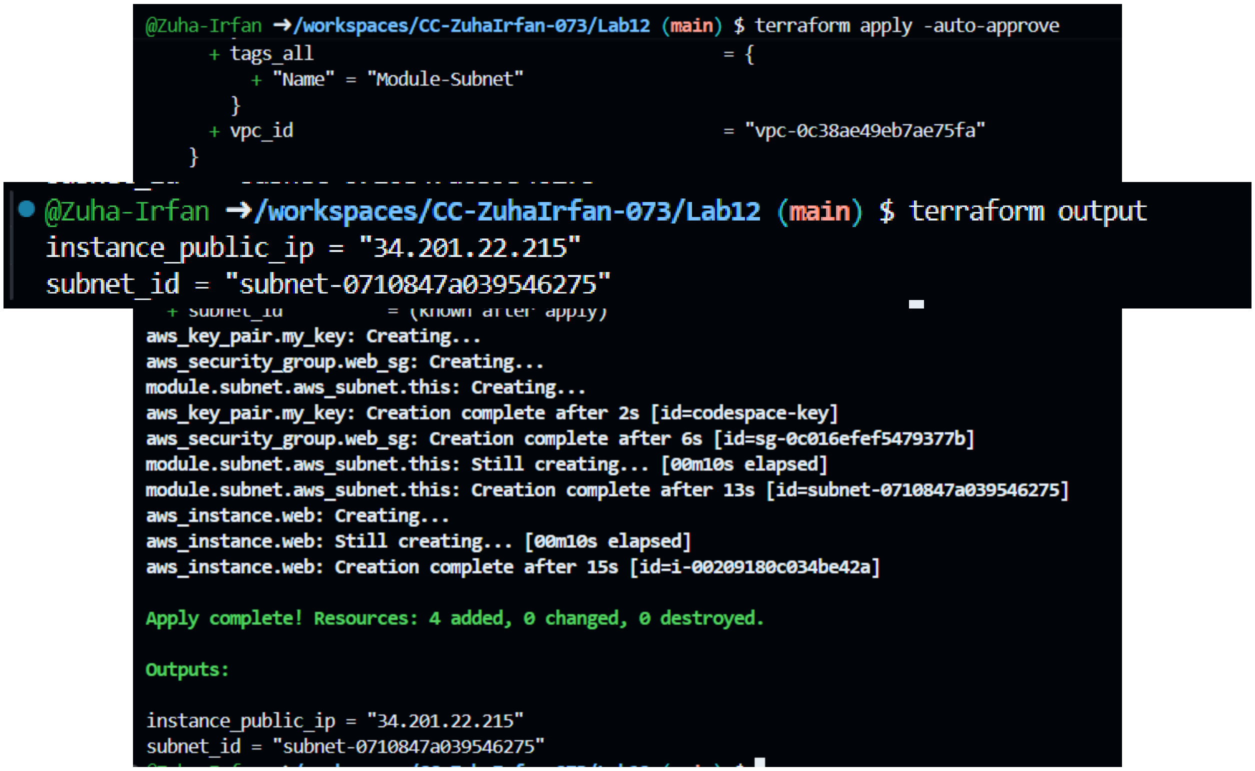
```
  GNU nano 7.2                                                         main.tf
provider "aws" {
  region = "us-east-1"
}

data "http" "my_ip" {
  url = "https://icanhazip.com"
}

data "aws_vpc" "default" {
  default = true
}

module "subnet" {
  source            = "./modules/subnet"
  vpc_id            = data.aws_vpc.default.id
  subnet_cidr       = "10.0.1.0/24"
  availability_zone = "us-east-1a"
}

resource "aws_key_pair" "my_key" {
  key_name   = "codespace-key"
  public_key = file("~/.ssh/id_ed25519.pub")
}

resource "aws_security_group" "web_sg" {
  name = "allow_ssh_http"
```

- task4_terraform_init.png



```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano modules/subnet/variables.tf
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano modules/subnet/main.tf
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano modules/subnet/outputs.tf
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano main.tf
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ nano outputs.tf
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform init
Initializing the backend...
Initializing modules...
- subnet in modules/subnet
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/http from the dependency lock file
- Using previously-installed hashicorp/aws v6.28.0
- Using previously-installed hashicorp/http v3.5.0

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.
```

- task4_terraform_apply.png



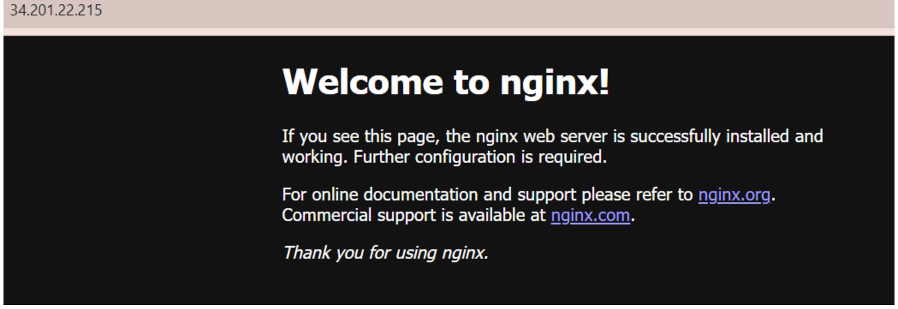
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform apply -auto-approve
      + tags_all                                    = {
          + "Name" = "Module-Subnet"
        }
      + vpc_id                                      = "vpc-0c38ae49eb7ae75fa"
    }
```

```
● @Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform output
  instance_public_ip = "34.201.22.215"
  subnet_id = "subnet-0710847a039546275"
```

```
  + subnet_id          = (known after apply)
aws_key_pair.my_key: Creating...
aws_security_group.web_sg: Creating...
module.subnet.aws_subnet.this: Creating...
aws_key_pair.my_key: Creation complete after 2s [id=codespace-key]
aws_security_group.web_sg: Creation complete after 6s [id=sg-0c016efef5479377b]
module.subnet.aws_subnet.this: Still creating... [00m10s elapsed]
module.subnet.aws_subnet.this: Creation complete after 13s [id=subnet-0710847a039546275]
aws_instance.web: Creating...
aws_instance.web: Still creating... [00m10s elapsed]
aws_instance.web: Creation complete after 15s [id=i-00209180c034be42a]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

Outputs:

instance_public_ip = "34.201.22.215"
subnet_id = "subnet-0710847a039546275"
```

- task4_terraform_output.png

- task4_nginx_browser.png



34.201.22.215

Welcome to nginx!

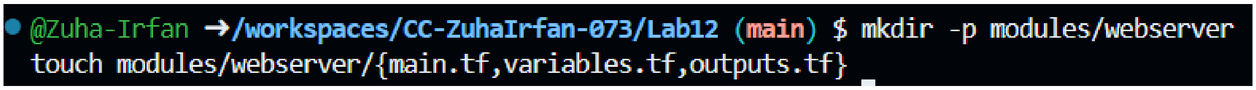
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

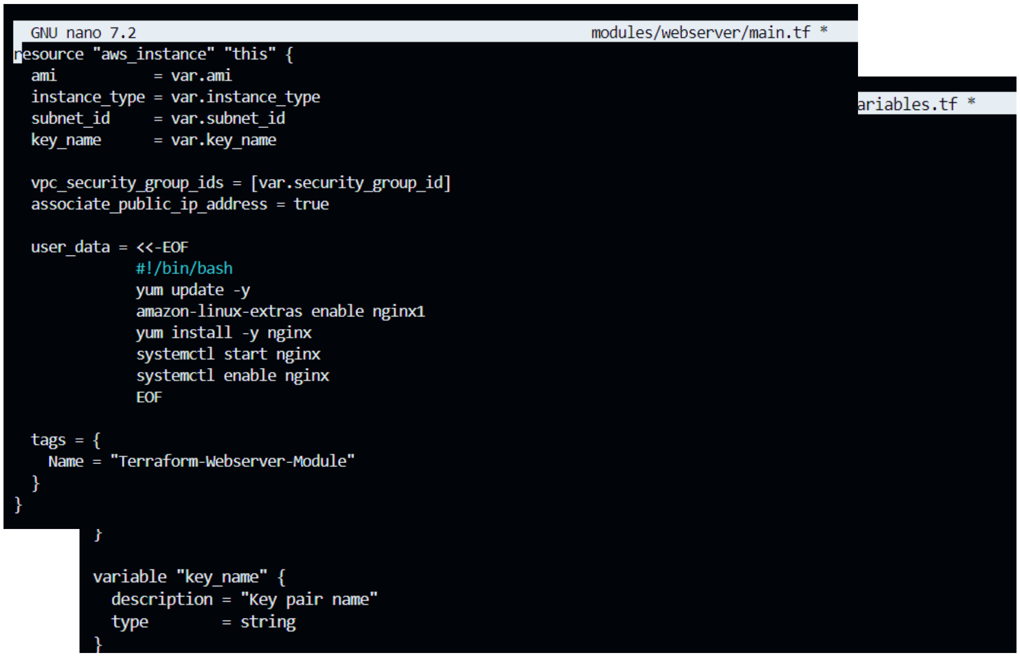
Task 5 — Create webserver module

- task5_webserver_module_structure.png



```
● @Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ mkdir -p modules/webserver
  touch modules/webserver/{main.tf,variables.tf,outputs.tf}
```

- task5_webserver_variables.png



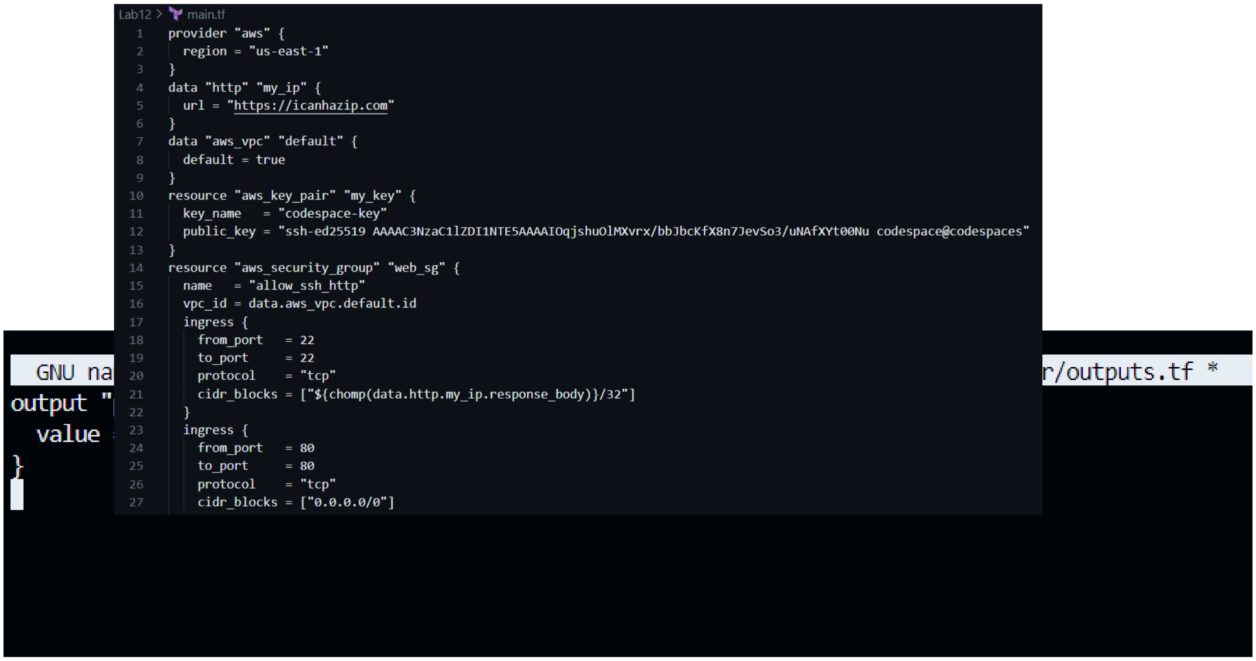
```
  GNU nano 7.2                                              modules/webserver/main.tf *
resource "aws_instance" "this" {
  ami           = var.ami
  instance_type = var.instance_type
  subnet_id     = var.subnet_id
  key_name      = var.key_name

  vpc_security_group_ids = [var.security_group_id]
  associate_public_ip_address = true

  user_data = <<-EOF
              #!/bin/bash
              yum update -y
              amazon-linux-extras enable nginx1
              yum install -y nginx
              systemctl start nginx
              systemctl enable nginx
              EOF

  tags = {
    Name = "Terraform-Webserver-Module"
  }
}
```

```
ariables.tf *
}

variable "key_name" {
  description = "Key pair name"
  type        = string
}
```

- task5_webserver_main.png

- task5_webserver_outputs.png

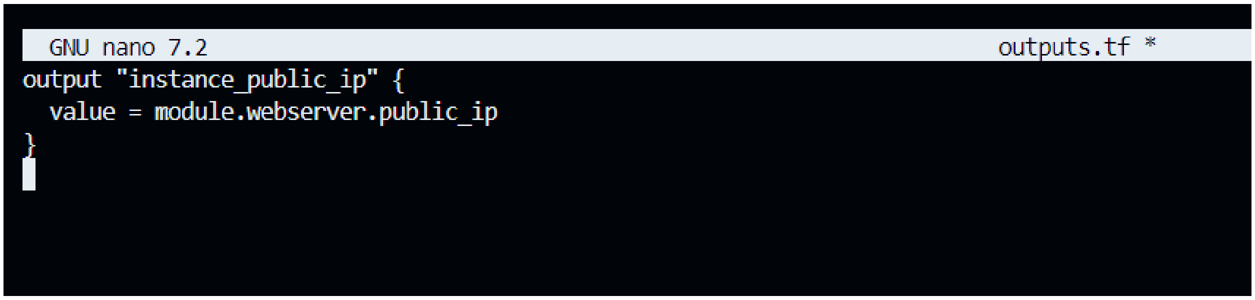
- task5_main_tf_webserver_module.png



```
Lab12 > main.tf
provider "aws" {
  region = "us-east-1"
}
data "http" "my_ip" {
  url = "https://icanhazip.com"
}
data "aws_vpc" "default" {
  default = true
}
resource "aws_key_pair" "my_key" {
  key_name   = "codespace-key"
  public_key = "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOqjshuOlMXvrx/bbJbcKfX8n7JevSo3/uNAfXYt00Nu codespace@codespaces"
}
resource "aws_security_group" "web_sg" {
  name   = "allow_ssh_http"
  vpc_id = data.aws_vpc.default.id
  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["${chomp(data.http.my_ip.response_body)}/32"]
  }
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
```

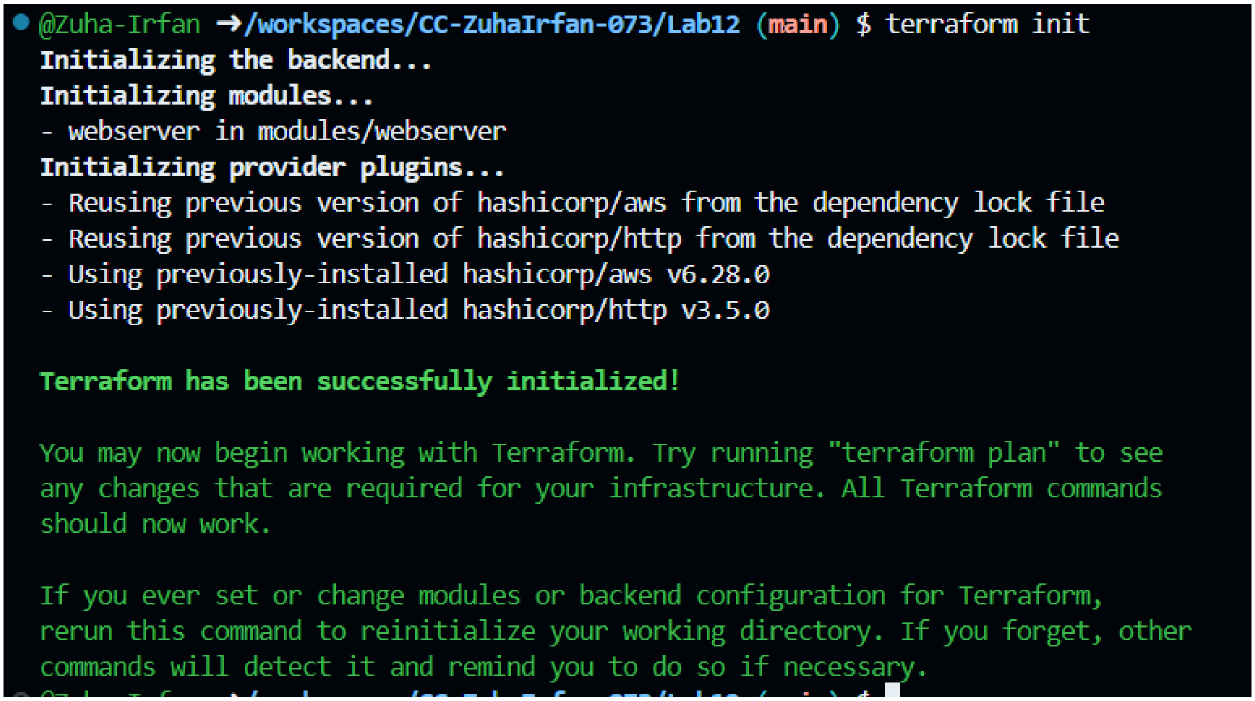
```
 GNU na                                   r/outputs.tf *
output "
  value
}
```

- task5_outputs_updated.png



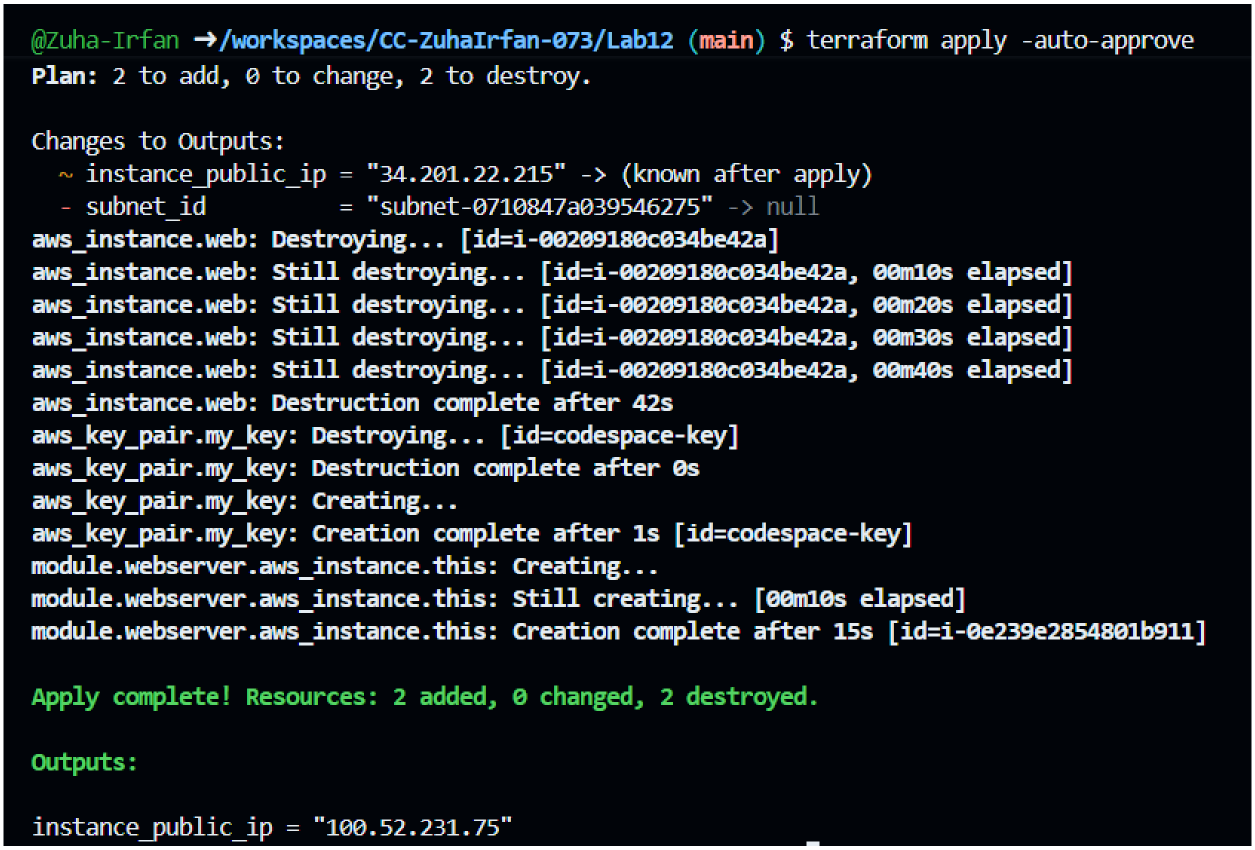
```
  GNU nano 7.2                                                          outputs.tf *
output "instance_public_ip" {
  value = module.webserver.public_ip
}
```

- task5_terraform_init.png



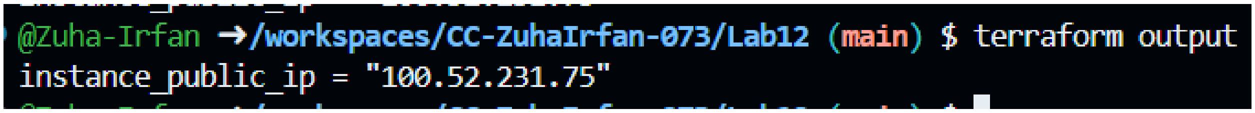
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform init
Initializing the backend...
Initializing modules...
- webserver in modules/webserver
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/http from the dependency lock file
- Using previously-installed hashicorp/aws v6.28.0
- Using previously-installed hashicorp/http v3.5.0

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.
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $
```

- task5_terraform_apply.png



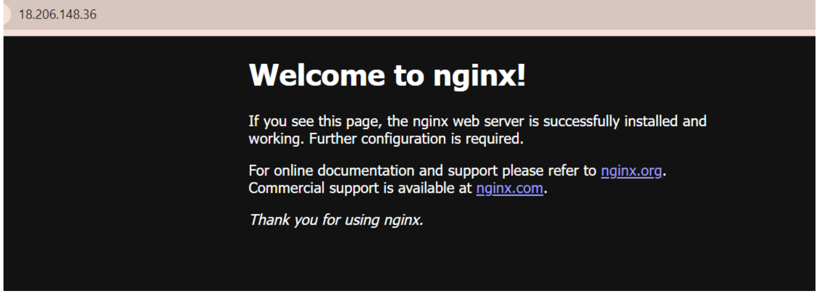
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform apply -auto-approve
Plan: 2 to add, 0 to change, 2 to destroy.

Changes to Outputs:
  ~ instance_public_ip = "34.201.22.215" -> (known after apply)
  - subnet_id          = "subnet-0710847a039546275" -> null
aws_instance.web: Destroying... [id=i-00209180c034be42a]
aws_instance.web: Still destroying... [id=i-00209180c034be42a, 00m10s elapsed]
aws_instance.web: Still destroying... [id=i-00209180c034be42a, 00m20s elapsed]
aws_instance.web: Still destroying... [id=i-00209180c034be42a, 00m30s elapsed]
aws_instance.web: Still destroying... [id=i-00209180c034be42a, 00m40s elapsed]
aws_instance.web: Destruction complete after 42s
aws_key_pair.my_key: Destroying... [id=codespace-key]
aws_key_pair.my_key: Destruction complete after 0s
aws_key_pair.my_key: Creating...
aws_key_pair.my_key: Creation complete after 1s [id=codespace-key]
module.webserver.aws_instance.this: Creating...
module.webserver.aws_instance.this: Still creating... [00m10s elapsed]
module.webserver.aws_instance.this: Creation complete after 15s [id=i-0e239e2854801b911]

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.

Outputs:

instance_public_ip = "100.52.231.75"
```

- task5_terraform_output.png



```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform output
instance_public_ip = "100.52.231.75"
```

- task5_nginx_browser.png



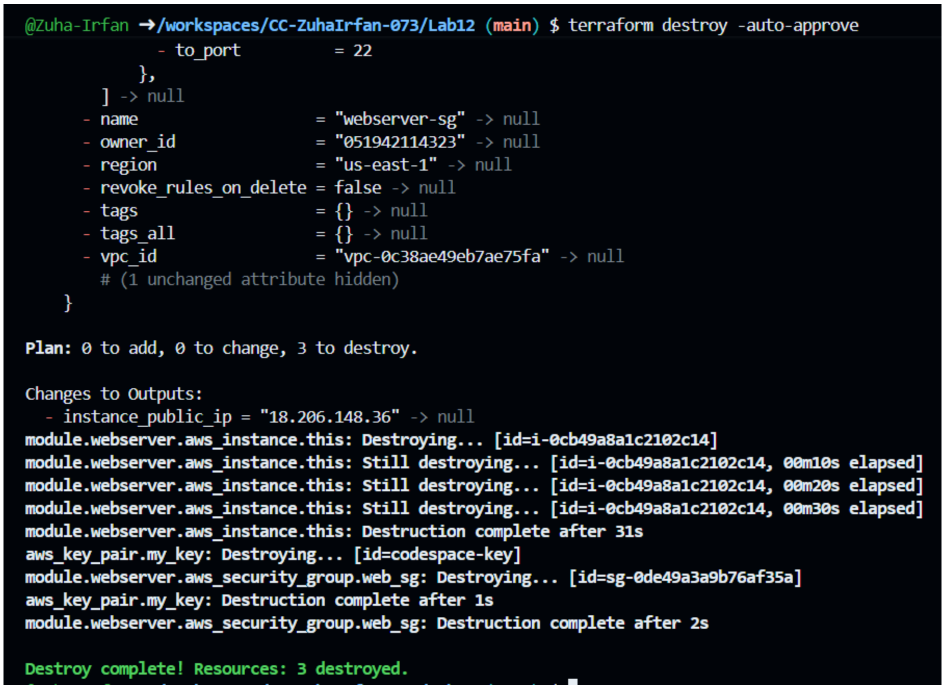
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

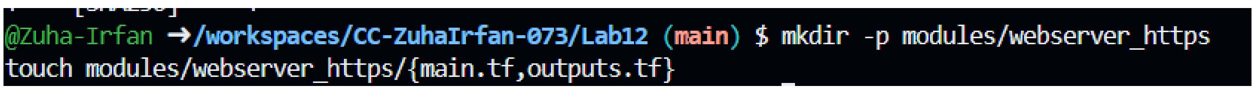
- task5_terraform_destroy.png



```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform destroy -auto-approve
              - to_port          = 22
            },
        ] -> null
      - name                   = "webserver-sg" -> null
      - owner_id               = "051942114323" -> null
      - region                 = "us-east-1" -> null
      - revoke_rules_on_delete = false -> null
      - tags                   = {} -> null
      - tags_all               = {} -> null
      - vpc_id                 = "vpc-0c38ae49eb7ae75fa" -> null
        # (1 unchanged attribute hidden)
    }

Plan: 0 to add, 0 to change, 3 to destroy.

Changes to Outputs:
  - instance_public_ip = "18.206.148.36" -> null
module.webserver.aws_instance.this: Destroying... [id=i-0cb49a8a1c2102c14]
module.webserver.aws_instance.this: Still destroying... [id=i-0cb49a8a1c2102c14, 00m10s elapsed]
module.webserver.aws_instance.this: Still destroying... [id=i-0cb49a8a1c2102c14, 00m20s elapsed]
module.webserver.aws_instance.this: Still destroying... [id=i-0cb49a8a1c2102c14, 00m30s elapsed]
module.webserver.aws_instance.this: Destruction complete after 31s
aws_key_pair.my_key: Destroying... [id=codespace-key]
module.webserver.aws_security_group.web_sg: Destroying... [id=sg-0de49a3a9b76af35a]
aws_key_pair.my_key: Destruction complete after 1s
module.webserver.aws_security_group.web_sg: Destruction complete after 2s

Destroy complete! Resources: 3 destroyed.
```

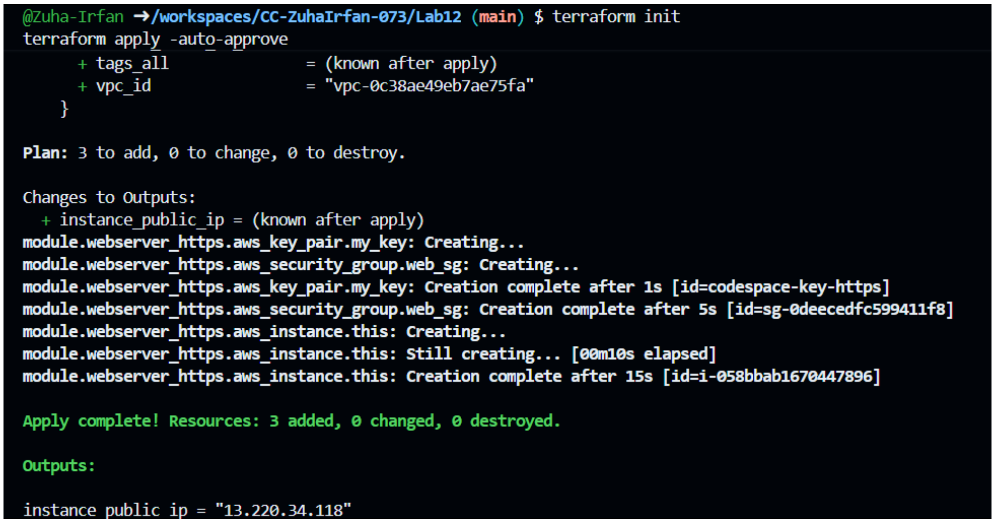
Task 6 — Configure HTTPS with self-signed certificates

- task6_entry_script_https.png



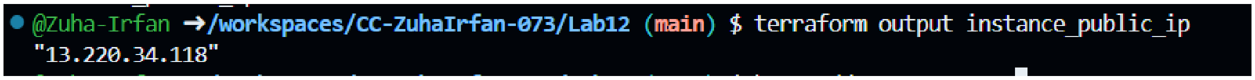
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ mkdir -p modules/webserver_https
touch modules/webserver_https/{main.tf,outputs.tf}
```

- task6_terraform_apply.png



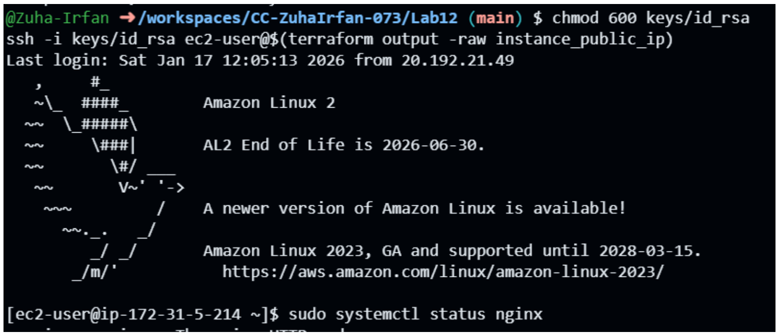
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform init
terraform apply -auto-approve
      + tags_all               = (known after apply)
      + vpc_id                 = "vpc-0c38ae49eb7ae75fa"
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + instance_public_ip = (known after apply)
module.webserver_https.aws_key_pair.my_key: Creating...
module.webserver_https.aws_security_group.web_sg: Creating...
module.webserver_https.aws_key_pair.my_key: Creation complete after 1s [id=codespace-key-https]
module.webserver_https.aws_security_group.web_sg: Creation complete after 5s [id=sg-0deecedfc599411f8]
module.webserver_https.aws_instance.this: Creating...
module.webserver_https.aws_instance.this: Still creating... [00m10s elapsed]
module.webserver_https.aws_instance.this: Creation complete after 15s [id=i-058bbab1670447896]

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

Outputs:

instance_public_ip = "13.220.34.118"
```

- task6_terraform_output.png



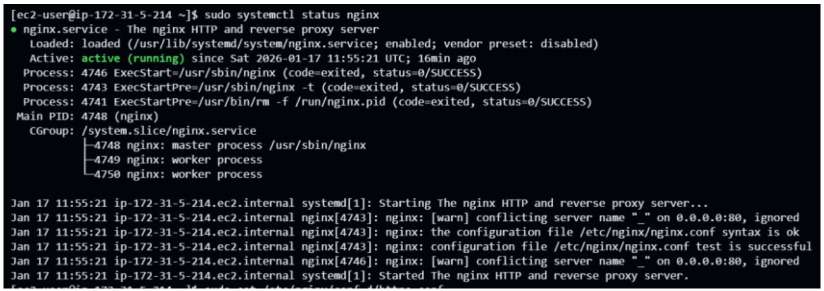
```
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform output instance_public_ip
"13.220.34.118"
```

- task6_browser_security_warning.png

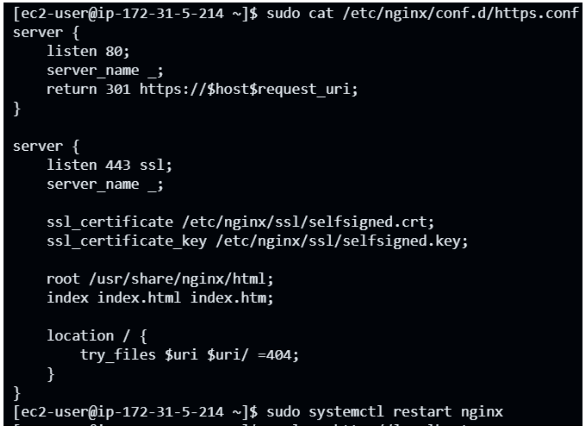


```
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ chmod 600 keys/id_rsa
ssh -i keys/id_rsa ec2-user@$(terraform output -raw instance_public_ip)
Last login: Sat Jan 17 12:05:13 2026 from 20.192.21.49
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-5-214 ~]$ sudo systemctl status nginx
```

- task6_nginx_https_browser.png

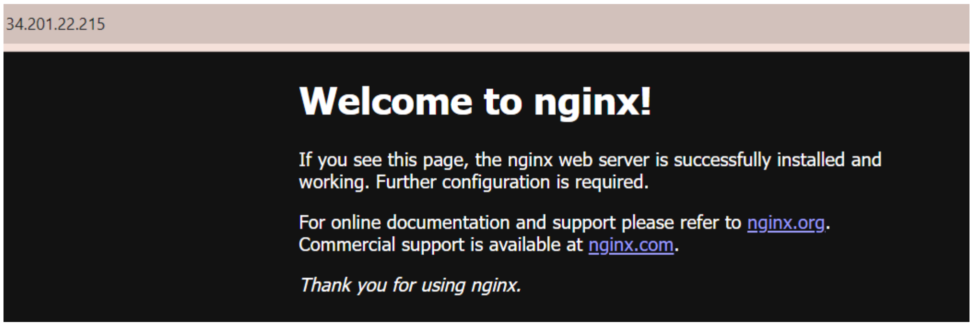


```
[ec2-user@ip-172-31-5-214 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2026-01-17 11:55:21 UTC; 16min ago
  Process: 4746 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
  Process: 4743 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
  Process: 4741 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
 Main PID: 4748 (nginx)
   CGroup: /system.slice/nginx.service
           ├─4748 nginx: master process /usr/sbin/nginx
           ├─4749 nginx: worker process
           └─4750 nginx: worker process

Jan 17 11:55:21 ip-172-31-5-214.ec2.internal systemd[1]: Starting The nginx HTTP and reverse proxy server...
Jan 17 11:55:21 ip-172-31-5-214.ec2.internal nginx[4743]: nginx: [warn] conflicting server name "_" on 0.0.0.0:80, ignored
Jan 17 11:55:21 ip-172-31-5-214.ec2.internal nginx[4743]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jan 17 11:55:21 ip-172-31-5-214.ec2.internal nginx[4743]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jan 17 11:55:21 ip-172-31-5-214.ec2.internal nginx[4746]: nginx: [warn] conflicting server name "_" on 0.0.0.0:80, ignored
Jan 17 11:55:21 ip-172-31-5-214.ec2.internal systemd[1]: Started The nginx HTTP and reverse proxy server.
```



```
[ec2-user@ip-172-31-5-214 ~]$ sudo cat /etc/nginx/conf.d/https.conf
server {
    listen 80;
    server_name _;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name _;

    ssl_certificate /etc/nginx/ssl/selfsigned.crt;
    ssl_certificate_key /etc/nginx/ssl/selfsigned.key;

    root /usr/share/nginx/html;
    index index.html index.htm;

    location / {
        try_files $uri $uri/ =404;
    }
}
[ec2-user@ip-172-31-5-214 ~]$ sudo systemctl restart nginx
```

- task6_http_redirect.png



34.201.22.215

Welcome to nginx!

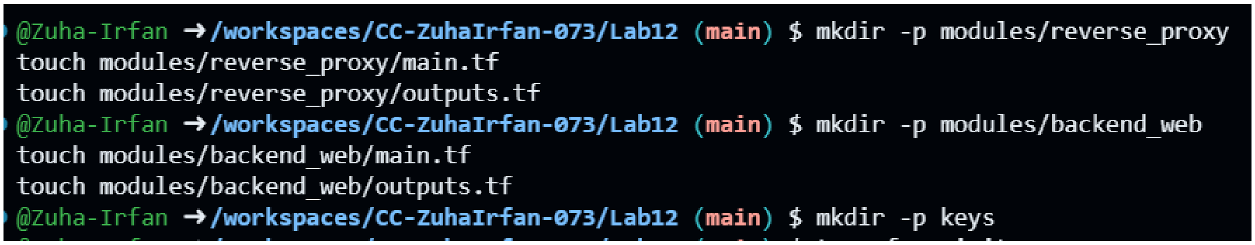
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

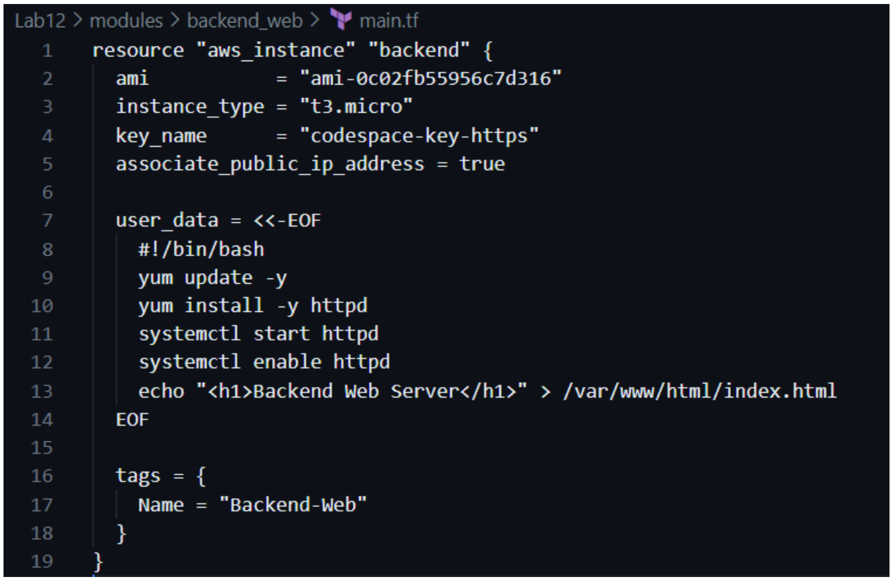
Task 7 — Configure Nginx as reverse proxy

- task7_apache_script.png



```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ mkdir -p modules/reverse_proxy
touch modules/reverse_proxy/main.tf
touch modules/reverse_proxy/outputs.tf
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ mkdir -p modules/backend_web
touch modules/backend_web/main.tf
touch modules/backend_web/outputs.tf
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ mkdir -p keys
```

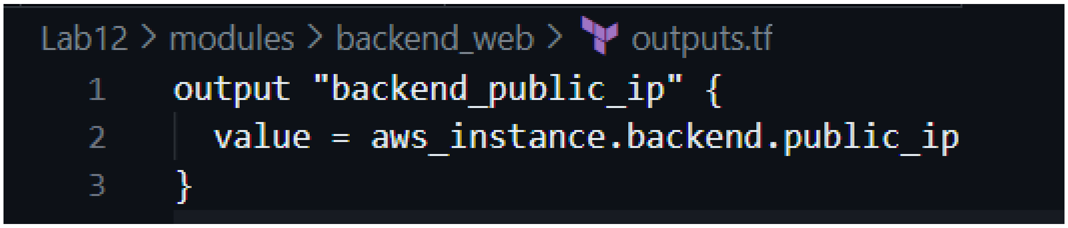
- task7_main_tf_web1.png



Lab12 > modules > backend_web > main.tf

```
resource "aws_instance" "backend" {
  ami           = "ami-0c02fb55956c7d316"
  instance_type = "t3.micro"
  key_name      = "codespace-key-https"
  associate_public_ip_address = true

  user_data = <<-EOF
    #!/bin/bash
    yum update -y
    yum install -y httpd
    systemctl start httpd
    systemctl enable httpd
    echo "<h1>Backend Web Server</h1>" > /var/www/html/index.html
  EOF

  tags = {
    Name = "Backend-Web"
  }
}
```

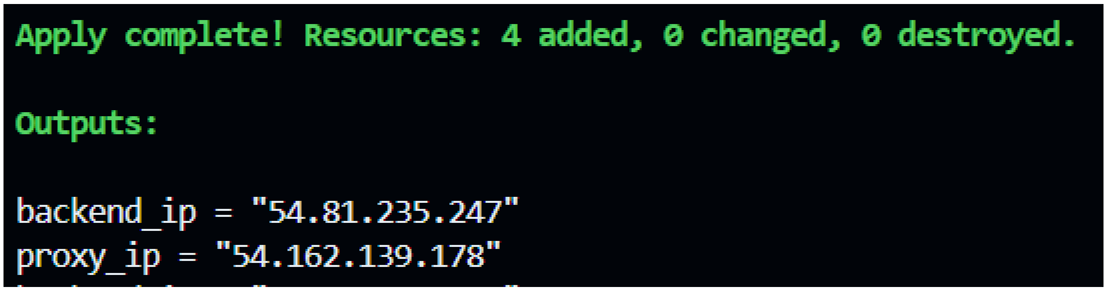
- task7_outputs_web1.png



Lab12 > modules > backend_web > outputs.tf

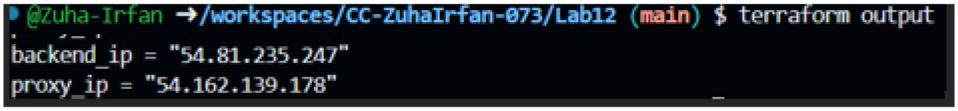
```
output "backend_public_ip" {
  value = aws_instance.backend.public_ip
}
```

- task7_terraform_apply.png



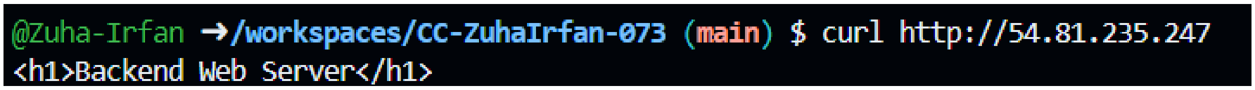
```
Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

Outputs:

backend_ip = "54.81.235.247"
proxy_ip = "54.162.139.178"
```

- task7_terraform_output.png



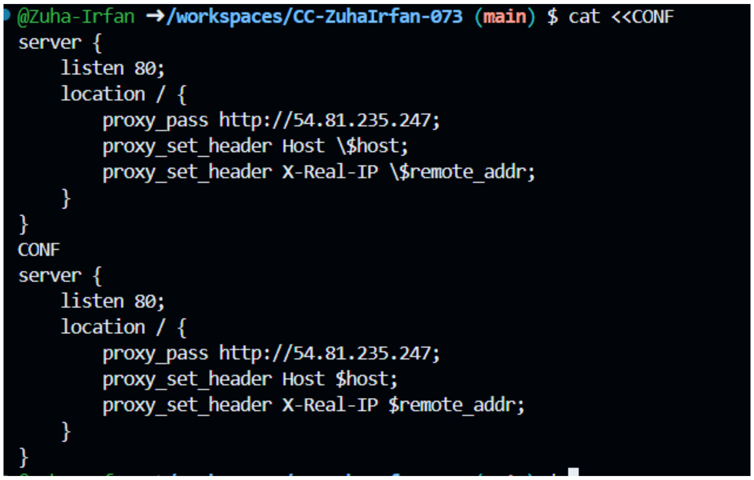
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform output
backend_ip = "54.81.235.247"
proxy_ip = "54.162.139.178"
```

- task7_ssh_webserver.png



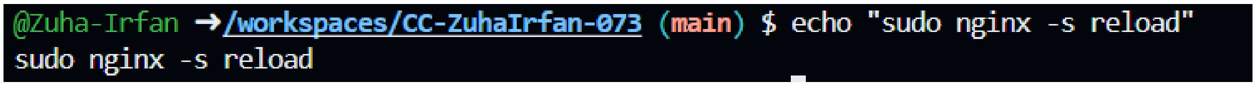
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ curl http://54.81.235.247
<h1>Backend Web Server</h1>
```

- task7_nginx_conf_reverse_proxy.png



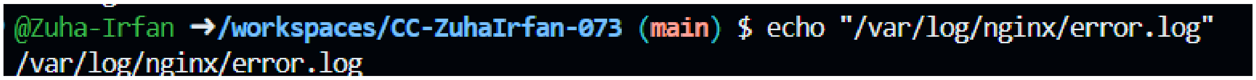
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ cat <<CONF
server {
    listen 80;
    location / {
        proxy_pass http://54.81.235.247;
        proxy_set_header Host \$host;
        proxy_set_header X-Real-IP \$remote_addr;
    }
}
CONF
server {
    listen 80;
    location / {
        proxy_pass http://54.81.235.247;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

- task7_nginx_restart.png



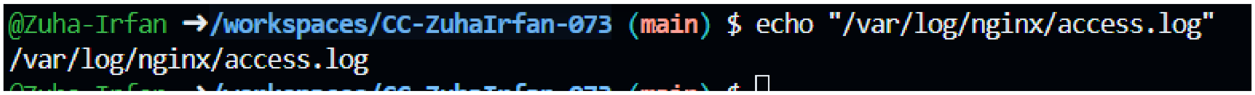
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $ echo "sudo nginx -s reload"
sudo nginx -s reload
```

- task7_error_log.png



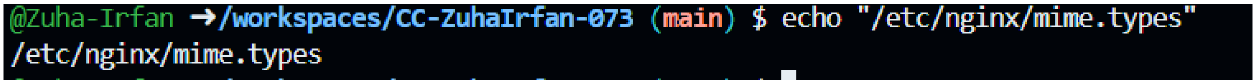
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $ echo "/var/log/nginx/error.log"
/var/log/nginx/error.log
```

- task7_access_log.png



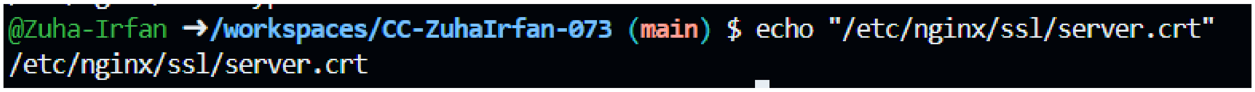
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $ echo "/var/log/nginx/access.log"
/var/log/nginx/access.log
```

- task7_mime_types.png



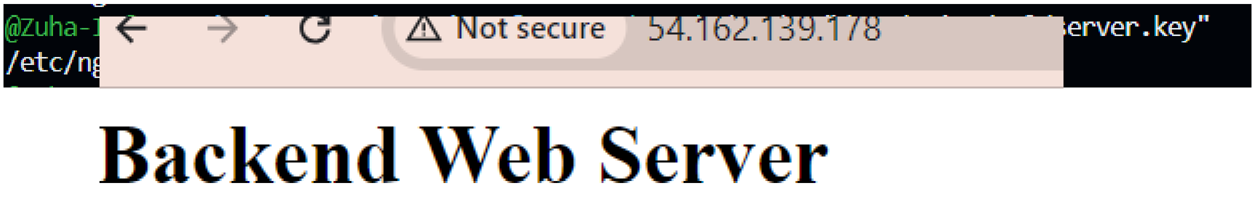
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $ echo "/etc/nginx/mime.types"
/etc/nginx/mime.types
```

- task7_ssl_cert.png



```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073 (main) $ echo "/etc/nginx/ssl/server.crt"
/etc/nginx/ssl/server.crt
```

- task7_ssl_key.png
- task7_reverse_proxy_browser.png

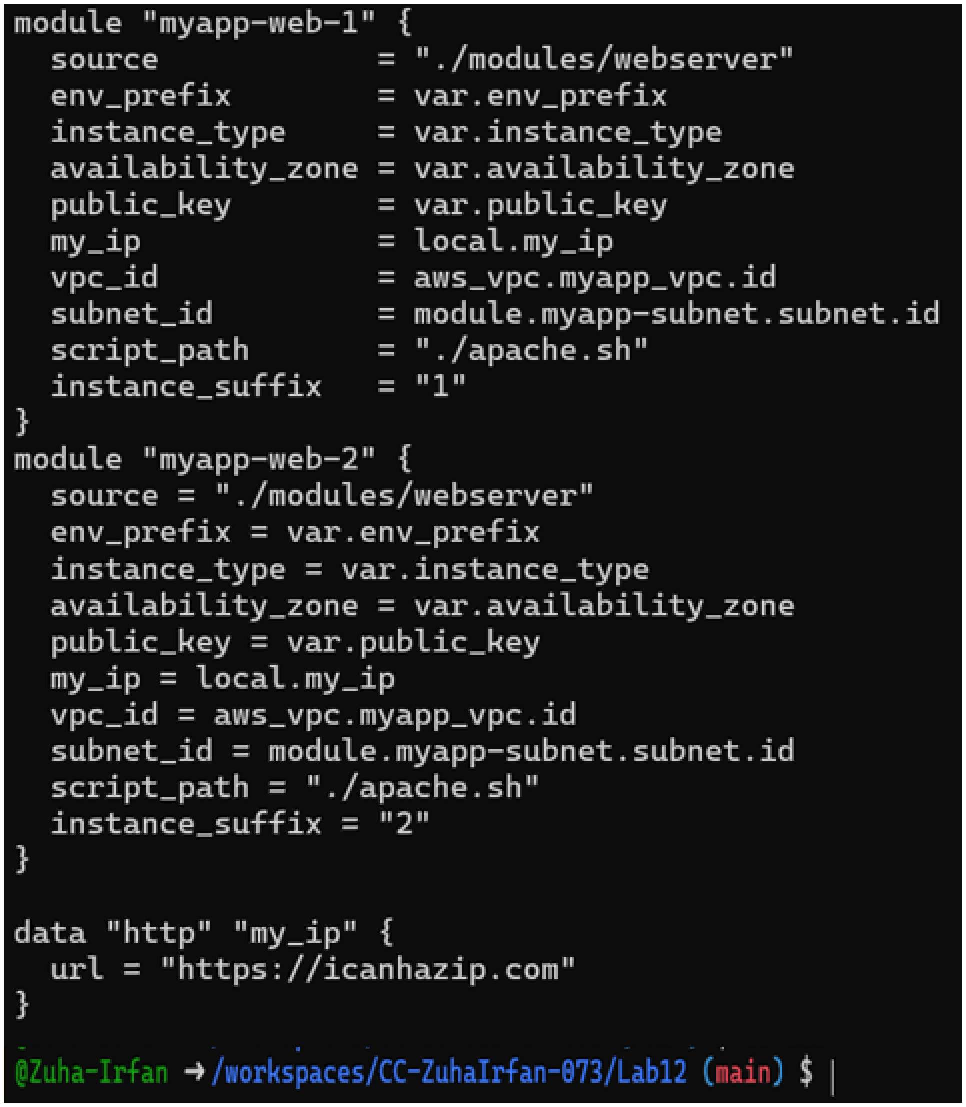


```
@Zuha-I                                              erver.key"
/etc/ng
```

Backend Web Server

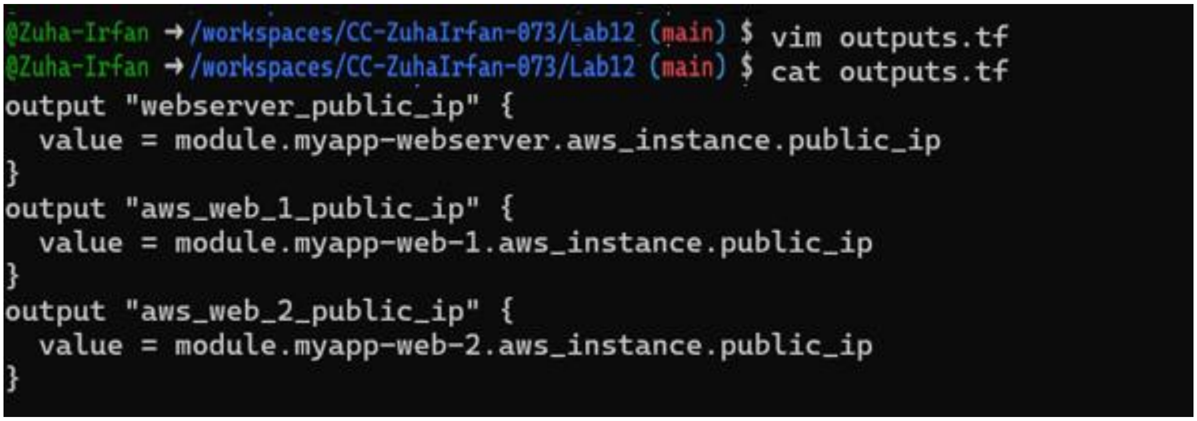
Task 8 — Configure Nginx as load balancer

- task8_main_tf_web2.png



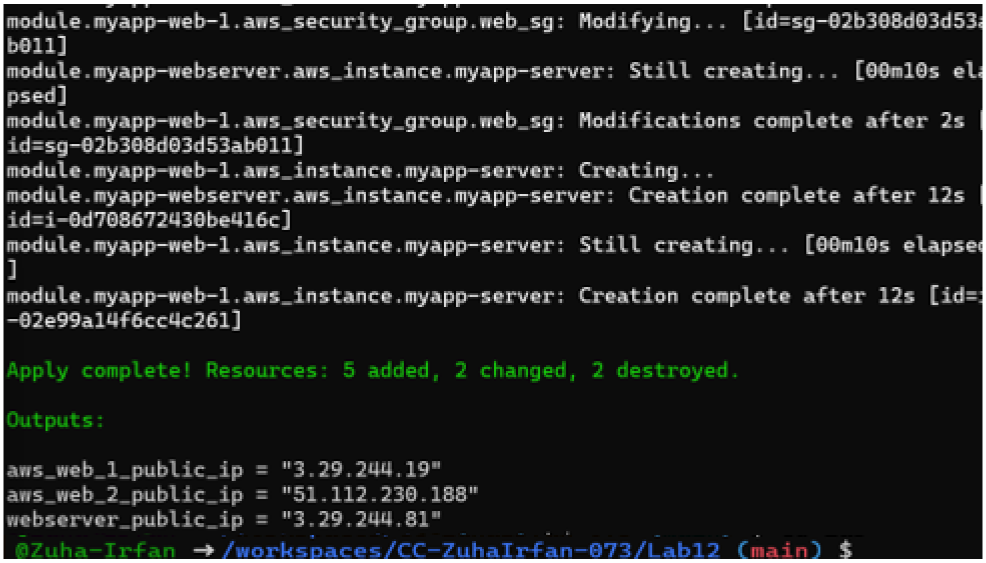
```
module "myapp-web-1" {
  source            = "./modules/webserver"
  env_prefix        = var.env_prefix
  instance_type     = var.instance_type
  availability_zone = var.availability_zone
  public_key        = var.public_key
  my_ip             = local.my_ip
  vpc_id            = aws_vpc.myapp_vpc.id
  subnet_id         = module.myapp-subnet.subnet.id
  script_path       = "./apache.sh"
  instance_suffix   = "1"
}
module "myapp-web-2" {
  source = "./modules/webserver"
  env_prefix = var.env_prefix
  instance_type = var.instance_type
  availability_zone = var.availability_zone
  public_key = var.public_key
  my_ip = local.my_ip
  vpc_id = aws_vpc.myapp_vpc.id
  subnet_id = module.myapp-subnet.subnet.id
  script_path = "./apache.sh"
  instance_suffix = "2"
}

data "http" "my_ip" {
  url = "https://icanhazip.com"
}

@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ |
```

- task8_outputs_web2.png



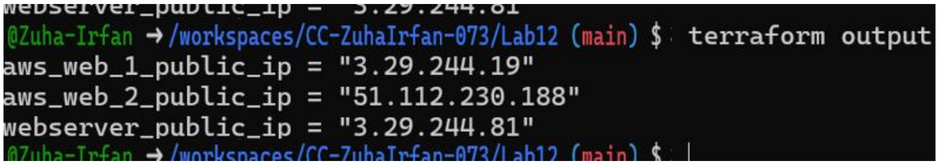
```
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ vim outputs.tf
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ cat outputs.tf
output "webserver_public_ip" {
  value = module.myapp-webserver.aws_instance.public_ip
}
output "aws_web_1_public_ip" {
  value = module.myapp-web-1.aws_instance.public_ip
}
output "aws_web_2_public_ip" {
  value = module.myapp-web-2.aws_instance.public_ip
}
```

- task8_terraform_apply.png



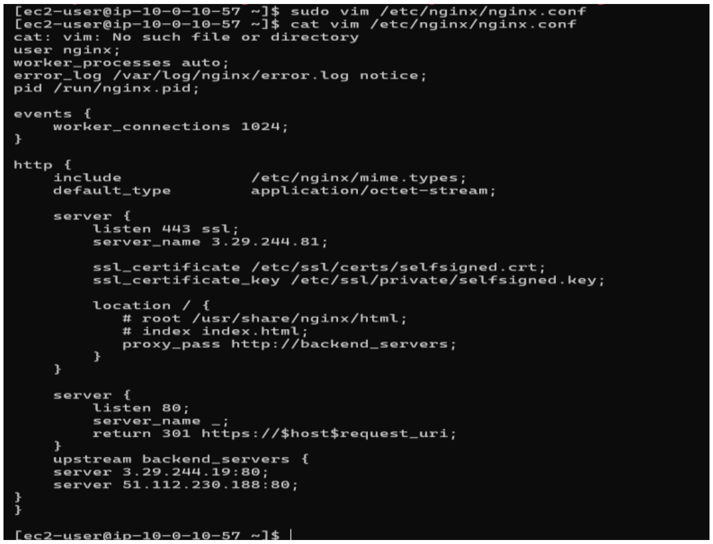
```
module.myapp-web-1.aws_security_group.web_sg: Modifying... [id=sg-02b308d03d53a
b011]
module.myapp-webserver.aws_instance.myapp-server: Still creating... [00m10s ela
psed]
module.myapp-web-1.aws_security_group.web_sg: Modifications complete after 2s [
id=sg-02b308d03d53ab011]
module.myapp-web-1.aws_instance.myapp-server: Creating...
module.myapp-webserver.aws_instance.myapp-server: Creation complete after 12s [
id=i-0d708672430be416c]
module.myapp-web-1.aws_instance.myapp-server: Still creating... [00m10s elapsed
]
module.myapp-web-1.aws_instance.myapp-server: Creation complete after 12s [id=i
-02e99a14f6cc4c261]

Apply complete! Resources: 5 added, 2 changed, 2 destroyed.

Outputs:

aws_web_1_public_ip = "3.29.244.19"
aws_web_2_public_ip = "51.112.230.188"
webserver_public_ip = "3.29.244.81"
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $
```

- task8_terraform_output.png



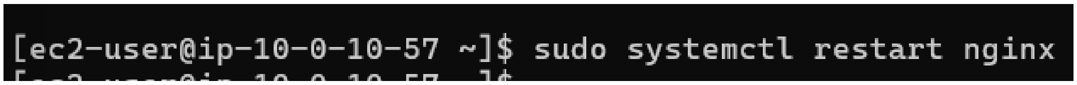
```
webserver_public_ip = "3.29.244.81"
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ terraform output
aws_web_1_public_ip = "3.29.244.19"
aws_web_2_public_ip = "51.112.230.188"
webserver_public_ip = "3.29.244.81"
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ |
```

- task8_nginx_conf_load_balancer.png



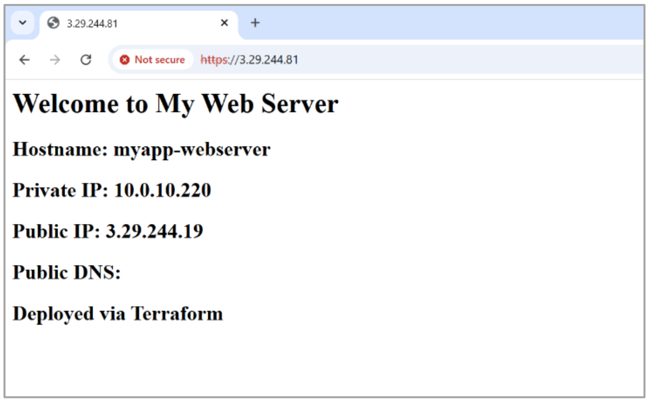
```
[ec2-user@ip-10-0-10-57 ~]$ sudo vim /etc/nginx/nginx.conf
[ec2-user@ip-10-0-10-57 ~]$ cat vim /etc/nginx/nginx.conf
cat: vim: No such file or directory
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    server {
        listen 443 ssl;
        server_name 3.29.244.81;

        ssl_certificate /etc/ssl/certs/selfsigned.crt;
        ssl_certificate_key /etc/ssl/private/selfsigned.key;

        location / {
           # root /usr/share/nginx/html;
           # index index.html;
           proxy_pass http://backend_servers;
        }
    }

    server {
        listen 80;
        server_name _;
        return 301 https://$host$request_uri;
    }
    upstream backend_servers {
    server 3.29.244.19:80;
    server 51.112.230.188:80;
}
}

[ec2-user@ip-10-0-10-57 ~]$ |
```

- task8_nginx_restart.png



```
[ec2-user@ip-10-0-10-57 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-57 ~]$
```

- task8_load_balancer_web1.png



Welcome to My Web Server

Hostname: myapp-webserver

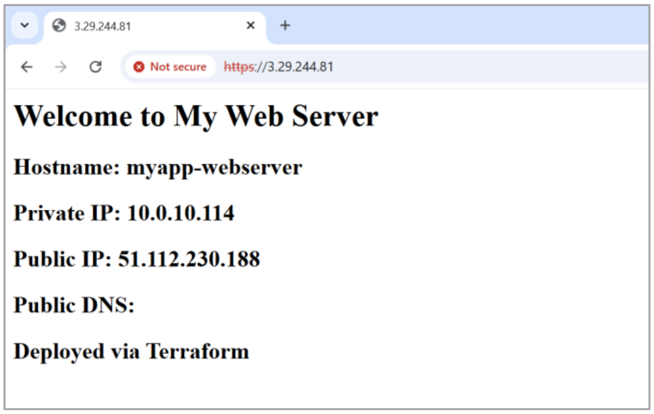
Private IP: 10.0.10.220

Public IP: 3.29.244.19

Public DNS:

Deployed via Terraform

- task8_load_balancer_web2.png



Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.0.10.114

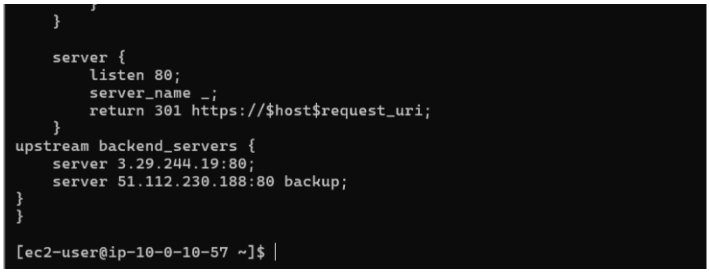
Public IP: 51.112.230.188

Public DNS:

Deployed via Terraform

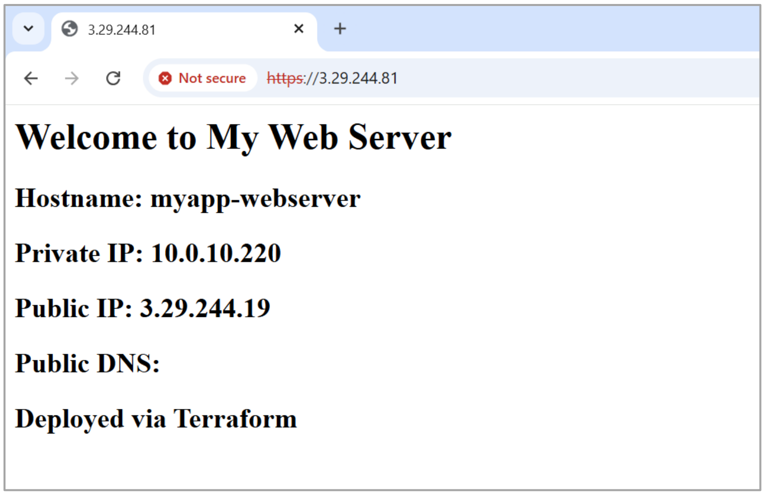
Task 9 — Configure high availability with backup servers

- task9_nginx_conf_ha_web1_primary.png



```
    }

    server {
        listen 80;
        server_name _;
        return 301 https://$host$request_uri;
    }
upstream backend_servers {
    server 3.29.244.19:80;
    server 51.112.230.188:80 backup;
}
}

[ec2-user@ip-10-0-10-57 ~]$ |
```

- task9_ha_web1_only.png



Welcome to My Web Server

Hostname: myapp-webserver

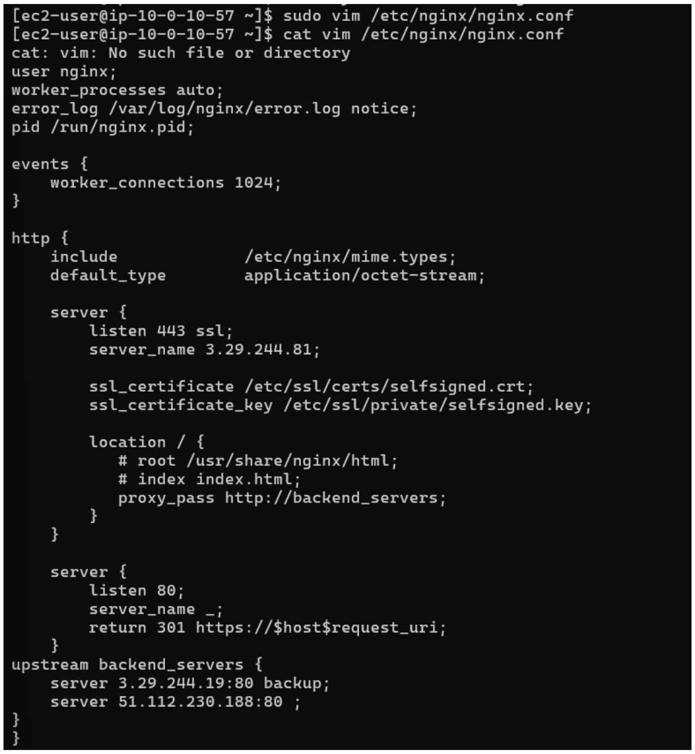
Private IP: 10.0.10.220

Public IP: 3.29.244.19

Public DNS:

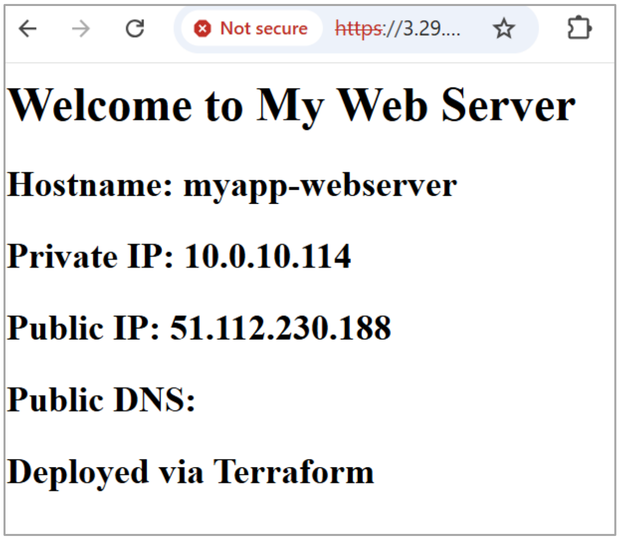
Deployed via Terraform

- task9_nginx_conf_ha_web2_primary.png



```
[ec2-user@ip-10-0-10-57 ~]$ sudo vim /etc/nginx/nginx.conf
[ec2-user@ip-10-0-10-57 ~]$ cat vim /etc/nginx/nginx.conf
cat: vim: No such file or directory
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    server {
        listen 443 ssl;
        server_name 3.29.244.81;

        ssl_certificate /etc/ssl/certs/selfsigned.crt;
        ssl_certificate_key /etc/ssl/private/selfsigned.key;

        location / {
           # root /usr/share/nginx/html;
           # index index.html;
           proxy_pass http://backend_servers;
        }
    }

    server {
        listen 80;
        server_name _;
        return 301 https://$host$request_uri;
    }
upstream backend_servers {
    server 3.29.244.19:80 backup;
    server 51.112.230.188:80 ;
}
}
```

- task9_ha_web2_only.png



Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.0.10.114

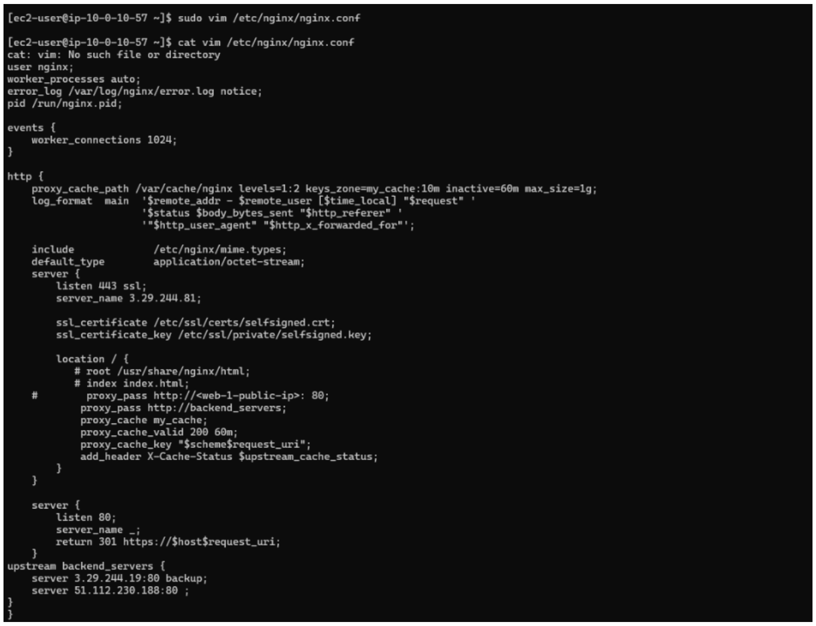
Public IP: 51.112.230.188

Public DNS:

Deployed via Terraform

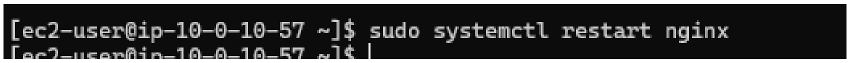
Task 10 — Enable Nginx caching

- task10_nginx_conf_cache.png



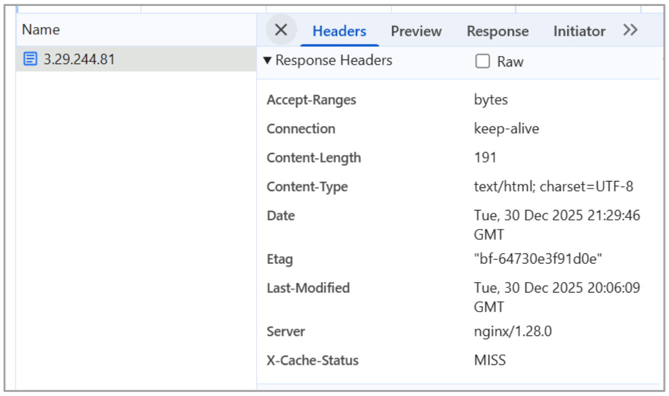
```
[ec2-user@ip-10-0-10-57 ~]$ sudo vim /etc/nginx/nginx.conf

[ec2-user@ip-10-0-10-57 ~]$ cat vim /etc/nginx/nginx.conf
cat: vim: No such file or directory
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=my_cache:10m inactive=60m max_size=1g;
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;
    server {
        listen 443 ssl;
        server_name 3.29.244.81;

        ssl_certificate /etc/ssl/certs/selfsigned.crt;
        ssl_certificate_key /etc/ssl/private/selfsigned.key;

        location / {
           # root /usr/share/nginx/html;
           # index index.html;
    #        proxy_pass http://<web-1-public-ip>: 80;
            proxy_pass http://backend_servers;
            proxy_cache my_cache;
            proxy_cache_valid 200 60m;
            proxy_cache_key "$scheme$request_uri";
            add_header X-Cache-Status $upstream_cache_status;
        }
    }

    server {
        listen 80;
        server_name _;
        return 301 https://$host$request_uri;
    }
upstream backend_servers {
    server 3.29.244.19:80 backup;
    server 51.112.230.188:80 ;
}
}
```

- task10_nginx_restart.png



```
[ec2-user@ip-10-0-10-57 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-57 ~]$ |
```

- task10_cache_miss.png

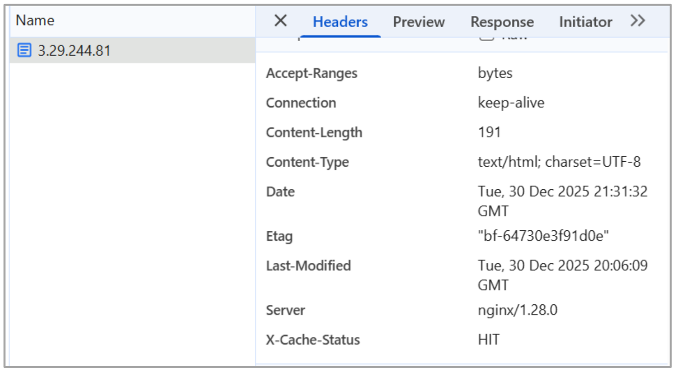


3.29.244.81

▾ Response Headers

Accept-Ranges	bytes
Connection	keep-alive
Content-Length	191
Content-Type	text/html; charset=UTF-8
Date	Tue, 30 Dec 2025 21:29:46 GMT
Etag	"bf-64730e3f91d0e"
Last-Modified	Tue, 30 Dec 2025 20:06:09 GMT
Server	nginx/1.28.0
X-Cache-Status	MISS

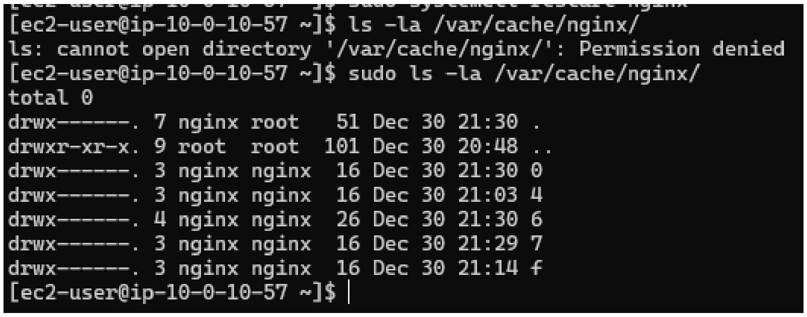
- task10_cache_hit.png



3.29.244.81

Accept-Ranges	bytes
Connection	keep-alive
Content-Length	191
Content-Type	text/html; charset=UTF-8
Date	Tue, 30 Dec 2025 21:31:32 GMT
Etag	"bf-64730e3f91d0e"
Last-Modified	Tue, 30 Dec 2025 20:06:09 GMT
Server	nginx/1.28.0
X-Cache-Status	HIT

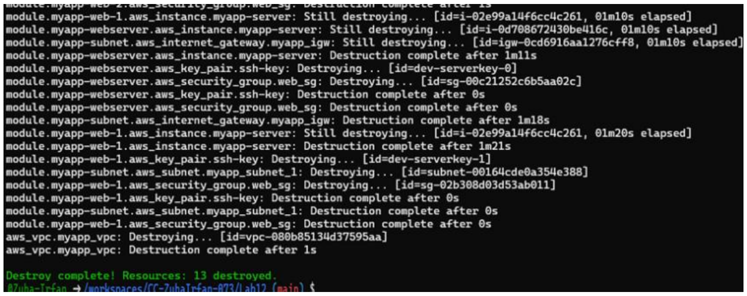
- task10_cache_directory.png



```
[ec2-user@ip-10-0-10-57 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-57 ~]$ ls -la /var/cache/nginx/
ls: cannot open directory '/var/cache/nginx/': Permission denied
[ec2-user@ip-10-0-10-57 ~]$ sudo ls -la /var/cache/nginx/
total 0
drwx------. 7 nginx root   51 Dec 30 21:30 .
drwxr-xr-x. 9 root  root  101 Dec 30 20:48 ..
drwx------. 3 nginx nginx  16 Dec 30 21:30 0
drwx------. 3 nginx nginx  16 Dec 30 21:03 4
drwx------. 4 nginx nginx  26 Dec 30 21:30 6
drwx------. 3 nginx nginx  16 Dec 30 21:29 7
drwx------. 3 nginx nginx  16 Dec 30 21:14 f
[ec2-user@ip-10-0-10-57 ~]$ |
```

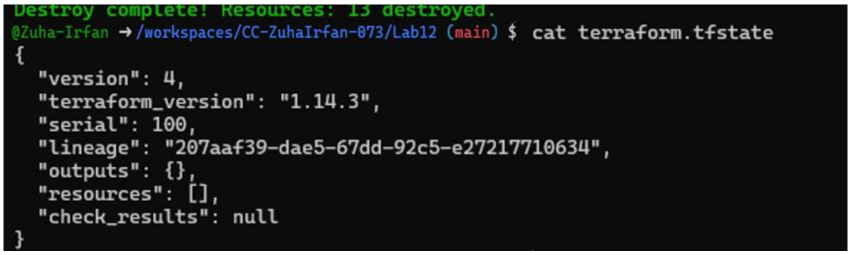
Cleanup

- cleanup_destroy_prompt.png



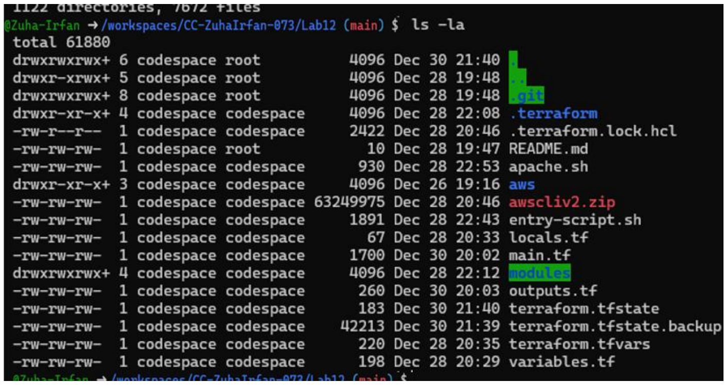
```
module.myapp-web-2.aws_security_group.web_sg: Destruction complete after 1s
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-02e99a14f6cc4c261, 01m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0d708672430be416c, 01m10s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0cd6916aa1276cff8, 01m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Destruction complete after 1m11s
module.myapp-webserver.aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-0]
module.myapp-webserver.aws_security_group.web_sg: Destroying... [id=sg-00c21252c6b5aa02c]
module.myapp-webserver.aws_key_pair.ssh-key: Destruction complete after 0s
module.myapp-webserver.aws_security_group.web_sg: Destruction complete after 0s
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destruction complete after 1m18s
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-02e99a14f6cc4c261, 01m20s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Destruction complete after 1m21s
module.myapp-web-1.aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-1]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destroying... [id=subnet-00164cde0a354e388]
module.myapp-web-1.aws_security_group.web_sg: Destroying... [id=sg-02b308d03d53ab011]
module.myapp-web-1.aws_key_pair.ssh-key: Destruction complete after 0s
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete after 0s
module.myapp-web-1.aws_security_group.web_sg: Destruction complete after 0s
aws_vpc.myapp_vpc: Destroying... [id=vpc-080b85134d37595aa]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 13 destroyed.
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $
```

- cleanup_destroy_complete.png



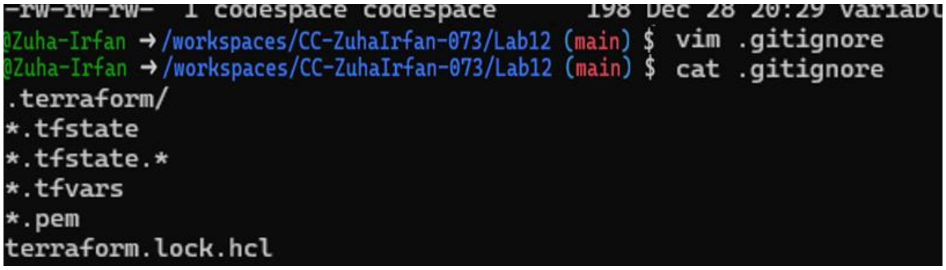
```
Destroy complete! Resources: 13 destroyed.
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 100,
  "lineage": "207aaf39-dae5-67dd-92c5-e27217710634",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

- cleanup_state_empty.png



```
1122 directories, 7672 files
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ ls -la
total 61880
drwxrwxrwx+ 6 codespace root          4096 Dec 30 21:40 .
drwxr-xrwx+ 5 codespace root          4096 Dec 28 19:48 ..
drwxrwxrwx+ 8 codespace root          4096 Dec 28 19:48 .git
drwxr-xr-x+ 4 codespace codespace     4096 Dec 28 22:08 .terraform
-rw-r--r--  1 codespace codespace     2422 Dec 28 20:46 .terraform.lock.hcl
-rw-rw-rw-  1 codespace root            10 Dec 28 19:47 README.md
-rw-rw-rw-  1 codespace codespace      930 Dec 28 22:53 apache.sh
drwxr-xr-x+ 3 codespace codespace     4096 Dec 26 19:16 aws
-rw-rw-rw-  1 codespace codespace 63249975 Dec 28 20:46 awscliv2.zip
-rw-rw-rw-  1 codespace codespace     1891 Dec 28 22:43 entry-script.sh
-rw-rw-rw-  1 codespace codespace       67 Dec 28 20:33 locals.tf
-rw-rw-rw-  1 codespace codespace     1700 Dec 30 20:02 main.tf
drwxrwxrwx+ 4 codespace codespace     4096 Dec 28 22:12 modules
-rw-rw-rw-  1 codespace codespace      260 Dec 30 20:03 outputs.tf
-rw-rw-rw-  1 codespace codespace      183 Dec 30 21:40 terraform.tfstate
-rw-rw-rw-  1 codespace codespace    42213 Dec 30 21:39 terraform.tfstate.backup
-rw-rw-rw-  1 codespace codespace      220 Dec 28 20:35 terraform.tfvars
-rw-rw-rw-  1 codespace codespace      198 Dec 28 20:29 variables.tf
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $
```

- cleanup_final_files.png



```
-rw-rw-rw-  1 codespace codespace      198 Dec 28 20:29 variabl
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ vim .gitignore
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/Lab12 (main) $ cat .gitignore
.terraform/
*.tfstate
*.tfstate.*
*.tfvars
*.pem
terraform.lock.hcl
```