# Name: Sadaf Riaz

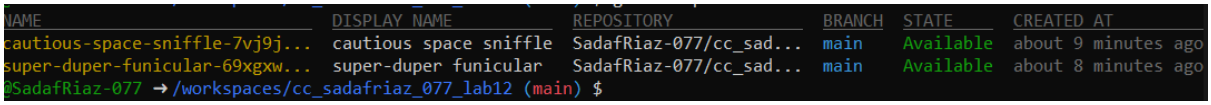
# Roll no: 2023-BSE-077

# Section: BSE5-B

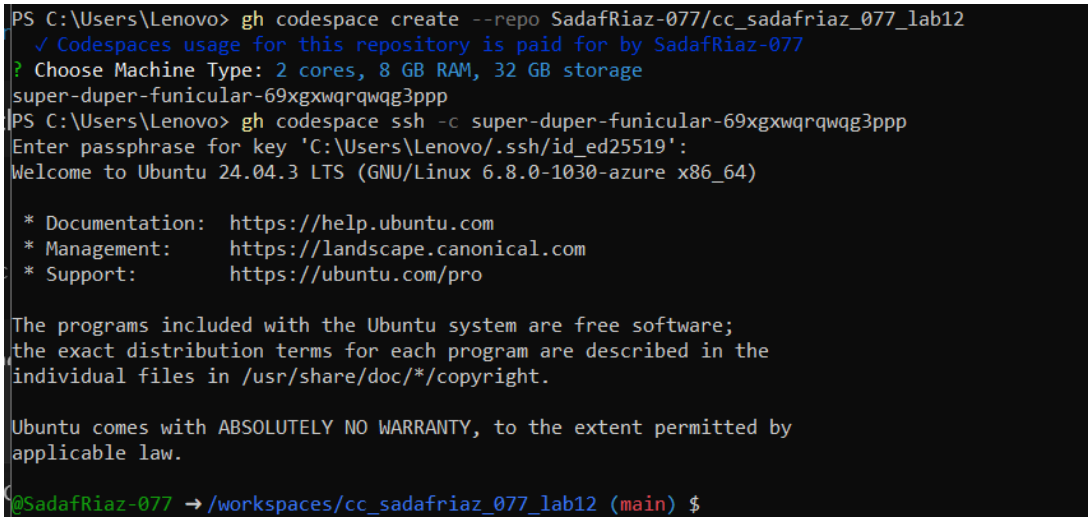
# Lab 12

## Task 0 – Lab Setup (Codespace & GH CLI)

### 0.1 Codespace Creation (task0_codespace_create_and_list.png)

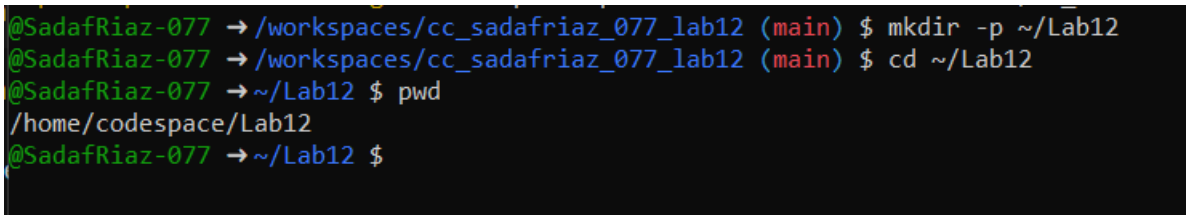



### 0.2 Codespace SSH Connection (task0_codespace_ssh_connected.png)

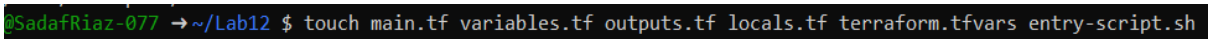



## Task 1 – Organize Terraform Code into Separate Files

### 1.1 Project Directory Structure (task1_dir_structure.png)




### 1.2 Creating Terraform Files (task1_terraform_files.png)




### 1.3 Defining Variables (task1_define_variables.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ cat 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" {}

@SadafRiaz-077 ➜ ~/Lab12 $
```

### 1.4 Defining Outputs (task1_define_outputs.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ nano outputs.tf
@SadafRiaz-077 ➜ ~/Lab12 $ cat outputs.tf
output "aws_instance_public_ip" {
  value = aws_instance.myapp-server.public_ip
}

@SadafRiaz-077 ➜ ~/Lab12 $
```

### 1.5 Using Local Values (task1_local_values.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ nano locals.tf
@SadafRiaz-077 ➜ ~/Lab12 $ cat locals.tf
locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"
}

@SadafRiaz-077 ➜ ~/Lab12 $
```

### 1.6 Terraform Variables File (task1_variables_file.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ nano terraform.tfvars
@SadafRiaz-077 ➜ ~/Lab12 $ cat 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"

@SadafRiaz-077 ➜ ~/Lab12 $
```

### 1.7 Main Terraform Configuration (task1_main_config.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ cat  main.tf
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
} cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_blockbnet-1"
  tags = {
     Name = "${var.env_prefix}-vpc"
  }
}esource "aws_default_route_table" "main_rt" {
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id
resource "aws_subnet" "myapp_subnet_1" {
  vpc_id     = aws_vpc.myapp_vpc.id
  cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone_igw.id
  tags = {
     Name = "${var.env_prefix}-subnet-1"
  }  Name = "${var.env_prefix}-rt"
} }
}
resource "aws_default_route_table" "main_rt" {
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id
  vpc_id = aws_vpc.myapp_vpc.id
  route {{
    cidr_block = "0.0.0.0/0"x}-igw"
    gateway_id = aws_internet_gateway.myapp_igw.id
  }
  tags = {
     Name = "${var.env_prefix}-rt"up" "default_sg" {
  }  _id      = aws_vpc.myapp_vpc.id
}
  ingress {
resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = aws_vpc.myapp_vpc.id
  tags = {ol     = "tcp"
     Name = "${var.env_prefix}-igw"
  }
```

### 1.8 User Data Script (Nginx Installation) (task1_userdata_nginx.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ nano entry-script.sh
@SadafRiaz-077 ➜ ~/Lab12 $ cat entry-script.sh
#!/bin/bash
set -e
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx

@SadafRiaz-077 ➜ ~/Lab12 $ chmod +x entry-script.sh
@SadafRiaz-077 ➜ ~/Lab12 $
```

### 1.9 SSH Key Generation (task1_ssh_key.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ 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:t/EqrKdnpg39vScfXTVbsfGVheC8aeqeZlaHb1lyp+A codespace@codespaces-018de5
The key's randomart image is:
+--[ED25519 256]--+
|            .. +=|
|           o  ..*|
|            o  o+|
|             o  =|
|        S o +. ..|
|       . . *+ o.=|
|      ... oo.+.*o|
|       oBo++E *. |
|      +O.** o*.  |
+----[SHA256]-----+
@SadafRiaz-077 ➜ ~/Lab12 $
```

### 1.10 Terraform Initialization (task1_tf_init.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/http...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (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.
@SadafRiaz-077 ➜ ~/Lab12 $
```

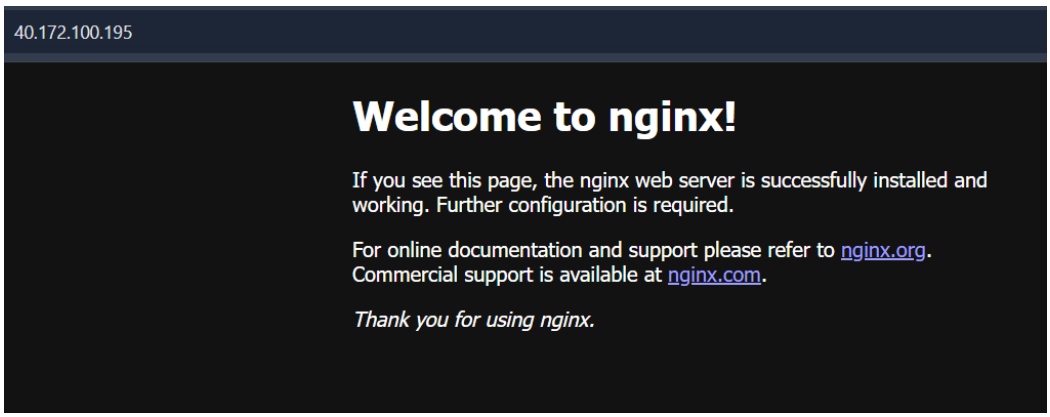
### 1.11 Terraform Apply (task1_tf_apply.png)

```
      + ipv6_cidr_block                     = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                 = (known after apply)
      + owner_id                            = (known after apply)
      + region                              = "me-central-1"
      + tags                                = {
          + "Name" = "dev-vpc"
        }
      + tags_all                            = {
          + "Name" = "dev-vpc"
        }
    }

Plan: 7 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + aws_instance_public_ip = (known after apply)
aws_key_pair.ssh-key: Creating...
aws_vpc.myapp_vpc: Creating...
aws_key_pair.ssh-key: Creation complete after 1s [id=serverkey]
aws_vpc.myapp_vpc: Creation complete after 2s [id=vpc-01fce2db4966cd0b8]
aws_internet_gateway.myapp_igw: Creating...
aws_subnet.myapp_subnet_1: Creating...
aws_default_security_group.default_sg: Creating...
aws_internet_gateway.myapp_igw: Creation complete after 0s [id=igw-06a02ab8c7234a16b]
aws_default_route_table.main_rt: Creating...
aws_subnet.myapp_subnet_1: Creation complete after 0s [id=subnet-0d0201fd919313783]
aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-01dd0139e291e502e]
aws_default_security_group.default_sg: Creation complete after 2s [id=sg-054dfcab2983d18ca]
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [00m10s elapsed]
aws_instance.myapp-server: Creation complete after 12s [id=i-0e312fb24ccb96733]

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

Outputs:

aws_instance_public_ip = "40.172.100.195"
```

### 1.12 Output Verification (task1_output_verify.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ terraform output
aws_instance_public_ip = "40.172.100.195"
@SadafRiaz-077 ➜ ~/Lab12 $
```

### 1.13 Nginx Browser Test (task1_nginx_browser.png)




### 1.14 Resource Cleanup (task1_cleanup.png)

```
aws_key_pair.ssh-key: Destruction complete after 1s
aws_subnet.myapp_subnet_1: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-01fce2db4966cd0b8]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
@SadafRiaz-077 ➜ ~/Lab12 $
```

## Task 2 – Using Remote-Exec Provisioner

### 2.1 task2_main_tf_remote_exec.png

```
@SadafRiaz-077 ➜ ~/Lab12_Task2 $ cat main.tf
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block

  tags = {
    Name = "${var.env_prefix}-vpc"
  }
}

resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = aws_vpc.myapp_vpc.id

  tags = {
    Name = "${var.env_prefix}-igw"
  }
}

resource "aws_subnet" "myapp_subnet_1" {
  vpc_id                  = aws_vpc.myapp_vpc.id
  cidr_block              = var.subnet_cidr_block
  availability_zone       = var.availability_zone
  map_public_ip_on_launch = true

  tags = {
    Name = "${var.env_prefix}-subnet-1"
  }
}

resource "aws_default_route_table" "main_rt" {
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.myapp_igw.id
  }
```

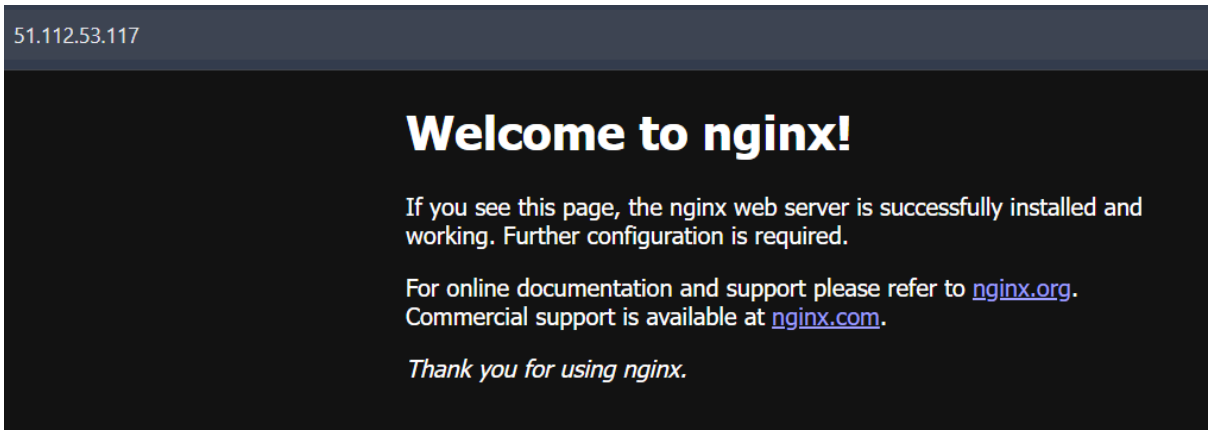
### 2.4 Terraform Apply (task2_tf_apply.png)

```
aws_instance.myapp-server (remote-exec): Complete!
aws_instance.myapp-server (remote-exec): Created symlink /etc/sys
aws_instance.myapp-server: Creation complete after 33s [id=i-03a5

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

Outputs:

aws_instance_public_ip = "51.112.53.117"
@SadafRiaz-077 ➜ ~/Lab12_Task2 $
```

### 2.5 Output Verification (task2_output_verify.png)

```
@SadafRiaz-077 ➜ ~/Lab12_Task2 $ terraform output
aws_instance_public_ip = "51.112.53.117"
@SadafRiaz-077 ➜ ~/Lab12_Task2 $
```

### 2.6 Nginx Validation (task2_nginx_validate.png)




## Task 3 – File and Local-Exec Provisioners

### 3.1 Main TF with All Provisioners (task3_main_tf_all_provisioners.png)

```
@SadafRiaz-077 ➜ ~/Lab12_Task2 $ mkdir -p ~/Lab12_Task3
@SadafRiaz-077 ➜ ~/Lab12_Task2 $ cd ~/Lab12_Task3
@SadafRiaz-077 ➜ ~/Lab12_Task3 $ touch main.tf variables.tf outputs.tf locals.tf terraform.tfvars entry-script.sh
```

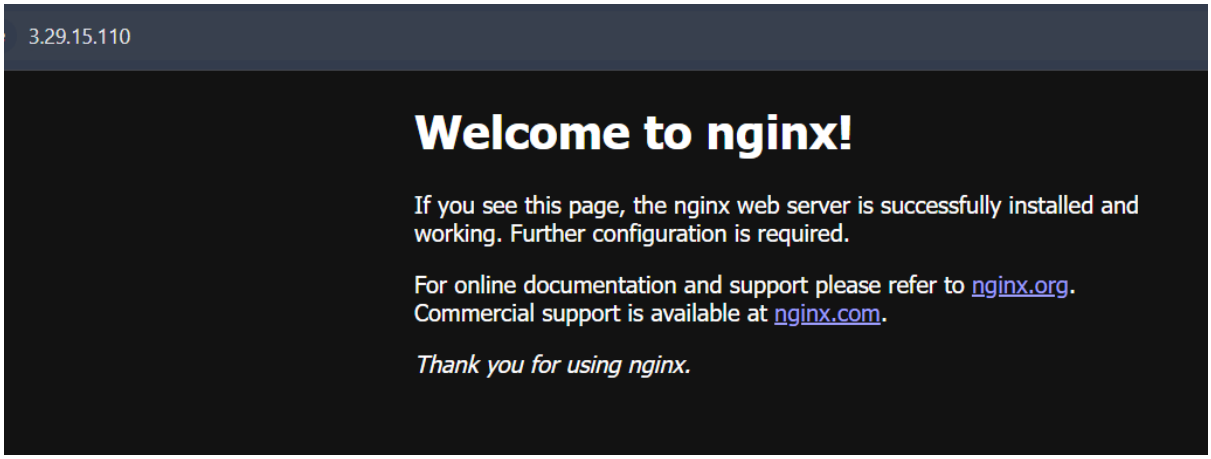
### 3.2 Terraform Apply (task3_terraform_apply.png)

```
aws_instance.myapp-server (local-exec): Instance i-0f1df6c3d89b9f678 with publ
aws_instance.myapp-server: Creation complete after 32s [id=i-0f1df6c3d89b9f678

Apply complete! Resources: 1 added, 0 changed, 2 destroyed.

Outputs:

aws_instance_public_ip = "3.29.15.110"
```

### 3.3 Terraform Output (task3_terraform_output.png)

```
@SadafRiaz-077 ➜ ~/Lab12_Task3 $ terraform output
aws_instance_public_ip = "3.29.15.110"
```

### 3.4 Nginx Browser Validation (task3_nginx_browser.png)



### 3.5 Terraform Destroy (task3_terraform_destroy.png)

```
aws_key_pair.ssh-key: Destruction complete after 0s
aws_subnet.myapp_subnet_1: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0f188dc49c255115b]
aws_vpc.myapp_vpc: Destruction complete after 0s

Destroy complete! Resources: 6 destroyed.
@SadafRiaz-077 ➜ ~/Lab12_Task3 $
```

### 3.6 Main TF Restored (task3_main_tf_restored.png)

```
resource "aws_key_pair" "ssh-key" {
  key_name   = "serverkey"
  public_key = file(var.public_key)
}
user_data = file("./entry-script.sh")

resource "aws_instance" "myapp-server" {
  ami                         = "ami-05524d6658fcf35b6"
  instance_type               = var.instance_type
  subnet_id                   = aws_subnet.myapp_subnet_1.id
  security_groups             = [aws_default_security_group.default_sg.id]
  associate_public_ip_address = true
  key_name                    = aws_key_pair.ssh-key.key_name
```

## Task 4 – Terraform Subnet Module

### 4.1 Module Directory Structure (task4_module_dir.png)

```
@SadafRiaz-077 ➜ ~/Lab12_Task3 $ mkdir -p ~/Lab12/modules/subnet
Lab12/modules/subnet
touch main.tf variables.tf outputs@SadafRiaz-077 ➜ ~/Lab12_Task3 $ cd ~/Lab12/modules/subnet
.tf
@SadafRiaz-077 ➜ ~/Lab12/modules/subnet $ touch main.tf variables.tf outputs.tf
@SadafRiaz-077 ➜ ~/Lab12/modules/subnet $
```

### 4.2 Subnet Module Variables (task4_module_vars.png)

```
@SadafRiaz-077 ➜ ~/Lab12/modules/subnet $ cat  variables.tf
variable "vpc_id" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "default_route_table_id" {}

@SadafRiaz-077 ➜ ~/Lab12/modules/subnet $
```

### 4.3 Subnet Module Resources (task4_module_res.png)

```
@SadafRiaz-077 ➜ ~/Lab12/modules/subnet $ @SadafRiaz-077 ➜ ~/Lab12/modules/subnet $ ca
resource "aws_subnet" "myapp_subnet_1" {
  vpc_id                  = var.vpc_id
  cidr_block              = var.subnet_cidr_block
  availability_zone       = var.availability_zone
  map_public_ip_on_launch = true

  tags = {
    Name = "${var.env_prefix}-subnet-1"
  }
}


resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = var.vpc_id

  tags = {
    Name = "${var.env_prefix}-igw"
  }
}


resource "aws_default_route_table" "main_rt" {
  default_route_table_id = var.default_route_table_id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.myapp_igw.id
  }

  tags = {
    Name = "${var.env_prefix}-rt"
  }
}


@SadafRiaz-077 ➜ ~/Lab12/modules/subnet $
```

### 4.4 Subnet Module Outputs (task4_module_out.png)

```
@SadafRiaz-077 ➜ ~/Lab12/modules/subnet $ nano outputs
@SadafRiaz-077 ➜ ~/Lab12/modules/subnet $ @SadafRiaz-077 ➜ ~/Lab12/modules/subnet $ cat  outputs
output "subnet" {
  value = aws_subnet.myapp_subnet_1
}

@SadafRiaz-077 ➜ ~/Lab12/modules/subnet $
```

### 4.5 Root Module Integration (task4_root_integ.png)

```
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "${var.env_prefix}-default-sg"
  }
}

#############################################################
# Key Pair
#############################################################
resource "aws_key_pair" "ssh-key" {
  key_name   = "serverkey"
  public_key = file(var.public_key)
}

#############################################################
# EC2 Instance
#############################################################
resource "aws_instance" "myapp-server" {
  ami           = "ami-05524d6658fcf35b6" # Amazon Linux 2023 Kernel 6.1 AMI
  instance_type = var.instance_type
  subnet_id     = module.myapp-subnet.subnet.id
  security_groups = [aws_default_security_group.default_sg.id]
  availability_zone = var.availability_zone
  associate_public_ip_address = true
  key_name = aws_key_pair.ssh-key.key_name

  # Simple user_data for Nginx
  user_data = file("./entry-script.sh")

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

@SadafRiaz-077 ➜ ~/Lab12 $
```

### 4.6 Terraform Initialization (task4_tf_init.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ terraform init
Initializing the backend...
Initializing modules...
- myapp-subnet in modules/subnet
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0
Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

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.
@SadafRiaz-077 ➜ ~/Lab12 $
```

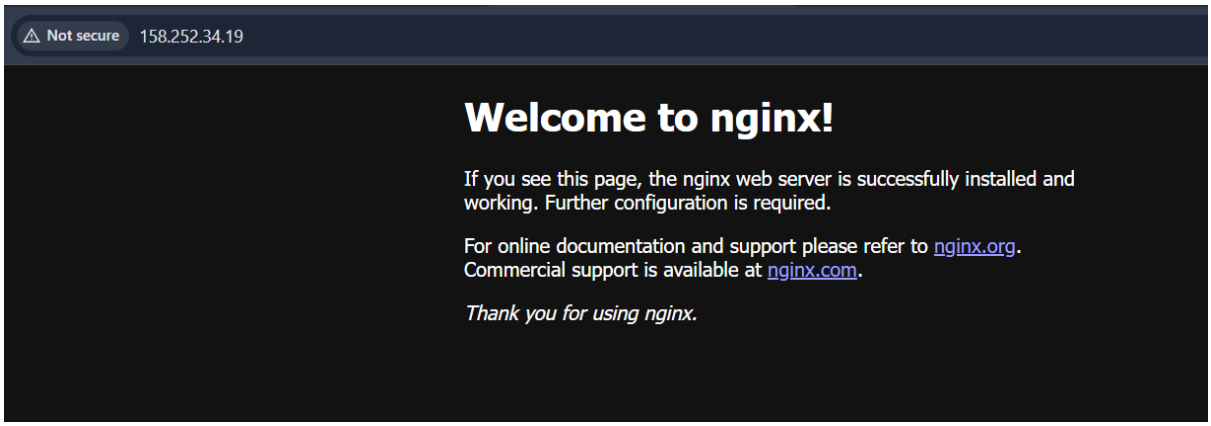
### 4.7 Terraform Apply (task4_tf_apply.png)

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

Outputs:

aws_instance_public_ip = "158.252.34.19"
@SadafRiaz-077 ➜ ~/Lab12 $
```

### 4.8 Output Verification (task4_output_verify.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ terraform output
aws_instance_public_ip = "158.252.34.19"
@SadafRiaz-077 ➜ ~/Lab12 $
```

### 4.9 Nginx Validation (task4_nginx_validate.png)




## Task 5 – Webserver Module

### 5.1 Webserver Module Structure – task5_webserver_module_structure.png

```
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $ cd modules/webserver
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12/modules/webserver (main) $ touch main.tf variables.tf outputs.tf
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12/modules/webserver (main) $ ls -la
total 8
drwxrwxrwx+ 2 codespace codespace 4096 Jan  2 11:13 .
drwxrwxrwx+ 3 codespace codespace 4096 Jan  2 11:13 ..
-rw-rw-rw-  1 codespace codespace    0 Jan  2 11:13 main.tf
-rw-rw-rw-  1 codespace codespace    0 Jan  2 11:13 outputs.tf
-rw-rw-rw-  1 codespace codespace    0 Jan  2 11:13 variables.tf
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12/modules/webserver (main) $
```

### 5.2 Webserver Module Variables – task5_webserver_variables.png

```
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12/modules/webserver (main) $ nano variables.tf
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12/modules/webserver (main) $ cat variables.tf
variable "env_prefix" {}
variable "instance_type" {}
variable "availability_zone" {}
variable "public_key" {}
variable "my_ip" {}
variable "vpc_id" {}
variable "subnet_id" {}
variable "script_path" {}
variable "instance_suffix" {}
```

### 5.3 Security Group Configuration – task5_webserver_main.png

```
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12/modules/webserver (main) $ nano main.tf
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12/modules/webserver (main) $ cat main.tf
resource "aws_security_group" "web_sg" {
  vpc_id      = var.vpc_id
  name        = "${var.env_prefix}-web-sg-${var.instance_suffix}"
  description = "Security group for web server allowing HTTP, HTTPS and SSH"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [var.my_ip]
  }
  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "${var.env_prefix}-default-sg"
  }
}

@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12/modules/webserver (main) $
```

### 5.4 EC2 Instance Configuration – task5_webserver_main.png *(same as above, combined)*

```
}
resource "aws_key_pair" "ssh-key" {
  key_name   = "${var.env_prefix}-serverkey-${var.instance_suffix}"
  public_key = file(var.public_key)
}

resource "aws_instance" "myapp-server" {
  ami                    = "ami-05524d6658fcf35b6" # Amazon Linux 2023
  instance_type          = var.instance_type
  subnet_id              = var.subnet_id
  vpc_security_group_ids = [aws_security_group.web_sg.id]
  availability_zone      = var.availability_zone
  associate_public_ip_address = true
  key_name               = aws_key_pair.ssh-key.key_name

  user_data = file(var.script_path)

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

### 5.5 Module Outputs – task5_webserver_outputs.png

```
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12/modules/webserver (main) $ cat outputs.tf
output "aws_instance" {
  value = aws_instance.myapp-server
}

@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12/modules/webserver (main) $
```

### 5.6 Root Module Integration – task5_main_tf_webserver_module.png

```
@SadafRiaz-077 ➜ ~/Lab12 $ nano main.tf
@SadafRiaz-077 ➜ ~/Lab12 $ @SadafRiaz-077 ➜ ~/Lab12 $ cat main.tf
# ======================
# PROVIDER CONFIGURATION
# ======================
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

# ======================
# VPC RESOURCE
# ======================
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block
  tags = {
    Name = "${var.env_prefix}-vpc"
  }
}

# ======================
# SUBNET MODULE
# ======================
module "myapp-subnet" {
  source = "./modules/subnet"
  vpc_id = aws_vpc.myapp_vpc.id
  subnet_cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  env_prefix = var.env_prefix
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id
}

# ======================
# WEB SERVER MODULE
# ======================
module "myapp-webserver" {
  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         = "./entry-script.sh"
  instance_suffix     = "0"
}

@SadafRiaz-077 ➜ ~/Lab12 $
```

### 5.7 Terraform Initialization – task5_terraform_init.png

```
@SadafRiaz-077 ➜ ~/Lab12 $ terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/http from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/http v3.5.0
- Using previously-installed hashicorp/aws v6.27.0

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.
@SadafRiaz-077 ➜ ~/Lab12 $
```

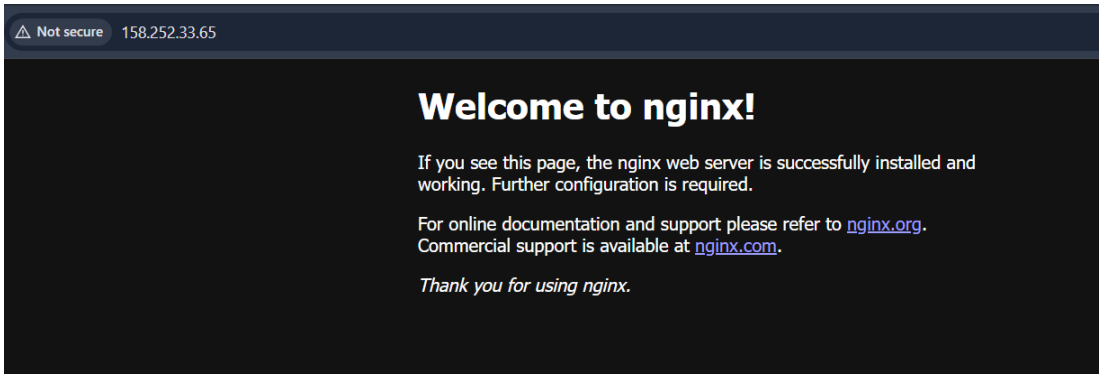
### 5.8 Terraform Apply – task5_terraform_apply.png

```
odule.myapp-webserver.aws_instance.myapp-server: Creation complete a

pply complete! Resources: 7 added, 0 changed, 0 destroyed.

utputs:

ebserver_public_ip = "158.252.33.65"
@SadafRiaz-077 ➜ ~/Lab12 $
```

### 5.9 Output Verification – task5_terraform_output.png

```
webserver_public_ip = "158.252.33.65"
@SadafRiaz-077 ➜ ~/Lab12 $ terraform output
webserver_public_ip = "158.252.33.65"
@SadafRiaz-077 ➜ ~/Lab12 $
```

### 5.10 Nginx Validation – task5_nginx_browser.png




### 5.11 Resource Cleanup – task5_terraform_destroy.png

```
module.myapp-webserver.aws_key_pair.ssh_key: Destruction complete after
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete aft
module.myapp-webserver.aws_security_group.web_sg: Destruction complete
aws_vpc.myapp_vpc: Destroying... [id=vpc-0bcfa74c41013c4fc]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
@SadafRiaz-077 ➜ ~/Lab12 $
```

## Task 6 – HTTPS Configuration with Self-Signed Certificates

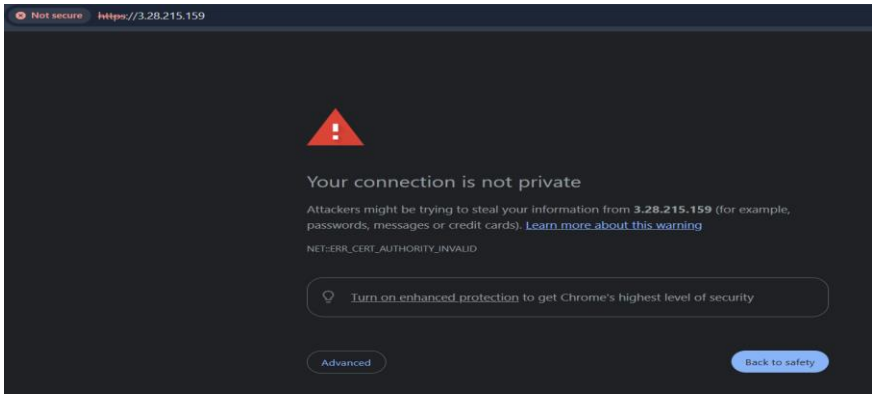
### 6.1 SSL Configuration Script (task6_ssl_config.png)

```
@SadafRiaz-077 ➜ ~/Lab12 $ nano entry-script.sh
@SadafRiaz-077 ➜ ~/Lab12 $ cat entry-script.sh
#!/bin/bash
set -e
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx

# Create directories for SSL certificates if they don't exist
mkdir -p /etc/ssl/private
mkdir -p /etc/ssl/certs

# Get IMDSv2 token
TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" \
  -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

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

# 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 "/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"

# Backup existing nginx.conf
cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak

# Overwrite nginx.conf with SSL configuration
cat <<EOF > /etc/nginx/nginx.conf
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;
```

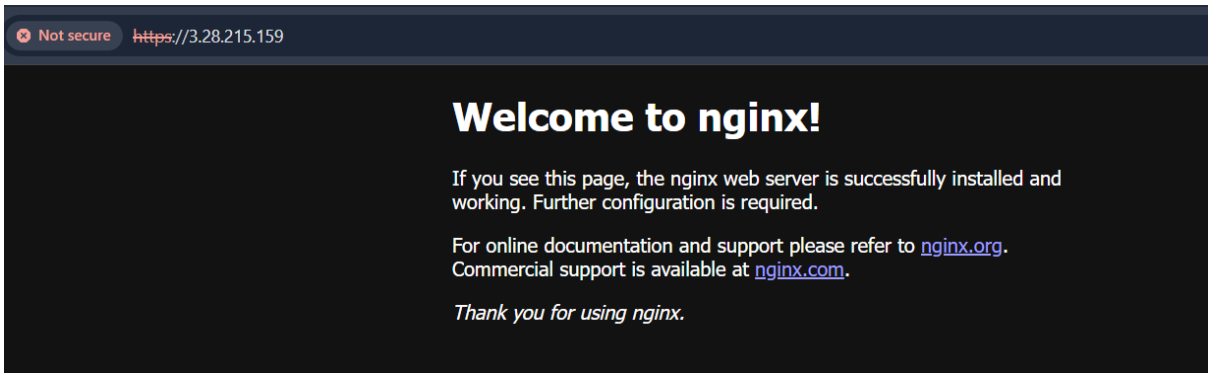
## 6.2 Terraform Apply (task6_tf_apply.png)

```
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creation complete after 0s [id=igw-0b210fe758f66e452]
module.myapp-subnet.aws_default_route_table.main_rt: Creating...
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-0a6ed38142104af17]
module.myapp-webserver.aws_security_group.web_sg: Creation complete after 2s [id=sg-050b0d03376f0e6a4]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [00m10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 11s [id=subnet-0b6e8085ebb5883ef]
module.myapp-webserver.aws_instance.myapp-server: Creating...
module.myapp-webserver.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Creation complete after 15s [id=i-0d98e9cde83d15524]

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

Outputs:

webserver_public_ip = "3.28.215.159"
@SadafRiaz-077 ➜ ~/Lab12 $
```

## 6.3 HTTPS Browser Testing (task6_browser_test.png)



### 6.4 HTTP to HTTPS Redirection (task6_http_https.png)

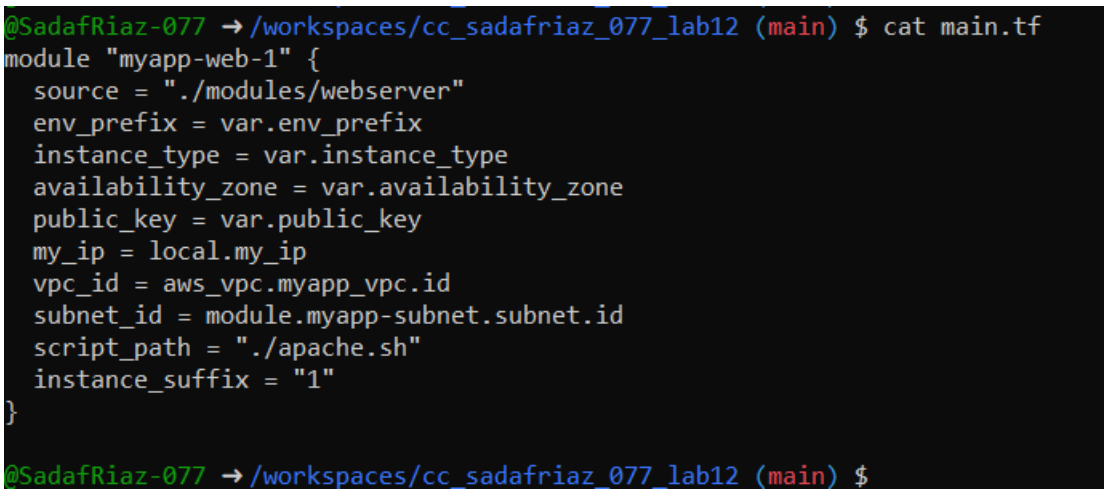



## Task 7 – Nginx Reverse Proxy Configuration

### 7.1 Apache Backend Server Script_task7_apache_script.png

```
@SadafRiaz-077 ➜ ~/Lab12 $ cat apache.sh
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd

echo "<h1>Welcome to My Web Server</h1>" > /var/www/html/index.html
hostnamectl set-hostname myapp-webserver

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

echo "<h2>Hostname: $(hostname)</h2>" >> /var/www/html/index.html
echo "<h2>Private IP: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/local-ipv4)</h2>" >> /
echo "<h2>Public IP: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-ipv4)</h2>" >> /
echo "<h2>Public DNS: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-hostname)</h2>"
echo "<h2>Deployed via Terraform</h2>" >> /var/www/html/index.html
```

### 7.2 Backend Webserver Deployment_task7_main_tf_web1.png



```
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $ cat main.tf
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"
}

@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $
```

### 7.4 Terraform Apply_task7_terraform_apply.png

```
module.myapp-web-1.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Creation complete after 13s [id=i-0554f895027a25567]

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

Outputs:

aws_web_1_public_ip = "51.112.253.145"
aws_webserver_public_ip = "3.28.135.180"
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $ chmod 400 mykey.pem
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $
```

### 7.6 SSH Access to Proxy Server_task7_ssh_webserver.png

```
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $ ssh -i mykey.pem ec2-user@3.28.135.18
     ,     #_
     ~\_  ####_        Amazon Linux 2023
    ~~  \_#####\
    ~~     \###|
    ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
     ~~       V~' '->
      ~~~         /
        ~~._.   _/
           _/ _/
         _/m/'
Last login: Sat Jan  3 16:47:10 2026 from 20.192.21.48
[ec2-user@myapp-webserver ~]$
```

### 7.7 Reverse Proxy Configuration_task7_nginx_conf_reverse_proxy.png

```
    * Official English Documentation: http://nginx.org/en/docs/
    * Official Russian Documentation: http://nginx.org/ru/docs/

ser nginx;
orker_processes auto;
rror_log /var/log/nginx/error.log notice;
id /run/nginx.pid;

 Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
nclude /usr/share/nginx/modules/*.conf;

vents {
    worker_connections 1024;


ttp {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include             /etc/nginx/mime.types;
/etc/nginx/nginx.conf" 82L, 2317B
```

### 7.8 Nginx Restart_task7_nginx_restart.png

```
[ec2-user@myapp-webserver ~]$ sudo systemctl restart nginx

sudo systemctl status nginx
[ec2-user@myapp-webserver ~]$
```

### 7.9 Error Log Verification_task7_error_log.png

```
2026/01/03 17:08:08 [notice] 27361#27361: try again to bind() after 500ms
2026/01/03 17:08:08 [emerg] 27361#27361: still could not bind()
2026/01/03 17:09:25 [notice] 27436#27436: using the "epoll" event method
2026/01/03 17:09:25 [notice] 27436#27436: nginx/1.28.0
2026/01/03 17:09:25 [notice] 27436#27436: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2026/01/03 17:09:25 [notice] 27436#27436: getrlimit(RLIMIT_NOFILE): 65535:65535
2026/01/03 17:09:25 [notice] 27437#27437: start worker processes
2026/01/03 17:09:25 [notice] 27437#27437: start worker process 27438
2026/01/03 17:09:25 [notice] 27437#27437: start worker process 27439
2026/01/03 17:14:27 [emerg] 27599#27599: "proxy_pass" directive is not allowed here in
2026/01/03 17:16:41 [warn] 27705#27705: could not build optimal types_hash, you should
ze: 1024 or types_hash_bucket_size: 64; ignoring types_hash_bucket_size
2026/01/03 17:16:54 [notice] 27437#27437: signal 3 (SIGQUIT) received from 1, shutting
2026/01/03 17:16:54 [notice] 27438#27438: gracefully shutting down
2026/01/03 17:16:54 [notice] 27438#27438: exiting
2026/01/03 17:16:54 [notice] 27438#27438: exit
2026/01/03 17:16:54 [notice] 27439#27439: gracefully shutting down
2026/01/03 17:16:54 [notice] 27439#27439: exiting
2026/01/03 17:16:54 [notice] 27439#27439: exit
2026/01/03 17:16:54 [notice] 27437#27437: signal 17 (SIGCHLD) received from 27438
2026/01/03 17:16:54 [notice] 27437#27437: worker process 27438 exited with code 0
2026/01/03 17:16:54 [notice] 27437#27437: signal 29 (SIGIO) received
2026/01/03 17:16:54 [notice] 27437#27437: signal 17 (SIGCHLD) received from 27439
2026/01/03 17:16:54 [notice] 27437#27437: worker process 27439 exited with code 0
2026/01/03 17:16:54 [notice] 27437#27437: exit
2026/01/03 17:16:54 [warn] 27715#27715: could not build optimal types_hash, you should
ze: 1024 or types_hash_bucket_size: 64; ignoring types_hash_bucket_size
2026/01/03 17:16:54 [warn] 27716#27716: could not build optimal types_hash, you should
ze: 1024 or types_hash_bucket_size: 64; ignoring types_hash_bucket_size
[ec2-user@myapp-webserver ~]$
```

### 7.10 Access Log Verification_task7_access_log.png

```
ze: 1024 or types_hash_bucket_size: 64; ignoring types_hash_buc
[ec2-user@myapp-webserver ~]$ cat /var/log/nginx/access.log
[ec2-user@myapp-webserver ~]$
```

Provisioners Setup

### 7.11 Mime Types Verification_task7_mime_types.png

```
image/x-cmu-raster                              ras;
image/x-portable-anymap                         pnm;
image/x-portable-bitmap                         pbm;
image/x-portable-graymap                        pgm;
image/x-portable-pixmap                         ppm;
image/x-rgb                                     rgb;
image/x-targa                                   tga;
image/x-xbitmap                                 xbm;
image/x-xpixmap                                 xpm;
image/x-xwindowdump                             xwd;
text/html-sandboxed                             sandboxed;
text/x-pod                                      pod;
text/x-setext                                   etx;
video/webm                                      webm;
video/x-annodex                                 axv;
video/x-flv                                     flv;
video/x-javafx                                  fxm;
video/x-matroska                                mkv;
video/x-matroska-3d                             mk3d;
video/x-ms-asf                                  asx;
video/x-ms-wm                                   wm;
video/x-ms-wmv                                  wmv;
video/x-ms-wmx                                  wmx;
video/x-ms-wvx                                  wvx;
video/x-msvideo                                 avi;
video/x-sgi-movie                               movie;
x-conference/x-cooltalk                         ice;
x-epoc/x-sisx-app                               sisx;
}
[ec2-user@myapp-webserver ~]$
```

### 7.12 SSL Certificate Verification_task7_ssl_cert.png

```
[ec2-user@myapp-webserver ~]$ sudo cat /etc/ssl/private/selfsigned.key
-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCwp3Vos7/PClig
f5GTQ+UbMGgtP4gb2ODCi+j3ib/ogqU/2f/N1rNDqF99CP1DhVMiVgSZwftQHUE
/+6Cxd2vQKhI1H1Z6eOxwNne1XQMoQdVeh4foEPpMxgzhUgallHEKu39D8P9y9lh
OI60IrlWD9oxv+sw8EvwAN1HQ8iING1mW9TqFfTlHH0uPA8cQB8IoxykgvgGFSU
mVi3p0WdoqHjblTdqaQLS0BCphhuAxz2G5y5qpxATwNAGrBr2Nc6PfWN6prllFd
gi4vwIk39Ck5IXHfsmZomKAj78qSSxcw7sm7OPQcvkfL+Ov2jMC9K6k9//63C81
ETVXXFxAgMBAAECggEAR/jiF/dsBWF0vFMqbuAEofcxaFV1WCepxoDG6h+fifUu
fIh+iXe3EeaNGzD12WiUeutWK6E4lIv3YP6mWTdeW4GfTpTsQU7OiuREn8odjuuo
fkXTshTstj0v9gWuW860n5TSA5GBy4xbE4SlS0sd29T5eKnjDzNuKRM1FWjyendw
T+cQ01EE9LKNGd+jo0zMQpV0pNDKLfzm3K9Tp8+H7Ao34CUKePj1PwojGdEoEuU
vYpxh4blYUzoFV8B7LNU6BO9kDX//vPCC4wNESe/AcT2cDylbTL3o2Wh11O9mqb
sSOuQ+82Nf7VdQmzbbe1nG37Db7bMPgV3CtlPtmlQKBgQDjYAiQFaiHGJQ4UOws
e/hRYNEpYJ//n3IBI0Zt9C8gD01N1Yt7gY/a4x8v2T4sjGvzm9BJt/k4ipc1MSM
cX21/LARdX++HBHeBZWNnsDqK9C/BUOnGzF2Ryh/zj3Ln3JQuPDmUBqFVwDU8RQ
1DE1Msli5Y1M85GODwhmDF5DwKBgQDG5MMRbzUoTaLBPHeMvxZxSn4KmCLUaX0N
4ti6i8DwUeG9iL4/P2aXI11y4xIh0iSaJS1GnqESeziQPoLgFOvA+flpESnsW6M
W4Cis9UPY//6FSgDabbhNW9sk4BXkrnD90YeLN8JqLVLeF60vUsEjYXg8varSLR
FFcUUltfwKBgCNfyXVBKVr7Q/QPAu9aVxuhCO8hINBw83iu3JTnuB3bhapF1a6H
Yk4Xj76N+dJMdm8MCretW6SkSyoHXwTnwzsQ+QN+WEVncJAuJjpGPJUyA+P0GbZ
0KwqHe7kU1DemZZdb30lcTrVcleuugH3AT5cTGCKfhU+v6JuTEw0GGZAoGASed3
GNePTub3nG5enMGw9IKQ6WES6lm7mO7R2X9UkffbCbJl/tpwJHjfoHD/q9GKCD0
NdV7xIJE4lRYDSu47lv0cTYPfeQPIAcgJVTtNmrSEvsJgzsk2UohvRaR+fhrToX
aAMEXXJy6zgu+LndxjNQK+YcAklhxV2pfcOccECgYBxFf9lZDAEOXUTcbZ9QeSa
J4/76erZY4DI8Zm0px7kJXtzsqquD9dV9xt7SCT3/EVR81AlH5Z5iZJRtv4WfCY
GDtcG9RZsfQDl2Fgmp5+2/pWLz1phrh4cPf56wymTKLPODF3H1w793Ur/CvCvFm
k7sL3FzWZlOhm5NkHP1kQ==
-----END PRIVATE KEY-----
[ec2-user@myapp-webserver ~]$
```

### 7.13 SSL Key Verification_task7_ssl_key.png

```
[ec2-user@myapp-webserver ~]$ sudo cat /etc/ssl/certs/selfsigned.crt
-----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIUSJpcSDBKMXfMN9iA0normt34SGYwDQYJKoZIhvcNAQEL
BQAwZzELMAkGA1UEBhMCUEsxDzANBgNVBAgMBlB1bmphYjEPMA0GA1UEBwwGTGFo
b3JlMQ4wDAYDVQQKDAVNeU9yZzEMMAoGA1UECwwDRGV2MRgwFgYDVQQDDA9teWFw
cC13ZWJzZXJ2ZXIwHhcNMjYwMTAzMTcyMTExWhcNMjcwMTAzMTcyMTExWjBnMQsw
CQYDVQQGEwJQSzEPMA0GA1UECAwGUHVuamFiMQ8wDQYDVQQHDAZMYWhvcmUxDjAM
BgNVBAoMBU15T3JnMQwwCgYDVQQLDANEZXYxGDAWBgNVBAMMD215YXBwLXdlYnNl
cnZlcjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALCndWizv88KWKBN
/kZND5RswaC0/iBvY4MKL6PeJv+iCpT/Z/83Ws0OoX30I/UOFUyJWBJnB+1AdQRX
7oLF3a9AqEjUfVnp47HA2d7VdAyhB1V6Hh+gQ+kzGDOFSBqWUcQq7f0Pw/3L2WE0
4jrQiuVYP2jG/6zDwS/AA3UdDyIg0bWZb1OoV9OUcfS48DxxAHwijHKSC+AYVJQ6
ZWLenRZ2ioeNuVN2ppAtLQEKmGG4DHPYbnLmqnEBPA0AasGvY1zo99Y3qmuWUV16
CLi/AiTf0KTkhcd+yZmiYoCPvypJLFzDuybs49By+R8v46/aMwL0rqT3//rcLzVw
RNVdcXECAwEAAaNTMFEwHQYDVR0OBBYEFD7OEvi7diQL44itWW0OvsH3xfdvMB8G
A1UdIwQYMBaAFD7OEvi7diQL44itWW0OvsH3xfdvMA8GA1UdEwEB/wQFMAMBAf8w
DQYJKoZIhvcNAQELBQADggEBAF8GXJEMaKgELi4q+lefWdh4/iTXZVzoKEypNAbQ
9XWXWpdpfLXnqA23MiuwVylUxNUvhUJi9uumWvPf538WnnMBz/LpBh7aUKezmT91
07mBXrgjJCOSR4mX25A4HbMVThenv1IeHraB5cg4zVojZ+Gp/vl4deLywgCZD/5e
QIgbOgIkTGG5LkCEkw7EQUwgI3F8vokvFZgNyVnNaTV3pebgKWVMrZnKGFlGmxK6
2aVhuZNXsdolpySuJA1eV1/U/p7w/iSTiLAH8mEaXGkzZwvOi/rvrbnf+zUcVJOL
hZkCs9mb0LA7vmL2BThcLjp1qsp0M6fUa9Jovxc0b6FIGbw=
-----END CERTIFICATE-----
[ec2-user@myapp-webserver ~]$
```

**7.14 Reverse Proxy Testing_task7_reverse_proxy_browser.png**


**Task 8 – Nginx Load Balancer Configuration**

**8.1 Second Backend Server Deployment – task8_main_tf_web2.png**

```
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"
}
```

**8.2 Output Updates – task8_outputs_web2.png**

```
output "nginx_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
}
```

**8.3 Terraform Apply – task8_terraform_apply.png**

```
Apply complete! Resources: 12 added, 0 changed, 0 destroyed.

Outputs:

aws_web-1_public_ip = "51.112.230.24"
aws_web-2_public_ip = "3.28.41.141"
nginx_public_ip = "3.29.137.238"
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $
```

**8.5 Load Balancer Configuration – task8_nginx_conf_load_balancer.png**

```
#    }
}
[ec2-user@ip-10-0-10-80 ~]$ sudo vim /etc/nginx/nginx.conf
[ec2-user@ip-10-0-10-80 ~]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-0-10-80 ~]$
```

**8.6 Nginx Restart – task8_nginx_restart.png**

```
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-0-10-80 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-80 ~]$
```

**8.7 Load Balancing Verification Web 1 – task8_load_balancer_web1.png**

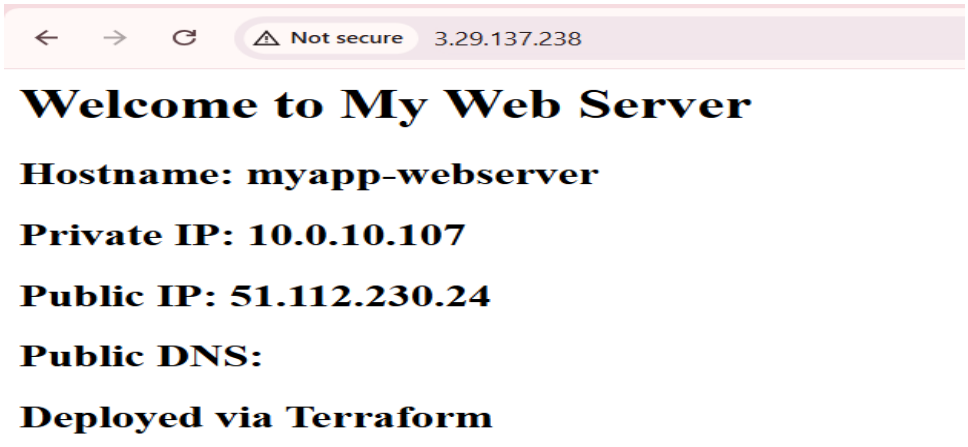
### 8.8 Load Balancing Verification Web 2 – task8_load_balancer_web2.png


## Task 9 – High Availability Configuration

### 9.1 Primary & Backup Server Setup (task9_primary_backup.png)

```
upstream backend_servers {
    server 51.112.230.24:80;        # Web-1 Public IP
    server 3.28.41.141:80 backup;   # Web-2 Public IP (Backup Server)
}

# Main server block (HTTPS)
server {
    listen 443 ssl;
    server_name 3.29.137.238;  # Use your Nginx Public IP or domain

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

    location / {
        proxy_pass http://backend_servers;  # Use the upstream backend_servers for load balancing
    }
}
```

### 9.2 Web-1 as Primary Server (task9_web1_primary.png)



### 9.3 Web-2 as Primary Server (task9_web2_primary.png)

```
http {
    upstream backend_servers {
        # Web-1 is the backup server
        server 51.112.230.24:80 backup;

        # Web-2 is the primary server
        server 3.28.41.141:80;
    }
```

### 9.4 High Availability Validation (task9_ha_validate.png)


## Task 10 – Nginx Caching Configuration

### 10.1 Cache Configuration (task10_cache_config.png)

```
[ec2-user@ip-10-0-10-80 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-80 ~]$ cat  nano /etc/nginx/nginx.conf
cat: nano: No such file or directory
# /etc/nginx/nginx.conf

worker_processes auto;
error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

# Load dynamic modules. See /usr/share/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    # Define proxy cache settings
    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"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

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

    # Backend servers configuration for load balancing
    upstream backend_servers {
        server 51.112.230.24:80;  # web-1 public IP
        server 3.28.41.141:80;    # web-2 public IP
    }

    # Server block for Nginx reverse proxy
    server {
        listen 443 ssl;
        server_name $PUBLIC_IP;

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

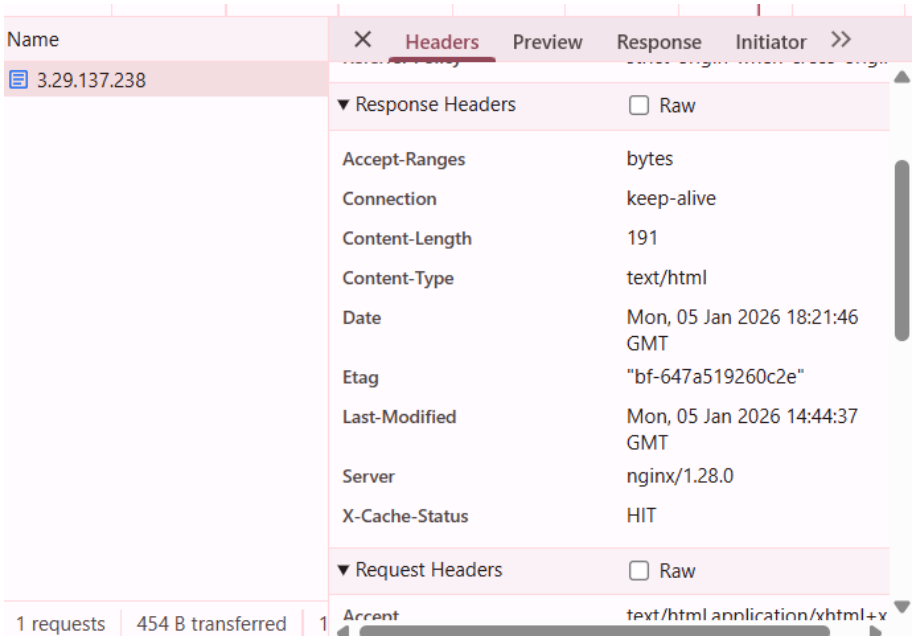
### 10.2 Nginx Restart (task10_nginx_restart.png)

```
[ec2-user@ip-10-0-10-80 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-80 ~]$
```

### 10.3 Cache MISS Verification (task10_cache_miss.png)

| | |
|---|---|
| Server | nginx/1.28.0 |
| X-Cache-Status | MISS |

### 10.4 Cache HIT Verification (task10_cache_hit.png)




### 10.5 Cache Directory Validation (task10_cache_dir.png)

```
[ec2-user@ip-10-0-10-80 ~]$ sudo ls -la /var/cache/nginx/
total 0
drwx------.  4 nginx root   24 Jan  5 18:09 .
drwxr-xr-x. 10 root  root  114 Jan  5 15:57 ..
drwx------.  3 nginx nginx  16 Jan  5 18:09 4
drwx------.  3 nginx nginx  16 Jan  5 17:27 9
[ec2-user@ip-10-0-10-80 ~]$
```

## Cleanup

### Cleanup.1 SSH Exit (cleanup_ssh_exit.png)

```
[ec2-user@myapp-webserver ~]$ exit
logout
Connection to 51.112.230.24 closed.
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $
```

### Cleanup.2 Terraform Destroy (cleanup_tf_destroy.png)

```
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete after 1s
module.myapp-webserver.aws_security_group.web_sg: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0c6e0c03fe31322f9]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 12 destroyed.
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $
```

**Cleanup.3 State File Verification (cleanup_state_verify.png)**

```
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 86,
  "lineage": "121aee07-c791-5c2b-4ff3-4f9dcca42adc",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

**Cleanup.4 Final Directory Structure (cleanup_final_dir.png)**

```
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $ ls -la
total 208
drwxrwxrwx+ 5 codespace root      4096 Jan  5 18:42 .
drwxr-xrwx+ 5 codespace root      4096 Dec 31 12:05 ..
drwxrwxrwx+ 8 codespace root      4096 Dec 31 12:05 .git
drwxrwxrwx+ 4 codespace codespace 4096 Jan  4 18:07 .terraform
-rw-r--r--  1 codespace codespace 2422 Jan  4 17:56 .terraform.lock.hcl
-rw-rw-rw-  1 codespace root        25 Dec 31 12:04 README.md
-rwxrwxrwx  1 codespace codespace  931 Jan  4 16:02 apache.sh
-rw-rw-rw-  1 codespace codespace    0 Jan  3 20:50 aws_key_pair
-rw-rw-rw-  1 codespace codespace 1769 Jan  4 17:50 entry-script.sh
-r--------  1 codespace codespace 1876 Jan  3 16:08 finalkey
-rw-r--r--  1 codespace codespace  409 Jan  3 16:08 finalkey.pub
-rw-rw-rw-  1 codespace codespace  126 Jan  4 17:47 locals.tf
-rw-rw-rw-  1 codespace codespace 1892 Jan  5 18:37 main.tf
-rw-rw-rw-  1 codespace codespace 1606 Jan  4 15:45 main.tf.bak
drwxrwxrwx+ 4 codespace codespace 4096 Jan  4 17:05 modules
-r--------  1 codespace codespace  389 Jan  2 18:31 mykey.pem
-r--------  1 codespace codespace 1896 Jan  3 15:40 mynewkey.pem
-rw-r--r--  1 codespace codespace  409 Jan  3 15:40 mynewkey.pem.pub
-rw-rw-rw-  1 codespace codespace   57 Jan  4 17:53 output
-rw-rw-rw-  1 codespace codespace  258 Jan  5 18:38 outputs.tf
-rw-rw-rw-  1 codespace codespace  146 Jan  4 15:34 outputs.tf.bak
-rw-rw-rw-  1 codespace codespace  212 Jan  3 20:50 resource
-rw-rw-rw-  1 codespace codespace  182 Jan  5 18:42 terraform.tfstate
-rw-rw-rw-  1 codespace codespace 5912 Jan  4 18:16 terraform.tfstate.1767550617.backup
-rw-rw-rw-  1 codespace codespace 4567 Jan  4 18:17 terraform.tfstate.1767550619.backup
-rw-rw-rw-  1 codespace codespace 3222 Jan  4 18:17 terraform.tfstate.1767550621.backup
-rw-rw-rw-  1 codespace codespace 5912 Jan  4 18:18 terraform.tfstate.1767550733.backup
-rw-rw-rw-  1 codespace codespace 4567 Jan  4 18:18 terraform.tfstate.1767550734.backup
-rw-rw-rw-  1 codespace codespace 3222 Jan  4 18:18 terraform.tfstate.1767550736.backup
-rw-rw-rw-  1 codespace codespace 41023 Jan  5 18:42 terraform.tfstate.backup
-rw-rw-rw-  1 codespace codespace  266 Jan  4 17:58 terraform.tfvars
-rw-rw-rw-  1 codespace codespace  234 Jan  4 16:00 terraform.tfvars.bak
-rw-rw-rw-  1 codespace codespace  261 Jan  4 17:45 variable
-rw-------  1 codespace codespace  199 Jan  4 15:30 variable.tf.save
-rw-rw-rw-  1 codespace codespace  198 Jan  4 17:44 variables.tf
-rw-rw-rw-  1 codespace codespace  255 Jan  4 15:43 variables.tf.bak
-rw-rw-rw-  1 codespace codespace  212 Jan  3 20:51 web1_key
-rw-rw-rw-  1 codespace codespace   32 Jan  4 17:53 '{'
-rw-rw-rw-  1 codespace codespace  261 Jan  4 17:45 {}
@SadafRiaz-077 ➜ /workspaces/cc_sadafriaz_077_lab12 (main) $
```