CLOUD COMPUTING LAB

LAB 12

Submitted By: **Urwa Zahra**

Roll No: **2023-BSE-068**

Submitted To: **Sir Muhammad Shoaib**

Task 0 Lab Setup (Codespace & GH CLI)

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068 (main) $ gh codespace list
NAME                                DISPLAY NAME        REPOSITORY                          BRANCH  STATE      CREATED AT
studious-guide-r45rj7gx6699f56wg    studious guide      Urwa012/cc-urwazahra-2023-BSE-068   main*   Shutdown   about 2 days ago
turbo-space-yodel-q759wvqx4gjp35g9  turbo space yodel   Urwa012/cc-urwazahra-2023-BSE-068   main    Shutdown   about 1 day ago
bookish-goldfish-pj5gx7r6w6pgc795w  bookish goldfish    Urwa012/cc-urwazahra-2023-BSE-068   main    Available  about 13 hours ago
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068 (main) $ gh codespace ssh -c bookish-goldfish-pj5gx7r6w6pgc795w
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068 (main) $ gh codespace ssh -c bookish-goldfish-pj5gx7r6w6pgc795w
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: Tue Jan  6 09:43:03 2026 from ::1
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068 (main) $
```

Task 1 — Organize Terraform code into separate files

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068 (main) $ mkdir -p ~/Lab12
cd ~/Lab12
```

```
cd ~/Lab12
@Urwa012 ➜~/Lab12 $ touch main.tf variables.tf outputs.tf locals.tf terraform.tfvars entry-script.sh
```

```
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "instance_type" {}
variable "public_key" {}
variable "private_key" {}
~
~
~
```

```
output "aws_instance_public_ip" {
  value = aws_instance.my_pp-server.public_ip
}
~
~
~
~
```

```
locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"
}

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

```
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"
~
~
~
~
```

```
resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = aws_vpc.myapp_vpc.id
    tags = {
     Name = "${var.env_prefix}-igw"
  }
}

resource "aws_default_security_group" "default_sg" {
  vpc_id      = aws_vpc.myapp_vpc.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [local.my_ip]
  }
  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"]
    prefix_list_ids = []
}
  tags = {
     Name = "${var.env_prefix}-default-sg"
  }
}

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

resource "aws_instance" "myapp-server" {
  ami           = "ami-05524d6658fcf35b6" # Amazon Linux 2023 Kernel 6.1 AMI
  instance_type = var.instance_type
  subnet_id     = aws_subnet.myapp_subnet_1.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

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

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

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

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068 (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:+QOMk4Gd2uDczfX59UjZ3IMq0of2lucJ9a/ABzTFHsg codespace@codespaces-3f967b
The key's randomart image is:
+--[ED25519 256]--+
|           . o.  |
|     o .    E.o  |
|    o +   . o. . |
|   o = B o o oo+.|
|    + * S   +oo.=|
|       ..o..o+.oo|
|       . =o++ +..|
|        o ++.+. .|
|          ..oo...|
+----[SHA256]-----+
+----[SHA256]-----+
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068 (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/http...
- Installing hashicorp/aws v6.27.0...
- 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.
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068 (main) $
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform apply -auto-approve
    }

  # aws_vpc.myapp_vpc will be created
  + resource "aws_vpc" "myapp_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + 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 0s [id=serverkey]
aws_vpc.myapp_vpc: Creation complete after 1s [id=vpc-0387ea55f51cab535]
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 1s [id=igw-0a884bed05a0395bf]
aws_default_route_table.main_rt: Creating...
aws_subnet.myapp_subnet_1: Creation complete after 1s [id=subnet-0fcb32f21afec245b]
aws_default_route_table.main_rt: Creation complete after 0s [id=rtb-0094ef33d921a250a]
aws_default_security_group.default_sg: Creation complete after 3s [id=sg-097aab856ed6e88b3]
aws_instance.myapp_server: Creating...
aws_instance.myapp_server: Still creating... [00m10s elapsed]
aws_instance.myapp_server: Creation complete after 13s [id=i-0fe7794598fc6e4f6]

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

Outputs:

aws_instance_public_ip = "40.172.186.89"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform outp
aws_instance_public_ip = "40.172.186.89"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

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.

```
aws_vpc.myapp_vpc: Destroying... [id=vpc-0387ea55f51cab535]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main)
```

Task 2 — Use remote-exec provisioner

```
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]
  availability_zone           = var.availability_zone
  associate_public_ip_address = true
  key_name                    = aws_key_pair.ssh_key.key_name

  connection {
    type        = "ssh"
    user        = "ec2-user"
    private_key = file(var.private_key)
    host        = self.public_ip
  }

  provisioner "remote-exec" {
    inline = [
      "sudo yum update -y",
      "sudo yum install -y nginx",
      "sudo systemctl start nginx",
      "sudo systemctl enable nginx"
     ]
  }

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

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform apply -auto-approve
aws_instance.myapp_server (remote-exec):   Installing       : gperf [      ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperf [=     ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperf [==    ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperf [===   ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperf [====  ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperf [===== ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperftools-1   4/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [      ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [=     ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [==    ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [===   ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [====  ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [===== ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx-core-1   5/7
aws_instance.myapp_server (remote-exec):   Installing       : gener [      ] 6/7
aws_instance.myapp_server (remote-exec):   Installing       : gener [====  ] 6/7
aws_instance.myapp_server (remote-exec):   Installing       : gener [===== ] 6/7
aws_instance.myapp_server (remote-exec):   Installing       : generic-logo   6/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [      ] 7/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [==    ] 7/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [===   ] 7/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [===== ] 7/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx-1:1.28   7/7
aws_instance.myapp_server (remote-exec):   Running scriptlet: nginx-1:1.28   7/7
aws_instance.myapp_server: Still creating... [00m30s elapsed]
aws_instance.myapp_server (remote-exec):   Verifying        : generic-logo   1/7
aws_instance.myapp_server (remote-exec):   Verifying        : gperftools-1   2/7
aws_instance.myapp_server (remote-exec):   Verifying        : libunwind-1.   3/7
aws_instance.myapp_server (remote-exec):   Verifying        : nginx-1:1.28   4/7
aws_instance.myapp_server (remote-exec):   Verifying        : nginx-core-1   5/7
aws_instance.myapp_server (remote-exec):   Verifying        : nginx-filesy   6/7
aws_instance.myapp_server (remote-exec):   Verifying        : nginx-mimety   7/7

aws_instance.myapp_server (remote-exec): Installed:
aws_instance.myapp_server (remote-exec):   generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
aws_instance.myapp_server (remote-exec):   gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
aws_instance.myapp_server (remote-exec):   libunwind-1.4.0-5.amzn2023.0.3.x86_64
aws_instance.myapp_server (remote-exec):   nginx-1:1.28.0-1.amzn2023.0.2.x86_64
aws_instance.myapp_server (remote-exec):   nginx-core-1:1.28.0-1.amzn2023.0.2.x86_64
aws_instance.myapp_server (remote-exec):   nginx-filesystem-1:1.28.0-1.amzn2023.0.2.noarch
aws_instance.myapp_server (remote-exec):   nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

aws_instance.myapp_server (remote-exec): Complete!
aws_instance.myapp_server (remote-exec): Created symlink /etc/systemd/system/multi-user.target.
ervice.
aws_instance.myapp_server: Creation complete after 31s [id=i-08448bef3dc1c7dd2]

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

Outputs:

aws_instance_public_ip = "3.28.42.111"
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform output
aws_instance_public_ip = "3.28.42.111"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

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

```
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]
  availability_zone           = var.availability_zone
  associate_public_ip_address = true
  key_name                    = aws_key_pair.ssh_key.key_name

  connection {
    type        = "ssh"
    user        = "ec2-user"
    private_key = file(var.private_key)
    host        = self.public_ip
  }

  provisioner "file" {
    source = "./entry-script.sh"
    destination = "/home/ec2-user/entry-script-on-ec2.sh"
  }

  provisioner "remote-exec" {
    inline = [
      "sudo chmod +x /home/ec2-user/entry-script-on-ec2.sh",
      "sudo /home/ec2-user/entry-script-on-ec2.sh"
     ]
  }

  provisioner "local-exec" {
    command = <<-EOF
      echo Instance ${self.id} with public IP ${self.public_ip} has been create
    EOF
  }

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

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform apply -auto-approve
aws_instance.myapp_server (remote-exec):   Installing       : gperf [=     ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperf [==    ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperf [===   ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperf [====  ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperf [===== ] 4/7
aws_instance.myapp_server (remote-exec):   Installing       : gperftools-l   4/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [      ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [=     ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [==    ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [===   ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [====  ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [===== ] 5/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx-core-1   5/7
aws_instance.myapp_server (remote-exec):   Installing       : gener [      ] 6/7
aws_instance.myapp_server (remote-exec):   Installing       : gener [====  ] 6/7
aws_instance.myapp_server (remote-exec):   Installing       : gener [===== ] 6/7
aws_instance.myapp_server (remote-exec):   Installing       : generic-logo   6/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [      ] 7/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [==    ] 7/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [===   ] 7/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx [===== ] 7/7
aws_instance.myapp_server (remote-exec):   Installing       : nginx-1:1.28   7/7
aws_instance.myapp_server (remote-exec):   Running scriptlet: nginx-1:1.28   7/7
aws_instance.myapp_server (remote-exec):   Verifying        : generic-logo   1/7
aws_instance.myapp_server (remote-exec):   Verifying        : gperftools-l   2/7
aws_instance.myapp_server (remote-exec):   Verifying        : libunwind-1.   3/7
aws_instance.myapp_server (remote-exec):   Verifying        : nginx-1:1.28   4/7
aws_instance.myapp_server (remote-exec):   Verifying        : nginx-core-1   5/7
aws_instance.myapp_server (remote-exec):   Verifying        : nginx-filesy   6/7
aws_instance.myapp_server (remote-exec):   Verifying        : nginx-mimety   7/7

aws_instance.myapp_server (remote-exec): Installed:
aws_instance.myapp_server (remote-exec):   generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
aws_instance.myapp_server (remote-exec):   gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
aws_instance.myapp_server (remote-exec):   libunwind-1.4.0-5.amzn2023.0.3.x86_64
aws_instance.myapp_server (remote-exec):   nginx-1:1.28.0-1.amzn2023.0.2.x86_64
aws_instance.myapp_server (remote-exec):   nginx-core-1:1.28.0-1.amzn2023.0.2.x86_64
aws_instance.myapp_server (remote-exec):   nginx-filesystem-1:1.28.0-1.amzn2023.0.2.noarch
aws_instance.myapp_server (remote-exec):   nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

aws_instance.myapp_server (remote-exec): Complete!
aws_instance.myapp_server (remote-exec): Created symlink /etc/systemd/system/multi-user.target.wants/nginx.s
ervice.
aws_instance.myapp_server: Provisioning with 'local-exec'...
aws_instance.myapp_server (local-exec): Executing: ["/bin/sh" "-c" "echo Instance i-09a84c3ec508e1fcf with p
n"]
aws_instance.myapp_server (local-exec): Instance i-09a84c3ec508e1fcf with public IP 51.112.53.106 has been c
aws_instance.myapp_server: Creation complete after 33s [id=i-09a84c3ec508e1fcf]

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

Outputs:

aws_instance_public_ip = "51.112.53.106"
```

```
aws_instance_public_ip = "51.112.53.106"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform output
aws_instance_public_ip = "51.112.53.106"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

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.

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform destroy
      - id                                   = "vpc-0af2d2d4d707582fc" -> null
      - instance_tenancy                     = "default" -> null
      - ipv6_netmask_length                  = 0 -> null
      - main_route_table_id                  = "rtb-03ca519681b097704" -> null
      - owner_id                             = "443915509636" -> null
      - region                               = "me-central-1" -> null
      - tags                                 = {
          - "Name" = "dev-vpc"
        } -> null
      - tags_all                             = {
          - "Name" = "dev-vpc"
        } -> null
        # (4 unchanged attributes hidden)
    }

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

Changes to Outputs:
  - aws_instance_public_ip = "51.112.53.106" -> null

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_default_route_table.main_rt: Destroying... [id=rtb-03ca519681b097704]
aws_default_route_table.main_rt: Destruction complete after 0s
aws_instance.myapp_server: Destroying... [id=i-09a84c3ec508e1fcf]
aws_internet_gateway.myapp_igw: Destroying... [id=igw-05ddffe08dfc864bb]
aws_instance.myapp_server: Still destroying... [id=i-09a84c3ec508e1fcf, 00m10s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-05ddffe08dfc864bb, 00m10s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-09a84c3ec508e1fcf, 00m20s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-05ddffe08dfc864bb, 00m20s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-09a84c3ec508e1fcf, 00m30s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-05ddffe08dfc864bb, 00m30s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-09a84c3ec508e1fcf, 00m40s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-05ddffe08dfc864bb, 00m40s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-09a84c3ec508e1fcf, 00m50s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-05ddffe08dfc864bb, 00m50s elapsed]
aws_internet_gateway.myapp_igw: Destruction complete after 58s
aws_instance.myapp_server: Still destroying... [id=i-09a84c3ec508e1fcf, 01m00s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-09a84c3ec508e1fcf, 01m10s elapsed]
aws_instance.myapp_server: Destruction complete after 1m11s
aws_key_pair.ssh_key: Destroying... [id=serverkey]
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-0d5fe07a873a83e5b]
aws_default_security_group.default_sg: Destroying... [id=sg-023402b63a85efbb9]
aws_default_security_group.default_sg: Destruction complete after 0s
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-0af2d2d4d707582fc]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

```
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]
  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"
  }
}
~
```

Task 4 — Create Terraform modules (subnet module)

```
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ mkdir -p modules/subnet
touch modules/subnet/main.tf
touch modules/subnet/variables.tf
touch modules/subnet/outputs.tf
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ ls -R
.:
entry-script.sh  locals.tf  main.tf  modules  outputs.tf  terraform.tfstate  terraform.tfstate.backup  terraform.tfvars  variables.tf  yes  yes.pub

./modules:
subnet

./modules/subnet:
main.tf  outputs.tf  variables.tf
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

```
variable "vpc_id" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "default_route_table_id" {}
~
~
~
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 4

```
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_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"
  }
}

resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = var.vpc_id
    tags = {
     Name = "${var.env_prefix}-igw"
  }
}
~
```

```
output "subnet" {
  value = aws_subnet.myapp_subnet_1
}
~
```

```
resource "aws_instance" "myapp_server" {
  ami                         = "ami-05524d6658fcf35b6"
  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

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


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

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ 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
- Reusing previous version of hashicorp/http from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0
- 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.
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform apply -auto-approve
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform apply -auto-approve
  # module.myapp-subnet.aws_subnet.myapp_subnet_1 will be created
  + resource "aws_subnet" "myapp_subnet_1" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "me-central-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.10.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "me-central-1"
      + tags                                           = {
          + "Name" = "dev-subnet-1"
        }
      + tags_all                                       = {
          + "Name" = "dev-subnet-1"
        }
      + vpc_id                                         = (known after apply)
    }

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-0f79ef9dd2cb76fc4]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creating...
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creating...
aws_default_security_group.default_sg: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creation complete after 0s [id=igw-09dad83b66b8a9173]
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-0297eb4e0fbc79acf]
aws_default_security_group.default_sg: Creation complete after 3s [id=sg-00d304253a1d7c3a7]
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-05ab509aed8abee75]
aws_instance.myapp_server: Creating...
aws_instance.myapp_server: Still creating... [00m10s elapsed]
aws_instance.myapp_server: Creation complete after 12s [id=i-008f7b0a1b8ae5810]

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

Outputs:

aws_instance_public_ip = "51.112.42.119"
```

```
aws_instance_public_ip = "51.112.42.119"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform output
aws_instance_public_ip = "51.112.42.119"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

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

```
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ tree
.
├── entry-script.sh
├── locals.tf
├── main.tf
├── modules
│   ├── subnet
│   │   ├── main.tf
│   │   ├── outputs.tf
│   │   └── variables.tf
│   └── webserver
│       ├── main.tf
│       ├── outputs.tf
│       └── variables.tf
├── outputs.tf
├── terraform.tfstate
├── terraform.tfstate.backup
├── terraform.tfvars
├── variables.tf
├── yes
└── yes.pub

4 directories, 16 files
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

```
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" {}
~
~
~
```

```
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"]
    prefix_list_ids = []
}
  tags = {
     Name = "${var.env_prefix}-default-sg"
  }
}

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 Kernel 6.1 AMI
  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}"
  }
}
:wq
```

```
output "aws_instance" {
  value = aws_instance.myapp_server
}
~
~
~
```

```
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"
  }
}


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
}

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"
}
~
```

```
output "webserver_public_ip" {
  value = module.myapp-webserver.aws_instance.public_ip
}:wq
~
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform init
Initializing the backend...
Initializing modules...
- myapp-webserver in modules/webserver
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/aws v6.27.0
- 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.
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform apply -auto-approve
          + {
              + cidr_blocks      = [
                  + "4.240.18.225/32",
                ]
              + from_port        = 22
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 22
                # (1 unchanged attribute hidden)
            },
        ]
      + name                   = "dev-web-sg-0"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + region                 = "me-central-1"
      + revoke_rules_on_delete = false
      + tags                   = {
          + "Name" = "dev-default-sg"
        }
      + tags_all               = {
          + "Name" = "dev-default-sg"
        }
      + vpc_id                 = "vpc-0f79ef9dd2cb76fc4"
    }

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

Changes to Outputs:
  - aws_instance_public_ip = "51.112.42.119" -> null
  + webserver_public_ip    = (known after apply)
aws_instance.myapp_server: Destroying... [id=i-008f7b0a1b8ae5810]
module.myapp-webserver.aws_key_pair.ssh-key: Creating...
module.myapp-webserver.aws_security_group.web_sg: Creating...
module.myapp-webserver.aws_key_pair.ssh-key: Creation complete after 1s [id=dev-serverkey-0]
module.myapp-webserver.aws_security_group.web_sg: Creation complete after 3s [id=sg-00af055874a58a2d3
module.myapp-webserver.aws_instance.myapp-server: Creating...
aws_instance.myapp_server: Still destroying... [id=i-008f7b0a1b8ae5810, 00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Creation complete after 13s [id=i-04eb9d1e3cab560b9
aws_instance.myapp_server: Still destroying... [id=i-008f7b0a1b8ae5810, 00m20s elapsed]
aws_instance.myapp_server: Destruction complete after 30s
aws_key_pair.ssh_key: Destroying... [id=serverkey]
aws_default_security_group.default_sg: Destroying... [id=sg-00d304253a1d7c3a7]
aws_default_security_group.default_sg: Destruction complete after 0s
aws_key_pair.ssh_key: Destruction complete after 0s

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

Outputs:

webserver_public_ip = "51.112.178.50"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

```
  webserver_public_ip = "51.112.178.50"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform output
  webserver_public_ip = "51.112.178.50"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

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.

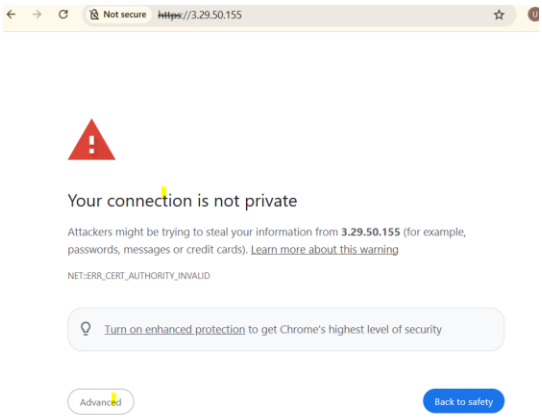
Thank you for using nginx.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS 4                                  bash - LA

@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform destroy
              - security_groups  = []
              - self             = false
              - to_port          = 22
                # (1 unchanged attribute hidden)
            },
        ] -> null
      - name                   = "dev-web-sg-0" -> null
      - owner_id               = "443915509636" -> null
      - region                 = "me-central-1" -> null
      - revoke_rules_on_delete = false -> null
      - tags                   = {
          - "Name" = "dev-default-sg"
        } -> null
      - tags_all               = {
          - "Name" = "dev-default-sg"
        } -> null
      - vpc_id                 = "vpc-0f79ef9dd2cb76fc4" -> null
        # (1 unchanged attribute hidden)
    }

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

Changes to Outputs:
  - webserver_public_ip = "51.112.178.50" -> null

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

module.myapp-subnet.aws_default_route_table.main_rt: Destroying... [id=rtb-0297eb4e0fbc79acf]
module.myapp-subnet.aws_default_route_table.main_rt: Destruction complete after 0s
module.myapp-webserver.aws_instance.myapp-server: Destroying... [id=i-04eb9d1e3cab560b9]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destroying... [id=igw-09dad83b66b8a9173]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-04eb9d1e3cab560b9, 00m10s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-09dad83b66b8a9173, 00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-04eb9d1e3cab560b9, 00m20s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-09dad83b66b8a9173, 00m20s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-04eb9d1e3cab560b9, 00m30s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-09dad83b66b8a9173, 00m30s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destruction complete after 37s
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-04eb9d1e3cab560b9, 00m40s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Destruction complete after 40s
module.myapp-webserver.aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-0]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destroying... [id=subnet-05ab509aed8abee75]
module.myapp-webserver.aws_security_group.web_sg: Destroying... [id=sg-00af055874a58a2d3]
module.myapp-webserver.aws_key_pair.ssh-key: Destruction complete after 1s
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-0f79ef9dd2cb76fc4]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

Task 6 — Configure HTTPS with self-signed certificates

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

events {
    worker_connections 1024;
}

http {
    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;
    default_type        application/octet-stream;

    upstream backend_servers {
        server 158.252.94.241:80;
        server 158.252.94.242:80 backup;
    }

    server {
        listen 443 ssl;
        server_name $PUBLIC_IP;
        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://158.252.94.241:80;
    #       proxy_pass http://backend_servers;

        }
    }

    server {
        listen 80;
        server_name _;
        return 301 https://\$host\$request_uri;
    }
}
EOF

# Test and restart Nginx
systemctl restart nginx
:wq
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform apply -auto-approve
                  + "4.240.18.225/32",
                ]
              + from_port        = 22
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 22
                # (1 unchanged attribute hidden)
            },
        ]
      + name                   = "dev-web-sg-0"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + region                 = "me-central-1"
      + revoke_rules_on_delete = false
      + tags                   = {
          + "Name" = "dev-default-sg"
        }
      + tags_all               = {
          + "Name" = "dev-default-sg"
        }
      + vpc_id                 = (known after apply)
    }

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

Changes to Outputs:
  + webserver_public_ip = (known after apply)
module.myapp-webserver.aws_key_pair.ssh-key: Creating...
aws_vpc.myapp_vpc: Creating...
module.myapp-webserver.aws_key_pair.ssh-key: Creation complete after 0s [id=dev-serverkey-0]
aws_vpc.myapp_vpc: Creation complete after 1s [id=vpc-0a1cc1ad3499cd631]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creating...
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creating...
module.myapp-webserver.aws_security_group.web_sg: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creation complete after 1s [id=igw-006ef71cff13aeea4]
module.myapp-subnet.aws_default_route_table.main_rt: Creating...
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 0s [id=rtb-0b730fea8efb53759]
module.myapp-webserver.aws_security_group.web_sg: Creation complete after 3s [id=sg-0cf32b6094365c265]
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-0d98bec772a349c9f]
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 13s [id=i-0a452e5fde50e0206]

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

Outputs:

webserver_public_ip = "3.29.50.155"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform output
web1_public_ip = "51.112.52.212"
webserver_public_ip = "3.29.50.155"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```



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

```
#!/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
echo "<h2>Hostname:  $(hostname)</h2>" >> /var/www/html/index.html
TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" \
  -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
echo "<h2>Private IP: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN"  http://169.254.169.2
54/latest/meta-data/local-ipv4)</h2>" >> /var/www/html/index.html
echo "<h2>Public IP: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN"  http://169.254.169.25
4/latest/meta-data/public-ipv4)</h2>" >> /var/www/html/index.html
echo "<h2>Public DNS: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN"  http://169.254.169.2
54/latest/meta-data/public-hostname)</h2>" >> /var/www/html/index.html
echo "<h2>Deployed via Terraform</h2>" >> /var/www/html/index. html
~
~
~
```

```
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"
}
```

```
output "webserver_public_ip" {
  value = module.myapp-webserver.public_ip
}

output "web1_public_ip" {
  value = module.myapp-web-1.public_ip
}
~
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform apply -auto-approve
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
module.myapp-webserver.aws_key_pair.ssh-key: Refreshing state... [id=dev-serverkey-0]
module.myapp-web-1.aws_key_pair.ssh-key: Refreshing state... [id=dev-serverkey-1]
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0a1cc1ad3499cd631]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-006ef71cff13aeea4]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0d98bec772a349c9f]
module.myapp-webserver.aws_security_group.web_sg: Refreshing state... [id=sg-0cf32b6094365c265]
module.myapp-web-1.aws_security_group.web_sg: Refreshing state... [id=sg-0c2b2f21a814ec585]
module.myapp-subnet.aws_default_route_table.main_rt: Refreshing state... [id=rtb-0b730fea8efb53759]
module.myapp-webserver.aws_instance.myapp-server: Refreshing state... [id=i-0a452e5fde50e0206]
module.myapp-web-1.aws_instance.myapp-server: Refreshing state... [id=i-01d592ffe86310339]

Changes to Outputs:
  - aws_web-1_public_ip = "51.112.52.212" -> null
  + web1_public_ip      = "51.112.52.212"
  + webserver_public_ip = "3.29.50.155"

You can apply this plan to save these new output values to the Terraform state, without changing any real infrastructure.

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

Outputs:

web1_public_ip = "51.112.52.212"
webserver_public_ip = "3.29.50.155"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform output
```

```
webserver_public_ip = "3.29.50.155"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform output
web1_public_ip = "51.112.52.212"
webserver_public_ip = "3.29.50.155"
```

```
    ssl_certificate_key /etc/ssl/private/selfsigned.key;

        location / {
      #  root /usr/share/nginx/html;
       # index index.html;
proxy_pass http://51.112.52.212:80;
 #       proxy_pass http://backend_servers;

    }
 }
 server {
    listen 80;
    server_name _;
    return 301 https://$host$request_uri;
 }
```

```
[ec2-user@ip-10-0-10-149 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-149 ~]$
```

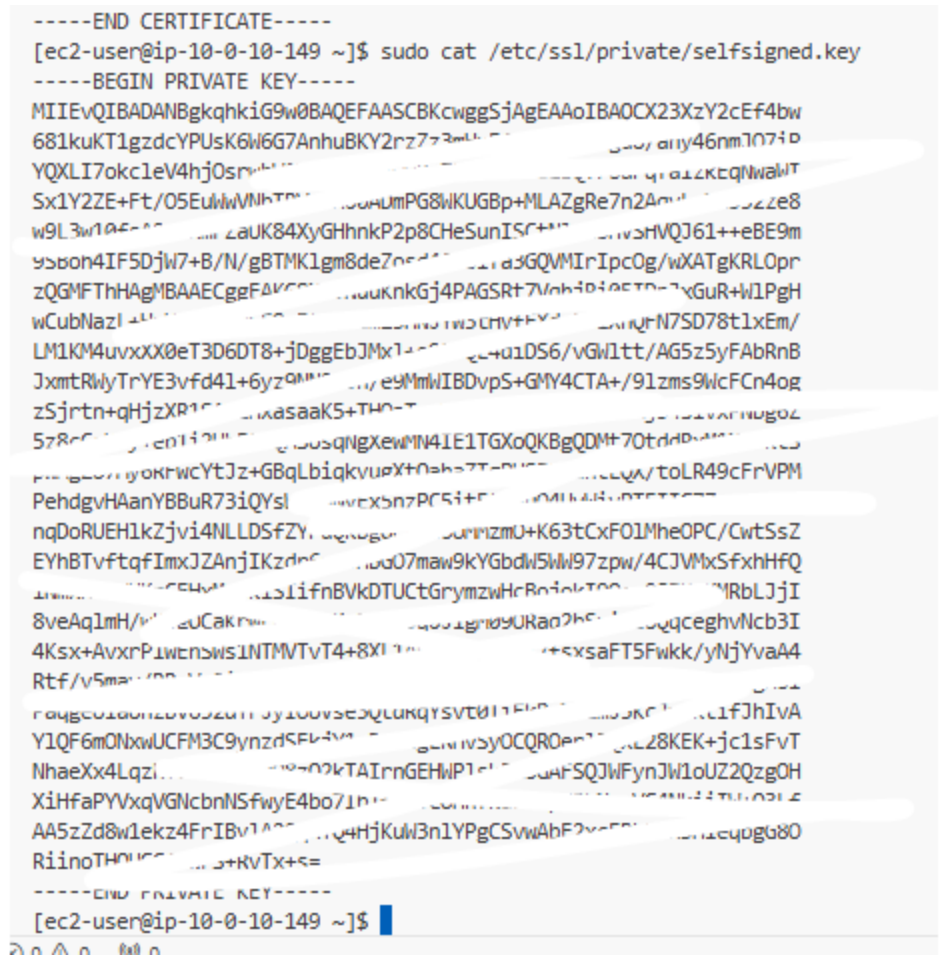
```
2026/01/06 15:25:10 [notice] 1936#1936: exit
2026/01/06 15:25:10 [notice] 1935#1935: signal 17 (SIGCHLD) received from 1936
2026/01/06 15:25:10 [notice] 1935#1935: worker process 1936 exited with code 0
2026/01/06 15:25:10 [notice] 1935#1935: signal 29 (SIGIO) received
2026/01/06 15:25:10 [notice] 1935#1935: signal 17 (SIGCHLD) received from 1937
2026/01/06 15:25:10 [notice] 1935#1935: worker process 1937 exited with code 0
2026/01/06 15:25:10 [notice] 1935#1935: exit
2026/01/06 15:26:21 [notice] 1574#1574: using the "epoll" event method
2026/01/06 15:26:21 [notice] 1574#1574: nginx/1.28.0
2026/01/06 15:26:21 [notice] 1574#1574: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2026/01/06 15:26:21 [notice] 1574#1574: getrlimit(RLIMIT_NOFILE): 65535:65535
2026/01/06 15:26:21 [notice] 1578#1578: start worker processes
2026/01/06 15:26:21 [notice] 1578#1578: start worker process 1579
2026/01/06 15:26:21 [notice] 1578#1578: start worker process 1580
2026/01/06 16:52:40 [error] 1580#1580: *21 open() "/usr/share/nginx/html/autodiscover/autodiscover.json" failed (2: No su
tory), client: 40.76.125.17, server: 40.172.122.60, request: "GET /autodiscover/autodiscover.json?@zdi/Powershell HTTP/1.
50.155"
2026/01/06 16:55:30 [error] 1580#1580: *23 open() "/usr/share/nginx/html/Core/Skin/Login.aspx" failed (2: No such file or
ent: 101.32.218.31, server: 40.172.122.60, request: "HEAD /Core/Skin/Login.aspx HTTP/1.1", host: "3.29.50.155", referrer:
.155:80/Core/Skin/Login.aspx"
2026/01/06 17:59:25 [error] 1580#1580: *34 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or director
184.76.28, server: 40.172.122.60, request: "GET /favicon.ico HTTP/1.1", host: "3.29.50.155"
2026/01/06 18:44:14 [error] 1580#1580: *39 open() "/usr/share/nginx/html/+CSCOL+/Java.jar" failed (2: No such file or dir
 162.243.20.133, server: 40.172.122.60, request: "GET /+CSCOL+/Java.jar HTTP/1.1", host: "3.29.50.155"
2026/01/06 18:44:16 [error] 1580#1580: *40 open() "/usr/share/nginx/html/+CSCOE+/logon_forms.js" failed (2: No such file
lient: 162.243.20.133, server: 40.172.122.60, request: "GET /+CSCOE+/logon_forms.js HTTP/1.1", host: "3.29.50.155"
2026/01/06 18:44:17 [error] 1580#1580: *41 open() "/usr/share/nginx/html/+CSCOL+/a1.jar" failed (2: No such file or direc
62.243.20.133, server: 40.172.122.60, request: "GET /+CSCOL+/a1.jar HTTP/1.1", host: "3.29.50.155"
2026/01/06 18:44:18 [error] 1579#1579: *42 open() "/usr/share/nginx/html/+CSCOE+/transfer.js" failed (2: No such file or
nt: 162.243.20.133, server: 40.172.122.60, request: "GET /+CSCOE+/transfer.js HTTP/1.1", host: "3.29.50.155"
2026/01/06 18:51:38 [error] 1579#1579: *43 open() "/usr/share/nginx/html/admin/config.php" failed (2: No such file or dir
 102.22.20.125, server: 40.172.122.60, request: "GET /admin/config.php HTTP/1.0", host: "0.0.0.0"
2026/01/06 19:43:29 [error] 1579#1579: *56 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or director
3.167.238, server: 40.172.122.60, request: "GET /favicon.ico HTTP/1.1", host: "3.29.50.155", referrer: "https://3.29.50.1
2026/01/06 19:51:24 [notice] 1578#1578: signal 3 (SIGQUIT) received from 1, shutting down
2026/01/06 19:51:24 [notice] 1580#1580: gracefully shutting down
2026/01/06 19:51:24 [notice] 1579#1579: gracefully shutting down
2026/01/06 19:51:24 [notice] 1580#1580: exiting
2026/01/06 19:51:24 [notice] 1579#1579: exiting
2026/01/06 19:51:24 [notice] 1580#1580: exit
2026/01/06 19:51:24 [notice] 1579#1579: exit
2026/01/06 19:51:24 [notice] 1578#1578: signal 17 (SIGCHLD) received from 1579
2026/01/06 19:51:24 [notice] 1578#1578: worker process 1579 exited with code 0
2026/01/06 19:51:24 [notice] 1578#1578: signal 29 (SIGIO) received
2026/01/06 19:51:24 [notice] 1578#1578: signal 17 (SIGCHLD) received from 1580
2026/01/06 19:51:24 [notice] 1578#1578: worker process 1580 exited with code 0
2026/01/06 19:51:24 [notice] 1578#1578: exit
2026/01/06 19:51:24 [notice] 8310#8310: using the "epoll" event method
2026/01/06 19:51:24 [notice] 8310#8310: nginx/1.28.0
2026/01/06 19:51:24 [notice] 8310#8310: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2026/01/06 19:51:24 [notice] 8310#8310: getrlimit(RLIMIT_NOFILE): 65535:65535
2026/01/06 19:51:24 [notice] 8311#8311: start worker processes
2026/01/06 19:51:24 [notice] 8311#8311: start worker process 8312
2026/01/06 19:51:24 [notice] 8311#8311: start worker process 8313
```

```
180.76.172.156 - - [06/Jan/2026:15:23:04 +0000] "GET /app/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.1"404 15
ail-http" "-"
180.76.172.156 - - [06/Jan/2026:15:23:15 +0000] "GET /index.php?s=/index/\x5Cthink\x5Capp/invokefunction&function=call_user_
rs[0]=md5&vars[1][]=Hello HTTP/1.1"404 153 "-""libredtail-http" "-"
180.76.172.156 - - [06/Jan/2026:15:23:16 +0000] "GET /public/index.php?s=/index/\x5Cthink\x5Capp/invokefunction&function=cal
rray&vars[0]=md5&vars[1][]=Hello HTTP/1.1"404 153 "-""libredtail-http" "-"
180.76.172.156 - - [06/Jan/2026:15:23:17 +0000] "GET /index.php?lang=../../../../../../../../usr/local/lib/php/pearcmd&+conf
<?echo(md5(\x22hi\x22));?>+/tmp/index1.php HTTP/1.1"404 153 "-""libredtail-http" "-"
180.76.172.156 - - [06/Jan/2026:15:23:17 +0000] "GET /index.php?lang=../../../../../../../../tmp/index1 HTTP/1.1"404 153 "-"
ttp" "-"
180.76.172.156 - - [06/Jan/2026:15:23:17 +0000] "GET /containers/json HTTP/1.1"404 153 "-""libredtail-http" "-"
64.227.97.195 - - [06/Jan/2026:15:27:59 +0000] "SSTP_DUPLEX_POST /sra_{BA195980-CD49-458b-9E23-C84EE0ADCD75}/ HTTP/1.1"400 1
20.83.49.34 - - [06/Jan/2026:15:34:43 +0000] "GET / HTTP/1.1"200 615 "-""Mozilla/5.0 zgrab/0.x" "-"
20.83.49.34 - - [06/Jan/2026:15:34:44 +0000] "MGLNDD_3.29.50.155_443"400 157 "-""-" "-"
138.68.84.216 - - [06/Jan/2026:15:57:42 +0000] "\x00\x0E8\xDC\x12V\xA1)Z\xAC\x83\x00\x00\x00\x00\x00"400 157 "-""-" "-"
138.68.84.216 - - [06/Jan/2026:15:57:43 +0000] "\x00\x0E8\xF0\x83\xA4$$\xD0\xEA\x88\x00\x00\x00\x00\x00"400 157 "-""-" "-"
87.236.176.192 - - [06/Jan/2026:16:37:14 +0000] "GET / HTTP/1.1"200 615 "-""Mozilla/5.0 (compatible; InternetMeasurement/1.0
ternet-measurement.com/)" "-"
40.76.125.17 - - [06/Jan/2026:16:52:40 +0000] "GET /autodiscover/autodiscover.json?@zdi/Powershell HTTP/1.1"404 153 "-""Mozi
/0.x" "-"
101.32.218.31 - - [06/Jan/2026:16:55:29 +0000] "HEAD /Core/Skin/Login.aspx HTTP/1.1"301 0 "-""Mozilla/5.0 (Windows NT 10.0;
ppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36" "-"
101.32.218.31 - - [06/Jan/2026:16:55:30 +0000] "HEAD /Core/Skin/Login.aspx HTTP/1.1"404 0 "http://3.29.50.155:80/Core/Skin/L
zilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36" "-"
185.242.226.111 - - [06/Jan/2026:17:48:07 +0000] "GET / HTTP/1.1"200 615 "-""Mozilla/5.0 (Windows NT 10.0; Win64; x64) Apple
 (KHTML, like Gecko) Chrome/88.0.4324.190 Safari/537.36" "-"
198.235.24.26 - - [06/Jan/2026:17:54:43 +0000] "GET / HTTP/1.1"301 169 "-""Hello from Palo Alto Networks, find out more abou
n https://docs-cortex.paloaltonetworks.com/r/1/Cortex-Xpanse/Scanning-activity" "-"
198.235.24.26 - - [06/Jan/2026:17:54:44 +0000] "GET / HTTP/1.1"200 615 "http://3.29.50.155:80/""Hello from Palo Alto Network
ore about our scans in https://docs-cortex.paloaltonetworks.com/r/1/Cortex-Xpanse/Scanning-activity" "-"
91.230.168.79 - - [06/Jan/2026:17:57:04 +0000] "GET / HTTP/1.1"200 615 "-""Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:134.0)
01 Firefox/134.0" "-"
195.184.76.28 - - [06/Jan/2026:17:59:25 +0000] "GET /favicon.ico HTTP/1.1"404 153 "-""Mozilla/5.0 (X11; Ubuntu; Linux x86_64
ecko/20100101 Firefox/134.0" "-"
102.22.20.125 - - [06/Jan/2026:18:28:42 +0000] "GET /admin/config.php HTTP/1.0"301 169 "-""xfa1,nvdorz,nvd0rz" "-"
162.243.20.133 - - [06/Jan/2026:18:44:14 +0000] "GET /+CSCOL+/Java.jar HTTP/1.1"404 555 "-""Mozilla/5.0 (Windows NT 10.0; Wi
leWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36" "-"
162.243.20.133 - - [06/Jan/2026:18:44:16 +0000] "GET /+CSCOE+/logon_forms.js HTTP/1.1"404 555 "-""Mozilla/5.0 (Windows NT 10
4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36" "-"
162.243.20.133 - - [06/Jan/2026:18:44:17 +0000] "GET /+CSCOL+/a1.jar HTTP/1.1"404 555 "-""Mozilla/5.0 (Windows NT 10.0; Win6
WebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36" "-"
162.243.20.133 - - [06/Jan/2026:18:44:18 +0000] "GET /+CSCOE+/transfer.js HTTP/1.1"404 555 "-""Mozilla/5.0 (Windows NT 10.0;
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36" "-"
102.22.20.125 - - [06/Jan/2026:18:51:38 +0000] "GET /admin/config.php HTTP/1.0"404 153 "-""xfa1,nvdorz,nvd0rz" "-"
60.163.213.55 - - [06/Jan/2026:19:18:36 +0000] "CONNECT www.baidu.com:443 HTTP/1.1"400 157 "-""-" "-"
60.163.213.55 - - [06/Jan/2026:19:18:36 +0000] "CONNECT www.baidu.com:443 HTTP/1.1"400 157 "-""-" "-"
198.235.24.222 - - [06/Jan/2026:19:32:25 +0000] "GET / HTTP/1.1"200 615 "-""Hello from Palo Alto Networks, find out more abo
in https://docs-cortex.paloaltonetworks.com/r/1/Cortex-Xpanse/Scanning-activity" "-"
39.33.167.238 - - [06/Jan/2026:19:43:01 +0000] "GET / HTTP/1.1"301 169 "-""Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
39.33.167.238 - - [06/Jan/2026:19:43:29 +0000] "GET / HTTP/1.1"200 615 "-""Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
39.33.167.238 - - [06/Jan/2026:19:43:29 +0000] "GET /favicon.ico HTTP/1.1"404 555 "https://3.29.50.155/""Mozilla/5.0 (Window
```

```
chemical/x-xyz                          xyz;
image/webp                              webp;
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@ip-10-0-10-149 ~]$
```

⊗ 0 ⚠ 0 0

```
[ec2-user@ip-10-0-10-149 ~]$ cat /etc/ssl/certs/selfsigned.crt
-----BEGIN CERTIFICATE-----
MIIDPzCCAiegAwIBAgIUZmXmRpXP5OoscnoGEMu0GIQ2LUgwDQYJKoZIhvcNAQEL
BQAwGDEWMBQGA1UEAwwNNDAuMTcyLjEyMi42MDAeFw0yNjAxMDYxNDM0NDJaFw0y
NzAxMDYxNDM0NDJaMBgxFjAUBgNVBAMMDTQwLjE3Mi4xMjIuNjAwggEiMA0GCSqG
SIb3DQEBAQU[illegible]
G7AnhuBKY2rzZz3mHxEf3mJ[illegible]
W1VMe25Yx1WvBpZL9k611[illegible]
TRVprH00ADmPG8WKUGBp+MLAZgRe7n2AqvL+NLD32ze8w9L3w10[illegible]
84XyGHhnkP2p8CHeSunISCtNlrzbHVSHVQJ61++eBE9m9SBoh4IF5DjW7+B/N/gB
TMKlgm8deZosd4C2oifa3GQ[illegible]
fjAdBgNVHQ4EFgQUgnGwriRffcOAs8u3sJ5AIshHUyAwHwYDVR0[illegible]
riRffcOAs8u3sJ5AIshHUyAwDwYDVR0RBAgwBocEKKx6PDAJBgNVHRMEAjAAMAsG
A1UdDwQEAwIFoDATBgNVHSUEDDAKBggrBgEFBQcDATANBgkqhkiG9w0BAQsFAAOC
AQEAg5A7Mz18E67ol/DP4q7ED15m2ygW1CBJcOA3D9SrKeZdpIAjuy+FmTs8wP9s
kiK5VFxOCP4d6X2lzspmno8GwHi/6sUj3GRS4W3orowoj12meOq031JRGwSkcbY0
7KCwWuDd5wo6hZ9SfnGa1HORf+S6mcY7SViBqiYL65DN2LjUHQsJyZggnQFjM4LE
8M6XEyM0IEB1rx3v+lXKd7oioxsQJfBGonx2lI02/Ag16Bi5ANmPi/ta1G08yJpb
2oGJdH6MHM1LMXUSB8NC0E6XrpONgWnFviZP8MVmabznXV8OhsoZvnx9tWHGeeDl
/cUcR6gw0uFY4bqE613/sqBa7A==
-----END CERTIFICATE-----
[ec2-user@ip-10-0-10-149 ~]$
```



Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.0.10.236

Public IP: 51.112.52.212

Public DNS:

Task 8 — Configure Nginx as load balancer

```
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"
  inst nce_suffix = "2"
}
```

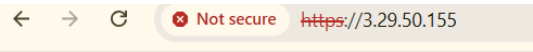
```
  output "webserver_public_ip" {
    value = module.myapp-webserver.public_ip
  }

  output "web1_public_ip" {
    value = module.myapp-web-1.public_ip
  }

  output "aws_web-2_public_ip" {
    value = module. myapp-web-2.public_ip
  }
  ~
  ~
  ~
  ~
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform apply -auto-approve
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 80
              # (1 unchanged attribute hidden)
          },
        + {
            + cidr_blocks      = [
              + "4.240.39.200/32",
              ]
            + from_port        = 22
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 22
              # (1 unchanged attribute hidden)
          },
      ]
    + name                   = "dev-web-sg-2"
    + name_prefix            = (known after apply)
    + owner_id               = (known after apply)
    + region                 = "me-central-1"
    + revoke_rules_on_delete = false
    + tags                   = {
        + "Name" = "dev-default-sg"
      }
    + tags_all               = {
        + "Name" = "dev-default-sg"
      }
    + vpc_id                 = "vpc-0a1cc1ad3499cd631"
  }

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

Changes to Outputs:
  + aws_web-2_public_ip = (known after apply)
module.myapp-web-2.aws_key_pair.ssh-key: Creating...
module.myapp-web-2.aws_security_group.web_sg: Creating...
module.myapp-web-2.aws_key_pair.ssh-key: Creation complete after 1s [id=dev-serverkey-2]
module.myapp-web-2.aws_security_group.web_sg: Creation complete after 3s [id=sg-06a5382ddd14e
module.myapp-web-2.aws_instance.myapp-server: Creating...
module.myapp-web-2.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Creation complete after 13s [id=i-0411fb648bdf7

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

Outputs:

aws_web-2_public_ip = "51.112.46.108"
web1_public_ip = "51.112.52.212"
webserver_publi _ip = "3.29.50.155"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

```
web1_public_ip = "51.112.52.212"
webserver_public_ip = "3.29.50.155"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ vim outputs.tf
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ terraform output
aws_web-2_public_ip = "51.112.46.108"
web1_public_ip = "51.112.52.212"
webserver_public_ip = "3.29.50.155"
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

```
    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;
    default_type        application/octet-stream;

    upstream backend_servers {
        server 51.112.46.108:80;
        server 51.112.52.212:80;
    }

    server {
        listen 443 ssl;
        server_name 40.172.122.60;
        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://51.112.52.212:80;
    #       proxy_pass http://backend_servers;

        }
    }
    server {
        listen 80;
        server_name _;
        return 301 https://$host$request_uri;
    }
}
```

```
[ec2-user@ip-10-0-10-149 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-149 ~]$
```



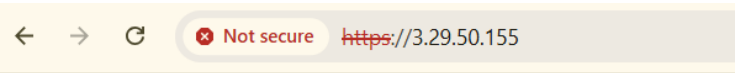
Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.0.10.236

Public IP: 51.112.52.212

Public DNS:



Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.0.10.130

Public IP: 51.112.46.108

Public DNS:

Task 9 — Configure high availability with backup servers

```
    default_type        application/octet-stream;

    upstream backend_servers {
        server 51.112.46.108:80;
        server 51.112.52.212:80 backup;
    }

    server {
        listen 443 ssl;
        server_name 40.172.122.60;
        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://51.112.52.212:80;
            proxy_pass http://backend_servers;

        }
    }
    server {
```

Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.0.10.130

Public IP: 51.112.46.108

Public DNS:

```
upstream backend_servers {
    server 51.112.46.108:80 backup;
    server 51.112.52.212:80;
}

server {
    listen 443 ssl;
    server_name 40.172.122.60;
    ssl_certificate /etc/ssl/certs/selfsigned.crt;
```

Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.0.10.236

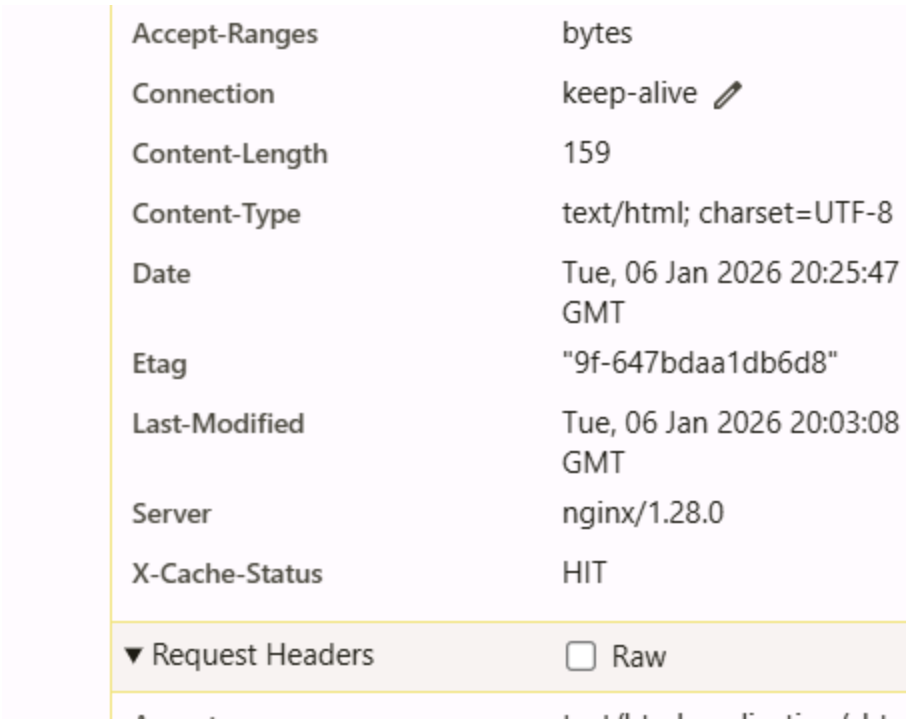
Public IP: 51.112.52.212

Public DNS:

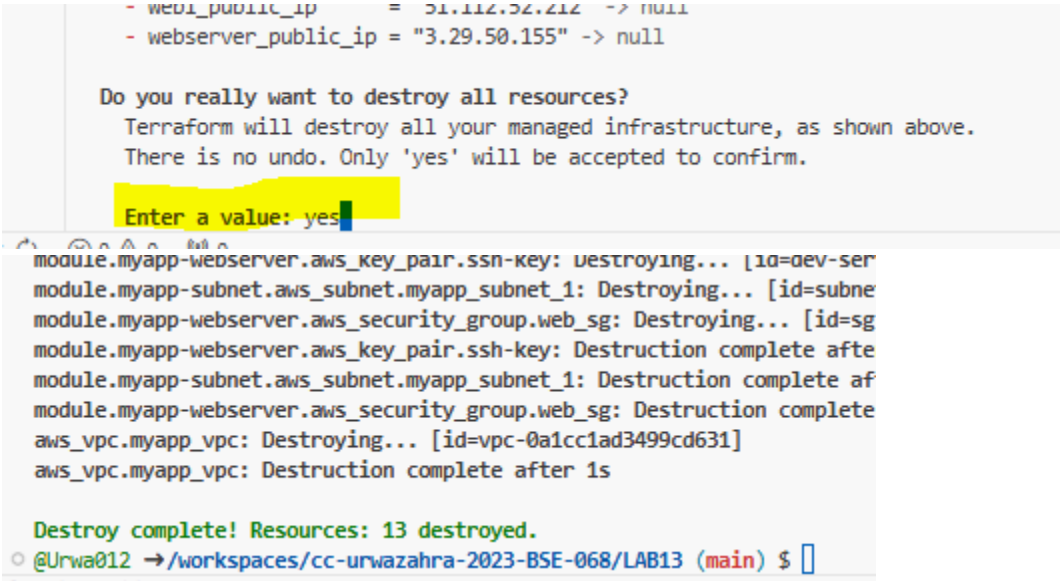
Task 10 — Enable Nginx caching

```
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"';

    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;
    default_type        application/octet-stream;

    upstream backend_servers {
        server 51.112.46.108:80;
        server 51.112.52.212:80;
    }

    server {
        listen 443 ssl;
        server_name 40.172.122.60;
        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://51.112.52.212:80;
           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;
    }
}
~
```

```
[ec2-user@ip-10-0-10-149 ~]$ sudo vim /etc/nginx/nginx.conf
[ec2-user@ip-10-0-10-149 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-149 ~]$
```


Accept-Ranges bytes
Connection keep-alive
Content-Length 159
Content-Type text/html; charset=UTF-8
Date Tue, 06 Jan 2026 20:25:47 GMT
Etag "9f-647bdaa1db6d8"
Last-Modified Tue, 06 Jan 2026 20:03:08 GMT
Server nginx/1.28.0
X-Cache-Status HIT
▼ Request Headers
Raw

Cleanup:


- webserver_public_ip = "3.29.50.155" -> null
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.
Enter a value: yes
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destroying... [id=subne
module.myapp-webserver.aws_security_group.web_sg: Destroying... [id=sg
module.myapp-webserver.aws_key_pair.ssh-key: Destruction complete afte
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete af
module.myapp-webserver.aws_security_group.web_sg: Destruction complete
aws_vpc.myapp_vpc: Destroying... [id=vpc-0a1cc1ad3499cd631]
aws_vpc.myapp_vpc: Destruction complete after 1s
Destroy complete! Resources: 13 destroyed.
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 109,
  "lineage": "1a4ffb4d-1fe5-6a69-68b1-9031e08e44c3",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```

```
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $ tree
.
├── apache.sh
├── entry-script.sh
├── locals.tf
├── main.tf
├── modules
│   ├── subnet
│   │   ├── main.tf
│   │   ├── outputs.tf
│   │   └── variables.tf
│   └── webserver
│       ├── main.tf
│       ├── outputs.tf
│       └── variables.tf
├── outputs.tf
├── terraform.tfstate
├── terraform.tfstate.backup
├── terraform.tfvars
├── variables.tf
├── yes
└── yes.pub

4 directories, 17 files
@Urwa012 ➜/workspaces/cc-urwazahra-2023-BSE-068/LAB13 (main) $
```