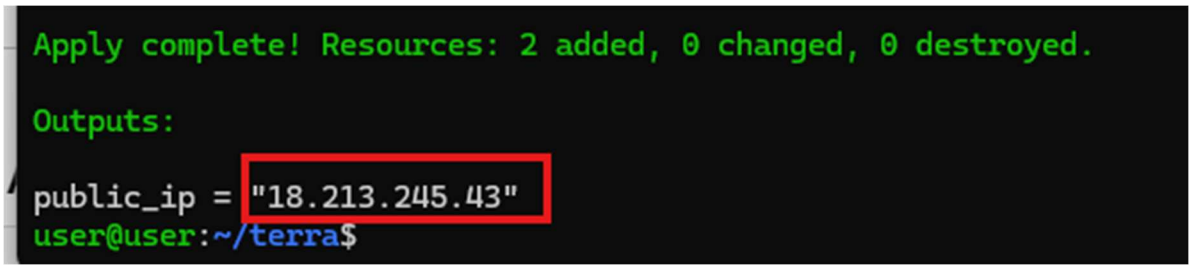
Terraform Assignment – 5

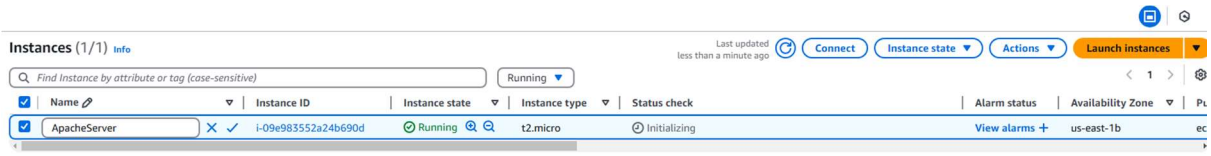
user@user: ~/terra

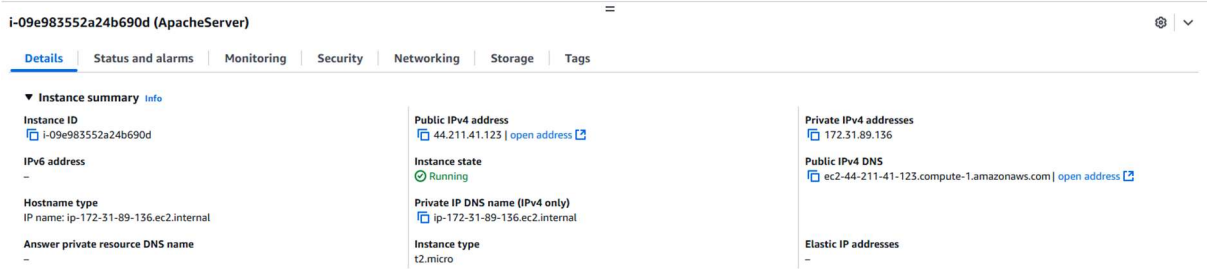
```
  GNU nano 6.2
provider "aws" {
  region = "us-east-1"  # Change to your preferred region
  access_key = "AKIA4MTWGW7C33ESVYP3"
  secret_key  = "v6YxMMlvT+8GXtAZPasiyIjsRz4y4k4K5z+P4SmA"
}

resource "aws_instance" "apache" {
  ami           = "ami-04b4f1a9cf54c11d0"
  instance_type = "t2.micro"
  tags = {
    Name = "ApacheServer"
  }

  user_data = <<-EOF
              #!/bin/bash
              sudo apt update -y
              sudo apt install apache2 -y
              sudo systemctl start apache2
              sudo systemctl enable apache2
              EOF

  security_groups = [aws_security_group.allow_http.name]
}

resource "aws_security_group" "allow_http" {
  name        = "allow_http"
  description = "Allow HTTP inbound traffic"

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

output "instance_ip" {
  value = aws-instance.apache.public_ip
}

File Name to Write: demo.tf
```

```
user@user:~/terra$ terraform apply

Terraform used the selected providers to generate the following executic
indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.apache will be created
  + resource "aws_instance" "apache" {
      + ami                                  = "ami-04b4f1a9cf54c11d0"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = [
          + "allow_http",
        ]
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "ApacheServer"
```


user@user: ~/terra
+ "0.0.0.0/0",
]
+ from_port = 22
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "tcp"
+ security_groups = []
+ self = false
+ to_port = 22
# (1 unchanged attribute hidden)
},
+ {
+ cidr_blocks = [
+ "0.0.0.0/0",
]
+ from_port = 80
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "tcp"
+ security_groups = []
+ self = false
+ to_port = 80
# (1 unchanged attribute hidden)
},
]
+ name = "allow_http"
+ name_prefix = (known after apply)
+ owner_id = (known after apply)
+ revoke_rules_on_delete = false
+ tags_all = (known after apply)
+ vpc_id = (known after apply)
}
Plan: 2 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value: yes
aws_security_group.allow_http: Creating...
aws_security_group.allow_http: Creation complete after 7s [id=sg-06721
aws_instance.apache: Creating...
aws_instance.apache: Still creating... [10s elapsed]
aws_instance.apache: Creation complete after 17s [id=i-09e983552a24b69
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
user@user:~/terra$


Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
Outputs:
public_ip = "18.213.245.43"
user@user:~/terra$


Instances (1/1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Running
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
ApacheServer
i-09e983552a24b690d
Running
t2.micro
Initializing
View alarms
us-east-1b


i-09e983552a24b690d (ApacheServer)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-09e983552a24b690d
IPv6 address
–
Hostname type
IP name: ip-172-31-89-136.ec2.internal
Answer private resource DNS name
–
Public IPv4 address
44.211.41.123 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
ip-172-31-89-136.ec2.internal
Instance type
t2.micro
Private IPv4 addresses
172.31.89.136
Public IPv4 DNS
ec2-44-211-41-123.compute-1.amazonaws.com | open address
Elastic IP addresses
–

Apache2 Default Page

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively