Terraform : VPC

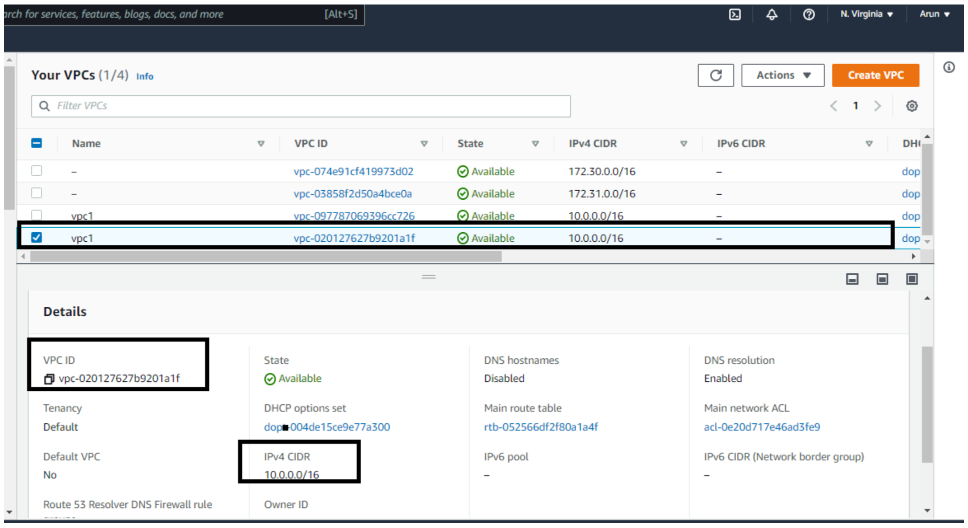
Step 1 : Provide security credentials

```
provider "aws" {
  region  ="us-east-1"
  access_key="AKIA6LFFGBWBZMAQFML"
  secret_key="VITGSj+3VwbCmmdnjajxMgOwNHlb9DEqJuZCZaS"
}
```

Step 2 : Create VPC

```
resource "aws_vpc" "vpc1" {
  cidr_block = "10.0.0.0/16"
  tags={
     Name="vpc1"
  }


 }
```



Your VPCs (1/4) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DH
–	vpc-074e91cf419973d02	Available	172.30.0.0/16	–	dop
–	vpc-03858f2d50a4bce0a	Available	172.31.0.0/16	–	dop
vpc1	vpc-097787069396cc726	Available	10.0.0.0/16	–	dop
vpc1	vpc-020127627b9201a1f	Available	10.0.0.0/16	–	dop

Details

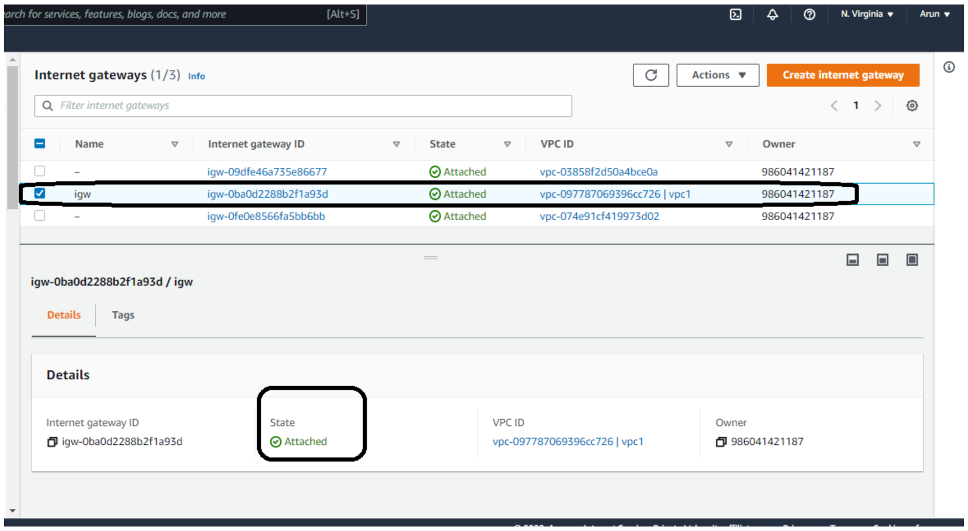
VPC ID vpc-020127627b9201a1f	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP options set dop■004de15ce9e77a300	Main route table rtb-052566df2f80a1a4f	Main network ACL acl-0e20d717e46ad3fe9
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –	IPv6 CIDR (Network border group) –
Route 53 Resolver DNS Firewall rule	Owner ID		

Step3 : Create internet Gateway

```
 resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.vpc1.id



  tags = {
    Name = "igw"
  }
}
```



Internet gateways (1/3) Info

	Name	Internet gateway ID	State	VPC ID	Owner
☐	–	igw-09dfe46a735e86677	✓ Attached	vpc-03858f2d50a4bce0a	986041421187
☑	igw	igw-0ba0d2288b2f1a93d	✓ Attached	vpc-097787069396cc726 | vpc1	986041421187
☐	–	igw-0fe0e8566fa5bb6bb	✓ Attached	vpc-074e91cf419973d02	986041421187

igw-0ba0d2288b2f1a93d / igw

Details | Tags

Details

Internet gateway ID	State	VPC ID	Owner
igw-0ba0d2288b2f1a93d	✓ Attached	vpc-097787069396cc726 | vpc1	986041421187

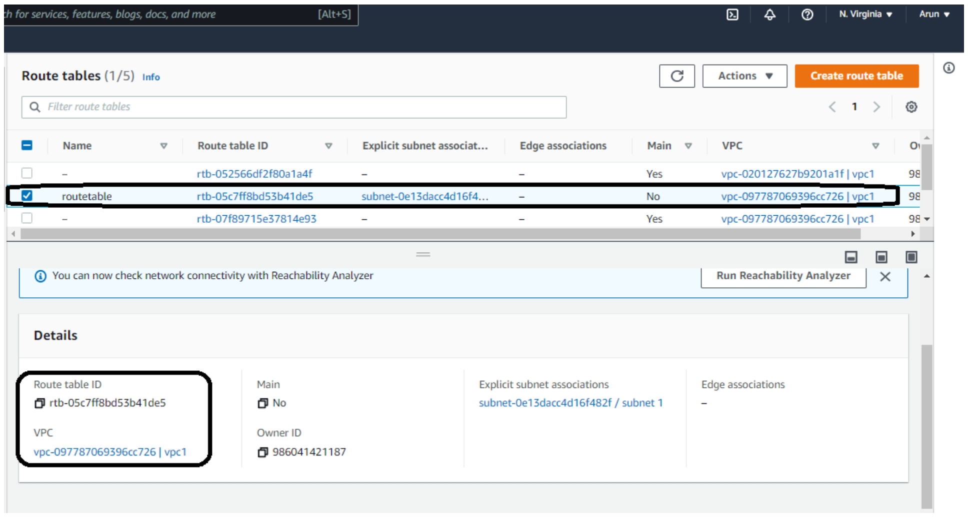
Step 4: Create route table

```
resource "aws_route_table" "rt" {
  vpc_id = aws_vpc.vpc1.id


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


  tags = {
    Name = "routetable"


  }
}
```


h for services, features, blogs, docs, and more [Alt+S]
N. Virginia
Arun
Actions
Create route table
Filter route tables
Run Reachability Analyzer

Route tables (1/5) Info

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	O
☐	–	rtb-052566df2f80a1a4f	–	–	Yes	vpc-020127627b9201a1f | vpc1	98
✓	routetable	rtb-05c7ff8bd53b41de5	subnet-0e13dacc4d16f4...	–	No	vpc-097787069396cc726 | vpc1	98
☐	–	rtb-07f89715e37814e93	–	–	Yes	vpc-097787069396cc726 | vpc1	98

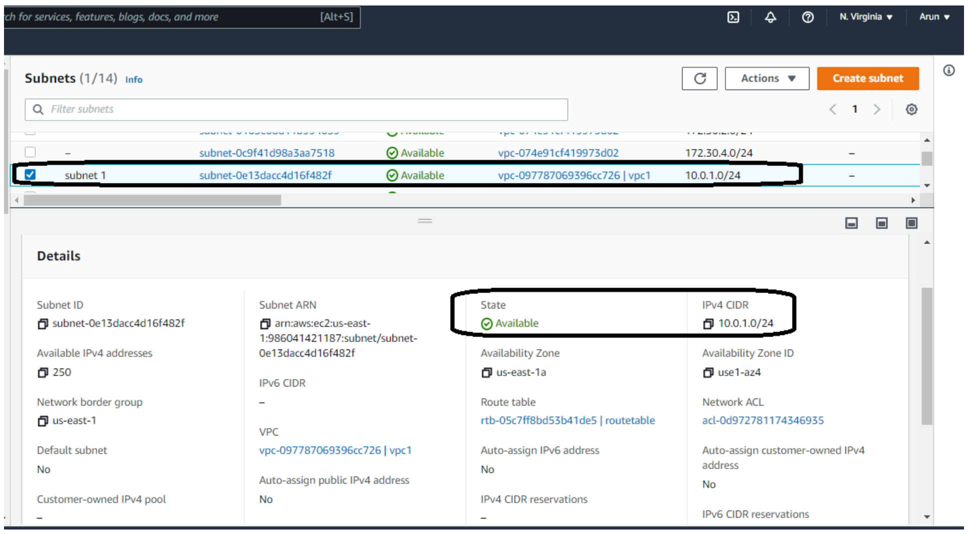
You can now check network connectivity with Reachability Analyzer

Details

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-05c7ff8bd53b41de5	No	subnet-0e13dacc4d16f482f / subnet 1	–
VPC	Owner ID		
vpc-097787069396cc726 | vpc1	986041421187		

Step 5 : Create subnet

```
resource "aws_subnet" "subnet1" {
   cidr_block="10.0.0.0/20"
   vpc_id = aws_vpc.vpc1.id
   tags = {
      Name="subnet 1"
   }
}
```



Subnets (1/14) Info

☐	–	subnet-0c9f41d98a3aa7518	Available	vpc-074e91cf419973d02	172.30.4.0/24	–
✓	subnet 1	subnet-0e13dacc4d16f482f	Available	vpc-097787069396cc726 | vpc1	10.0.1.0/24	–

Details

Subnet ID subnet-0e13dacc4d16f482f	Subnet ARN arn:aws:ec2:us-east-1:986041421187:subnet/subnet-0e13dacc4d16f482f	State Available	IPv4 CIDR 10.0.1.0/24
Available IPv4 addresses 250	IPv6 CIDR –	Availability Zone us-east-1a	Availability Zone ID use1-az4
Network border group us-east-1	VPC vpc-097787069396cc726 | vpc1	Route table rtb-05c7ff8bd53b41de5 | routetable	Network ACL acl-0d972781174346935
Default subnet No	Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No
Customer-owned IPv4 pool –		IPv4 CIDR reservations –	IPv6 CIDR reservations

Step 6 : Route table association with subnet

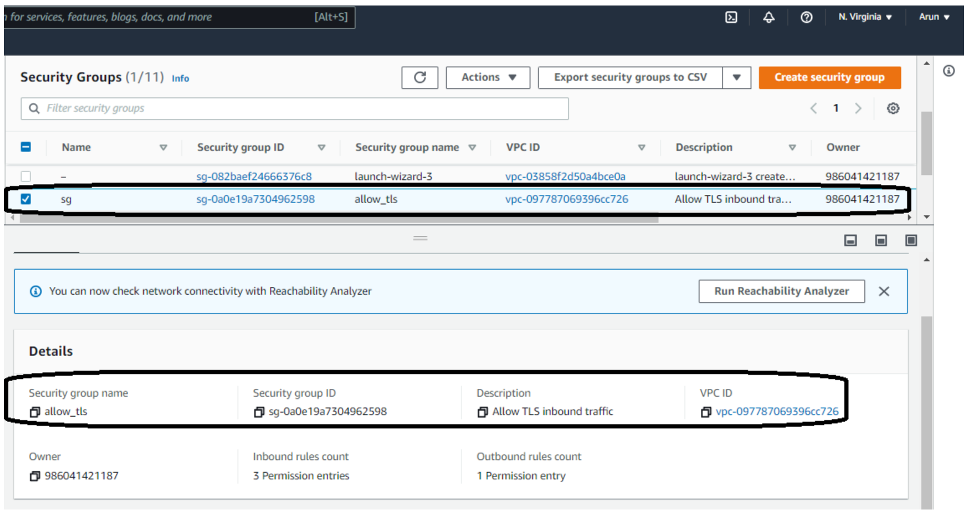
```
resource "aws_route_table_association" "a" {
  subnet_id      = aws_subnet.subnet1.id
  route_table_id = aws_route_table.rt.id
}
```

Step7: Create security group allowing ports 443, 80, 22

```
resource "aws_security_group" "web" {
  name        = "allow_tls"
  description = "Allow TLS inbound traffic"
  vpc_id      = aws_vpc.vpc1.id



  ingress {
    description      = "TLS from VPC"
    from_port        = 443
    to_port          = 443
    protocol         = "tcp"
    cidr_blocks      = [aws_vpc.vpc1.cidr_block]
  }
  ingress {
    description      = "TLS from VPC"
    from_port        = 80
    to_port          = 80
    protocol         = "tcp"
    cidr_blocks      = [aws_vpc.vpc1.cidr_block]
  }
  ingress {
    description      = "TLS from VPC"
    from_port        = 22
    to_port          = 22
    protocol         = "tcp"
    cidr_blocks      = [aws_vpc.vpc1.cidr_block]
  }
}
```


for services, features, blogs, docs, and more [Alt+S]
N. Virginia
Arun
Actions
Export security groups to CSV
Create security group
Filter security groups
1

Security Groups (1/11) Info

	Name	Security group ID	Security group name	VPC ID	Description	Owner
☐	–	sg-082baef24666376c8	launch-wizard-3	vpc-03858f2d50a4bce0a	launch-wizard-3 create…	986041421187
✓	sg	sg-0a0e19a7304962598	allow_tls	vpc-097787069396cc726	Allow TLS inbound tra…	986041421187

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Details

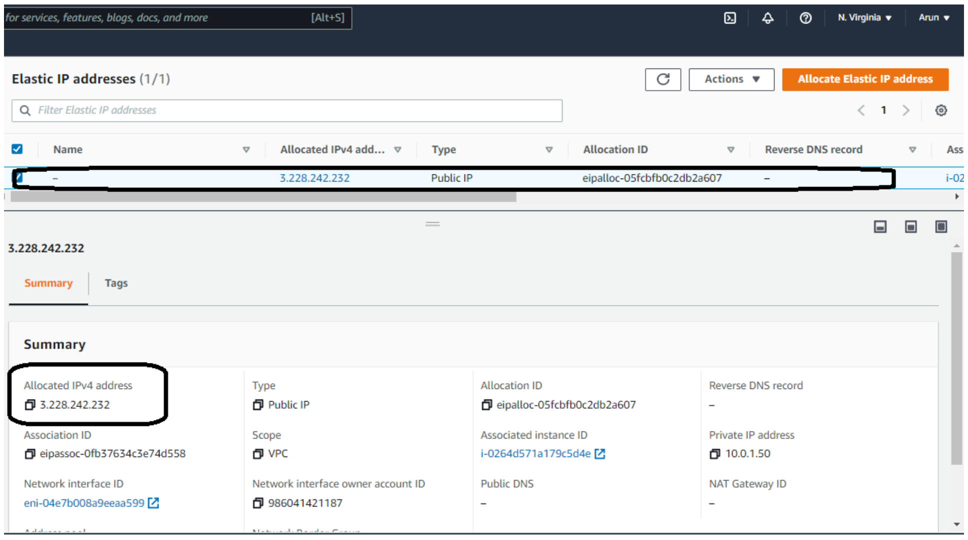
Security group name	Security group ID	Description	VPC ID
allow_tls	sg-0a0e19a7304962598	Allow TLS inbound traffic	vpc-097787069396cc726
Owner	Inbound rules count	Outbound rules count	
986041421187	3 Permission entries	1 Permission entry	

Step 8 : Create network interface

```
resource "aws_network_interface" "net" {
  subnet_id       = aws_subnet.subnet1.id
  private_ips     = ["10.0.0.50"]
  security_groups = [aws_security_group.web.id]
```

Step 09 : Assign elastic ip to an network interface

```
resource "aws_eip" "prodeip" {
  vpc                       = true
  network_interface         = aws_network_interface.net.id
  associate_with_private_ip = "10.0.0.50"
  depends_on = [ aws_internet_gateway.gw]
}
```


for services, features, blogs, docs, and more [Alt+S]
N. Virginia
Arun
Actions
Allocate Elastic IP address
Filter Elastic IP addresses
Tags

Elastic IP addresses (1/1)

Name	Allocated IPv4 add...	Type	Allocation ID	Reverse DNS record	Ass
–	3.228.242.232	Public IP	eipalloc-05fcbfb0c2db2a607	–	i-02

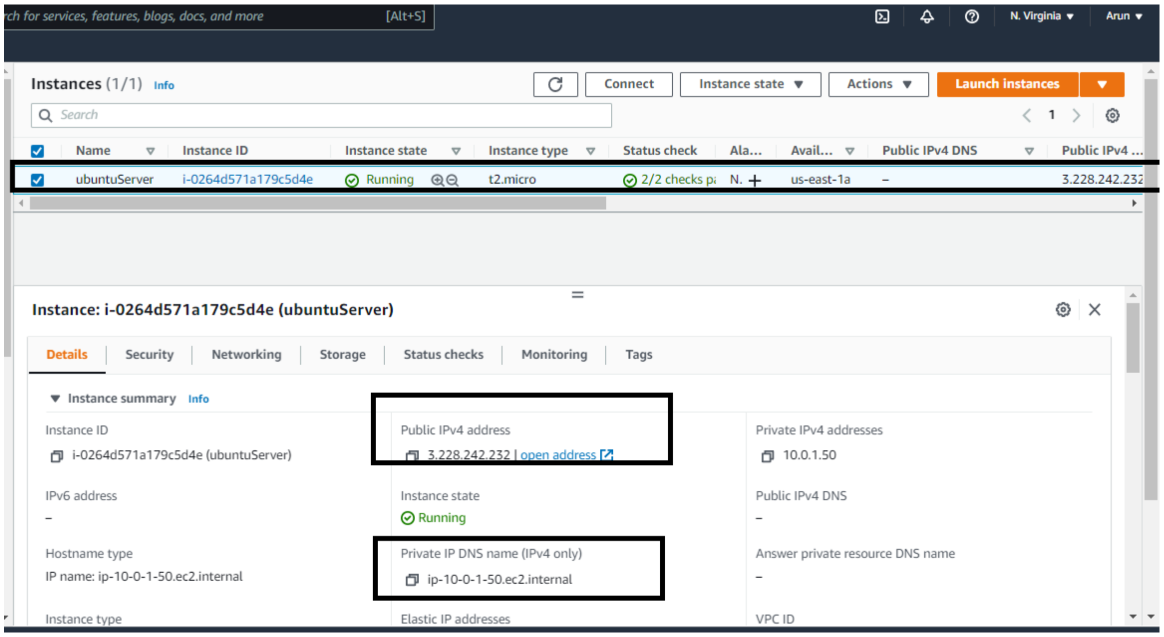
3.228.242.232

Summary

Allocated IPv4 address 3.228.242.232	Type Public IP	Allocation ID eipalloc-05fcbfb0c2db2a607	Reverse DNS record –
Association ID eipassoc-0fb37634c3e74d558	Scope VPC	Associated instance ID i-0264d571a179c5d4e	Private IP address 10.0.1.50
Network interface ID eni-04e7b008a9eeaa599	Network interface owner account ID 986041421187	Public DNS –	NAT Gateway ID –

Step 10 : create ubuntu ec2

```
}
resource "aws_instance" "ubuntu" {
  ami           = "ami-04505e74c0741db8d"
  instance_type = "t2.micro"
  tags = {
    Name = "ubuntuServer"
  }
}
```



ch for services, features, blogs, docs, and more [Alt+S]

N. Virginia ▾ Arun ▾

Instances (1/1) Info

Connect | Instance state ▾ | Actions ▾ | Launch instances ▾

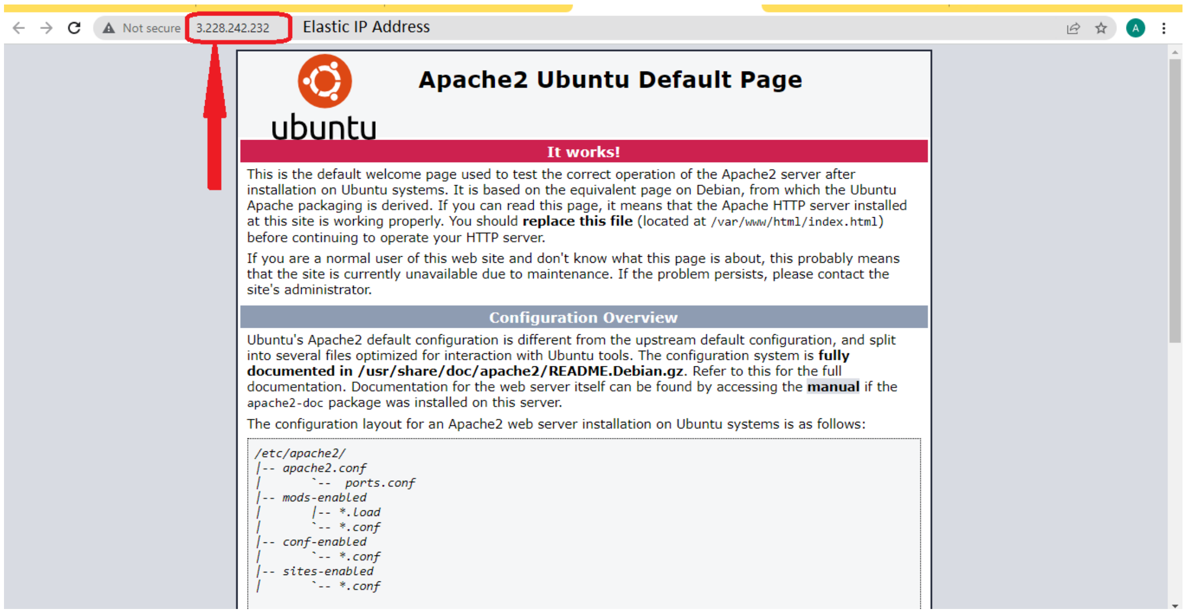
Name	Instance ID	Instance state	Instance type	Status check	Ala...	Avail...	Public IPv4 DNS	Public IPv4 ...
ubuntuServer	i-0264d571a179c5d4e	Running	t2.micro	2/2 checks pa	N. +	us-east-1a	–	3.228.242.232

Instance: i-0264d571a179c5d4e (ubuntuServer)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Instance ID i-0264d571a179c5d4e (ubuntuServer)	Public IPv4 address 3.228.242.232 | open address	Private IPv4 addresses 10.0.1.50
IPv6 address –	Instance state Running	Public IPv4 DNS –
Hostname type IP name: ip-10-0-1-50.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-1-50.ec2.internal	Answer private resource DNS name –
Instance type	Elastic IP addresses	VPC ID



Not secure 3.228.242.232 Elastic IP Address

Apache2 Ubuntu Default Page

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
```

Successfully Launched Ubuntu Server, Installed Apache server, Attached EIP, Created VPC