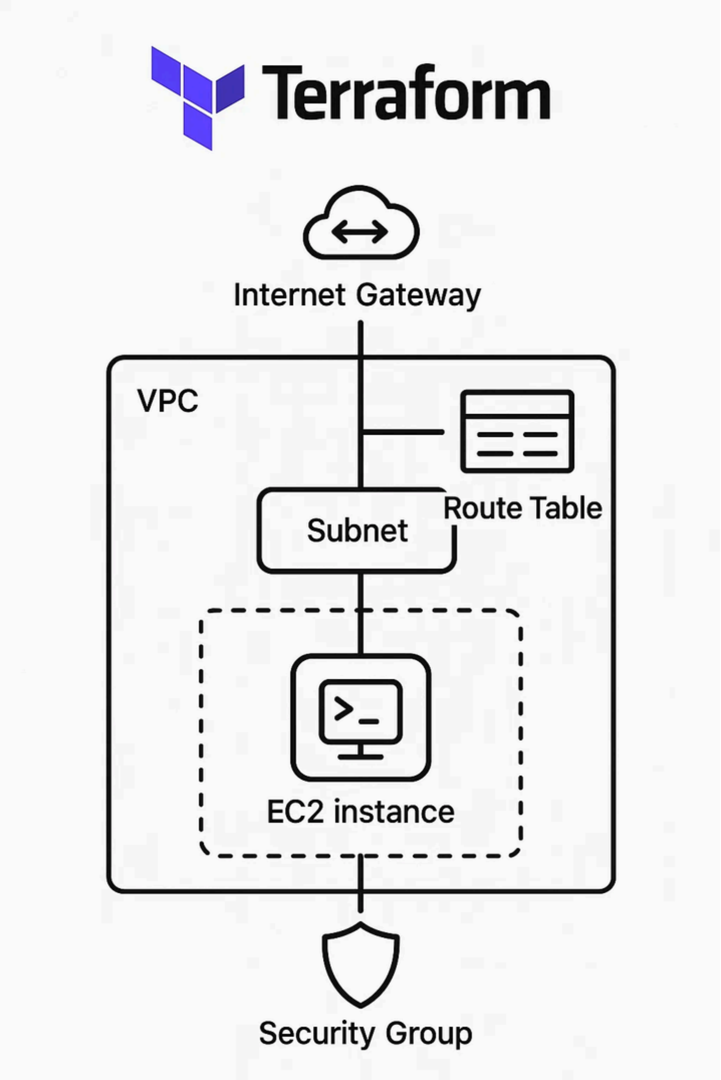
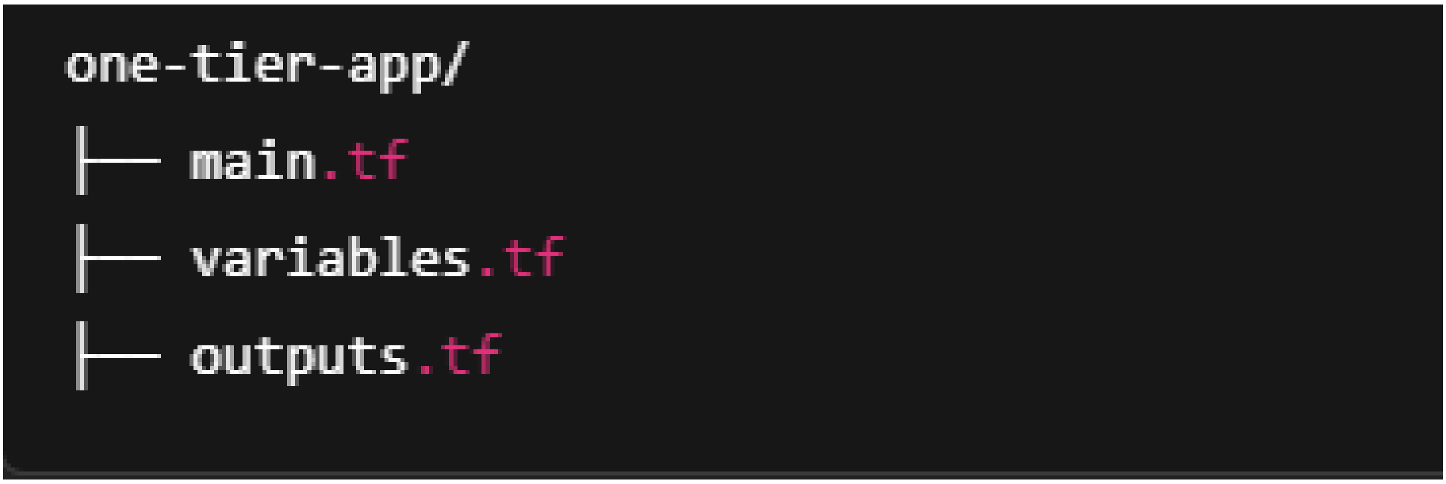
1-tier AWS architecture using Terraform

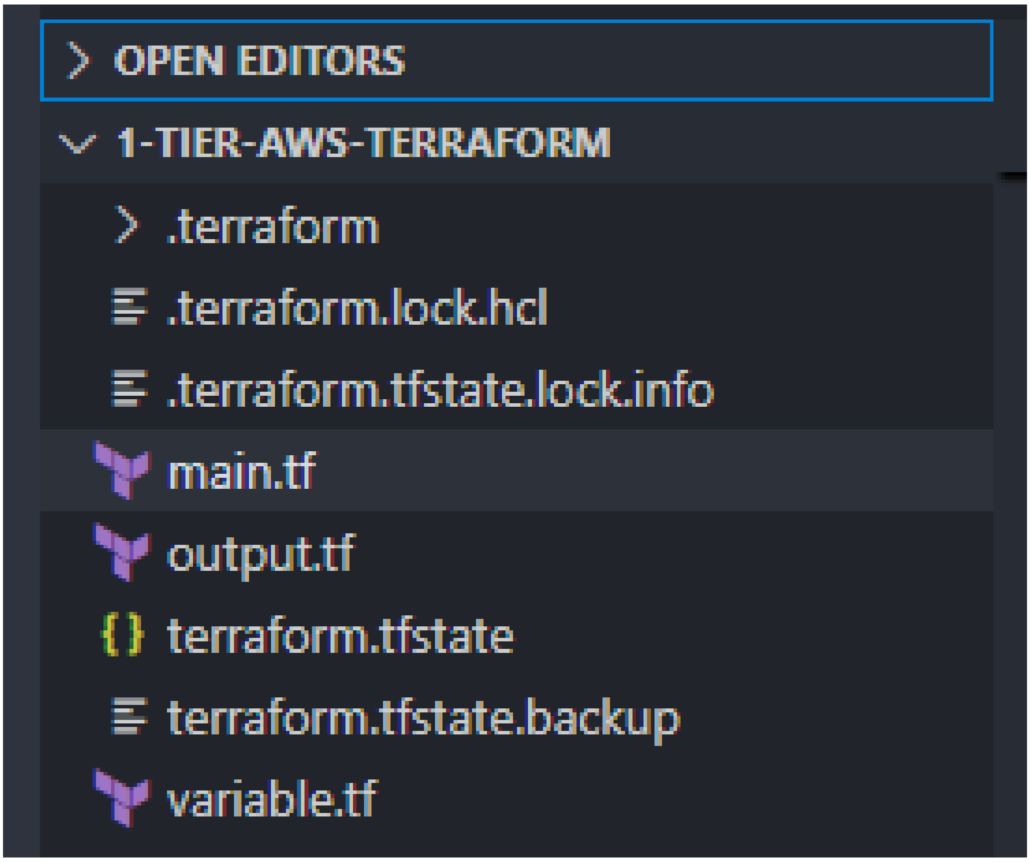


🧱 Components

- AWS VPC
- Public Subnet
- Internet Gateway
- Route Table
- Security Group (allows SSH and HTTP)
- EC2 Instance (Ubuntu with Apache)
- Terraform Provisioning



```
one-tier-app/
├── main.tf
├── variables.tf
├── outputs.tf
```



```
> OPEN EDITORS
v 1-TIER-AWS-TERRAFORM
  > .terraform
  .terraform.lock.hcl
  .terraform.tfstate.lock.info
  main.tf
  output.tf
  terraform.tfstate
  terraform.tfstate.backup
  variable.tf
```

main.tf:

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


resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
  tags = {
    Name = "one-tier-vpc"
  }
}

resource "aws_subnet" "main" {
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "us-east-1a"
  map_public_ip_on_launch = true
  tags = {
    Name = "one-tier-subnet"
  }
}

resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.main.id
  tags = {
    Name = "one-tier-igw"
  }
}

resource "aws_route_table" "public" {
  vpc_id = aws_vpc.main.id

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

  tags = {
    Name = "public-route-table"
  }
}

resource "aws_route_table_association" "a" {
  subnet_id      = aws_subnet.main.id
  route_table_id = aws_route_table.public.id
}

resource "aws_security_group" "allow_ssh_http" {
  name        = "allow_ssh_http"
  description = "Allow SSH and HTTP inbound traffic"
  vpc_id      = aws_vpc.main.id

  ingress {
    description = "SSH"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    description = "HTTP"
    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 = "one-tier-sg"
  }
}

resource "aws_instance" "web" {
  ami                    = "ami-0c7217cdde317cfec" # Ubuntu 22.04 in us-east-1
  instance_type          = "t2.micro"
  subnet_id              = aws_subnet.main.id
  vpc_security_group_ids = [aws_security_group.allow_ssh_http.id]
  key_name               = var.key_name


  user_data = <<-EOF
            #!/bin/bash
            apt update -y
            apt install -y apache2
            systemctl start apache2
            systemctl enable apache2
            echo "Hello from Terraform EC2" > /var/www/html/index.html
            EOF


  tags = {
    Name = "web-server"
  }
}
```

variables.tf:

```
variable "key_name" {
  description = "Name of the existing EC2 Key Pair to use"
  type        = string
}
```

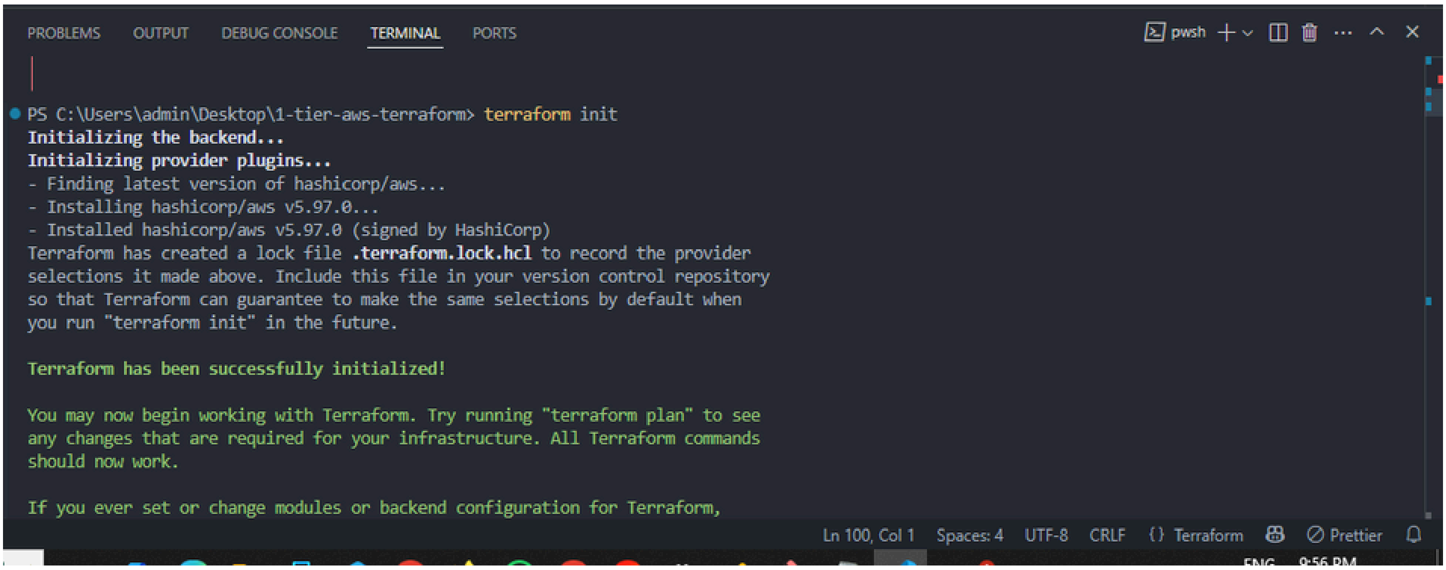
outputs.tf:

```
output "public_ip" {
  description = "Public IP of the EC2 instance"
  value       = aws_instance.web.public_ip
}



output "web_url" {

  value = "http://${aws_instance.web.public_ip}"

}
```

How to Run It

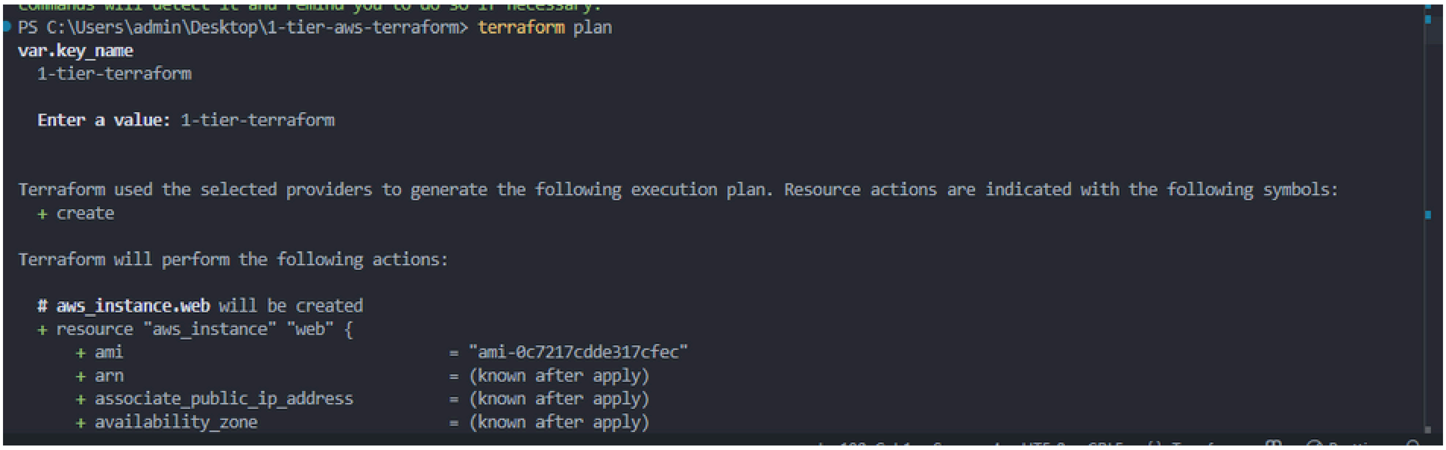
1. ***Initialize Terraform***

terraform init



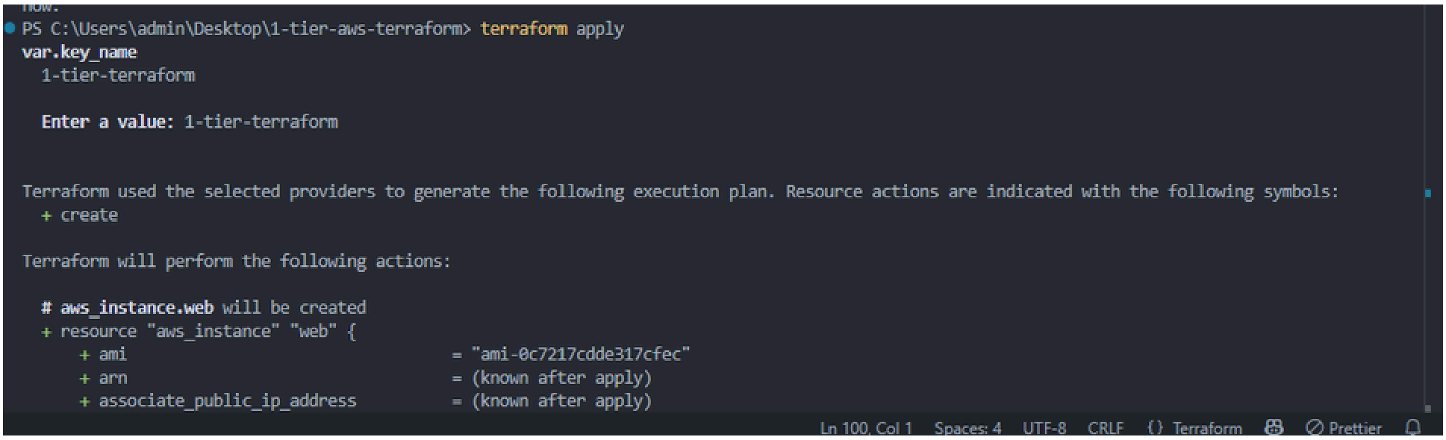
```
PS C:\Users\admin\Desktop\1-tier-aws-terraform> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.97.0...
- Installed hashicorp/aws v5.97.0 (signed by HashiCorp)
Terraform has created a lock file .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,
```

terraform plan
it will ask us to verify key-pair name , verify all resources which we need to create

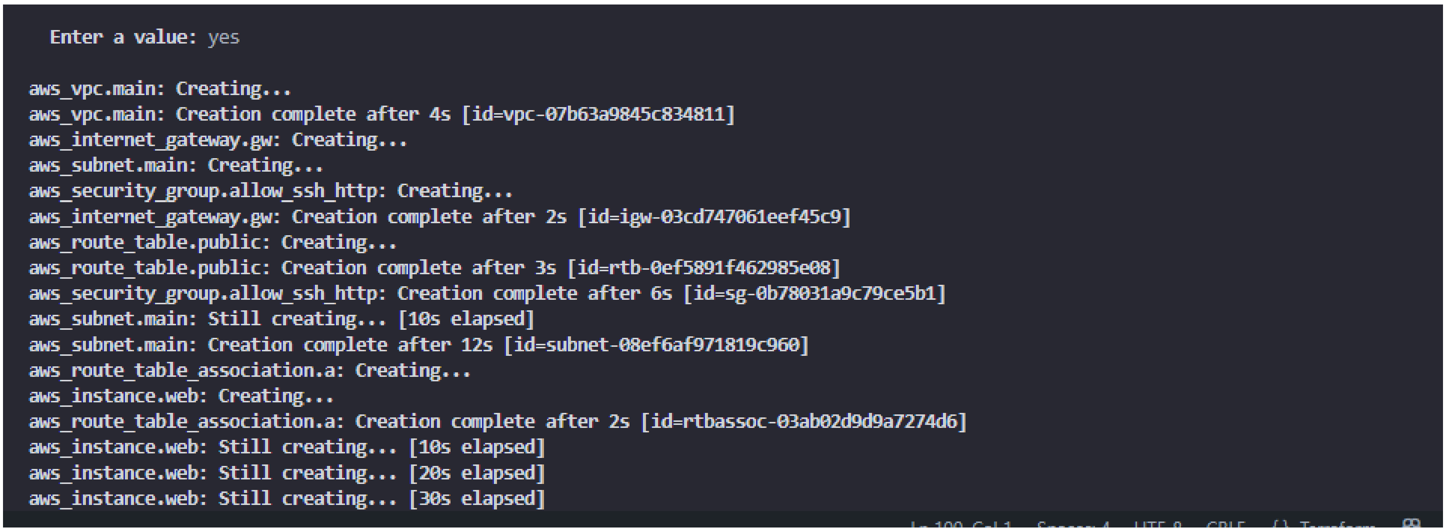


```
commands will detect it and remind you to do so if necessary.
PS C:\Users\admin\Desktop\1-tier-aws-terraform> terraform plan
var.key_name
  1-tier-terraform

  Enter a value: 1-tier-terraform


Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.web will be created
  + resource "aws_instance" "web" {
      + ami                                  = "ami-0c7217cdde317cfec"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
```

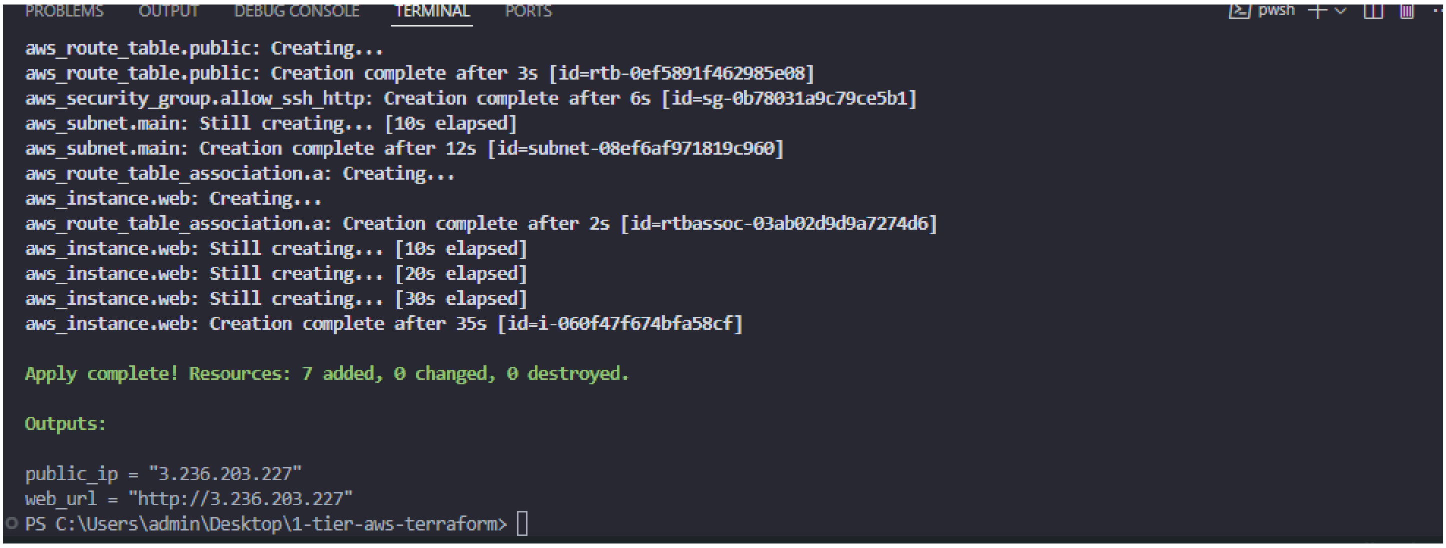
Terraform apply



```
now:
PS C:\Users\admin\Desktop\1-tier-aws-terraform> terraform apply
var.key_name
  1-tier-terraform

  Enter a value: 1-tier-terraform


Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.web will be created
  + resource "aws_instance" "web" {
      + ami                                  = "ami-0c7217cdde317cfec"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
```



```
  Enter a value: yes

aws_vpc.main: Creating...
aws_vpc.main: Creation complete after 4s [id=vpc-07b63a9845c834811]
aws_internet_gateway.gw: Creating...
aws_subnet.main: Creating...
aws_security_group.allow_ssh_http: Creating...
aws_internet_gateway.gw: Creation complete after 2s [id=igw-03cd747061eef45c9]
aws_route_table.public: Creating...
aws_route_table.public: Creation complete after 3s [id=rtb-0ef5891f462985e08]
aws_security_group.allow_ssh_http: Creation complete after 6s [id=sg-0b78031a9c79ce5b1]
aws_subnet.main: Still creating... [10s elapsed]
aws_subnet.main: Creation complete after 12s [id=subnet-08ef6af971819c960]
aws_route_table_association.a: Creating...
aws_instance.web: Creating...
aws_route_table_association.a: Creation complete after 2s [id=rtbassoc-03ab02d9d9a7274d6]
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Still creating... [20s elapsed]
aws_instance.web: Still creating... [30s elapsed]
```

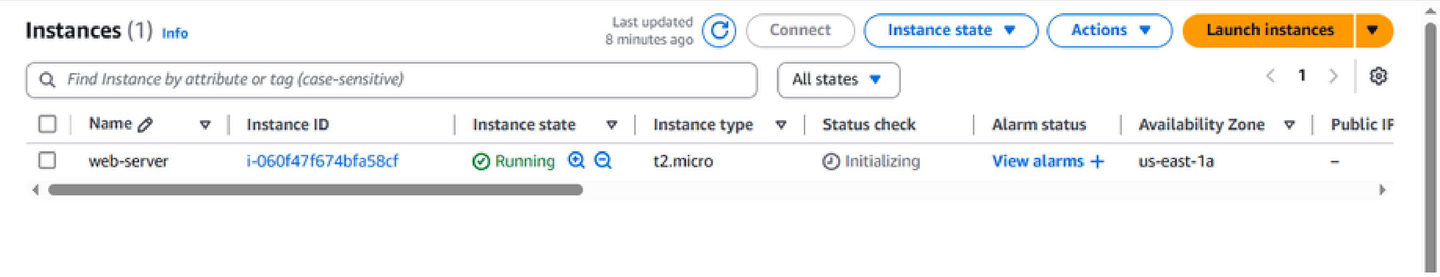


```
aws_route_table.public: Creating...
aws_route_table.public: Creation complete after 3s [id=rtb-0ef5891f462985e08]
aws_security_group.allow_ssh_http: Creation complete after 6s [id=sg-0b78031a9c79ce5b1]
aws_subnet.main: Still creating... [10s elapsed]
aws_subnet.main: Creation complete after 12s [id=subnet-08ef6af971819c960]
aws_route_table_association.a: Creating...
aws_instance.web: Creating...
aws_route_table_association.a: Creation complete after 2s [id=rtbassoc-03ab02d9d9a7274d6]
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Still creating... [20s elapsed]
aws_instance.web: Still creating... [30s elapsed]
aws_instance.web: Creation complete after 35s [id=i-060f47f674bfa58cf]

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

Outputs:

public_ip = "3.236.203.227"
web_url = "http://3.236.203.227"
PS C:\Users\admin\Desktop\1-tier-aws-terraform>
```

we can see all required resources are created

Let’s gooooo…. and see our resources are created in AWS console

Ec2 instance (web-server)

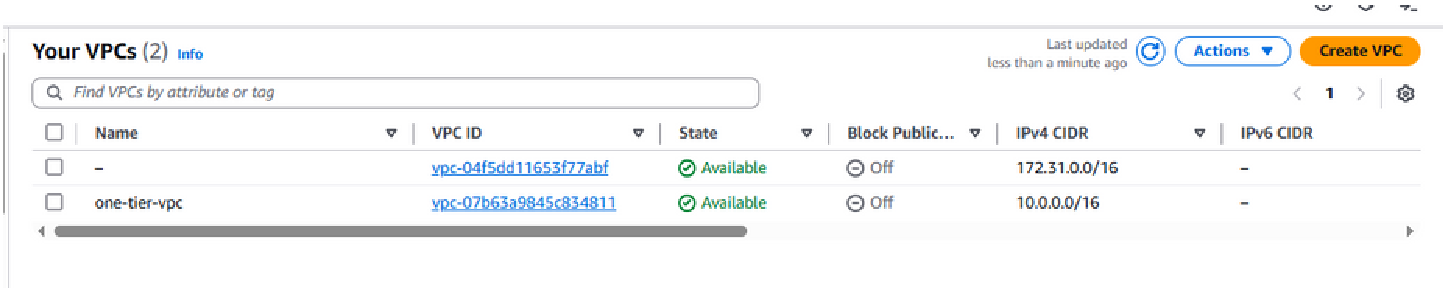


Instances (1) Info

Find Instance by attribute or tag (case-sensitive)

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IF
□	web-server	i-060f47f674bfa58cf	✓ Running	t2.micro	Initializing	View alarms +	us-east-1a	–

VPC (one-tier-vpc)

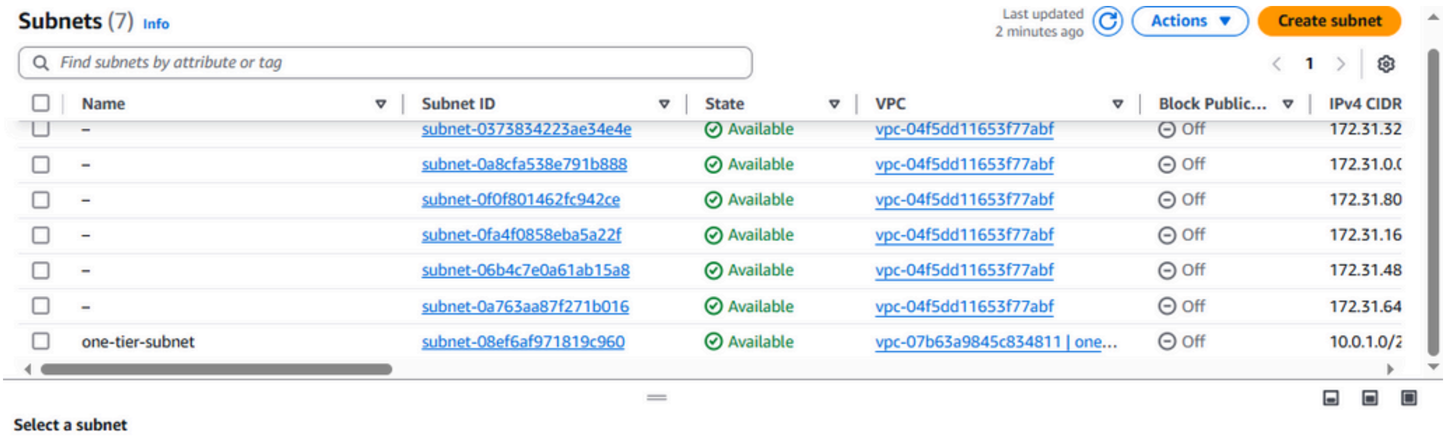


Your VPCs (2) Info

Find VPCs by attribute or tag

□	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
□	–	vpc-04f5dd11653f77abf	✓ Available	⊖ Off	172.31.0.0/16	–
□	one-tier-vpc	vpc-07b63a9845c834811	✓ Available	⊖ Off	10.0.0.0/16	–

Subnet (one-tier-subnet)



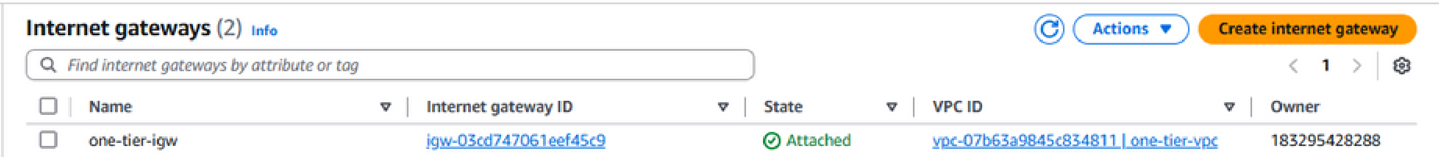
Subnets (7) Info

Find subnets by attribute or tag

□	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
□	–	subnet-0373834223ae34e4e	✓ Available	vpc-04f5dd11653f77abf	⊖ Off	172.31.32
□	–	subnet-0a8cfa538e791b888	✓ Available	vpc-04f5dd11653f77abf	⊖ Off	172.31.0.(
□	–	subnet-0f0f801462fc942ce	✓ Available	vpc-04f5dd11653f77abf	⊖ Off	172.31.80
□	–	subnet-0fa4f0858eba5a22f	✓ Available	vpc-04f5dd11653f77abf	⊖ Off	172.31.16
□	–	subnet-06b4c7e0a61ab15a8	✓ Available	vpc-04f5dd11653f77abf	⊖ Off	172.31.48
□	–	subnet-0a763aa87f271b016	✓ Available	vpc-04f5dd11653f77abf	⊖ Off	172.31.64
□	one-tier-subnet	subnet-08ef6af971819c960	✓ Available	vpc-07b63a9845c834811 | one...	⊖ Off	10.0.1.0/2

Select a subnet

Internet Gateway (one-tier-igw)

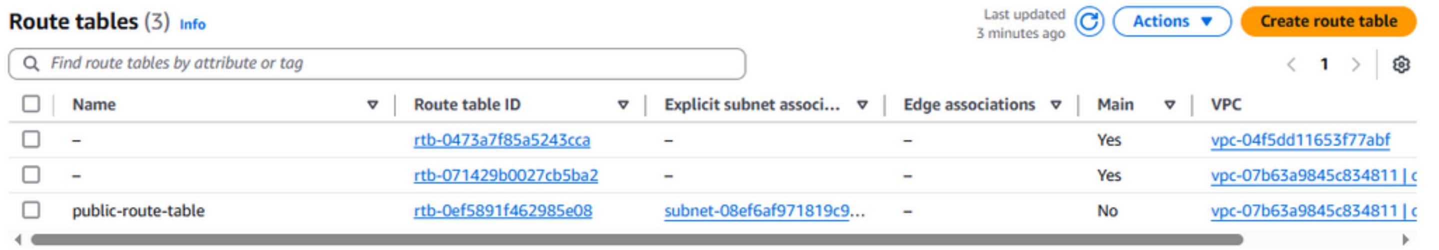


Internet gateways (2) Info

Find internet gateways by attribute or tag

□	Name	Internet gateway ID	State	VPC ID	Owner
□	one-tier-igw	igw-03cd747061eef45c9	✓ Attached	vpc-07b63a9845c834811 | one-tier-vpc	183295428288

Route Table (public-route-table)

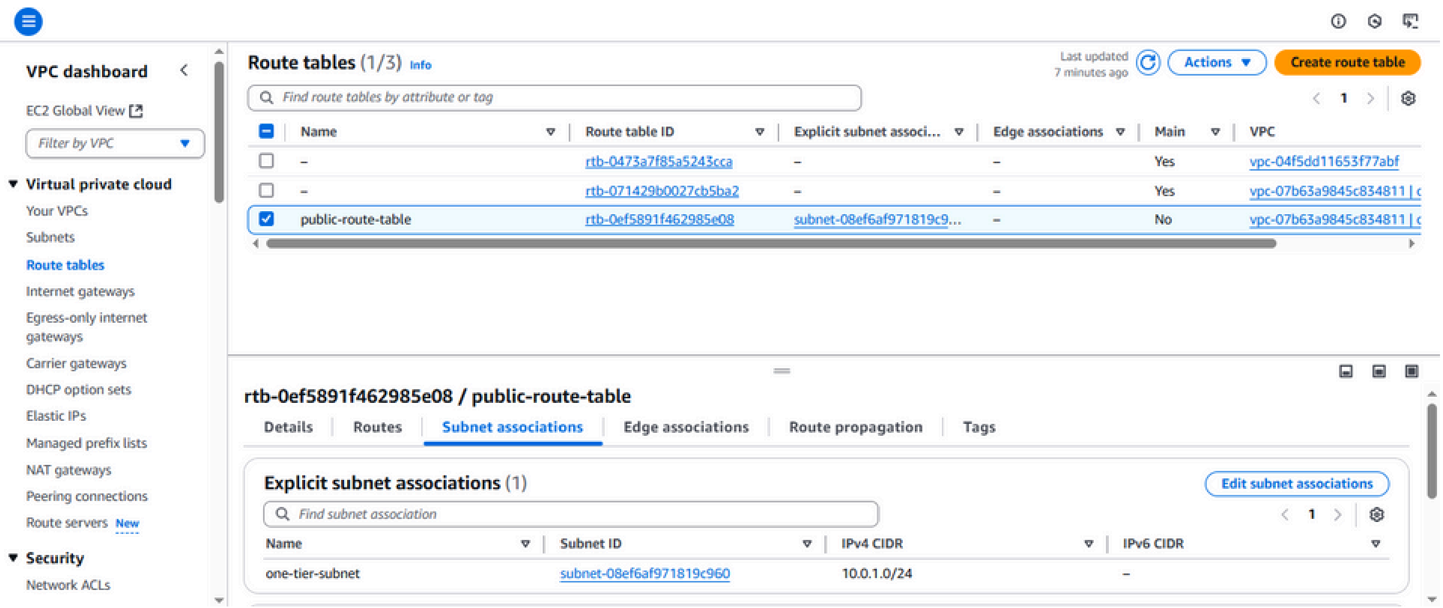


Route tables (3) Info

Find route tables by attribute or tag

□	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
□	–	rtb-0473a7f85a5243cca	–	–	Yes	vpc-04f5dd11653f77abf
□	–	rtb-071429b0027cb5ba2	–	–	Yes	vpc-07b63a9845c834811 | c
□	public-route-table	rtb-0ef5891f462985e08	subnet-08ef6af971819c9...	–	No	vpc-07b63a9845c834811 | c

Subnet Association



Route tables (1/3) Info

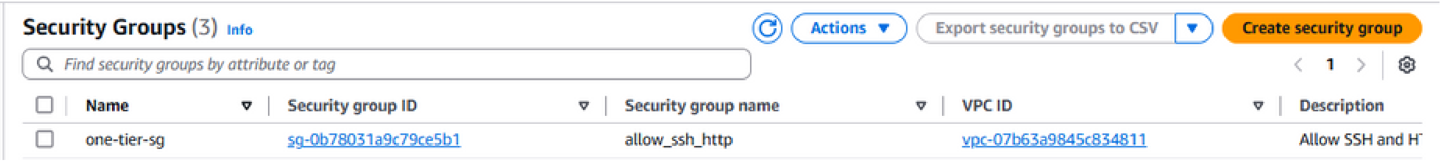
	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
□	–	rtb-0473a7f85a5243cca	–	–	Yes	vpc-04f5dd11653f77abf
□	–	rtb-071429b0027cb5ba2	–	–	Yes	vpc-07b63a9845c834811 | c
☑	public-route-table	rtb-0ef5891f462985e08	subnet-08ef6af971819c9...	–	No	vpc-07b63a9845c834811 | c

rtb-0ef5891f462985e08 / public-route-table

Explicit subnet associations (1)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
one-tier-subnet	subnet-08ef6af971819c960	10.0.1.0/24	–

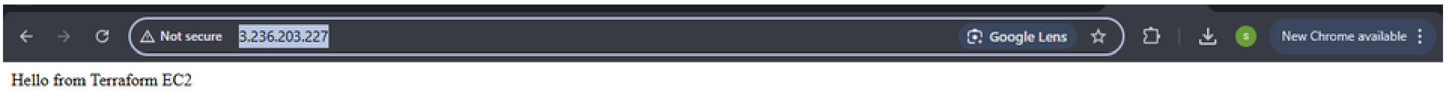
Security Group (one-tier-sg)


Actions
Export security groups to CSV
Create security group
Find security groups by attribute or tag

Security Groups (3) Info

□	Name	Security group ID	Security group name	VPC ID	Description
□	one-tier-sg	sg-0b78031a9c79ce5b1	allow_ssh_http	vpc-07b63a9845c834811	Allow SSH and H

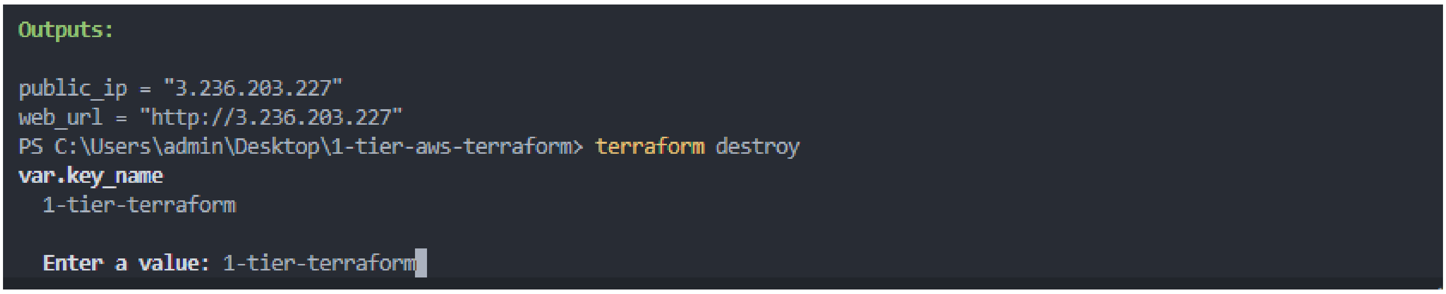
Access Your App http://3.236.203.227



Hello from Terraform EC2

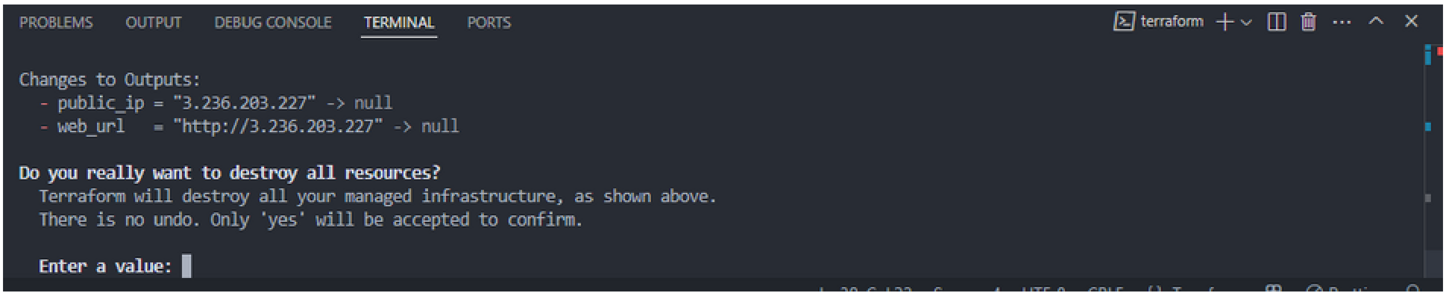
Successfully we can access .... we have provisioned with Terraform

lets cleanup our resources with *terraform destroy*

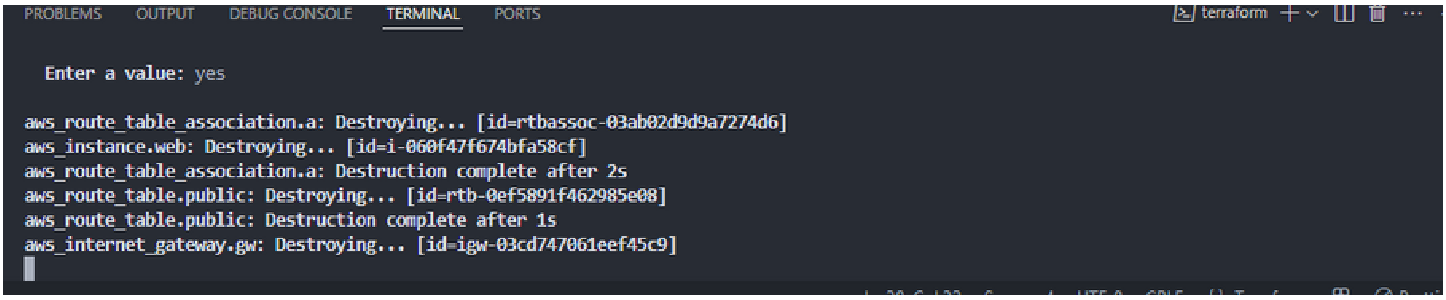


```
Outputs:

public_ip = "3.236.203.227"
web_url = "http://3.236.203.227"
PS C:\Users\admin\Desktop\1-tier-aws-terraform> terraform destroy
var.key_name
  1-tier-terraform

  Enter a value: 1-tier-terraform
```

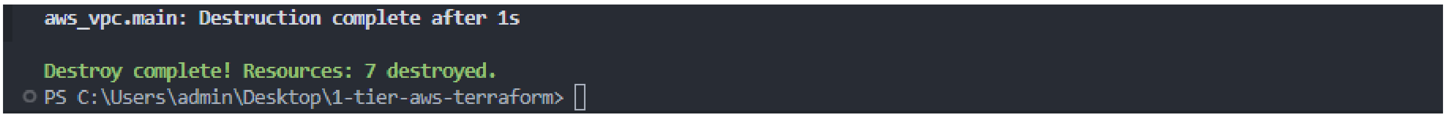
confirm to clean up



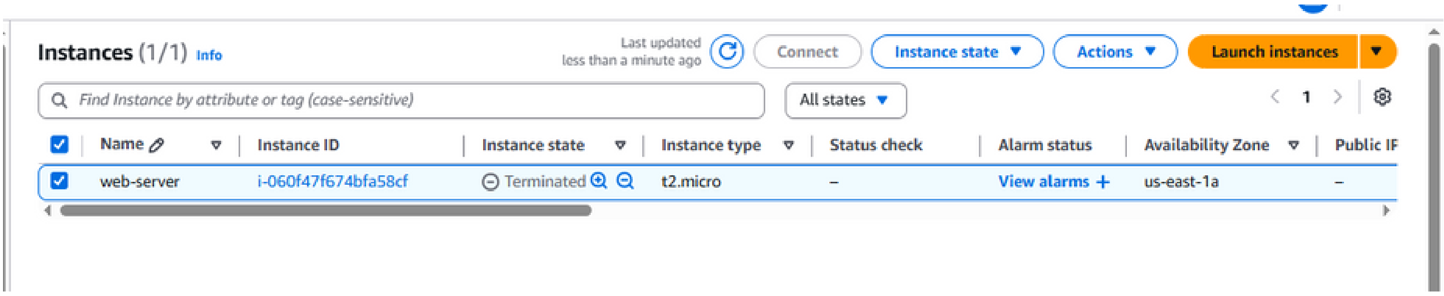
```
Changes to Outputs:
  - public_ip = "3.236.203.227" -> null
  - web_url   = "http://3.236.203.227" -> null

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:
```



```
  Enter a value: yes

aws_route_table_association.a: Destroying... [id=rtbassoc-03ab02d9d9a7274d6]
aws_instance.web: Destroying... [id=i-060f47f674bfa58cf]
aws_route_table_association.a: Destruction complete after 2s
aws_route_table.public: Destroying... [id=rtb-0ef5891f462985e08]
aws_route_table.public: Destruction complete after 1s
aws_internet_gateway.gw: Destroying... [id=igw-03cd747061eef45c9]
```



```
  aws_vpc.main: Destruction complete after 1s

  Destroy complete! Resources: 7 destroyed.
○ PS C:\Users\admin\Desktop\1-tier-aws-terraform>
```



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
☑	web-server	i-060f47f674bfa58cf	⊖ Terminated	t2.micro	–	View alarms +	us-east-1a	–