MAKE THE PUBLIC IP HIT AN INDEX.HTML

IMPORTING THE INSTANCE TO MODIFY IT

```
root@advika:~/terraform# vi main.tf
root@advika:~/terraform# cat main.tf
provider "aws" {
  region = "us-east-1"
}
resource "aws_instance" "Terraform_IAM" {
        instance_id="i-0b639827f13ad24dc"
}
```

```
root@advika:~/terraform# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.1.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to se
e
any changes that are required for your infrastructure. All Terraform command
s
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, ot
her
commands will detect it and remind you to do so if necessary.
```

```
root@advika:~/terraform# terraform import aws_instance.Terraform_IAM i-0b639
827f13ad24dc
aws_instance.Terraform_IAM: Importing from ID "i-0b639827f13ad24dc"...
aws_instance.Terraform_IAM: Import prepared!
  Prepared aws_instance for import
aws_instance.Terraform_IAM: Refreshing state... [id=i-0b639827f13ad24dc]

Import successful!

The resources that were imported are shown above. These resources are now in
your Terraform state and will henceforth be managed by Terraform.
```

CREATING THE INDEX.HTML FILE

background should be | index.html

File Edit View

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Terraform Built Instance</title>
</head>
<body>
    <h1>Welcome to the file built using Terraform</h1>
    <p>This web page is hosted on the EC2 instance configured and deployed
with Terraform i.e, Terraform_IAM.</p>
    <p>Thank you for visiting!</p>
</body>
</html>
```

Saved it to a local path on my system

UPDATED main.tf FILE

```
terraform{
        required_providers {
                aws={
                        source="hashicorp/aws"
                        version="5.1.0"
                }
        }
}
provider "aws" {
  region = "us-east-1"
}

data "local_file" "index_html" {
        filename="C:\\Users\\advik\\Downloads\\index.html"

resource "aws_instance" "Terraform_IAM" {
        ami="ami-04b70fa74e45c3917"
        instance_type="t2.micro"

        user_data = <<-EOF
                #!/bin/bash
                apt-get update
                apt-get install -y apache2
                systemctl start apache2
                systemctl enable apache2
                echo '${data.local_file.index_html.content}' > /var/www/html
/index.html
                EOF
        }

        tags={
                Name="Terraform_IAM"
        }
}
output "ec2_public_ip" {
        value=aws_instance.Terraform_IAM.public_ip
}
```

CHECKING STATE FILES BEFORE RUNNING

```
root@advika:~/terraform# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Installing hashicorp/local v2.5.1...
- Installed hashicorp/local v2.5.1 (signed by HashiCorp)
- Using previously-installed hashicorp/aws v5.1.0

Terraform has made some changes to the provider dependency selections record
ed
in the .terraform.lock.hcl file. Review those changes and commit them to you
r
version control system if they represent changes you intended to make.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to se
e
any changes that are required for your infrastructure. All Terraform command
s
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, ot
her
commands will detect it and remind you to do so if necessary.
```

```
root@advika:~/terraform# terraform validate
Success! The configuration is valid.

root@advika:~/terraform# terraform plan
data.local_file.index_html: Reading...
data.local_file.index_html: Read complete after 1s [id=58702a1d3e6bf68fbdd91
9d1fce0f5d2e9221f88]
aws_instance.Terraform_IAM: Refreshing state... [id=i-0b639827f13ad24dc]
aws_instance.demo-server: Refreshing state... [id=i-0b639827f13ad24dc]

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  ~ update in-place
  - destroy

Terraform will perform the following actions:

  # aws_instance.Terraform_IAM will be updated in-place
  ~ resource "aws_instance" "Terraform_IAM" {
        id                                   = "i-0b639827f13ad24dc"
        tags                                 = {
            "Name" = "Terraform_IAM"
        }
      + user_data                            = "df231c64b9b5dc6c5c60bb54e498
7c2c4e7d627c"
      + user_data_replace_on_change          = false
        # (36 unchanged attributes hidden)

        # (8 unchanged blocks hidden)
    }
```

```
root@advika:~/terraform# terraform validate
Success! The configuration is valid.

root@advika:~/terraform# terraform state list
data.local_file.index_html
aws_instance.Terraform_IAM
root@advika:~/terraform# terraform fmt
main.tf
```

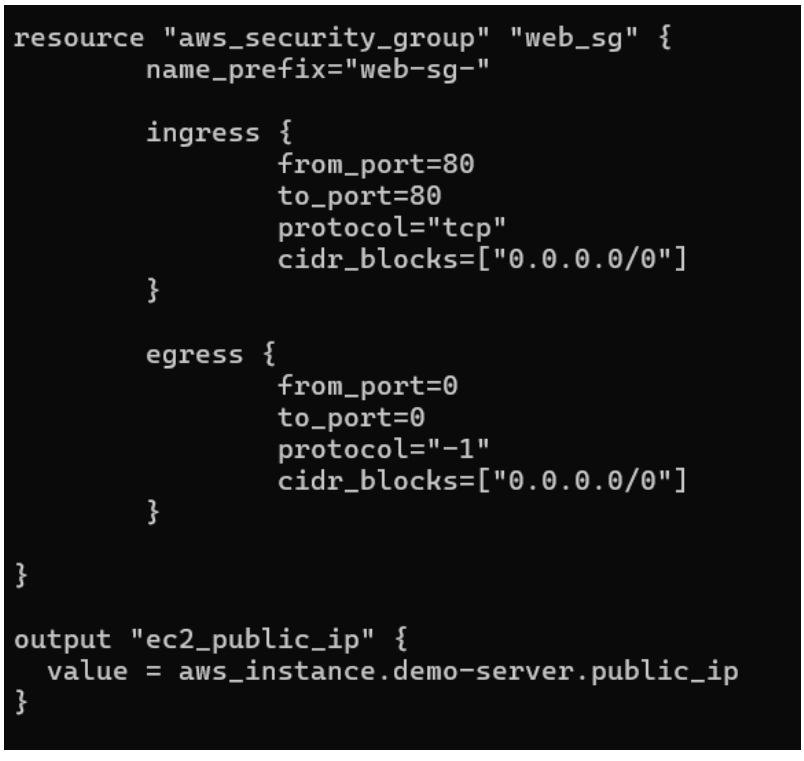
```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

ec2_public_ip = "54.221.62.223"
root@advika:~/terraform#
```

Had to destroy that instance because I couldn't modify security group permissions in place
Made a new instance called Tf_IAM

```
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_instance.demo-server: Creating...
aws_instance.demo-server: Still creating... [10s elapsed]
aws_instance.demo-server: Still creating... [20s elapsed]
aws_instance.demo-server: Still creating... [30s elapsed]
aws_instance.demo-server: Still creating... [40s elapsed]
aws_instance.demo-server: Creation complete after 42s [id=i-0e2fcff3e9e1f755
9]

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

Outputs:

ec2_public_ip = "3.81.15.249"
```

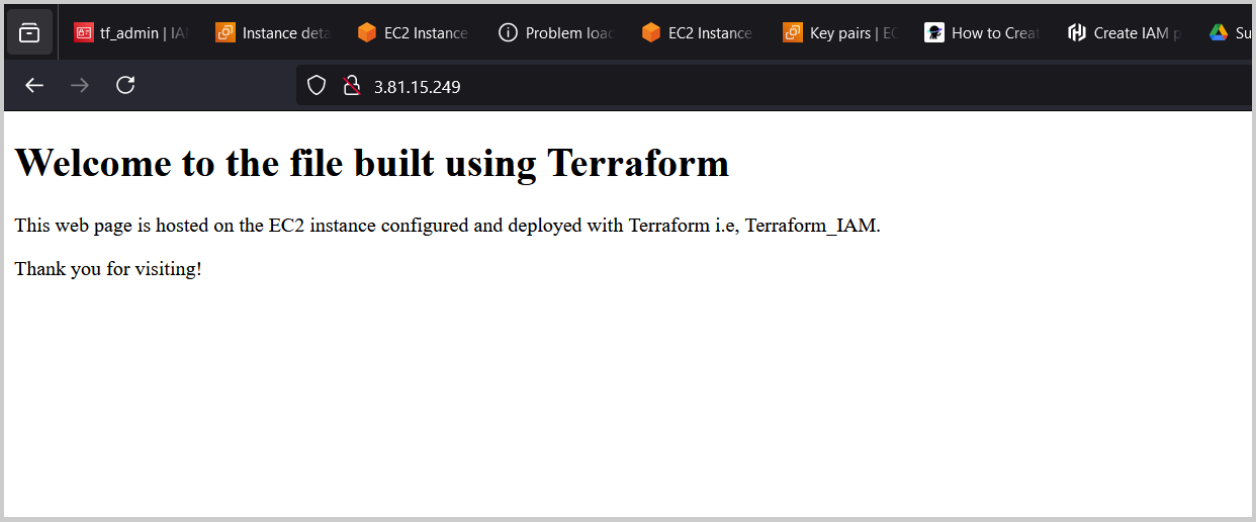
main.tf FILE FOR Tf_IAM creation:

```
ec2_public_ip = "3.81.15.249"
root@advika:~/terraform# cat main.tf
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "5.1.0"
    }
  }
}

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

data "local_file" "index_html" {
  filename = "/mnt/c/Users/advik/Downloads/index.html"
}


resource "aws_instance" "demo-server" {
  ami           = "ami-04b70fa74e45c3917"
  instance_type = "t2.micro"
  key_name      = "keyfor"

  security_groups=[aws_security_group.web_sg.name]


  user_data = <<-EOF
              #!/bin/bash
              apt-get update
              apt-get install -y apache2
              systemctl start apache2
              systemctl enable apache2
              echo '${data.local_file.index_html.content}' > /var/www/html
/index.html
              EOF


  tags = {
    Name = "Tf_IAM"
  }
}
```

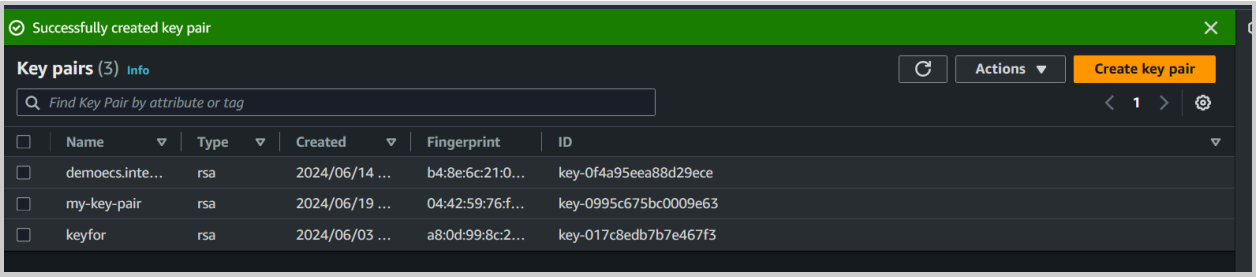


```
resource "aws_security_group" "web_sg" {
        name_prefix="web-sg-"

        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"]
        }

}

output "ec2_public_ip" {
  value = aws_instance.demo-server.public_ip
}
```

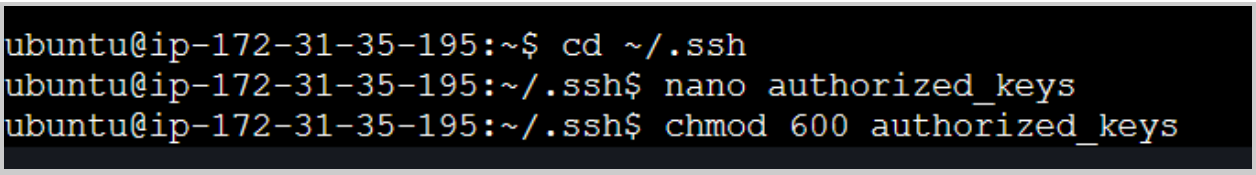
SUCCESSFUL IN HAVING THE EC2 INSTANCE DISPLAY THE INDEX.HTML FILE!



***HAD TO RECONFIGURE THE SECURITY RULES OF THE INSTANCE



Modified the key-pair being used by my current instance using AWS CLI



Edited the Security Rules within the AWS Management Console for EC2