Lab: Terraform Provisioners

Provisioners can be used to model specific actions on the local machine or on a remote machine in order to prepare servers or other infrastructure objects for service.

To this point the EC2 web server we have created is useless. We created a server without any running code with no useful services are running on it.

We will utilize Terraform provisoners to deploy a webapp onto the instance we've created. In order run these steps Terraform needs a connection block along with our generated SSH key from the previous labs in order to authenticate into our instance. Terraform can utilize both the `local-exec` provisioner to urn commands on our local workstation, and the `remote-exec` provisoner to install security updates along with our web application.

- Task 1: Upload your SSH keypair to AWS and associate to your instance.
- Task 2: Create a Security Group that allows SSH to your instance.
- Task 3: Create a connection block using your SSH keypair.
- Task 4: Use the `local-exec` provisioner to change permissions on your local SSH Key
- Task 5: Create a `remote-exec` provisioner block to pull down and install web application.
- Task 6: Apply your configuration and watch for the remote connection.
- Task 7: Pull up the web application and ssh into the web server (optional)

Task 1: Create an SSH keypair and associate it to your instance.

In `main.tf` add the following resource blocks to create a key pair in AWS that is associated with your generated key from the previous lab.

```
resource "aws_key_pair" "generated" {
  key_name   = "MyAWSKey"
  public_key = tls_private_key.generated.public_key_openssh

  lifecycle {
    ignore_changes = [key_name]
  }
}
```

```
terraform apply
```


Task 2: Create a Security Group that allows SSH to your instance.

Step 7.1.1

In `main.tf` add the following resource block to create a Security Group that allows SSH access.

```
# Security Groups

resource "aws_security_group" "ingress-ssh" {
  name   = "allow-all-ssh"
  vpc_id = aws_vpc.vpc.id
  ingress {
    cidr_blocks = [
      "0.0.0.0/0"
    ]
    from_port = 22
    to_port   = 22
    protocol  = "tcp"
  }
  // Terraform removes the default rule
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

In `main.tf` add the following resource block to create a Security Group that allows web traffic over the standard HTTP and HTTPS ports.

```
# Create Security Group - Web Traffic
resource "aws_security_group" "vpc-web" {
  name        = "vpc-web-${terraform.workspace}"
  vpc_id      = aws_vpc.vpc.id
  description = "Web Traffic"
  ingress {
    description = "Allow Port 80"
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    description = "Allow Port 443"
    from_port   = 443
    to_port     = 443
```

```
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    description = "Allow all ip and ports outbound"
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_security_group" "vpc-ping" {
  name        = "vpc-ping"
  vpc_id      = aws_vpc.vpc.id
  description = "ICMP for Ping Access"
  ingress {
    description = "Allow ICMP Traffic"
    from_port   = -1
    to_port     = -1
    protocol    = "icmp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    description = "Allow all ip and ports outboun"
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

Task 3: Create a connection block using your keypair module outputs.

Replace the aws_instance" "ubuntu_server" resource block in your `main.tf` with the code below to deploy and Ubuntu server, associate the AWS Key, Security Group and connection block for Terraform to connect to your instance:

```
resource "aws_instance" "ubuntu_server" {
  ami                         = data.aws_ami.ubuntu.id
  instance_type               = "t2.micro"
  subnet_id                   = aws_subnet.public_subnets["public_subnet_1
     "].id
  security_groups             = [aws_security_group.vpc-ping.id,
     aws_security_group.ingress-ssh.id, aws_security_group.vpc-web.id]
```

```
  associate_public_ip_address = true
  key_name                    = aws_key_pair.generated.key_name
  connection {
    user        = "ubuntu"
    private_key = tls_private_key.generated.private_key_pem
    host        = self.public_ip
  }

  tags = {
    Name = "Ubuntu EC2 Server"
  }

  lifecycle {
    ignore_changes = [security_groups]
  }

}
```

You will notice that we are referencing other resource blocks via Terraform interpolation syntax to associate the security group, keypair and private key for the connection to our instance. The value of `self` refers to the resource defined by the current block. So `self.public_ip` refers to the public IP address of our `aws_instance.web`.

Task 4: Use the `local-exec` provisioner to change permissions on your local SSH Key

The `local-exec` provisioner invokes a local executable after a resource is created. We will utilize a `local-exec` provisioner to make sure our private key is permissioned correctly. This invokes a process on the machine running Terraform, not on the resource.

Update the aws_instance" "ubuntu_server" resource block in your `main.tf` to call a `local-exec` provisioner:

```
# Leave the first part of the block unchanged and create our `local-exec`
   provisioner
  provisioner "local-exec" {
    command = "chmod 600 ${local_file.private_key_pem.filename}"
  }
```

Task 5: Create a `remote-exec` provisioner block to pull down web application.

The `remote-exec` provisioner runs remote commands on the instance provisoned with Terraform. We can use this provisioner to clone our web application code to the isntance and then invoke the

setup script.

```
  provisioner "remote-exec" {
    inline = [
      "sudo rm -rf /tmp",
      "sudo git clone https://github.com/hashicorp/demo-terraform-101 /tmp
         ",
      "sudo sh /tmp/assets/setup-web.sh",
    ]
  }
```

Make sure both provisioners are inside the *aws_instance* resource block.

Task 3: Apply your configuration and watch for the remote connection.

In order to create our security group, new web ubuntu instance with the associated public SSH Key and execute our `provisioners` we will validate our code and then initiate a `terraform apply`.

```
terraform validate
```

```
Success! The configuration is valid.
```

```
terraform apply
```

Upon running `terraform apply`, you should see new output which includes a connection to the EC2 instance

```
terraform apply
```

```
...
aws_instance.ubuntu_server: Provisioning with 'local-exec'...
aws_instance.ubuntu_server (local-exec): Executing: ["/bin/sh" "-c" "chmod
    600 MyAWSKey.pem"]
aws_instance.ubuntu_server: Provisioning with 'remote-exec'...
aws_instance.ubuntu_server (remote-exec): Connecting to remote host via
    SSH...
aws_instance.ubuntu_server (remote-exec):   Host: 3.236.92.141
aws_instance.ubuntu_server (remote-exec):   User: ubuntu
aws_instance.ubuntu_server (remote-exec):   Password: false
aws_instance.ubuntu_server (remote-exec):   Private key: true
aws_instance.ubuntu_server (remote-exec):   Certificate: false
aws_instance.ubuntu_server (remote-exec):   SSH Agent: true
aws_instance.ubuntu_server (remote-exec):   Checking Host Key: false
aws_instance.ubuntu_server (remote-exec):   Target Platform: unix
aws_instance.ubuntu_server (remote-exec): Connecting to remote host via
    SSH...
```

```
aws_instance.ubuntu_server (remote-exec):   Host: 3.236.92.141
aws_instance.ubuntu_server (remote-exec):   User: ubuntu
aws_instance.ubuntu_server (remote-exec):   Password: false
aws_instance.ubuntu_server (remote-exec):   Private key: true
aws_instance.ubuntu_server (remote-exec):   Certificate: false
aws_instance.ubuntu_server (remote-exec):   SSH Agent: true
aws_instance.ubuntu_server (remote-exec):   Checking Host Key: false
aws_instance.ubuntu_server (remote-exec):   Target Platform: unix
aws_instance.ubuntu_server (remote-exec): Connecting to remote host via
    SSH...
aws_instance.ubuntu_server (remote-exec):   Host: 3.236.92.141
aws_instance.ubuntu_server (remote-exec):   User: ubuntu
aws_instance.ubuntu_server (remote-exec):   Password: false
aws_instance.ubuntu_server (remote-exec):   Private key: true
aws_instance.ubuntu_server (remote-exec):   Certificate: false
aws_instance.ubuntu_server (remote-exec):   SSH Agent: true
aws_instance.ubuntu_server (remote-exec):   Checking Host Key: false
aws_instance.ubuntu_server (remote-exec):   Target Platform: unix
aws_instance.ubuntu_server (remote-exec): Connected!
aws_instance.ubuntu_server: Still creating... [50s elapsed]
aws_instance.ubuntu_server (remote-exec): Cloning into '/tmp'...
aws_instance.ubuntu_server (remote-exec): remote: Enumerating objects:
    417, done.
aws_instance.ubuntu_server (remote-exec): Receiving objects:   0% (1/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:   1% (5/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:   2% (9/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:   3% (13/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:   4% (17/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:   5% (21/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:   6% (26/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:   7% (30/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:   8% (34/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:   9% (38/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  10% (42/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  11% (46/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  12% (51/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  13% (55/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  14% (59/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  15% (63/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  16% (67/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  17% (71/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  18% (76/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  19% (80/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  20% (84/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  21% (88/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  22% (92/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  23% (96/417)
aws_instance.ubuntu_server (remote-exec): Receiving objects:  24%
    (101/417), 2.71 MiB | 5.37 MiB/s
```

```
aws_instance.ubuntu_server (remote-exec): Receiving objects:  25%
    (105/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  26%
    (109/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  27%
    (113/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  28%
    (117/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  29%
    (121/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  30%
    (126/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  31%
    (130/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  32%
    (134/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  33%
    (138/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  34%
    (142/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  35%
    (146/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  36%
    (151/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  37%
    (155/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  38%
    (159/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  39%
    (163/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  40%
    (167/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  41%
    (171/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  42%
    (176/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  43%
    (180/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  44%
    (184/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  45%
    (188/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  46%
    (192/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  47%
    (196/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): remote: Total 417 (delta 0),
    reused 0 (delta 0), pack-reused 417
aws_instance.ubuntu_server (remote-exec): Receiving objects:  48%
```

```
    (201/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  49%
    (205/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  50%
    (209/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  51%
    (213/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  52%
    (217/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  53%
    (222/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  54%
    (226/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  55%
    (230/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  56%
    (234/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  57%
    (238/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  58%
    (242/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  59%
    (247/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  60%
    (251/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  61%
    (255/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  62%
    (259/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  63%
    (263/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  64%
    (267/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  65%
    (272/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  66%
    (276/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  67%
    (280/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  68%
    (284/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  69%
    (288/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  70%
    (292/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  71%
    (297/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  72%
    (301/417), 2.71 MiB | 5.37 MiB/s
```

```
aws_instance.ubuntu_server (remote-exec): Receiving objects:  73%
    (305/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  74%
    (309/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  75%
    (313/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  76%
    (317/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  77%
    (322/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  78%
    (326/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  79%
    (330/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  80%
    (334/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  81%
    (338/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  82%
    (342/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  83%
    (347/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  84%
    (351/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  85%
    (355/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  86%
    (359/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  87%
    (363/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  88%
    (367/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  89%
    (372/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  90%
    (376/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  91%
    (380/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  92%
    (384/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  93%
    (388/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  94%
    (392/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  95%
    (397/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  96%
    (401/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  97%
```

```
    (405/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  98%
    (409/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects:  99%
    (413/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects: 100%
    (417/417), 2.71 MiB | 5.37 MiB/s
aws_instance.ubuntu_server (remote-exec): Receiving objects: 100%
    (417/417), 4.18 MiB | 5.76 MiB/s, done.
aws_instance.ubuntu_server (remote-exec): Resolving deltas:   0% (0/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:   2% (4/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:   3% (5/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:   4% (6/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:   5% (8/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:   6% (9/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:   7% (11/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  12% (18/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  13% (19/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  14% (20/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  15% (22/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  16% (23/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  17% (25/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  21% (31/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  25% (36/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  26% (37/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  30% (43/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  31% (45/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  34% (49/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  35% (50/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  39% (56/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  41% (59/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  42% (60/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  44% (63/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  48% (69/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  49% (70/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  63% (90/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  64% (91/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  66% (95/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  68% (97/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  69% (98/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  76% (108/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  80% (115/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  81% (116/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  82% (117/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  84% (120/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  88% (125/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  94% (134/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  95% (135/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas:  97% (139/142)
```

```
aws_instance.ubuntu_server (remote-exec): Resolving deltas: 100% (142/142)
aws_instance.ubuntu_server (remote-exec): Resolving deltas: 100% (142/142)
    , done.
aws_instance.ubuntu_server (remote-exec): Created symlink /etc/systemd/
    system/multi-user.target.wants/webapp.service -> /lib/systemd/system/
    webapp.service.
aws_instance.ubuntu_server: Creation complete after 54s [id=i-021
    cf7ae83ee067d1]
...
```

Task 7: Pull up the web application and ssh into the web server (optional)

You can now visit your web application by pointing your browser at the public_ip output for your EC2 instance. To get that address you can look at the state details of the EC2 instance by performing a `terraform state show aws_instance.ubuntu_server`

```
terraform state show aws_instance.ubuntu_server
```

```
resource "aws_instance" "ubuntu_server" {
    ami                                  = "ami-0964546d3da97e3ab"
    arn                                  = "arn:aws:ec2:us-west
        -2:508140242758:instance/i-00eccad2a464a4aa3"
    associate_public_ip_address          = true
    availability_zone                    = "us-west-2b"
    cpu_core_count                       = 1
    cpu_threads_per_core                 = 1
    disable_api_termination              = false
    ebs_optimized                        = false
    get_password_data                    = false
    hibernation                          = false
    id                                   = "i-00eccad2a464a4aa3"
    instance_initiated_shutdown_behavior = "stop"
    instance_state                       = "running"
    instance_type                        = "t2.micro"
    ipv6_address_count                   = 0
    ipv6_addresses                       = []
    key_name                             = "MyAWSKey"
    monitoring                           = false
    primary_network_interface_id         = "eni-00e236032e4f38e95"
    private_dns                          = "ip-10-0-101-238.us-west-2.
        compute.internal"
    private_ip                           = "10.0.101.238"
    public_ip                            = "35.86.144.200"
    secondary_private_ips                = []
    security_groups                      = [
        "sg-068db6e720cb80a46",
```

```
        "sg-0dbb6b4429d7730f2",
        "sg-0f64195ac2bfee1f2",
    ]
    source_dest_check                    = true
    subnet_id                            = "subnet-03977b3f439ccc2cb"
    tags                                 = {
        "Name" = "Ubuntu EC2 Server"
    }
    tags_all                             = {
        "Name"       = "Ubuntu EC2 Server"
        "Owner"      = "Acme"
        "Provisoned" = "Terraform"
    }
    tenancy                              = "default"
    vpc_security_group_ids               = [
        "sg-068db6e720cb80a46",
        "sg-0dbb6b4429d7730f2",
        "sg-0f64195ac2bfee1f2",
    ]

    capacity_reservation_specification {
        capacity_reservation_preference = "open"
    }

    credit_specification {
        cpu_credits = "standard"
    }

    enclave_options {
        enabled = false
    }

    metadata_options {
        http_endpoint               = "enabled"
        http_put_response_hop_limit = 1
        http_tokens                 = "optional"
    }

    root_block_device {
        delete_on_termination = true
        device_name           = "/dev/sda1"
        encrypted             = false
        iops                  = 100
        tags                  = {}
        throughput            = 0
        volume_id             = "vol-06c0a4100fe14914c"
        volume_size           = 8
        volume_type           = "gp2"
    }
```


```
}
```

Visit `http://<public_ip>`

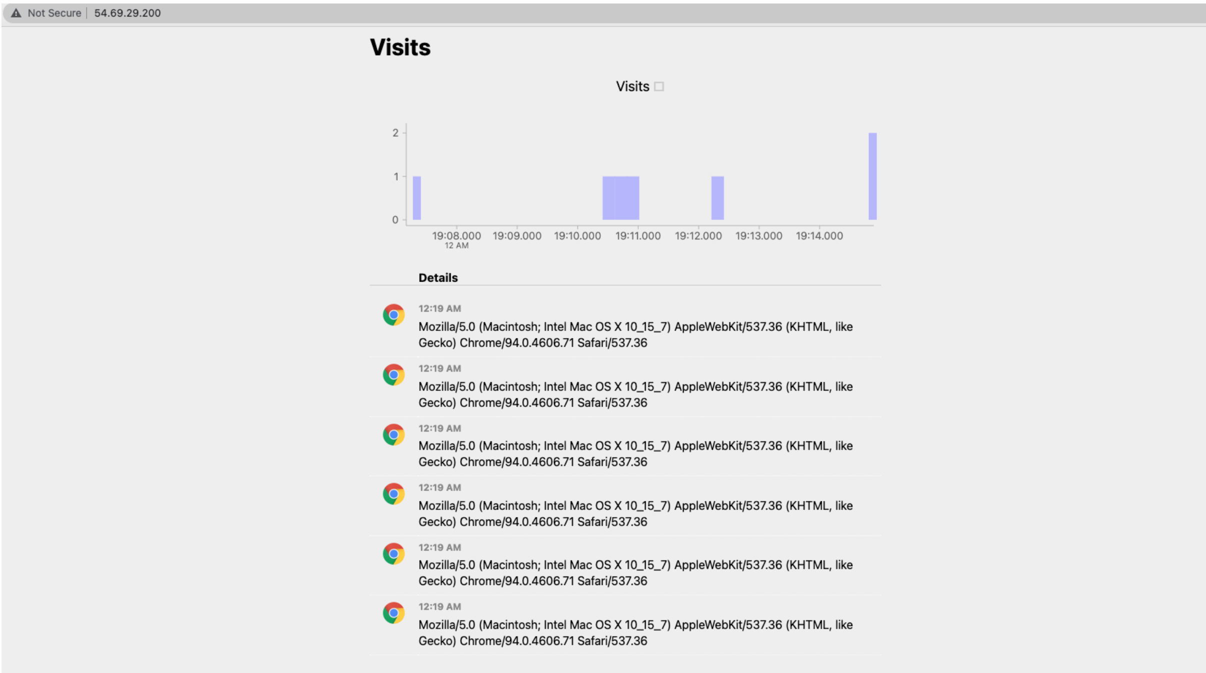

Figure 1: Web Application

Optional

If you want, you can also ssh to your EC2 instance with a command like `ssh -i MyAWSKey.pem ubuntu@<public_ip>`. Type yes when prompted to use the key. Type `exit` to quit the ssh session.

```
ssh -i MyAWSKey.pem ubuntu@<public_ip>
```

```
ssh -i MyAWSKey.pem ubuntu@54.69.29.200
The authenticity of host '54.69.29.200 (54.69.29.200)' can't be
    established.
ECDSA key fingerprint is SHA256:
    OgKh9TuNNuyFFBT96oiZbYTGhvtZKAoLOIcFgLw7Niw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.69.29.200' (ECDSA) to the list of known
    hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1019-aws x86_64)

 * Documentation:  https://help.ubuntu.com
```

```
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Oct 13 04:22:02 UTC 2021

  System load:  0.0               Processes:             102
  Usage of /:   18.4% of 7.69GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 10.0.101.134
  Swap usage:   0%


1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed Oct 13 04:17:21 2021 from 44.197.238.120
ubuntu@ip-10-0-101-134:~$ exit
logout
Connection to 54.69.29.200 closed.
```