ICT & Infra S3 Automation & Orchestration, week 4

Class:	CB01
Student number:	4961854
Student name:	Heiko Morales Aloria

Introduction

This week you will practice provisioning an infrastructure using Terraform configuration file. Additionally, Ansible Playbook(s) must be used to automate a complex process. Before executing the assignment, ensure that you have working Ansible control node. For this assignment, you must be familiar with preparing Apache server and hosting Flask application using Ansible.

This is **group** assignment.

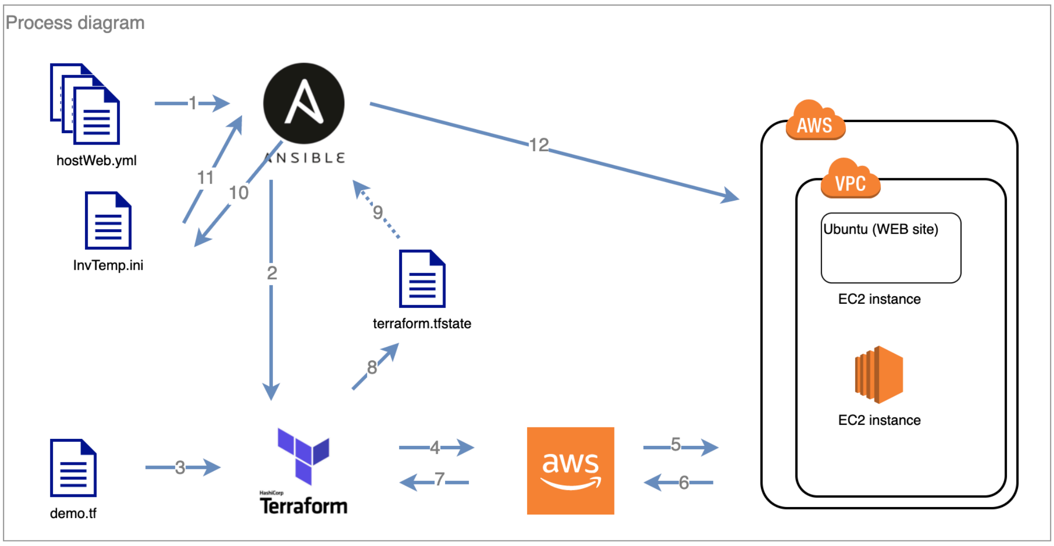
Assignment 1. Create a Terraform config file to provision an infrastructure

Difficulty: ★★★★★.

During the Week 4 lesson you saw an example how to use Terraform to provision a simple infrastructure. In the demo, we had no time to host a website. Follow the tutorial at https://learn.hashicorp.com/tutorials/terraform/infrastructure-as-code and complete all the steps to install and configure Terraform. Practise to build, change and destroy an infrastructure. Additionally, practise to define Input Variables and query Data with Outputs.

Finally, use Ansible to host a flask website, as you might already did in in Week 3 homework assignment.

The following process diagram shows possible process for your solution:

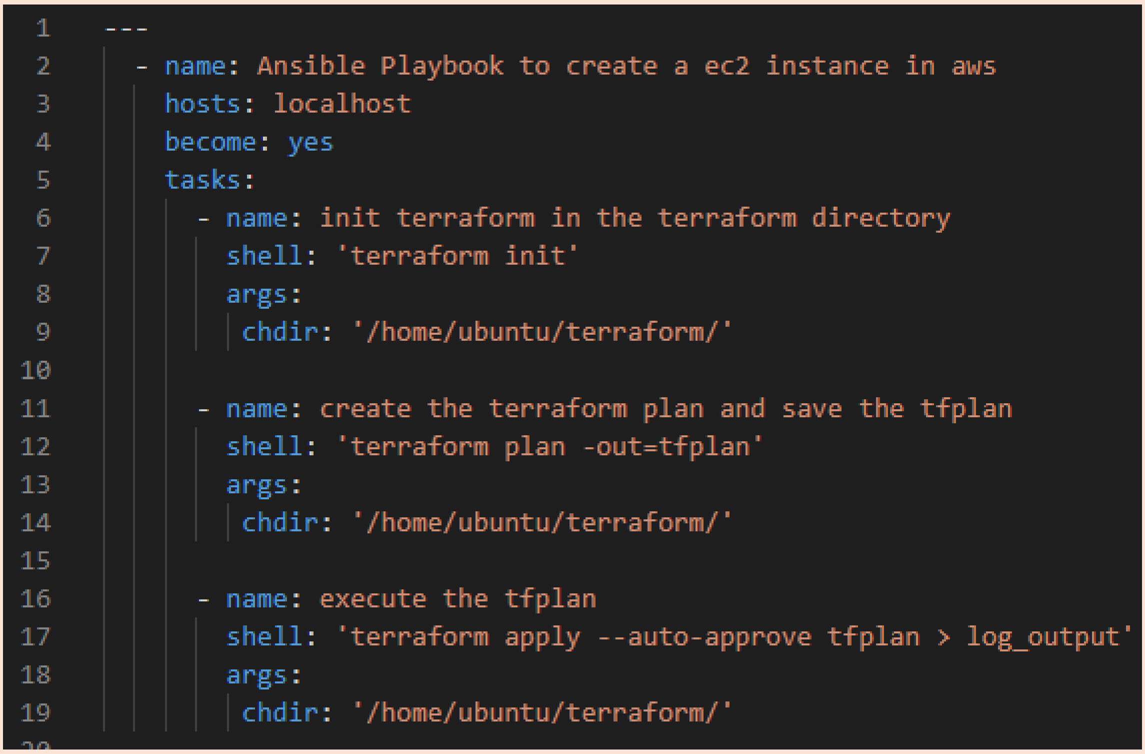


Provide screenshots (evidence) for your solution. Always explain your evidence! As a prof, we expect at least:

- A screenshot of running EC2 Ubuntu instance.
- Ansible Playbook file(s) that automates the required process.
- Ansible output after executing the playbook.
- A prof that the website is working.

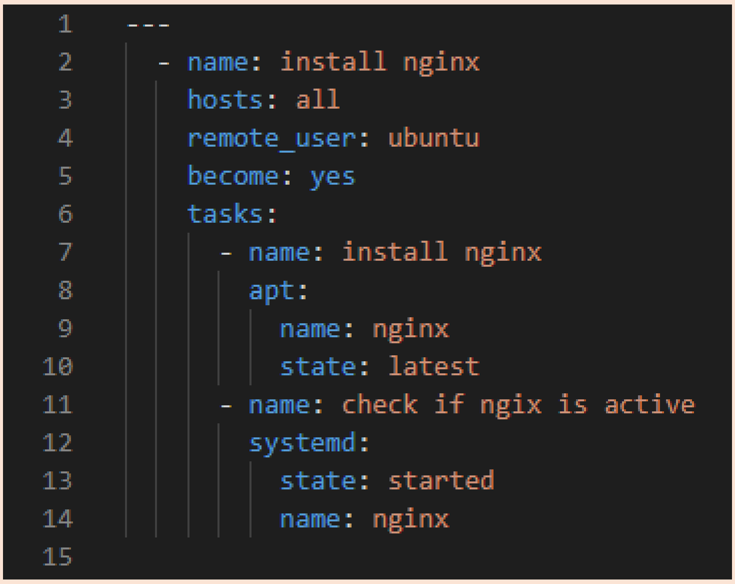
Solution:

To do this task we started by creating a first playbook to automate the first part of executing the commands to start and run terraform.



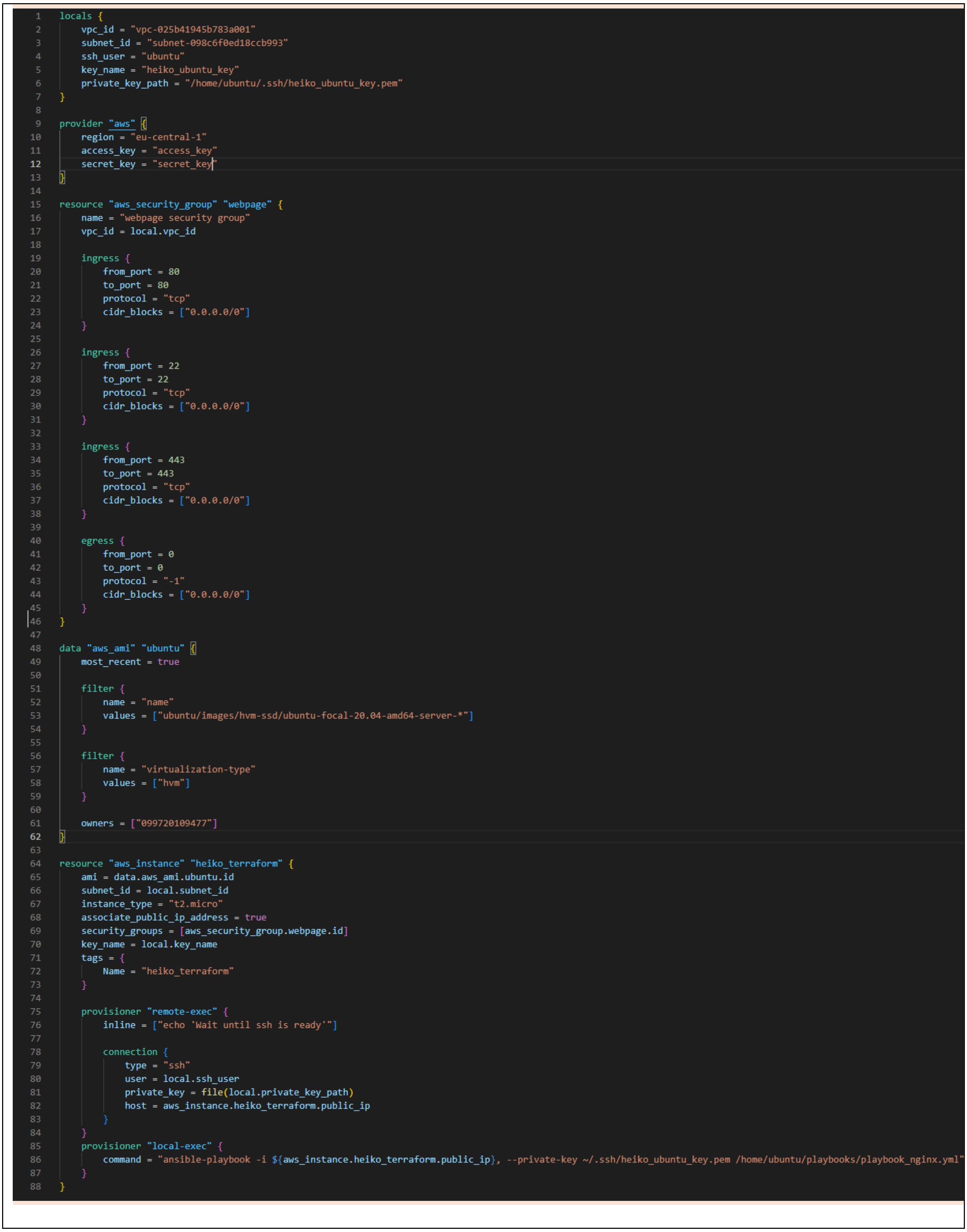
```
---
  - name: Ansible Playbook to create a ec2 instance in aws
    hosts: localhost
    become: yes
    tasks:
      - name: init terraform in the terraform directory
        shell: 'terraform init'
        args:
         chdir: '/home/ubuntu/terraform/'

      - name: create the terraform plan and save the tfplan
        shell: 'terraform plan -out=tfplan'
        args:
         chdir: '/home/ubuntu/terraform/'

      - name: execute the tfplan
        shell: 'terraform apply --auto-approve tfplan > log_output'
        args:
         chdir: '/home/ubuntu/terraform/'
```

Later we create a new, simpler playbook that installs nginx.



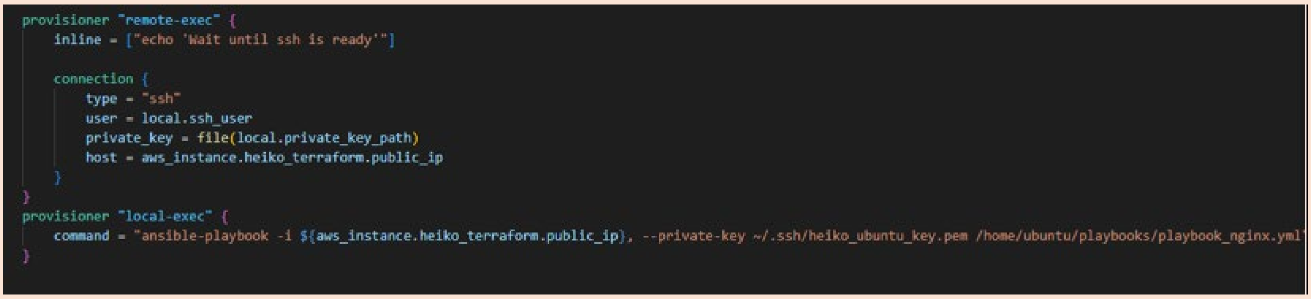
```
---
  - name: install nginx
    hosts: all
    remote_user: ubuntu
    become: yes
    tasks:
      - name: install nginx
        apt:
          name: nginx
          state: latest
      - name: check if ngix is active
        systemd:
          state: started
          name: nginx
```

Next, we generate a terraform file which creates an instance in aws with ports 80 and 443 open in order to access the web page that will be hosted on this server. Likewise, the instance that we create is configured to be an ubuntu with public ip.



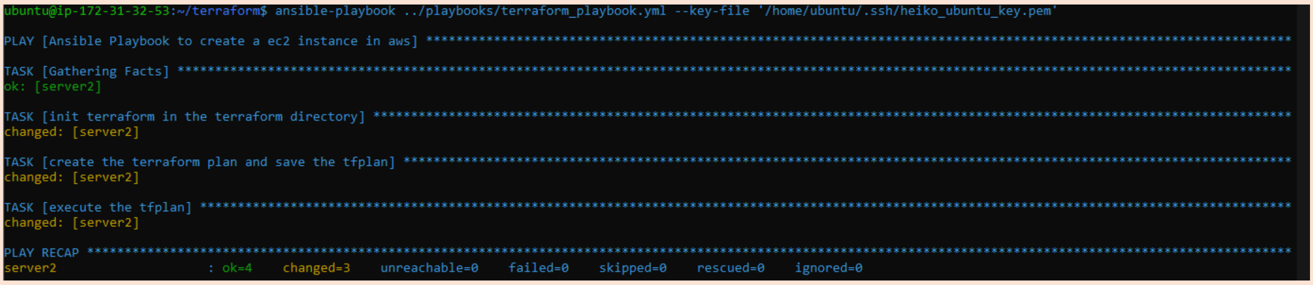
```
locals {
    vpc_id = "vpc-025b41945b783a001"
    subnet_id = "subnet-098c6f0ed18ccb993"
    ssh_user = "ubuntu"
    key_name = "heiko_ubuntu_key"
    private_key_path = "/home/ubuntu/.ssh/heiko_ubuntu_key.pem"
}

provider "aws" {
    region = "eu-central-1"
    access_key = "access_key"
    secret_key = "secret_key"
}

resource "aws_security_group" "webpage" {
    name = "webpage security group"
    vpc_id = local.vpc_id

    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"]
    }

    ingress {
        from_port = 443
        to_port = 443
        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"]
    }
}

data "aws_ami" "ubuntu" {
    most_recent = true

    filter {
        name = "name"
        values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*"]
    }

    filter {
        name = "virtualization-type"
        values = ["hvm"]
    }

    owners = ["099720109477"]
}

resource "aws_instance" "heiko_terraform" {
    ami = data.aws_ami.ubuntu.id
    subnet_id = local.subnet_id
    instance_type = "t2.micro"
    associate_public_ip_address = true
    security_groups = [aws_security_group.webpage.id]
    key_name = local.key_name
    tags = {
        Name = "heiko_terraform"
    }

    provisioner "remote-exec" {
        inline = ["echo 'Wait until ssh is ready'"]

        connection {
            type = "ssh"
            user = local.ssh_user
            private_key = file(local.private_key_path)
            host = aws_instance.heiko_terraform.public_ip
        }
    }
    provisioner "local-exec" {
        command = "ansible-playbook -i ${aws_instance.heiko_terraform.public_ip}, --private-key ~/.ssh/heiko_ubuntu_key.pem /home/ubuntu/playbooks/playbook_nginx.yml"
    }
}
```

The following part of the code is worth noting:



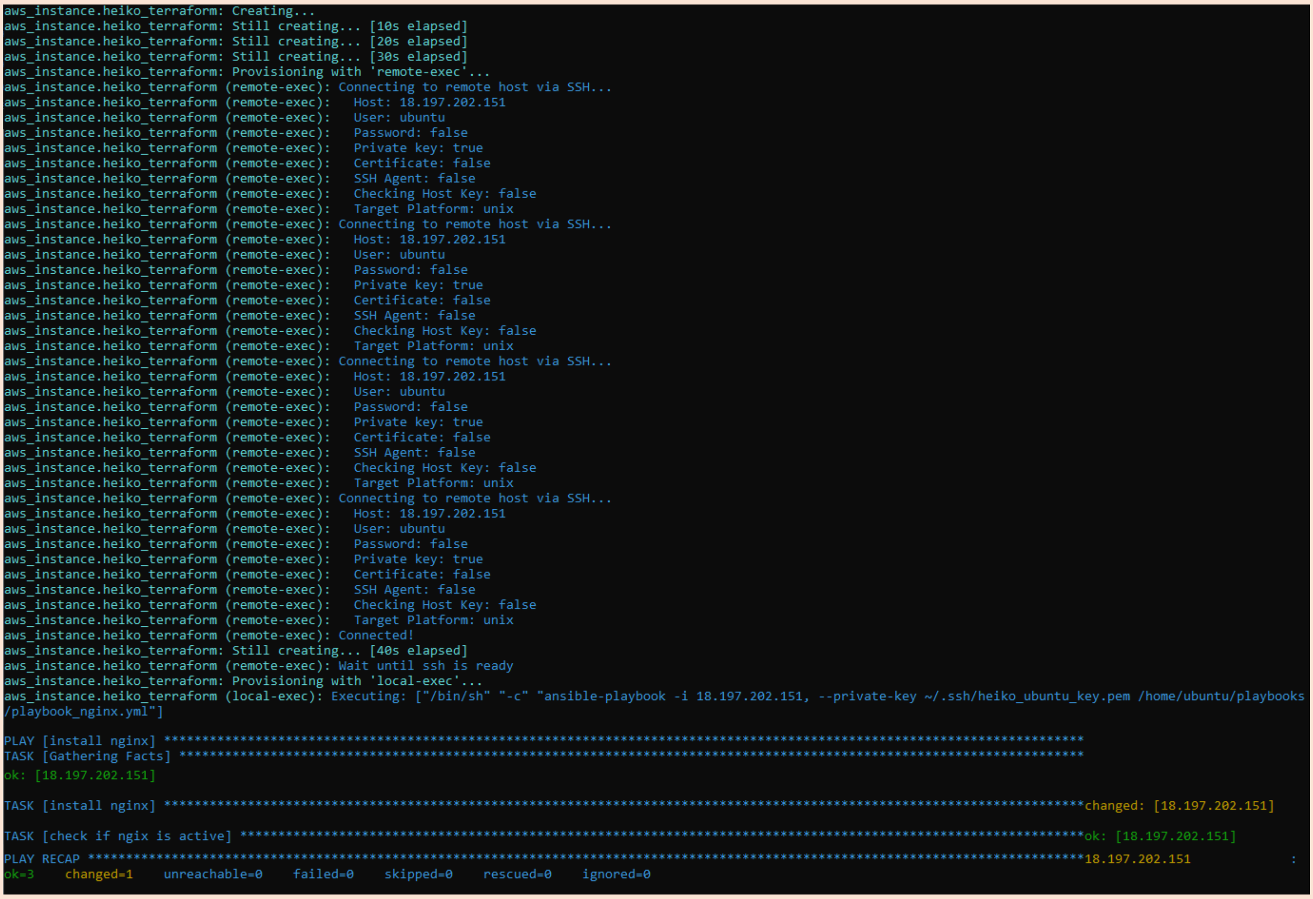
```
provisioner "remote-exec" {
    inline = ["echo 'Wait until ssh is ready'"]

    connection {
        type = "ssh"
        user = local.ssh_user
        private_key = file(local.private_key_path)
        host = aws_instance.heiko_terraform.public_ip
    }
}
provisioner "local-exec" {
    command = "ansible-playbook -i ${aws_instance.heiko_terraform.public_ip}, --private-key ~/.ssh/heiko_ubuntu_key.pem /home/ubuntu/playbooks/playbook_nginx.yml
}
```

In this part we force terraform to try to connect via ssh to the newly created instance so that we know exactly when the new instance is available. At that point terraform will run an ansible playbook with the public ip of the new instance so it can install nginx.



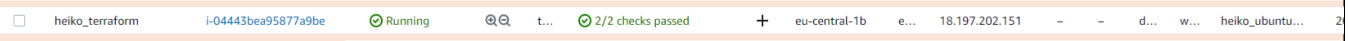
```
ubuntu@ip-172-31-32-53:~/terraform$ ansible-playbook ../playbooks/terraform_playbook.yml --key-file '/home/ubuntu/.ssh/heiko_ubuntu_key.pem'

PLAY [Ansible Playbook to create a ec2 instance in aws] *********************************************************************************************

TASK [Gathering Facts] ******************************************************************************************************************************
ok: [server2]

TASK [init terraform in the terraform directory] ****************************************************************************************************
changed: [server2]

TASK [create the terraform plan and save the tfplan] ************************************************************************************************
changed: [server2]

TASK [execute the tfplan] ***************************************************************************************************************************
changed: [server2]

PLAY RECAP ******************************************************************************************************************************************
server2                    : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

The playbook runs perfectly.



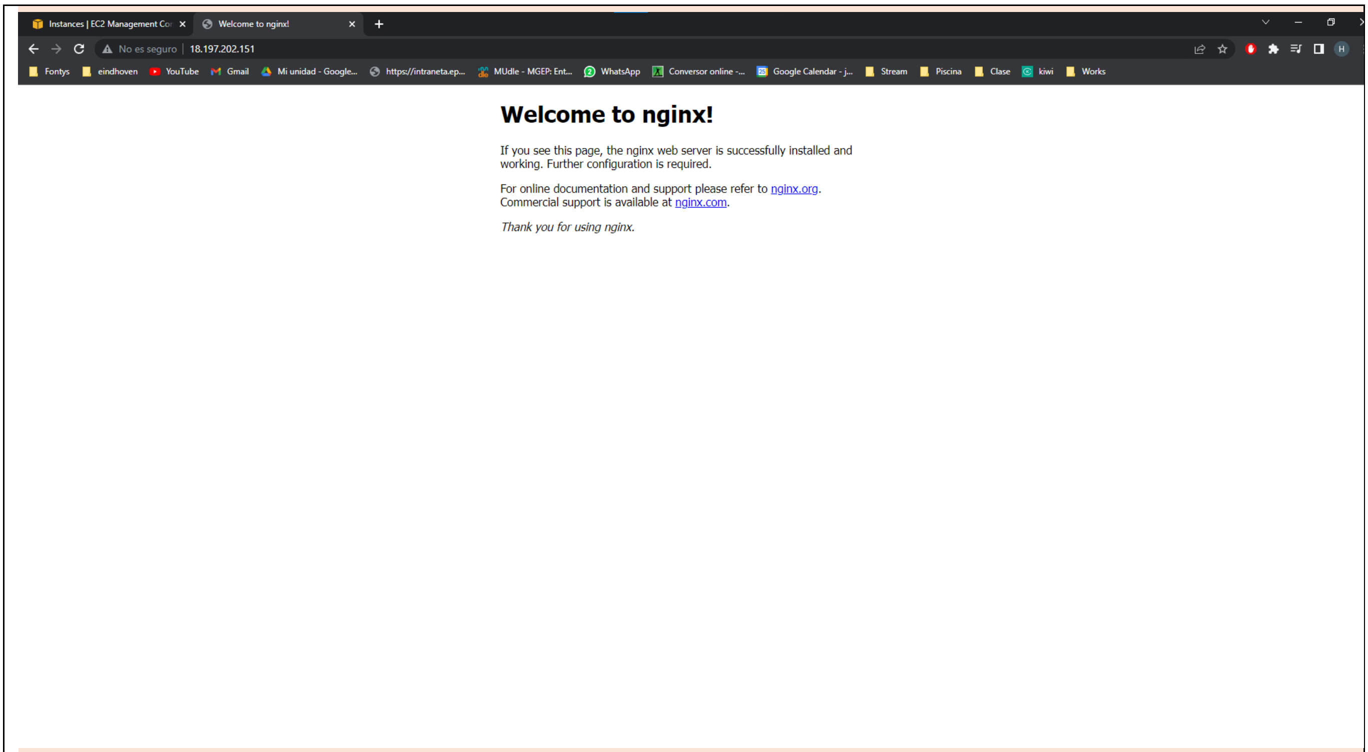
```
aws_instance.heiko_terraform: Creating...
aws_instance.heiko_terraform: Still creating... [10s elapsed]
aws_instance.heiko_terraform: Still creating... [20s elapsed]
aws_instance.heiko_terraform: Still creating... [30s elapsed]
aws_instance.heiko_terraform: Provisioning with 'remote-exec'...
aws_instance.heiko_terraform (remote-exec): Connecting to remote host via SSH...
aws_instance.heiko_terraform (remote-exec):   Host: 18.197.202.151
aws_instance.heiko_terraform (remote-exec):   User: ubuntu
aws_instance.heiko_terraform (remote-exec):   Password: false
aws_instance.heiko_terraform (remote-exec):   Private key: true
aws_instance.heiko_terraform (remote-exec):   Certificate: false
aws_instance.heiko_terraform (remote-exec):   SSH Agent: false
aws_instance.heiko_terraform (remote-exec):   Checking Host Key: false
aws_instance.heiko_terraform (remote-exec):   Target Platform: unix
aws_instance.heiko_terraform (remote-exec): Connecting to remote host via SSH...
aws_instance.heiko_terraform (remote-exec):   Host: 18.197.202.151
aws_instance.heiko_terraform (remote-exec):   User: ubuntu
aws_instance.heiko_terraform (remote-exec):   Password: false
aws_instance.heiko_terraform (remote-exec):   Private key: true
aws_instance.heiko_terraform (remote-exec):   Certificate: false
aws_instance.heiko_terraform (remote-exec):   SSH Agent: false
aws_instance.heiko_terraform (remote-exec):   Checking Host Key: false
aws_instance.heiko_terraform (remote-exec):   Target Platform: unix
aws_instance.heiko_terraform (remote-exec): Connecting to remote host via SSH...
aws_instance.heiko_terraform (remote-exec):   Host: 18.197.202.151
aws_instance.heiko_terraform (remote-exec):   User: ubuntu
aws_instance.heiko_terraform (remote-exec):   Password: false
aws_instance.heiko_terraform (remote-exec):   Private key: true
aws_instance.heiko_terraform (remote-exec):   Certificate: false
aws_instance.heiko_terraform (remote-exec):   SSH Agent: false
aws_instance.heiko_terraform (remote-exec):   Checking Host Key: false
aws_instance.heiko_terraform (remote-exec):   Target Platform: unix
aws_instance.heiko_terraform (remote-exec): Connecting to remote host via SSH...
aws_instance.heiko_terraform (remote-exec):   Host: 18.197.202.151
aws_instance.heiko_terraform (remote-exec):   User: ubuntu
aws_instance.heiko_terraform (remote-exec):   Password: false
aws_instance.heiko_terraform (remote-exec):   Private key: true
aws_instance.heiko_terraform (remote-exec):   Certificate: false
aws_instance.heiko_terraform (remote-exec):   SSH Agent: false
aws_instance.heiko_terraform (remote-exec):   Checking Host Key: false
aws_instance.heiko_terraform (remote-exec):   Target Platform: unix
aws_instance.heiko_terraform (remote-exec): Connected!
aws_instance.heiko_terraform: Still creating... [40s elapsed]
aws_instance.heiko_terraform (remote-exec): Wait until ssh is ready
aws_instance.heiko_terraform: Provisioning with 'local-exec'...
aws_instance.heiko_terraform (local-exec): Executing: ["/bin/sh" "-c" "ansible-playbook -i 18.197.202.151, --private-key ~/.ssh/heiko_ubuntu_key.pem /home/ubuntu/playbooks
/playbook_nginx.yml"]

PLAY [install nginx] ************************************************************************************************************
TASK [Gathering Facts] **********************************************************************************************************
ok: [18.197.202.151]

TASK [install nginx] ************************************************************************************************************changed: [18.197.202.151]

TASK [check if ngix is active] **************************************************************************************************ok: [18.197.202.151]
PLAY RECAP **********************************************************************************************************************18.197.202.151             :
ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

As we can see the result is successful.



□	heiko_terraform	i-04443bea95877a9be	✓ Running	⊕⊖	t...	✓ 2/2 checks passed	+	eu-central-1b	e...	18.197.202.151	–	–	d...	w...	heiko_ubuntu...

The virtual machine is created.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

you can access the new website