Final project

Project Description:

Description:

Nowadays, infrastructure automation is critical. We tend to put the most emphasis on software development processes, but infrastructure deployment strategy is just as important. Infrastructure automation not only aids disaster recovery, but it also facilitates testing and development.

Your organization is adopting the DevOps methodology and in order to automate provisioning of infrastructure there's a need to setup a centralised server for Jenkins.

Terraform is a tool that allows you to provision various infrastructure components. Ansible is a platform for managing configurations and deploying applications. It means you'll use Terraform to build a virtual machine, for example, and then use Ansible to instal the necessary applications on that machine.

Considering the Organizational requirement you are asked to automate the infrastructure using Terraform first and install other required automation tools in it.

Tools required: Terraform, AWS account with security credentials, Keypair

Expected Deliverables:

Launch an EC2 instance using Terraform

Connect to the instance

Install Jenkins, Java, and Python in the instance

Solution:

terraform script .tf format

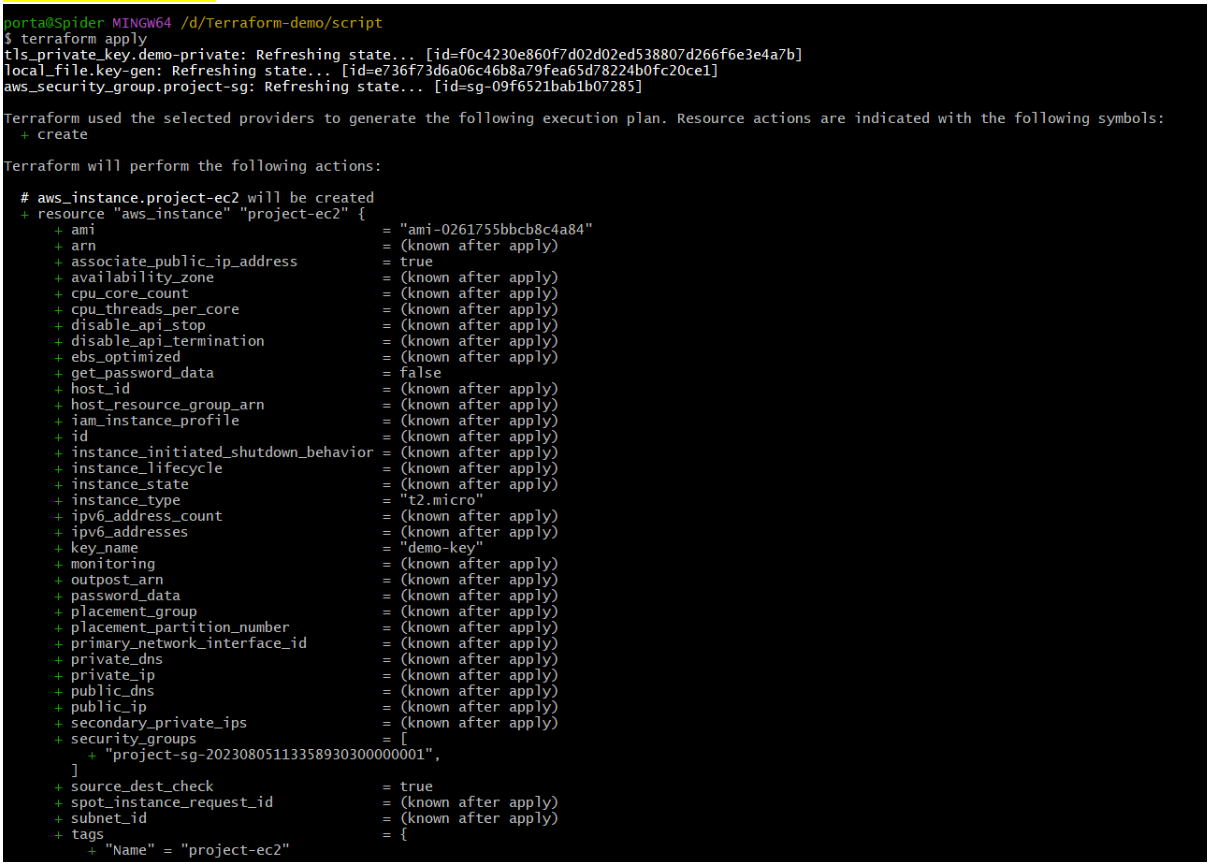
```
provider "aws" {
  region     = "us-east-1"  # North Virginia
  access_key = ""
  secret_key = ""
  token = "”

resource "tls_private_key" "demo-private" {
  algorithm = "RSA"
  rsa_bits  = 4096
}

resource "aws_security_group" "project-sg" {
  name_prefix = "project-sg-"
  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"]
  }

  ingress {
    description = "http"
    from_port   = 8080
    to_port     = 8080
    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 = "project-sg-tag"
  }
}

resource "aws_key_pair" "demo-key" {
  key_name   = "demo-key"
  public_key = tls_private_key.demo-private.public_key_openssh
}


resource "aws_instance" "project-ec2" {
  ami           = "ami-0261755bbcb8c4a84"
  associate_public_ip_address = true
  instance_type = "t2.micro"
  key_name      = "demo-key"
  security_groups = [aws_security_group.project-sg.name]
```

```
 depends_on  = [aws_key_pair.demo-key]

 tags = {
   Name = "project-ec2"
 }
}

resource "local_file" "key-gen" {
 content        = tls_private_key.demo-private.private_key_pem
 filename        = "demo-key.pem"
 file_permission = "0400"
}
```

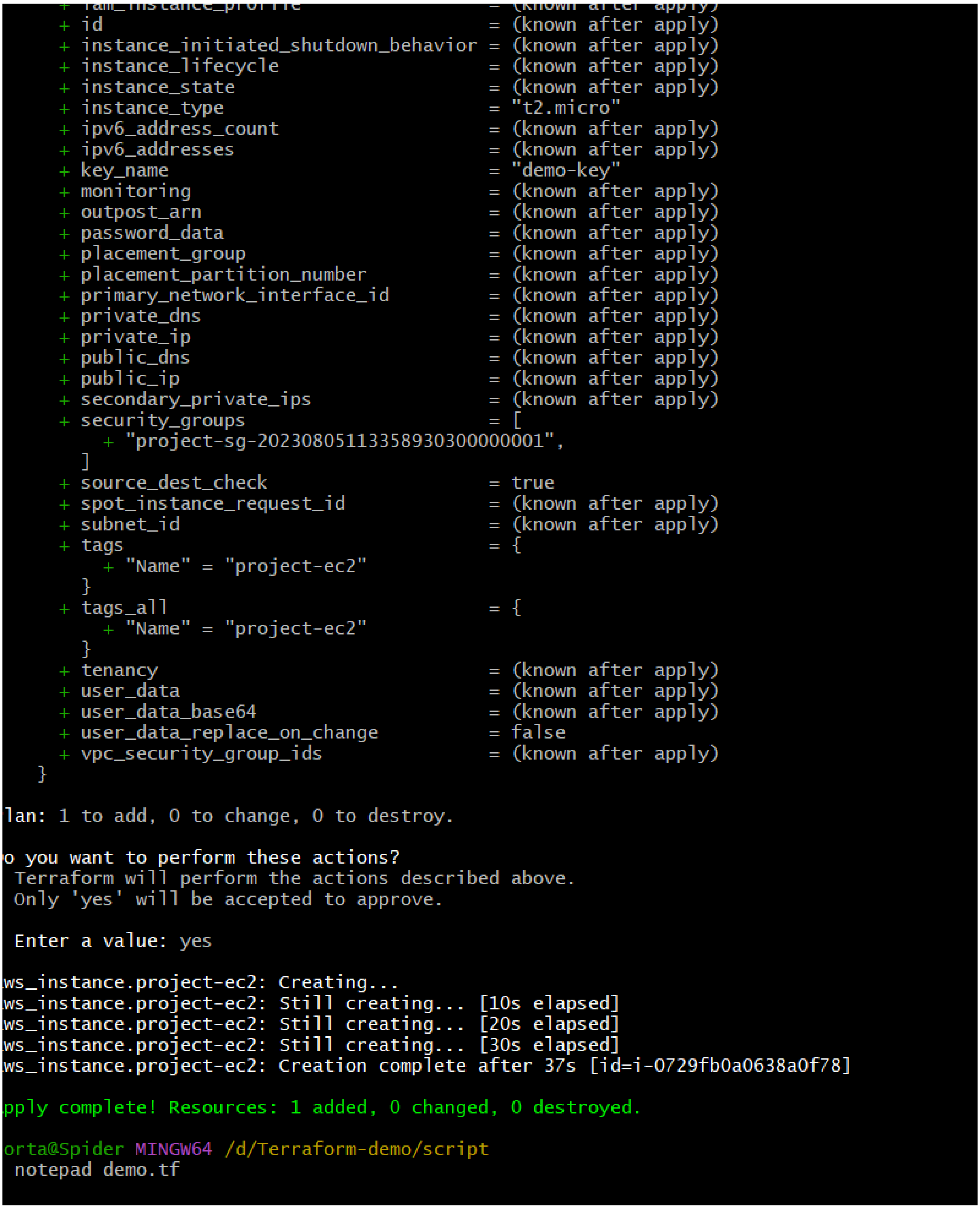
Command used for terraform

- notepad.demo.tf – open the script file
- terraform init – To in initiated provider
- terraform plan- to check plan
- terraform validate – to validate the script
- Terraform- apply – to apply the changes
- Terraform fmt – to set the format
- Terraform destroy- to delete all existing infrastructure.
- Terraform providers - to check initiated providers.

Script execution



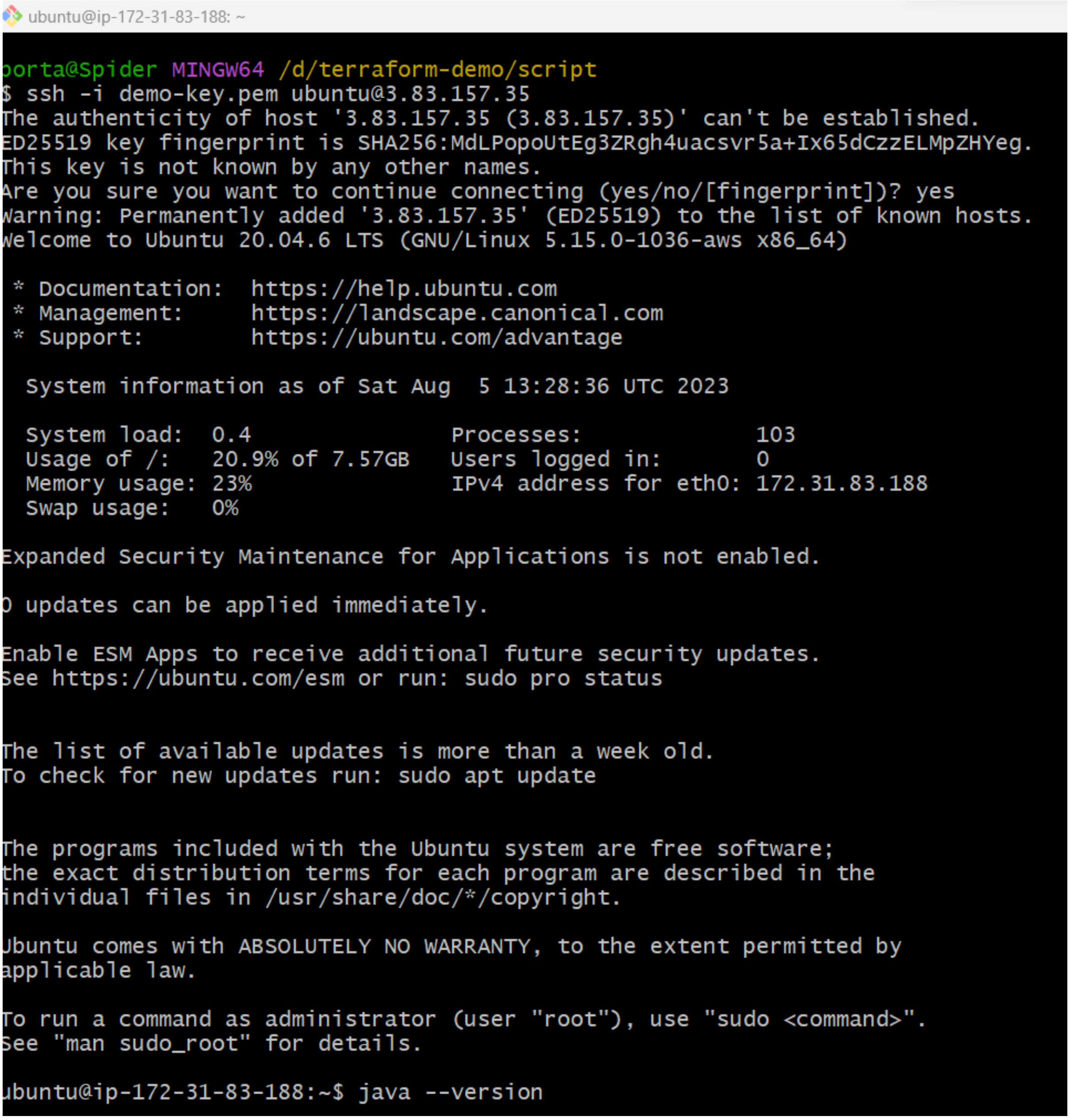
```
porta@Spider MINGW64 /d/Terraform-demo/script
$ terraform apply
tls_private_key.demo-private: Refreshing state... [id=f0c4230e860f7d02d02ed538807d266f6e3e4a7b]
local_file.key-gen: Refreshing state... [id=e736f73d6a06c46b8a79fea65d78224b0fc20ce1]
aws_security_group.project-sg: Refreshing state... [id=sg-09f6521bab1b07285]

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.project-ec2 will be created
  + resource "aws_instance" "project-ec2" {
      + ami                                  = "ami-0261755bbcb8c4a84"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "demo-key"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = [
          + "project-sg-20230805113358930300000001",
        ]
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "project-ec2"
```



```
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "demo-key"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = [
          + "project-sg-20230805113358930300000001",
        ]
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "project-ec2"
        }
      + tags_all                             = {
          + "Name" = "project-ec2"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

lan: 1 to add, 0 to change, 0 to destroy.

o you want to perform these actions?
 Terraform will perform the actions described above.
 Only 'yes' will be accepted to approve.

 Enter a value: yes

ws_instance.project-ec2: Creating...
ws_instance.project-ec2: Still creating... [10s elapsed]
ws_instance.project-ec2: Still creating... [20s elapsed]
ws_instance.project-ec2: Still creating... [30s elapsed]
ws_instance.project-ec2: Creation complete after 37s [id=i-0729fb0a0638a0f78]

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

orta@Spider MINGW64 /d/Terraform-demo/script
 notepad demo.tf
```

Connected with EC2-instance

Command using
ssh-I demo-key.pem ubunti@public_ip



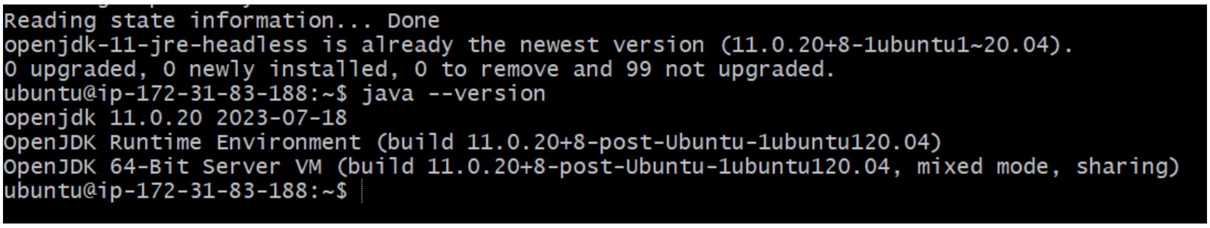
ubuntu@ip-172-31-83-188: ~

```
porta@Spider MINGW64 /d/terraform-demo/script
$ ssh -i demo-key.pem ubuntu@3.83.157.35
The authenticity of host '3.83.157.35 (3.83.157.35)' can't be established.
ED25519 key fingerprint is SHA256:MdLPopoUtEg3ZRgh4uacsvr5a+Ix65dCzzELMpZHYeg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.83.157.35' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1036-aws x86_64)

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

  System information as of Sat Aug  5 13:28:36 UTC 2023

  System load:  0.4               Processes:             103
  Usage of /:   20.9% of 7.57GB   Users logged in:       0
  Memory usage: 23%               IPv4 address for eth0: 172.31.83.188
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


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


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-83-188:~$ java --version
```

How to install Java

inside instance, we need to run the below commands

- sudo apt update – to check latest update
- sudo apt install openjdk-11-jre-headless – commands to install Java
- Java –version – to check installed Java

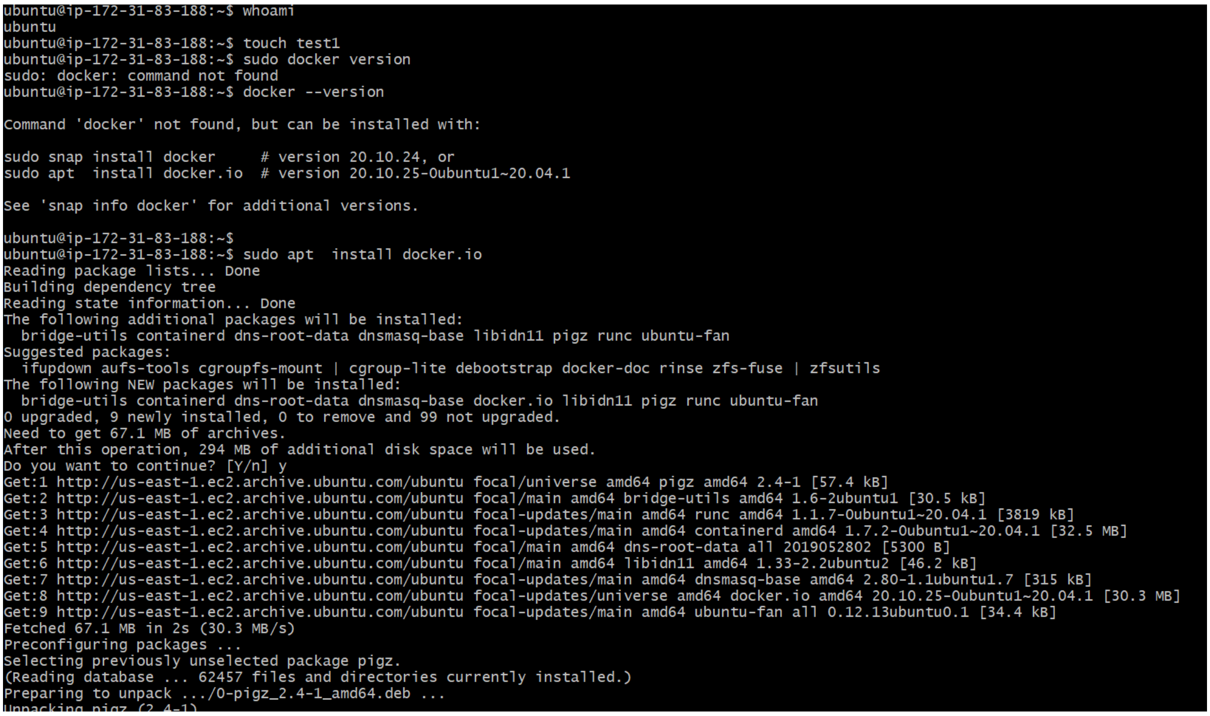


```
Reading state information... Done
openjdk-11-jre-headless is already the newest version (11.0.20+8-1ubuntu1~20.04).
0 upgraded, 0 newly installed, 0 to remove and 99 not upgraded.
ubuntu@ip-172-31-83-188:~$ java --version
openjdk 11.0.20 2023-07-18
OpenJDK Runtime Environment (build 11.0.20+8-post-Ubuntu-1ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.20+8-post-Ubuntu-1ubuntu120.04, mixed mode, sharing)
ubuntu@ip-172-31-83-188:~$
```

Docker installation

- docker –version – to check docker version if its installed
- sudo apt install docker.io- Docker installation

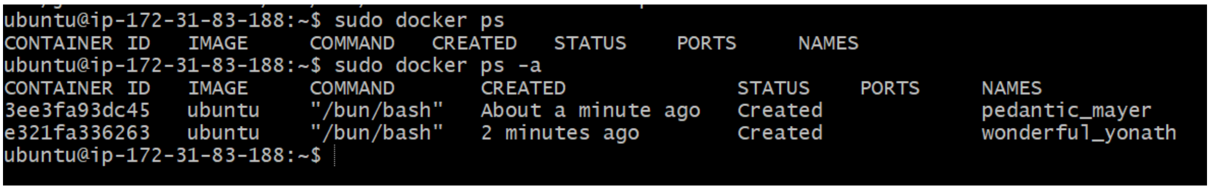
```
ubuntu@ip-172-31-83-188:~$ docker --version
Docker version 20.10.25, build 20.10.25-0ubuntu1~20.04.1
```



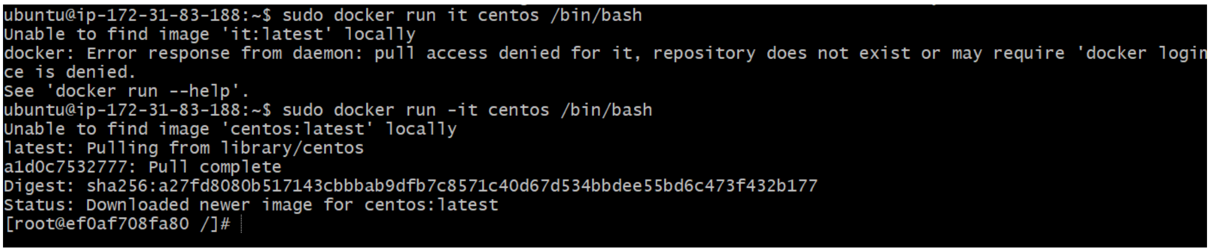
```
ubuntu@ip-172-31-83-188:~$ whoami
ubuntu
ubuntu@ip-172-31-83-188:~$ touch test1
ubuntu@ip-172-31-83-188:~$ sudo docker version
sudo: docker: command not found
ubuntu@ip-172-31-83-188:~$ docker --version

Command 'docker' not found, but can be installed with:

sudo snap install docker     # version 20.10.24, or
sudo apt  install docker.io  # version 20.10.25-0ubuntu1~20.04.1

See 'snap info docker' for additional versions.

ubuntu@ip-172-31-83-188:~$
ubuntu@ip-172-31-83-188:~$ sudo apt  install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 99 not upgraded.
Need to get 67.1 MB of archives.
After this operation, 294 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [30.5 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.1.7-0ubuntu1~20.04.1 [3819 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd amd64 1.7.2-0ubuntu1~20.04.1 [32.5 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-data all 2019052802 [5300 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 amd64 1.33-2.2ubuntu2 [46.2 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 dnsmasq-base amd64 2.80-1.1ubuntu1.7 [315 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 docker.io amd64 20.10.25-0ubuntu1~20.04.1 [30.3 MB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 ubuntu-fan all 0.12.13ubuntu0.1 [34.4 kB]
Fetched 67.1 MB in 2s (30.3 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 62457 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1)
```

Docker services / start/stop/run images

- **sudo docker run -it ubuntu /bun/bash – Docker run**
- **Sudo docker ps -a to know all services**
- **Sudo service Docker status – to know status of docker**
- **Sudo service Docker start – to start services**
- **Sudo service Docker stop – to stop the docker services**
- **sudo docker run -it centos /bin/bash**



```
ubuntu@ip-172-31-83-188:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ubuntu@ip-172-31-83-188:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED              STATUS    PORTS     NAMES
3ee3fa93dc45   ubuntu    "/bun/bash"   About a minute ago   Created             pedantic_mayer
e321fa336263   ubuntu    "/bun/bash"   2 minutes ago        Created             wonderful_yonath
ubuntu@ip-172-31-83-188:~$
```

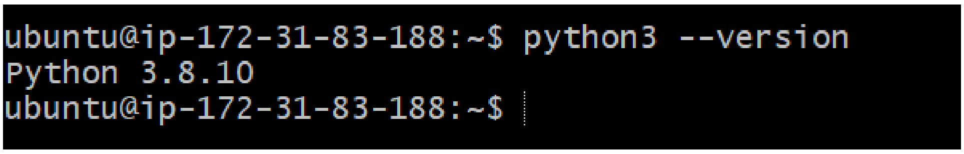


```
ubuntu@ip-172-31-83-188:~$ sudo docker run it centos /bin/bash
Unable to find image 'it:latest' locally
docker: Error response from daemon: pull access denied for it, repository does not exist or may require 'docker login
ce is denied.
See 'docker run --help'.
ubuntu@ip-172-31-83-188:~$ sudo docker run -it centos /bin/bash
Unable to find image 'centos:latest' locally
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
[root@ef0af708fa80 /]#
```

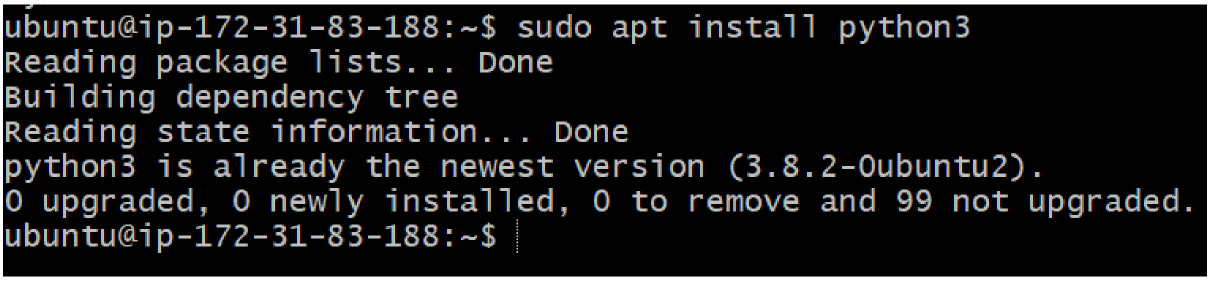
How to install Python on AWS instance

Sudo apt update- just to install latest update

sudo apt install python3- to install python on AWS instance
python3 –version – to check pyhthon version (also let know if already installed or not)

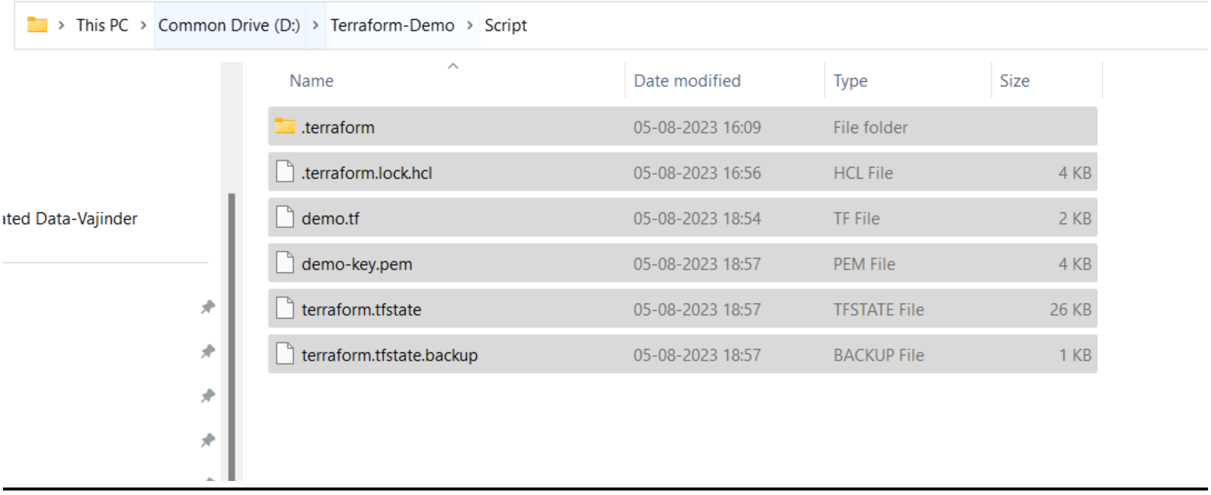


```
ubuntu@ip-172-31-83-188:~$ python3 --version
Python 3.8.10
ubuntu@ip-172-31-83-188:~$
```



```
ubuntu@ip-172-31-83-188:~$ sudo apt install python3
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.8.2-0ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 99 not upgraded.
ubuntu@ip-172-31-83-188:~$
```

Reference files:



Name	Date modified	Type	Size
.terraform	05-08-2023 16:09	File folder	
.terraform.lock.hcl	05-08-2023 16:56	HCL File	4 KB
demo.tf	05-08-2023 18:54	TF File	2 KB
demo-key.pem	05-08-2023 18:57	PEM File	4 KB
terraform.tfstate	05-08-2023 18:57	TFSTATE File	26 KB
terraform.tfstate.backup	05-08-2023 18:57	BACKUP File	1 KB