Seamless DevOps Orchestration: From Terraform Automation to Kubernetes Deployment.

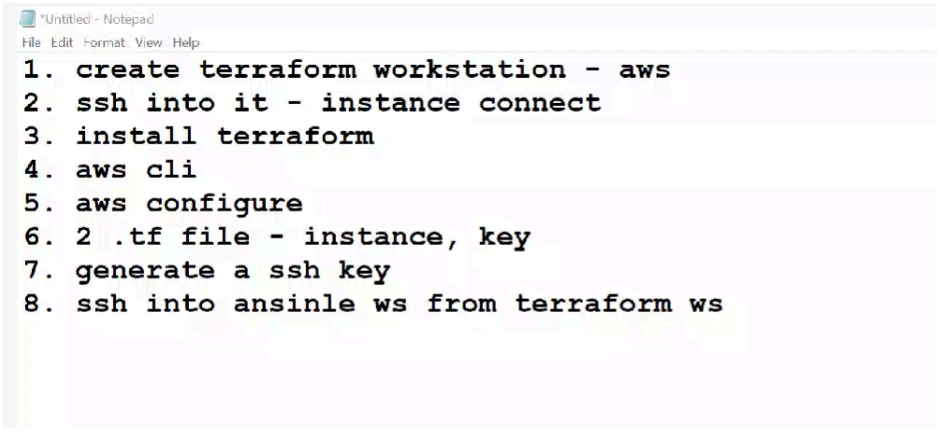
Problem statement:

Launch an Ubuntu EC2 instance (t2.micro) to be used as your Terraform workstation.

From that WS, using Terraform, launch an EC2 instance (instance type: t2.micro, OS: Red Hat Linux) to be used as an ansible workstation for the ansible task.

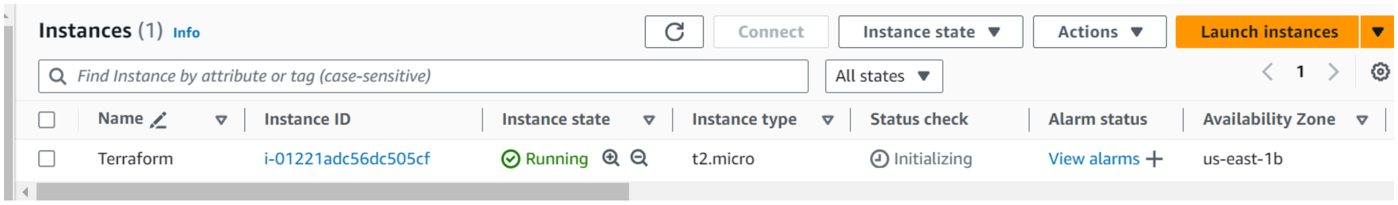
Ensure that you create a key (using ssh-keygen) and use it while launching the EC2 so that we can SSH into the ansible WS once created.

Prerequisites:
EC2
Terraform



```
1. create terraform workstation - aws
2. ssh into it - instance connect
3. install terraform
4. aws cli
5. aws configure
6. 2 .tf file - instance, key
7. generate a ssh key
8. ssh into ansinle ws from terraform ws
```

Launch an Ubuntu EC2 instance (t2.micro) to be used as your Terraform workstation



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Terraform	i-01221adc56dc505cf	Running	t2.micro	Initializing	View alarms	us-east-1b

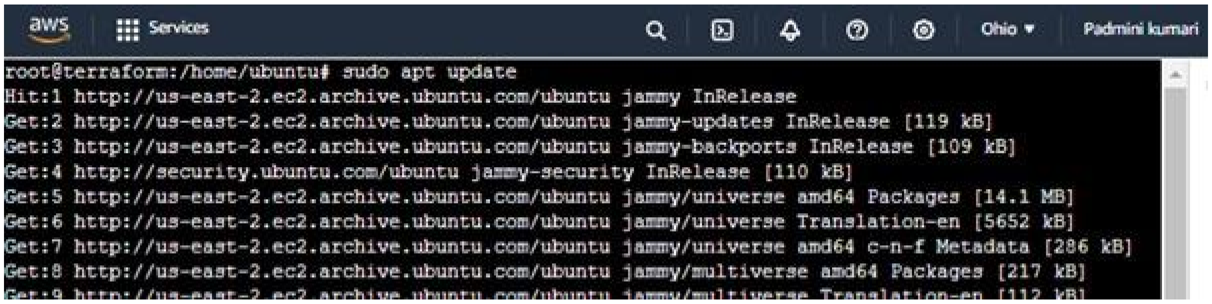
Performed SSH via Ec2 instance connect

To Set hostname,

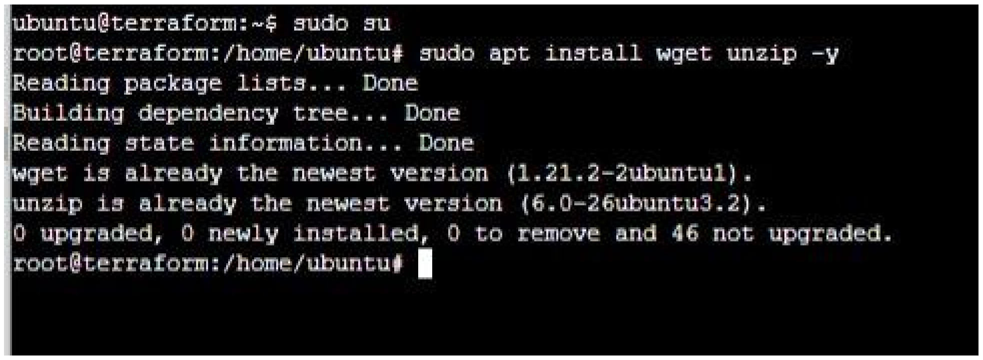
sudo hostnamectl set-hostname **terraform**

Bash

Run the command “**sudo apt update**” and **sudo apt install wget unzip –y**” to install the Ubuntu packages.

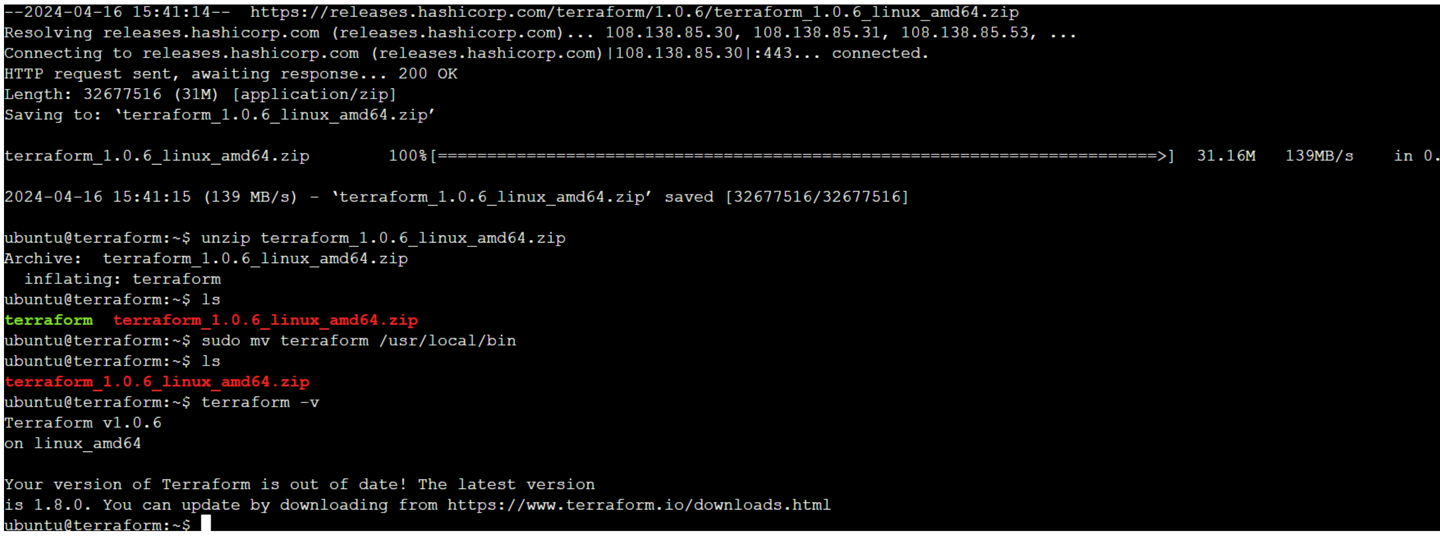


```
root@terraform:/home/ubuntu# sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
```



```
ubuntu@terraform:~$ sudo su
root@terraform:/home/ubuntu# sudo apt install wget unzip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.2-2ubuntu1).
unzip is already the newest version (6.0-26ubuntu3.2).
0 upgraded, 0 newly installed, 0 to remove and 46 not upgraded.
root@terraform:/home/ubuntu#
```

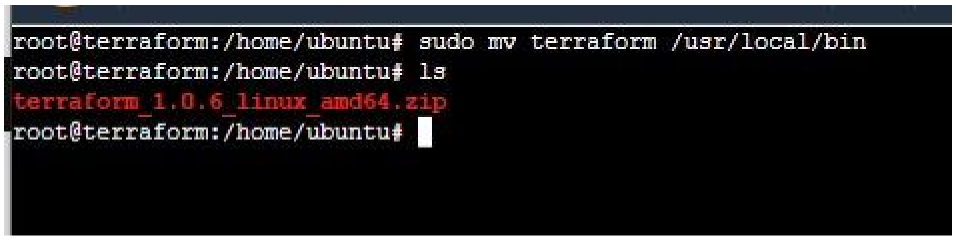
Terraform is installed using the command
“**wget https://releases.hashicorp.com/terraform/1.0.6/terraform_1.0.6_linux_amd64.zip**
“unzip terraform_1.0.6_linux_amd64.zip” and check if it is installed or not using the command “**ls**”



```
--2024-04-16 15:41:14--  https://releases.hashicorp.com/terraform/1.0.6/terraform_1.0.6_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 108.138.85.30, 108.138.85.31, 108.138.85.53, ...
Connecting to releases.hashicorp.com (releases.hashicorp.com)|108.138.85.30|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 32677516 (31M) [application/zip]
Saving to: ‘terraform_1.0.6_linux_amd64.zip’

terraform_1.0.6_linux_amd64.zip        100%[=========================================================================>]  31.16M   139MB/s    in 0.

2024-04-16 15:41:15 (139 MB/s) - ‘terraform_1.0.6_linux_amd64.zip’ saved [32677516/32677516]

ubuntu@terraform:~$ unzip terraform_1.0.6_linux_amd64.zip
Archive:  terraform_1.0.6_linux_amd64.zip
  inflating: terraform
ubuntu@terraform:~$ ls
terraform  terraform_1.0.6_linux_amd64.zip
ubuntu@terraform:~$ sudo mv terraform /usr/local/bin
ubuntu@terraform:~$ ls
terraform_1.0.6_linux_amd64.zip
ubuntu@terraform:~$ terraform -v
Terraform v1.0.6
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.8.0. You can update by downloading from https://www.terraform.io/downloads.html
ubuntu@terraform:~$
```

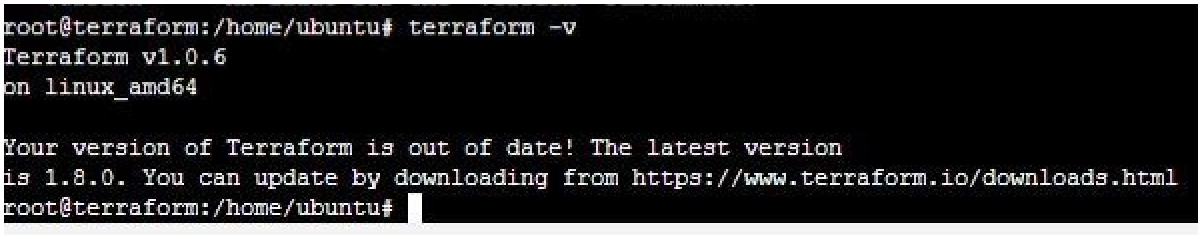
Moved Terraform file to /usr/local/bin location.

Sudo mv terraform /usr/local/bin



```
root@terraform:/home/ubuntu# sudo mv terraform /usr/local/bin
root@terraform:/home/ubuntu# ls
terraform_1.0.6_linux_amd64.zip
root@terraform:/home/ubuntu# 
```

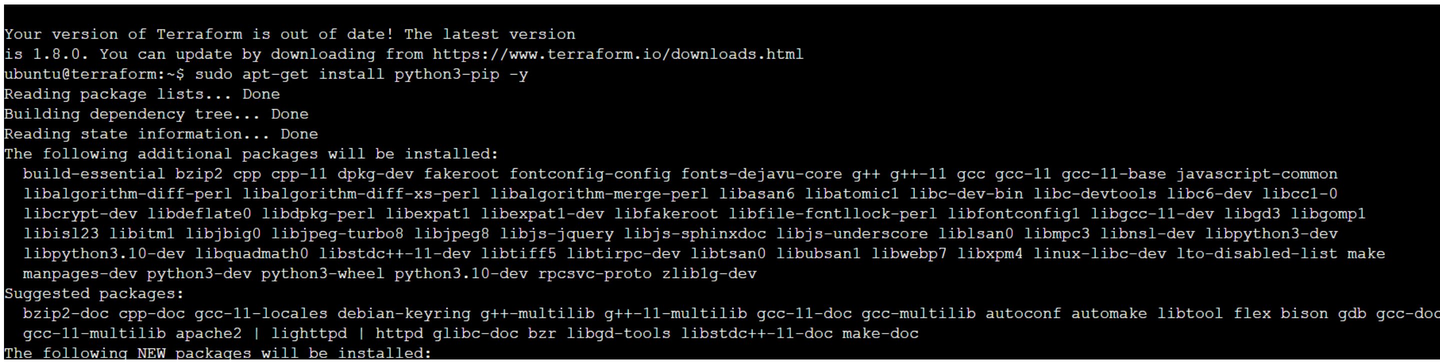
Checked the version of Terraform using the command “terraform and then terraform –v



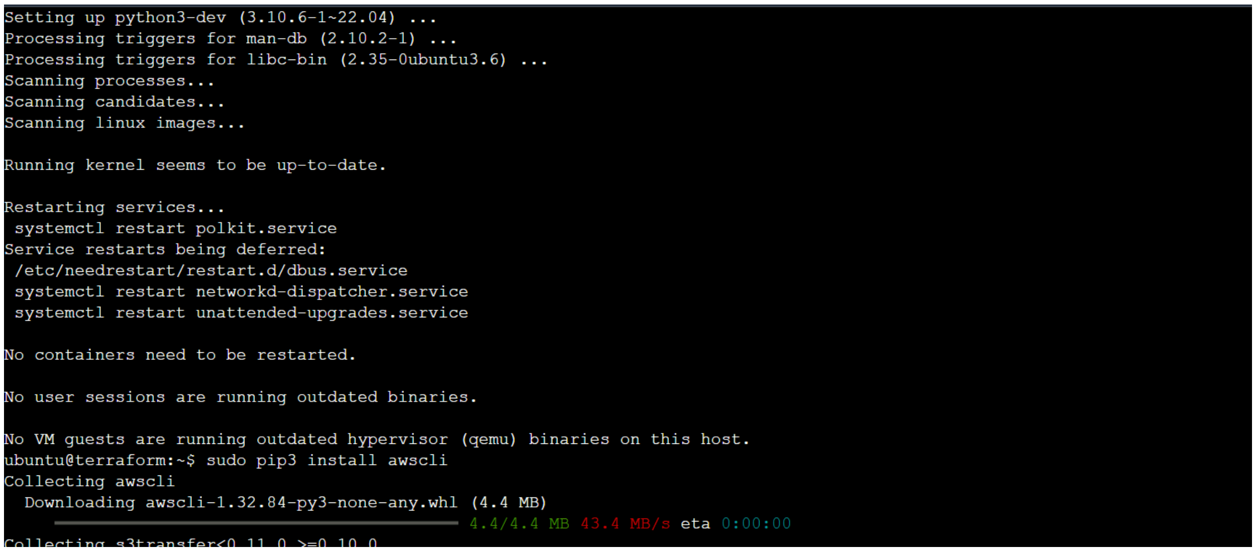
```
root@terraform:/home/ubuntu# terraform -v
Terraform v1.0.6
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.8.0. You can update by downloading from https://www.terraform.io/downloads.html
root@terraform:/home/ubuntu# 
```

Installed Python

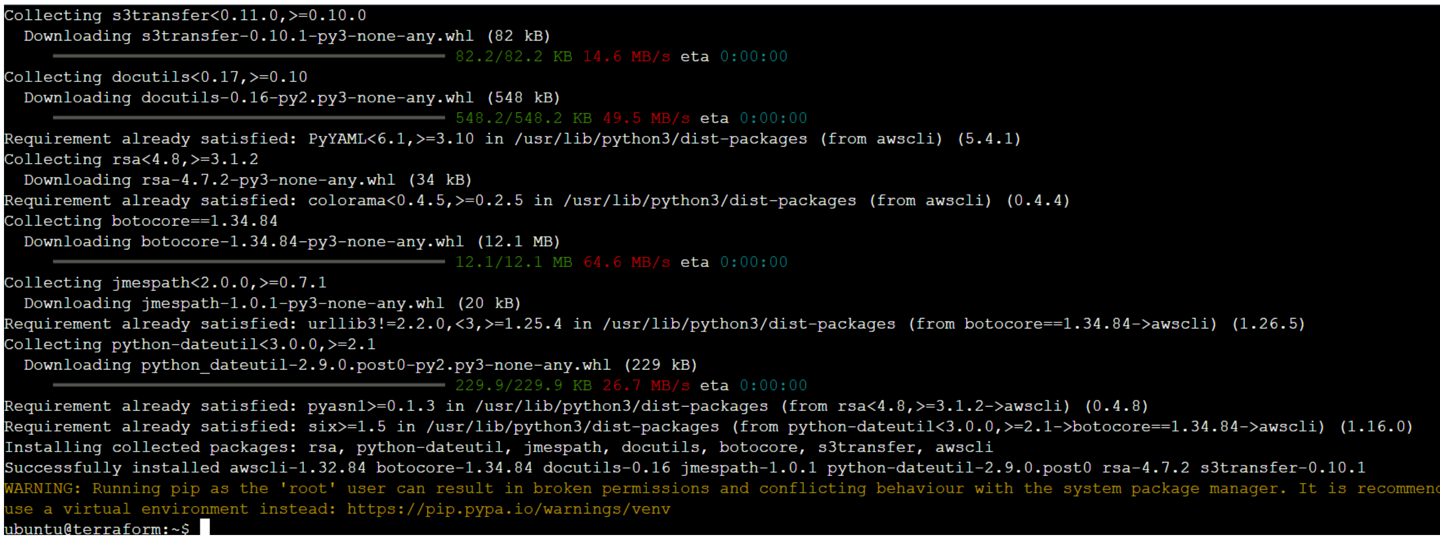
$ sudo apt-get install python3-pip -y
$ sudo pip3 install awscli



```
Your version of Terraform is out of date! The latest version
is 1.8.0. You can update by downloading from https://www.terraform.io/downloads.html
ubuntu@terraform:~$ sudo apt-get install python3-pip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11 gcc-11-base javascript-common
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1 libc-dev-bin libc-devtools libc6-dev libcc1-0
  libcrypt-dev libdeflate0 libdpkg-perl libexpat1 libexpat1-dev libfakeroot libfile-fcntllock-perl libfontconfig1 libgcc-11-dev libgd3 libgomp1
  libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libmpc3 libnsl-dev libpython3-dev
  libpython3.10-dev libquadmath0 libstdc++-11-dev libtiff5 libtirpc-dev libtsan0 libubsan1 libwebp7 libxpm4 linux-libc-dev lto-disabled-list make
  manpages-dev python3-dev python3-wheel python3.10-dev rpcsvc-proto zlib1g-dev
Suggested packages:
  bzip2-doc cpp-doc gcc-11-locales debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib autoconf automake libtool flex bison gdb gcc-doc
  gcc-11-multilib apache2 | lighttpd | httpd glibc-doc bzr libgd-tools libstdc++-11-doc make-doc
The following NEW packages will be installed:
```



```
Setting up python3-dev (3.10.6-1~22.04) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...
 systemctl restart polkit.service
Service restarts being deferred:
 /etc/needrestart/restart.d/dbus.service
 systemctl restart networkd-dispatcher.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@terraform:~$ sudo pip3 install awscli
Collecting awscli
  Downloading awscli-1.32.84-py3-none-any.whl (4.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.4/4.4 MB 43.4 MB/s eta 0:00:00
Collecting s3transfer<0.11.0,>=0.10.0
```



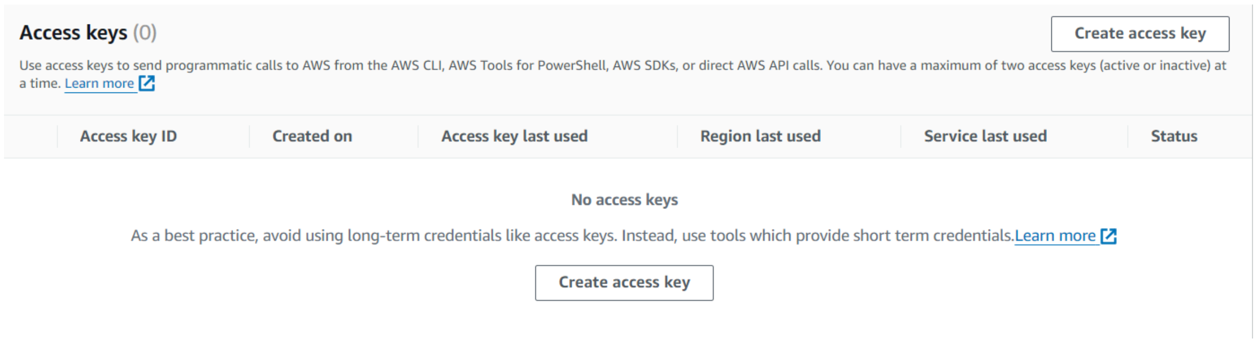
```
Collecting s3transfer<0.11.0,>=0.10.0
  Downloading s3transfer-0.10.1-py3-none-any.whl (82 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 82.2/82.2 KB 14.6 MB/s eta 0:00:00
Collecting docutils<0.17,>=0.10
  Downloading docutils-0.16-py2.py3-none-any.whl (548 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 548.2/548.2 KB 49.5 MB/s eta 0:00:00
Requirement already satisfied: PyYAML<6.1,>=3.10 in /usr/lib/python3/dist-packages (from awscli) (5.4.1)
Collecting rsa<4.8,>=3.1.2
  Downloading rsa-4.7.2-py3-none-any.whl (34 kB)
Requirement already satisfied: colorama<0.4.5,>=0.2.5 in /usr/lib/python3/dist-packages (from awscli) (0.4.4)
Collecting botocore==1.34.84
  Downloading botocore-1.34.84-py3-none-any.whl (12.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.1/12.1 MB 64.6 MB/s eta 0:00:00
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in /usr/lib/python3/dist-packages (from botocore==1.34.84->awscli) (1.26.5)
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 229.9/229.9 KB 26.7 MB/s eta 0:00:00
Requirement already satisfied: pyasn1>=0.1.3 in /usr/lib/python3/dist-packages (from rsa<4.8,>=3.1.2->awscli) (0.4.8)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil<3.0.0,>=2.1->botocore==1.34.84->awscli) (1.16.0)
Installing collected packages: rsa, python-dateutil, jmespath, docutils, botocore, s3transfer, awscli
Successfully installed awscli-1.32.84 botocore-1.34.84 docutils-0.16 jmespath-1.0.1 python-dateutil-2.9.0.post0 rsa-4.7.2 s3transfer-0.10.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommend
use a virtual environment instead: https://pip.pypa.io/warnings/venv
ubuntu@terraform:~$ 
```

AWS configure (This setting allows CLI to authenticate with AWS services and make API requests on your behalf, & these keys are crucial for securely accessing and interacting with various AWS services, such as EC2 .)

<u>Access Key</u>

ID: AKIAYZLMO7NR52UGQAMP
Key: 1QphXmsZwIqLdyHowFLJqMB4fqLmC8Y93fQTRa/z

To create the Access key, we should go to the profile icon in the AWS console click “security credentials” click Create access key, and follow the rest of the procedure to create it.


Access keys (0)

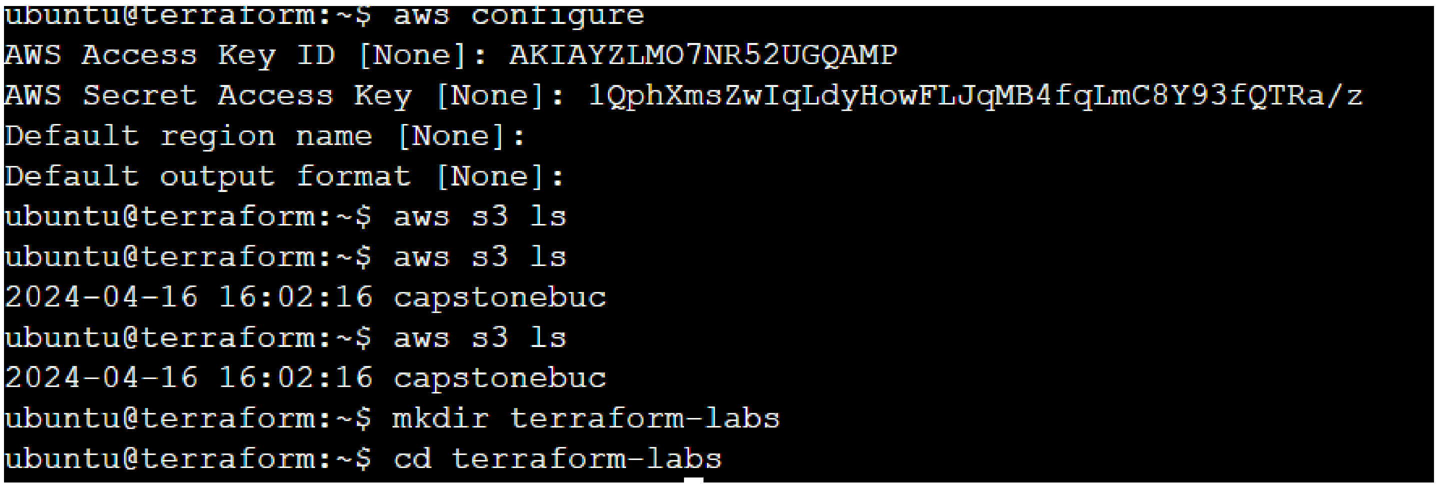
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

Access key ID	Created on	Access key last used	Region last used	Service last used	Status

No access keys

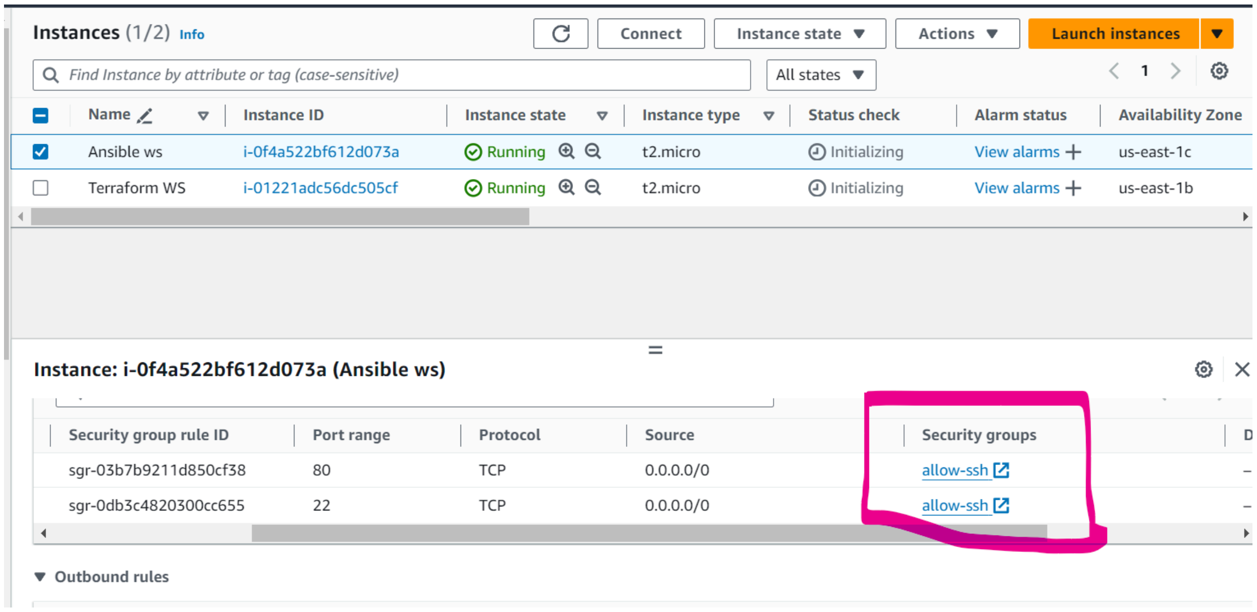
As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more

To list the **S3 buckets**, use the command "**aws s3 ls**" and create a new directory using the command "**mkdir** terraform-labs" Change the directory to the new one using the command "cd terraform-labs".


```
ubuntu@terraform:~$ aws configure
AWS Access Key ID [None]: AKIAYZLMO7NR52UGQAMP
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
ubuntu@terraform:~$ aws s3 ls
ubuntu@terraform:~$ aws s3 ls
2024-04-16 16:02:16 capstonebuc
ubuntu@terraform:~$ aws s3 ls
2024-04-16 16:02:16 capstonebuc
ubuntu@terraform:~$ mkdir terraform-labs
ubuntu@terraform:~$ cd terraform-labs
```

Create a (instancefile1.tf) file to run the below script to create a new instance of **Ansible** in the same region. Also, to get the AMI value, navigate to EC2, select the launch instance option, select the AMI of RedHat, and copy the AMI ID.

```
provider "aws" {
profile = "default"
region = "us-east-1"
}
resource "aws_instance" "example" {
ami= "ami-0fe630eb857a6ec83"
instance_type = "t2.micro"
vpc_security_group_ids = [aws_security_group.allow-ssh.id]
 key_name            = aws_key_pair.mykey.key_name
}

resource "aws_security_group" "allow-ssh" {
  vpc_id     = "vpc-06c7cfcba064509b7"
  name       = "allow-ssh"
 description = "security group that allows SSH and all egress traffic"

egress {
   from_port   = 0
   to_port     = 0
   protocol    = "-1"
   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   = 80
   to_port     = 80
   protocol    = "tcp"
   cidr_blocks = ["0.0.0.0/0"]
 }
}
```



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Ansible ws	i-0f4a522bf612d073a	Running	t2.micro	Initializing	View alarms	us-east-1c
Terraform WS	i-01221adc56dc505cf	Running	t2.micro	Initializing	View alarms	us-east-1b

Instance: i-0f4a522bf612d073a (Ansible ws)

Security group rule ID	Port range	Protocol	Source	Security groups
sgr-03b7b9211d850cf38	80	TCP	0.0.0.0/0	allow-ssh
sgr-0db3c4820300cc655	22	TCP	0.0.0.0/0	allow-ssh

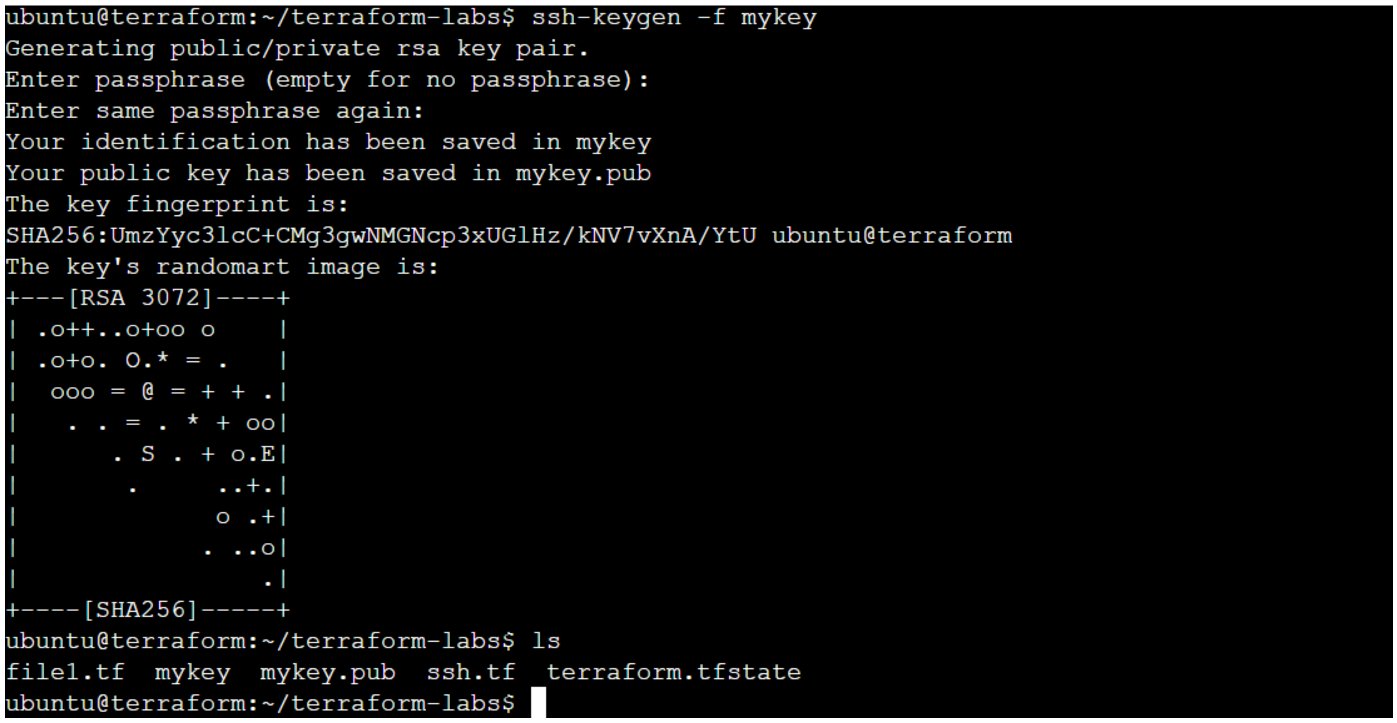
Generated the SSH-key using the command “ssh-keygen –f mykey”.

ls command to list the files

when you run "ssh-keygen -f mykey" in your terminal or command prompt, it will generate a new SSH key pair and save it with the filenames "mykey" (private key) and "mykey.pub" (public key) in the current directory.

ssh-keygen: This is the command-line tool used to generate SSH keys.

-f mykey: This part of the command specifies the filename for the generated key pair. In this case, "mykey" is the filename. If you run this command as is, it will generate a new SSH key pair with the filenames "mykey" for the private key and "mykey.pub" for the public key. The private key is typically kept secure and not shared, while the public key can be distributed to allow access to systems that you want to connect to securely via SSH.



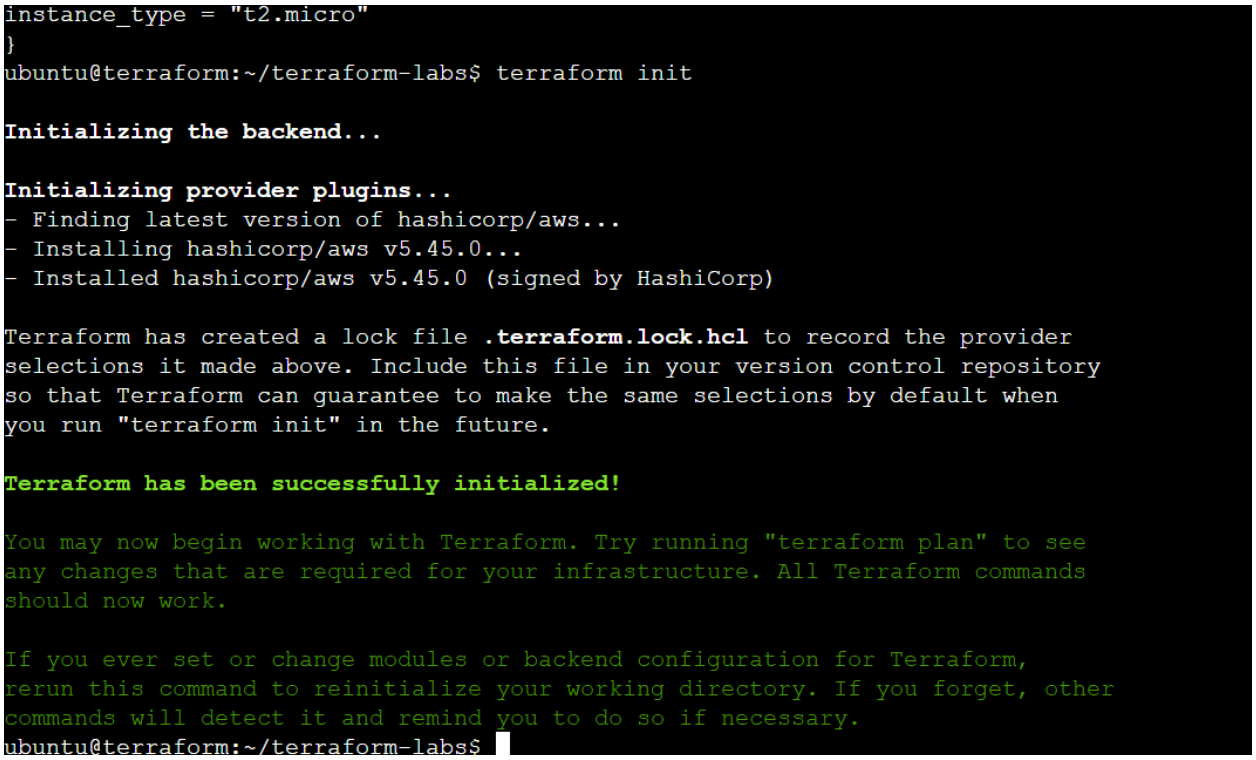
```
ubuntu@terraform:~/terraform-labs$ ssh-keygen -f mykey
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in mykey
Your public key has been saved in mykey.pub
The key fingerprint is:
SHA256:UmzYyc3lcC+CMg3gwNMGNcp3xUGlHz/kNV7vXnA/YtU ubuntu@terraform
The key's randomart image is:
+---[RSA 3072]----+
| .o++..o+oo o    |
| .o+o. O.* = .   |
|  ooo = @ = + + .|
|   . . = . * + oo|
|      . S . + o.E|
|       .     ..+.|
|             o .+|
|            . ..o|
|                .|
+----[SHA256]-----+
ubuntu@terraform:~/terraform-labs$ ls
file1.tf  mykey  mykey.pub  ssh.tf  terraform.tfstate
ubuntu@terraform:~/terraform-labs$ 
```

Creating a .tf for SSH key -

vi sshkey.tf

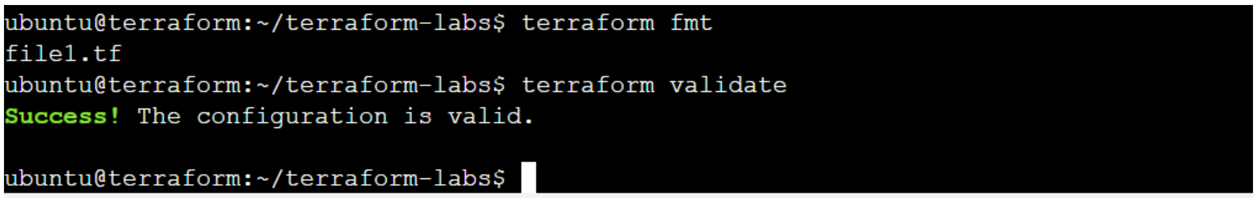
```
resource "aws_key_pair" "mykey" {
key_name = "mykey"
public_key = mykey.pub
}
```

Terraform init:
This command **initializes a working directory containing Terraform configuration files**.
It downloads and installs any necessary plugins defined in the configuration.



```
instance_type = "t2.micro"
}
ubuntu@terraform:~/terraform-labs$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.45.0...
- Installed hashicorp/aws v5.45.0 (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,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@terraform:~/terraform-labs$
```

terraform fmt:
This command is used to format Terraform configuration files. It automatically updates the formatting to follow Terraform's conventions, making the code easier to read and maintain.

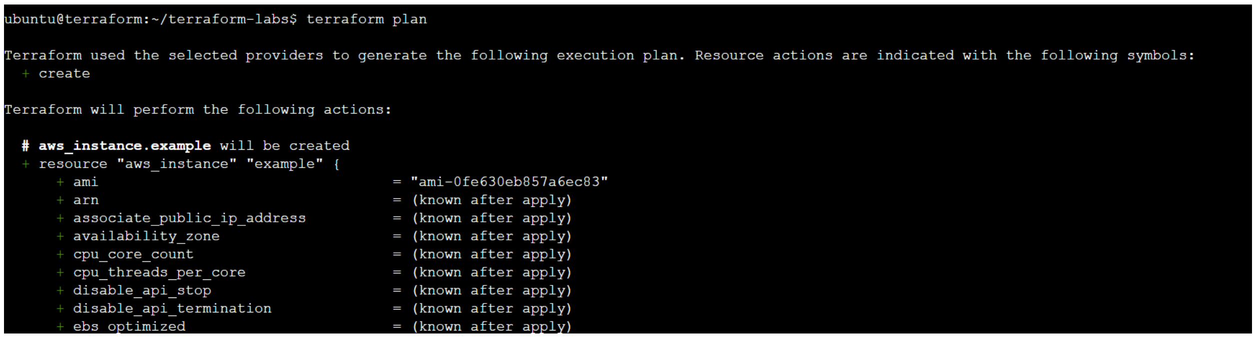


```
ubuntu@terraform:~/terraform-labs$ terraform fmt
file1.tf
ubuntu@terraform:~/terraform-labs$ terraform validate
Success! The configuration is valid.

ubuntu@terraform:~/terraform-labs$
```

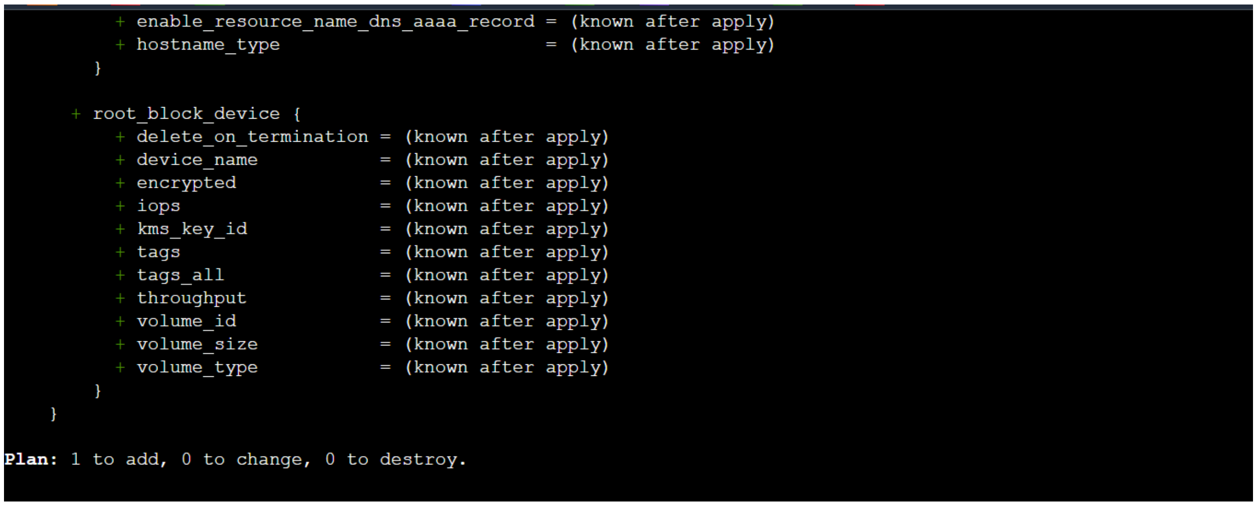
terraform validate:
This command is used to check the syntax and validity of Terraform configuration files. It verifies whether the configuration files are correctly written and if all necessary variables are defined.

terraform plan: (dry run)
This command creates an execution plan. It generates an execution plan based on the Terraform

configuration and shows what actions Terraform will take when you apply the configuration. This allows you to preview changes before actually applying them.

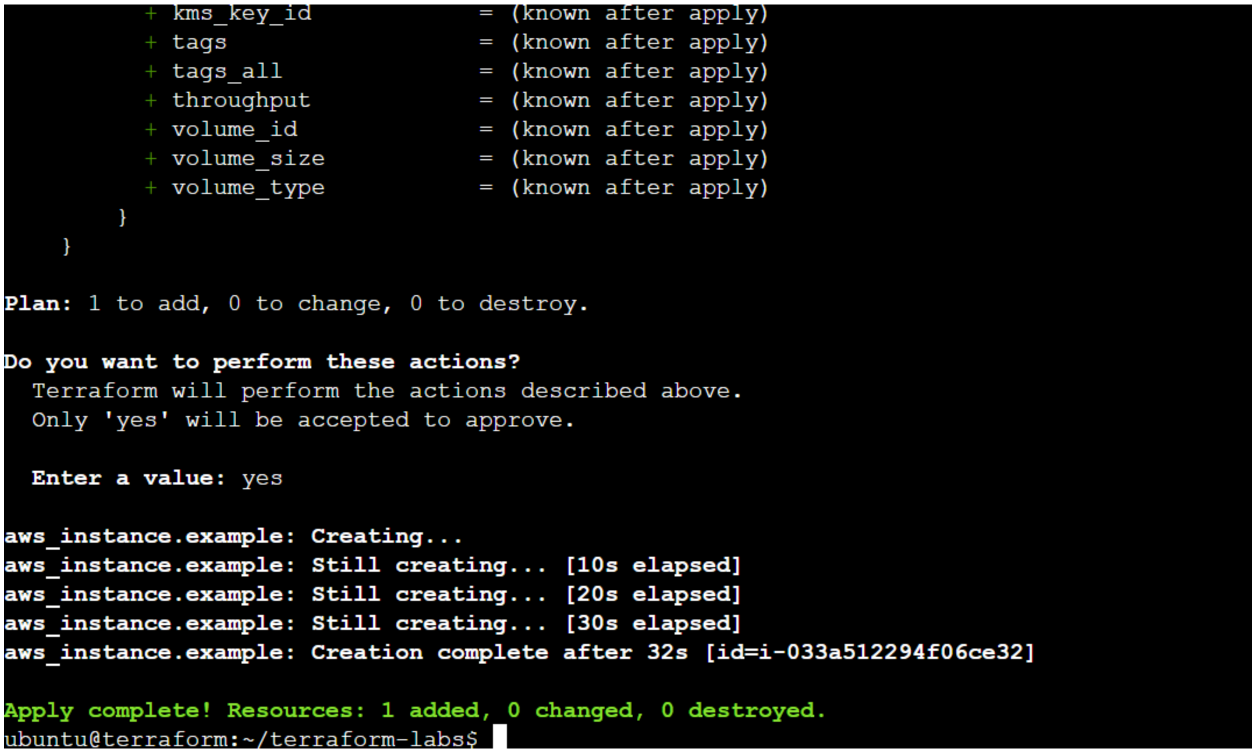


```
ubuntu@terraform:~/terraform-labs$ terraform plan

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.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-0fe630eb857a6ec83"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + 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)
```

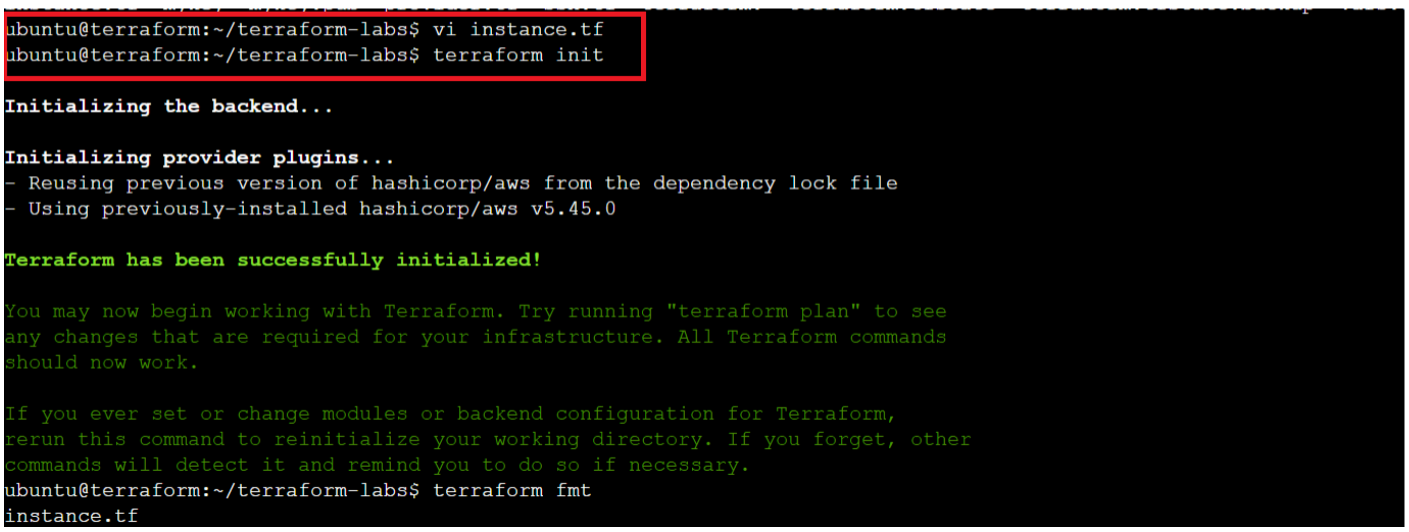


```
          + enable_resource_name_dns_aaaa_record = (known after apply)
          + hostname_type                        = (known after apply)
        }

      + root_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + tags_all              = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```

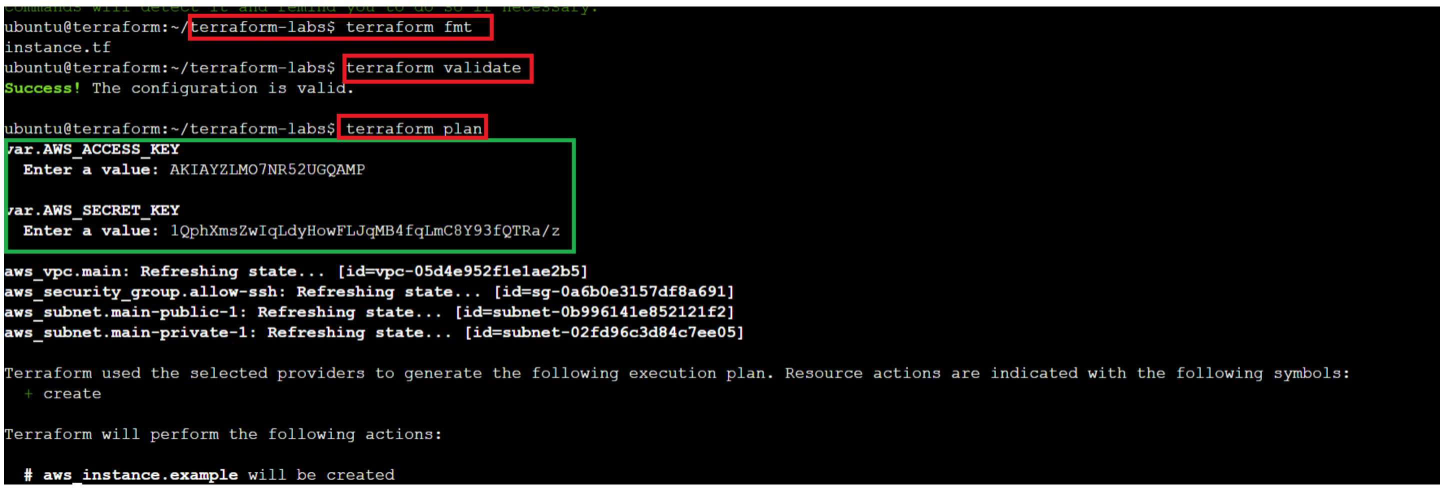
terraform apply: This command applies the changes described in a terraform execution plan. It creates, updates, or deletes infrastructure resources as defined in the Terraform configuration.



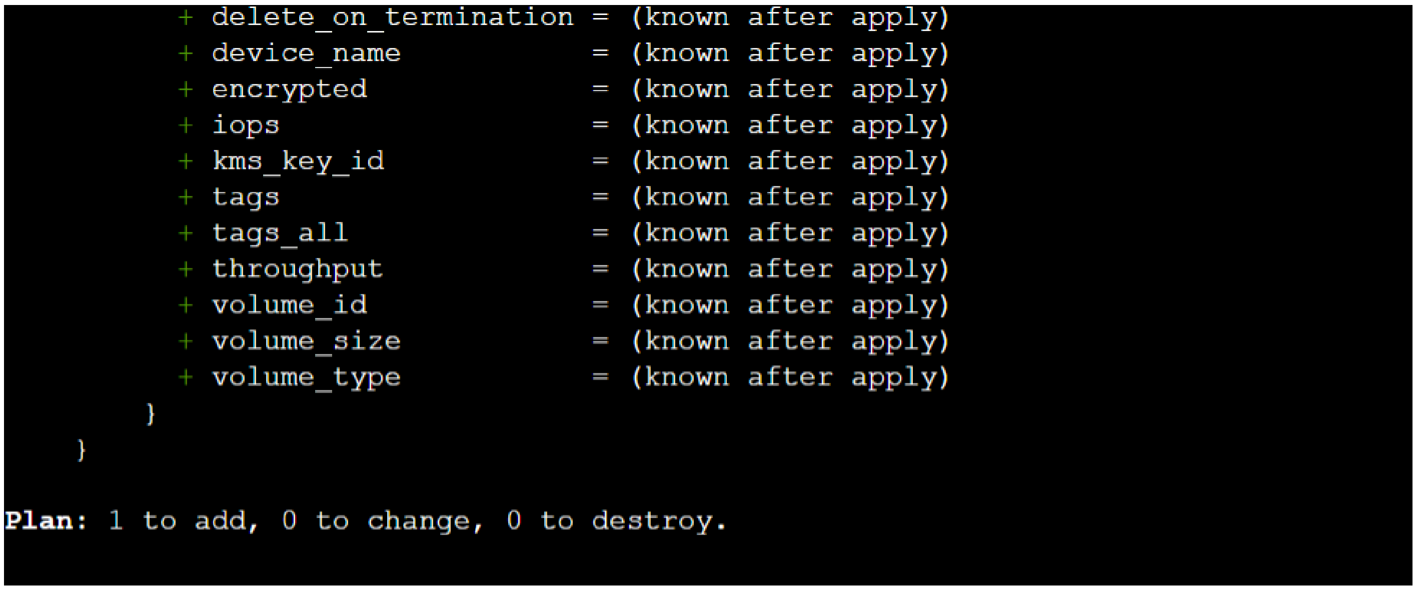
```
          + kms_key_id             = (known after apply)
          + tags                   = (known after apply)
          + tags_all               = (known after apply)
          + throughput             = (known after apply)
          + volume_id              = (known after apply)
          + volume_size            = (known after apply)
          + volume_type            = (known after apply)
        }
    }

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

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.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Still creating... [30s elapsed]
aws_instance.example: Creation complete after 32s [id=i-033a512294f06ce32]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
ubuntu@terraform:~/terraform-labs$
```



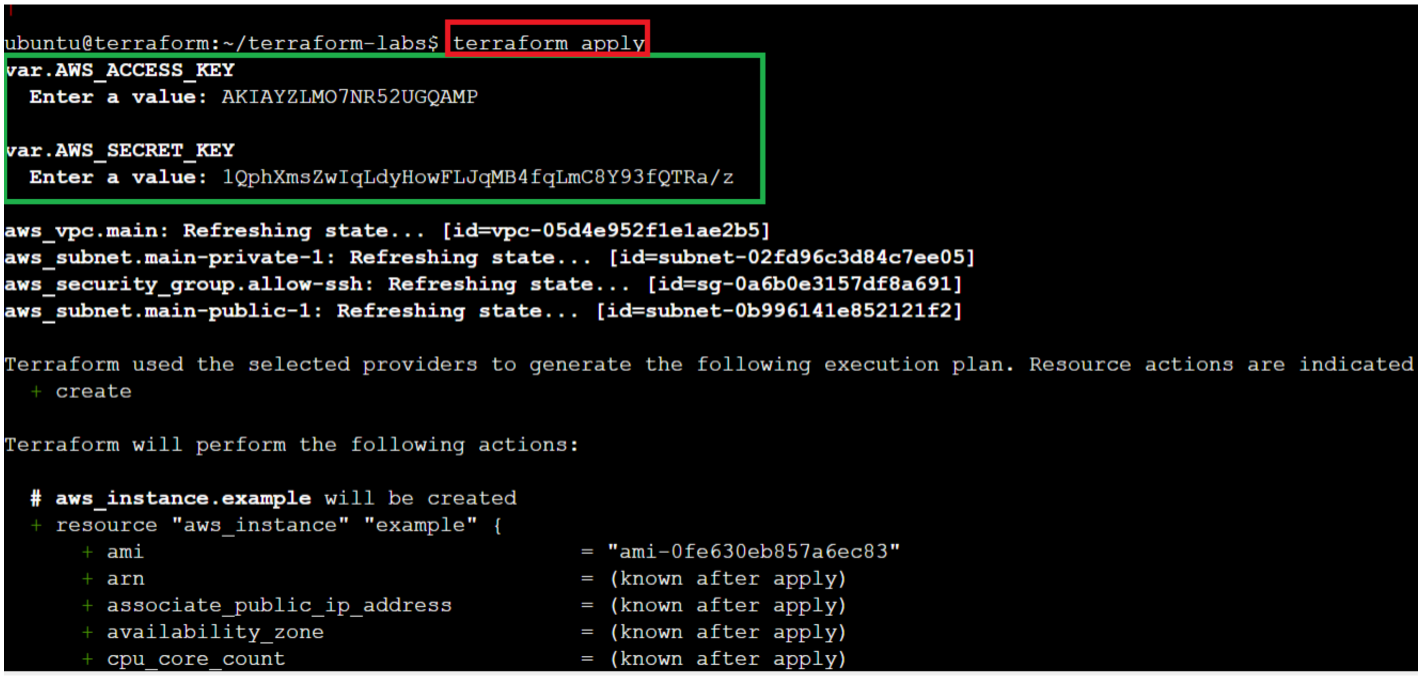
```
ubuntu@terraform:~/terraform-labs$ vi instance.tf
ubuntu@terraform:~/terraform-labs$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.45.0

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,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@terraform:~/terraform-labs$ terraform fmt
instance.tf
```



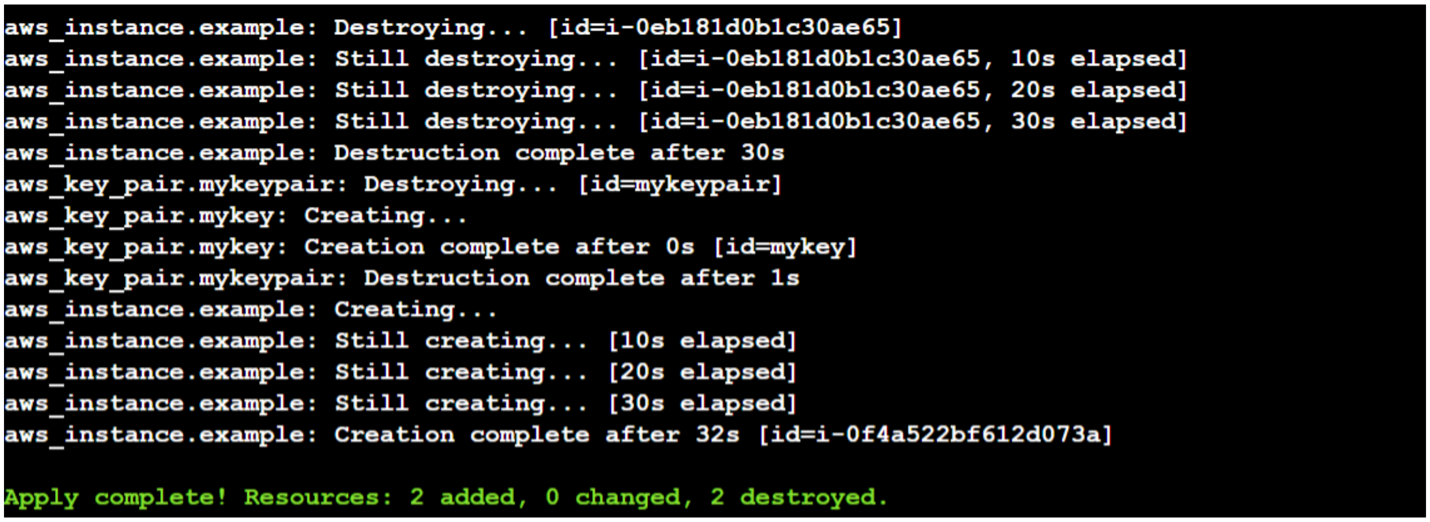
```
ubuntu@terraform:~/terraform-labs$ terraform fmt
instance.tf
ubuntu@terraform:~/terraform-labs$ terraform validate
Success! The configuration is valid.

ubuntu@terraform:~/terraform-labs$ terraform plan
var.AWS_ACCESS_KEY
  Enter a value: AKIAYZLMO7NR52UGQAMP

var.AWS_SECRET_KEY
  Enter a value: 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

aws_vpc.main: Refreshing state... [id=vpc-05d4e952f1e1ae2b5]
aws_security_group.allow-ssh: Refreshing state... [id=sg-0a6b0e3157df8a691]
aws_subnet.main-public-1: Refreshing state... [id=subnet-0b996141e852121f2]
aws_subnet.main-private-1: Refreshing state... [id=subnet-02fd96c3d84c7ee05]

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.example will be created
```



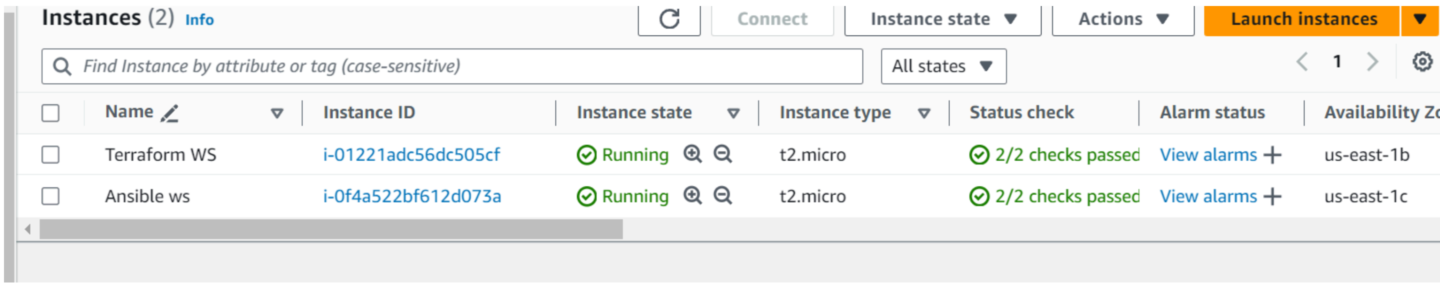
```
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + tags_all              = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```



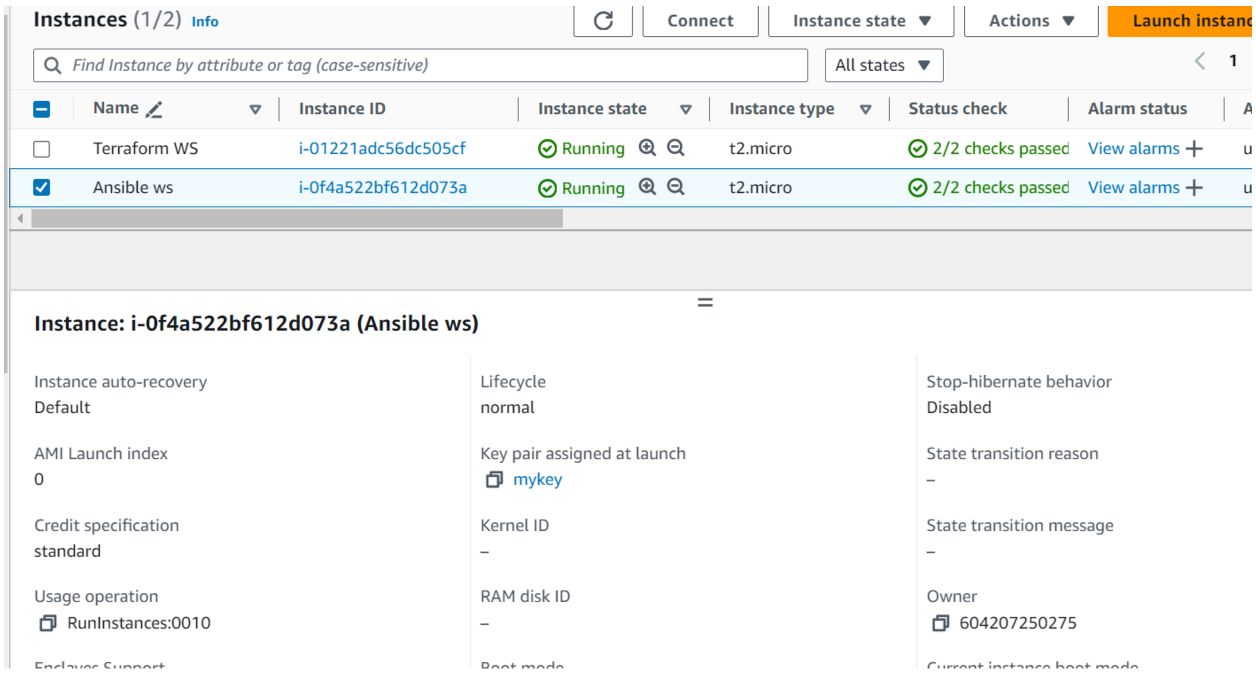
```
ubuntu@terraform:~/terraform-labs$ terraform apply
var.AWS_ACCESS_KEY
  Enter a value: AKIAYZLMO7NR52UGQAMP

var.AWS_SECRET_KEY
  Enter a value: 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

aws_vpc.main: Refreshing state... [id=vpc-05d4e952f1e1ae2b5]
aws_subnet.main-private-1: Refreshing state... [id=subnet-02fd96c3d84c7ee05]
aws_security_group.allow-ssh: Refreshing state... [id=sg-0a6b0e3157df8a691]
aws_subnet.main-public-1: Refreshing state... [id=subnet-0b996141e852121f2]

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

Terraform will perform the following actions:

  # aws_instance.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-0fe630eb857a6ec83"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
```



```
aws_instance.example: Destroying... [id=i-0eb181d0b1c30ae65]
aws_instance.example: Still destroying... [id=i-0eb181d0b1c30ae65, 10s elapsed]
aws_instance.example: Still destroying... [id=i-0eb181d0b1c30ae65, 20s elapsed]
aws_instance.example: Still destroying... [id=i-0eb181d0b1c30ae65, 30s elapsed]
aws_instance.example: Destruction complete after 30s
aws_key_pair.mykeypair: Destroying... [id=mykeypair]
aws_key_pair.mykey: Creating...
aws_key_pair.mykey: Creation complete after 0s [id=mykey]
aws_key_pair.mykeypair: Destruction complete after 1s
aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Still creating... [30s elapsed]
aws_instance.example: Creation complete after 32s [id=i-0f4a522bf612d073a]

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.
```



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z
☐	Terraform WS	i-01221adc56dc505cf	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-1b
☐	Ansible ws	i-0f4a522bf612d073a	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-1c

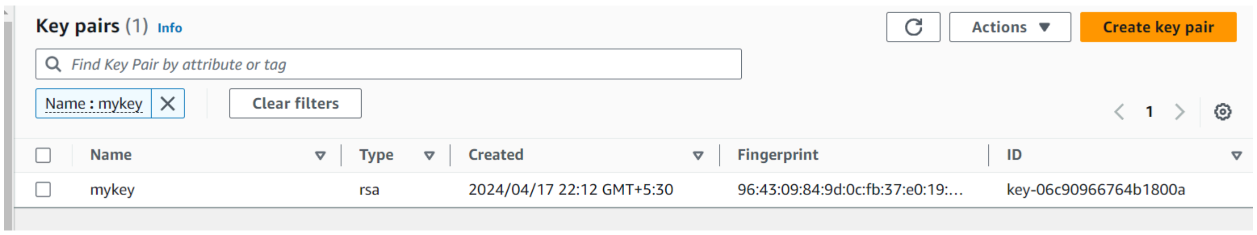


Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	Terraform WS	i-01221adc56dc505cf	Running	t2.micro	2/2 checks passed	View alarms
☑	Ansible ws	i-0f4a522bf612d073a	Running	t2.micro	2/2 checks passed	View alarms

Instance: i-0f4a522bf612d073a (Ansible ws)

Instance auto-recovery: Default	Lifecycle: normal	Stop-hibernate behavior: Disabled
AMI Launch index: 0	Key pair assigned at launch: mykey	State transition reason: –
Credit specification: standard	Kernel ID: –	State transition message: –
Usage operation: RunInstances:0010	RAM disk ID: –	Owner: 604207250275



Key pairs (1) Info

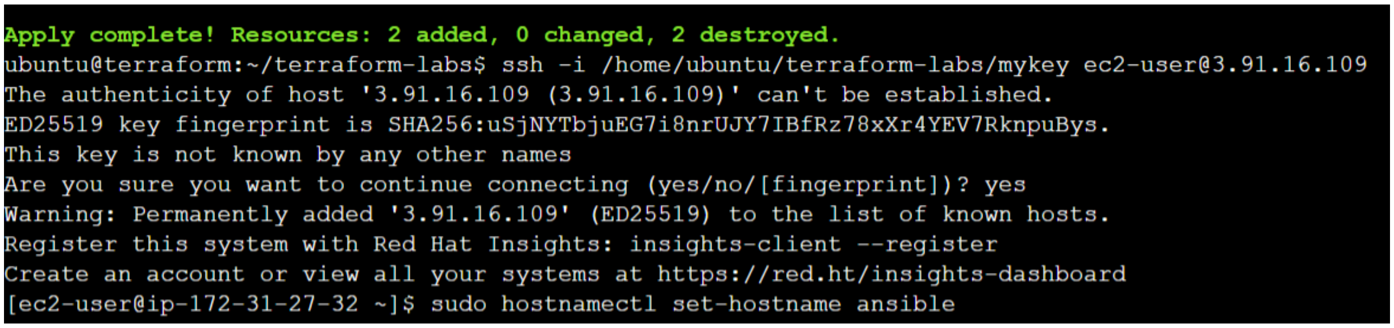
Name : mykey

	Name	Type	Created	Fingerprint	ID
☐	mykey	rsa	2024/04/17 22:12 GMT+5:30	96:43:09:84:9d:0c:fb:37:e0:19:...	key-06c90966764b1800a

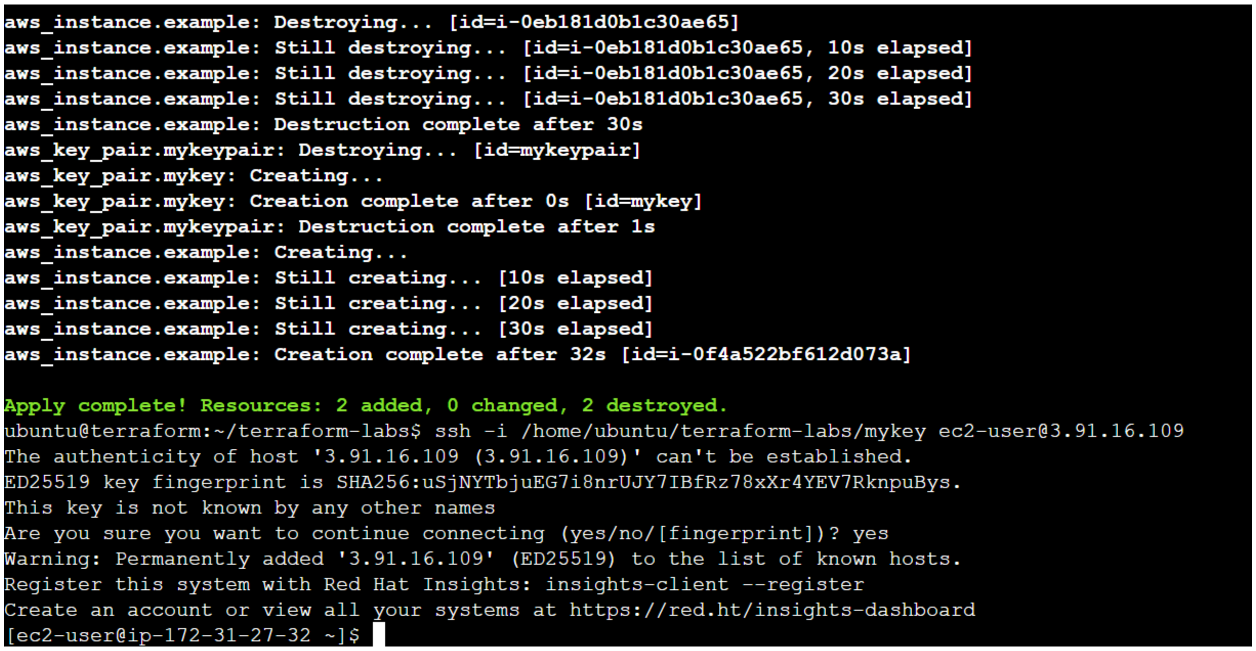
SSH into the Ansible workstation

ssh -i /home/ubuntu/terraform-labs/mykey ec2-user@ip address of target server

ssh -i /home/ubuntu/terraform-labs/mykey ec2-user@3.91.16.109



```
Apply complete! Resources: 2 added, 0 changed, 2 destroyed.
ubuntu@terraform:~/terraform-labs$ ssh -i /home/ubuntu/terraform-labs/mykey ec2-user@3.91.16.109
The authenticity of host '3.91.16.109 (3.91.16.109)' can't be established.
ED25519 key fingerprint is SHA256:uSjNYTbjuEG7i8nrUJY7IBfRz78xXr4YEV7RknpuBys.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.91.16.109' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-27-32 ~]$ sudo hostnamectl set-hostname ansible
```

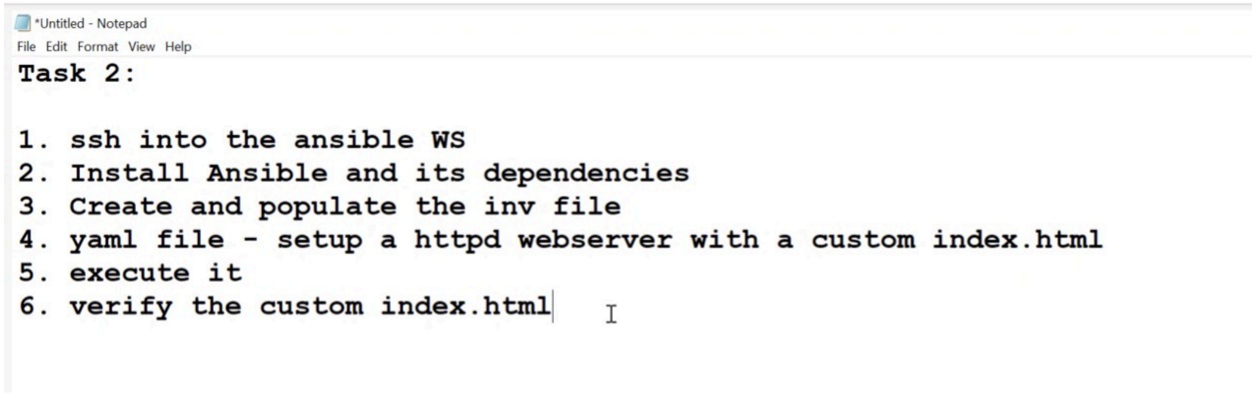


```
aws_instance.example: Destroying... [id=i-0eb181d0b1c30ae65]
aws_instance.example: Still destroying... [id=i-0eb181d0b1c30ae65, 10s elapsed]
aws_instance.example: Still destroying... [id=i-0eb181d0b1c30ae65, 20s elapsed]
aws_instance.example: Still destroying... [id=i-0eb181d0b1c30ae65, 30s elapsed]
aws_instance.example: Destruction complete after 30s
aws_key_pair.mykeypair: Destroying... [id=mykeypair]
aws_key_pair.mykey: Creating...
aws_key_pair.mykey: Creation complete after 0s [id=mykey]
aws_key_pair.mykeypair: Destruction complete after 1s
aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Still creating... [30s elapsed]
aws_instance.example: Creation complete after 32s [id=i-0f4a522bf612d073a]

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.
ubuntu@terraform:~/terraform-labs$ ssh -i /home/ubuntu/terraform-labs/mykey ec2-user@3.91.16.109
The authenticity of host '3.91.16.109 (3.91.16.109)' can't be established.
ED25519 key fingerprint is SHA256:uSjNYTbjuEG7i8nrUJY7IBfRz78xXr4YEV7RknpuBys.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.91.16.109' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-27-32 ~]$
```

Task-2

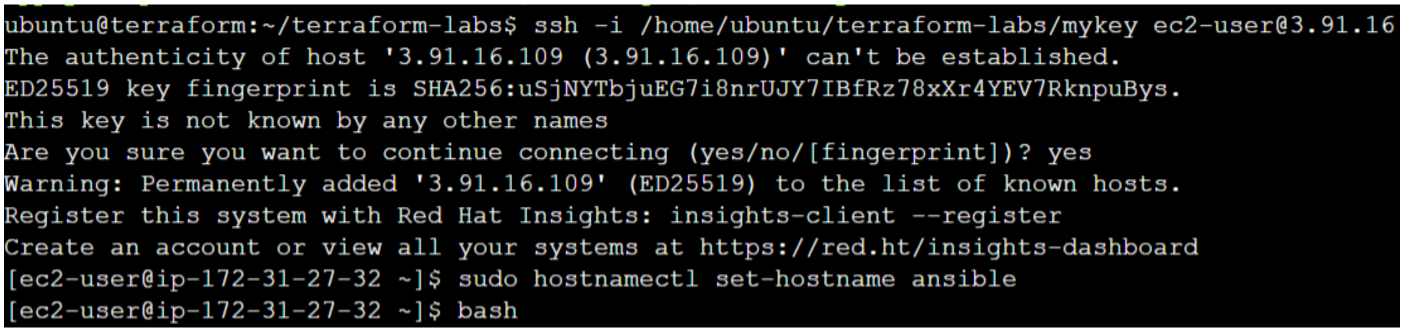
Problem Statement:

Once you have created a new instance using Terraform (as part of Terraform task), ssh into that instance and install Ansible in it.
After that, you have to install the httpd web server in the managed node.
You don't have separate managed nodes. So use your ansible workstation itself as the managed node by adding the below line in your host inventory file:
localhost ansible_connection = local



```
Task 2:

1. ssh into the ansible WS
2. Install Ansible and its dependencies
3. Create and populate the inv file
4. yaml file - setup a httpd webserver with a custom index.html
5. execute it
6. verify the custom index.html
```

1. ssh into the ansible WS

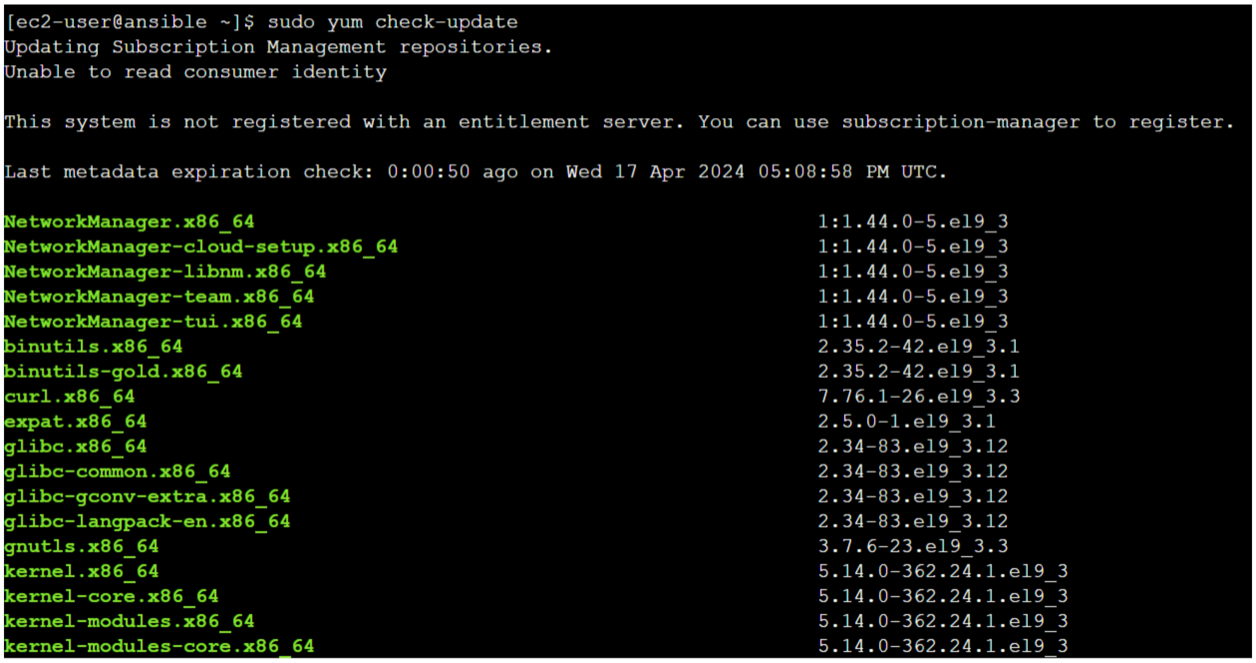


```
ubuntu@terraform:~/terraform-labs$ ssh -i /home/ubuntu/terraform-labs/mykey ec2-user@3.91.16
The authenticity of host '3.91.16.109 (3.91.16.109)' can't be established.
ED25519 key fingerprint is SHA256:uSjNYTbjuEG7i8nrUJY7IBfRz78xXr4YEV7RknpuBys.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.91.16.109' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-27-32 ~]$ sudo hostnamectl set-hostname ansible
[ec2-user@ip-172-31-27-32 ~]$ bash
```

2. Install Ansible and its dependencies

Update the package repository with the latest available versions

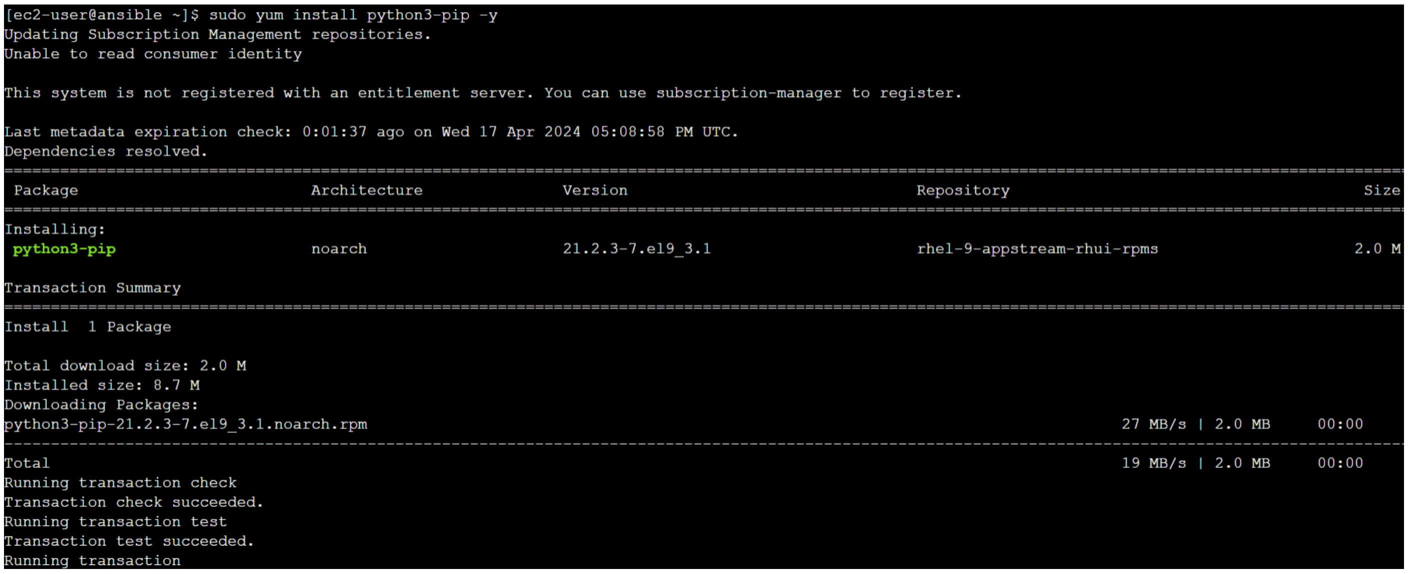
sudo yum check-update



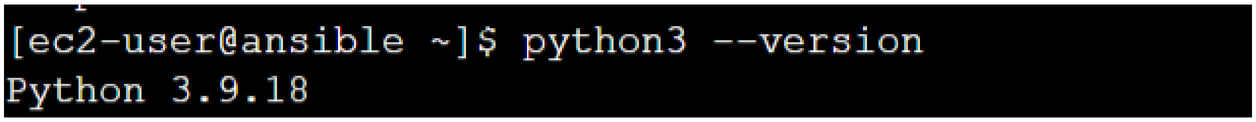
```
[ec2-user@ansible ~]$ sudo yum check-update
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:00:50 ago on Wed 17 Apr 2024 05:08:58 PM UTC.

NetworkManager.x86_64                                              1:1.44.0-5.el9_3
NetworkManager-cloud-setup.x86_64                                  1:1.44.0-5.el9_3
NetworkManager-libnm.x86_64                                        1:1.44.0-5.el9_3
NetworkManager-team.x86_64                                         1:1.44.0-5.el9_3
NetworkManager-tui.x86_64                                          1:1.44.0-5.el9_3
binutils.x86_64                                                    2.35.2-42.el9_3.1
binutils-gold.x86_64                                               2.35.2-42.el9_3.1
curl.x86_64                                                        7.76.1-26.el9_3.3
expat.x86_64                                                       2.5.0-1.el9_3.1
glibc.x86_64                                                       2.34-83.el9_3.12
glibc-common.x86_64                                                2.34-83.el9_3.12
glibc-gconv-extra.x86_64                                           2.34-83.el9_3.12
glibc-langpack-en.x86_64                                           2.34-83.el9_3.12
gnutls.x86_64                                                      3.7.6-23.el9_3.3
kernel.x86_64                                                      5.14.0-362.24.1.el9_3
kernel-core.x86_64                                                 5.14.0-362.24.1.el9_3
kernel-modules.x86_64                                              5.14.0-362.24.1.el9_3
kernel-modules-core.x86_64                                         5.14.0-362.24.1.el9_3
```

Install the latest version of Python.

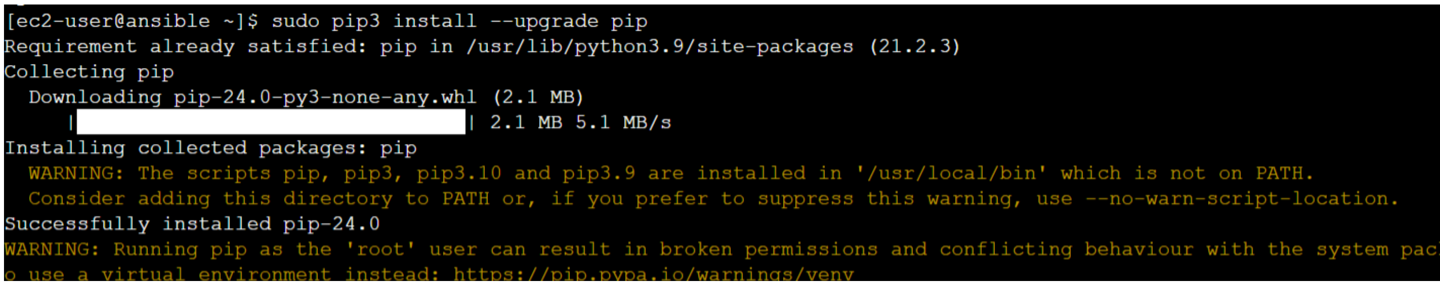
sudo yum install python3-pip -y


```
[ec2-user@ansible ~]$ sudo yum install python3-pip -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:01:37 ago on Wed 17 Apr 2024 05:08:58 PM UTC.
Dependencies resolved.
=====================================================================================================================================
 Package                        Architecture              Version                              Repository                       Size
=====================================================================================================================================
Installing:
 python3-pip                    noarch                    21.2.3-7.el9_3.1                     rhel-9-appstream-rhui-rpms      2.0 M

Transaction Summary
=====================================================================================================================================
Install  1 Package

Total download size: 2.0 M
Installed size: 8.7 M
Downloading Packages:
python3-pip-21.2.3-7.el9_3.1.noarch.rpm                                                             27 MB/s | 2.0 MB     00:00
-------------------------------------------------------------------------------------------------------------------------------------
Total                                                                                               19 MB/s | 2.0 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

python3 --version


```
[ec2-user@ansible ~]$ python3 --version
Python 3.9.18
```

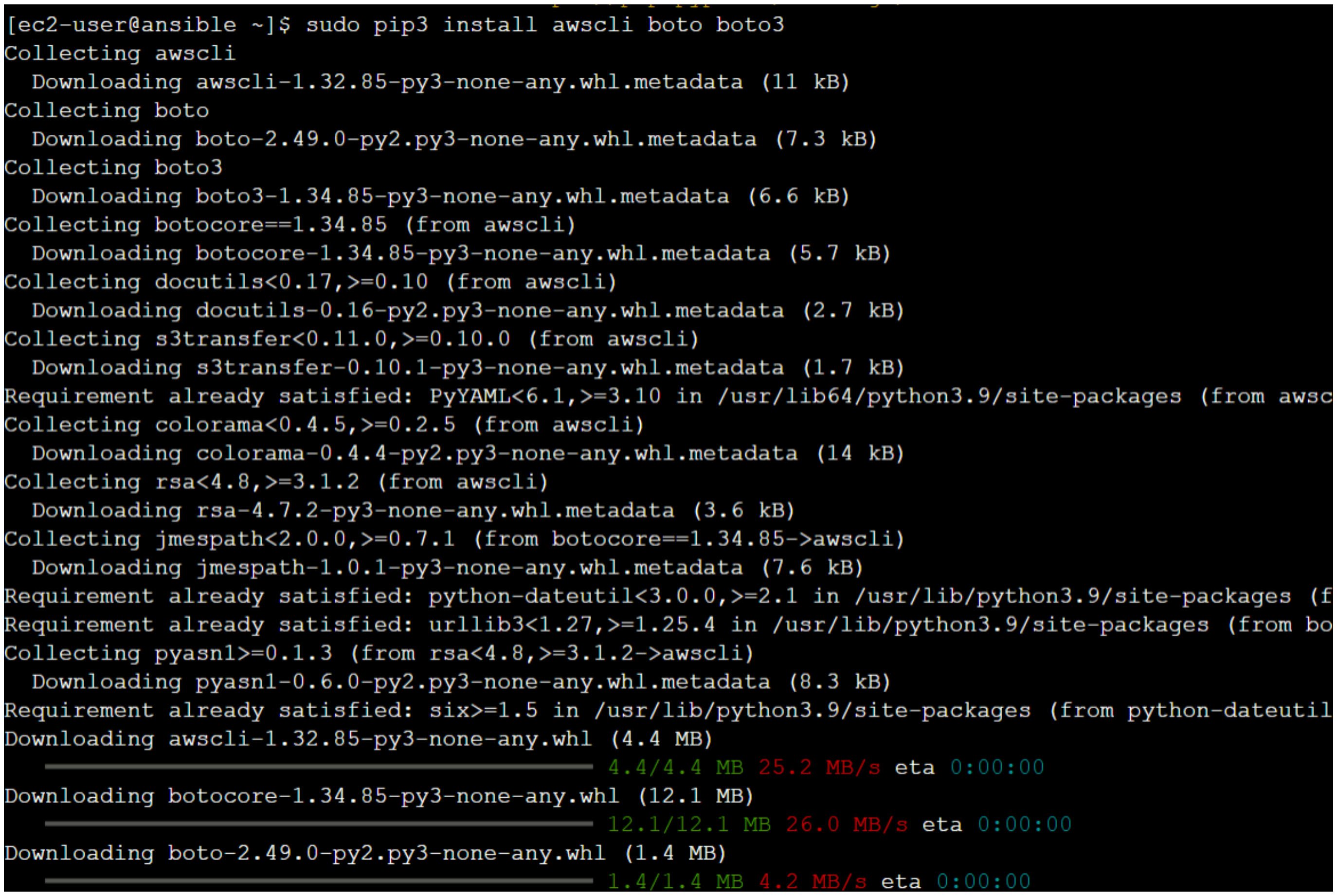
sudo pip3 install --upgrade pip


```
[ec2-user@ansible ~]$ sudo pip3 install --upgrade pip
Requirement already satisfied: pip in /usr/lib/python3.9/site-packages (21.2.3)
Collecting pip
  Downloading pip-24.0-py3-none-any.whl (2.1 MB)
     |████████████████████████████████| 2.1 MB 5.1 MB/s
Installing collected packages: pip
  WARNING: The scripts pip, pip3, pip3.10 and pip3.9 are installed in '/usr/local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-24.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system pac
o use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

Install AWS CLI, boto, boto3, and Ansible

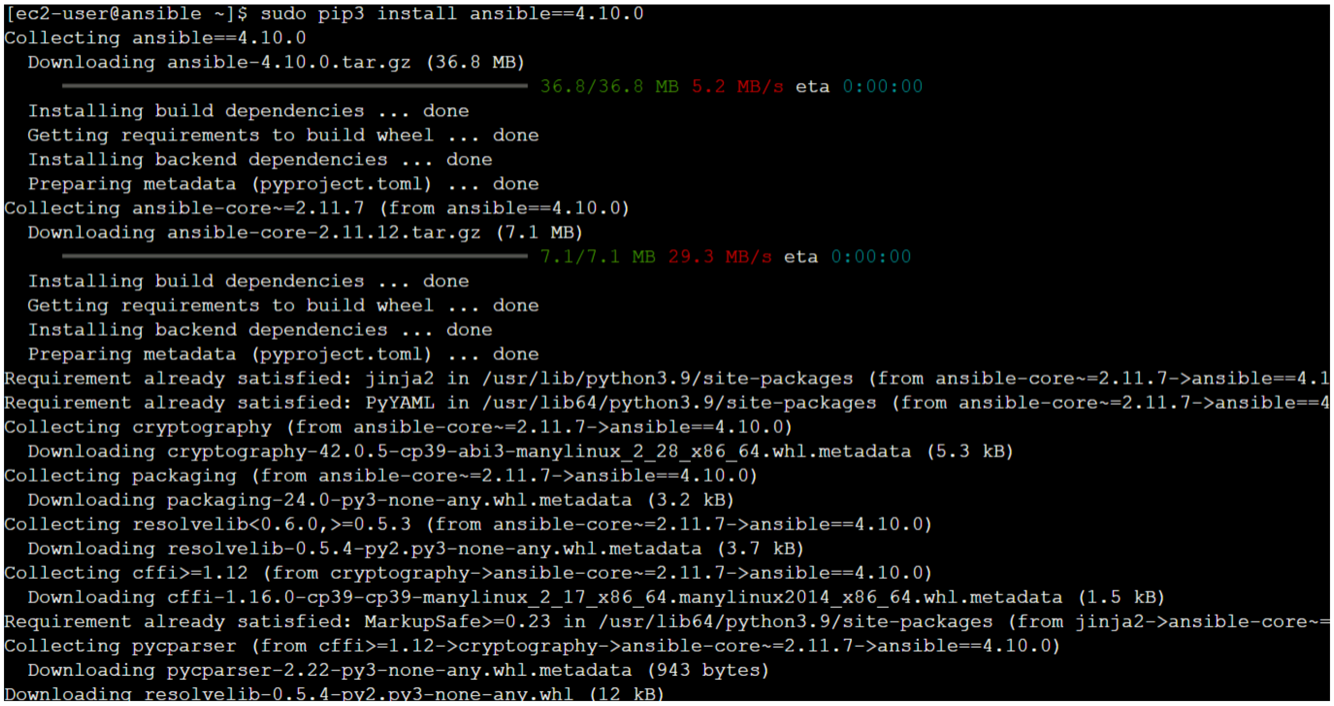
# Boto/Boto3 are AWS SDKs that will be needed while accessing AWS APIs

sudo pip3 install awscli boto boto3

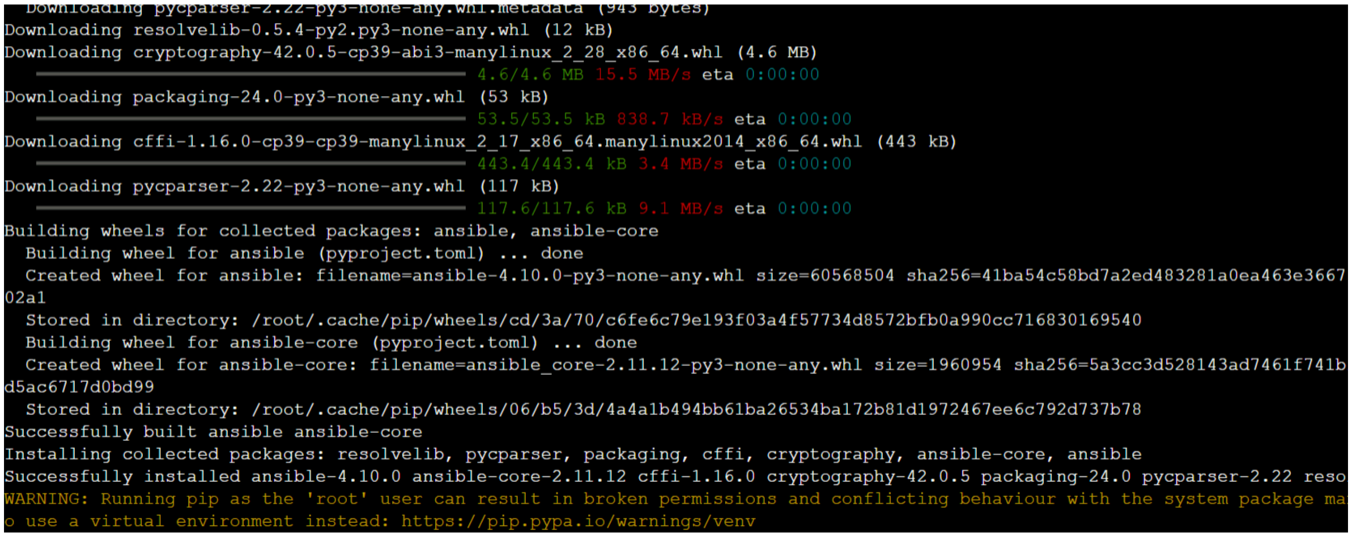


```
[ec2-user@ansible ~]$ sudo pip3 install awscli boto boto3
Collecting awscli
  Downloading awscli-1.32.85-py3-none-any.whl.metadata (11 kB)
Collecting boto
  Downloading boto-2.49.0-py2.py3-none-any.whl.metadata (7.3 kB)
Collecting boto3
  Downloading boto3-1.34.85-py3-none-any.whl.metadata (6.6 kB)
Collecting botocore==1.34.85 (from awscli)
  Downloading botocore-1.34.85-py3-none-any.whl.metadata (5.7 kB)
Collecting docutils<0.17,>=0.10 (from awscli)
  Downloading docutils-0.16-py2.py3-none-any.whl.metadata (2.7 kB)
Collecting s3transfer<0.11.0,>=0.10.0 (from awscli)
  Downloading s3transfer-0.10.1-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: PyYAML<6.1,>=3.10 in /usr/lib64/python3.9/site-packages (from awsc
Collecting colorama<0.4.5,>=0.2.5 (from awscli)
  Downloading colorama-0.4.4-py2.py3-none-any.whl.metadata (14 kB)
Collecting rsa<4.8,>=3.1.2 (from awscli)
  Downloading rsa-4.7.2-py3-none-any.whl.metadata (3.6 kB)
Collecting jmespath<2.0.0,>=0.7.1 (from botocore==1.34.85->awscli)
  Downloading jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/lib/python3.9/site-packages (f
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/lib/python3.9/site-packages (from bo
Collecting pyasn1>=0.1.3 (from rsa<4.8,>=3.1.2->awscli)
  Downloading pyasn1-0.6.0-py2.py3-none-any.whl.metadata (8.3 kB)
Requirement already satisfied: six>=1.5 in /usr/lib/python3.9/site-packages (from python-dateutil
Downloading awscli-1.32.85-py3-none-any.whl (4.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.4/4.4 MB 25.2 MB/s eta 0:00:00
Downloading botocore-1.34.85-py3-none-any.whl (12.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.1/12.1 MB 26.0 MB/s eta 0:00:00
Downloading boto-2.49.0-py2.py3-none-any.whl (1.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.4/1.4 MB 4.2 MB/s eta 0:00:00
```

sudo pip3 install ansible==4.10.0

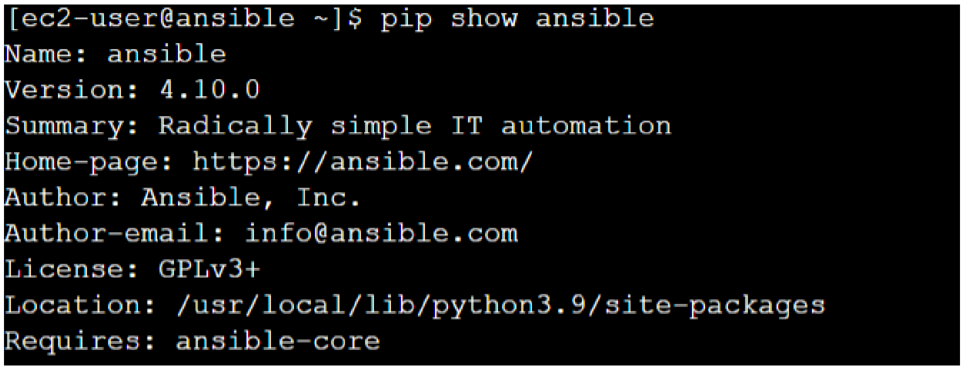


```
[ec2-user@ansible ~]$ sudo pip3 install ansible==4.10.0
Collecting ansible==4.10.0
  Downloading ansible-4.10.0.tar.gz (36.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 36.8/36.8 MB 5.2 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Collecting ansible-core~=2.11.7 (from ansible==4.10.0)
  Downloading ansible-core-2.11.12.tar.gz (7.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.1/7.1 MB 29.3 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: jinja2 in /usr/lib/python3.9/site-packages (from ansible-core~=2.11.7->ansible==4.1
Requirement already satisfied: PyYAML in /usr/lib64/python3.9/site-packages (from ansible-core~=2.11.7->ansible==4
Collecting cryptography (from ansible-core~=2.11.7->ansible==4.10.0)
  Downloading cryptography-42.0.5-cp39-abi3-manylinux_2_28_x86_64.whl.metadata (5.3 kB)
Collecting packaging (from ansible-core~=2.11.7->ansible==4.10.0)
  Downloading packaging-24.0-py3-none-any.whl.metadata (3.2 kB)
Collecting resolvelib<0.6.0,>=0.5.3 (from ansible-core~=2.11.7->ansible==4.10.0)
  Downloading resolvelib-0.5.4-py2.py3-none-any.whl.metadata (3.7 kB)
Collecting cffi>=1.12 (from cryptography->ansible-core~=2.11.7->ansible==4.10.0)
  Downloading cffi-1.16.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (1.5 kB)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/lib64/python3.9/site-packages (from jinja2->ansible-core~=
Collecting pycparser (from cffi>=1.12->cryptography->ansible-core~=2.11.7->ansible==4.10.0)
  Downloading pycparser-2.22-py3-none-any.whl.metadata (943 bytes)
Downloading resolvelib-0.5.4-py2.py3-none-any.whl (12 kB)
```



```
  Downloading pycparser-2.22-py3-none-any.whl.metadata (943 bytes)
Downloading resolvelib-0.5.4-py2.py3-none-any.whl (12 kB)
Downloading cryptography-42.0.5-cp39-abi3-manylinux_2_28_x86_64.whl (4.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.6/4.6 MB 15.5 MB/s eta 0:00:00
Downloading packaging-24.0-py3-none-any.whl (53 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.5/53.5 kB 838.7 kB/s eta 0:00:00
Downloading cffi-1.16.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (443 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 443.4/443.4 kB 3.4 MB/s eta 0:00:00
Downloading pycparser-2.22-py3-none-any.whl (117 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 117.6/117.6 kB 9.1 MB/s eta 0:00:00
Building wheels for collected packages: ansible, ansible-core
  Building wheel for ansible (pyproject.toml) ... done
  Created wheel for ansible: filename=ansible-4.10.0-py3-none-any.whl size=60568504 sha256=41ba54c58bd7a2ed483281a0ea463e3667
02a1
  Stored in directory: /root/.cache/pip/wheels/cd/3a/70/c6fe6c79e193f03a4f57734d8572bfb0a990cc716830169540
  Building wheel for ansible-core (pyproject.toml) ... done
  Created wheel for ansible-core: filename=ansible_core-2.11.12-py3-none-any.whl size=1960954 sha256=5a3cc3d528143ad7461f741b
d5ac6717d0bd99
  Stored in directory: /root/.cache/pip/wheels/06/b5/3d/4a4a1b494bb61ba26534ba172b81d1972467ee6c792d737b78
Successfully built ansible ansible-core
Installing collected packages: resolvelib, pycparser, packaging, cffi, cryptography, ansible-core, ansible
Successfully installed ansible-4.10.0 ansible-core-2.11.12 cffi-1.16.0 cryptography-42.0.5 packaging-24.0 pycparser-2.22 reso
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package ma
o use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

pip show ansible



```
[ec2-user@ansible ~]$ pip show ansible
Name: ansible
Version: 4.10.0
Summary: Radically simple IT automation
Home-page: https://ansible.com/
Author: Ansible, Inc.
Author-email: info@ansible.com
License: GPLv3+
Location: /usr/local/lib/python3.9/site-packages
Requires: ansible-core
```

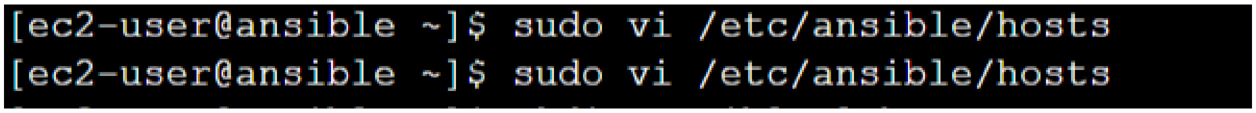
3. Create and populate the inv file

Create the Inventory File: Use a text editor

sudo vi /etc/ansible/hosts

# Add the given line, by pressing "INSERT"
# Add localhost and add the connection as local
localhost ansible_connection=local
# save the file using "ESCAPE + :wq!"

Note-   :1,%d, To erase the complete information from the VI editor

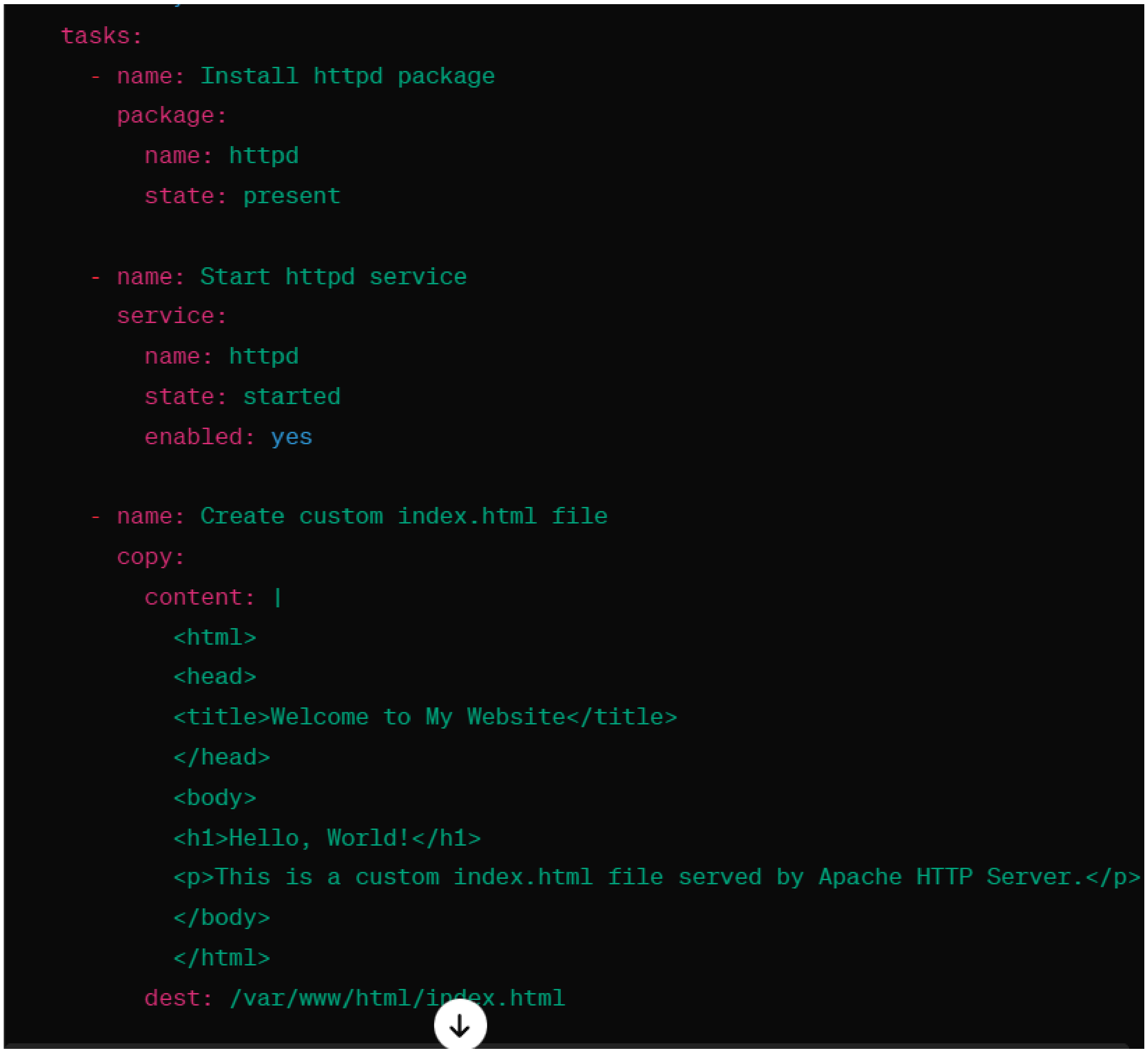


```
[ec2-user@ansible ~]$ sudo vi /etc/ansible/hosts
[ec2-user@ansible ~]$ sudo vi /etc/ansible/hosts
```

4. Yaml file - setup an httpd web server with a custom index.html

vi install_httpd.yml

```
---
- name: Setup HTTPD Web Server with Custom Index.html
  hosts: localhost
  become: yes
  tasks:
    - name: Install httpd package
      package:
        name: httpd
        state: present

    - name: Start httpd service
      service:
        name: httpd
        state: started
        enabled: yes

    - name: Create custom index.html file
      copy:
        content: |
          <html>
          <head>
          <title>Welcome to My Website</title>
          </head>
          <body>
          <h1>Hello, World!</h1>
          <p>This is a custom index.html file served by Apache HTTP Server.</p>
          </body>
          </html>
        dest: /var/www/html/index.html
```



```
  tasks:
    - name: Install httpd package
      package:
        name: httpd
        state: present

    - name: Start httpd service
      service:
        name: httpd
        state: started
        enabled: yes

    - name: Create custom index.html file
      copy:
        content: |
          <html>
          <head>
          <title>Welcome to My Website</title>
          </head>
          <body>
          <h1>Hello, World!</h1>
          <p>This is a custom index.html file served by Apache HTTP Server.</p>
          </body>
          </html>
        dest: /var/www/html/index.html
```

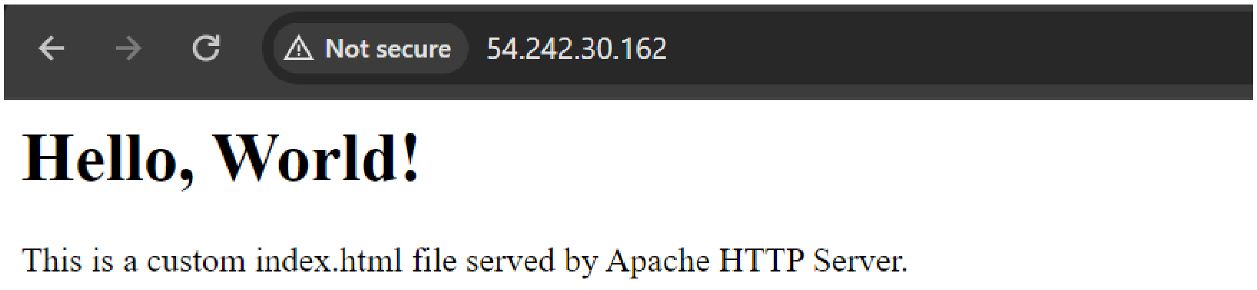
5. execute it

ansible-playbook install_httpd.yml
#This command will execute the playbook, which installs the httpd web server on the localhost.

6. verify the custom index.html

Open a web browser on your local machine.
Enter the IP address or hostname of your Ansible workstation in the address bar.
You should see the custom index.html page served by the Apache HTTP Server running on the Ansible workstation.

http://54.242.30.162/



Not secure 54.242.30.162

Hello, World!

This is a custom index.html file served by Apache HTTP Server.

Check httpd Service Status:
Verify that the httpd service is running on your workstation. You can do this by running the following command in your terminal:

```
systemctl status httpd
```

If the service is not running, you can start it with:

```
sudo systemctl start httpd
```

Make sure to enable the service to start automatically on boot:

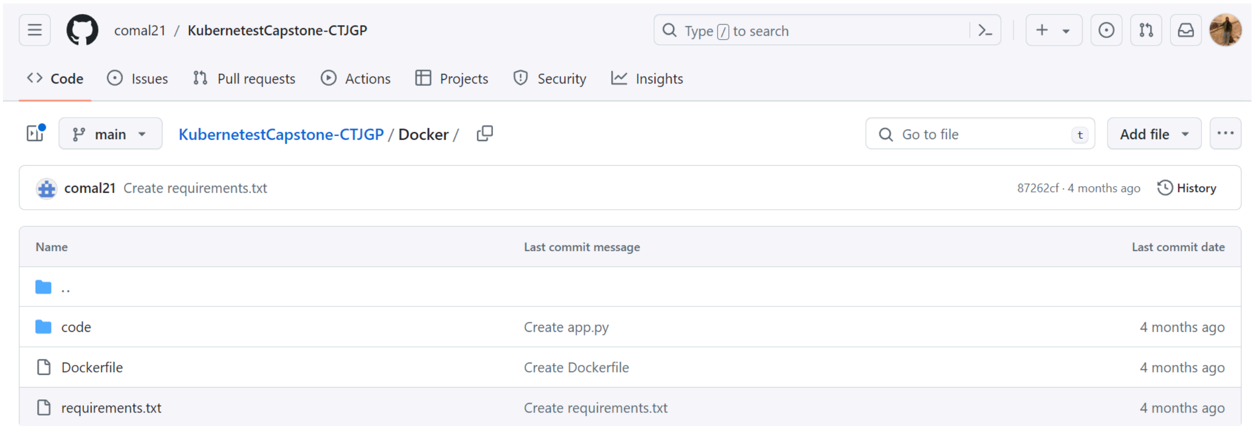
```
sudo systemctl enable httpd
```

TASK 3 - Docker & Kubernetes Task:

Build a docker image to use the Python API and push it to the DockerHub. Create a pod and node port service with that Docker image.

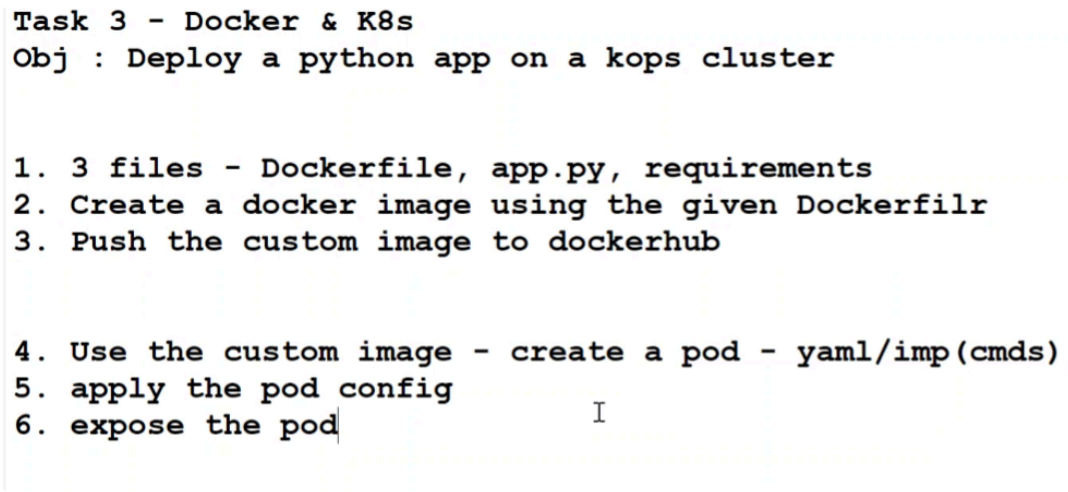
Hint: A KOPS cluster will be provided to you. You can use the worker nodes to write DockerFile and build an image
Hint: Use the DockerFile provided to you if needed to create the Docker image

https://github.com/comal21/KubernetestCapstone-CTJGP/tree/main/Docker



comal21 Create requirements.txt 87262cf · 4 months ago

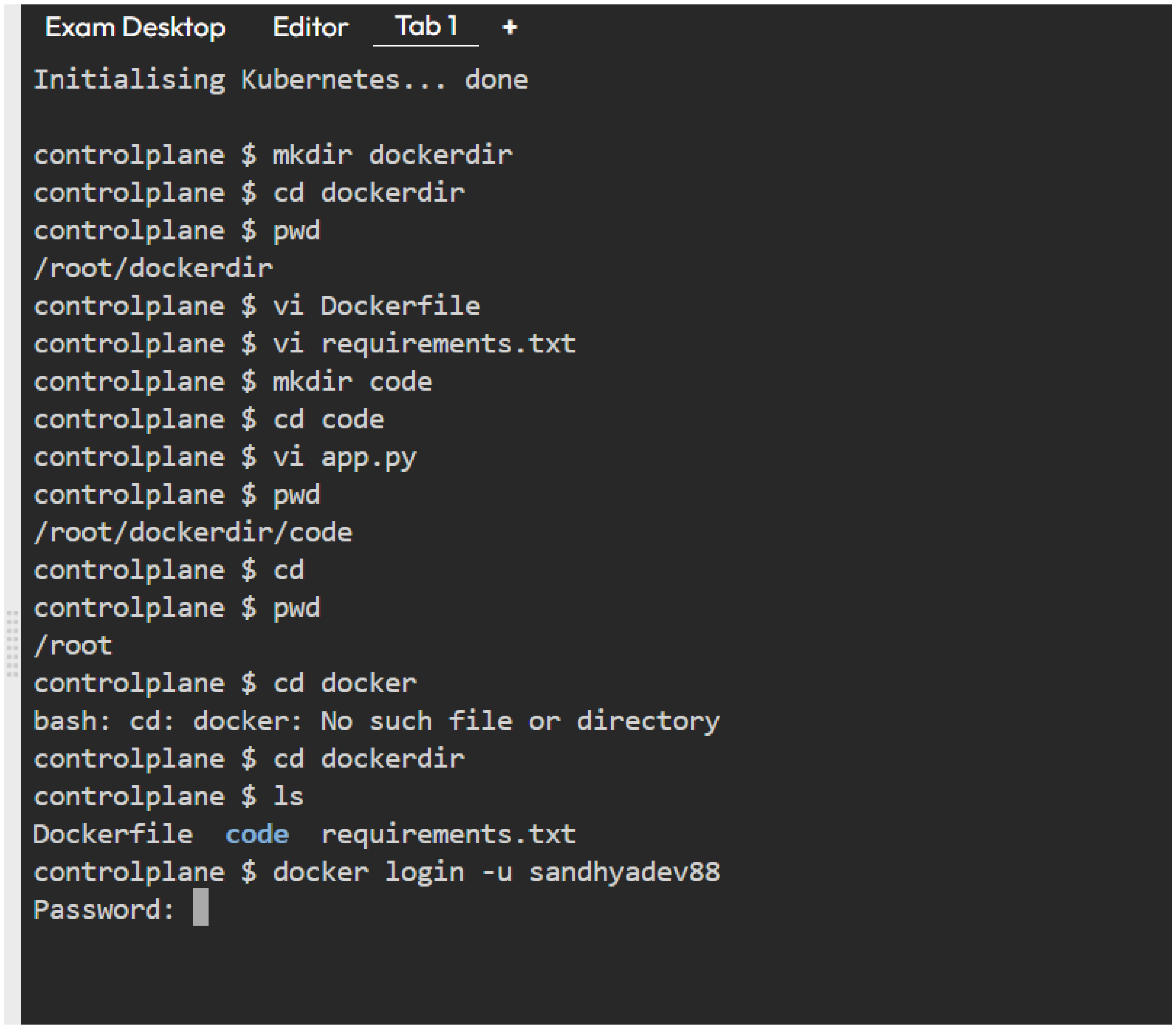
Name	Last commit message	Last commit date
..		
code	Create app.py	4 months ago
Dockerfile	Create Dockerfile	4 months ago
requirements.txt	Create requirements.txt	4 months ago



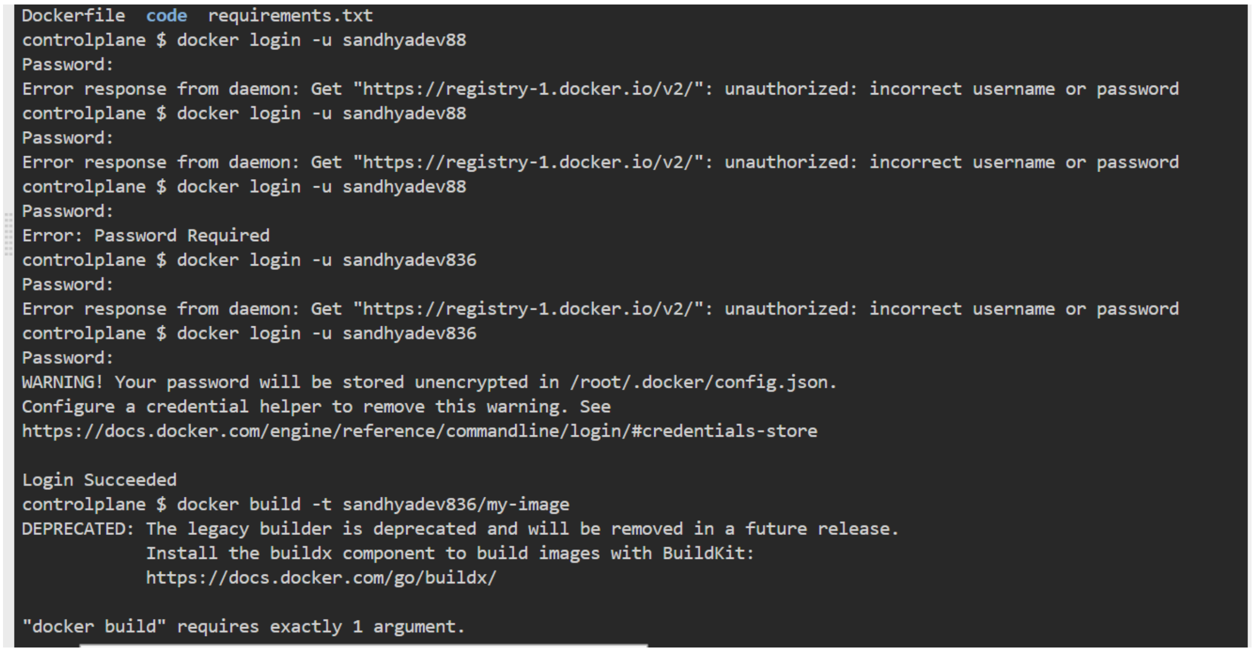
```
Task 3 - Docker & K8s
Obj : Deploy a python app on a kops cluster


1. 3 files - Dockerfile, app.py, requirements
2. Create a docker image using the given Dockerfilr
3. Push the custom image to dockerhub


4. Use the custom image - create a pod - yaml/imp(cmds)
5. apply the pod config
6. expose the pod
```

Due to the short timeline, the project was completed in Killercoda.

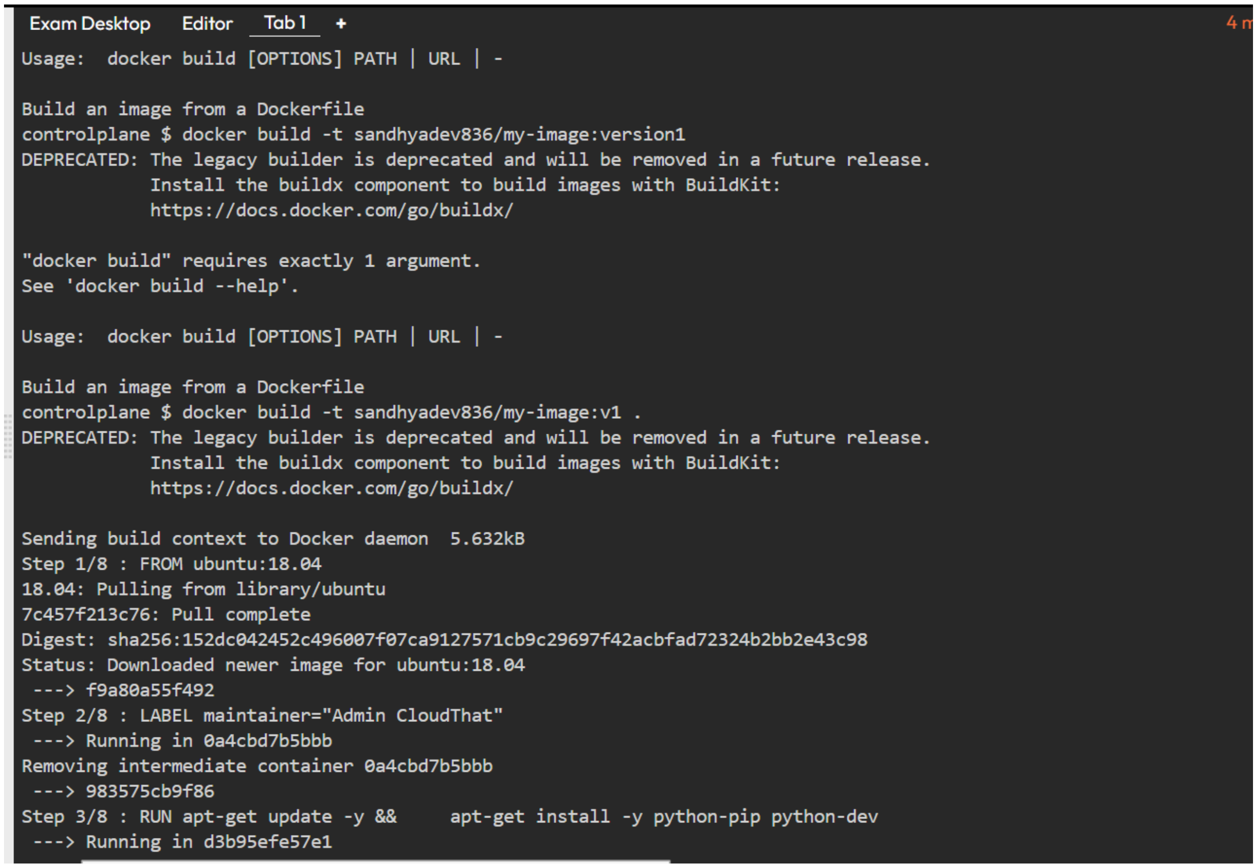
Building a Docker Image:
Using the Dockerfile, you'll build a Docker image. This image will contain the Python API and all the necessary dependencies packaged together.
Pushing to DockerHub: Once the Docker image is built, you'll make it to DockerHub, which is a cloud-based registry service where you can share Docker images.
Creating a Kubernetes Cluster: You'll use KOPS, which stands for Kubernetes Operations, to create a Kubernetes cluster. KOPS helps you create, destroy, upgrade, and maintain production-grade, highly available, Kubernetes clusters from the command line.
Deploying a Pod: You'll create a pod that runs your Docker image.
Exposing the Pod with a NodePort Service: To make your application accessible from outside the Kubernetes cluster, you'll create a NodePort service. This type of service exposes the pod to the external network by opening a specific port on all nodes (VMs) and any traffic sent to this port is forwarded to the service.



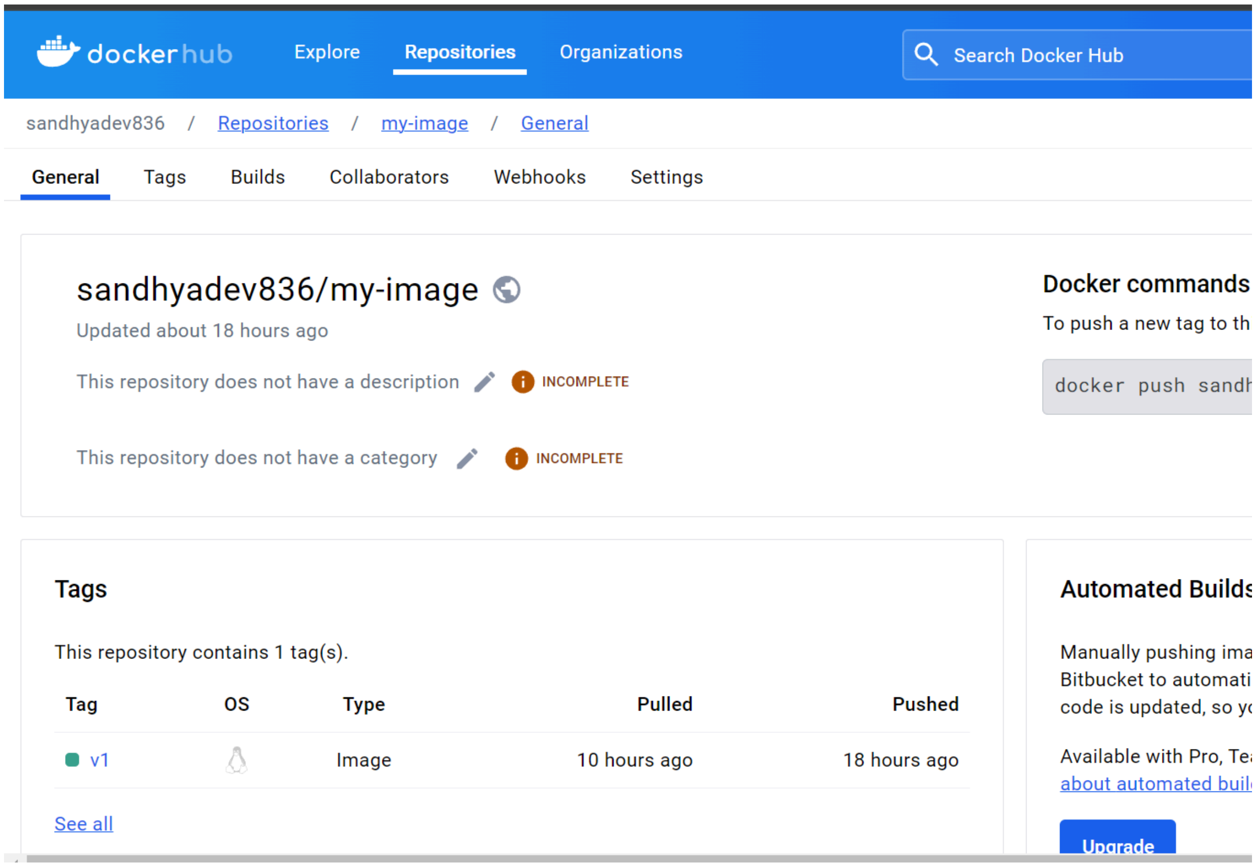
```
Initialising Kubernetes... done

controlplane $ mkdir dockerdir
controlplane $ cd dockerdir
controlplane $ pwd
/root/dockerdir
controlplane $ vi Dockerfile
controlplane $ vi requirements.txt
controlplane $ mkdir code
controlplane $ cd code
controlplane $ vi app.py
controlplane $ pwd
/root/dockerdir/code
controlplane $ cd
controlplane $ pwd
/root
controlplane $ cd docker
bash: cd: docker: No such file or directory
controlplane $ cd dockerdir
controlplane $ ls
Dockerfile   code   requirements.txt
controlplane $ docker login -u sandhyadev88
Password:
```



```
Dockerfile  code  requirements.txt
controlplane $ docker login -u sandhyadev88
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
controlplane $ docker login -u sandhyadev88
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
controlplane $ docker login -u sandhyadev88
Password:
Error: Password Required
controlplane $ docker login -u sandhyadev836
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
controlplane $ docker login -u sandhyadev836
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
controlplane $ docker build -t sandhyadev836/my-image
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

"docker build" requires exactly 1 argument.
```



```
Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
controlplane $ docker build -t sandhyadev836/my-image:version1
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

"docker build" requires exactly 1 argument.
See 'docker build --help'.

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
controlplane $ docker build -t sandhyadev836/my-image:v1 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  5.632kB
Step 1/8 : FROM ubuntu:18.04
18.04: Pulling from library/ubuntu
7c457f213c76: Pull complete
Digest: sha256:152dc042452c496007f07ca9127571cb9c29697f42acbfad72324b2bb2e43c98
Status: Downloaded newer image for ubuntu:18.04
 ---> f9a80a55f492
Step 2/8 : LABEL maintainer="Admin CloudThat"
 ---> Running in 0a4cbd7b5bbb
Removing intermediate container 0a4cbd7b5bbb
 ---> 983575cb9f86
Step 3/8 : RUN apt-get update -y &&     apt-get install -y python-pip python-dev
 ---> Running in d3b95efe57e1
```



sandhyadev836 / Repositories / my-image / General

sandhyadev836/my-image

Updated about 18 hours ago

This repository does not have a description INCOMPLETE

This repository does not have a category INCOMPLETE

Docker commands

To push a new tag to th

```
docker push sandh
```

Tags

This repository contains 1 tag(s).

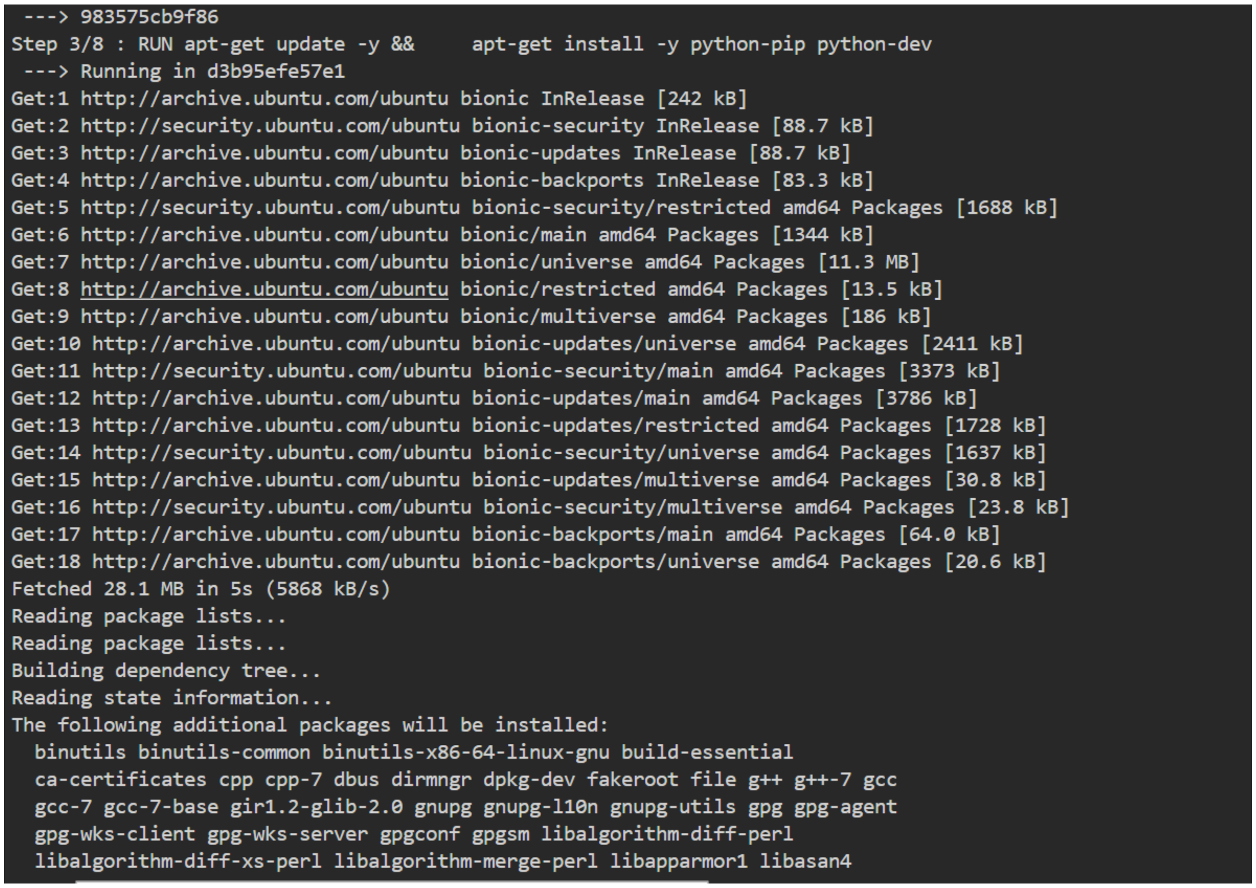
Tag	OS	Type	Pulled	Pushed
v1		Image	10 hours ago	18 hours ago

See all

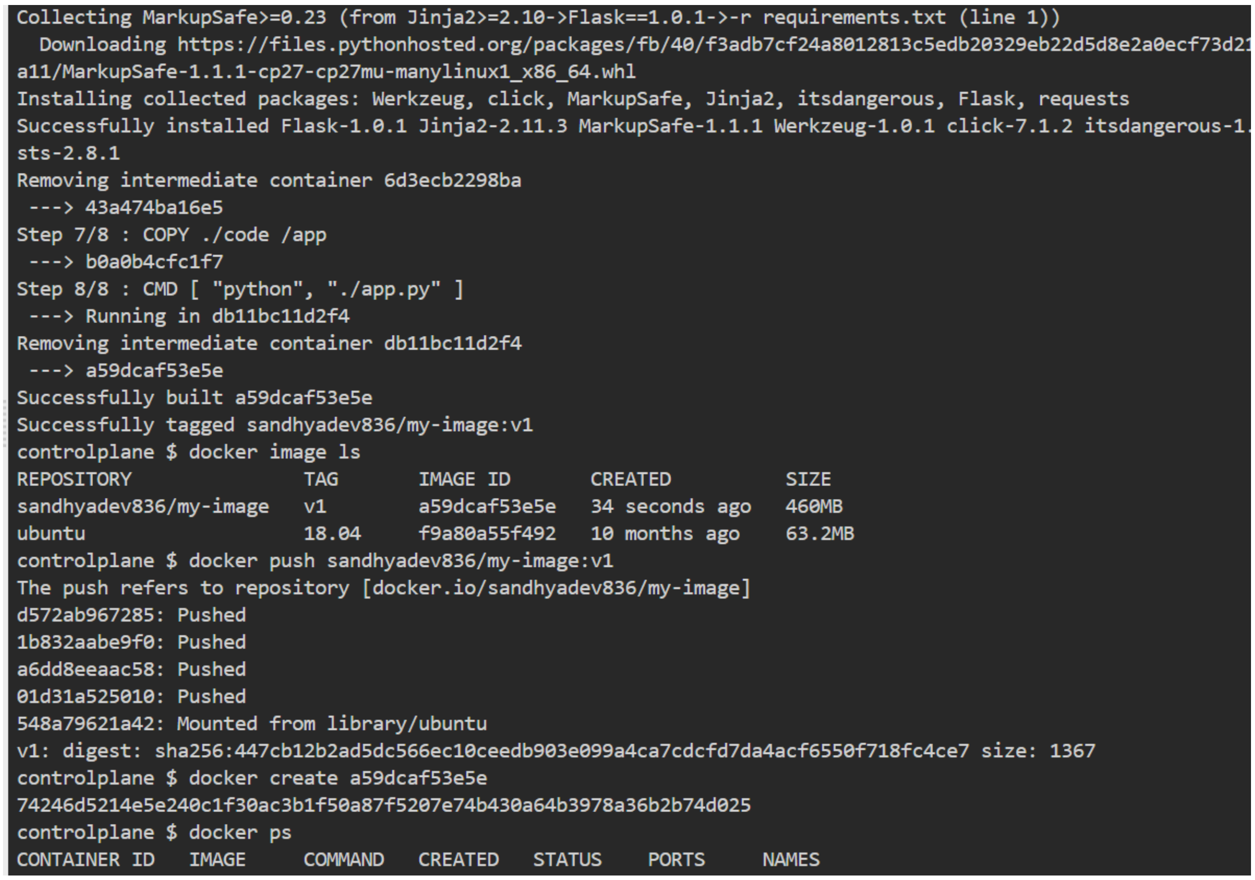
Automated Builds

Manually pushing ima
Bitbucket to automati
code is updated, so yo

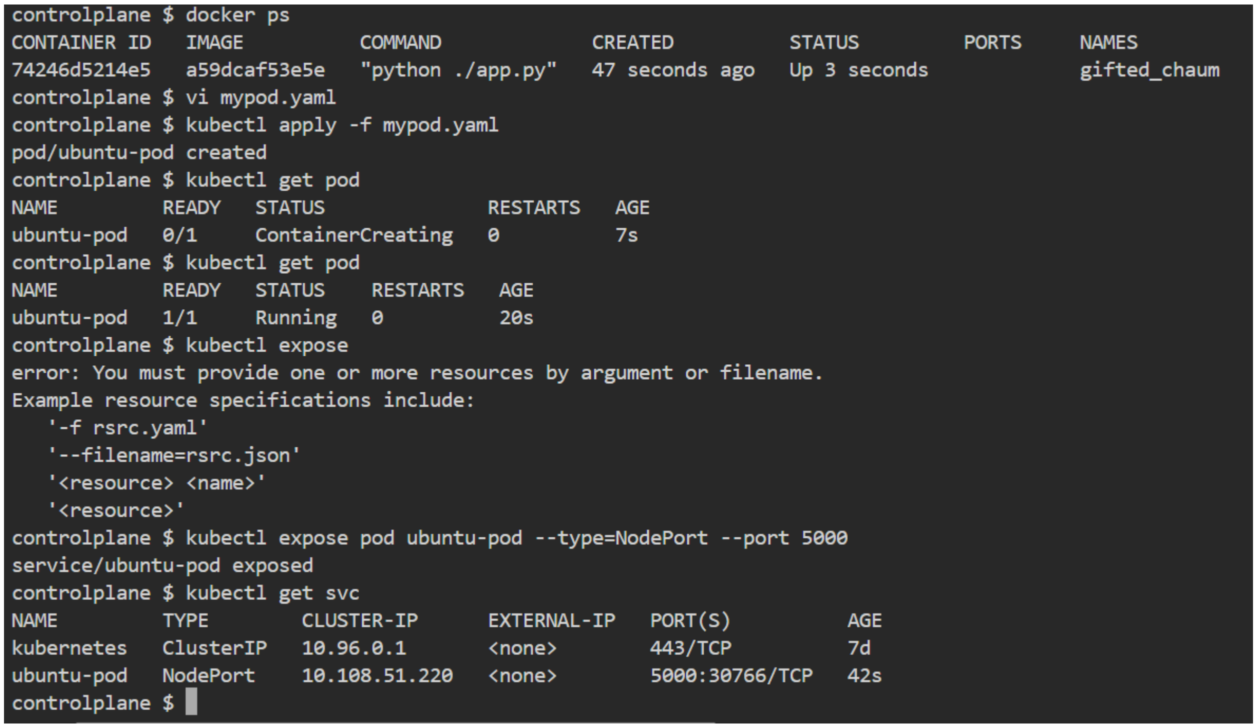
Available with Pro, Tea
about automated buil



```
 ---> 983575cb9f86
Step 3/8 : RUN apt-get update -y &&     apt-get install -y python-pip python-dev
 ---> Running in d3b95efe57e1
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [1688 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [2411 kB]
Get:11 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [3373 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [3786 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [1728 kB]
Get:14 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1637 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [30.8 kB]
Get:16 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [23.8 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [64.0 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [20.6 kB]
Fetched 28.1 MB in 5s (5868 kB/s)
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential
  ca-certificates cpp cpp-7 dbus dirmngr dpkg-dev fakeroot file g++ g++-7 gcc
  gcc-7 gcc-7-base gir1.2-glib-2.0 gnupg gnupg-l10n gnupg-utils gpg gpg-agent
  gpg-wks-client gpg-wks-server gpgconf gpgsm libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libapparmor1 libasan4
```



```
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10->Flask==1.0.1->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/fb/40/f3adb7cf24a8012813c5edb20329eb22d5d8e2a0ecf73d21
a11/MarkupSafe-1.1.1-cp27-cp27mu-manylinux1_x86_64.whl
Installing collected packages: Werkzeug, click, MarkupSafe, Jinja2, itsdangerous, Flask, requests
Successfully installed Flask-1.0.1 Jinja2-2.11.3 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 itsdangerous-1.
sts-2.8.1
Removing intermediate container 6d3ecb2298ba
 ---> 43a474ba16e5
Step 7/8 : COPY ./code /app
 ---> b0a0b4cfc1f7
Step 8/8 : CMD [ "python", "./app.py" ]
 ---> Running in db11bc11d2f4
Removing intermediate container db11bc11d2f4
 ---> a59dcaf53e5e
Successfully built a59dcaf53e5e
Successfully tagged sandhyadev836/my-image:v1
controlplane $ docker image ls
REPOSITORY               TAG       IMAGE ID       CREATED          SIZE
sandhyadev836/my-image   v1        a59dcaf53e5e   34 seconds ago   460MB
ubuntu                   18.04     f9a80a55f492   10 months ago    63.2MB
controlplane $ docker push sandhyadev836/my-image:v1
The push refers to repository [docker.io/sandhyadev836/my-image]
d572ab967285: Pushed
1b832aabe9f0: Pushed
a6dd8eeaac58: Pushed
01d31a525010: Pushed
548a79621a42: Mounted from library/ubuntu
v1: digest: sha256:447cb12b2ad5dc566ec10ceedb903e099a4ca7cdcfd7da4acf6550f718fc4ce7 size: 1367
controlplane $ docker create a59dcaf53e5e
74246d5214e5e240c1f30ac3b1f50a87f5207e74b430a64b3978a36b2b74d025
controlplane $ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```



```
controlplane $ docker ps
CONTAINER ID   IMAGE          COMMAND             CREATED          STATUS         PORTS     NAMES
74246d5214e5   a59dcaf53e5e   "python ./app.py"   47 seconds ago   Up 3 seconds             gifted_chaum
controlplane $ vi mypod.yaml
controlplane $ kubectl apply -f mypod.yaml
pod/ubuntu-pod created
controlplane $ kubectl get pod
NAME         READY   STATUS              RESTARTS   AGE
ubuntu-pod   0/1     ContainerCreating   0          7s
controlplane $ kubectl get pod
NAME         READY   STATUS    RESTARTS   AGE
ubuntu-pod   1/1     Running   0          20s
controlplane $ kubectl expose
error: You must provide one or more resources by argument or filename.
Example resource specifications include:
   '-f rsrc.yaml'
   '--filename=rsrc.json'
   '<resource> <name>'
   '<resource>'
controlplane $ kubectl expose pod ubuntu-pod --type=NodePort --port 5000
service/ubuntu-pod exposed
controlplane $ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP          7d
ubuntu-pod   NodePort    10.108.51.220   <none>        5000:30766/TCP   42s
controlplane $ 
```

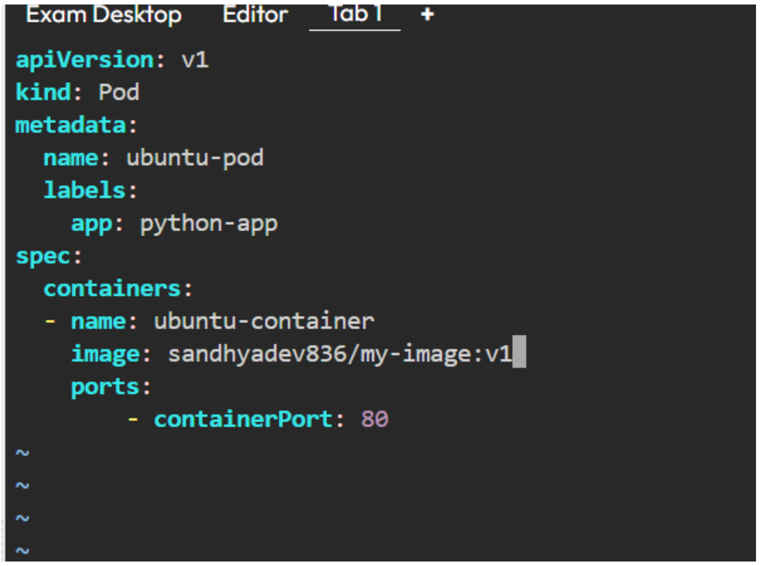
Use of NodePort:

- NodePort service in Kubernetes is a service that is used to **expose the application to the internet**, from where end-users can access it.
- When you create a NodePort service, Kubernetes allocates a port from a **default range (30000-32767)**. This port is opened on every node in your cluster. The application can be accessed by end-users using the node's IP address.
- You can access your application using the IP address of any node in the cluster, along with the allocated NodePort. For example, if a node's IP address is 192.168.1.1 and the NodePort is 31000, you would access your application via http://192.168.1.1:31000
- Any traffic that comes to this "NodePort" on any node gets routed to your service, and then to the pod(s) running your application

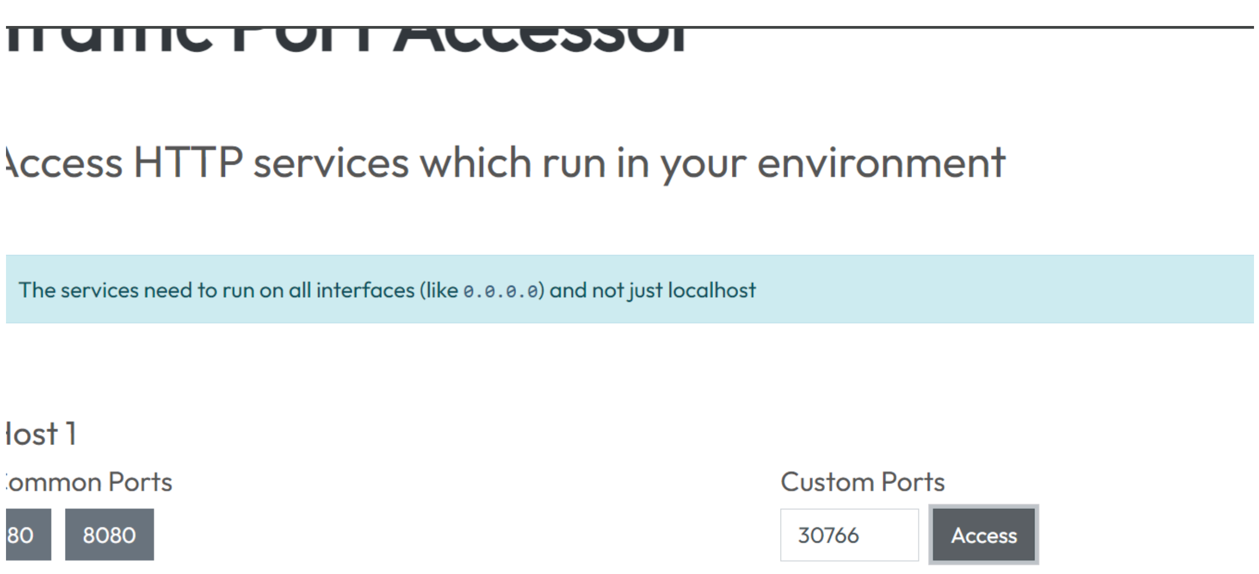
Use of ClusterIP: K8S service, which **exposes the service on an internal IP** in the cluster. It makes the service only reachable from **within the cluster**.

vi mypod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-pod
  labels:
     app: python-app
spec:
  containers:
  - name: ubuntu-container
    image: ubuntu
    ports:
       - containerPort: 80
```



```
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-pod
  labels:
    app: python-app
spec:
  containers:
  - name: ubuntu-container
    image: sandhyadev836/my-image:v1
    ports:
        - containerPort: 80
~
~
~
~
```



raffic Port Accessor

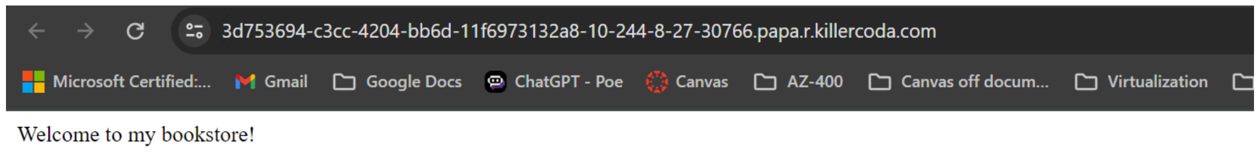
ccess HTTP services which run in your environment

The services need to run on all interfaces (like `0.0.0.0`) and not just localhost

ost 1

ommon Ports

Custom Ports



3d753694-c3cc-4204-bb6d-11f6973132a8-10-244-8-27-30766.papa.r.killercoda.com

Welcome to my bookstore!

COMMANDS:

```
controlplane $ mkdir code

controlplane $ cd code

controlplane $ vi app.py

controlplane $ pwd

/root/dockerdir/code

controlplane $ cd

controlplane $ pwd
/root

controlplane $ cd dockerdir
controlplane $ ls
Dockerfile  code  requirements.txt

controlplane $ docker login -u sandhyadev836
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

controlplane $ docker build -t sandhyadev836/my-image:v1 .
```

Note: The dot at the end specifies the build context as the current directory, which should contain the Dockerfile.

```
Successfully tagged sandhyadev836/my-image:v1

controlplane $ docker image ls
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
sandhyadev836/my-image   v1       a59dcaf53e5e   34 seconds ago   460MB
ubuntu                 18.04     f9a80a55f492   10 months ago    63.2MB

controlplane $ docker push sandhyadev836/my-image:v1
```

This will upload the image to your DockerHub repository, making it available for others to pull and use

```
The push refers to repository [docker.io/sandhyadev836/my-image]
d572ab967285: Pushed
1b832aabe9f0: Pushed
a6dd8eeaac58: Pushed
01d31a525010: Pushed
548a79621a42: Mounted from library/ubuntu
v1: digest: sha256:447cb12b2ad5dc566ec10ceedb903e099a4ca7cdcfd7da4acf6550f718fc4ce7 size: 1367

controlplane $ docker create a59dcaf53e5e
74246d5214e5e240c1f30ac3b1f50a87f5207e74b430a64b3978a36b2b74d025

controlplane $ docker ps -a
CONTAINER ID   IMAGE          COMMAND            CREATED          STATUS    PORTS     NAMES
74246d5214e5   a59dcaf53e5e   "python ./app.py"   29 seconds ago   Created             gifted_chaum

controlplane $ docker start 74246d5214e5
74246d5214e5

controlplane $ docker ps
CONTAINER ID   IMAGE          COMMAND            CREATED          STATUS         PORTS     NAMES
74246d5214e5   a59dcaf53e5e   "python ./app.py"   47 seconds ago   Up 3 seconds             gifted_chaum
```

```
controlplane $ vi mypod.yaml

controlplane $ kubectl apply -f mypod.yaml
pod/ubuntu-pod created

controlplane $ kubectl get pod
NAME        READY   STATUS            RESTARTS   AGE
ubuntu-pod   0/1     ContainerCreating   0          7s

controlplane $ kubectl get pod
NAME        READY   STATUS    RESTARTS   AGE
ubuntu-pod   1/1     Running   0          20s

controlplane $ kubectl expose pod ubuntu-pod --type=NodePort --port 5000

service/ubuntu-pod exposed

controlplane $ kubectl get svc
NAME        TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP          7d
ubuntu-pod   NodePort    10.108.51.220   <none>        5000:30766/TCP   42s
```

Traffic that comes to port 30766 on any node in the Kubernetes cluster will be forwarded to port 5000 of the Ubuntu-pod.