Automating Infrastructure using Terraform

Project 1

DESCRIPTION

Use Terraform to provision infrastructure

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 centralized 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 install 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, Key pair

Expected Deliverables:

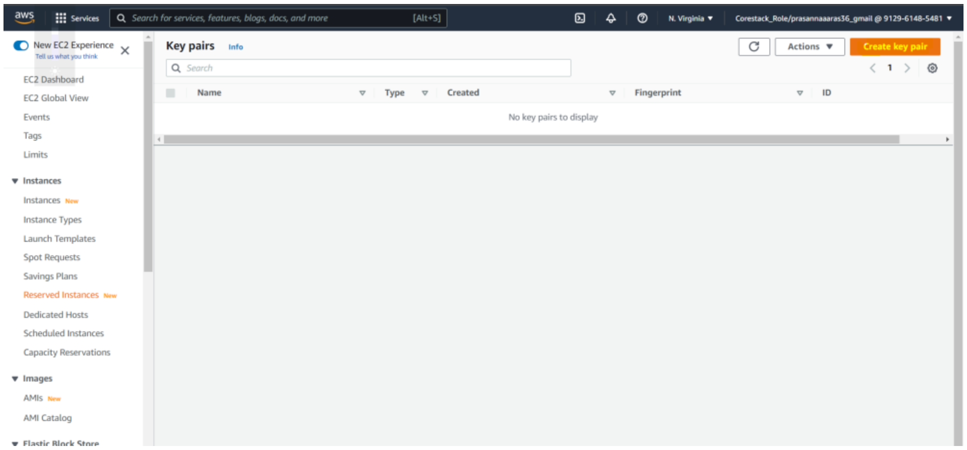
- Launch an EC2 instance using Terraform
- Connect to the instance
- Install Jenkins, Java and Python in the instance

NOTE: This Project was completed from my Personal System.

Procedure:

Step 1: Get the AWS web services Access key and Secret.

Step 2: Launch your AWS web console and navigate to Key-Pair. To generate a Key-pair Value for the server login once the EC2 instance in launched. Since AWS doesn’t allow password login initially.

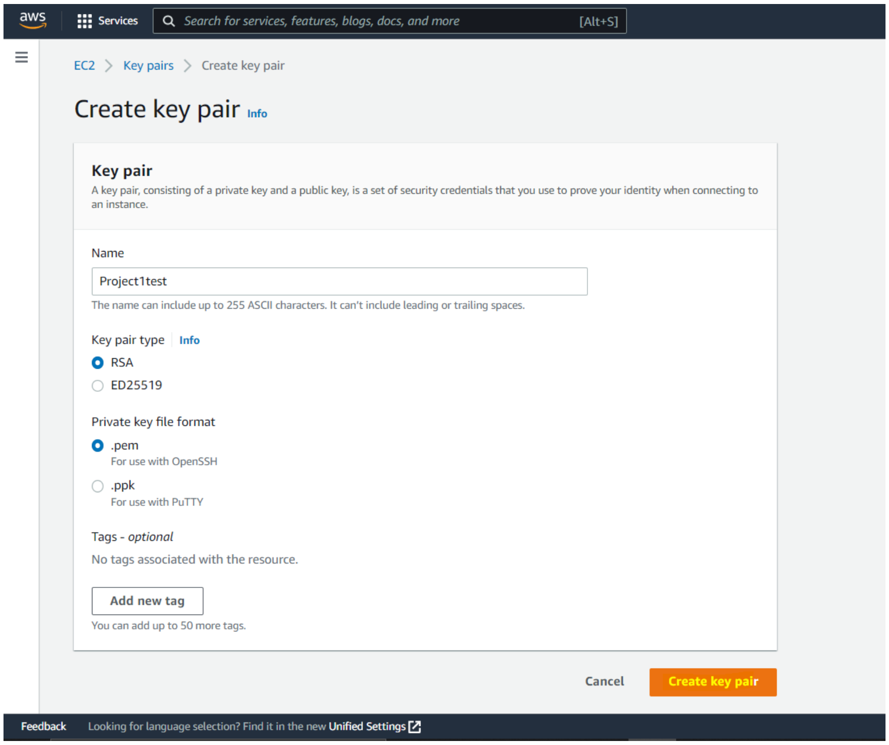


Key pairs

Name	Type	Created	Fingerprint	ID

No key pairs to display

Generate the Key-Value Pair and download the .pem.


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
EC2 > Key pairs > Create key pair
Info
Add new tag
Cancel
Create key pair
Feedback
Looking for language selection? Find it in the new Unified Settings

Create key pair

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

Project1test

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type Info

● RSA

○ ED25519

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

Tags - *optional*

No tags associated with the resource.

You can add up to 50 more tags.

Once you click on create key pair it will generate a certificate which we can use to login to the server later once deployed. Attached is the sample for the same.

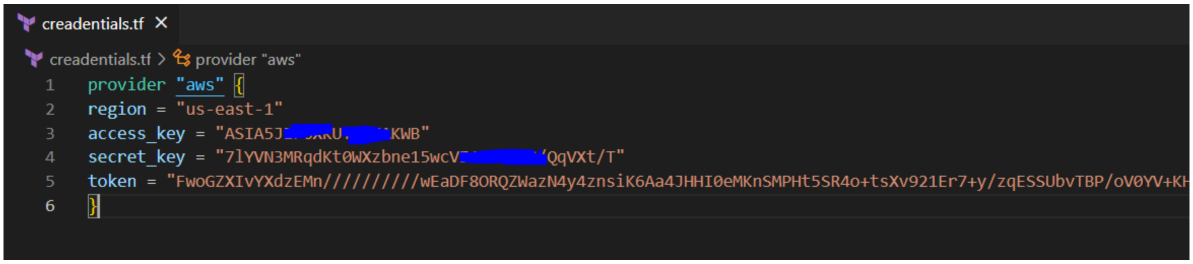
Project1test.pem

After the Key Pair is generated, we can proceed with the Instance deployment.

Step 3: Below is the terraform script which we can use for the instance deployment.

credentials.tf

This will authenticate for your web console to deploy the resources.



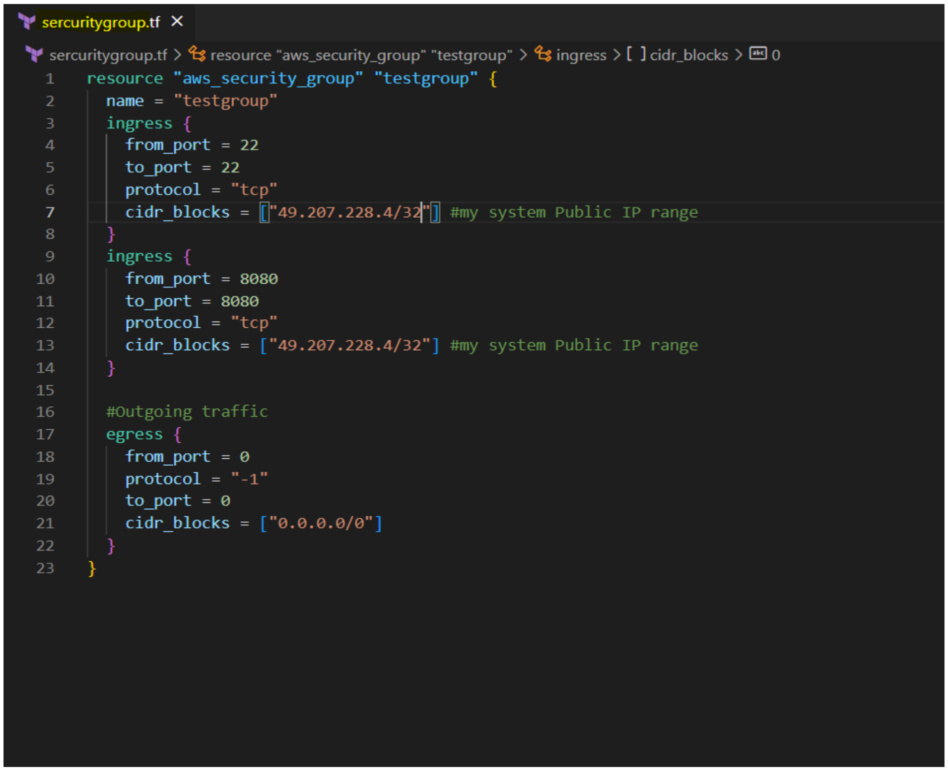
```
provider "aws" {
region = "us-east-1"
access_key = "ASIA5J[redacted]KWB"
secret_key = "7lYVN3MRqdKt0WXzbne15wcV[redacted]/QqVXt/T"
token = "FwoGZXIvYXdzEMn//////////wEaDF8ORQZWazN4y4znsiK6Aa4JHHI0eMKnSMPHt5SR4o+tsXv921Er7+y/zqESSUbvTBP/oV0YV+KH
}
```

creadentials.tf

securitygroup.tf

This is used to deploy a security group, which allows only required traffic to connect to your instance.

Ingress point are for the **inbound rules** which let traffic from internet to the server and we have allowed **port 22 with certain IP sets which will allow connection securely** and also we have allowed **port 8080 for the Jenkins connections** once the deployment is completed and **egress** is for **outbound rule** which allow traffic from server to the internet and the traffic is allowed to go out from the server.

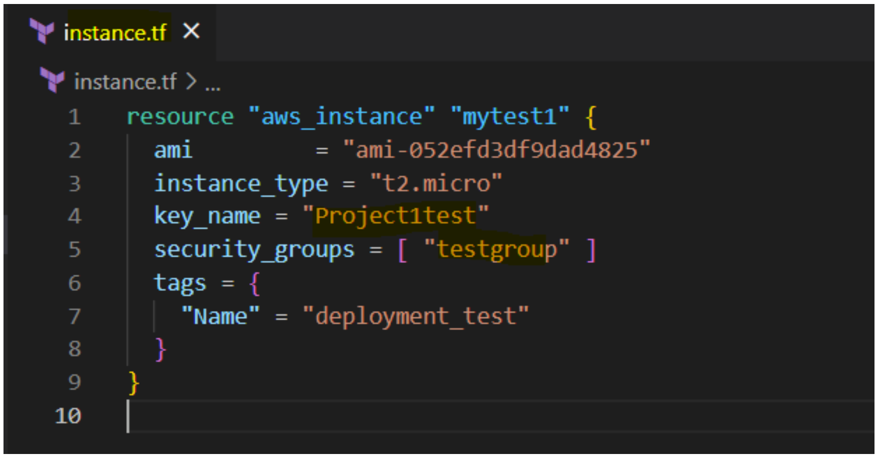

sercuritygroup.tf ×
sercuritygroup.tf > resource "aws_security_group" "testgroup" > ingress > [ ] cidr_blocks > 0

```
resource "aws_security_group" "testgroup" {
  name = "testgroup"
  ingress {
    from_port = 22
    to_port = 22
    protocol = "tcp"
    cidr_blocks = ["49.207.228.4/32"] #my system Public IP range
  }
  ingress {
    from_port = 8080
    to_port = 8080
    protocol = "tcp"
    cidr_blocks = ["49.207.228.4/32"] #my system Public IP range
  }

  #Outgoing traffic
  egress {
    from_port = 0
    protocol = "-1"
    to_port = 0
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

sercuritygroup.tf

instance.tf

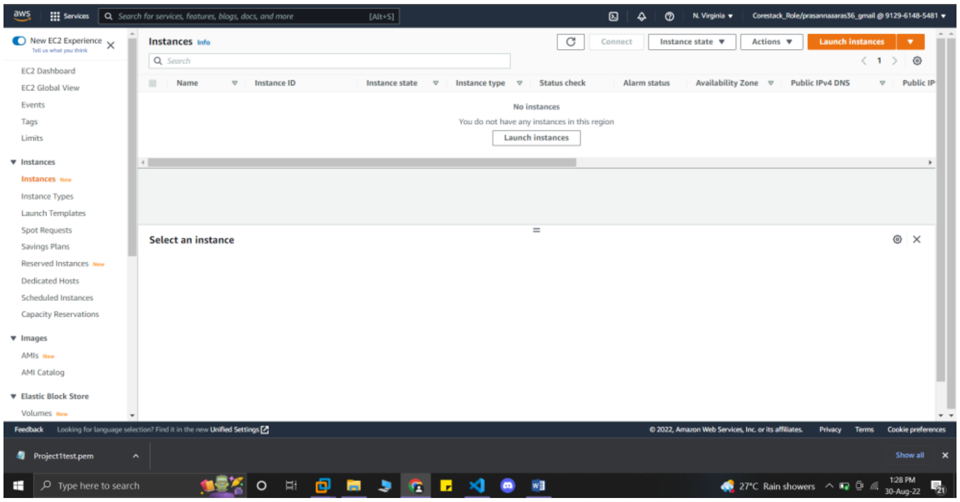
Terraform code which will deploy the intances.



```
resource "aws_instance" "mytest1" {
  ami           = "ami-052efd3df9dad4825"
  instance_type = "t2.micro"
  key_name = "Project1test"
  security_groups = [ "testgroup" ]
  tags = {
    "Name" = "deployment_test"
  }
}

```

instance.tf

No Instance:



Instances Info

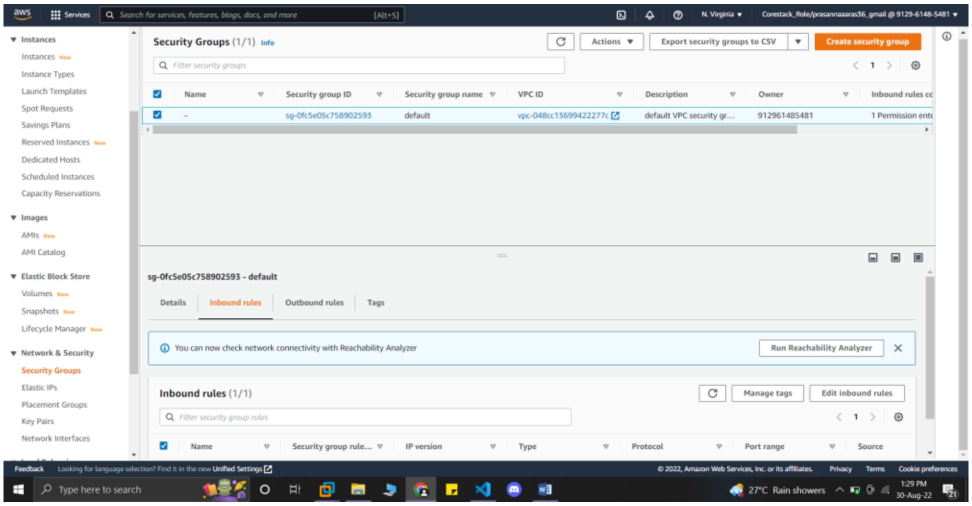
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP

No instances

You do not have any instances in this region

Launch instances

Select an instance



Security Groups (1/1) Info

☑	Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules co
☑	–	sg-0fc5e05c758902593	default	vpc-048cc13699422277c	default VPC security gr...	912961485481	1 Permission entr

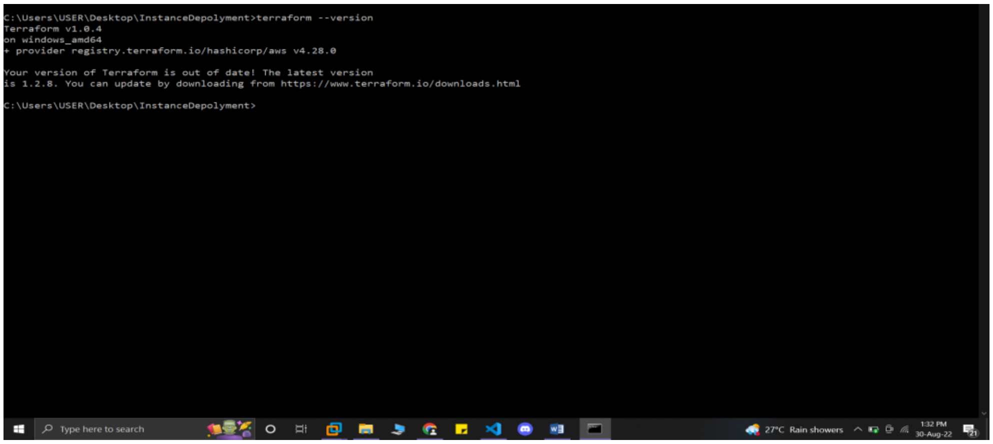
sg-0fc5e05c758902593 - default

You can now check network connectivity with Reachability Analyzer

Inbound rules (1/1)

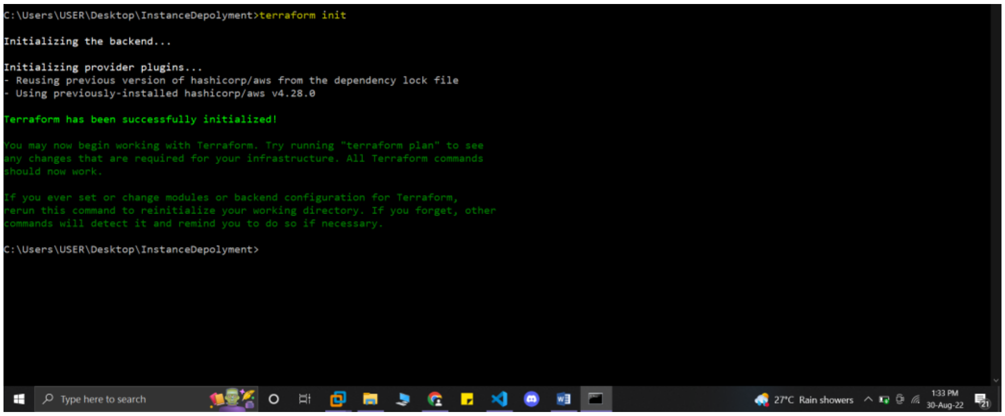
☑	Name	Security group rule...	IP version	Type	Protocol	Port range	Source

Once the Script is ready we can run the terraform command to deploy the resources.



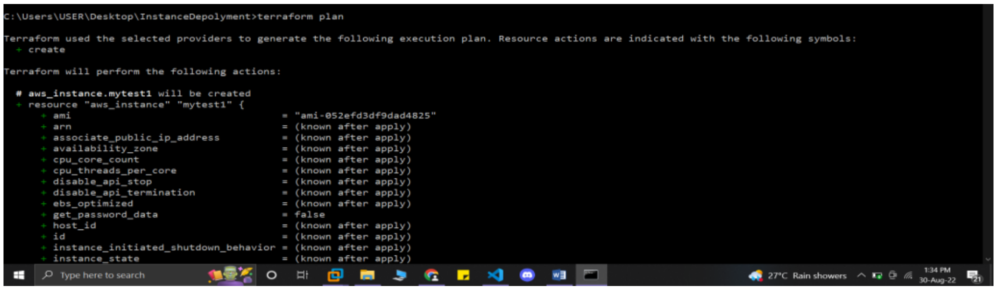
```
C:\Users\USER\Desktop\InstanceDepolyment>terraform --version
Terraform v1.0.4
on windows_amd64
+ provider registry.terraform.io/hashicorp/aws v4.28.0

Your version of Terraform is out of date! The latest version
is 1.2.8. You can update by downloading from https://www.terraform.io/downloads.html

C:\Users\USER\Desktop\InstanceDepolyment>
```

Navigate the folder where all the script are stored and run **terraform init**

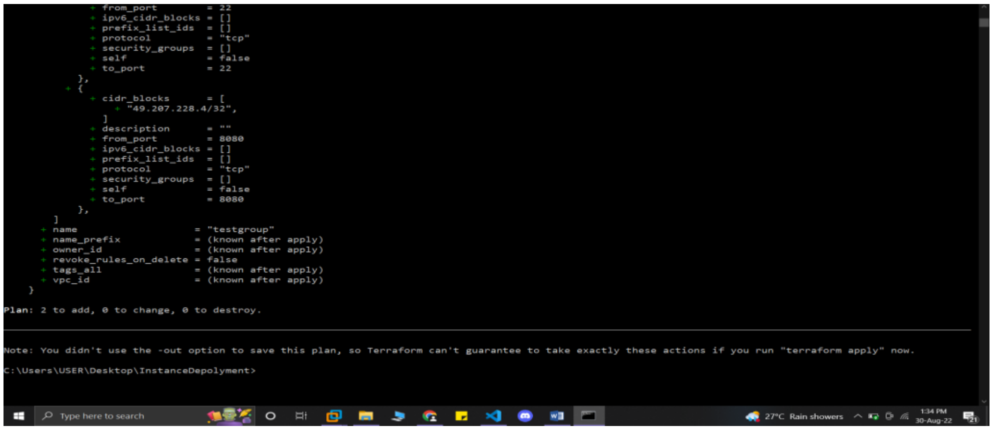


```
C:\Users\USER\Desktop\InstanceDepolyment>terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.28.0

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.

C:\Users\USER\Desktop\InstanceDepolyment>
```

Now once the terraform init is successful, run **terraform plan**

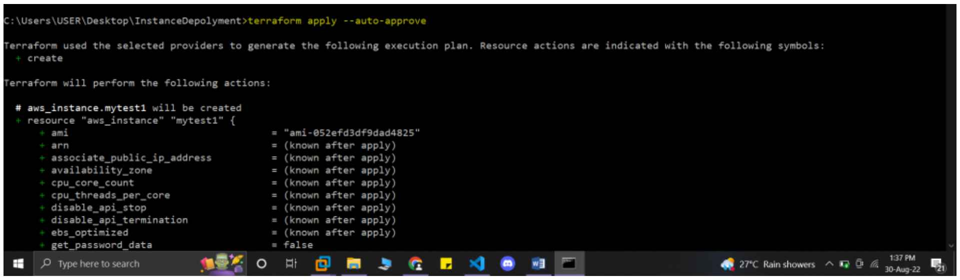


```
C:\Users\USER\Desktop\InstanceDepolyment>terraform plan

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.mytest1 will be created
  + resource "aws_instance" "mytest1" {
      + ami                                  = "ami-052efd3df9dad4825"
      + 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)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
```

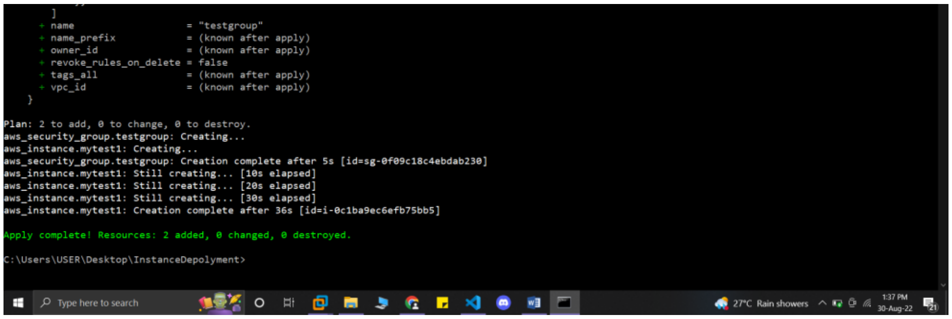


```
              + from_port        = 22
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 22
            },
          + {
              + cidr_blocks      = [
                  + "49.207.228.4/32",
                ]
              + description      = ""
              + from_port        = 8080
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 8080
            },
        ]
      + name                   = "testgroup"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + revoke_rules_on_delete = false
      + tags_all               = (known after apply)
      + vpc_id                 = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.

─────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

C:\Users\USER\Desktop\InstanceDepolyment>
```

After the terraform plan got successfully execute run, **terraform apply** to deploy the resources.

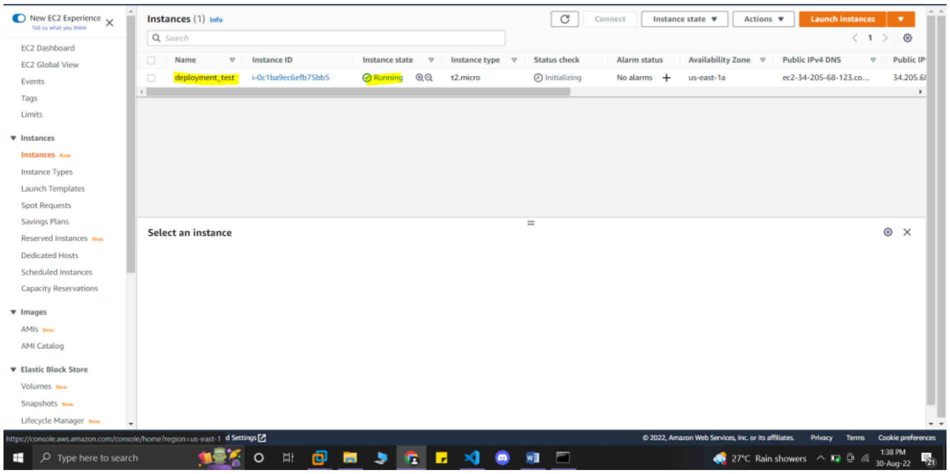


```
C:\Users\USER\Desktop\InstanceDepolyment>terraform apply --auto-approve

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.mytest1 will be created
  + resource "aws_instance" "mytest1" {
      + ami                                  = "ami-052efd3df9dad4825"
      + 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)
      + get_password_data                    = false
```



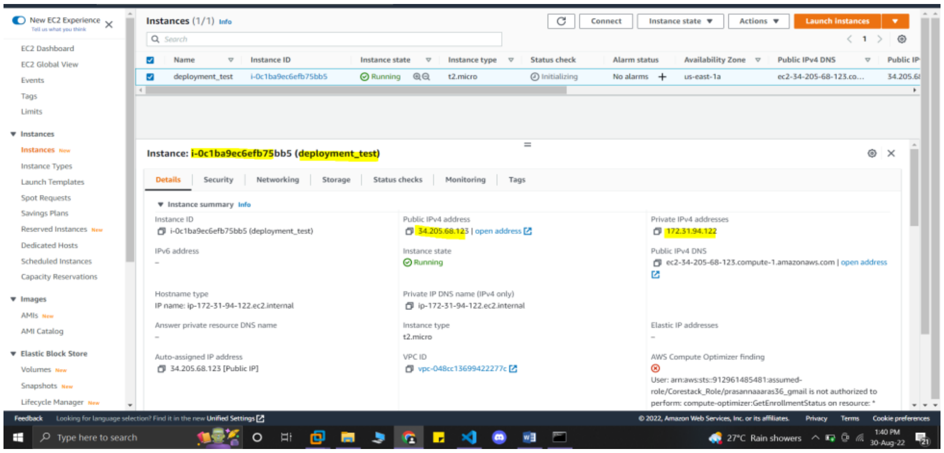
```
        ]
      + name                   = "testgroup"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + revoke_rules_on_delete = false
      + tags_all               = (known after apply)
      + vpc_id                 = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.
aws_security_group.testgroup: Creating...
aws_instance.mytest1: Creating...
aws_security_group.testgroup: Creation complete after 5s [id=sg-0f09c18c4ebdab230]
aws_instance.mytest1: Still creating... [10s elapsed]
aws_instance.mytest1: Still creating... [20s elapsed]
aws_instance.mytest1: Still creating... [30s elapsed]
aws_instance.mytest1: Creation complete after 36s [id=i-0c1ba9ec6efb75bb5]

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

C:\Users\USER\Desktop\InstanceDepolyment>
```

Now Login to your AWS web console to verify the deployed resources



Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
deployment_test	i-0c1ba9ec6efb75bb5	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-34-205-68-123.co...	34.205.6



Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
☑ deployment_test	i-0c1ba9ec6efb75bb5	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-34-205-68-123.co...	34.205.6

Instance: i-0c1ba9ec6efb75bb5 (deployment_test)

Instance summary Info

Instance ID
i-0c1ba9ec6efb75bb5 (deployment_test)

Public IPv4 address
34.205.68.123 | open address

Private IPv4 addresses
172.31.94.122

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-34-205-68-123.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-94-122.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-94-122.ec2.internal

Answer private resource DNS name
–

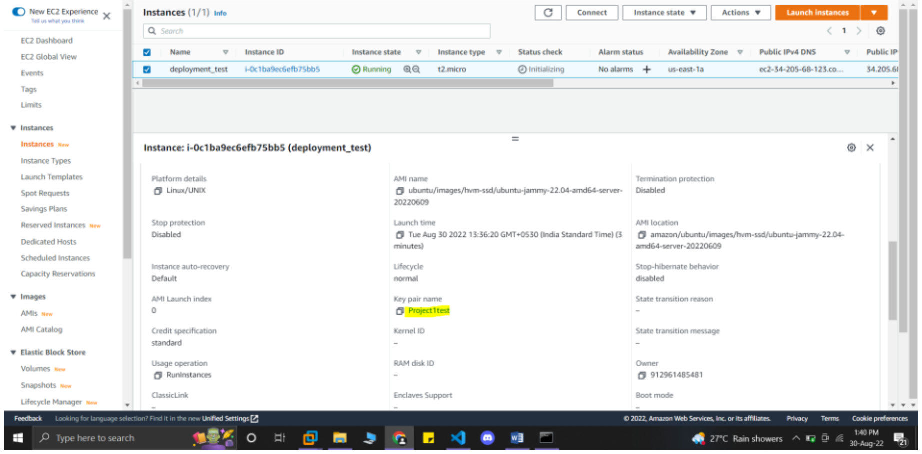
Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
34.205.68.123 [Public IP]

VPC ID
vpc-048cc13699422277c

AWS Compute Optimizer finding
User: arn:aws:sts::912961485481:assumed-role/Corestack_Role/prasannaaras36_gmail is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: *



Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
☑	deployment_test	i-0c1ba9ec6efb75bb5	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-34-205-68-123.co...	34.205.6

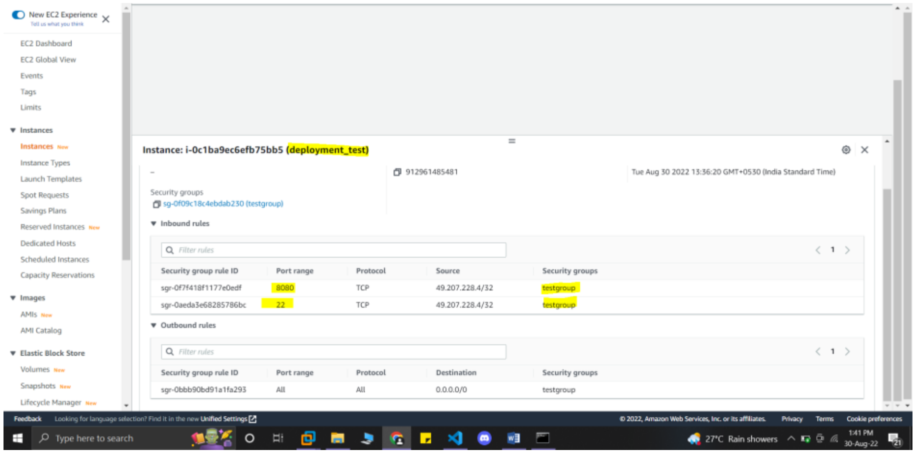
Instance: i-0c1ba9ec6efb75bb5 (deployment_test)

Platform details Linux/UNIX	AMI name ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20220609	Termination protection Disabled
Stop protection Disabled	Launch time Tue Aug 30 2022 13:36:20 GMT+0530 (India Standard Time) (3 minutes)	AMI location amazon/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20220609
Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled
AMI Launch index 0	Key pair name Project1test	State transition reason –
Credit specification standard	Kernel ID –	State transition message –
Usage operation RunInstances	RAM disk ID –	Owner 912961485481
ClassicLink	Enclaves Support	Boot mode

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

27°C Rain showers 1:40 PM 30-Aug-22



Instance: i-0c1ba9ec6efb75bb5 (deployment_test)

– 912961485481 Tue Aug 30 2022 13:36:20 GMT+0530 (India Standard Time)

Security groups
sg-0f09c18c4ebdab230 (testgroup)

Inbound rules

Security group rule ID	Port range	Protocol	Source	Security groups
sgr-0f7f418f1177e0edf	8080	TCP	49.207.228.4/32	testgroup
sgr-0aeda3e68285786bc	22	TCP	49.207.228.4/32	testgroup

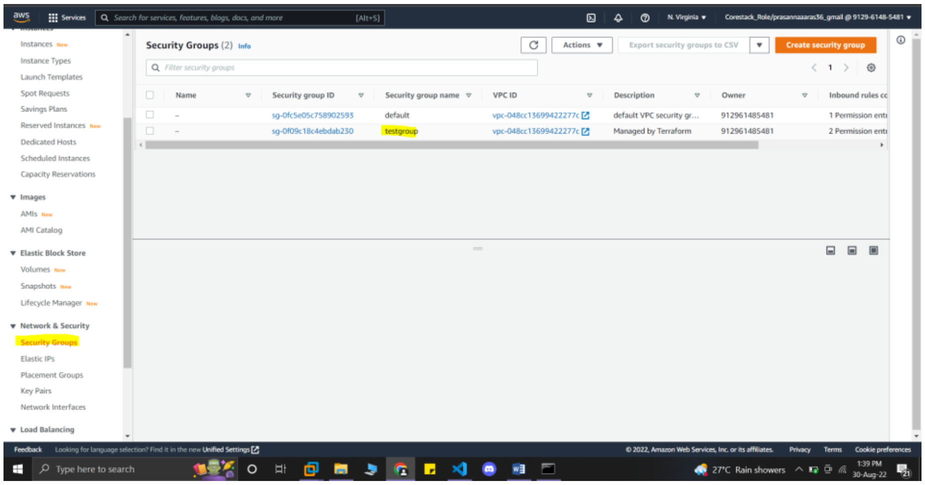
Outbound rules

Security group rule ID	Port range	Protocol	Destination	Security groups
sgr-0bbb90bd91a1fa293	All	All	0.0.0.0/0	testgroup

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

27°C Rain showers 1:41 PM 30-Aug-22



Security Groups (2) Info

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules co
–	sg-0fc5e05c758902593	default	vpc-048cc13699422277c	default VPC security gr...	912961485481	1 Permission ent
–	sg-0f09c18c4ebdab230	testgroup	vpc-048cc13699422277c	Managed by Terraform	912961485481	2 Permission ent

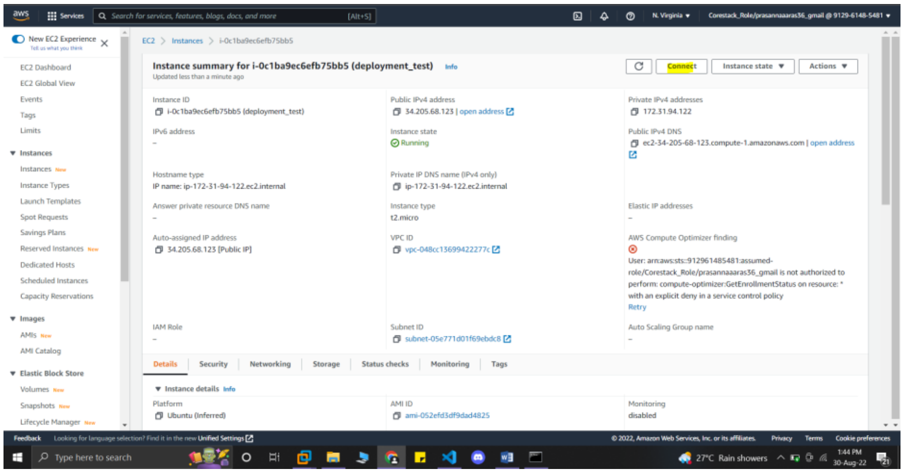
Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

27°C Rain showers 1:39 PM 30-Aug-22

Connecting to the Instance:

Go to instance and click on Connect.



EC2 > Instances > i-0c1ba9ec6efb75bb5

Instance summary for i-0c1ba9ec6efb75bb5 (deployment_test) Info

Updated less than a minute ago

Instance ID
i-0c1ba9ec6efb75bb5 (deployment_test)

Public IPv4 address
34.205.68.123 | open address

Private IPv4 addresses
172.31.94.122

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-34-205-68-123.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-94-122.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-94-122.ec2.internal

Answer private resource DNS name
–

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
34.205.68.123 [Public IP]

VPC ID
vpc-048cc13699422277c

AWS Compute Optimizer finding
User: arn:aws:sts::912961485481:assumed-role/Corestack_Role/prasannaaaras36_gmail is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * with an explicit deny in a service control policy
Retry

IAM Role
–

Subnet ID
subnet-05e771d01f69ebdc8

Auto Scaling Group name
–

Instance details Info

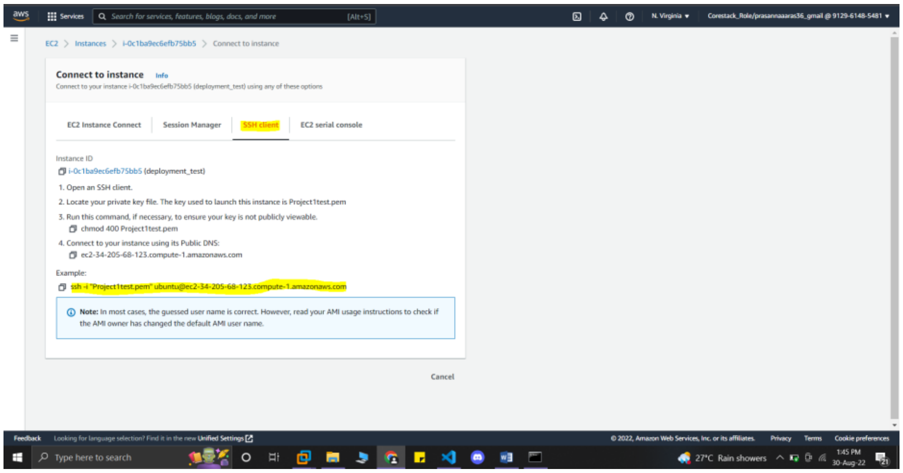
Platform
Ubuntu (Inferred)

AMI ID
ami-052efd3df9dad4825

Monitoring
disabled

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

27°C Rain showers 1:44 PM 30-Aug-22



EC2 > Instances > i-0c1ba9ec6efb75bb5 > Connect to instance

Connect to instance Info

Connect to your instance i-0c1ba9ec6efb75bb5 (deployment_test) using any of these options

Instance ID
i-0c1ba9ec6efb75bb5 (deployment_test)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Project1test.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 Project1test.pem
4. Connect to your instance using its Public DNS:
   ec2-34-205-68-123.compute-1.amazonaws.com

Example:

```
ssh -i "Project1test.pem" ubuntu@ec2-34-205-68-123.compute-1.amazonaws.com
```

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

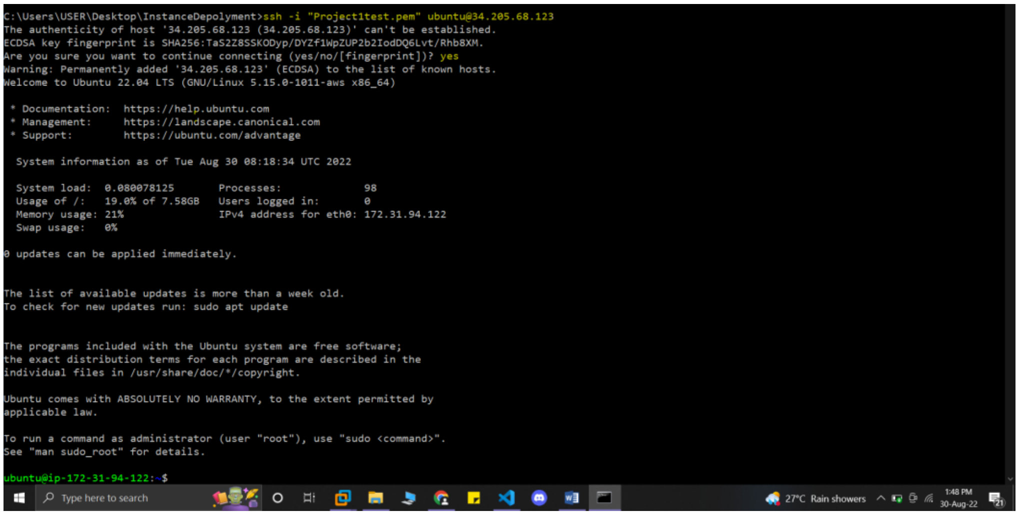
© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

27°C Rain showers 1:45 PM 30-Aug-22

We can either use the above option to connect or, We can use

ssh –i “Project1test.pem” ubuntu@34.205.68.123

NOTE: Ubuntu is the default username for the login and 34.205.68.123 is the public IP of the server.



```
C:\Users\USER\Desktop\InstanceDepolyment>ssh -i "Project1test.pem" ubuntu@34.205.68.123
The authenticity of host '34.205.68.123 (34.205.68.123)' can't be established.
ECDSA key fingerprint is SHA256:TaS2Z8SSKODyp/DYZf1WpZUP2b2IodDQ6Lvt/Rhb8XM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.205.68.123' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1011-aws x86_64)

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

  System information as of Tue Aug 30 08:18:34 UTC 2022

  System load:  0.080078125       Processes:             98
  Usage of /:   19.0% of 7.58GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for eth0: 172.31.94.122
  Swap usage:   0%

0 updates can be applied immediately.


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-172-31-94-122:~$
```

27°C Rain showers 1:48 PM 30-Aug-22

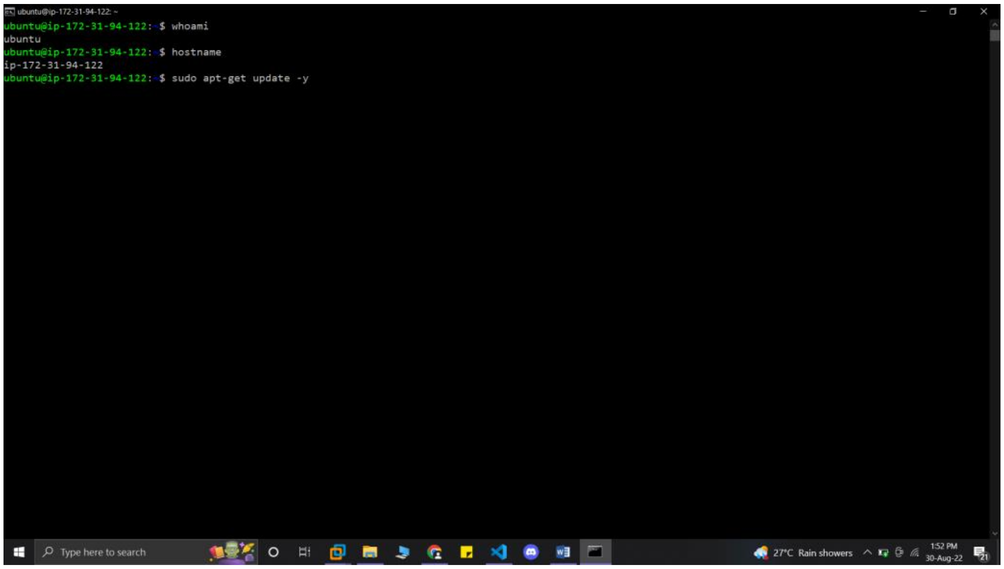
We are successfully logged into the server.

Now we need to install Jenkins, Java and Python.

Jenkins Installation:

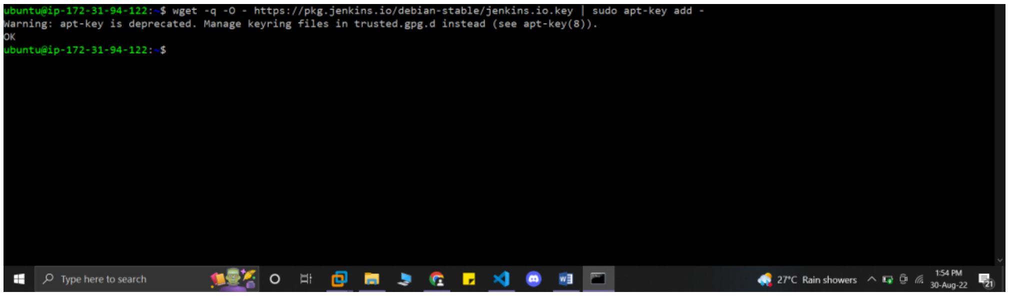
Run the below commands:

sudo apt-get update -y



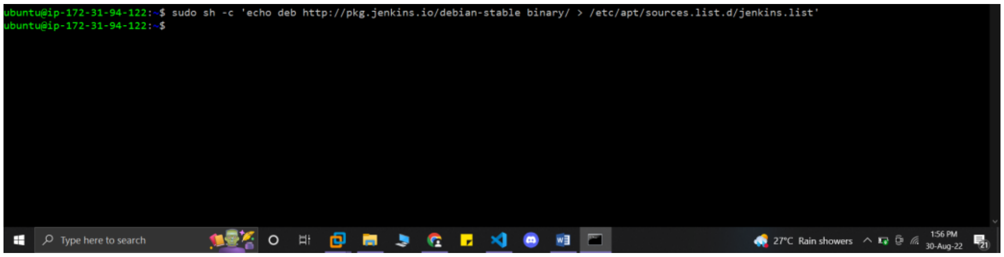
```
ubuntu@ip-172-31-94-122:~$ whoami
ubuntu
ubuntu@ip-172-31-94-122:~$ hostname
ip-172-31-94-122
ubuntu@ip-172-31-94-122:~$ sudo apt-get update -y
```

wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -



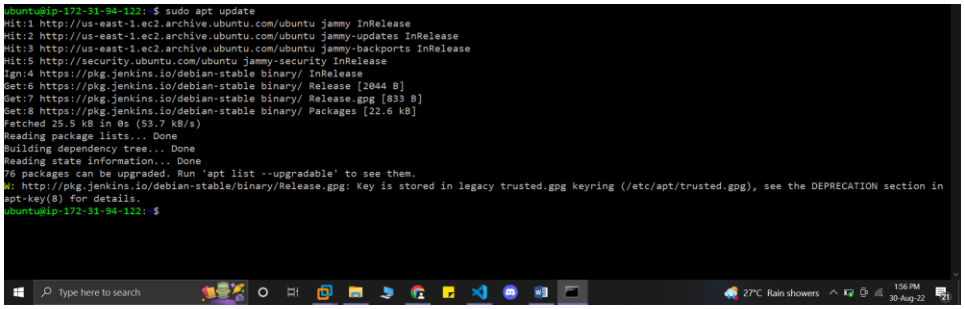
```
ubuntu@ip-172-31-94-122:~$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
ubuntu@ip-172-31-94-122:~$
```

sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'



```
ubuntu@ip-172-31-94-122:~$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
ubuntu@ip-172-31-94-122:~$
```

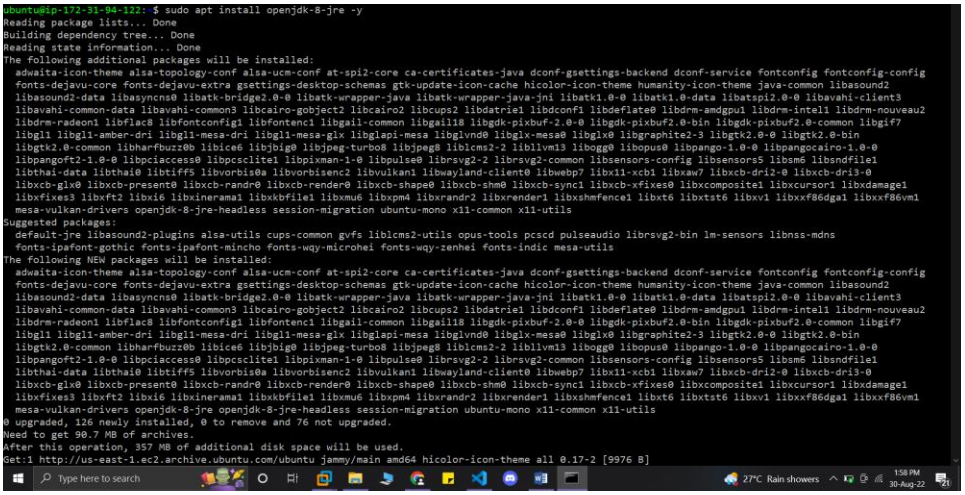
sudo apt update



```
ubuntu@ip-172-31-94-122:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:7 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [22.6 kB]
Fetched 25.5 kB in 0s (53.7 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
76 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: http://pkg.jenkins.io/debian-stable/binary/Release.gpg: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in
apt-key(8) for details.
ubuntu@ip-172-31-94-122:~$
```

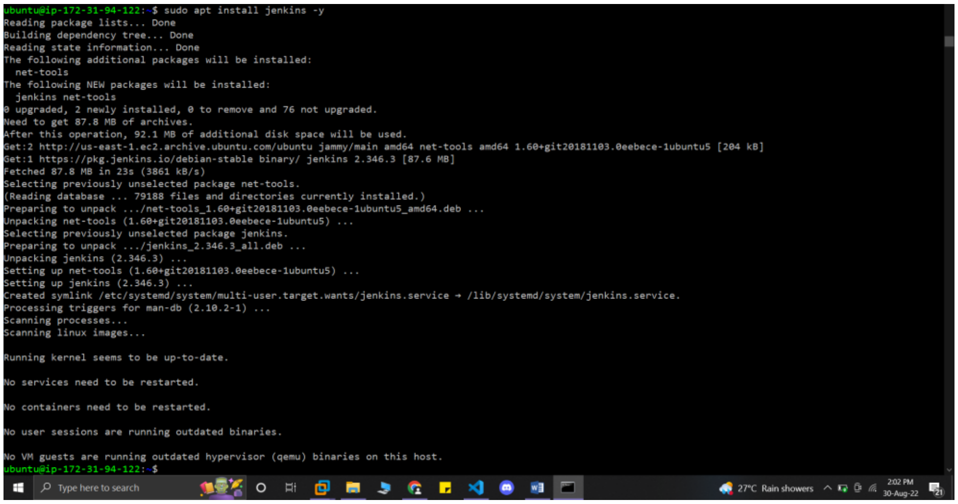
sudo apt install openjdk-8-jre –y

Note: This installs Java and is also a dependencies for Jenkins:



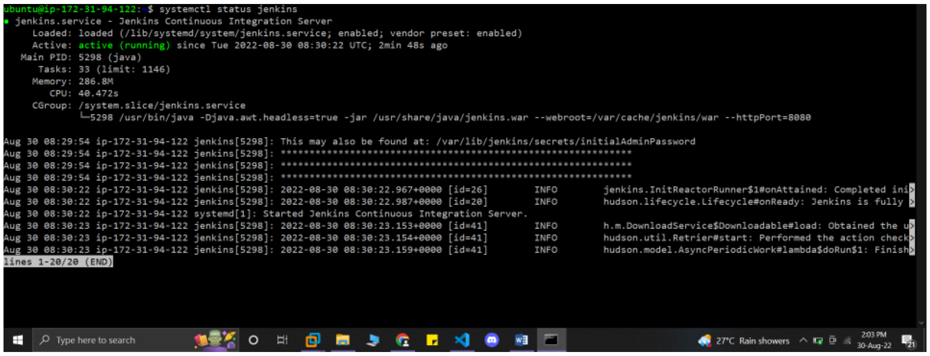
```
ubuntu@ip-172-31-94-122:~$ sudo apt install openjdk-8-jre -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig fontconfig-config
  fonts-dejavu-core fonts-dejavu-extra gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme java-common libasound2
  libasound2-data libasyncns0 libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3
  libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcups2 libdatrie1 libdconf1 libdeflate0 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2
  libdrm-radeon1 libflac8 libfontconfig1 libfontenc1 libgail-common libgail18 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgif7
  libgl1 libgl1-amber-dri libgl1-mesa-dri libgl1-mesa-glx libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libgtk2.0-0 libgtk2.0-bin
  libgtk2.0-common libharfbuzz0b libice6 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 libllvm13 libogg0 libopus0 libpango-1.0-0 libpangocairo-1.0-0
  libpangoft2-1.0-0 libpciaccess0 libpcsclite1 libpixman-1-0 libpulse0 librsvg2-2 librsvg2-common libsensors-config libsensors5 libsm6 libsndfile1
  libthai-data libthai0 libtiff5 libvorbis0a libvorbisenc2 libvulkan1 libwayland-client0 libwebp7 libx11-xcb1 libxaw7 libxcb-dri2-0 libxcb-dri3-0
  libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcomposite1 libxcursor1 libxdamage1
  libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt6 libxtst6 libxv1 libxxf86dga1 libxxf86vm1
  mesa-vulkan-drivers openjdk-8-jre-headless session-migration ubuntu-mono x11-common x11-utils
Suggested packages:
  default-jre libasound2-plugins alsa-utils cups-common gvfs liblcms2-utils opus-tools pcscd pulseaudio librsvg2-bin lm-sensors libnss-mdns
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic mesa-utils
The following NEW packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig fontconfig-config
  fonts-dejavu-core fonts-dejavu-extra gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme java-common libasound2
  libasound2-data libasyncns0 libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3
  libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcups2 libdatrie1 libdconf1 libdeflate0 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2
  libdrm-radeon1 libflac8 libfontconfig1 libfontenc1 libgail-common libgail18 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgif7
  libgl1 libgl1-amber-dri libgl1-mesa-dri libgl1-mesa-glx libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libgtk2.0-0 libgtk2.0-bin
  libgtk2.0-common libharfbuzz0b libice6 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 libllvm13 libogg0 libopus0 libpango-1.0-0 libpangocairo-1.0-0
  libpangoft2-1.0-0 libpciaccess0 libpcsclite1 libpixman-1-0 libpulse0 librsvg2-2 librsvg2-common libsensors-config libsensors5 libsm6 libsndfile1
  libthai-data libthai0 libtiff5 libvorbis0a libvorbisenc2 libvulkan1 libwayland-client0 libwebp7 libx11-xcb1 libxaw7 libxcb-dri2-0 libxcb-dri3-0
  libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcomposite1 libxcursor1 libxdamage1
  libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt6 libxtst6 libxv1 libxxf86dga1 libxxf86vm1
  mesa-vulkan-drivers openjdk-8-jre openjdk-8-jre-headless session-migration ubuntu-mono x11-common x11-utils
0 upgraded, 126 newly installed, 0 to remove and 76 not upgraded.
Need to get 90.7 MB of archives.
After this operation, 357 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 hicolor-icon-theme all 0.17-2 [9976 B]
```

sudo apt install jenkins –y #### Install Jenkins



```
ubuntu@ip-172-31-94-122:~$ sudo apt install jenkins -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 76 not upgraded.
Need to get 87.8 MB of archives.
After this operation, 92.1 MB of additional disk space will be used.
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.346.3 [87.6 MB]
Fetched 87.8 MB in 23s (3861 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 79188 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.346.3_all.deb ...
Unpacking jenkins (2.346.3) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.346.3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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@ip-172-31-94-122:~$
```

Jenkins service is running:

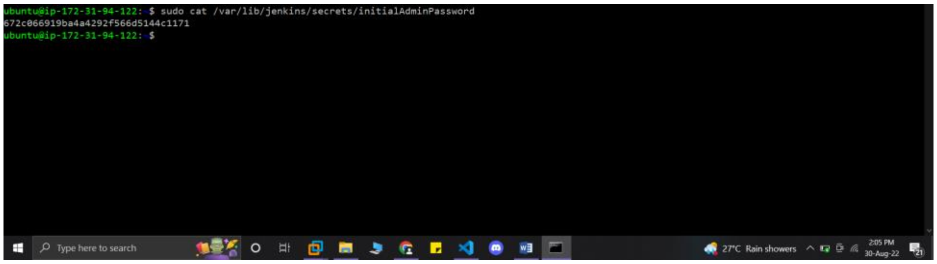


```
ubuntu@ip-172-31-94-122:~$ systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2022-08-30 08:30:22 UTC; 2min 48s ago
   Main PID: 5298 (java)
      Tasks: 33 (limit: 1146)
     Memory: 286.8M
        CPU: 40.472s
     CGroup: /system.slice/jenkins.service
             └─5298 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Aug 30 08:29:54 ip-172-31-94-122 jenkins[5298]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Aug 30 08:29:54 ip-172-31-94-122 jenkins[5298]: *************************************************************
Aug 30 08:29:54 ip-172-31-94-122 jenkins[5298]: *************************************************************
Aug 30 08:29:54 ip-172-31-94-122 jenkins[5298]: *************************************************************
Aug 30 08:30:22 ip-172-31-94-122 jenkins[5298]: 2022-08-30 08:30:22.967+0000 [id=26]        INFO        jenkins.InitReactorRunner$1#onAttained: Completed ini>
Aug 30 08:30:22 ip-172-31-94-122 jenkins[5298]: 2022-08-30 08:30:22.987+0000 [id=20]        INFO        hudson.lifecycle.Lifecycle#onReady: Jenkins is fully >
Aug 30 08:30:22 ip-172-31-94-122 systemd[1]: Started Jenkins Continuous Integration Server.
Aug 30 08:30:23 ip-172-31-94-122 jenkins[5298]: 2022-08-30 08:30:23.153+0000 [id=41]        INFO        h.m.DownloadService$Downloadable#load: Obtained the u>
Aug 30 08:30:23 ip-172-31-94-122 jenkins[5298]: 2022-08-30 08:30:23.154+0000 [id=41]        INFO        hudson.util.Retrier#start: Performed the action check>
Aug 30 08:30:23 ip-172-31-94-122 jenkins[5298]: 2022-08-30 08:30:23.159+0000 [id=41]        INFO        hudson.model.AsyncPeriodicWork#lambda$doRun$1: Finish>
lines 1-20/20 (END)
```

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

672c066919ba4a4292f566d5144c1171

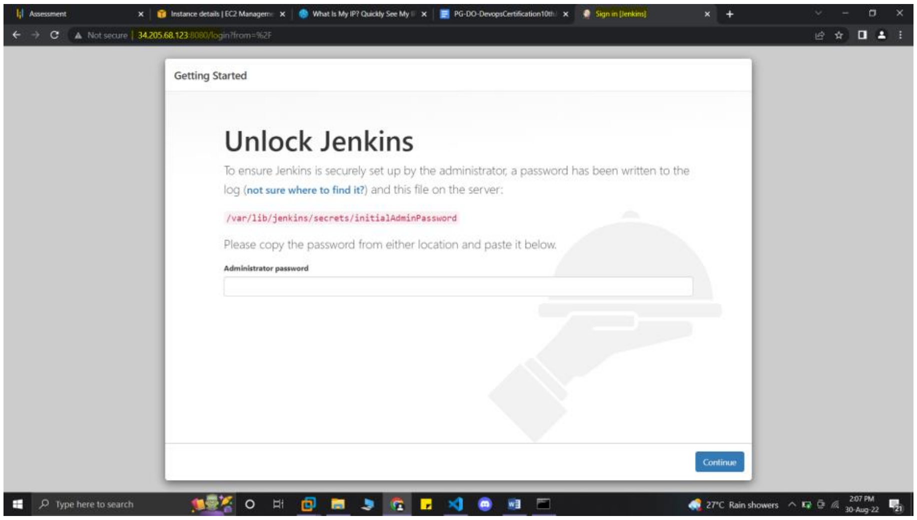
#### This command will provide the initial admin password to setup up Jenkins.



```
ubuntu@ip-172-31-94-122:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
672c066919ba4a4292f566d5144c1171
ubuntu@ip-172-31-94-122:~$
```

Open Browser and browser the public IP of the Instance with port 8080

[**http://**34.205.68.123:8080](http://34.205.68.123:8080)



Assessment
Instance details | EC2 Managem
What Is My IP? Quickly See My I
PG-DO-DevopsCertification10th
Sign in [Jenkins]
Not secure | 34.205.68.123:8080/login?from=%2F

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

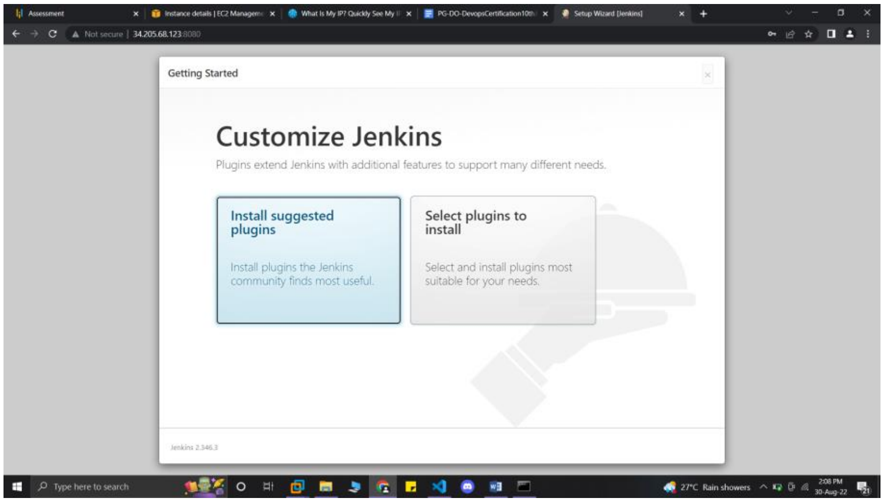
`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

Continue
Type here to search
27°C Rain showers
2:07 PM
30-Aug-22

Enter the initial admin password which we copied previously and install the Jenkins



Assessment
Instance details | EC2 Manageme
What Is My IP? Quickly See My I
PG-DO-DevopsCertification10th
Setup Wizard [Jenkins]
Not secure | 34.205.68.123:8080
Type here to search
27°C Rain showers
2:08 PM
30-Aug-22

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

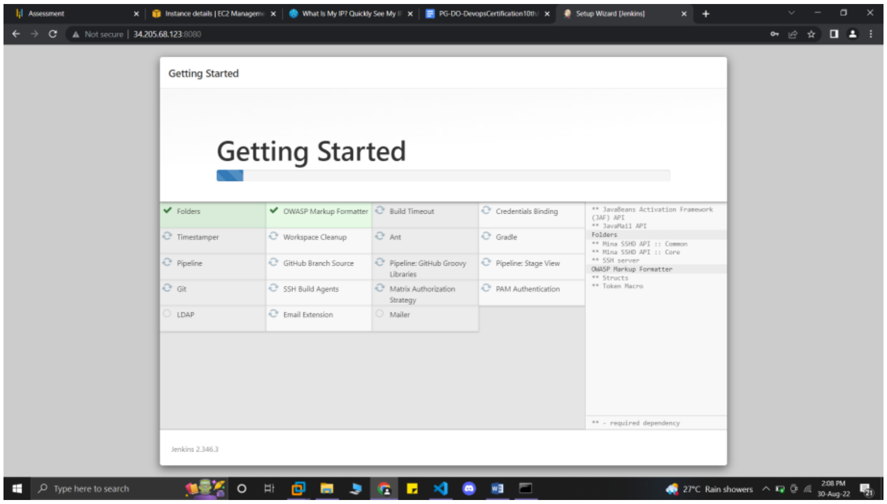
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.346.3



Assessment
Instance details | EC2 Manageme
What Is My IP? Quickly See My I
PG-DO-DevopsCertification10th
Setup Wizard [Jenkins]
Not secure | 34.205.68.123:8080
Type here to search
27°C Rain showers
2:08 PM
30-Aug-22

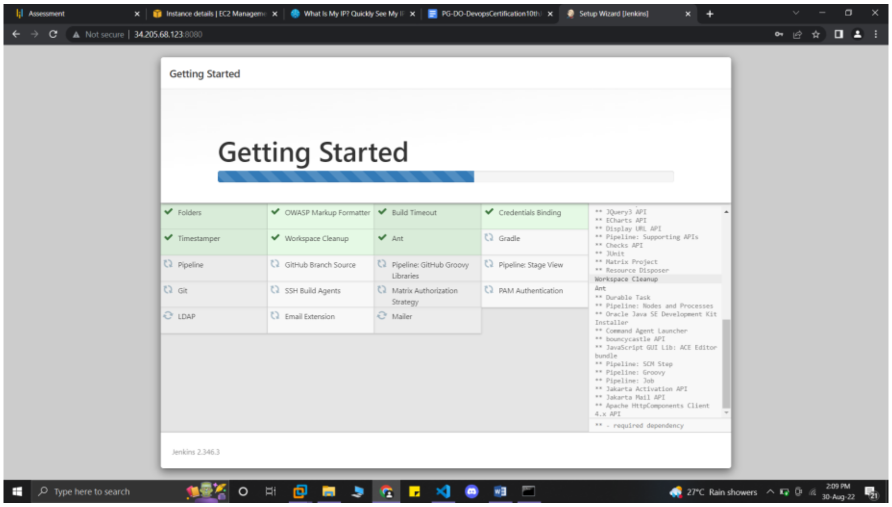
Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
○ LDAP	Email Extension	○ Mailer	

```
** JavaBeans Activation Framework
(JAF) API
** JavaMail API
Folders
** Mina SSHD API :: Common
** Mina SSHD API :: Core
** SSH server
OWASP Markup Formatter
** Structs
** Token Macro

** - required dependency
```

Jenkins 2.346.3



Assessment
Instance details | EC2 Manageme
What Is My IP? Quickly See My I
PG-DO-DevopsCertification10th
Setup Wizard [Jenkins]
Not secure | 34.205.68.123:8080
Type here to search
27°C Rain showers
2:09 PM
30-Aug-22

Getting Started

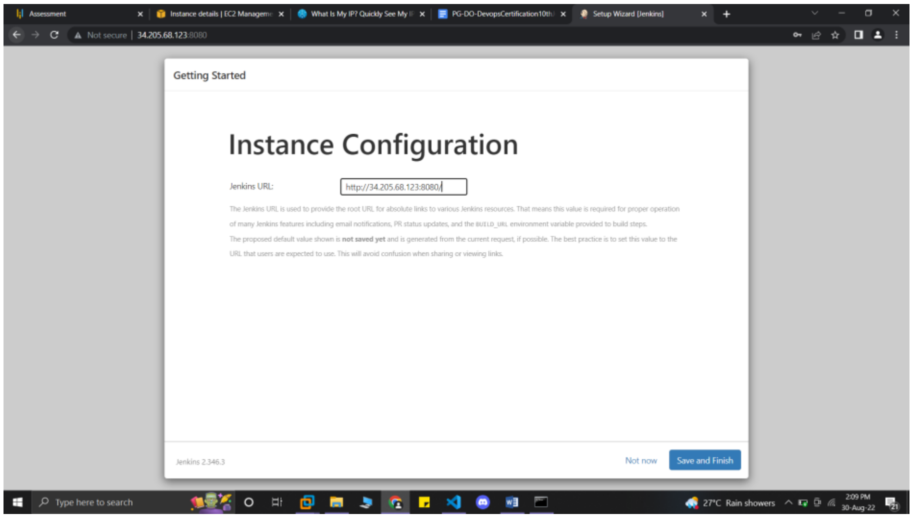
Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

```
** JQuery3 API
** ECharts API
** Display URL API
** Pipeline: Supporting APIs
** Checks API
** JUnit
** Matrix Project
** Resource Disposer
Workspace Cleanup
Ant
** Durable Task
** Pipeline: Nodes and Processes
** Oracle Java SE Development Kit
Installer
** Command Agent Launcher
** bouncycastle API
** JavaScript GUI Lib: ACE Editor
bundle
** Pipeline: SCM Step
** Pipeline: Groovy
** Pipeline: Job
** Jakarta Activation API
** Jakarta Mail API
** Apache HttpComponents Client
4.x API

** - required dependency
```

Jenkins 2.346.3

After this steps we have create a Admin account for the jenkins login and finsihed the setup.


Assessment
Instance details | EC2 Management
What Is My IP? Quickly See My IP
PG-DO-DevopsCertification10th
Setup Wizard [Jenkins]
Not secure | 34.205.68.123:8080
Type here to search
27°C Rain showers
2:09 PM
30-Aug-22

Getting Started

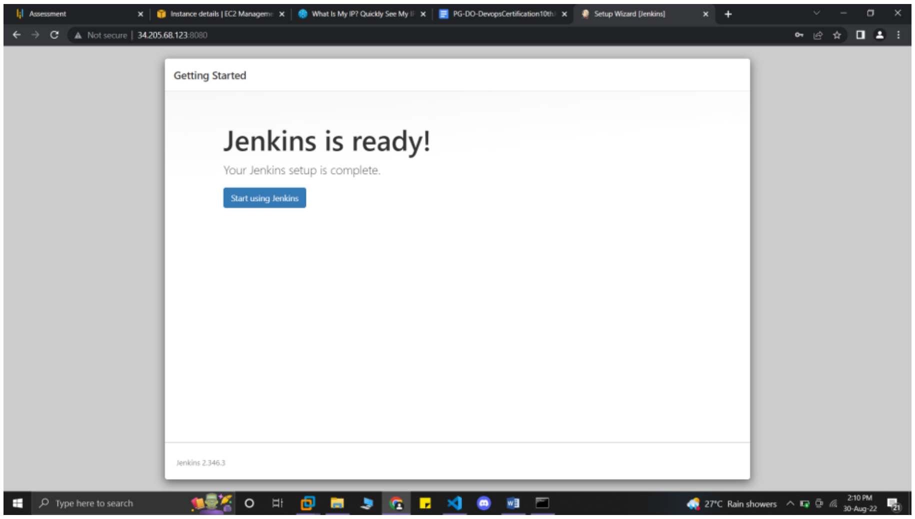
Instance Configuration

Jenkins URL: http://34.205.68.123:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.346.3

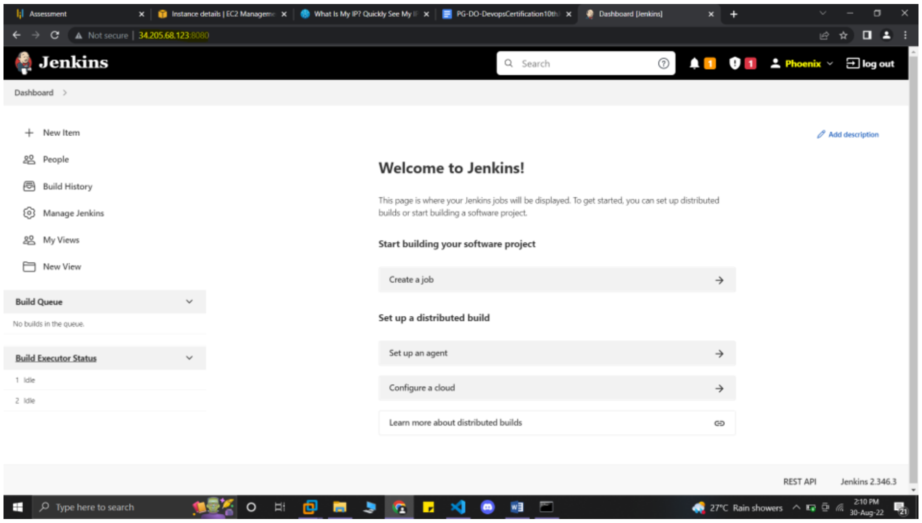

Assessment
Instance details | EC2 Management
What Is My IP? Quickly See My IP
PG-DO-DevopsCertification10th
Setup Wizard [Jenkins]
Not secure | 34.205.68.123:8080
Type here to search
27°C Rain showers
2:10 PM
30-Aug-22

Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Jenkins 2.346.3



Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

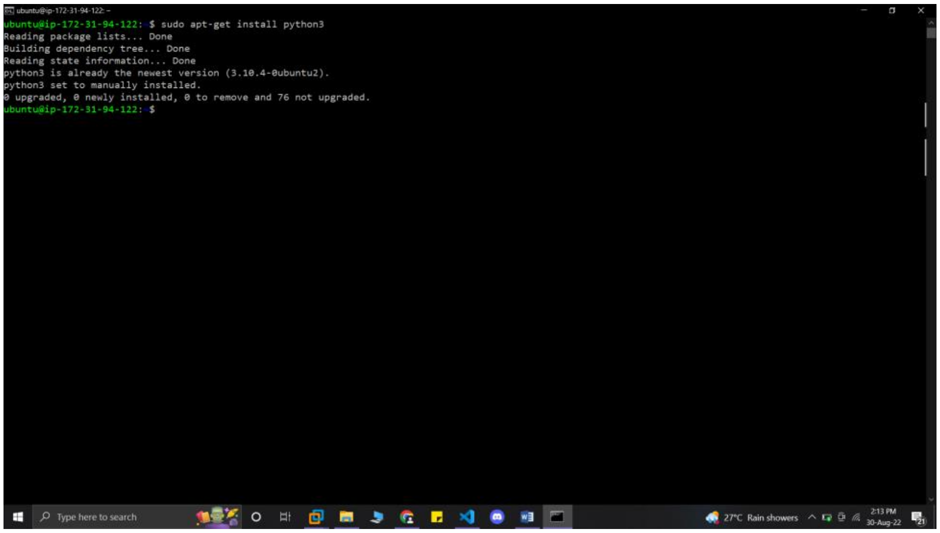
Start building your software project

Set up a distributed build

Jenkins 2.346.3

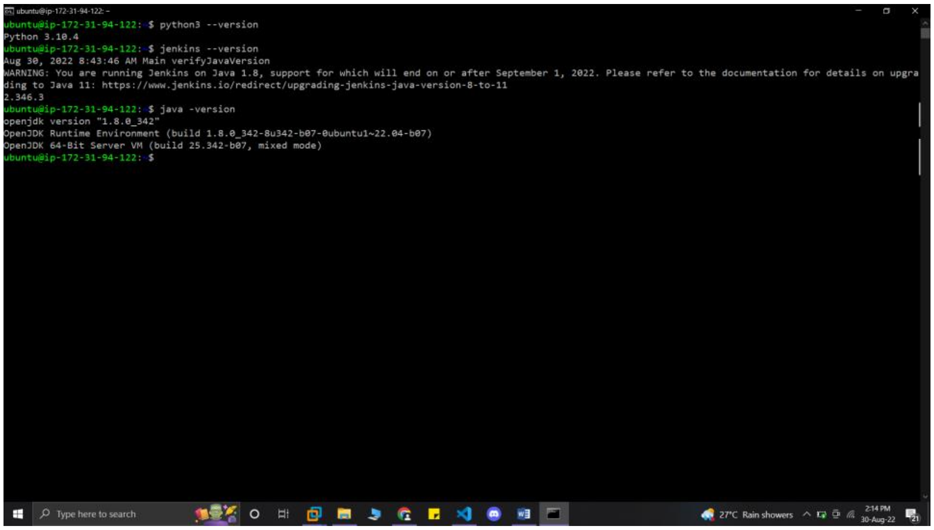
To Install Python

Run the command: sudo apt-get install python3



```
ubuntu@ip-172-31-94-122:~$ sudo apt-get install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.4-0ubuntu2).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 76 not upgraded.
ubuntu@ip-172-31-94-122:~$
```

We see all the 3 packages are installed



```
ubuntu@ip-172-31-94-122:~$ python3 --version
Python 3.10.4
ubuntu@ip-172-31-94-122:~$ jenkins --version
Aug 30, 2022 8:43:46 AM Main verifyJavaVersion
WARNING: You are running Jenkins on Java 1.8, support for which will end on or after September 1, 2022. Please refer to the documentation for details on upgra
ding to Java 11: https://www.jenkins.io/redirect/upgrading-jenkins-java-version-8-to-11
2.346.3
ubuntu@ip-172-31-94-122:~$ java -version
openjdk version "1.8.0_342"
OpenJDK Runtime Environment (build 1.8.0_342-8u342-b07-0ubuntu1~22.04-b07)
OpenJDK 64-Bit Server VM (build 25.342-b07, mixed mode)
ubuntu@ip-172-31-94-122:~$
```