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 centralised server for Jenkins.

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

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

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

Expected Deliverables:

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

Table of Contents

Introduction

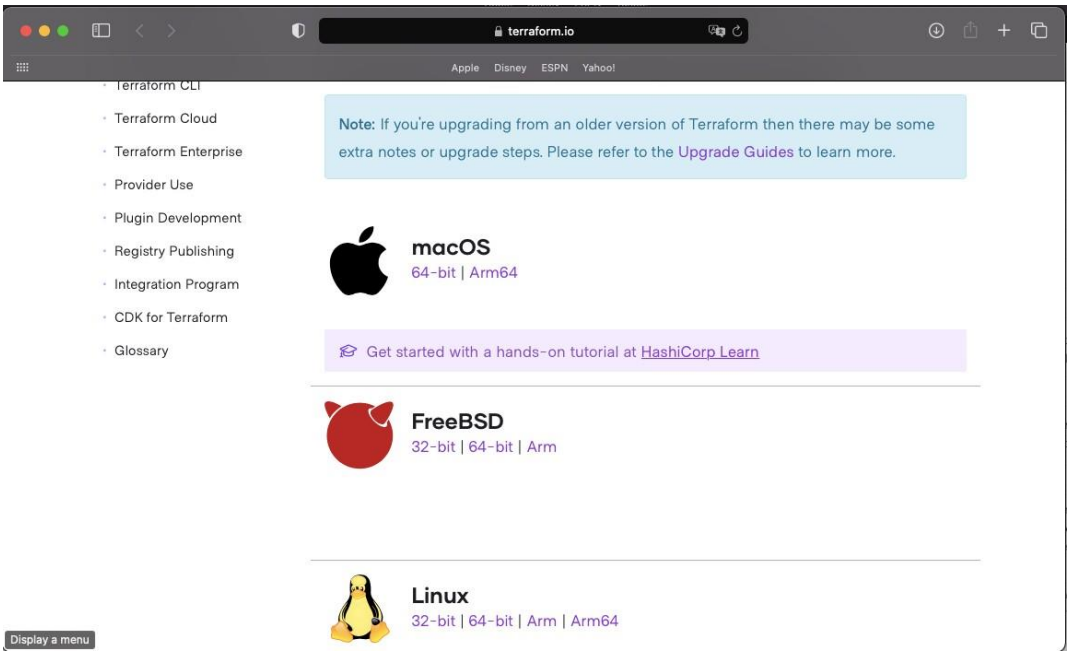
This Project has the objective to create an EC2 with terraform source-code

Installation of pre-requisites

Install terraform

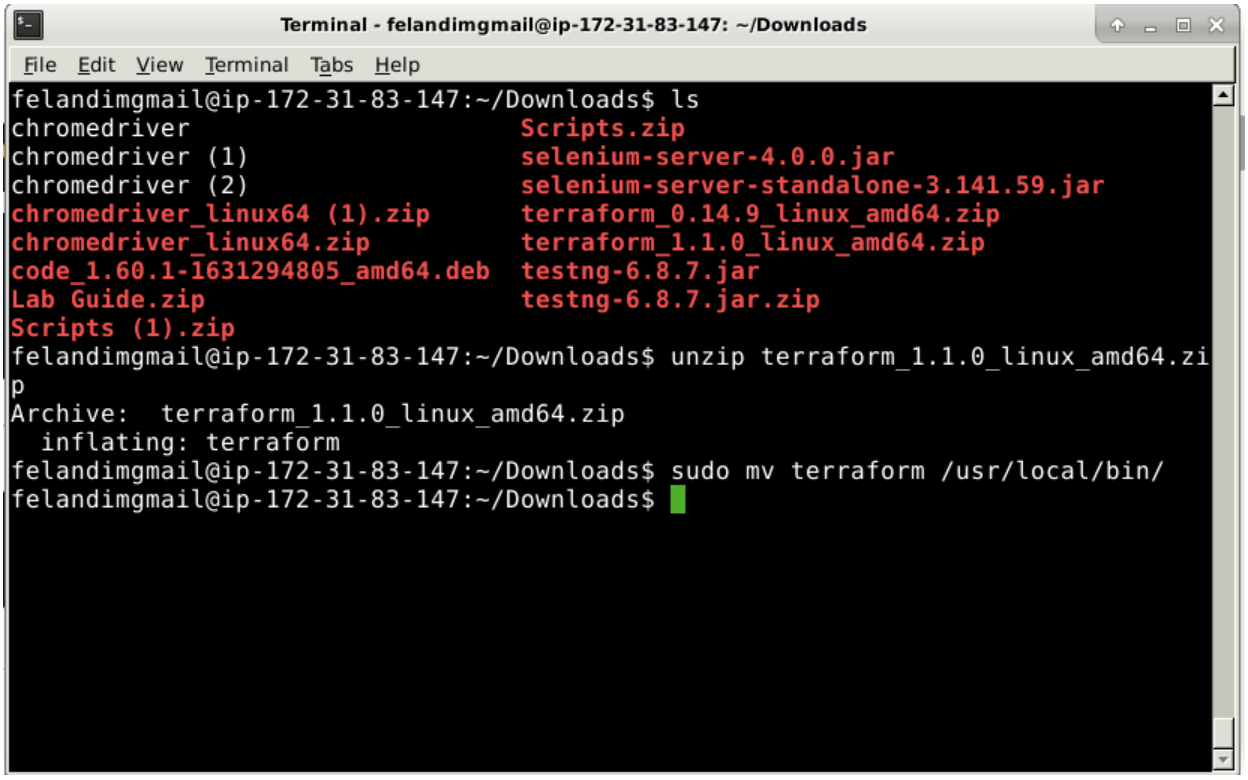
Install the Terraform package from the link

https://www.terraform.io/downloads.html



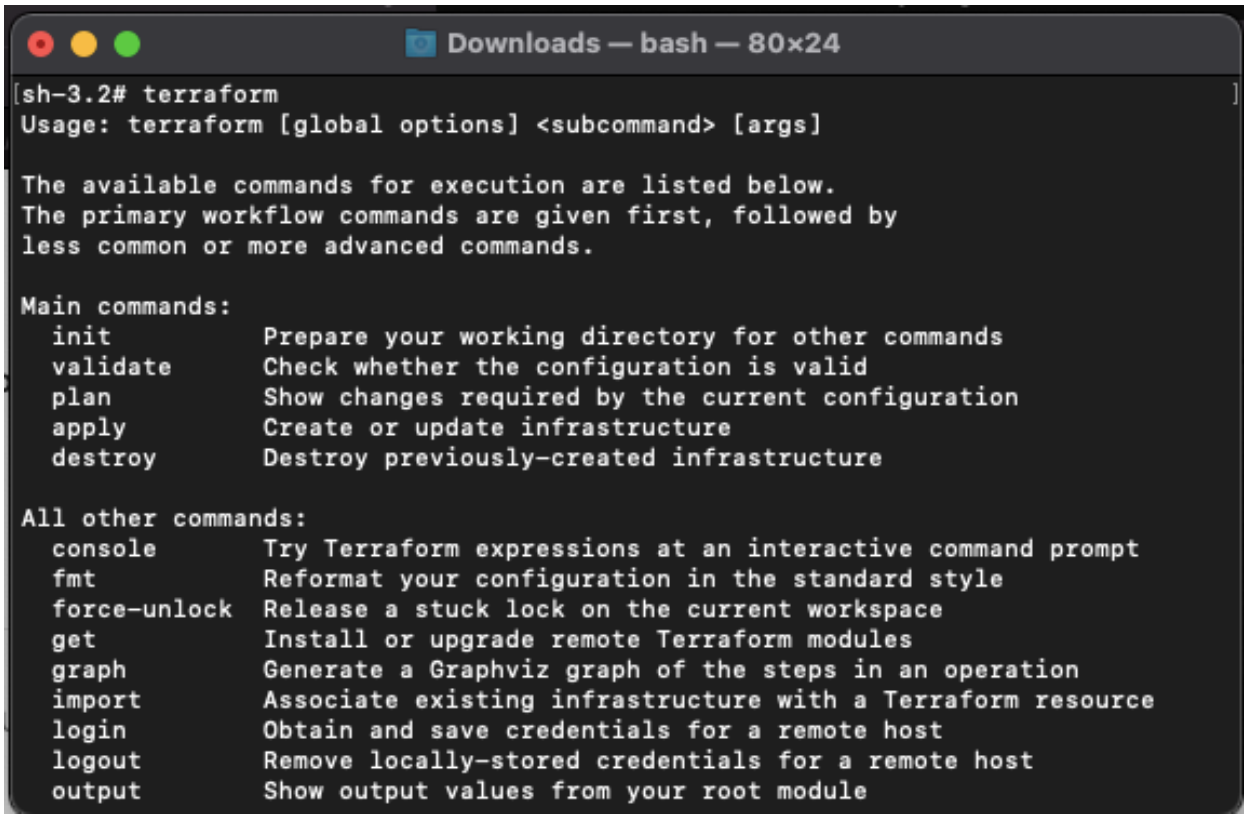
In this step you need to extract the downloaded Terraform file, remember that you downloaded the Terraform package:

Check if the file is located on Donwloads Folder, Unzip and move the binary to /usr/local/bin directory



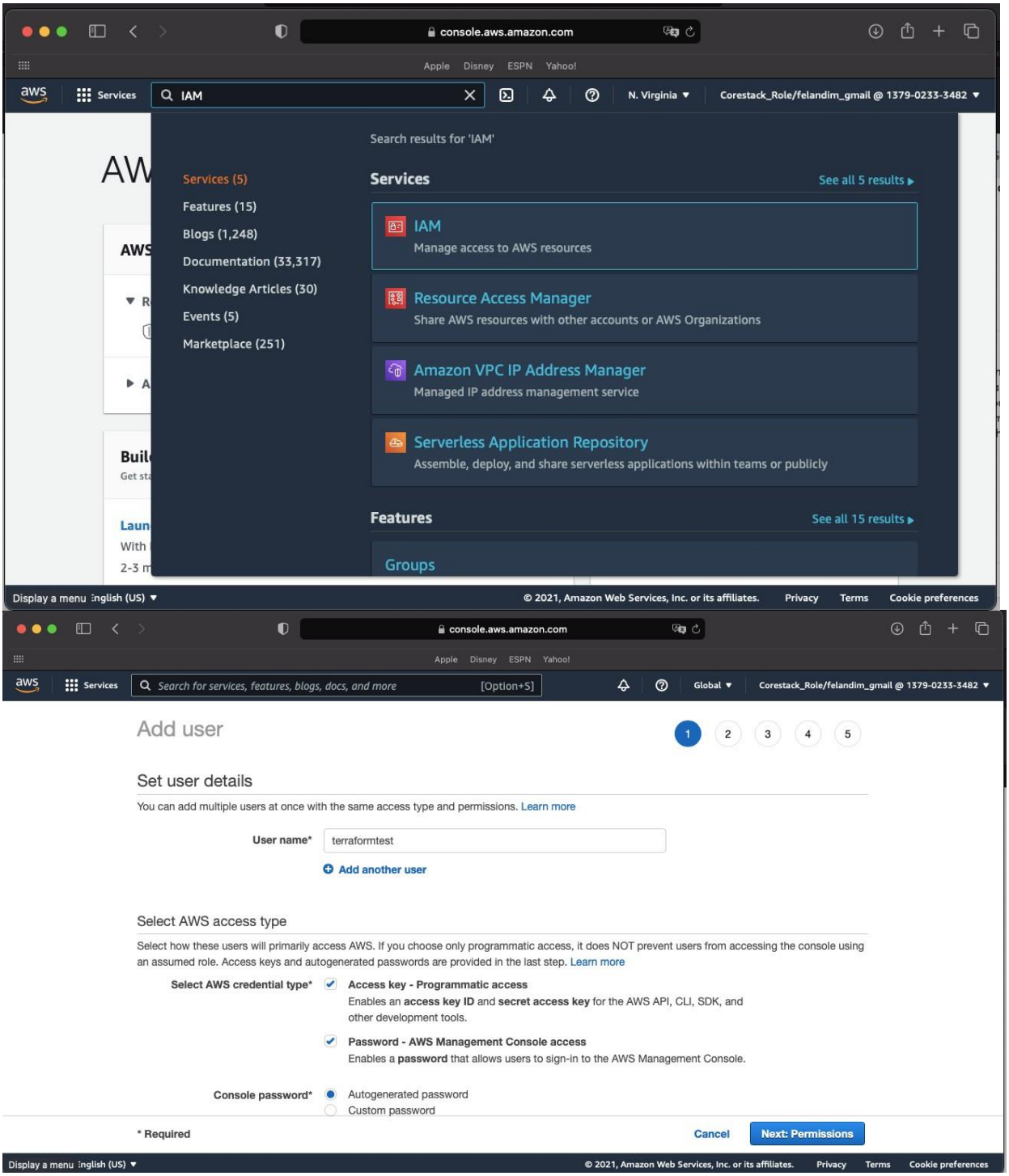
And check the terraform version:

See the Terraform command usage options, just run # terraform

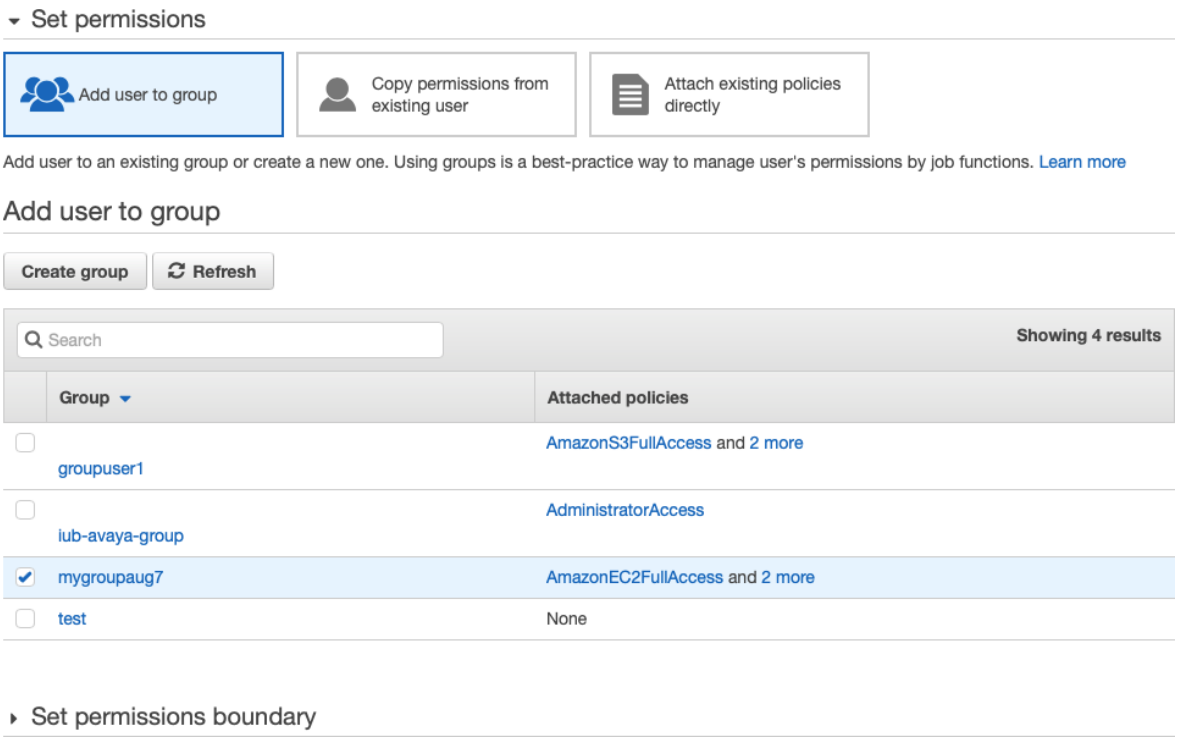


Create IAM user on AWS environment

Based on Link
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html



After that, click on: Next Permissions, and select the group with EC2 full access (and create a new group with this specific access type.



On last screen you can visualize all information about the user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	terraformtest
AWS access type	Programmatic access and AWS Management Console access
Console password type	Autogenerated
Require password reset	Yes
Permissions boundary	Permissions boundary is not set

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	mygroupaug7
Managed policy	IAMUserChangePassword

Tags

No tags were added.

Cancel Previous Create user

The user has been created

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

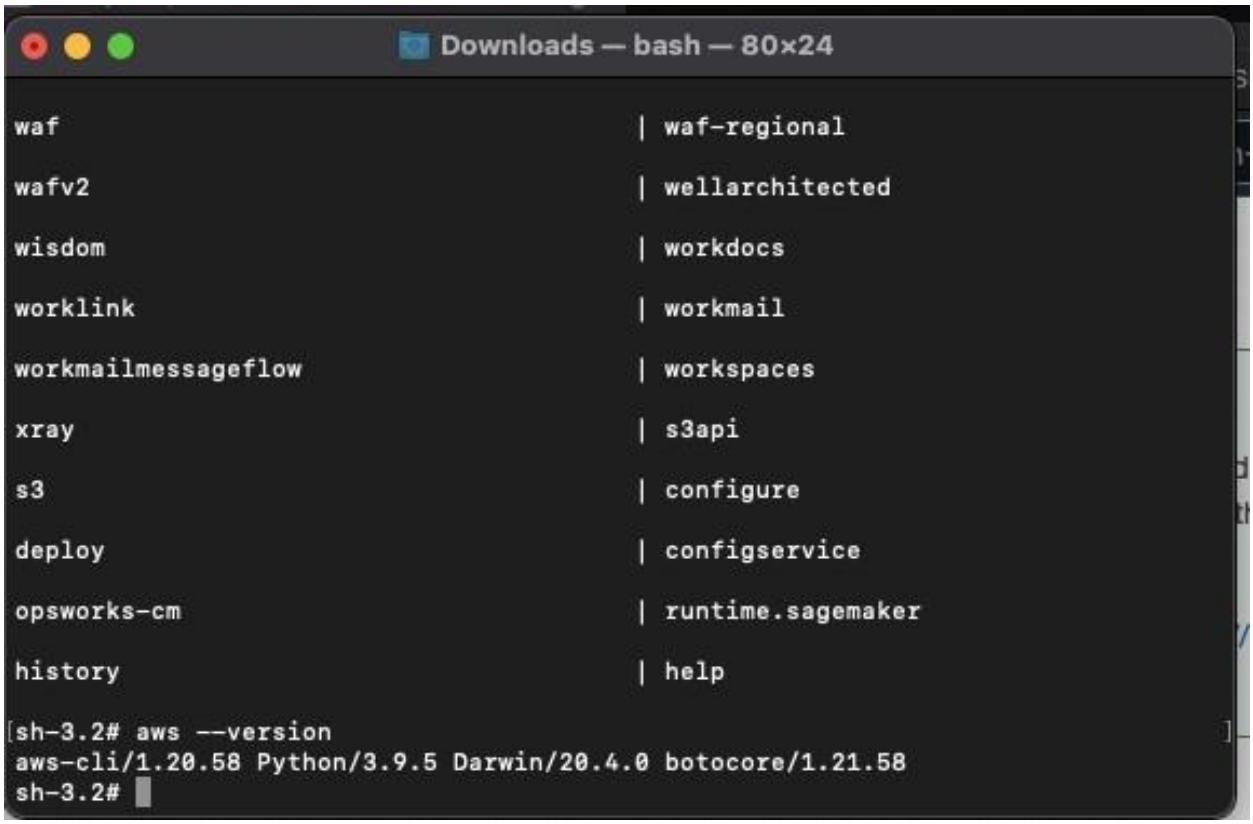
Users with AWS Management Console access can sign-in at: https://137902333482.signin.aws.amazon.com/console

Download .csv

User	Access key ID	Secret access key
terraformtest	AKIASAG46TYVLIYN45W2	CXfNLPswDmGSLQXjKjdO/DuVzrsTrqdsYbUMFbGi Hide

With this Access Key and Secret key, is possible to access your AWS account from your local machine.

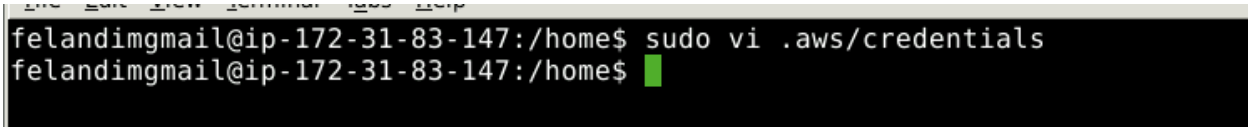
The machine needs to have AWS package installed



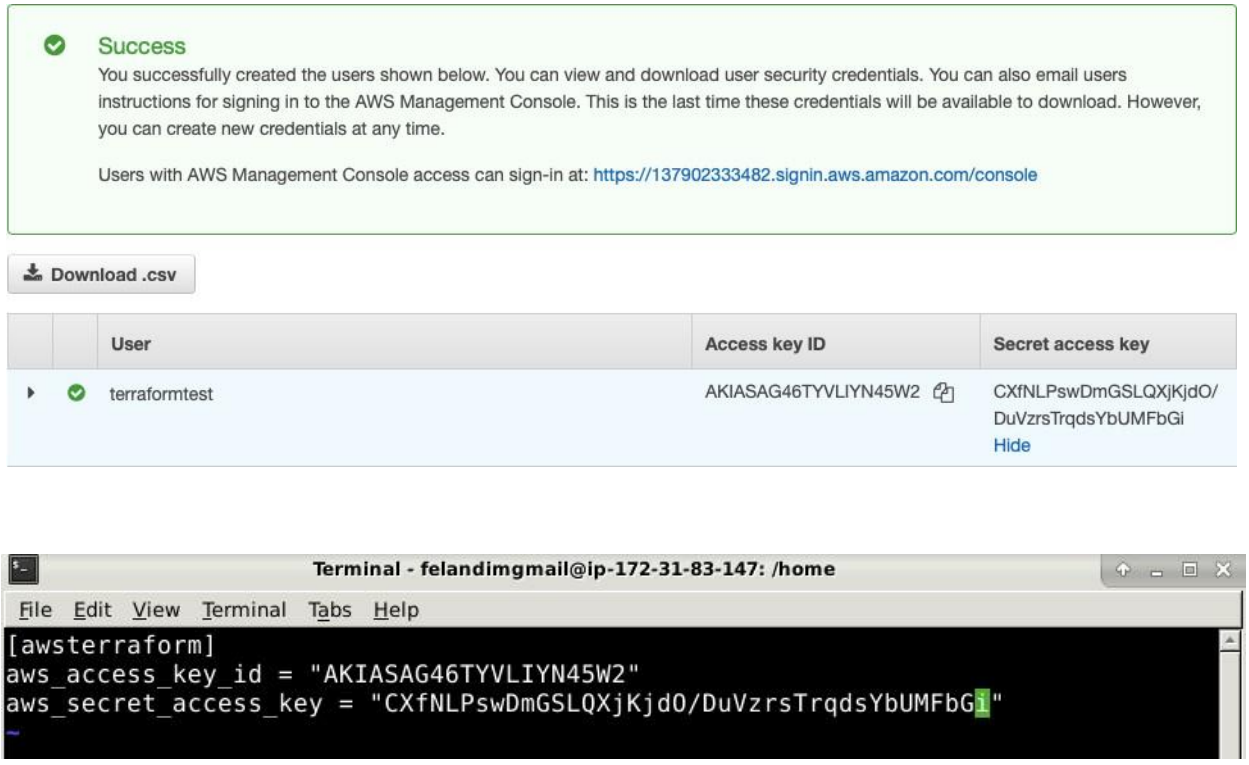
Terraform

Credentials file creation

Let's now create a file called credentials in your /home/.aws and add the keys that were created on AWS Console. To do this, execute:



The access key ID and Access key are requested by this screen on AWS



After that press ESC, :wq

Create PEM file

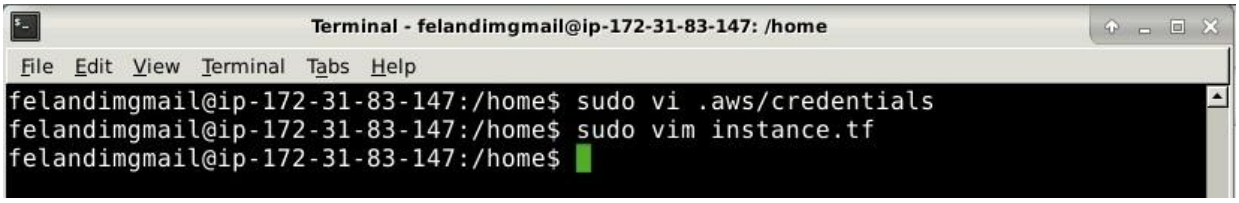
Later we are going to associate both public and private keys with AWS EC2 instances.
Let us generate the key pair using the following command:

Sudo ssh-keygen -t rsa -b 2048
By default, the above command will generate the public as well as private key at location ***/root/.ssh/id_rsa***, but we override the destination with a custom ***/home/felandimgmail/key/tfkey***

```
felandimgmail@ip-172-31-83-147:/home$ sudo ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /home/felandimgmail/key
/tfkey
/home/felandimgmail/key/tfkey already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/felandimgmail/key/tfkey.
Your public key has been saved in /home/felandimgmail/key/tfkey.pub.
The key fingerprint is:
SHA256:jd81bSqMjDdL9G5sHPCYfQLHsMgcZzy+NClNg5phwhw root@ip-172-31-83-147
The key's randomart image is:
+---[RSA 2048]----+
|   oE.   o       |
|    + o o O      |
|     o * O B     |
|      o =oX o  . |
|        Soo@  o o|
|         =+==..+ |
|        . Bo=+.  |
|         o +=.   |
|          .o.    |
+----[SHA256]-----+
felandimgmail@ip-172-31-83-147:/home$
```

Create TF File

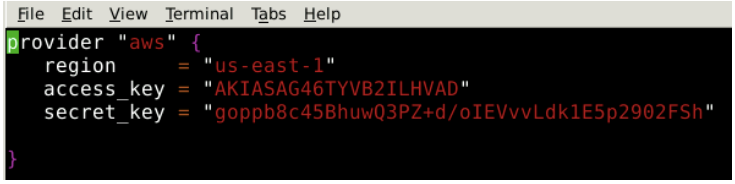
Now must create a TF file that is responsible for create a EC2 machine



Here in this file, we specify that we are using the AWS provider, we are using the “us-east-1” region, we share our access keys, which we got there at user creation in AWS.

We pass the AMI (it would be an identifier) to say that we are creating an Ubuntu distro, and using the instance type “t2.micro”.

The first part is to define the **access_key** and **secret_key** for AWS environment



Define what is the AMI, Instance, and private key to access the EC2 environment

```
resource "aws_instance" "ec2_example" {

    ami = "ami-083654bd07b5da81d"
    instance_type = "t2.micro"
    key_name= "tfkey"

  connection {
      type        = "ssh"
      host        = self.public_ip
      user        = "ubuntu"
      private_key = file("/home/felandimgmail/key/tfkey")
      timeout     = "4m"
   }
}
```

Security group, the open the ports to access this EC2 machine (this case all ports are open)

```
resource "aws_security_group" "main" {
  ingress {
    from_port = 22
    to_port = 22
    protocol = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

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

The keypar to access the EC2 machine

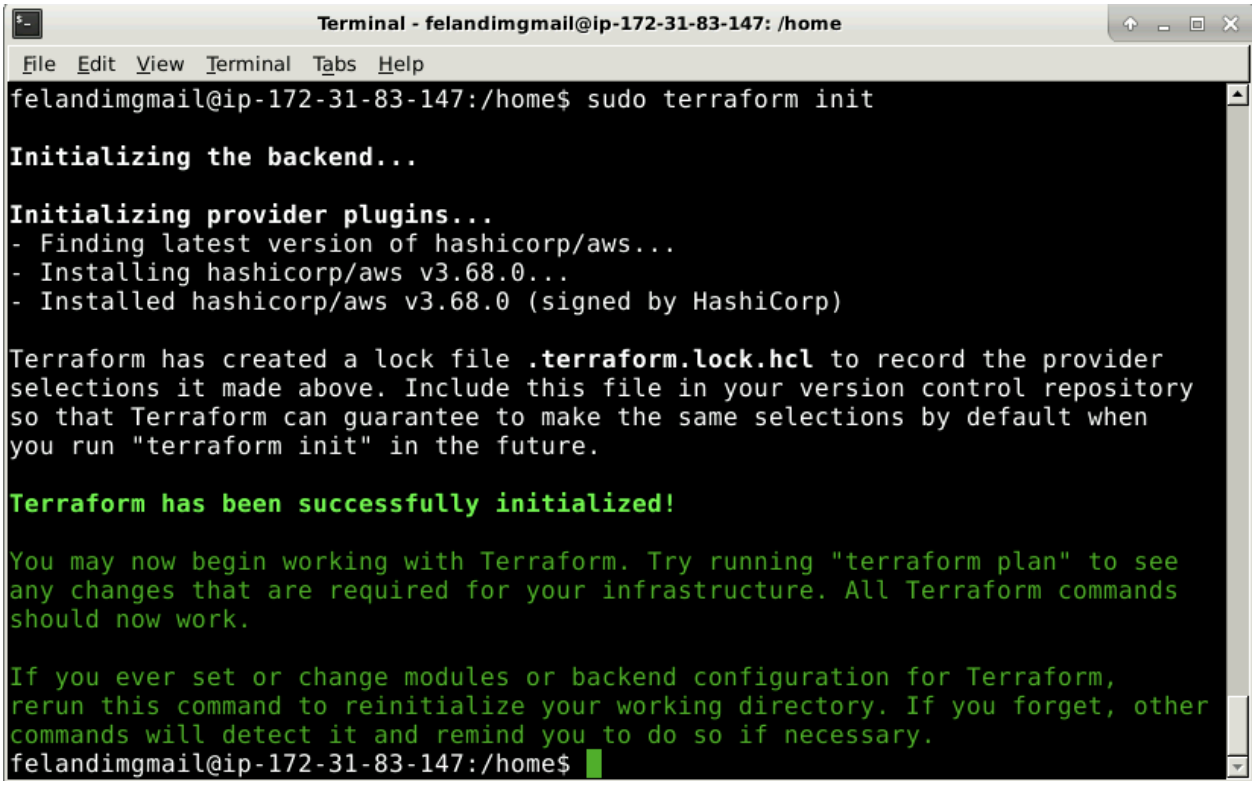
```
resource "aws_key_pair" "deployer" {
  key_name   = "tfkey"
  public_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDHRV3GjKzL5ZOXI8BybidJ4IUN
uTBVvF5txlkVC5q3foag9vU4+4U/SGu+IQTD3zf08voc7UDXSLSckG50NewdeURBOKHBUKRfBprehpEM
WGE8FOZtRCtWlMiCG4unyFz8L5Z3k73/QdQo3YKUa+4vkAo8Enl3wSzdRJNKq33pmS2HeZh5jl3I79yg
c388xbKgVLDqJY+WNlHAxGT5c27gvjt2GlJxR1HV+APWOAd7bzQWR3NIoeSH7RKkz+8cB978vV3Wr6Pn
zdEfbWZLh6DT2QxNdC+nlxJO+V6zZyAGmttqT4DHdU7IjixSEGfs6TCwVqVzj5Vd8GV7y6HIEv5Z root
@ip-172-31-83-147"
}
-- INSERT --                                                    36,2          Bot
```

This is the entire file:

```
resource "aws_key_pair" "deployer" {
  key_name   = "tfkey"
  public_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDHRV3GjKzL5ZOXI8BybidJ
4IUNuTBVvF5txlkVC5q3foag9vU4+4U/SGu+IQTD3zf08voc7UDXSLSckG50NewdeURBOKHBUKRf
BprehpEMWGE8FOZtRCtWlMiCG4unyFz8L5Z3k73/QdQo3YKUa+4vkAo8Enl3wSzdRJNKq33pmS2H
eZh5jl3I79ygc388xbKgVLDqJY+WNlHAxGT5c27gvjt2GlJxR1HV+APWOAd7bzQWR3NIoeSH7RKk
z+8cB978vV3Wr6PnzdEfbWZLh6DT2QxNdC+nlxJO+V6zZyAGmttqT4DHdU7IjixSEGfs6TCwVqVz
j5Vd8GV7y6HIEv5Z root@ip-172-31-83-147"
}
-- INSERT --                                                    36,2          Bot
```

Starting the case with terraform

Now we are executing the command **terraform Init**, that this command download the artifacts from AWS.



Now let's run the terraform plan command. This will allow us to see what Terraform will do before we decide to define the infrastructure.

The result of Terraform plan is the following:

```
aws_key_pair.deployer: Refreshing state... [id=tfkey]
aws_security_group.main: Refreshing state... [id=sg-07e436701e43c9a02]
aws_instance.ec2_example: Refreshing state... [id=i-02a06bcc6c63f8556]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since
the last "terraform apply":

 # aws_instance.ec2_example has been deleted
 - resource "aws_instance" "ec2_example" {
    - ami                          = "ami-083654bd07b5da81d" -> null
    - arn                          = "arn:aws:ec2:us-east-1:137902333482:instance/i-
02a06bcc6c63f8556" -> null
    - associate_public_ip_address        = true -> null
    - availability_zone                  = "us-east-1a" -> null
    - cpu_core_count                      = 1 -> null
```

```
    - instance_initiated_shutdown_behavior = "stop" -> null
    - instance_state                       = "running" -> null
    - instance_type                        = "t2.micro" -> null
    - ipv6_address_count                   = 0 -> null
    - ipv6_addresses                       = [] -> null
    - key_name                             = "tfkey" -> null
    - monitoring                           = false -> null
    - primary_network_interface_id         = "eni-08e87aa3baf226201" -> null
    - private_dns                          = "ip-172-31-80-74.ec2.internal" -> null
    - private_ip                           = "172.31.80.74" -> null
    - public_dns                           = "ec2-52-91-226-251.compute-1.amazonaws.com" -> null
    - public_ip                            = "52.91.226.251" -> null
    - secondary_private_ips                = [] -> null
    - security_groups                      = [
        - "default",
      ] -> null
    - source_dest_check                    = true -> null
    - subnet_id                            = "subnet-00b85fb07e260121b" -> null
    - tags_all                             = {} -> null
    - tenancy                              = "default" -> null
    - vpc_security_group_ids               = [
        - "sg-0c6bf99bd9fa7ff67",
      ] -> null

    - capacity_reservation_specification {
        - capacity_reservation_preference = "open" -> null
      }

    - credit_specification {
        - cpu_credits = "standard" -> null
      }

    - enclave_options {
        - enabled = false -> null
      }

    - metadata_options {
        - http_endpoint               = "enabled" -> null
```

```
      - http_put_response_hop_limit = 1 -> null
      - http_tokens               = "optional" -> null
    }

  - root_block_device {
      - delete_on_termination = true -> null
      - device_name          = "/dev/sda1" -> null
      - encrypted            = false -> null
      - iops                 = 100 -> null
      - tags                 = {} -> null
      - throughput           = 0 -> null
      - volume_id            = "vol-08b481ab35590ae36" -> null
      - volume_size          = 8 -> null
      - volume_type          = "gp2" -> null
    }
  }

 # aws_key_pair.deployer has been deleted
 - resource "aws_key_pair" "deployer" {
    - arn        = "arn:aws:ec2:us-east-1:137902333482:key-pair/tfkey" -> null
    - fingerprint = "35:9d:d5:ac:22:33:08:1c:07:f5:08:16:56:3a:d0:e5" -> null
    - id         = "tfkey" -> null
    - key_name    = "tfkey" -> null
    - key_pair_id = "key-06b840832c9c9a102" -> null
    - public_key = "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDJdKjDGDryfHMHfXJ0dwiZbI3pjA42xg3Bkg2zIIuY4073
QCcvtp+VNpvGWfahsxpOXbUPjcoM5QGNSCwSmLZGkrlE5kTJQnJV2O5xrTM9+9kRKdf7OrBSvkVi
c0n7n/6m8EKigeqbA+T5WbntBnw2cz09fN+436VPOzuRWOI+KFMz4desNmSZgmEYxA46wURBQ
TzNMJQH7SQwW2R3ontwxHhp8QELbBPWNPCit3159B7iwSjKCnJ+VBtYyZvNXaVOzhap0e0rIXqo
MQ50iHJl5/QgUaMmgHCVY+fTjYpPleIcIZLaU0qoEnpmGnyCkRX9K/FBkI0TV1HjD1ju9uft
root@ip-172-31-83-147" -> null
    - tags_all    = {} -> null
  }

 # aws_security_group.main has been deleted
 - resource "aws_security_group" "main" {
    - arn                  = "arn:aws:ec2:us-east-1:137902333482:security-group/sg-
07e436701e43c9a02" -> null
    - description          = "Managed by Terraform" -> null
    - egress               = [
      - {
          - cidr_blocks     = [
              - "0.0.0.0/0",
            ]
```

```
          - ipv6_cidr_blocks = [
              - "::/0",
            ]
          - prefix_list_ids  = []
          - protocol         = "-1"
          - security_groups  = []
          - self             = false
          - to_port          = 0
        },
    ] -> null
    - id                   = "sg-07e436701e43c9a02" -> null
    - ingress              = [] -> null
    - name                 = "terraform-20211210153346670000000001" -> null
    - name_prefix          = "terraform-" -> null
    - owner_id             = "137902333482" -> null
    - revoke_rules_on_delete = false -> null
    - tags_all             = {} -> null
    - vpc_id               = "vpc-0e82d013a522d5d53" -> null
  }
```

Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the following plan may include actions to undo or respond to these changes.

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.ec2_example will be created
  + resource "aws_instance" "ec2_example" {
      + ami                          = "ami-083654bd07b5da81d"
      + 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)
```

```
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_state                       = (known after apply)
    + instance_type                        = "t2.micro"
    + ipv6_address_count                   = (known after apply)
    + ipv6_addresses                       = (known after apply)
    + key_name                             = "tfkey"
    + monitoring                           = (known after apply)
    + outpost_arn                          = (known after apply)
    + password_data                        = (known after apply)
    + placement_group                      = (known after apply)


    + private_dns                          = (known after apply)
    + private_ip                           = (known after apply)
    + public_dns                           = (known after apply)
    + public_ip                            = (known after apply)
    + secondary_private_ips                = (known after apply)
    + security_groups                      = (known after apply)
    + source_dest_check                    = true
    + subnet_id                            = (known after apply)
    + tags_all                             = (known after apply)
    + tenancy                              = (known after apply)
    + user_data                            = (known after apply)
    + user_data_base64                     = (known after apply)
    + vpc_security_group_ids               = (known after apply)

    + capacity_reservation_specification {
        + capacity_reservation_preference = (known after apply)

        + capacity_reservation_target {
            + capacity_reservation_id = (known after apply)
          }
      }

    + ebs_block_device {
        + delete_on_termination = (known after apply)
        + device_name           = (known after apply)
        + encrypted             = (known after apply)
        + iops                  = (known after apply)
```

```
    }

  + enclave_options {
      + enabled = (known after apply)
    }

  + ephemeral_block_device {
      + device_name  = (known after apply)
      + no_device    = (known after apply)
      + virtual_name = (known after apply)
    }

  + metadata_options {
      + http_endpoint               = (known after apply)
      + http_put_response_hop_limit = (known after apply)
      + http_tokens                 = (known after apply)
    }

  + network_interface {
      + delete_on_termination = (known after apply)
      + device_index          = (known after apply)
      + network_interface_id  = (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)
      + throughput            = (known after apply)
      + volume_id             = (known after apply)
      + volume_size           = (known after apply)
      + volume_type           = (known after apply)
    }
}
```

```
  # aws_key_pair.deployer will be created
  + resource "aws_key_pair" "deployer" {
      + arn            = (known after apply)
      + fingerprint    = (known after apply)
      + id             = (known after apply)
      + key_name       = "tfkey"
      + key_name_prefix = (known after apply)
      + key_pair_id    = (known after apply)
      + public_key     = "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDHRV3GjKzL5ZOXI8BybidJ4IUNuTBVvF5txlkVC5q3foag
9vU4+4U/SGu+IQTD3zf08voc7UDXSLSckG50NewdeURBOKHBUKRfBprehpEMWGE8FOZtRCtWI
MiCG4unyFz8L5Z3k73/QdQo3YKUa+4vkAo8Enl3wSzdRJNKq33pmS2HeZh5jl3I79ygc388xbKgVLD
qJY+WNlHAxGT5c27gvjt2GlJxR1HV+APWOAd7bzQWR3NIoeSH7RKkz+8cB978vV3Wr6PnzdEfbW
ZLh6DT2QxNdC+nlxJO+V6zZyAGmttqT4DHdU7IjixSEGfs6TCwVqVzj5Vd8GV7y6HIEv5Z root@ip-
172-31-83-147"
      + tags_all       = (known after apply)
    }

  # aws_security_group.main will be created
  + resource "aws_security_group" "main" {
      + arn                    = (known after apply)
      + description            = "Managed by Terraform"
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = ""
              + from_port        = 0
              + ipv6_cidr_blocks = [
                  + "::/0",
                ]
              + prefix_list_ids  = []
              + protocol         = "-1"
              + security_groups  = []
              + self             = false
              + to_port          = 0
            },
        ]
      + id                     = (known after apply)
      + ingress                = (known after apply)
      + name                   = (known after apply)
      + name_prefix            = (known after apply)
```

```
    + owner_id             = (known after apply)
    + revoke_rules_on_delete = false


  }

Plan: 3 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.
```

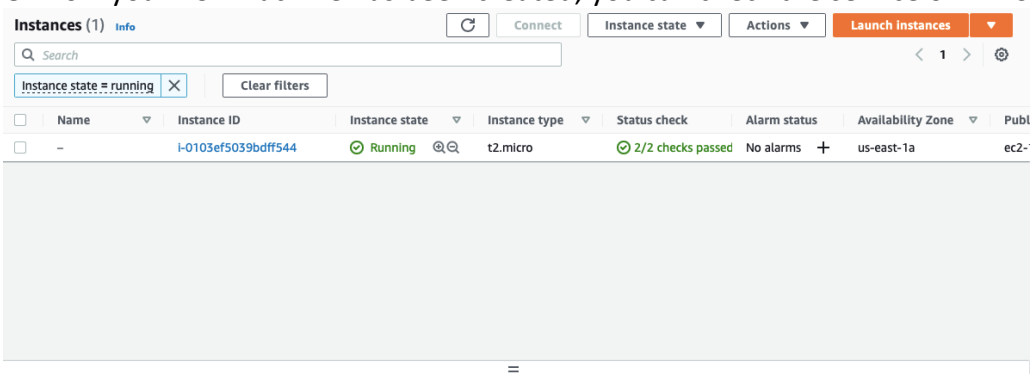
And Finally type: ***sudo terraform apply***

```
  Enter a value: yes

aws_key_pair.deployer: Creating...
aws_security_group.main: Creating...
aws_instance.ec2_example: Creating...
aws_key_pair.deployer: Creation complete after 0s [id=tfkey]
aws_security_group.main: Creation complete after 3s [id=sg-0e3690a86dbd762d9
]
aws_instance.ec2_example: Still creating... [10s elapsed]
aws_instance.ec2_example: Still creating... [20s elapsed]
aws_instance.ec2_example: Creation complete after 23s [id=i-0103ef5039bdff54
4]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
felandimgmail@ip-172-31-83-147:/home$
```

Ok now your EC2 machine has been created, you can check the service on AWS Console



Configure EC2 environment

To install Java, Jenkins and python, you need to connect to EC2 machine from ssh, bellow the executed to connect:

```
felandimgmail@ip-172-31-83-147:~/key$ sudo su
root@ip-172-31-83-147:/home/felandimgmail/key# chmod 400 tfkey
root@ip-172-31-83-147:/home/felandimgmail/key# ssh -i "tfkey" ubuntu@ec2-3-95-15
2-35.compute-1.amazonaws.com
The authenticity of host 'ec2-3-95-152-35.compute-1.amazonaws.com (3.95.152.35)'
 can't be established.
ECDSA key fingerprint is SHA256:4Ntr/c1XZh46WiV3p3Ec/lBZQ8o2d+jWc5GKEOig1v0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-3-95-152-35.compute-1.amazonaws.com,3.95.152.35'
 (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)
```

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

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

  System information as of Fri Dec 10 23:51:28 UTC 2021

  System load:  0.0               Processes:             98
  Usage of /:   17.9% of 7.69GB   Users logged in:       0
  Memory usage: 19%               IPv4 address for eth0: 172.31.83.80
  Swap usage:   0%

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


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


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

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

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

ubuntu@ip-172-31-83-80:~$
```

Java Run Time

First important item about these processes, JRE is very important for all artifacts, because of this, this is the first step.

sudo apt-get update

```
ubuntu@ip-172-31-83-80:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [11
4 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [
108 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packag
es [8628 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-
en [5124 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f
Metadata [265 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Pack
ages [144 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translatio
```

Verify the installation of java using the command "***java -version***", if Java is not currently installed, you will see the following output:

```
ubuntu@ip-172-31-83-80:~$ java -version

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

sudo apt install openjdk-11-jre-headless  # version 11.0.11+9-0ubuntu2~20.04, or
sudo apt install default-jre              # version 2:1.11-72
sudo apt install openjdk-13-jre-headless  # version 13.0.7+5-0ubuntu1~20.04
sudo apt install openjdk-16-jre-headless  # version 16.0.1+9-1~20.04
sudo apt install openjdk-17-jre-headless  # version 17.0.1+12-1~20.04
sudo apt install openjdk-8-jre-headless   # version 8u292-b10-0ubuntu1~20.04
```

Run the following command to install the default Java Runtime Environment (JRE), which will install the OpenJDK 11 JRE:

sudo apt install default-jre

```
ubuntu@ip-172-31-83-80:~$ sudo apt install default-jre
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  at-spi2-core ca-certificates-java default-jre-headless fontconfig-config
  fonts-dejavu-core fonts-dejavu-extra java-common 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
  libcups2 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1
  libfontconfig1 libfontenc1 libgif7 libgl1 libgl1-mesa-dri libglapi-mesa
  libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libharfbuzz0b libice6
```

The JRE will allow you to run almost any Java software.

Verify installation with:

java -version

```
ubuntu@ip-172-31-83-80:~$ java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sha
ring)
ubuntu@ip-172-31-83-80:~$
```

Ansible

Use the below command to install Ansible

sudo apt-get install ansible

```
root@ip-172-31-83-80:/home/ubuntu# sudo apt-get install ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ieee-data python3-argcomplete python3-crypto python3-dnspython
  python3-jmespath python3-kerberos python3-libcloud python3-lockfile
  python3-netaddr python3-ntlm-auth python3-requests-kerberos
  python3-requests-ntlm python3-selinux python3-winrm python3-xmltodict
Suggested packages:
  cowsay sshpass python-lockfile-doc ipython3 python-netaddr-docs
The following NEW packages will be installed:
  ansible ieee-data python3-argcomplete python3-crypto python3-dnspython
  python3-jmespath python3-kerberos python3-libcloud python3-lockfile
  python3-netaddr python3-ntlm-auth python3-requests-kerberos
  python3-requests-ntlm python3-selinux python3-winrm python3-xmltodict
0 upgraded, 16 newly installed, 0 to remove and 38 not upgraded.
Need to get 9644 kB of archives.
After this operation, 90.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Abort.
root@ip-172-31-83-80:/home/ubuntu#
```

Jenkins

On the same terminal that executes the java installation, can be execute the Jenkins installation, with the command:

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

```
root@ip-172-31-83-80:/home/ubuntu# wget -q -O - https://pkg.jenkins.io/debian/je
nkins.io.key | sudo apt-key add -
OK
```

And run the command ***sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian binary/ > /etc/apt/sources.list.d/jenkins.list'*** and after the command ***sudo apt-get update***

```
root@ip-172-31-83-80:/home/ubuntu# sudo sh -c 'echo deb http://pkg.jenkins-ci.or
g/debian binary/ > /etc/apt/sources.list.d/jenkins.list'
root@ip-172-31-83-80:/home/ubuntu# sudo apt-get update
Ign:1 http://pkg.jenkins-ci.org/debian binary/ InRelease
Get:2 http://pkg.jenkins-ci.org/debian binary/ Release [2044 B]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:6 http://pkg.jenkins-ci.org/debian binary/ Release.gpg [833 B]
Get:7 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:8 http://pkg.jenkins-ci.org/debian binary/ Packages [40.7 kB]
Fetched 157 kB in 0s (337 kB/s)
Reading package lists... Done
```

Has installed all pre-requisites to install Jenkins, to install Jenkins execute the command ***sudo apt-get install jenkins***

```
root@ip-172-31-83-80:/home/ubuntu# sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon net-tools
The following NEW packages will be installed:
  daemon jenkins net-tools
0 upgraded, 3 newly installed, 0 to remove and 40 not upgraded.
Need to get 73.4 MB of archives.
After this operation, 74.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 daemon
 amd64 0.6.4-1build2 [96.3 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 net-tools
amd64 1.60+git20180626.aebd88e-1ubuntu1 [196 kB]
Get:1 http://pkg.jenkins-ci.org/debian binary/ jenkins 2.324 [73.1 MB]
Fetched 73.4 MB in 11s (6756 kB/s)
Selecting previously unselected package daemon.
(Reading database ... 64755 files and directories currently installed.)
Preparing to unpack .../daemon_0.6.4-1build2_amd64.deb ...
Unpacking daemon (0.6.4-1build2) ...
Selecting previously unselected package net-tools.
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ..
.
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../archives/jenkins_2.324_all.deb ...
Unpacking jenkins (2.324) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up daemon (0.6.4-1build2) ...
Setting up jenkins (2.324) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
root@ip-172-31-83-80:/home/ubuntu#
```

After Jenkins installation, is necessary to start the Jenkins, execute following command:

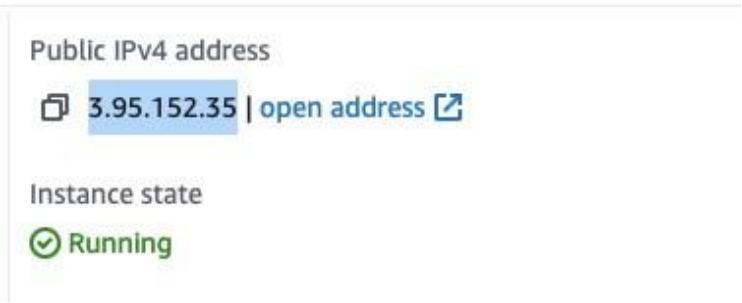
sudo systemctl start jenkinsTo verify the start status of Jenkins, execute the following command

sudo systemctl status jenkins

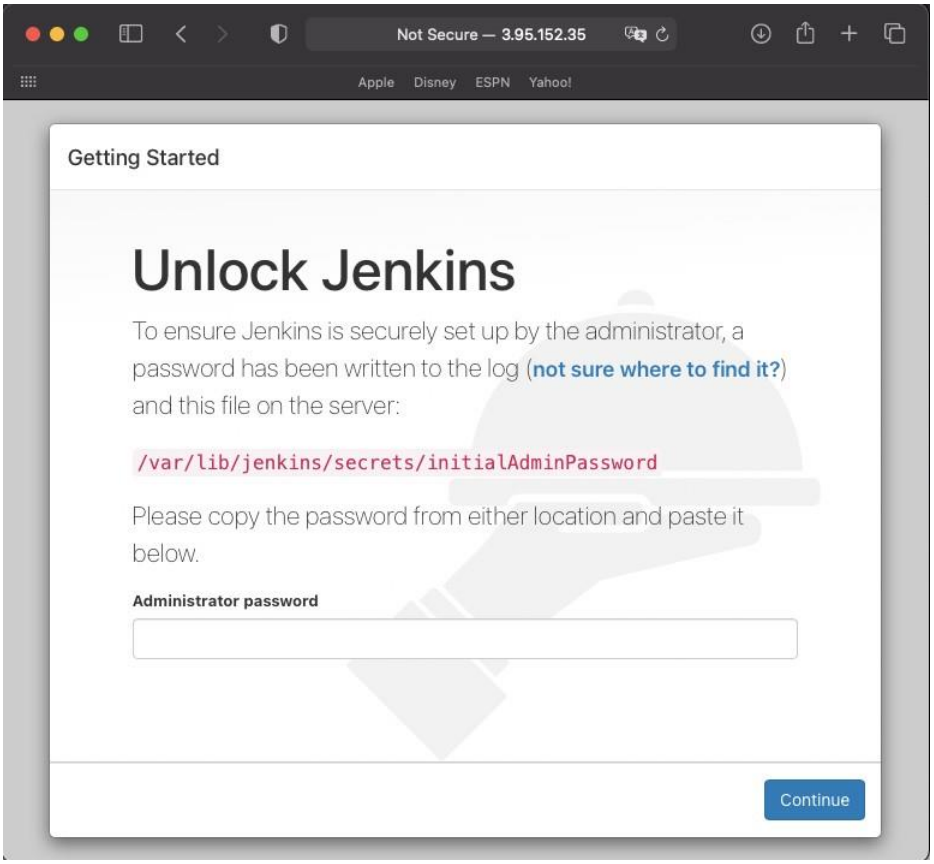
```
root@ip-172-31-83-80:/home/ubuntu# sudo systemctl start jenkins
root@ip-172-31-83-80:/home/ubuntu# sudo systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
     Loaded: loaded (/etc/init.d/jenkins; generated)
     Active: active (exited) since Sat 2021-12-11 00:19:48 UTC; 5min ago
       Docs: man:systemd-sysv-generator(8)
      Tasks: 0 (limit: 1154)
     Memory: 0B
     CGroup: /system.slice/jenkins.service

Dec 11 00:19:47 ip-172-31-83-80 systemd[1]: Starting LSB: Start Jenkins at boot>
Dec 11 00:19:47 ip-172-31-83-80 jenkins[16733]: Correct java version found
Dec 11 00:19:47 ip-172-31-83-80 jenkins[16733]:  * Starting Jenkins Automation >
Dec 11 00:19:47 ip-172-31-83-80 su[16767]: (to jenkins) root on none
Dec 11 00:19:47 ip-172-31-83-80 su[16767]: pam_unix(su-l:session): session open>
Dec 11 00:19:47 ip-172-31-83-80 su[16767]: pam_unix(su-l:session): session clos>
Dec 11 00:19:48 ip-172-31-83-80 jenkins[16733]:    ...done.
Dec 11 00:19:48 ip-172-31-83-80 systemd[1]: Started LSB: Start Jenkins at boot >
lines 1-16/16 (END)
```

Copy the IPV4 of this Server



And access the website with port 8080 and is show the following page



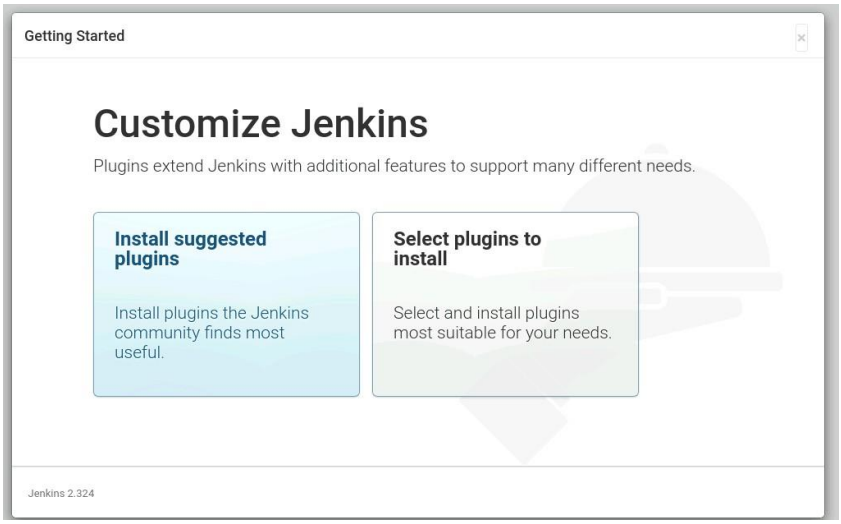
On terminal type the following command to verify Jenkins Initial admin password, is necessary to execute this command:

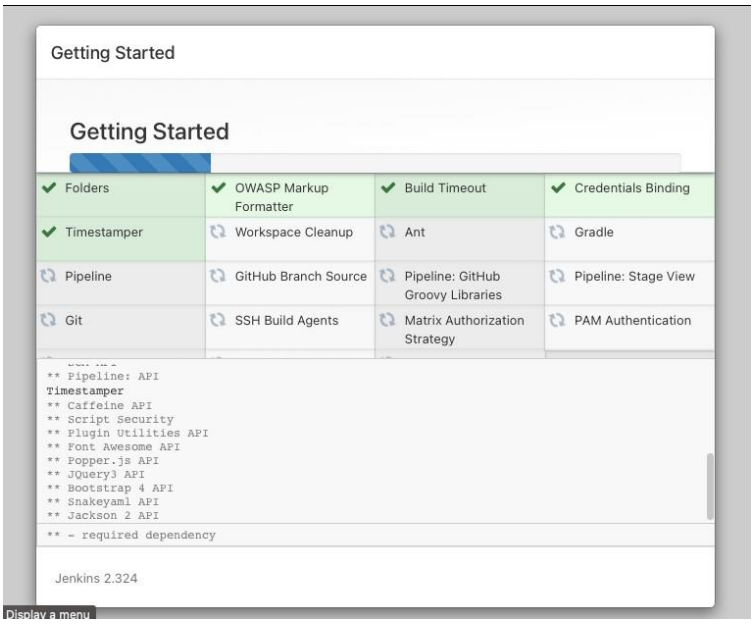
sudo cat /var/lib/jenkins/secrets/initialAdminPassword

```
root@ip-172-31-83-80:/home/ubuntu# sudo cat /var/lib/jenkins/secrets/initialAdmi
nPassword
2d2759daa4c44d509f953a7959714789
root@ip-172-31-83-80:/home/ubuntu#
```

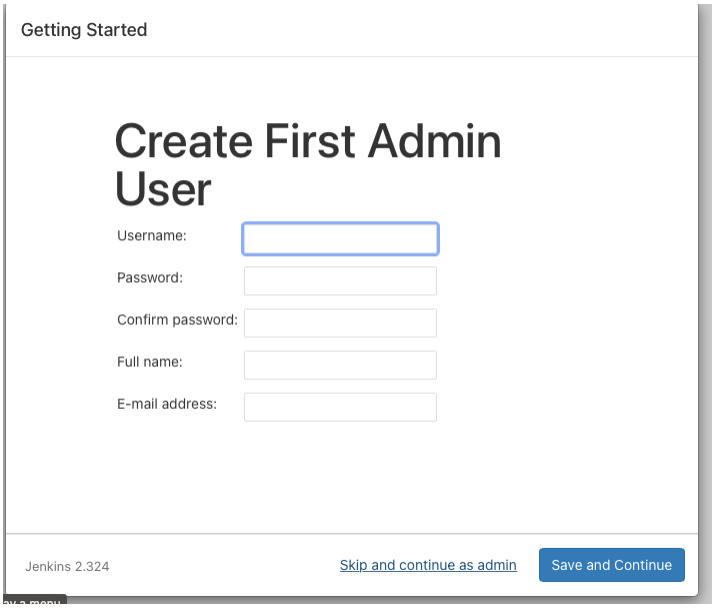
Copy this password and paste on the field Administrator password on browser

Click on Install suggested plugins

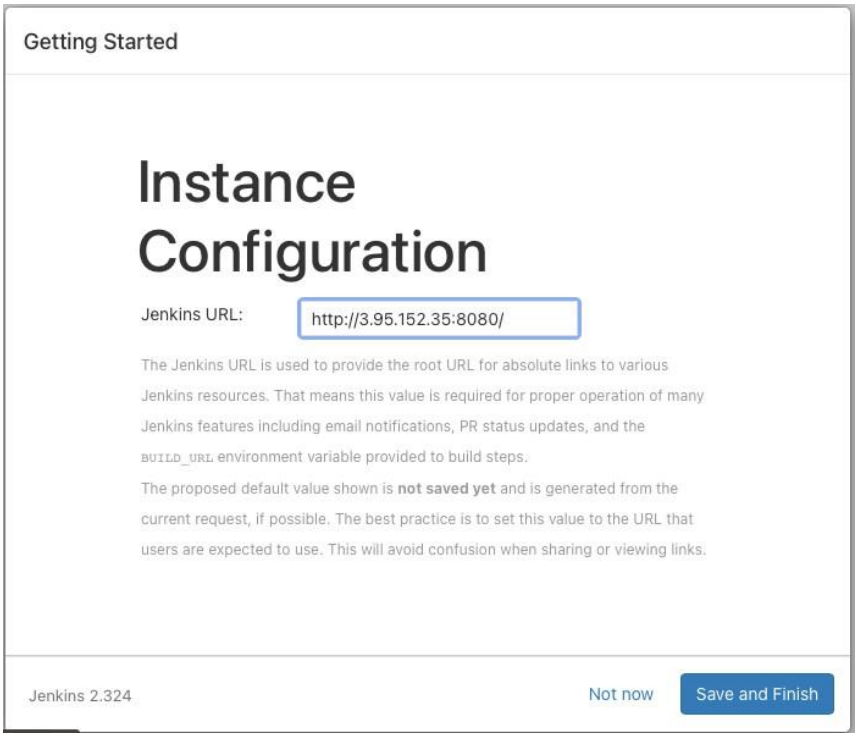




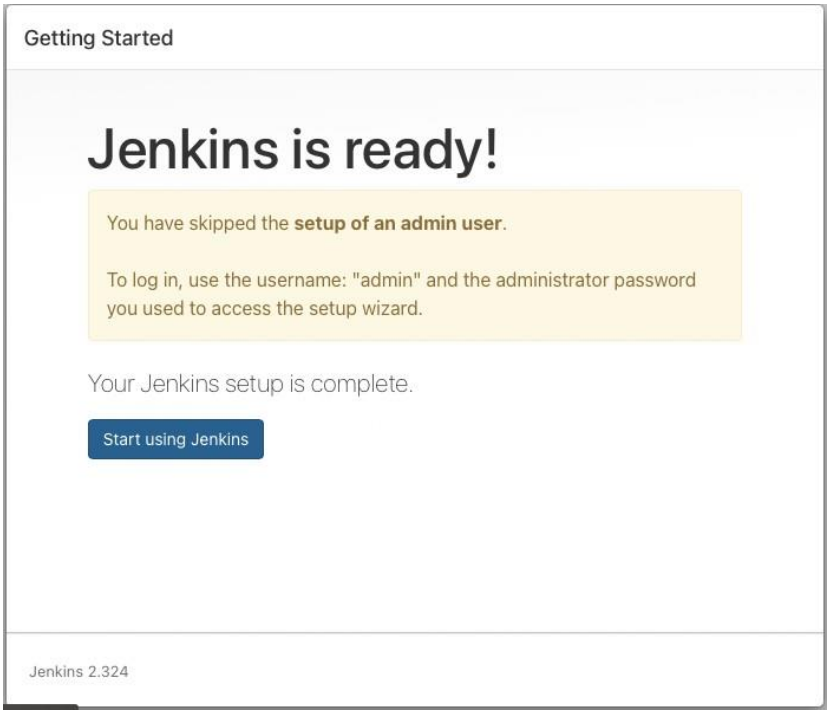
To facility the currently process, we going to proceed with admin access

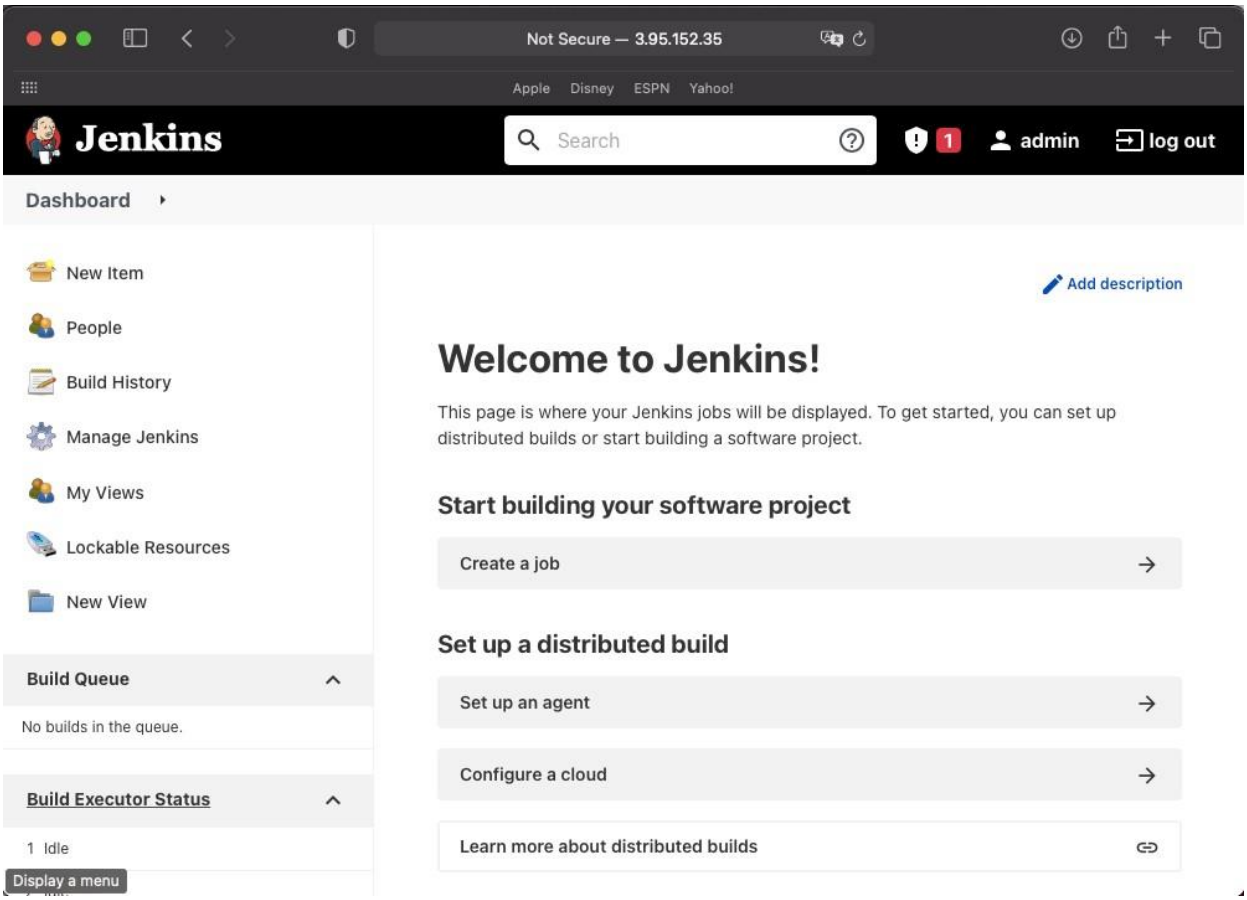


Is possible to change the Jenkins default URL, but at this moment we proceed with this url and port



The default installation has executed


Getting Started
Jenkins is ready!
You have skipped the setup of an admin user.
To log in, use the username: "admin" and the administrator password you used to access the setup wizard.
Your Jenkins setup is complete.
Start using Jenkins
Jenkins 2.324



Python

Get updates with command ***sudo apt update***

```
root@ip-172-31-83-80:/home/ubuntu# sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [11
4 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [
108 kB]
Ign:4 http://pkg.jenkins-ci.org/debian binary/ InRelease
Hit:5 http://pkg.jenkins-ci.org/debian binary/ Release
Get:6 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 336 kB in 1s (655 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
40 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-83-80:/home/ubuntu#
```

The software-properties-common package gives you better control over your package manager by letting you add PPA (Personal Package Archive) repositories. Install the supporting software with the command:

sudo apt install software-properties-common

```
root@ip-172-31-83-80:/home/ubuntu# sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties
The following packages will be upgraded:
  python3-software-properties software-properties-common
2 upgraded, 0 newly installed, 0 to remove and 38 not upgraded.
Need to get 35.5 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 so
ftware-properties-common all 0.99.9.8 [10.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 py
thon3-software-properties all 0.99.9.8 [24.9 kB]
Fetched 35.5 kB in 0s (1565 kB/s)
(Reading database ... 64823 files and directories currently installed.)
Preparing to unpack .../software-properties-common_0.99.9.8_all.deb ...
Unpacking software-properties-common (0.99.9.8) over (0.98.9.5) ...
Preparing to unpack .../python3-software-properties_0.99.9.8_all.deb ...
Unpacking python3-software-properties (0.99.9.8) over (0.98.9.5) ...
Setting up python3-software-properties (0.99.9.8) ...
Setting up software-properties-common (0.99.9.8) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
root@ip-172-31-83-80:/home/ubuntu#
```

Deadsnakes is a PPA with neweí íeleases than the default Ubuntu íepositoíies. Add thePPA by enteíing the following:

sudo add-apī-íeposiītoíy ppa:deadsnakes/ppa

1The system will píompt you to píess enteí to continue. Do so, and allow it to finish. Refíesh the package lists again:

```
root@ip-172-31-83-80:/home/ubuntu# sudo add-apt-repository ppa:deadsnakes/ppa
 This PPA contains more recent Python versions packaged for Ubuntu.

Disclaimer: there's no guarantee of timely updates in case of security problems
or other issues. If you want to use them in a security-or-otherwise-critical env
ironment (say, on a production server), you do so at your own risk.

Update Note
===========
Please use this repository instead of ppa:fkrull/deadsnakes.

Reporting Issues
================

Issues can be reported in the master issue tracker at:
https://github.com/deadsnakes/issues/issues

Supported Ubuntu and Python Versions
====================================
```

sudo apī updaīe

```
root@ip-172-31-83-80:/home/ubuntu# sudo apt update
Ign:1 http://pkg.jenkins-ci.org/debian binary/ InRelease
Hit:2 http://pkg.jenkins-ci.org/debian binary/ Release
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [1
4 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
108 kB]
Hit:7 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu focal InRelease
Get:8 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 336 kB in 1s (611 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
38 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Now you can install python with command

```
root@ip-172-31-83-80:/home/ubuntu# sudo apt install -y python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9
  dpkg-dev fakeroot g++ g++-9 gcc gcc-9 gcc-9-base libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0
```

And last execute the command ***sudo apī-geī insīall pyīhon-is-pyīhon3***

```
root@ip-172-31-83-80:/home/ubuntu# sudo apt-get install python-is-python3
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  python-is-python3
0 upgraded, 1 newly installed, 0 to remove and 38 not upgraded.
Need to get 2364 B of archives.
After this operation, 10.2 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 python-is-
python3 all 3.8.2-4 [2364 B]
Fetched 2364 B in 0s (151 kB/s)
Selecting previously unselected package python-is-python3.
(Reading database ... 70936 files and directories currently installed.)
Preparing to unpack .../python-is-python3_3.8.2-4_all.deb ...
Unpacking python-is-python3 (3.8.2-4) ...
Setting up python-is-python3 (3.8.2-4) ...
root@ip-172-31-83-80:/home/ubuntu# python --version
Python 3.8.10
root@ip-172-31-83-80:/home/ubuntu#
```

And ins possible to verify that python is installed

```
root@ip-172-31-83-80:/home/ubuntu# python --version
Python 3.8.10
root@ip-172-31-83-80:/home/ubuntu#
```