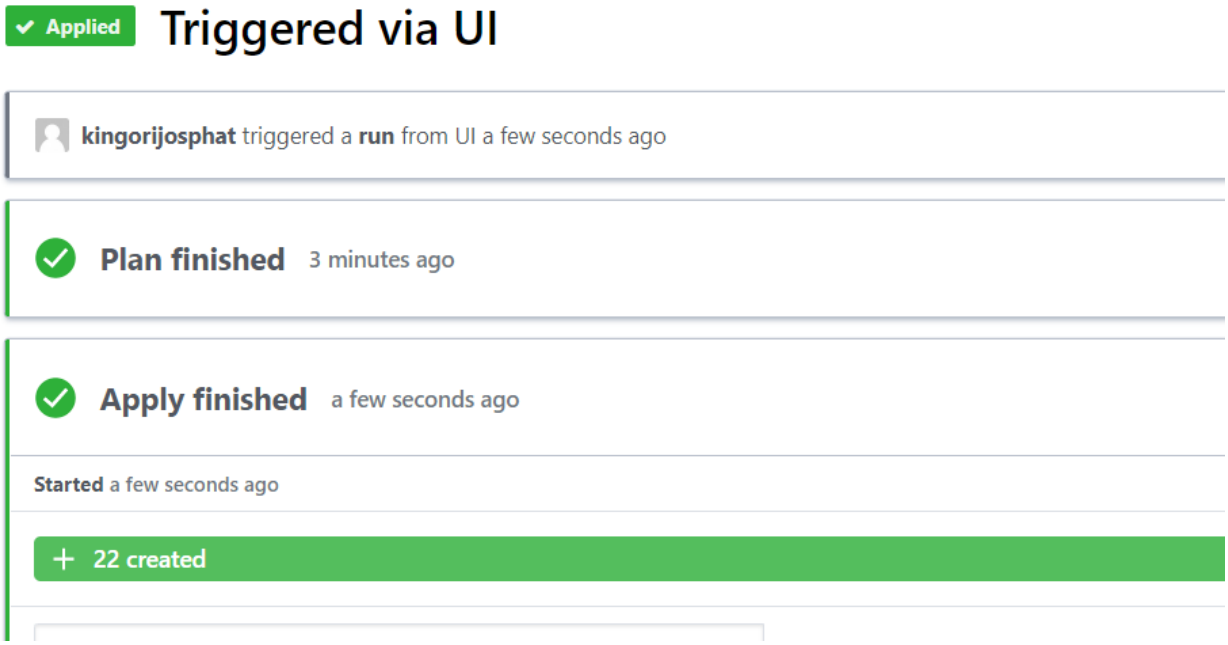
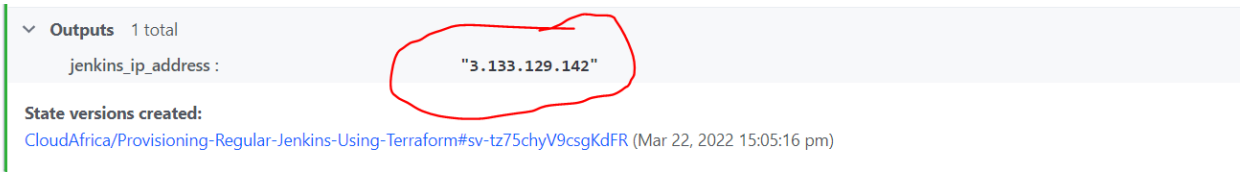
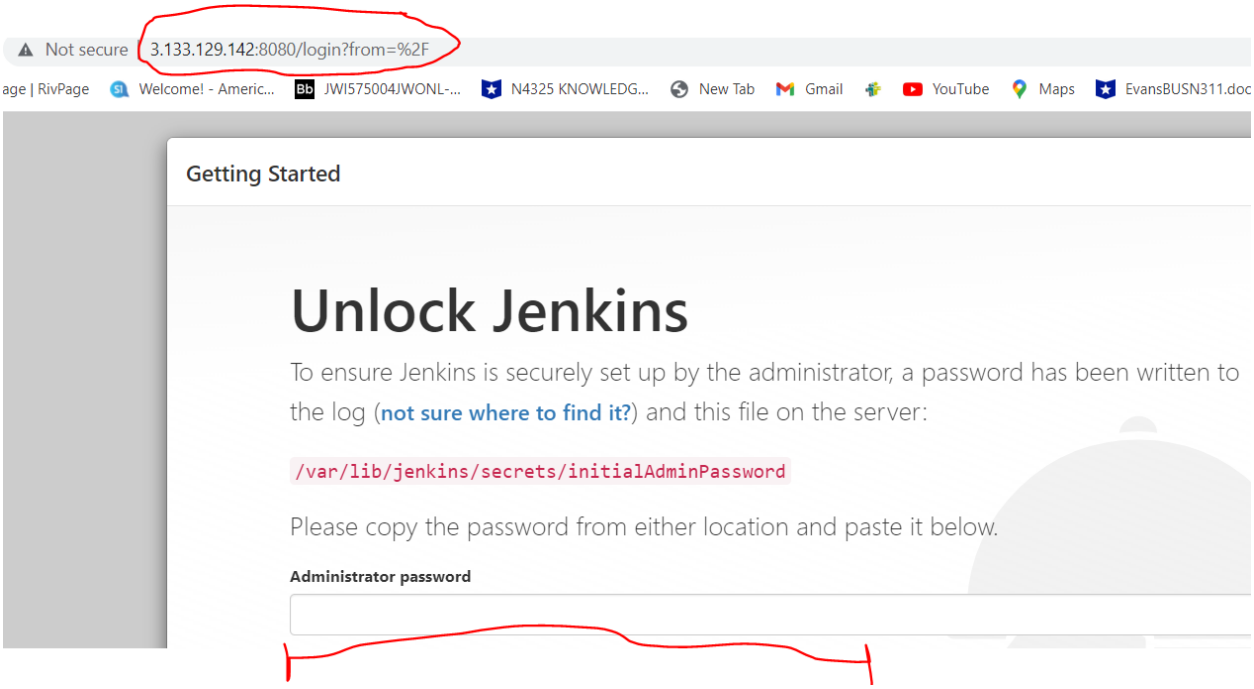
Provision war file Jenkins using terraform cloud



Get the output generated at the end of the terraform life cycle.



Paste the jenkins_ip_address on browser

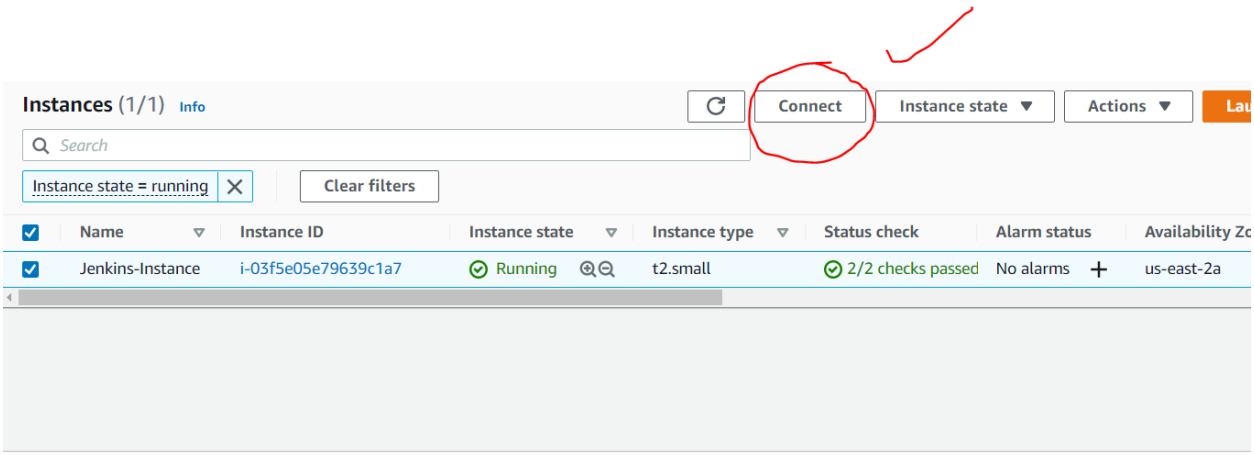


We need to go and get the admin password for the Jenkins from the Jenkins console

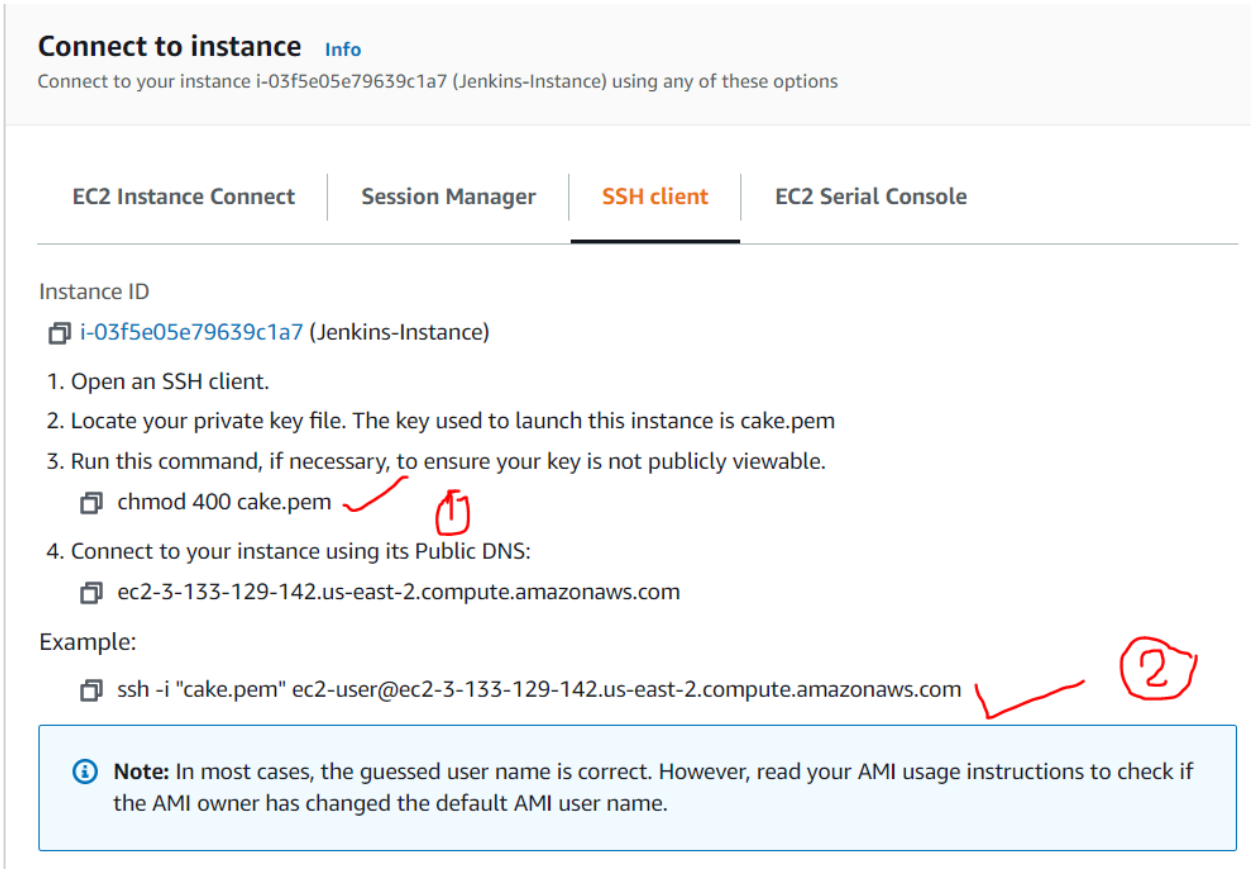
How do we do this

1. We need to ssh to the jenkins instance on aws

Go to aws console and locate your jenkins instance



2. Locate connect and click on it



Run the commands in the Jenkins terminal

```
Forex@LAPTOP-4C614INS MINGW64 ~/OneDrive/Desktop/Regular _ jenkin
sing-Terraform (master)
$ chmod 400 cake.pem

Forex@LAPTOP-4C614INS MINGW64 ~/OneDrive/Desktop/Regular _ jenkin
sing-Terraform (master)
$ ssh -i "cake.pem" ec2-user@ec2-3-133-129-142.us-east-2.compute.
Last login: Tue Mar 22 20:02:48 2022 from 204.108.237.194

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-132 ~]$
```

Now you are inside Jenkins's terminal

We need to carry out the following commands to configure Jenkins's username and obtain the administrator password for the Jenkins console

1. Check whether your environment is ready

   Sudo docker -v
   Sudo terraform -v
   Sudo ansible –version
   Sudo aws –version
   Sudo java –version

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-132 ~]$ sudo docker -v
Docker version 20.10.7, build f0df350
[ec2-user@ip-10-0-1-132 ~]$ sudo terraform -v
Terraform v1.1.2
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.1.7. You can update by downloading from https://www.terraform.io/downloads.html
```

2. Check the users in the system

```
tcpdump:x:72:72::/:/sbin/nologin
ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash
nginx:x:995:993:Nginx web server:/var/lib/nginx:/sbin/nologin
jenkins:x:994:990:Jenkins Automation Server:/var/lib/jenkins:/bin/false
```

   Jenkins come with bin/false shell
   We need to change this to /bin/bash

3. Sudo usermod –shell /bin/bash jenkins

```
[ec2-user@ip-10-0-1-132 ~]$ sudo usermod --shell /bin/bash jenkins
```

4. Rerun the sudo cat /etc/passwd

```
chrony:x:996:994::/var/lib/chrony:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash
nginx:x:995:993:Nginx web server:/var/lib/nginx:/sbin/nologin
jenkins:x:994:990:Jenkins Automation Server:/var/lib/jenkins:/bin/bash
```

Now jenkins has a /bin/bash shell

5. Next, we need to add jenkins user to the sudoers file so that it can have the right to run commands as administrator.
   Run the command: sudo vi /etc/sudoers

   Go to the end of the file and add the following information

```
## Allows members of the users
# %users  localhost=/sbin/shut

## Read drop-in files from /et
#includedir /etc/sudoers.d

jenkins ALL=(ALL) PASSWD: ALL
```

   Save: wq!

6. Now, let add jenkins to the docker group.

   Do some research and find out why we need to add jenkins to docker group

   Command: sudo usermod -aG docker jenkins or sudo usermod -a -G docker jenkins

```
[ec2-user@ip-10-0-1-132 ~]$ sudo usermod -a -G docker jenkins
[ec2-user@ip-10-0-1-132 ~]$
```

7. Now we can switch to jenkins user

   Run the command: sudo su – jenkins

```
[ec2-user@ip-10-0-1-132 ~]$ sudo su - jenkins
-bash-4.2$
-bash-4.2$
-bash-4.2$
-bash-4.2$
-bash-4.2$
```

Now you can run commands without using sudo

Let us try

Run: docker -v

```
-bash-4.2$ docker -v
Docker version 20.10.7, build f0df350
-bash-4.2$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
-bash-4.2$ docker pa
docker: 'pa' is not a docker command.
See 'docker --help'
-bash-4.2$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
-bash-4.2$
```

We will continue from here