**Roll No:** 2023510059 **Batch:** C

**Name:** Vivek Tiwari

**Experiment No:** 11

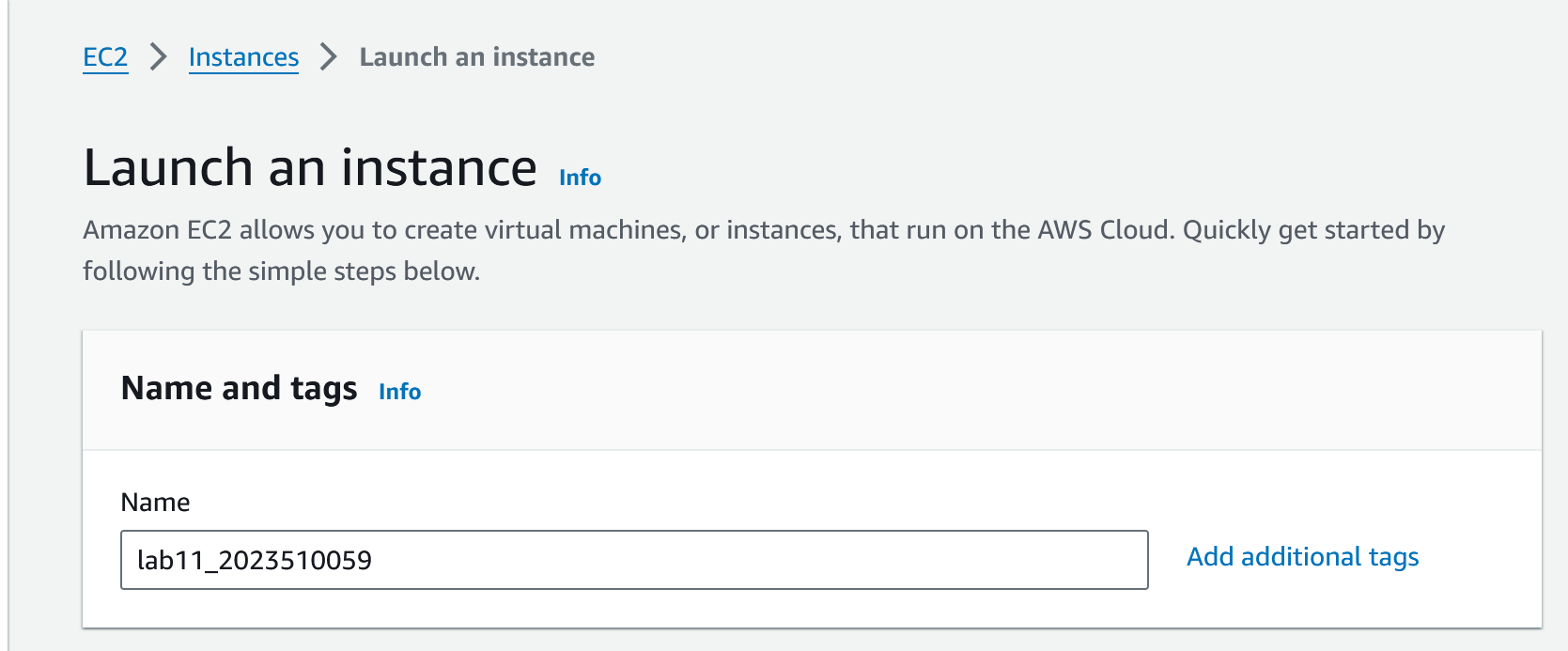
**Experiment Title:** CI/CD Tools,

**Objectives:**

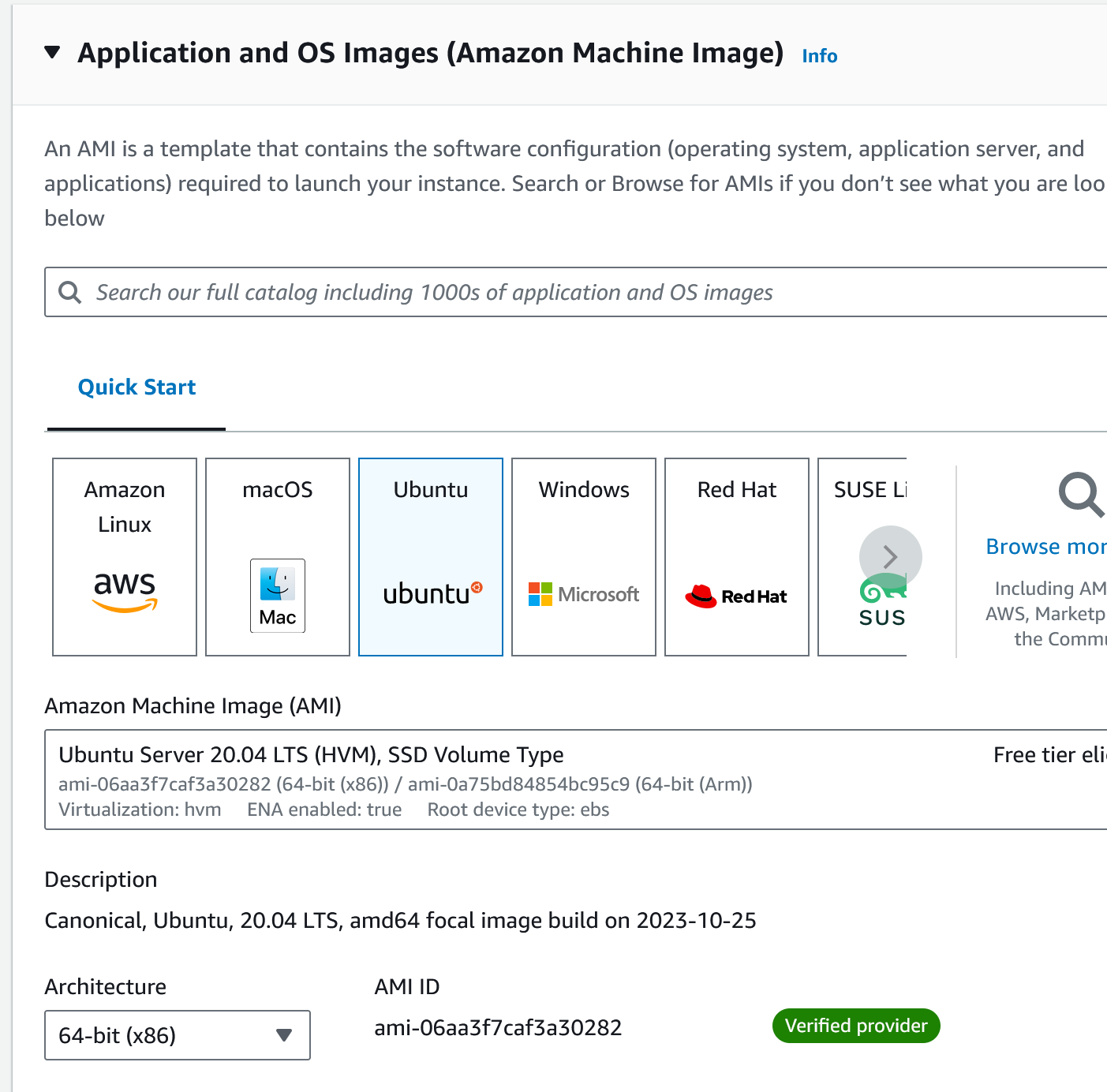
1. To create EC2 instance on AWS.
2. To deploy Jenkins service on the instance.
3. To create a pipeline in Jenkins.

**Steps:**

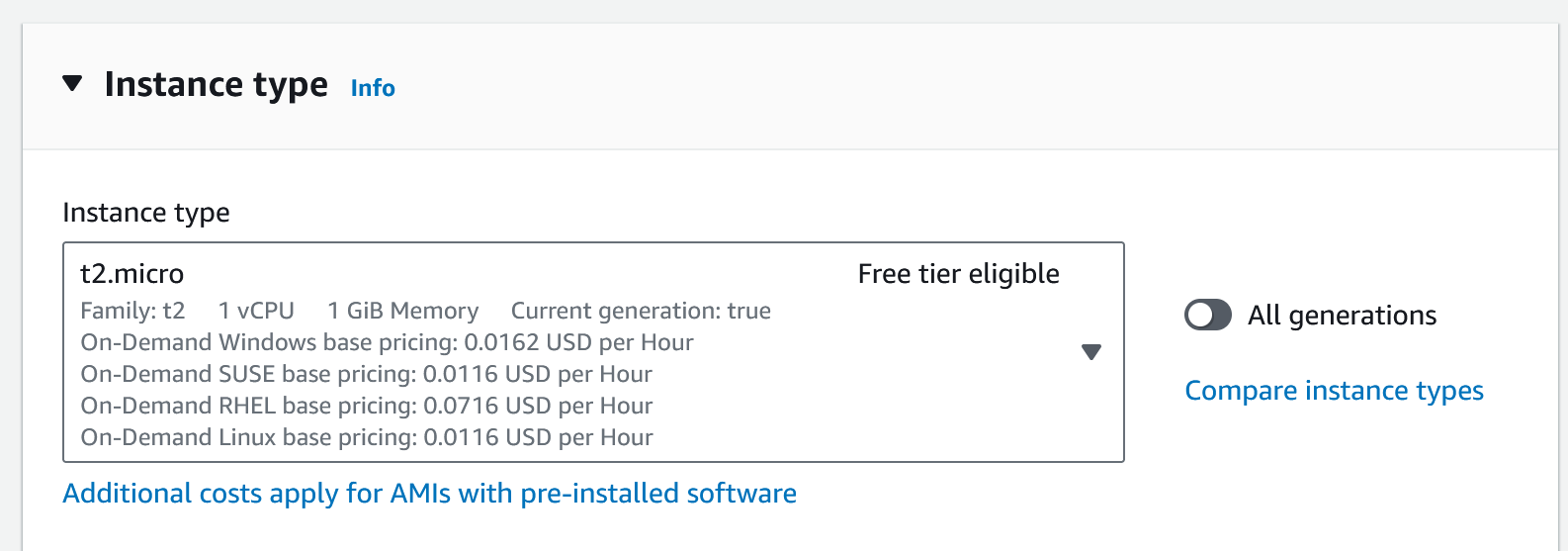
1. Search for EC2 on AWS and launch an instance.



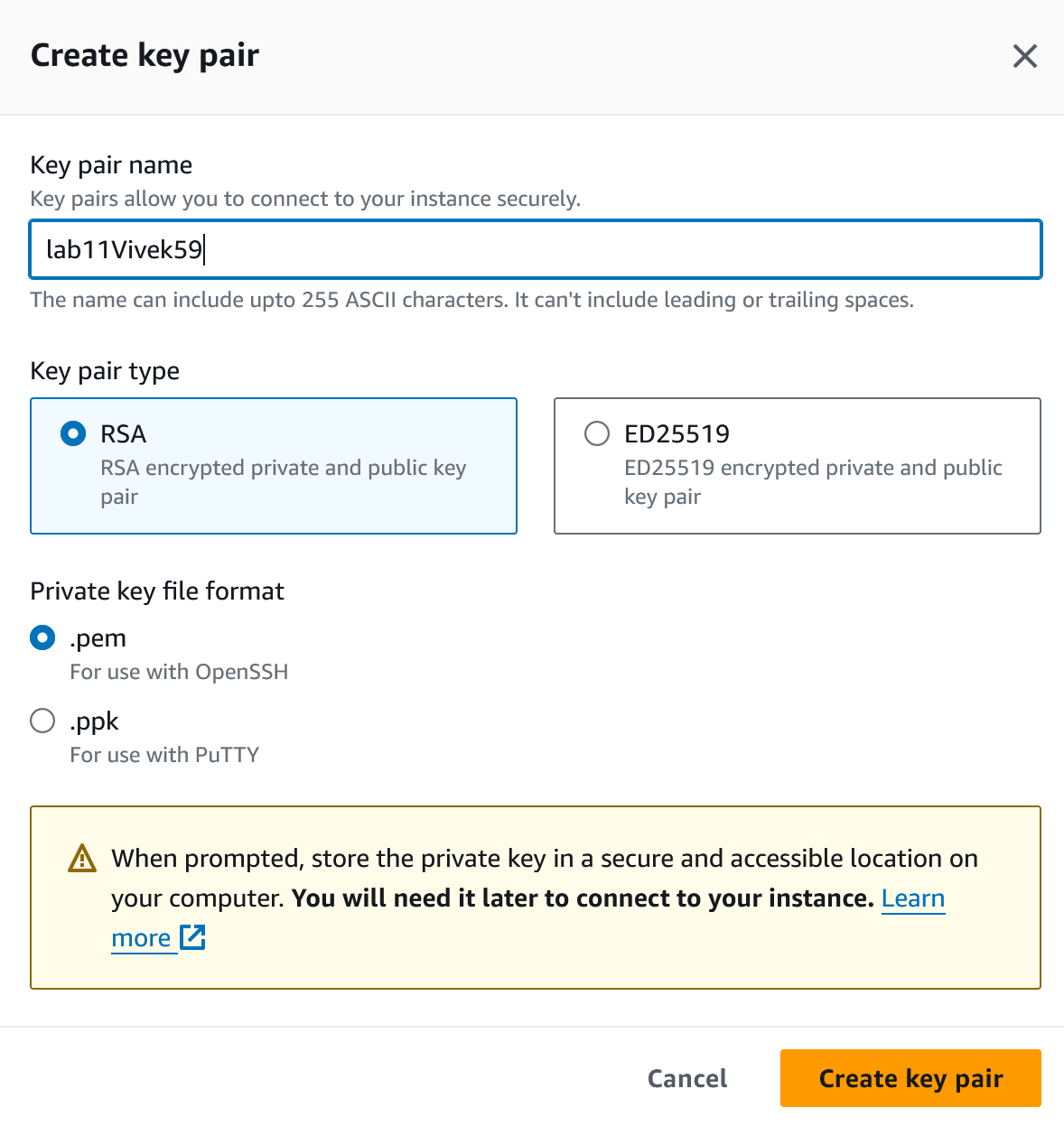
1. Select OS of your choice:

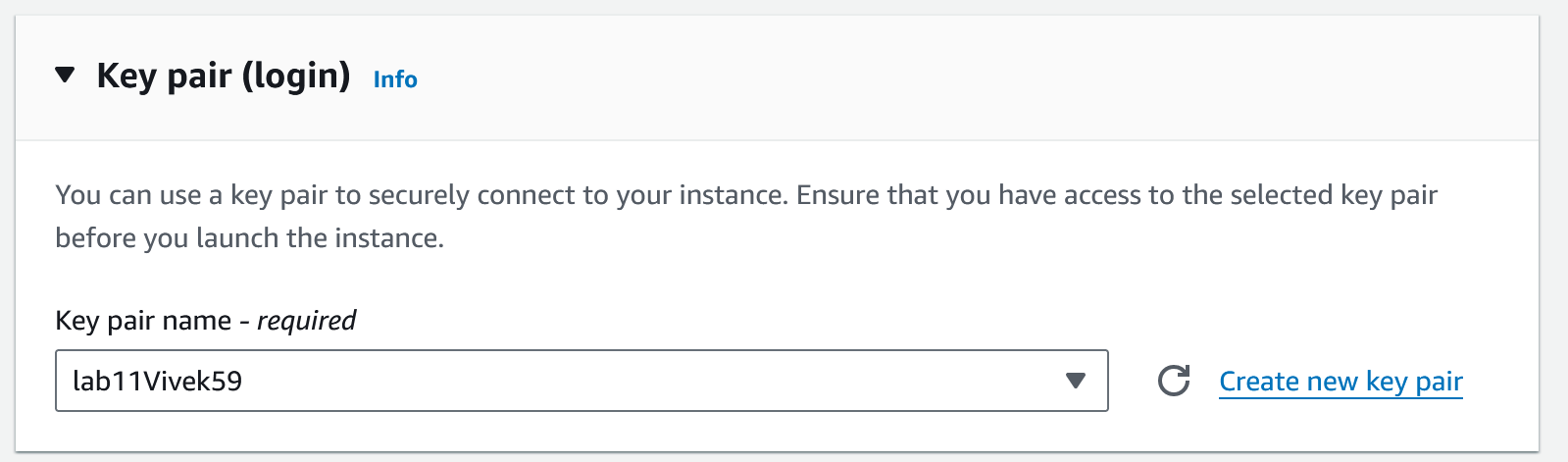


1. Confirm Instance type:

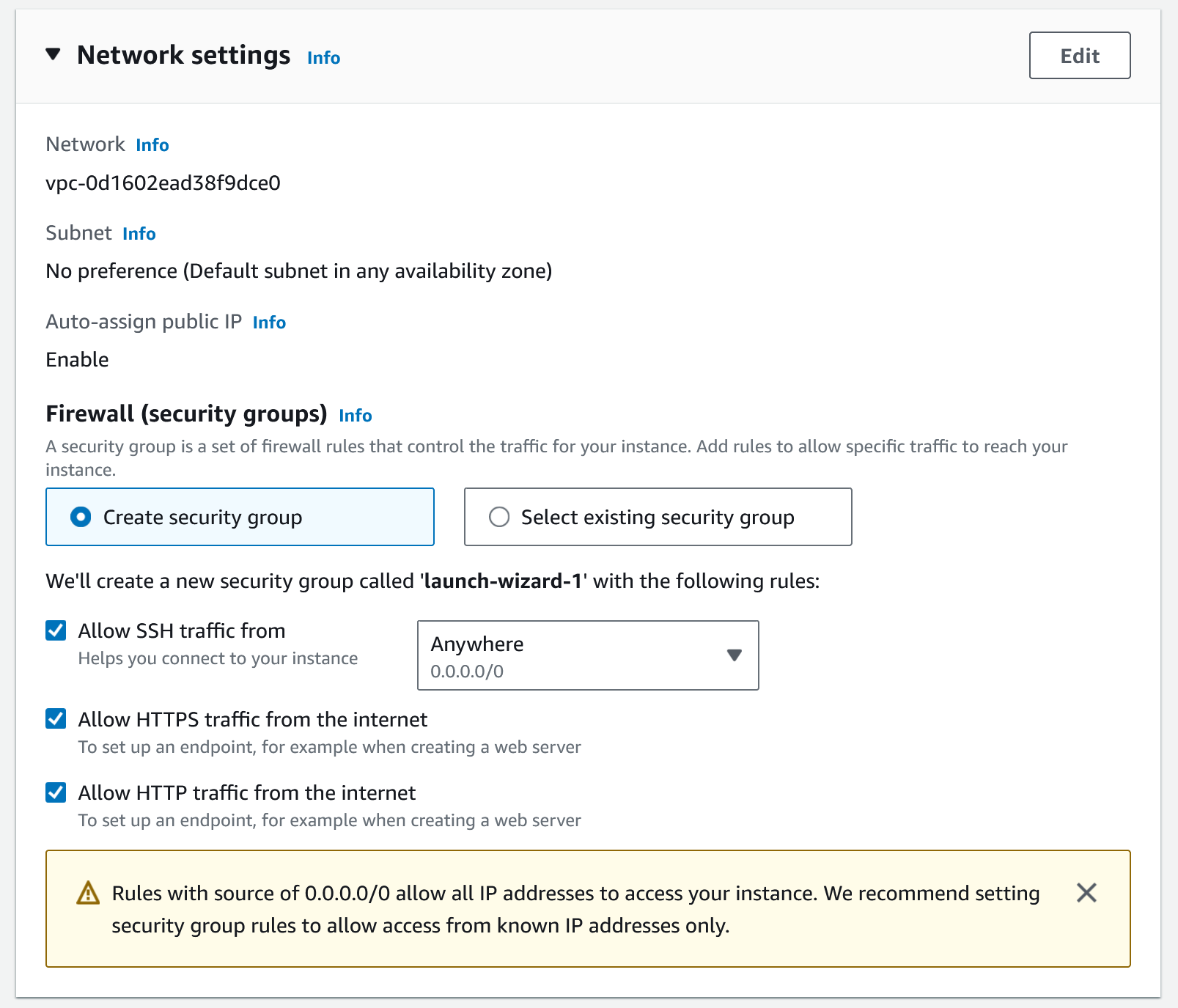


1. Create a key pair and save the downloaded file to a organised folder:

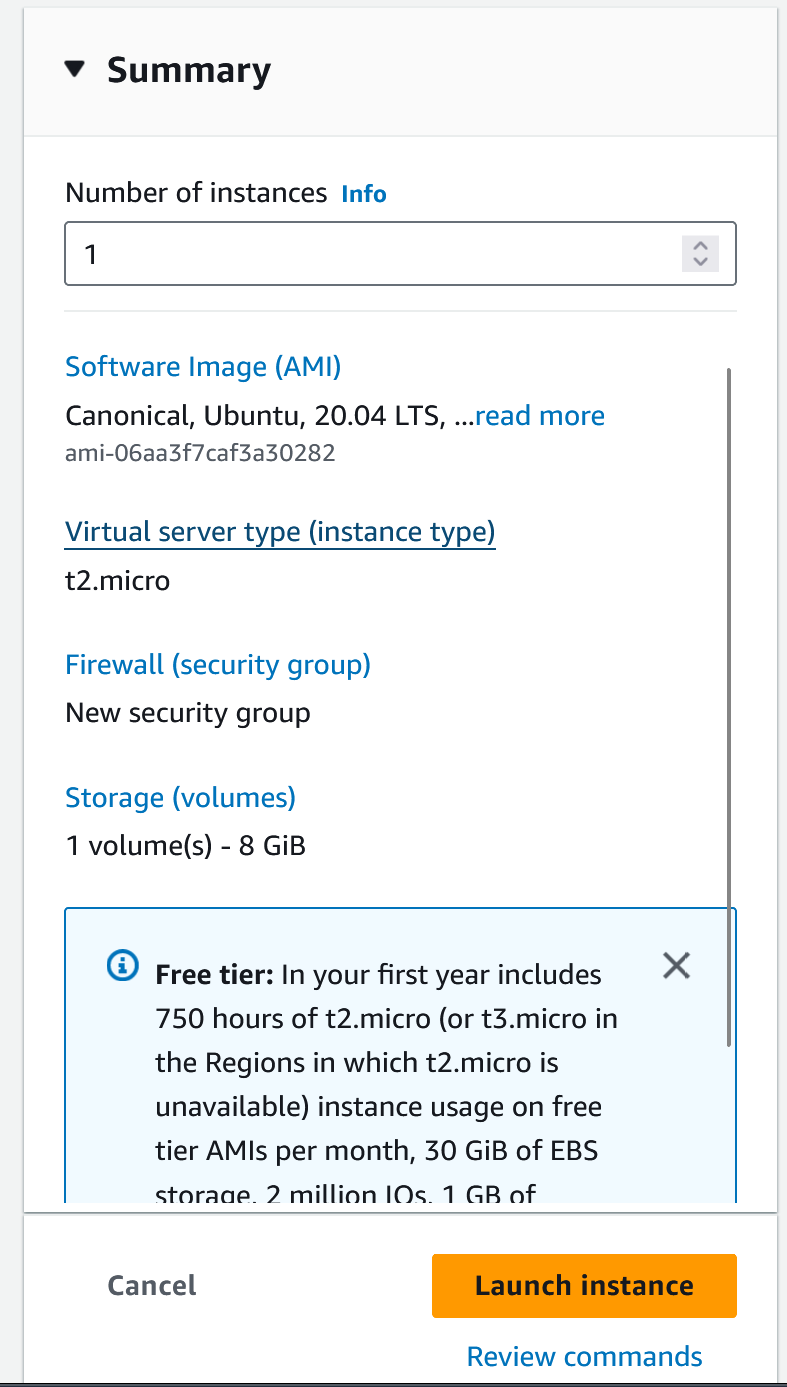




1. Enable all network settings:



1. Confirm your choices, so as not to get billed unnecessarily:



**API Calls to the instance:**

{

"MaxCount": 1,

"MinCount": 1,

"ImageId": "ami-06aa3f7caf3a30282",

"InstanceType": "t2.micro",

"KeyName": "lab11Vivek59",

"EbsOptimized": false,

"NetworkInterfaces": [

{

"AssociatePublicIpAddress": true,

"DeviceIndex": 0,

"Groups": [

"<groupId of the new security group created below>"

]

}

],

"TagSpecifications": [

{

"ResourceType": "instance",

"Tags": [

{

"Key": "Name",

"Value": "lab11\_2023510059"

}

]

}

],

"MetadataOptions": {

"HttpTokens": "required",

"HttpEndpoint": "enabled",

"HttpPutResponseHopLimit": 2

},

"PrivateDnsNameOptions": {

"HostnameType": "ip-name",

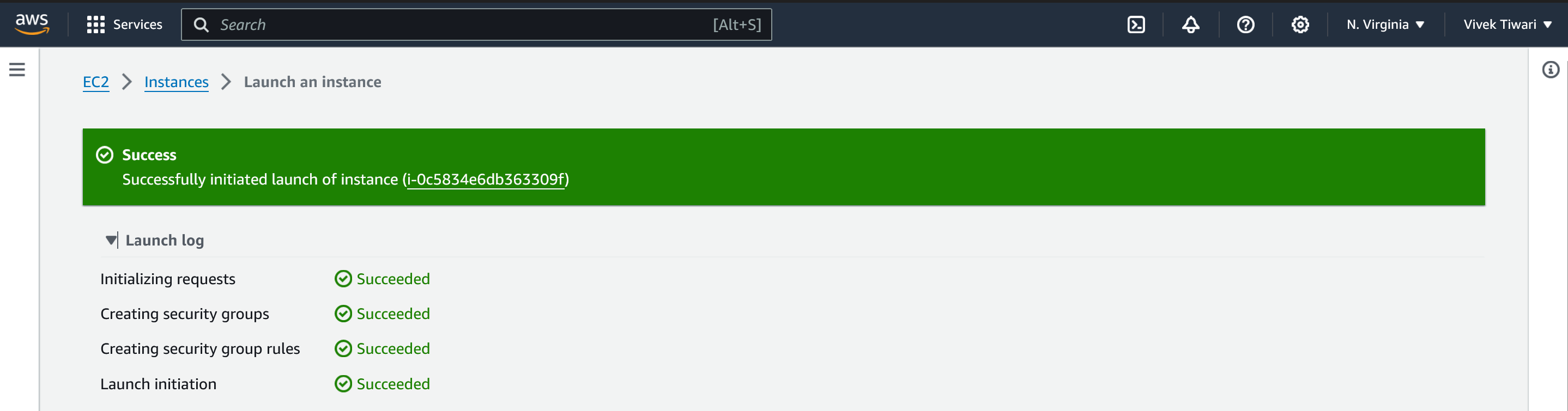
"EnableResourceNameDnsARecord": true,

"EnableResourceNameDnsAAAARecord": false

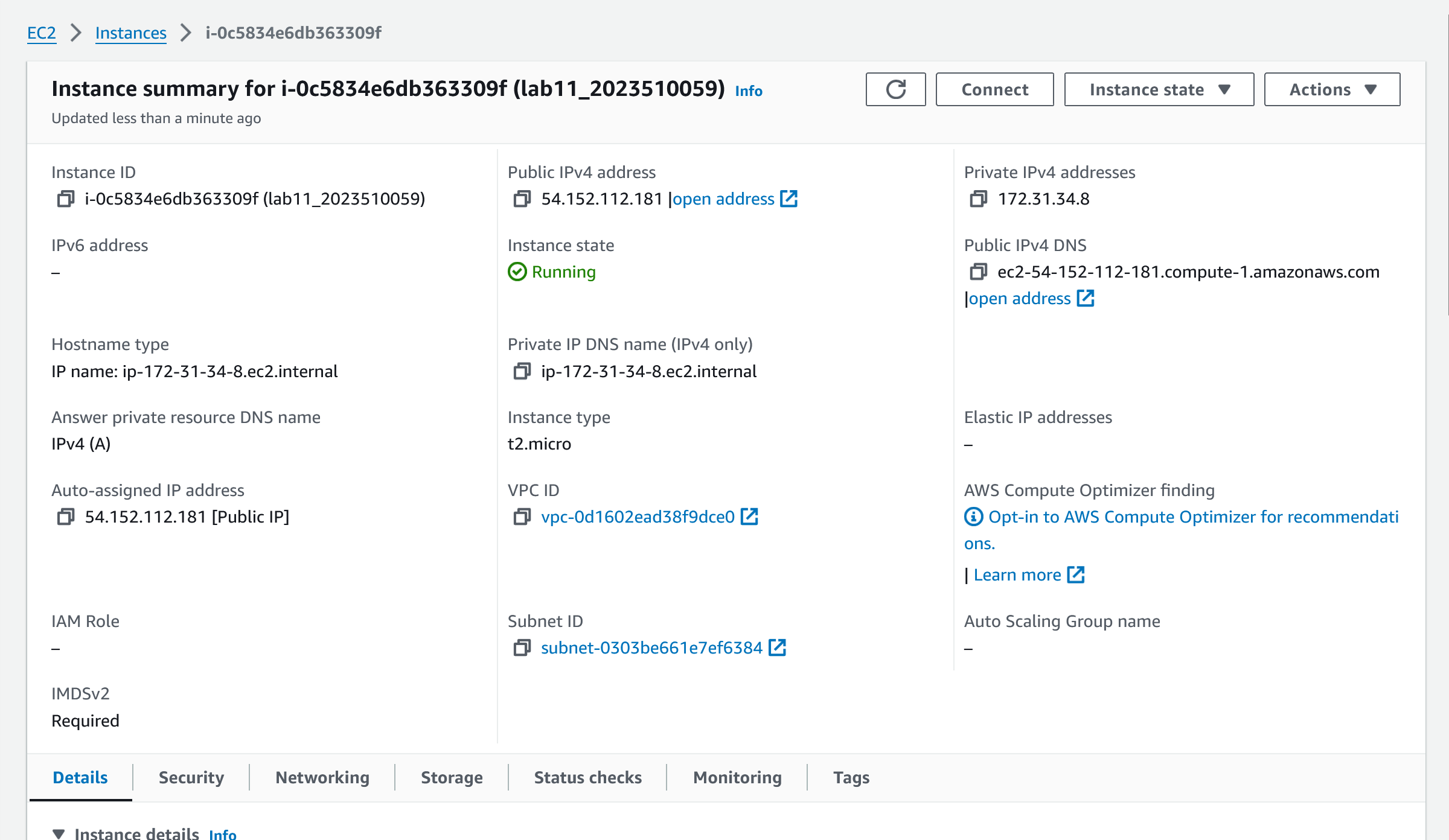
}

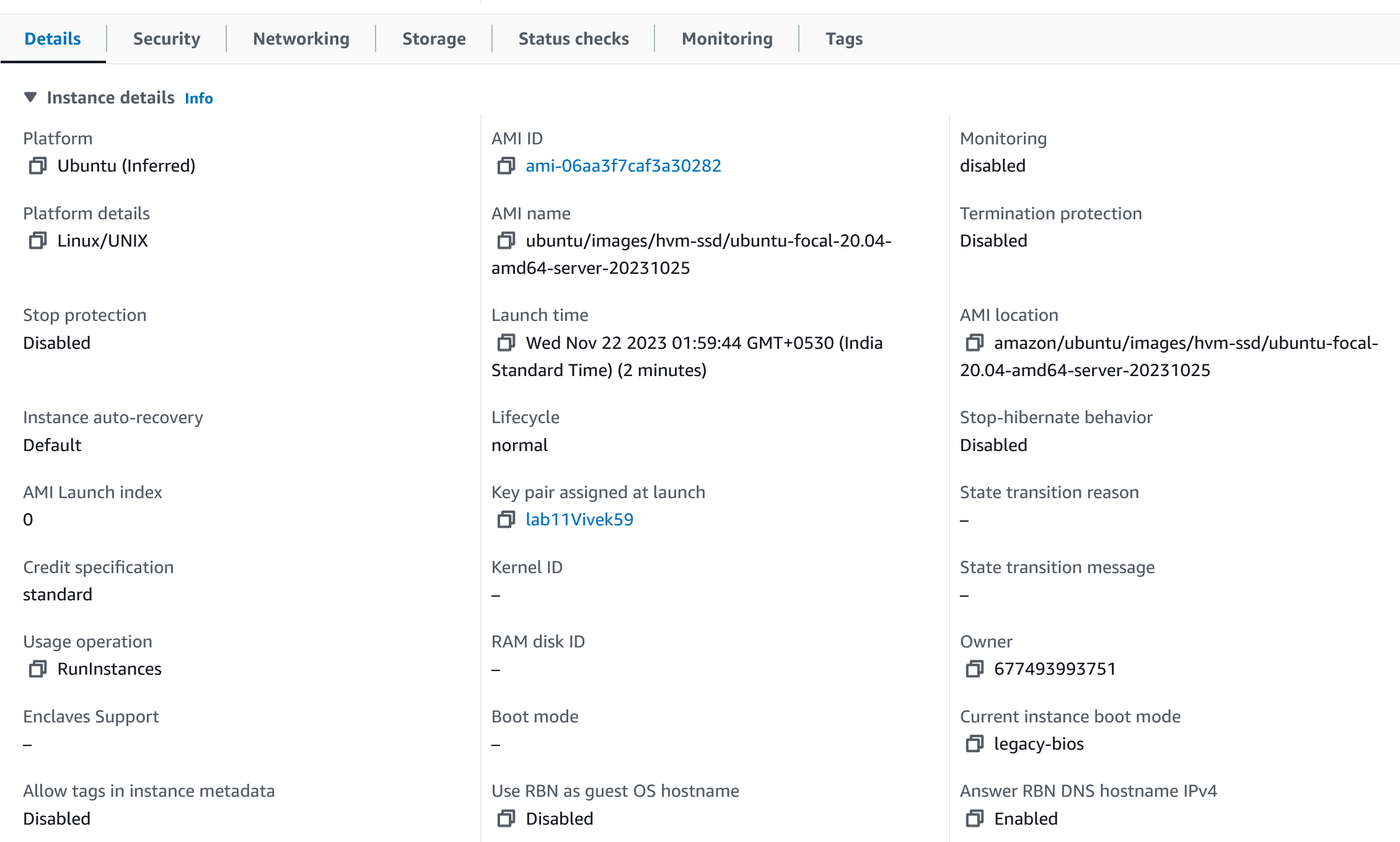
}

1. You’ve successfully created an EC2 instance:

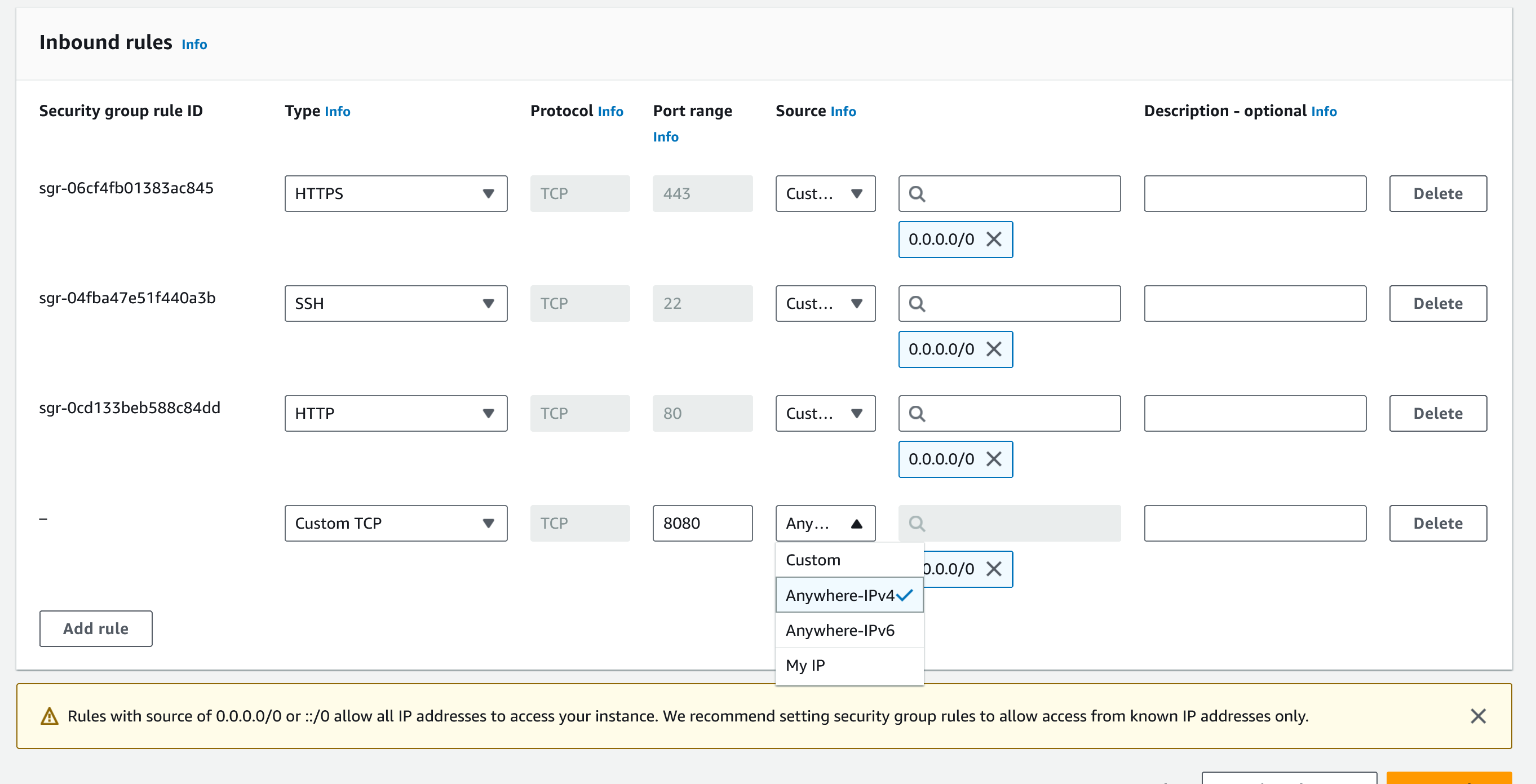


1. Check all the details:





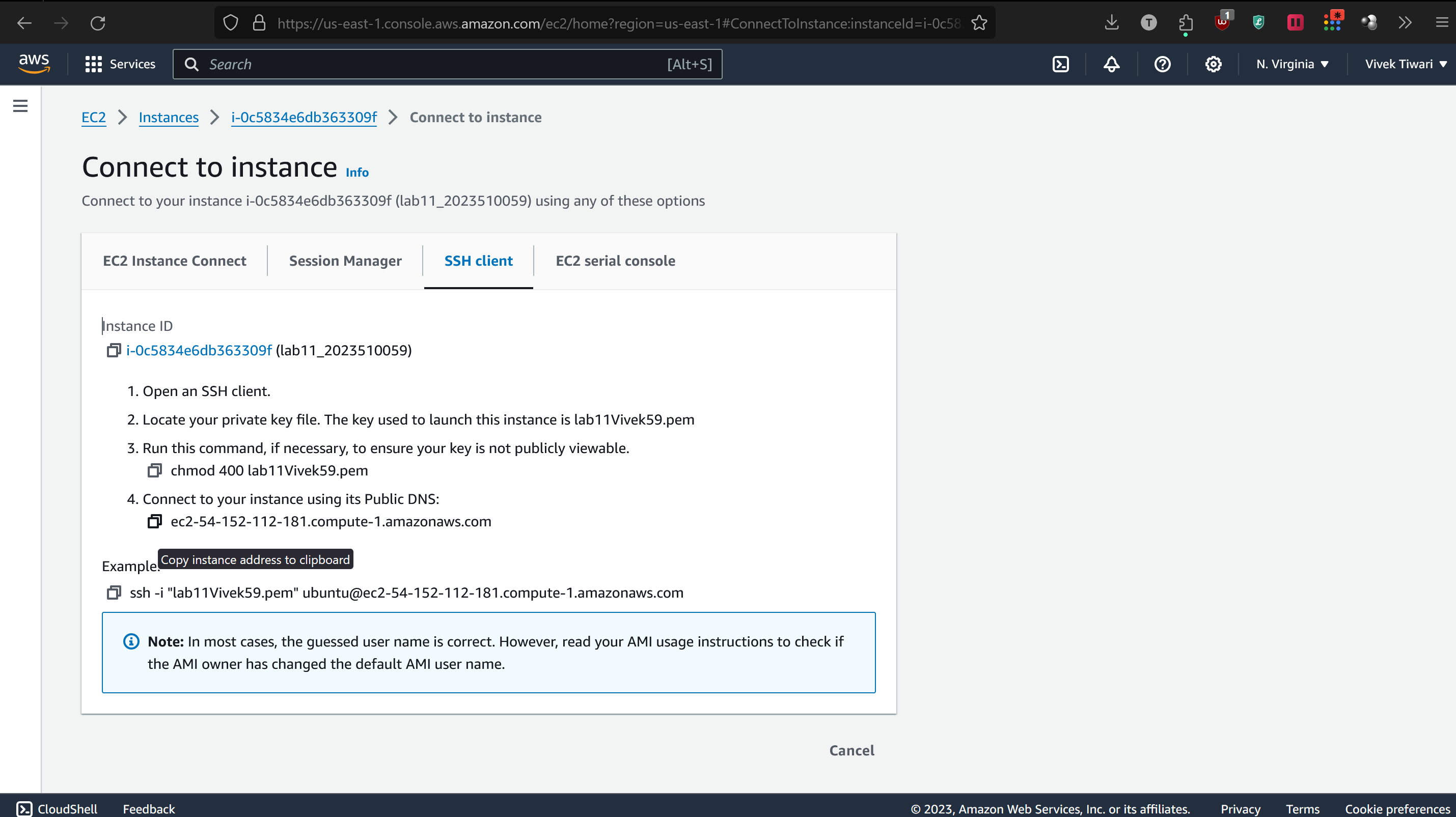
1. Now create a custom TCP protocol using any port number you like, 8080 has been used here:



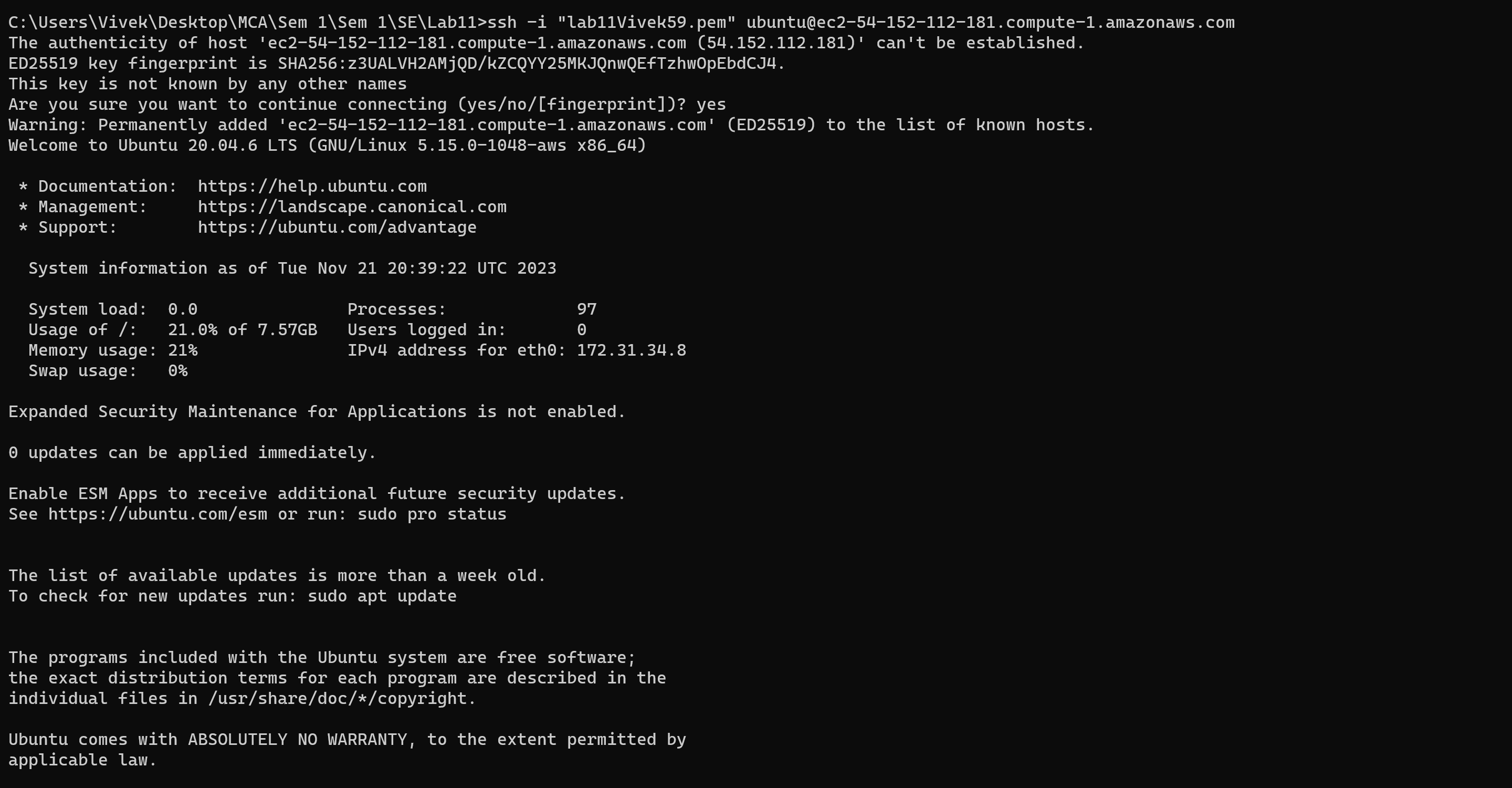
1. Now connect to the EC2 instance using CLI via SSH protocol:

To do so:

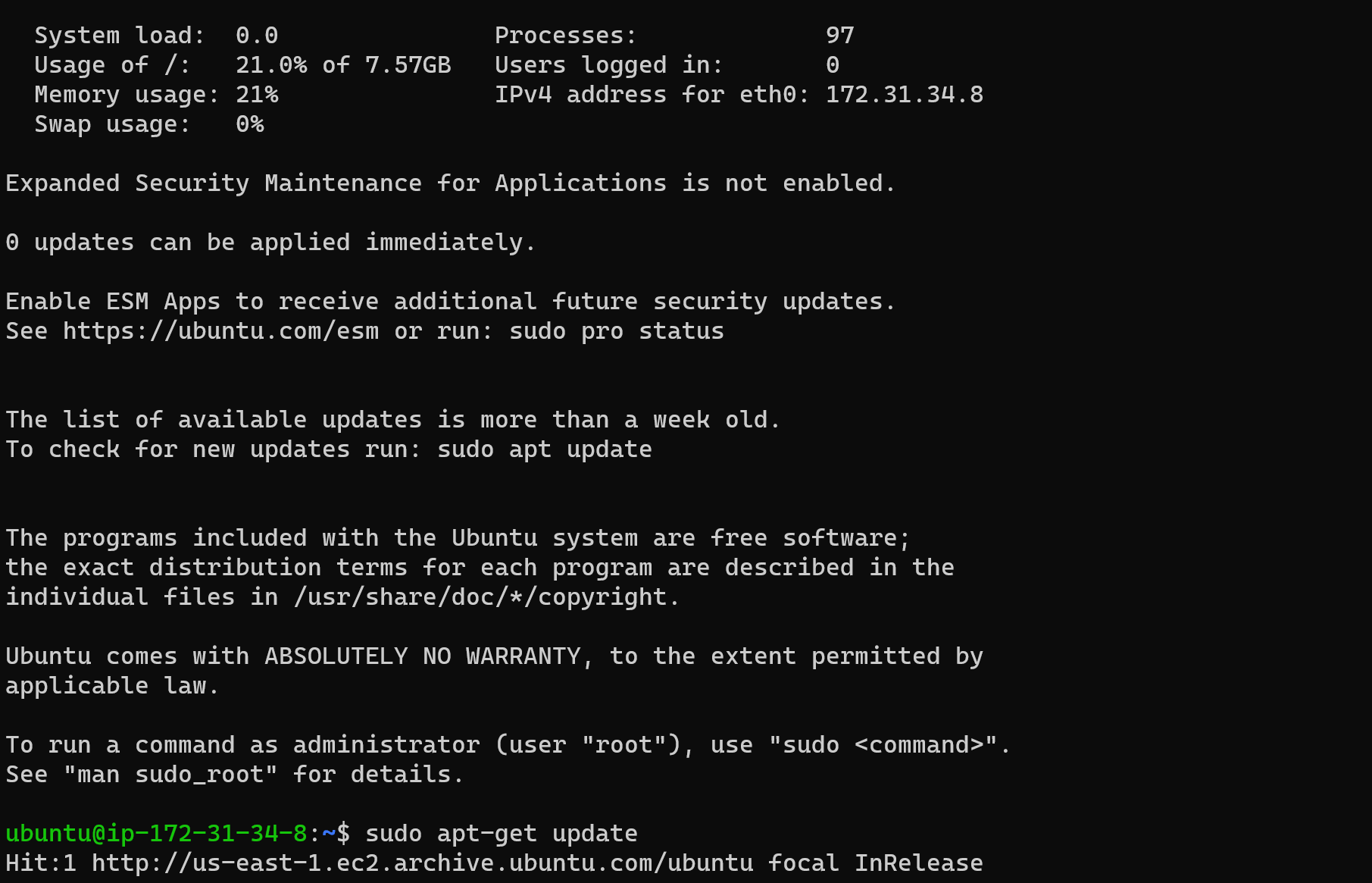
* Head over to your EC2 instance and click on connect.
* Click on the SSH tab and copy the command present there.



* Open the directory where you’ve stored the .pem file.
* Open a CLI in that directory
* Now run the copied SSH command in the CLI window.



* Run the command ‘sudo apt-get update’ to update your Linux machine.



1. Now adding Jenkins repository for ubuntu:

Command 1:

sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

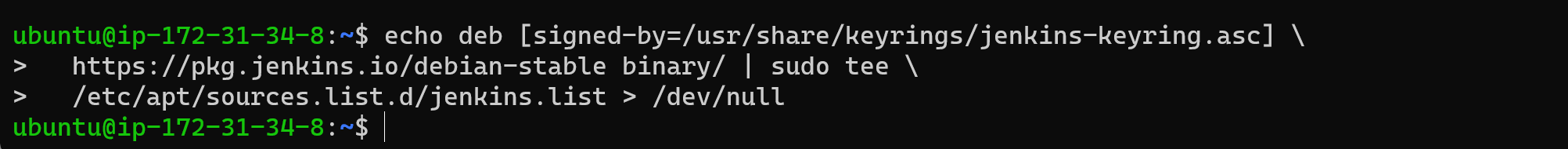


Command 2:

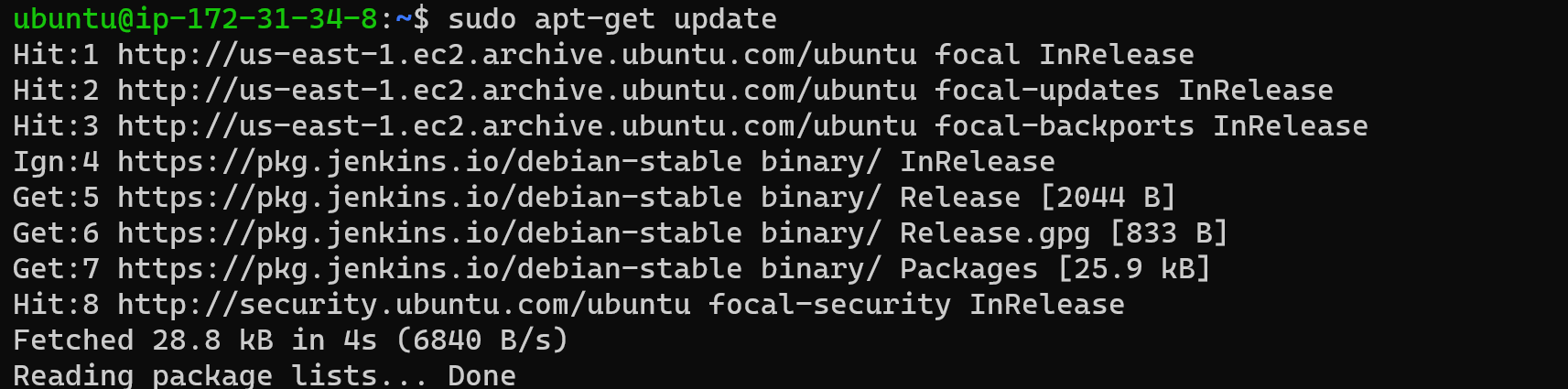
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null



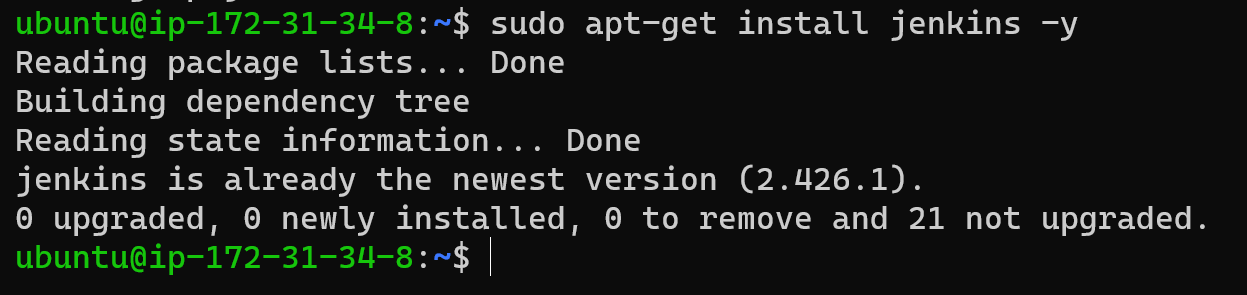
1. Now run ‘sudo apt-get update’ to update the Jenkins repository



Before installing Jenkins, install openjdk -17 using the command ‘sudo apt install fontconfig openjdk-17-jre’



1. Now install Jenkins using the command ‘sudo apt-get install Jenkins -y’

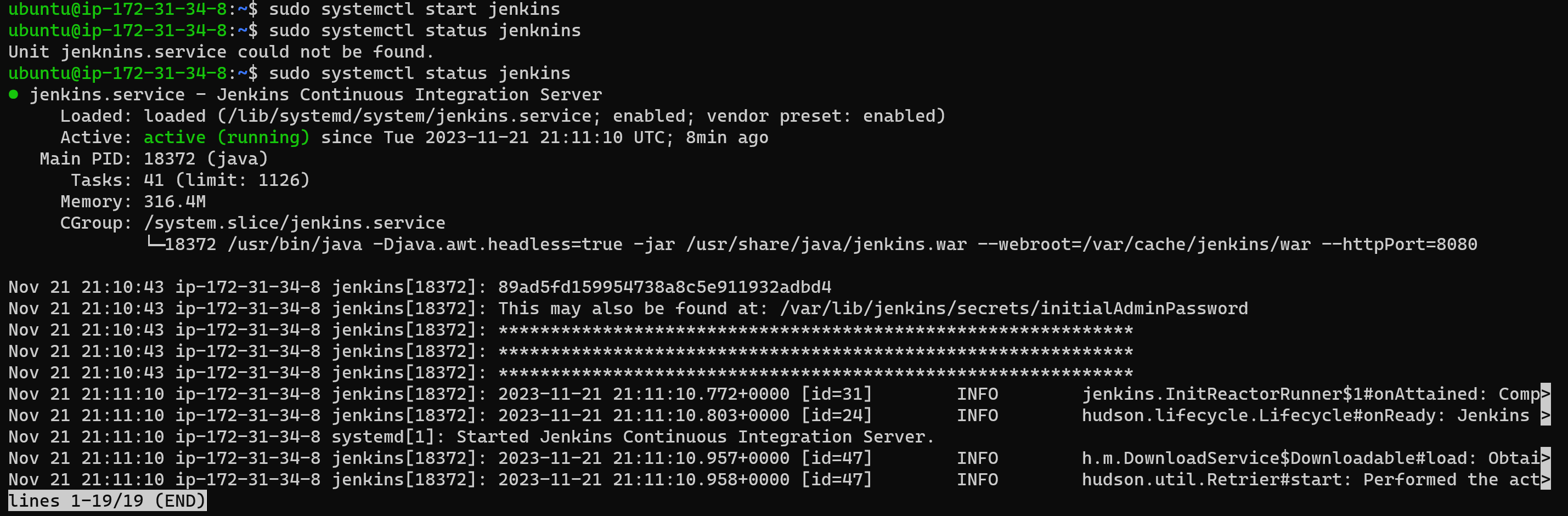


1. Now run the command ‘sudo systemctl enable jenkins’

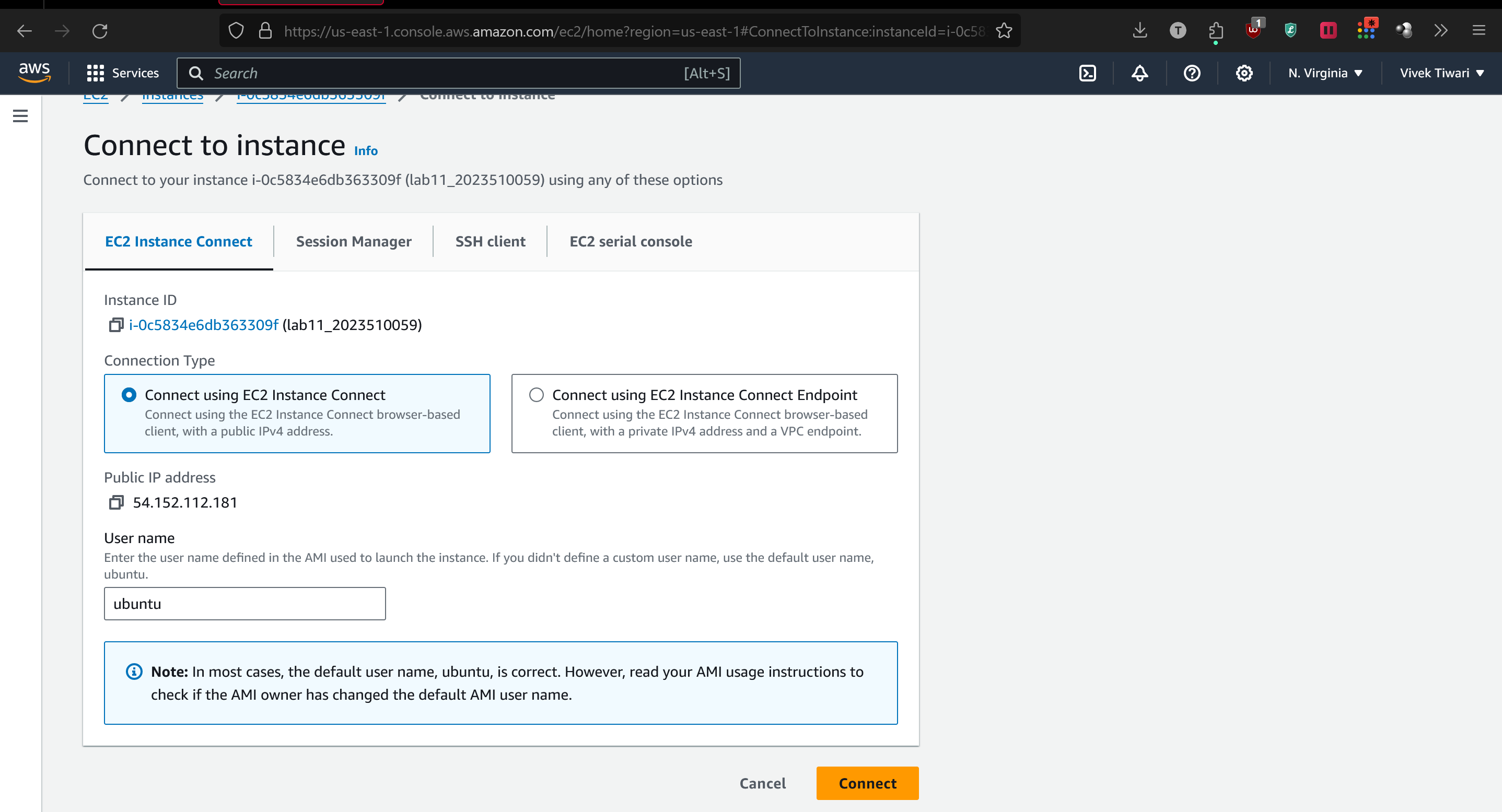


1. Start Jenkins service using the command ‘sudo systemctl start Jenkins’

To check if Jenkins is running or not, use the command ‘sudo systemctl status jenkins’

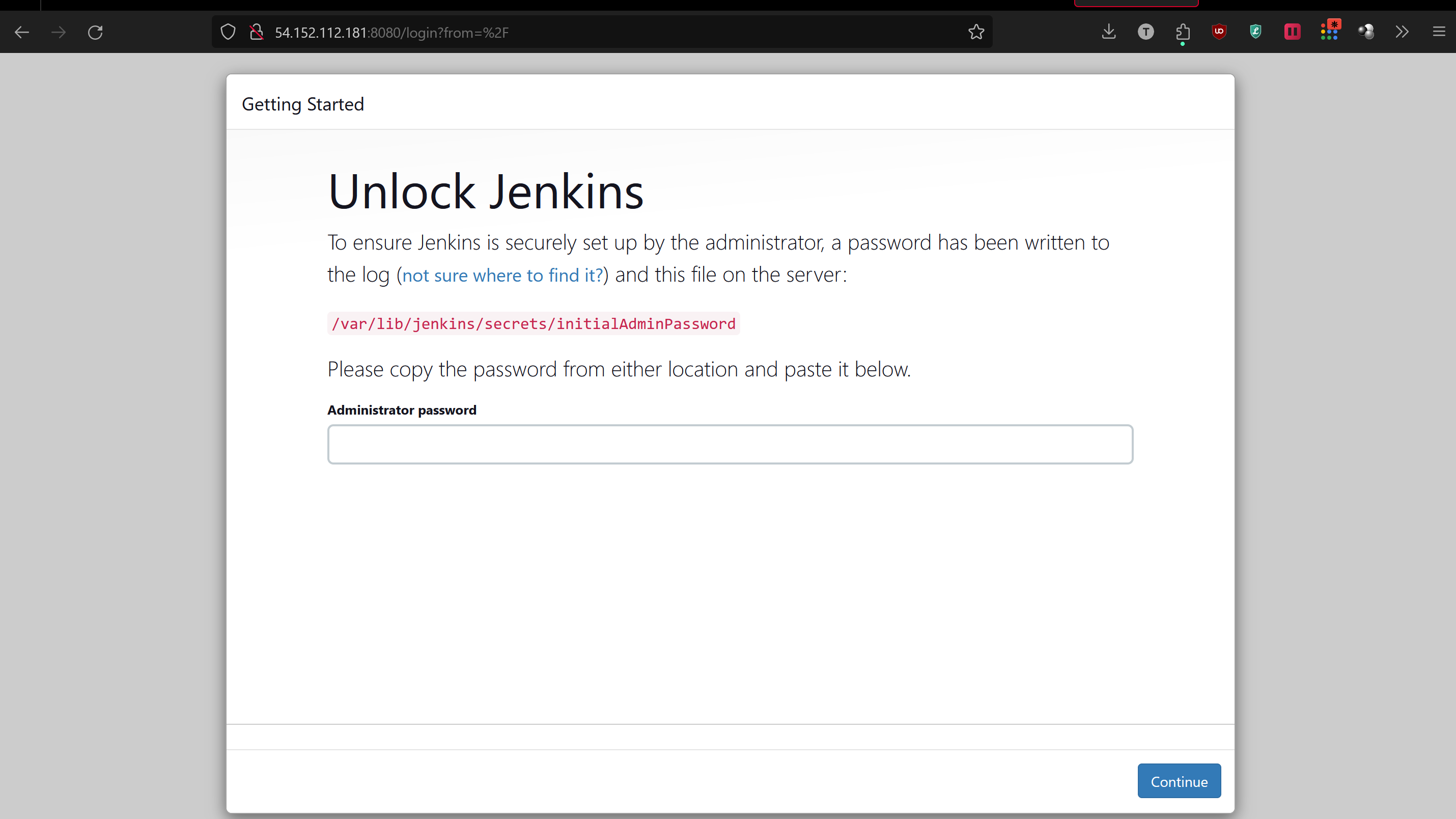


1. Now copy the IP address mentioned under the ec2 connect tab

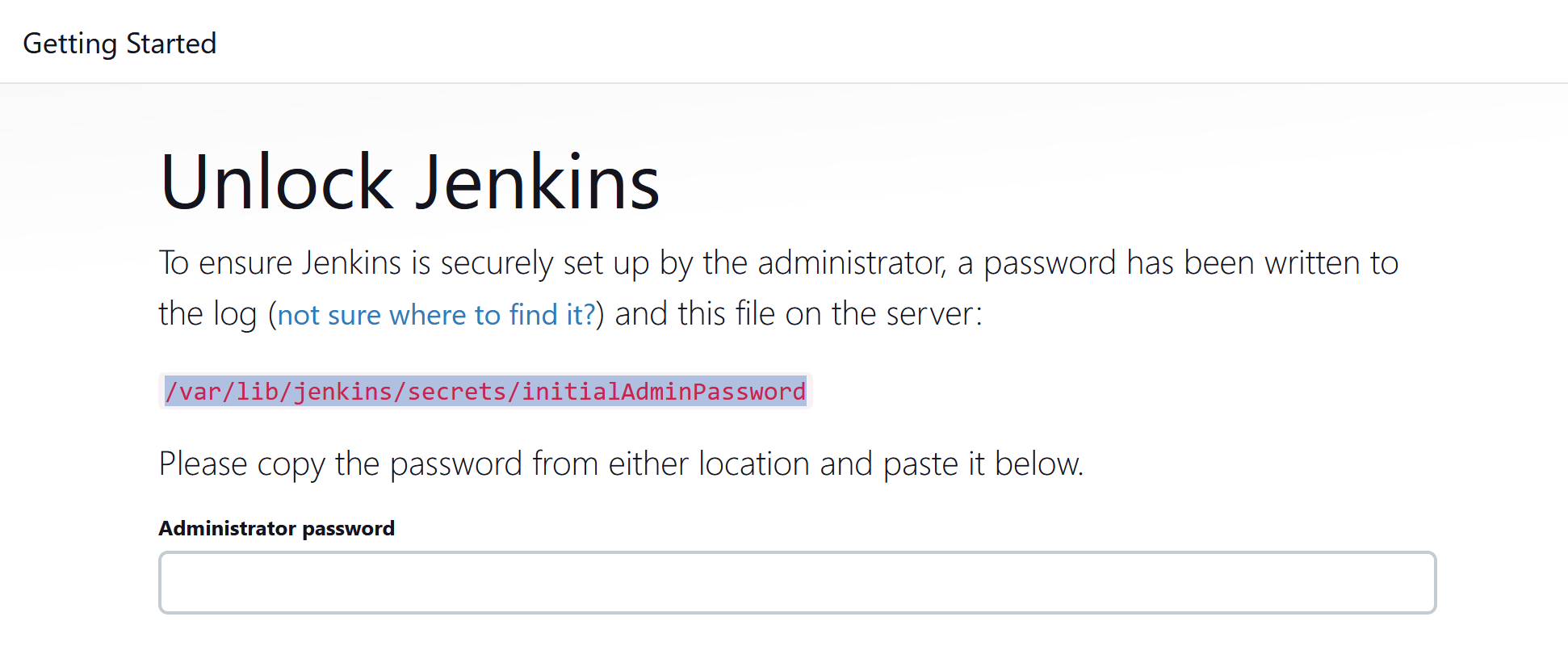


1. Append the port 8080 to the IP address as we had previously defined:

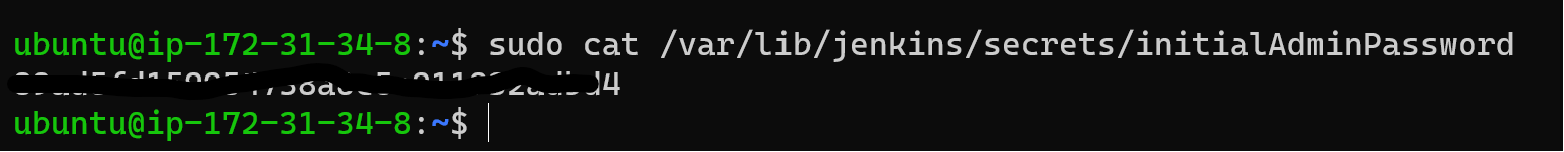
* 54.152.112.181:8080, and paste the IP address in your browser.
* The browser should load unlock Jenkins page as below.



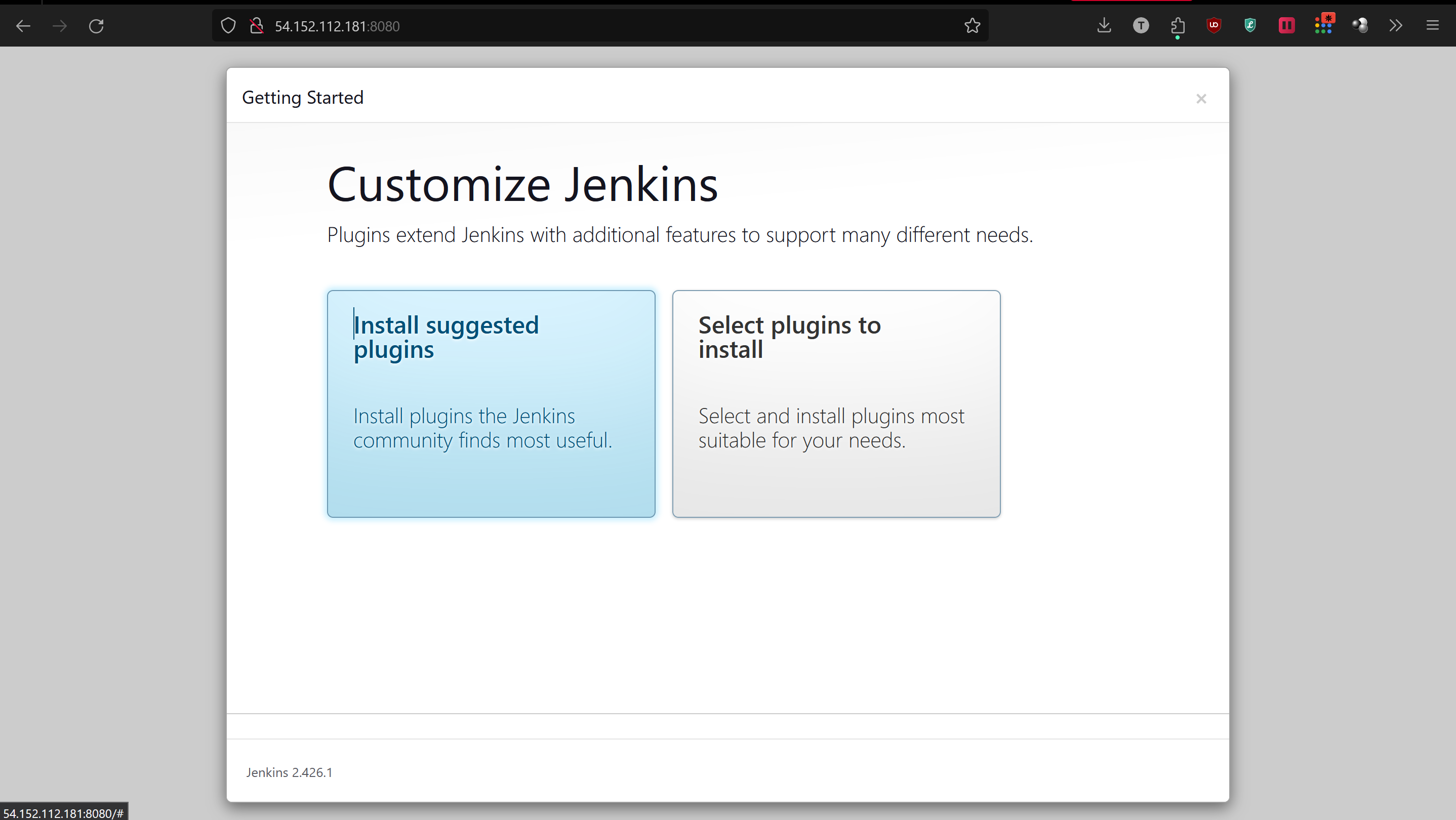
1. Copy the path:

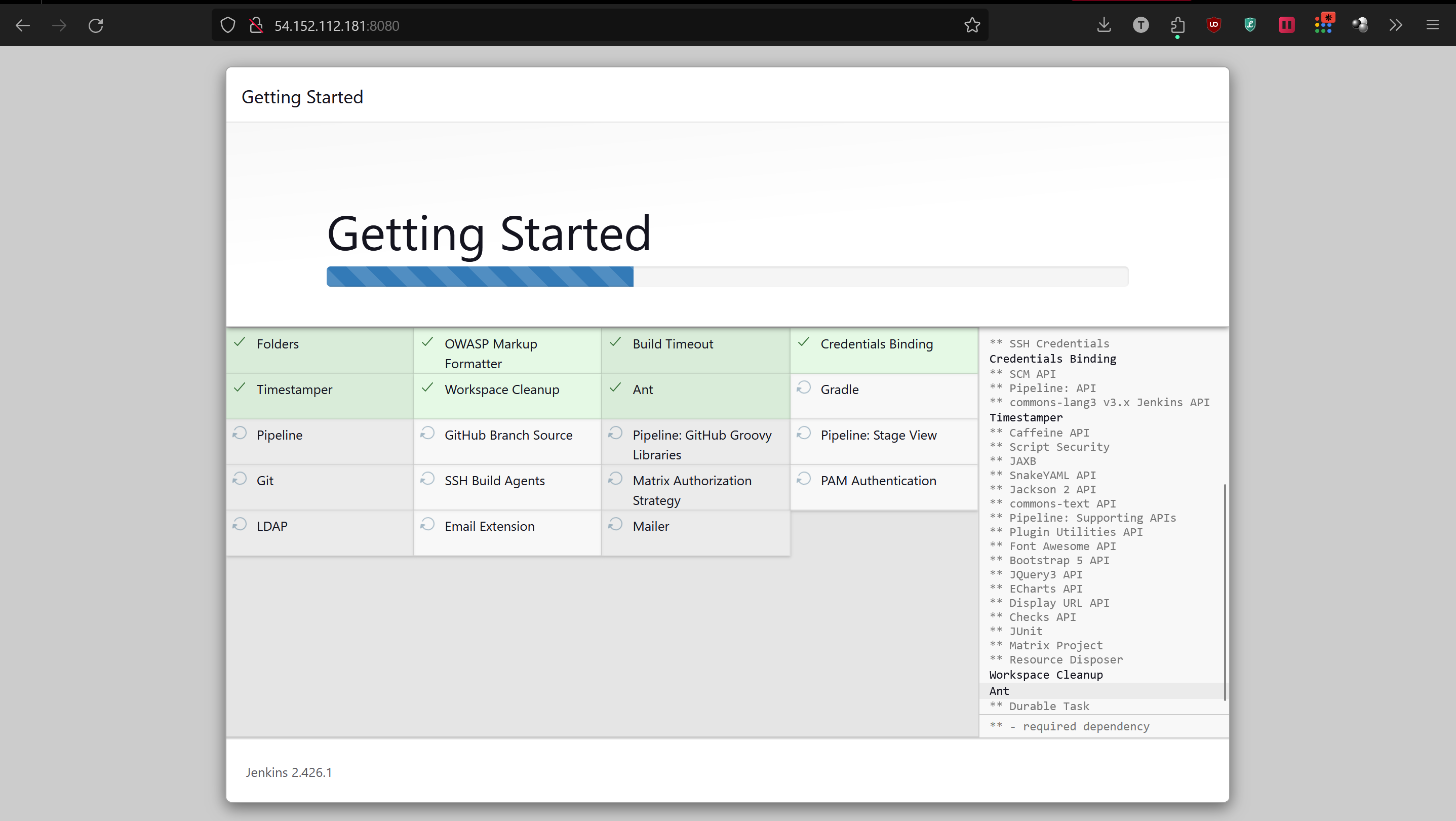


1. Run the command ‘sudo cat <path>’. This command will return a password.

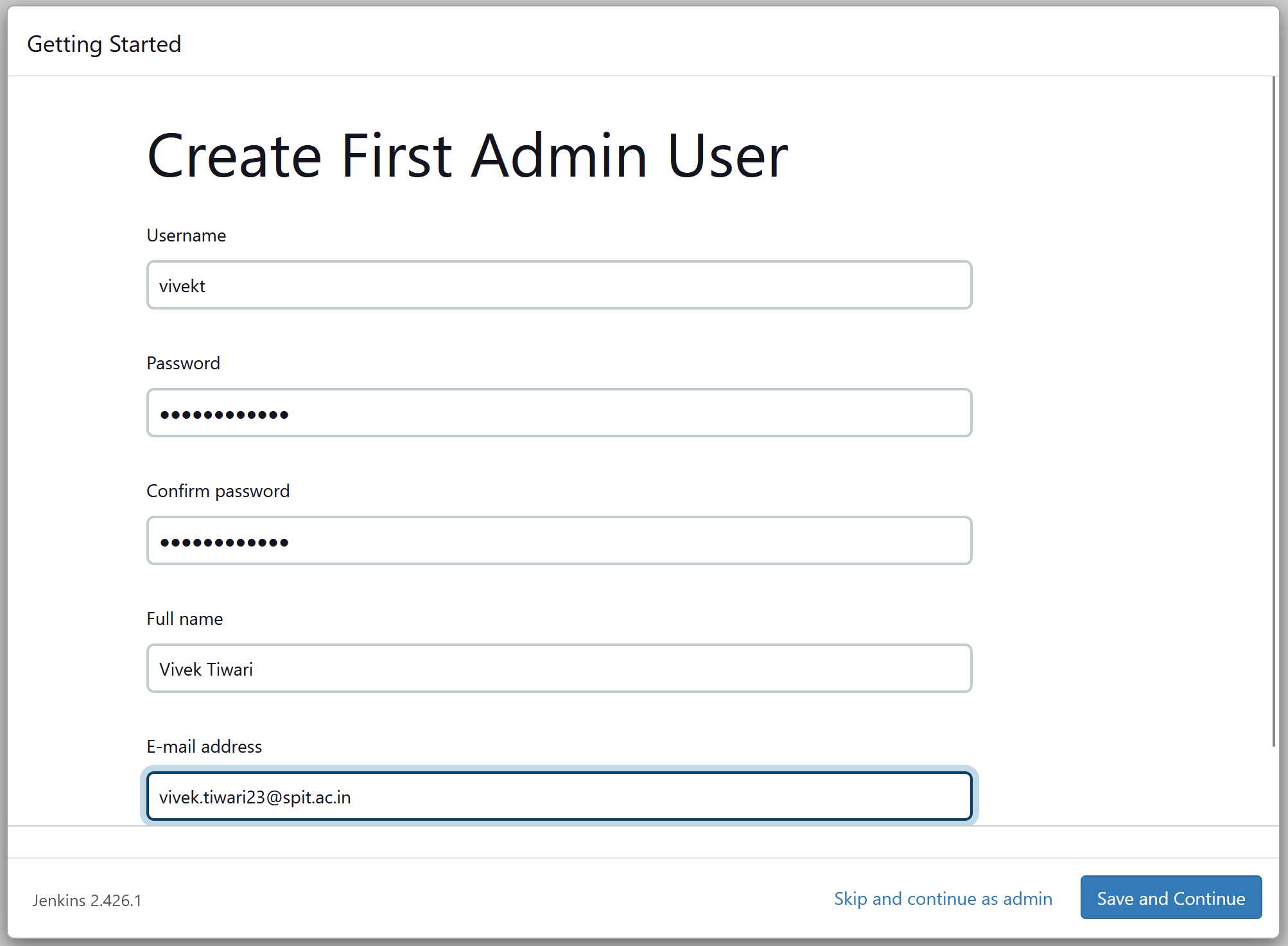


1. After entering the password, select install suggested plugins:



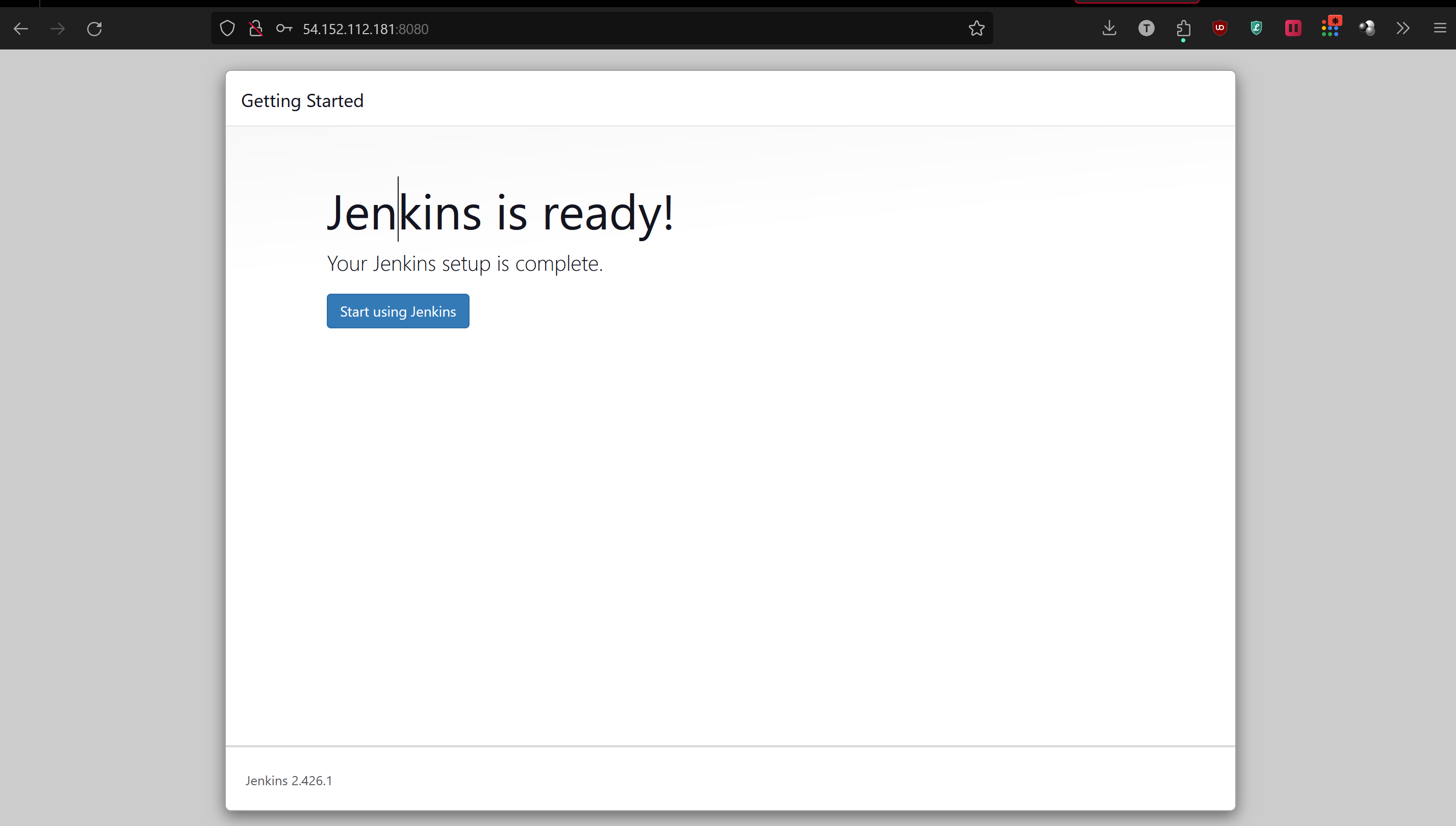


1. Now create an admin user:

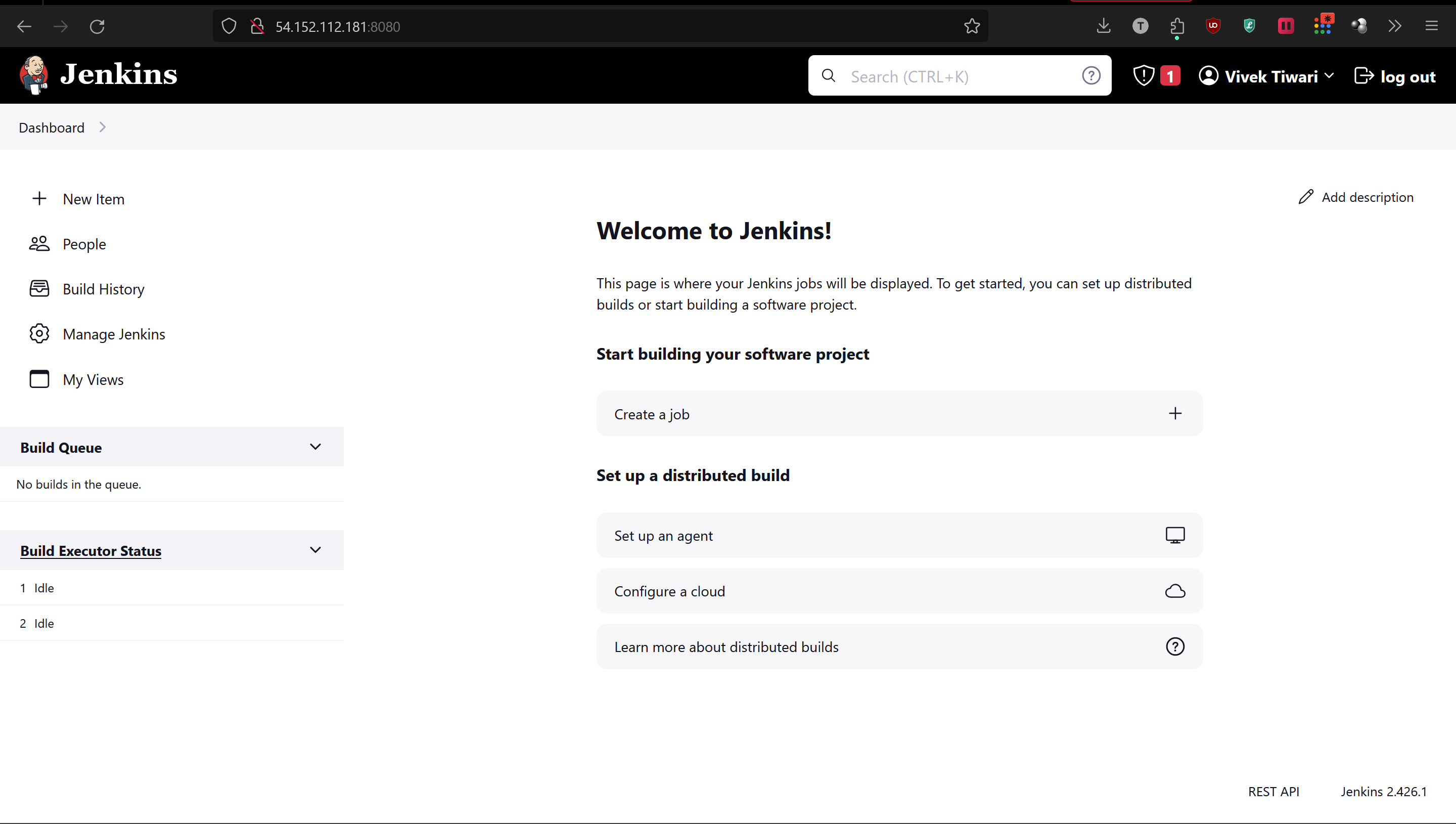


Make sure to copy your Jenkins URL on the next step after saving the user details.

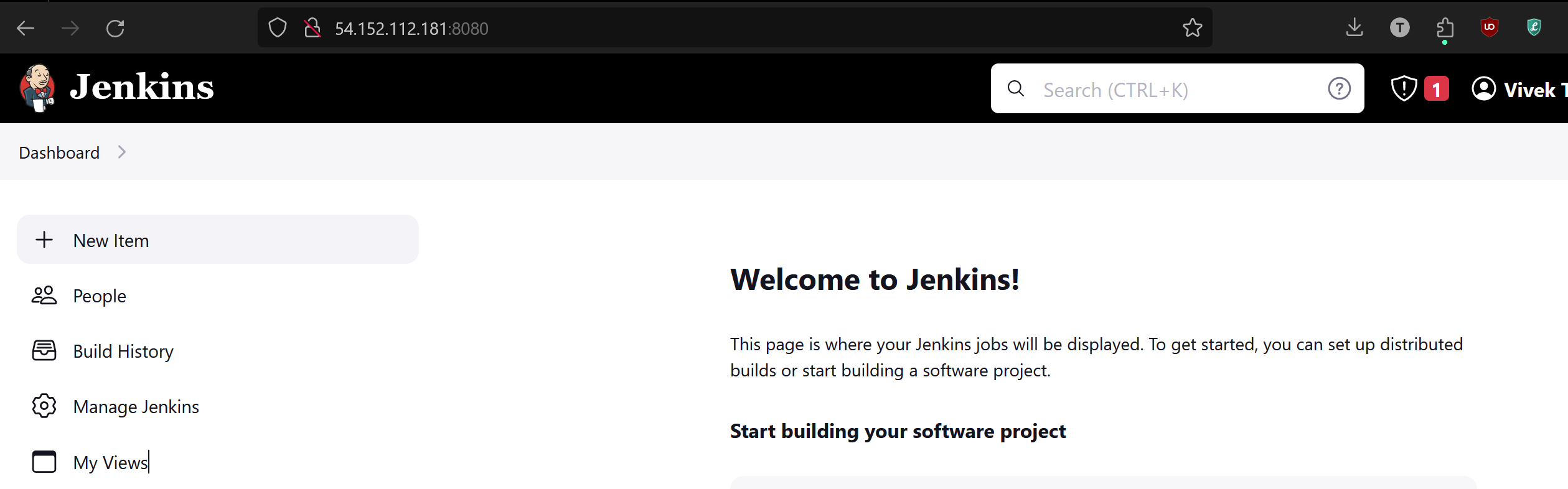
1. Once done, you’ll be greeted with the following message.



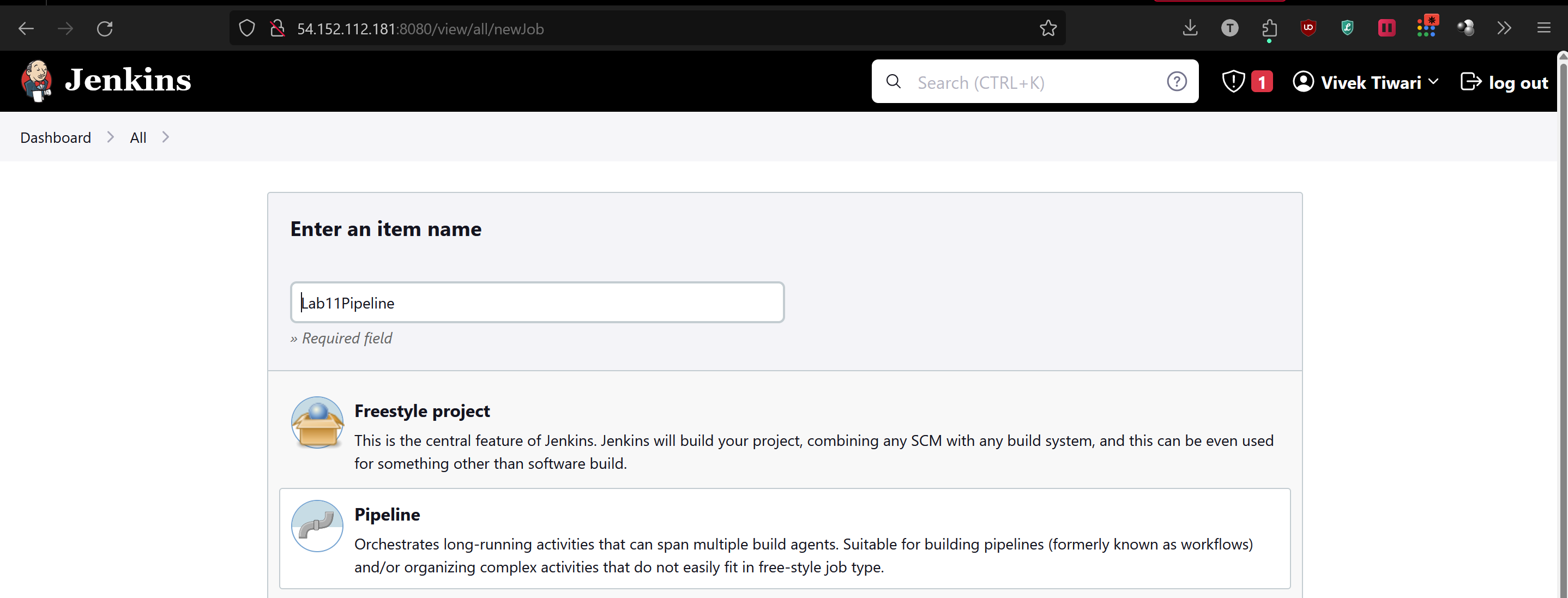
Now Jenkins is ready to use:



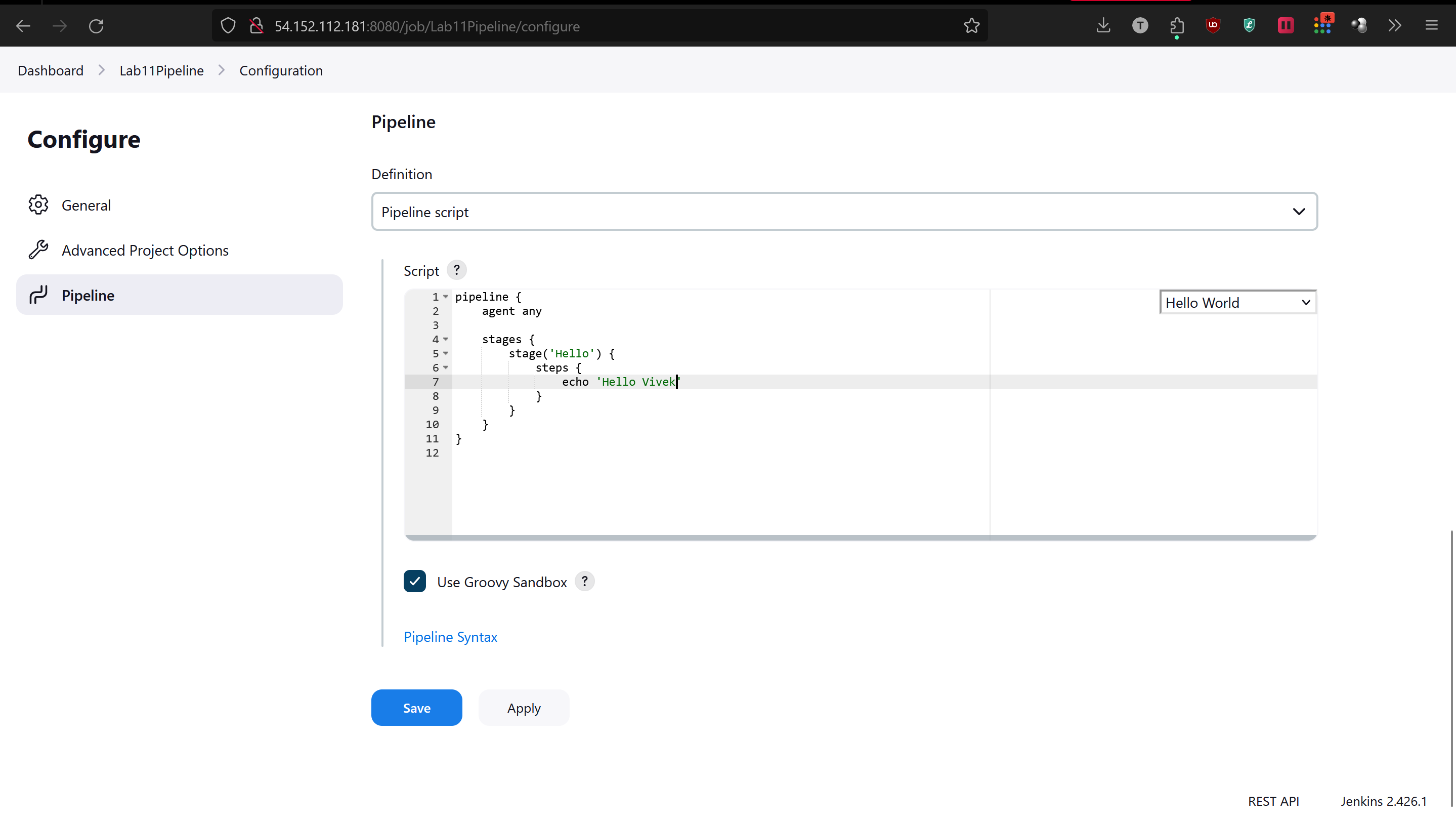
1. Click on new item:



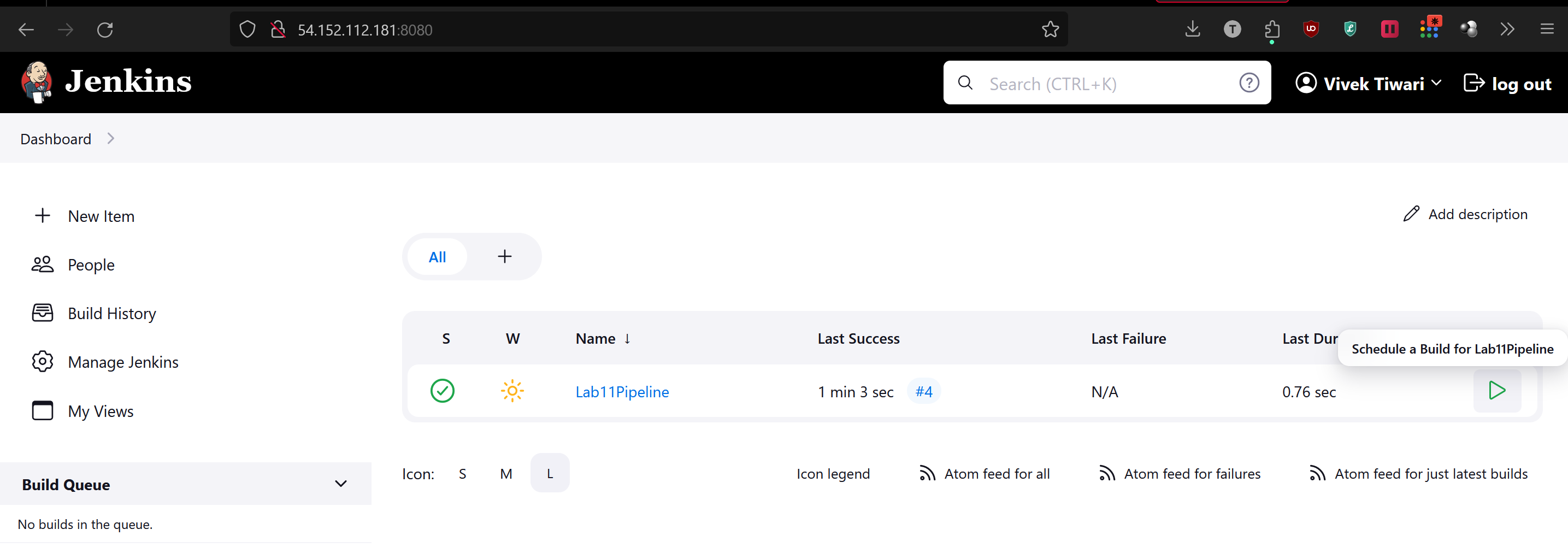
1. Now enter any suitable name for your pipeline and select the pipeline options from below:



1. On the next page, scroll down to Pipeline, and select ‘hello world’ from the top right drop down menu.



1. After saving it, go back to your dashboard and schedule a build:



1. Now you can check if your pipeline is functional or not:

