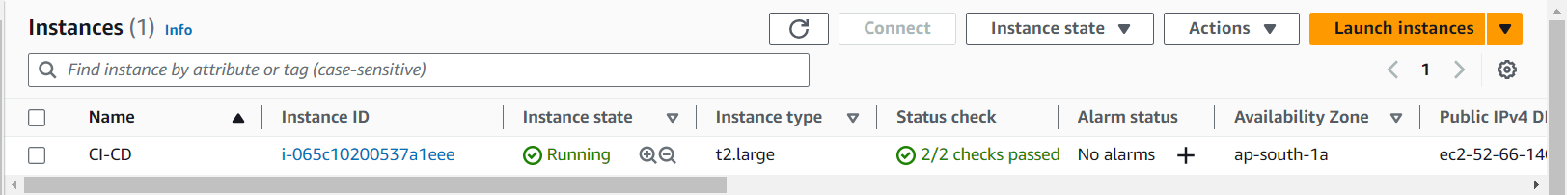


**Automate EC2 provisioning in AWS using Jenkins and Ansible Playbook**

We will learn how to create new EC2 instances using the Ansible playbook and automate using Jenkins Pipeline. in the end, we will play the game of 2048.

**Launch an Ubuntu(22.04) T2 medium Instance**

Launch an AWS T2 medium Instance. Use the image as Ubuntu. You can create a new key pair or use an existing one. Enable HTTP and HTTPS settings in the Security Group.



**Install Jenkins and Trivy**

**To Install Jenkins**

Connect to your console, and enter these commands to Install Jenkins

vi jenkins.sh

#!/bin/bash

sudo apt update -y

wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/public | tee /etc/apt/keyrings/adoptium.asc

echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.adoptium.net/artifactory/deb $(awk -F= '/^VERSION\_CODENAME/{print$2}' /etc/os-release) main" | tee /etc/apt/sources.list.d/adoptium.list

sudo apt update -y

sudo apt install temurin-17-jdk -y

/usr/bin/java --version

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \

/usr/share/keyrings/jenkins-keyring.asc > /dev/null

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

sudo apt-get update -y

sudo apt-get install jenkins -y

sudo systemctl start jenkins

sudo systemctl status jenkins

sudo chmod 777 jenkins.sh

./jenkins.sh # this will installl jenkins

Once Jenkins is installed, you will need to go to your AWS EC2 Security Group and open Inbound Port 8080, since Jenkins works on Port 8080.

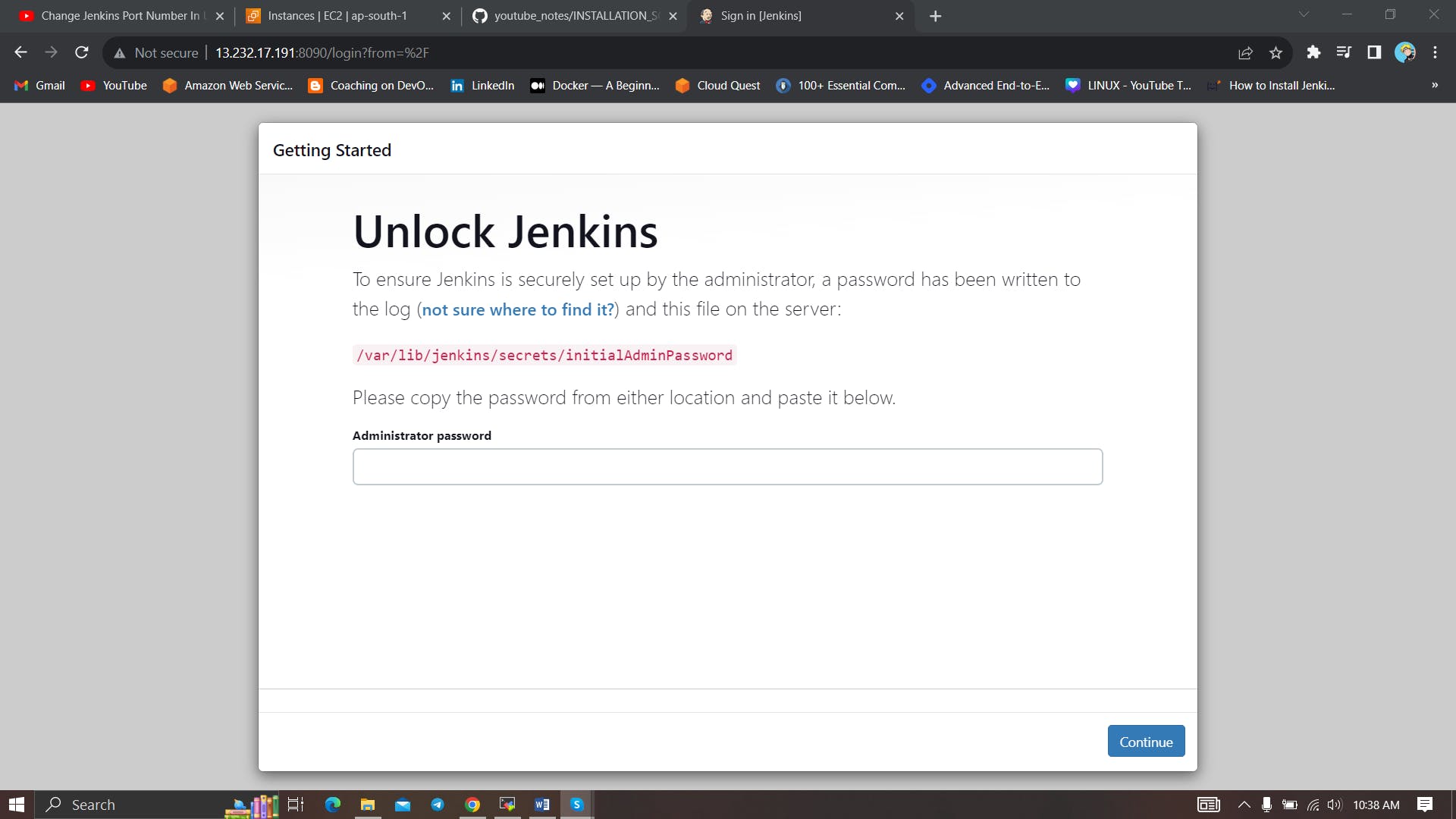
Now, grab your Public IP Address

COPY

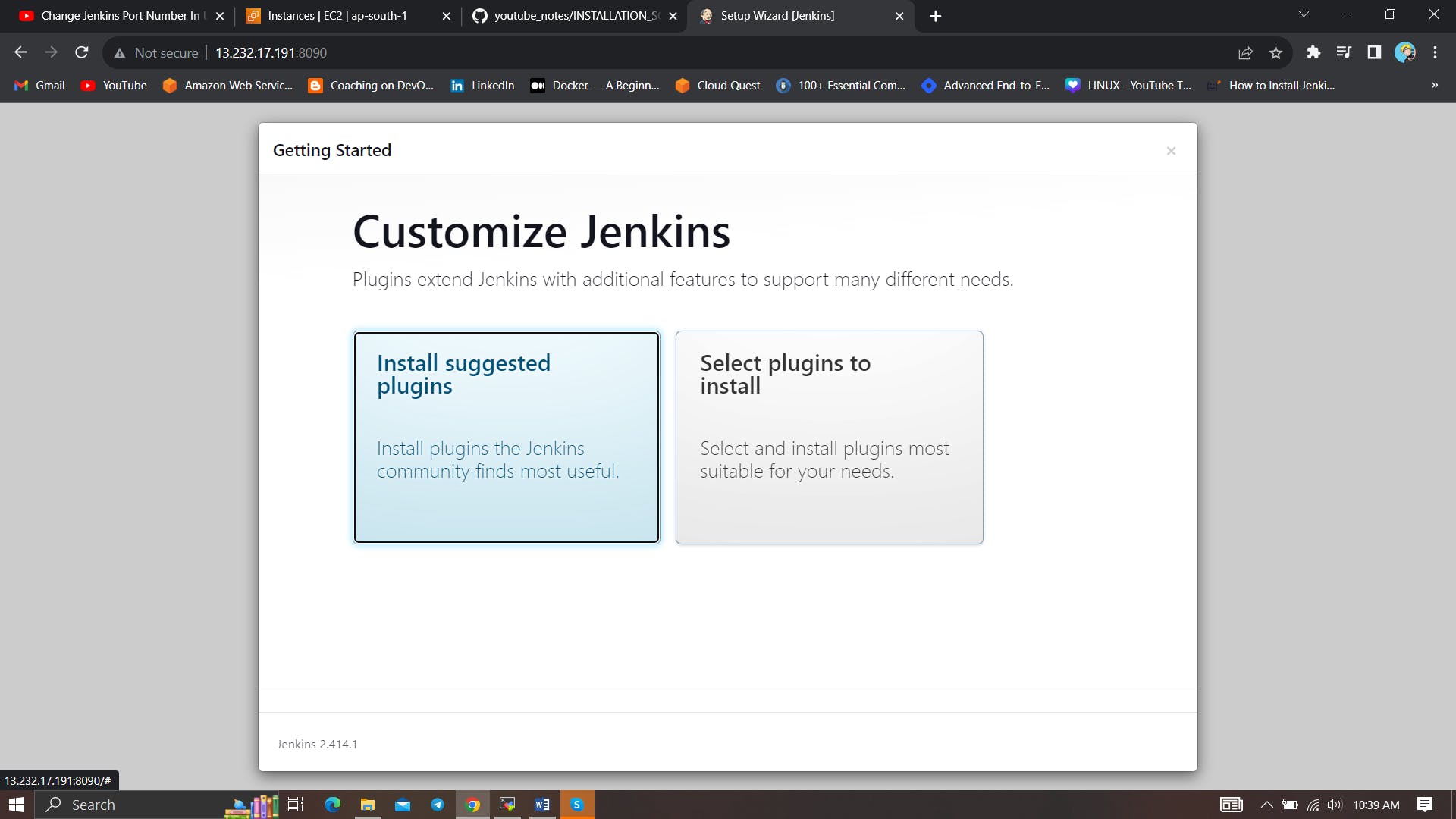
COPY

<EC2 Public IP Address:8080>

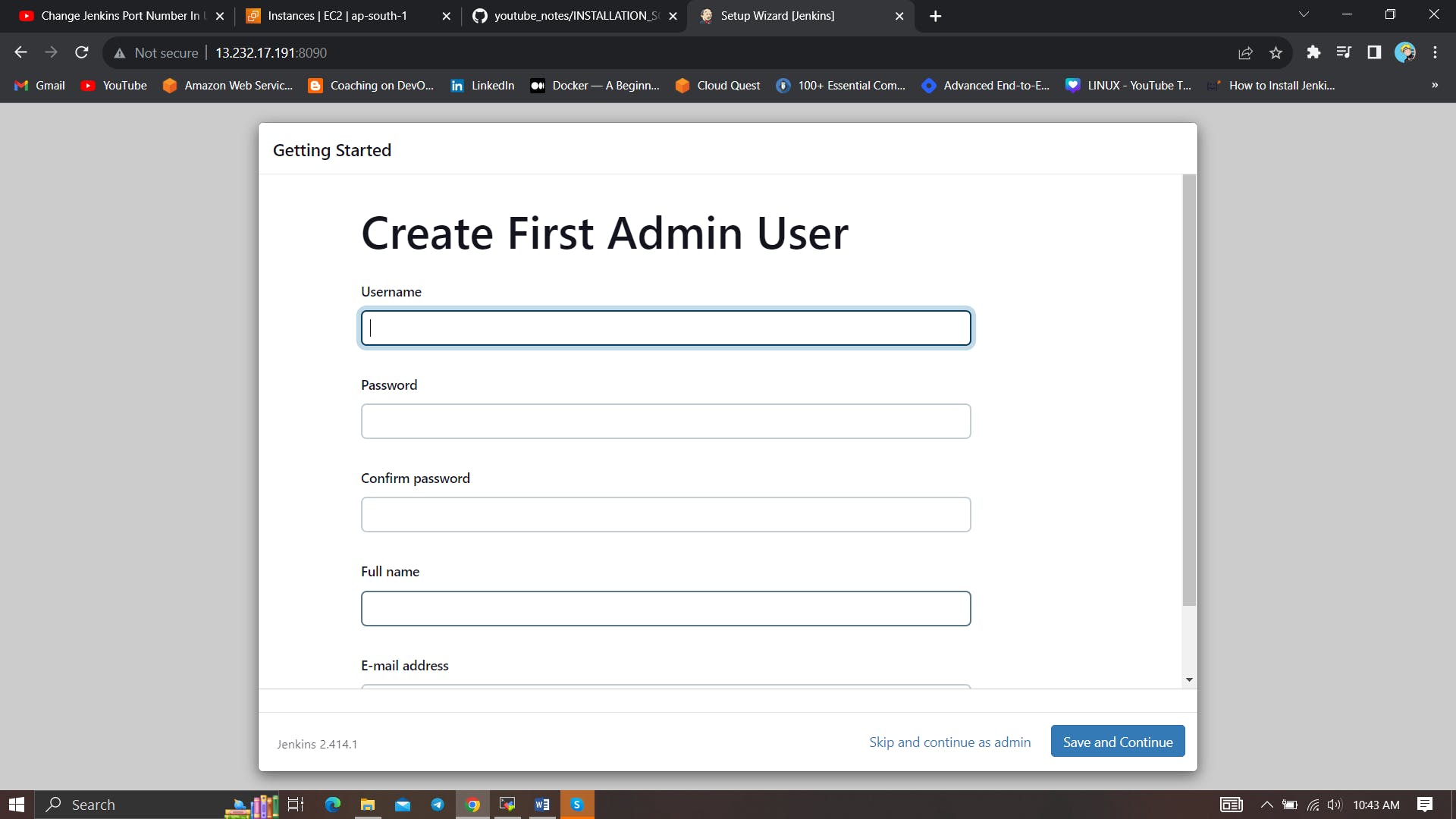
sudo cat /var/lib/jenkins/secrets/initialAdminPassword



Unlock Jenkins using an administrative password and install the suggested plugins.

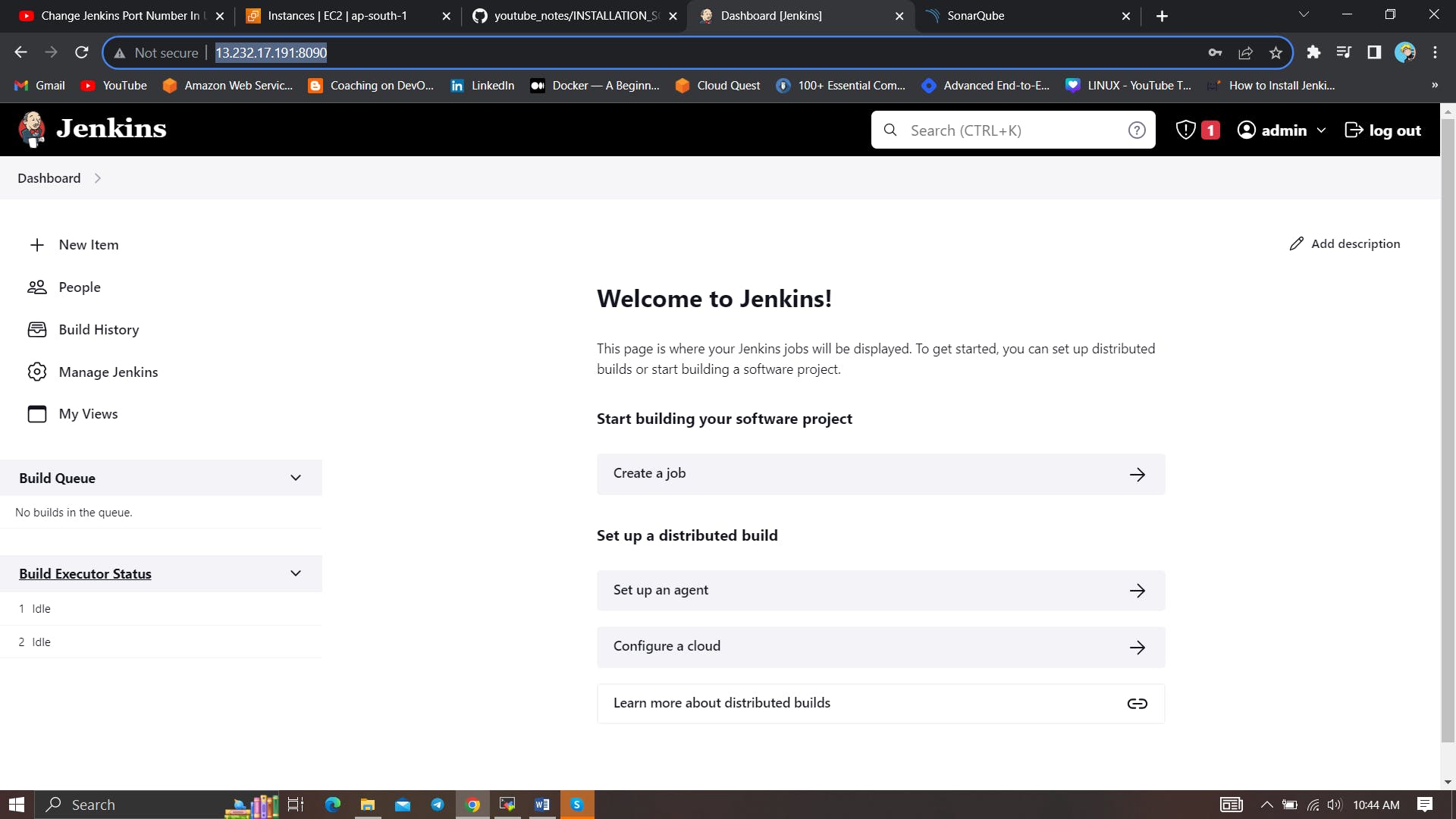


Jenkins will now get installed and install all the libraries.



Create a user click on save and continue.

Jenkins Getting Started Screen.



**Install Trivy**

vi trivy.sh

sudo apt-get install wget apt-transport-https gnupg lsb-release -y

wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null

echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $(lsb\_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list

sudo apt-get update

sudo apt-get install trivy -y

**Install Ansible**

connect to your Jenkins machine using Putty or Mobaxtreme

Now we are going to run the below commands on the Jenkins machine

Step1:Update your system packages:

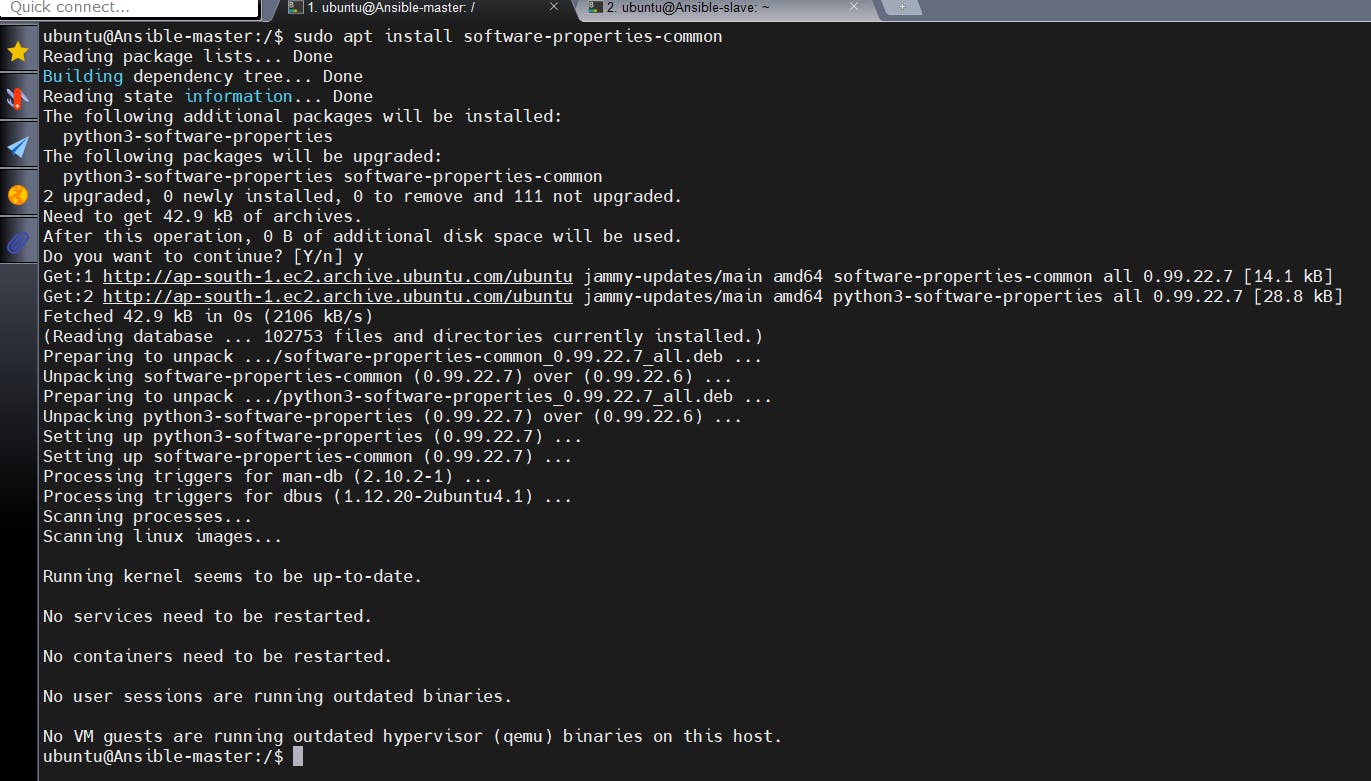
COPY

COPY

sudo apt-get update

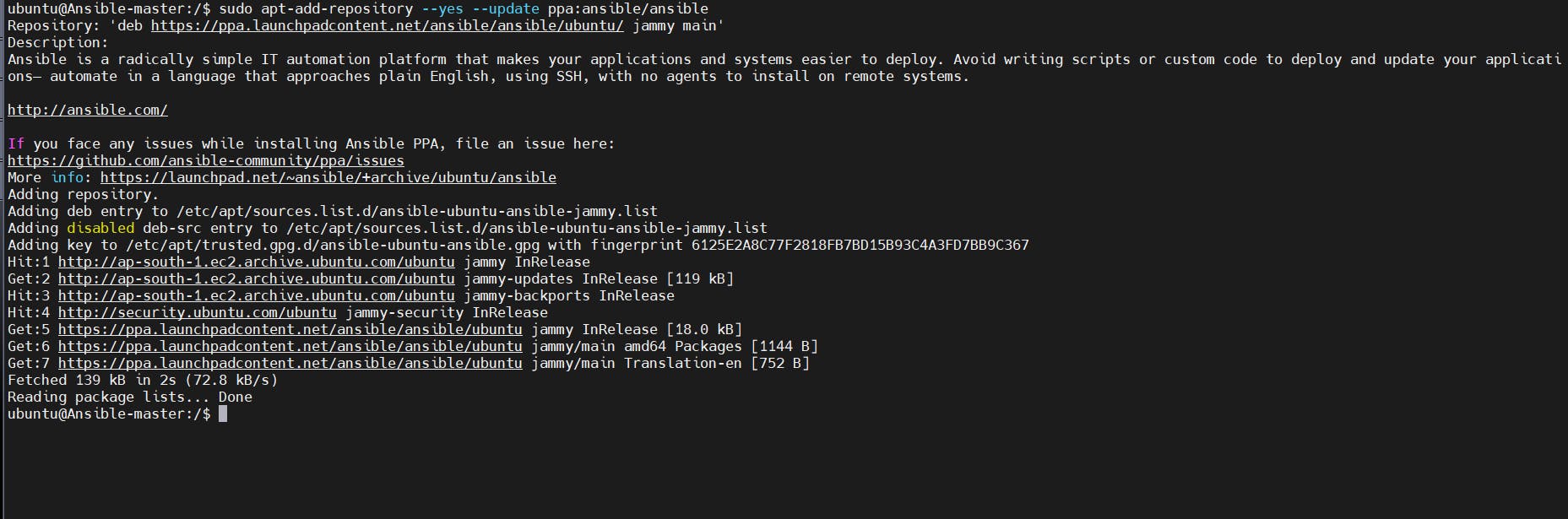
Step 2: First Install Required packages to install Ansible.

sudo apt install software-properties-common



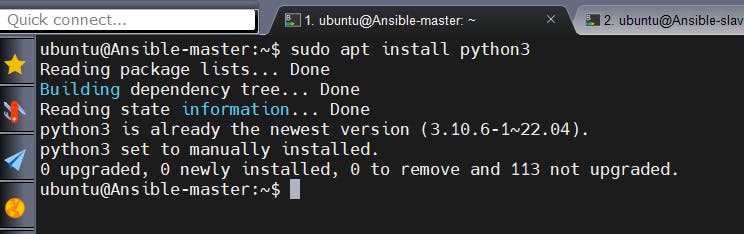
Step3: Add the ansible repository via PPA

sudo add-apt-repository --yes --update ppa:ansible/ansible



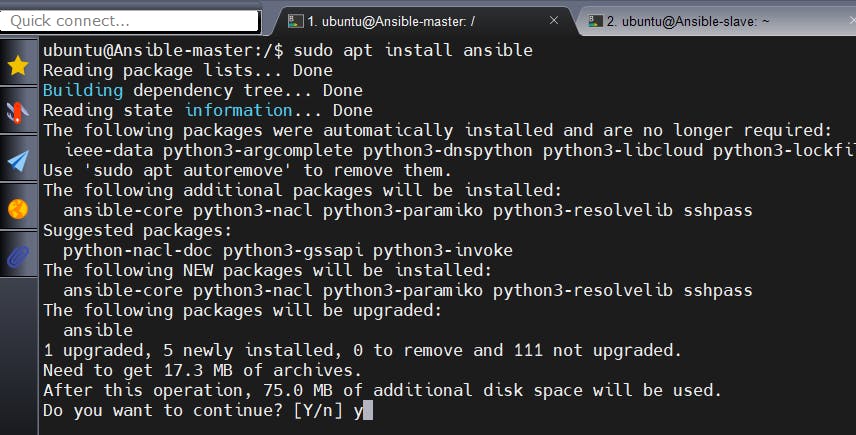
Install Python3 on Jenkins for Ansible

sudo apt install python3

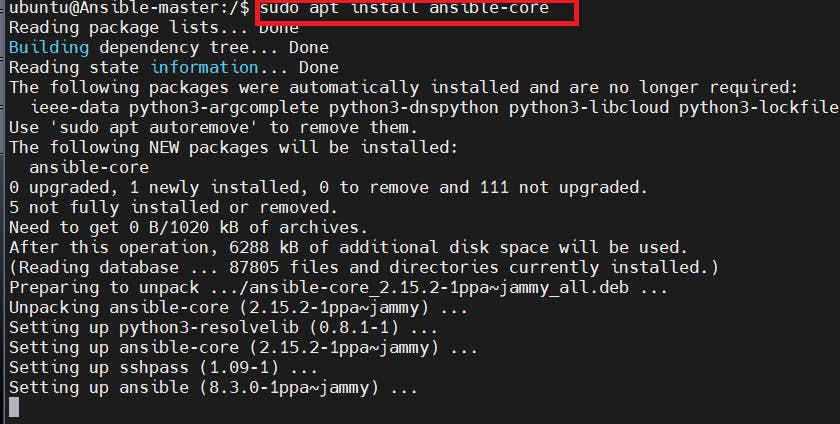


Install Ansible on Ubuntu 22.04 LTS

sudo apt install ansible -y



sudo apt install ansible-core



Step2: To check version :

ansible --version

**Install Python-pip3:**

sudo apt install python3-pip -y

( this is just comment - Package manager for python)Install Boto Framework - AWS SDKsudo pip3 install boto boto3

**Install Boto Framework - AWS SDK**

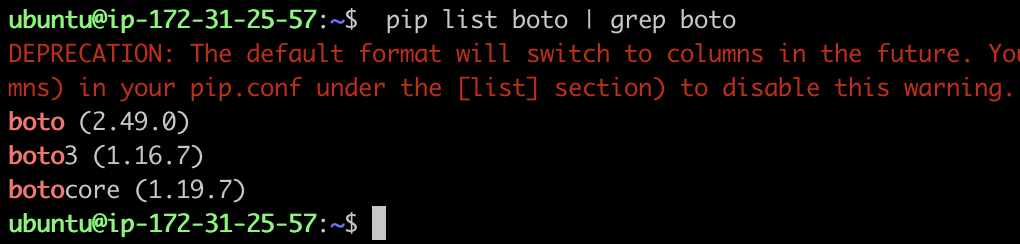
sudo pip3 install boto boto3

Ansible will access AWS resources using Boto SDK.

sudo apt-get install python3-boto -y

pip list boto | grep boto

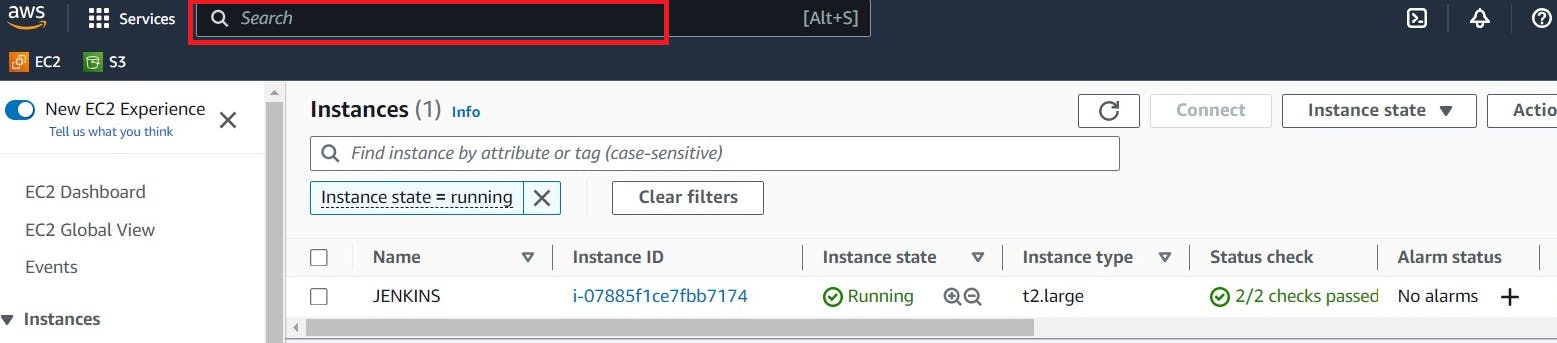
(the above command should display the below output)



let's create and attach an IAM role to Jenkins machines for the provision of a new ec2 instance

Navigate to **AWS CONSOLE**

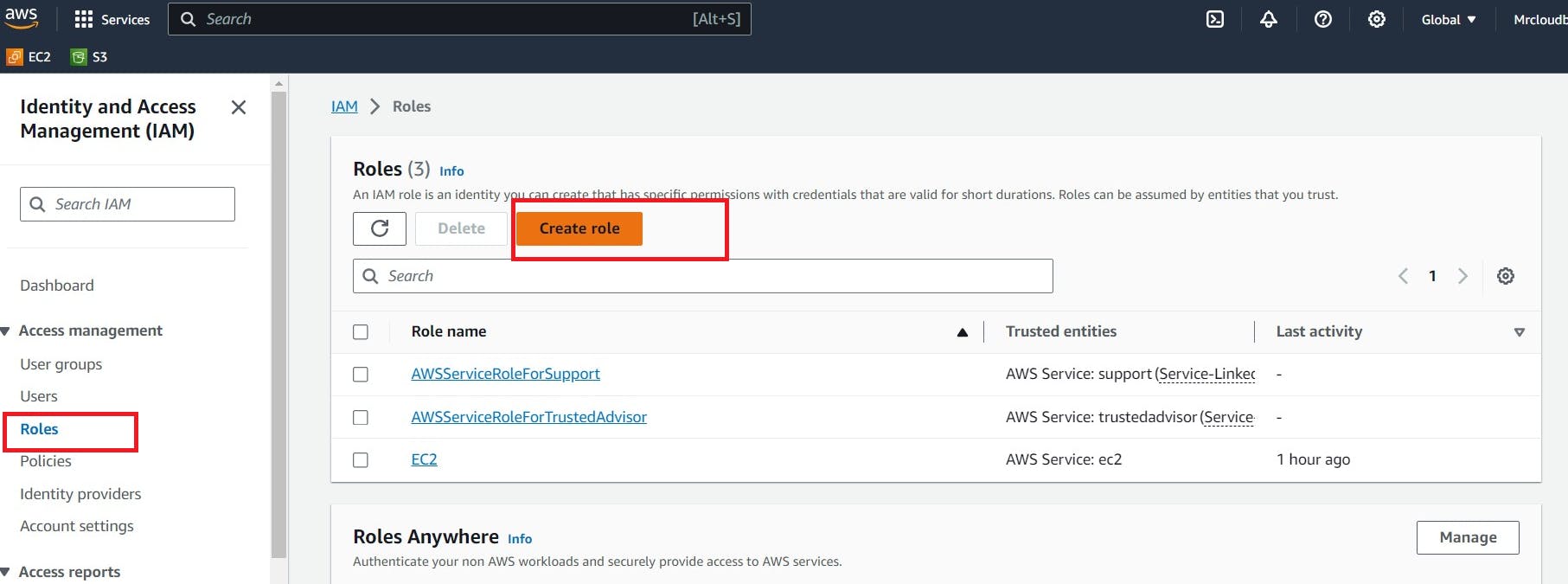
Click the "Search" field.



Type "IAM **enter**"

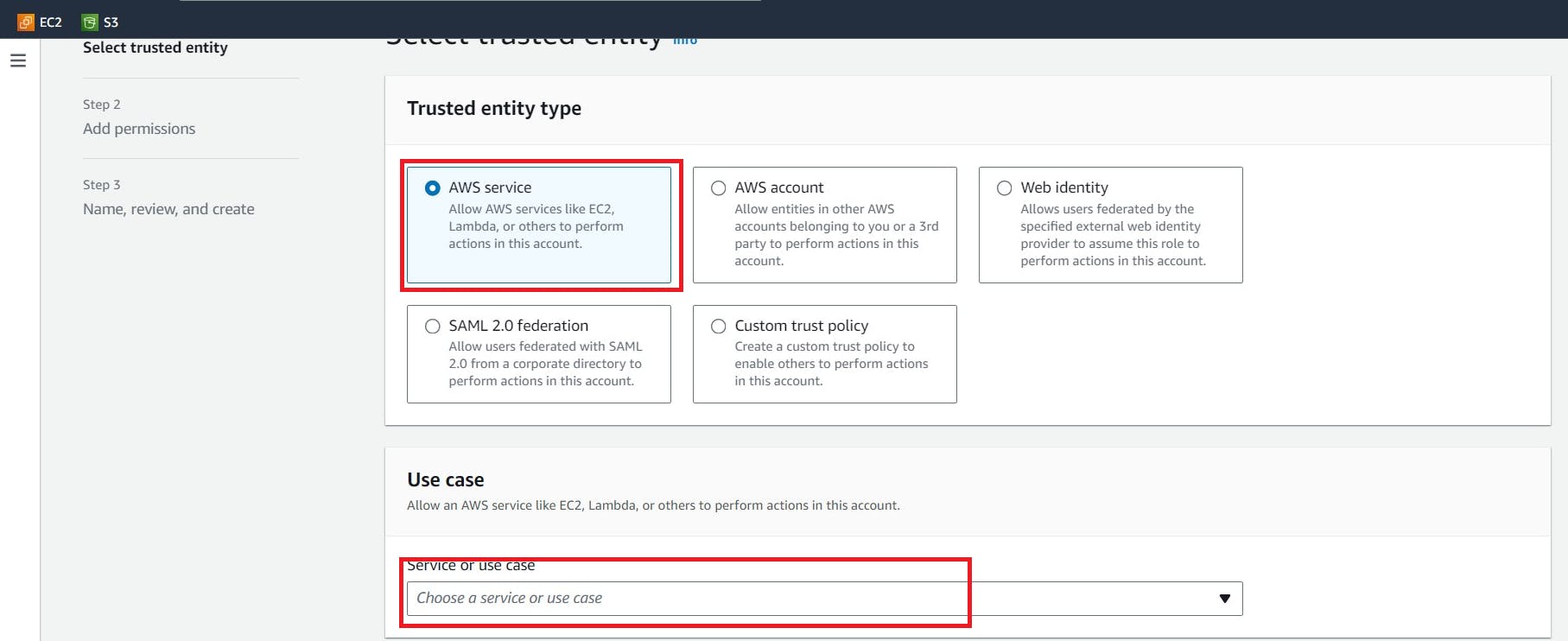
Click "Roles"

Click "Create role"



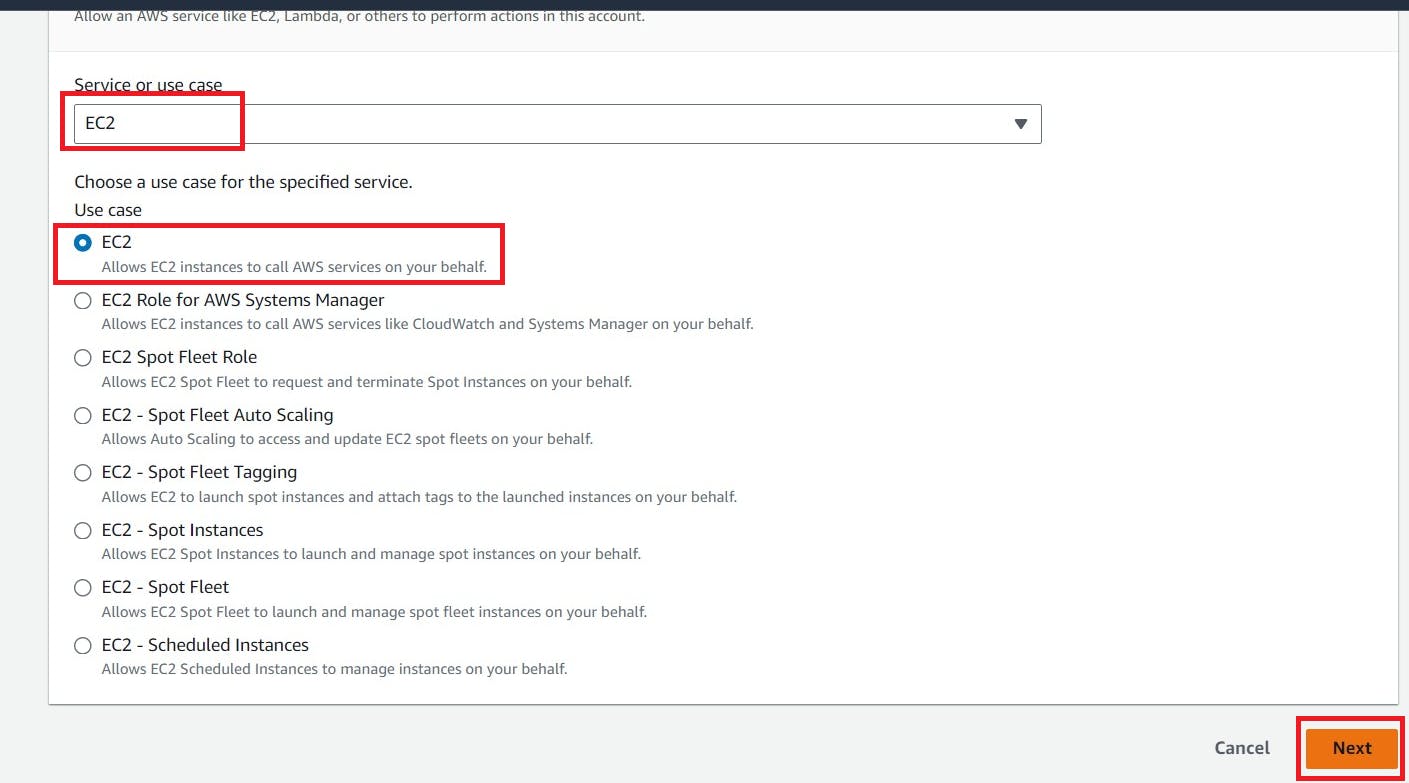
Click "AWS service"

Click "Choose a service or use case"



Click "EC2"

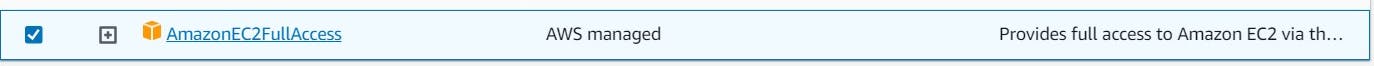
Click "Next"



Click the "Search" field.

Add permissions policies

AmazonEC2FullAccess



click Next

Click the "Role name" field.

Type "Jenkins-cicd"

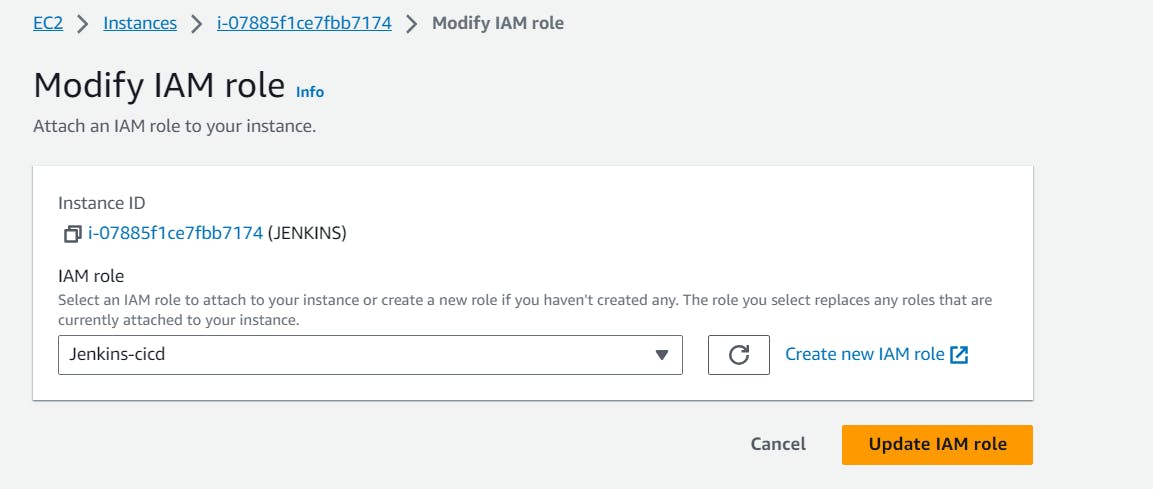
Click "Create role"

Click "EC2"

go to the Jenkins instance and add this role to the Ec2 instance.

select Jenkins instance --> Actions --> Security --> Modify IAM role

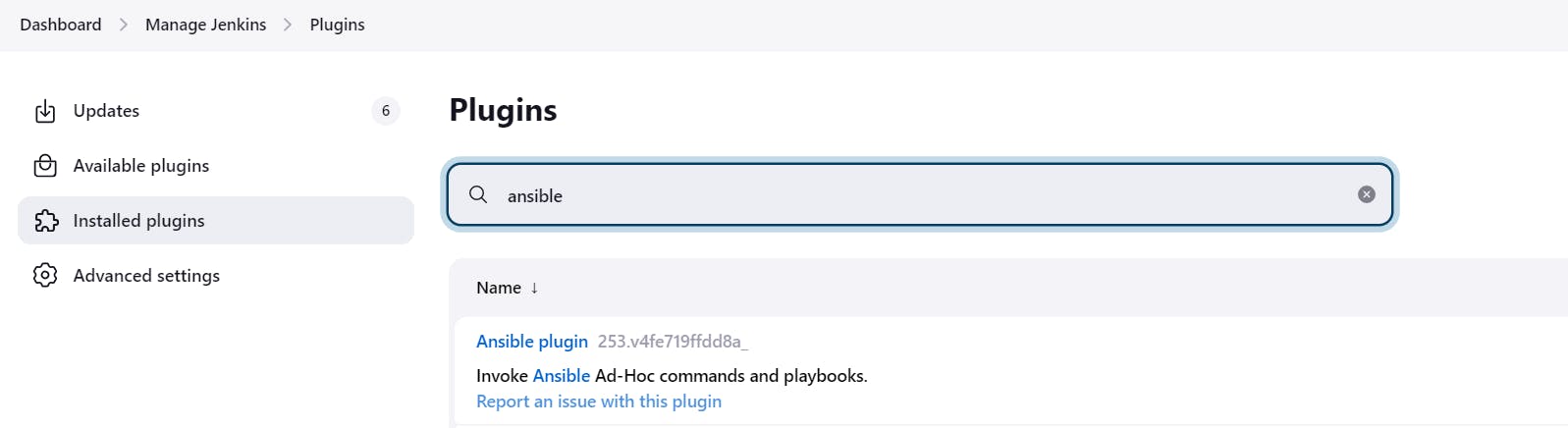
Add a newly created Role and click on Update IAM role.



Let's go to the Jenkins machine and add the Ansible Plugin

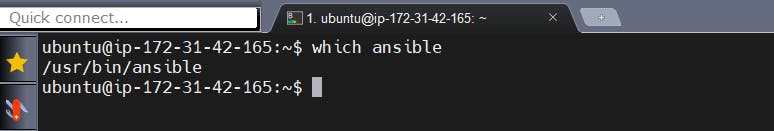
Manage Jenkins --> Plugins --> Available Plugins

search for Ansible and install

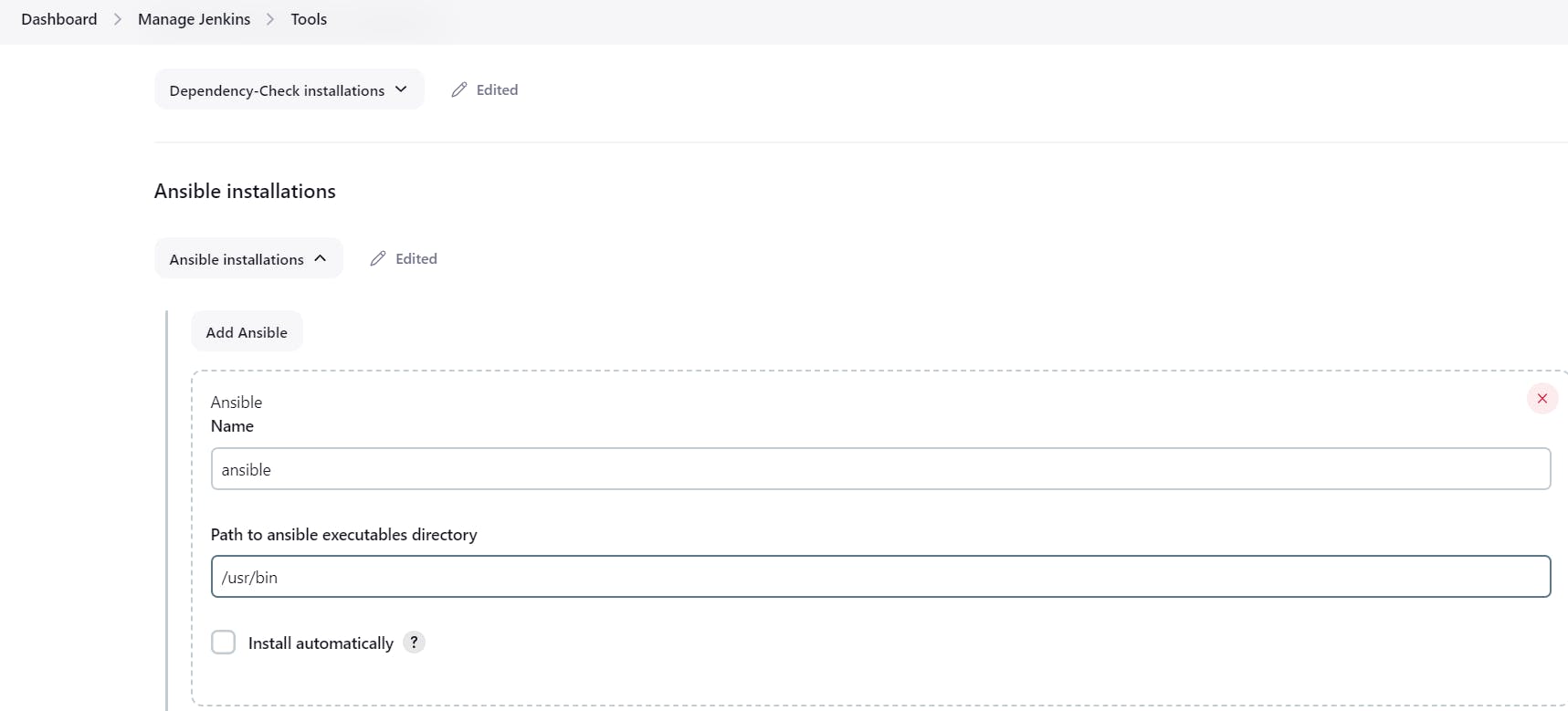


Give this command in your Jenkins machine to find the path of your ansible which is used in the tool section of Jenkins.

which ansible



Copy that path and add it to the tools section of Jenkins at ansible installations.



**Ansible Playbook**

---

- name: Provisioning a new EC2 instance and security group

hosts: localhost

connection: local

gather\_facts: False

tags: provisioning

pre\_tasks:

- name: Gather facts

setup:

- name: Print python version

debug:

msg: "Using Python {{ ansible\_python\_version }}"

- name: Install dependencies

shell: "/usr/bin/python3.10 -m pip install {{ item }}"

loop:

- boto3

- botocore

vars:

ansible\_python\_interpreter: /usr/bin/python3.10

keypair: Mumbai

instance\_type: t2.micro

image\_id: ami-0f5ee92e2d63afc18

wait: yes

group: webserver

count: 1

region: ap-south-1

security\_group: ec2-security-group

tag\_name:

Name: Aj-ec2

tasks:

- name: Create a security group

amazon.aws.ec2\_group:

name: "{{ security\_group }}"

description: Security Group for webserver Servers

region: "{{ region }}"

rules:

- proto: tcp

from\_port: 22

to\_port: 22

cidr\_ip: 0.0.0.0/0

- proto: tcp

from\_port: 8080

to\_port: 8080

cidr\_ip: 0.0.0.0/0

- proto: tcp

from\_port: 3000

to\_port: 3000

cidr\_ip: 0.0.0.0/0

- proto: tcp

from\_port: 80

to\_port: 80

cidr\_ip: 0.0.0.0/0

- proto: tcp

from\_port: 443

to\_port: 443

cidr\_ip: 0.0.0.0/0

rules\_egress:

- proto: all

cidr\_ip: 0.0.0.0/0

register: basic\_firewall

- name: Launch the new EC2 Instance

amazon.aws.ec2\_instance:

security\_group: "{{ security\_group }}"

instance\_type: "{{ instance\_type }}"

image\_id: "{{ image\_id }}"

wait: "{{ wait }}"

region: "{{ region }}"

key\_name: "{{ keypair }}"

count: "{{ count }}"

tags: "{{ tag\_name }}"

user\_data: |

#!/bin/bash

sudo apt update -y

sudo apt install docker.io -y

sudo systemctl start docker

sudo systemctl enable docker

sudo docker run -d --name 2048 -p 3000:3000 sevenajay/2048:latest

register: ec2

Write a sample pipeline for Provision

pipeline {

agent any

tools{

ansible 'ansible'

}

stages {

stage('cleanws') {

steps {

cleanWs()

}

}

stage('checkout'){

steps{

git branch: 'main', url: 'https://github.com/Aj7Ay/ANSIBLE.git'

}

}

stage('TRIVY FS SCAN') {

steps {

sh "trivy fs . > trivyfs.txt"

}

}

stage('ansible provision') {

steps {

// To suppress warnings when you execute the playbook

sh "pip install --upgrade requests==2.20.1"

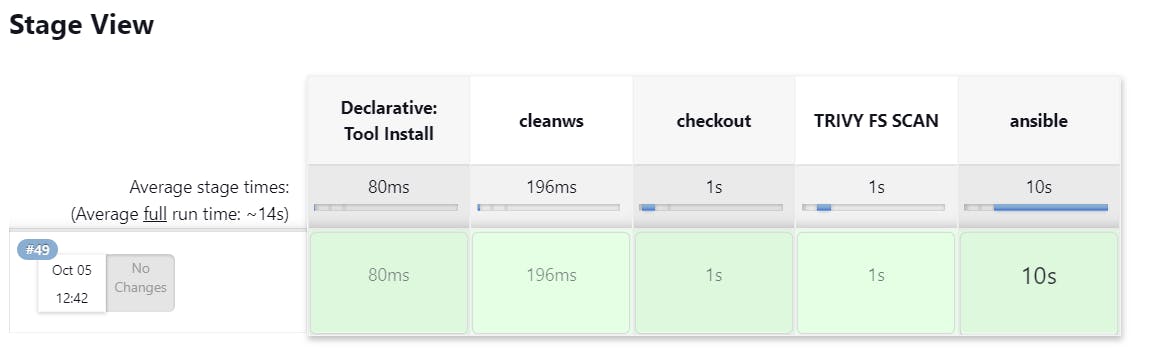
ansiblePlaybook playbook: 'ec2.yaml'

}

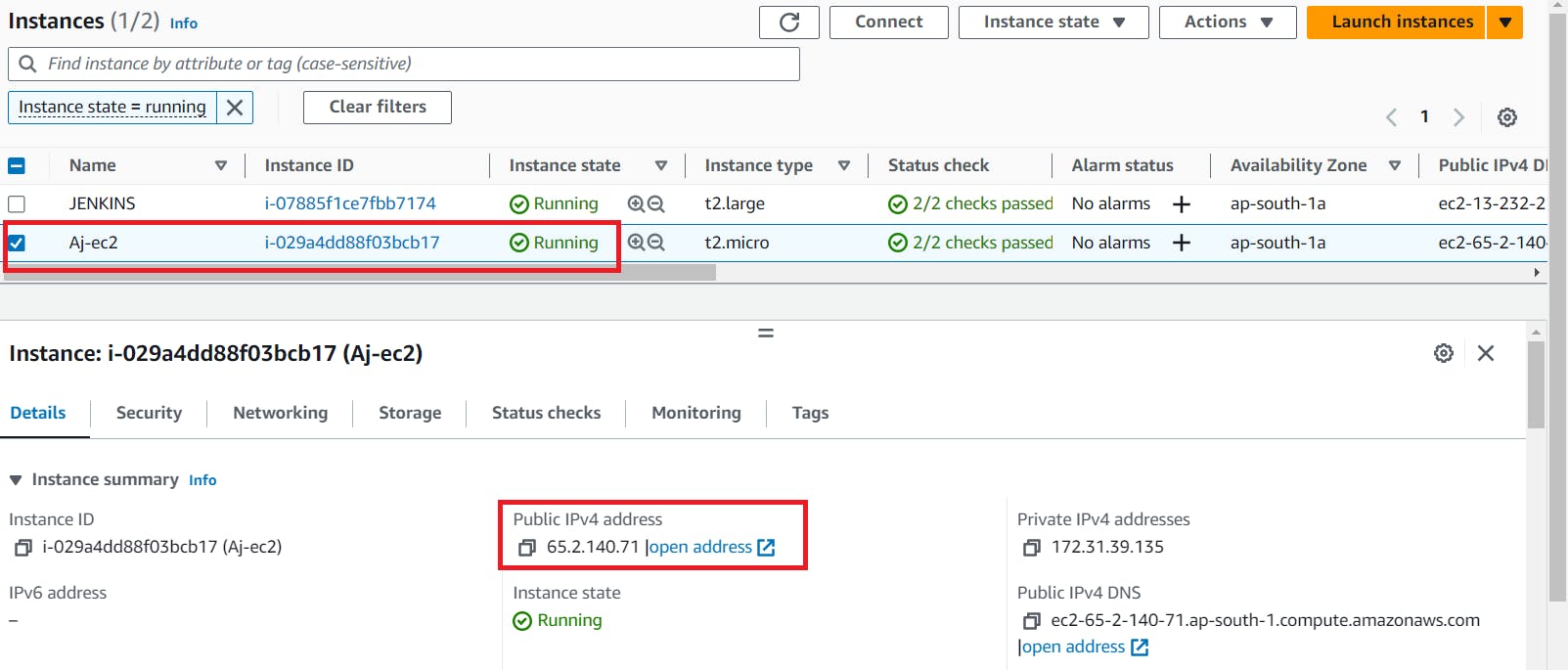
}

}

}



Provision Ec2-instance



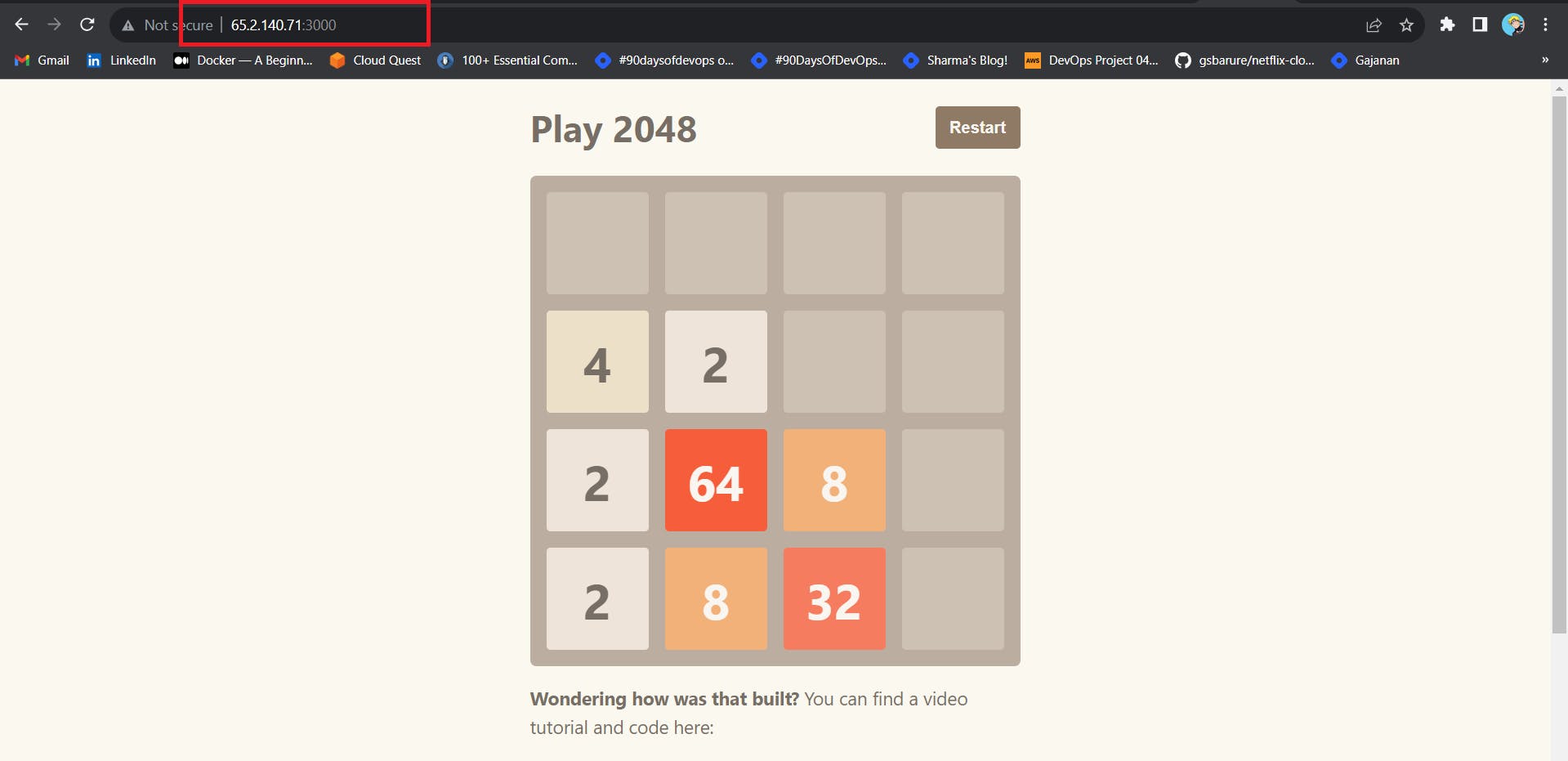
copy the Public IP of the provisioned instance

COPY

COPY

<public-ip:3000>

Play Game 2048



Delete the instances.

Continuous improvement is not just about doing the same thing better. It's about reimagining what's possible and striving for excellence in every line of code, every pipeline, and every deployment. In the world of Jenkins and Ansible, automation is the key to unlocking innovation. So, keep coding, keep automating, and keep pushing the boundaries of what's achievable. Your journey has just begun.

Thanks for Reading my Blog.