**APPLICATION DEPLOYMENT**

**USING DOCKER-JENKINS-AWS EKS,ECR**

**DESCRIPTION:**

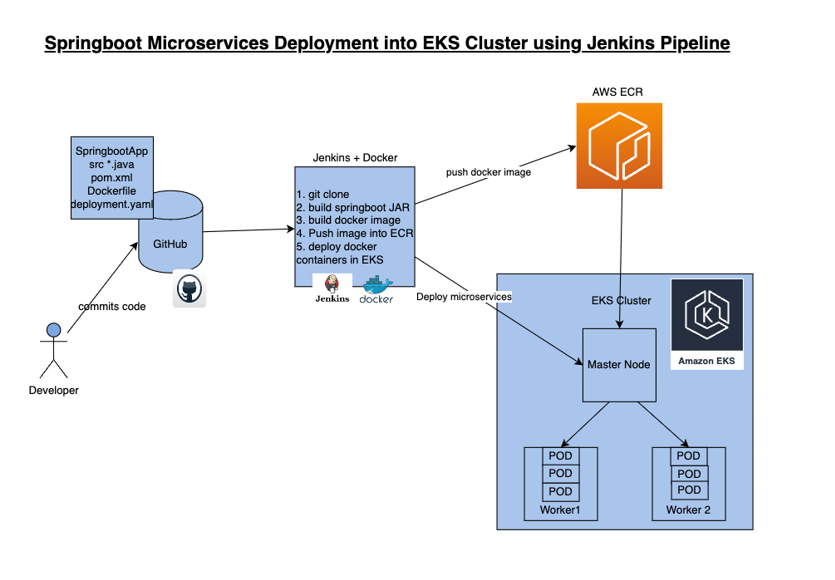
**Automate springboot microservices builds using Jenkins pipeline and Deploy into AWS EKS Cluster.**

**- Automating builds using Jenkins**

**- Automating Docker image creation**

**- Automating Docker image upload into AWS ECR**

**- Automating Docker Containers Deployments to Kubernetes Cluster**

****

**Repo Link -**

[**https://github.com/SAARA213/EKS\_CICD\_PROJECT.git**](https://github.com/SAARA213/EKS_CICD_PROJECT.git)

**Pre-requisites:**

**1. Amazon EKS Cluster is Setup and running.**

**2. Create ECR repo in AWS**

**3.Jenkins is up and running**

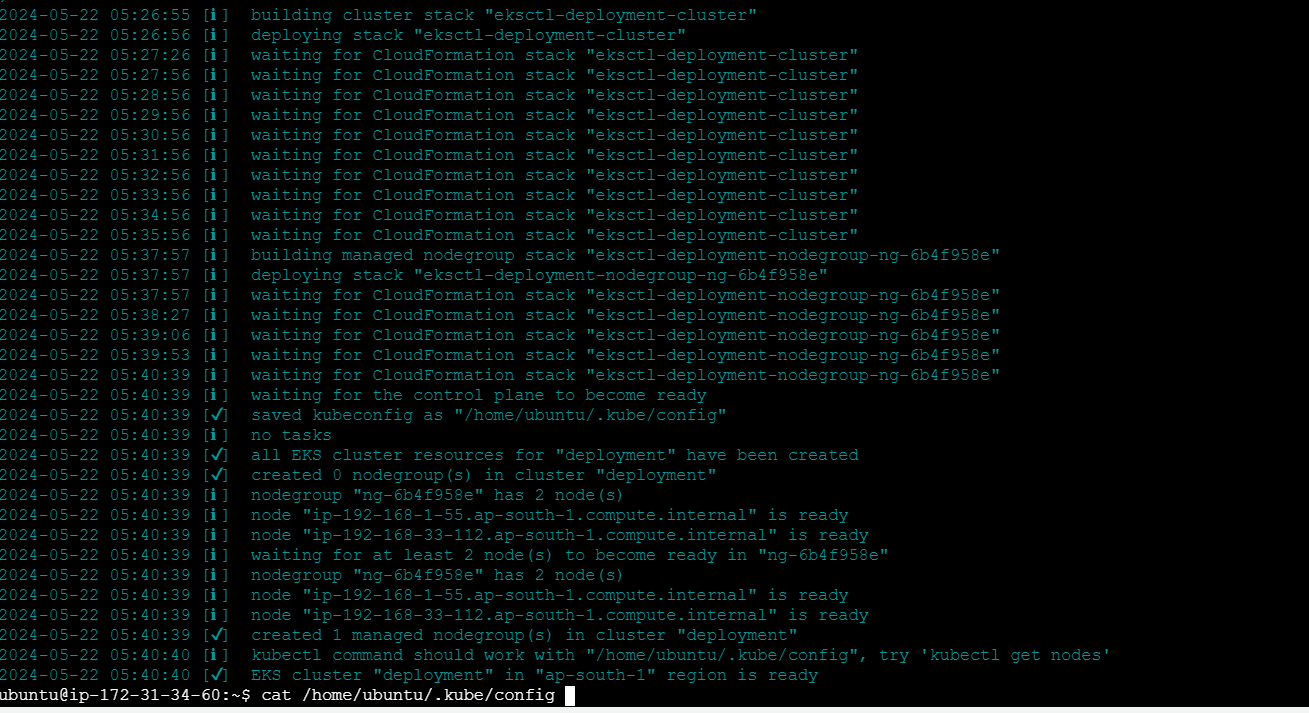
**4. Docker installed on Jenkins instance**

**5. Docker, Docker pipeline and Kubernetes CLI plug-ins are installed in Jenkins**

**6.Install kubectl on ec2 instance**

**Execute the below command to get kubeconfig info, copy the entire content of the file:**

**cat /var/lib/jenkins/.kube/config**



**Open your text editor or notepad, copy and paste the entire content and save in a file.**

**We will upload this file.**

[**Jenkins Master is up and running**](https://www.coachdevops.com/2020/04/install-jenkins-ubuntu-1804-setup.html)

### **Install Jenkins on Ubuntu 22.0.4**

**Jenkins is an open source continuous integration/continuous delivery and deployment (CI/CD) automation software DevOps tool written in the Java programming language. It is used to implement CI/CD workflows, called pipelines.**

**Change Host Name to Jenkins**

**sudo hostnamectl set-hostname Jenkins**

**Perform update first**

**sudo apt update**

**Install Java 11**

**sudo apt install default-jdk -y**

**Once install java, enter the below command**

**Verify Java Version**

**java -version**

**Now lets start Jenkins installation**

**Jenkins Setup**

**Add Repository key to the system**

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

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

**Append debian package repo address to the system**

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

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

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

**Update Ubuntu package**

sudo apt update

**Install Jenkins**

sudo apt install jenkins -y

**Access Jenkins in web browser**

**Now Go to AWS console. Click on EC2, click on running instances link. Select the checkbox of EC2 you are installing Java and Jenkins. Click on Action. Copy the value from step 4 that says --> Connect to your instance using its Public DNS:**

[**Docker installed on Jenkins instance**](https://www.coachdevops.com/2020/05/docker-jenkins-integration-building.html)

**Every time developer makes code changes, you would want the Jenkins to automate Docker images creation and pushing into Docker registry**

**Install docker**

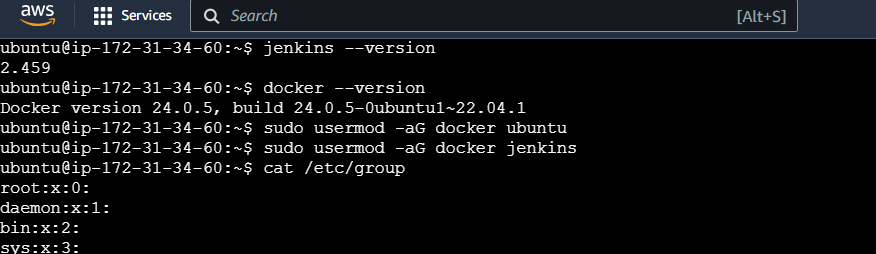
**sudo apt install docker.io -y**

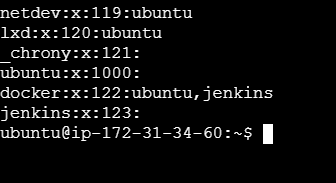
**Add Ubuntu user to Docker group**

**sudo usermod -aG docker $USER**

**Add jenkins user to Docker group**

**sudo usermod -a -G docker jenkins**



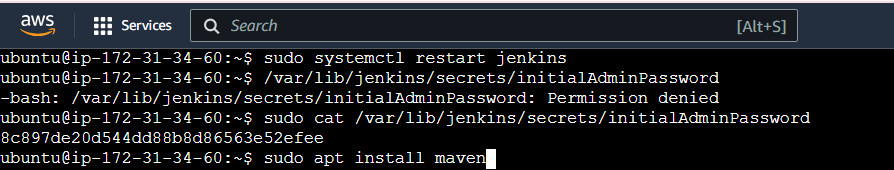


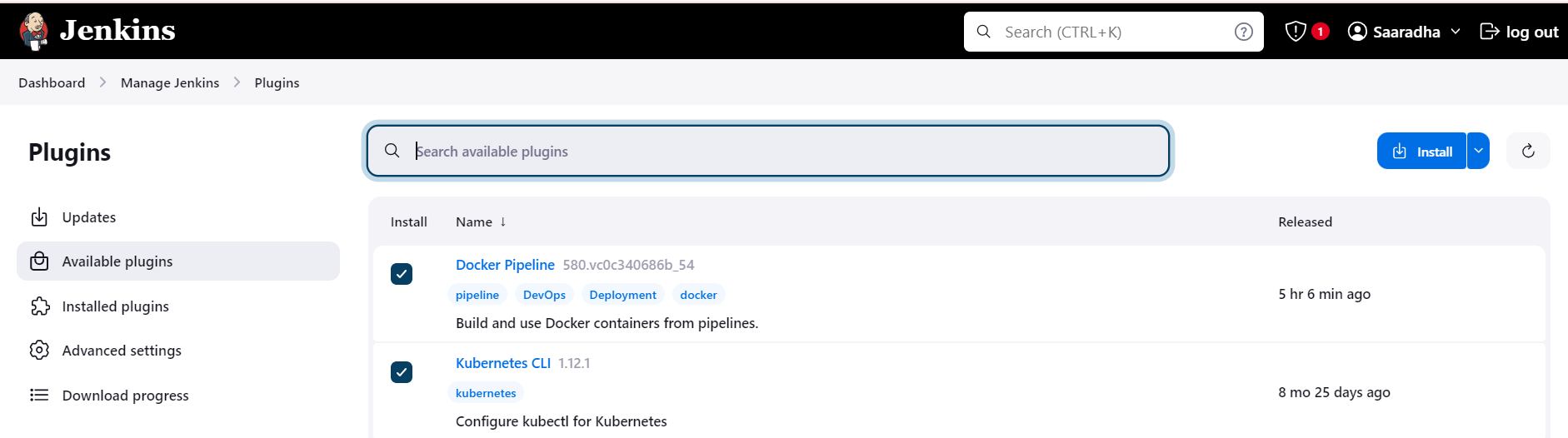
**Maven Installation**

Maven is a popular build tool used for building Java applications. You can install Maven by executing below command:

sudo apt install maven -y

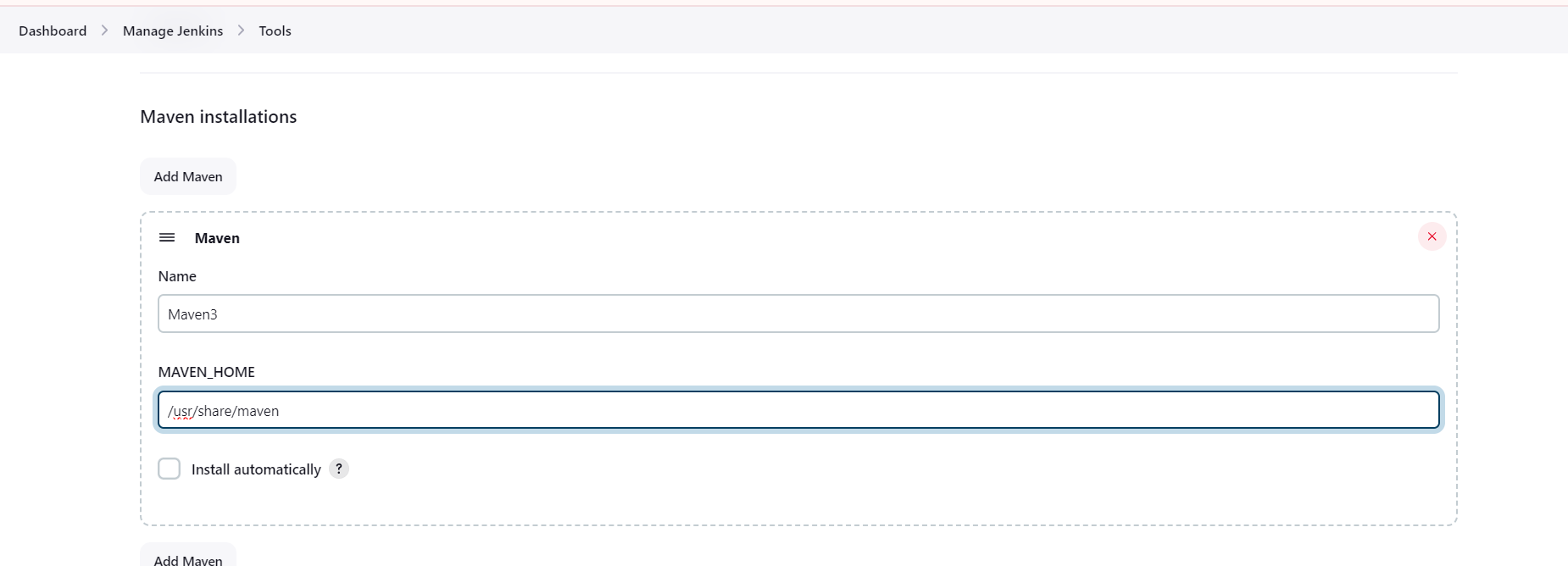
mvn --version



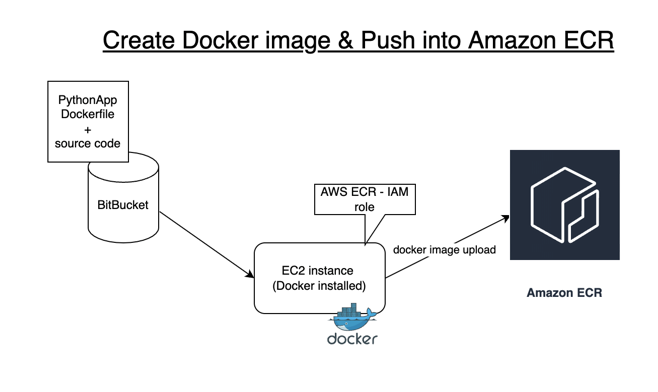


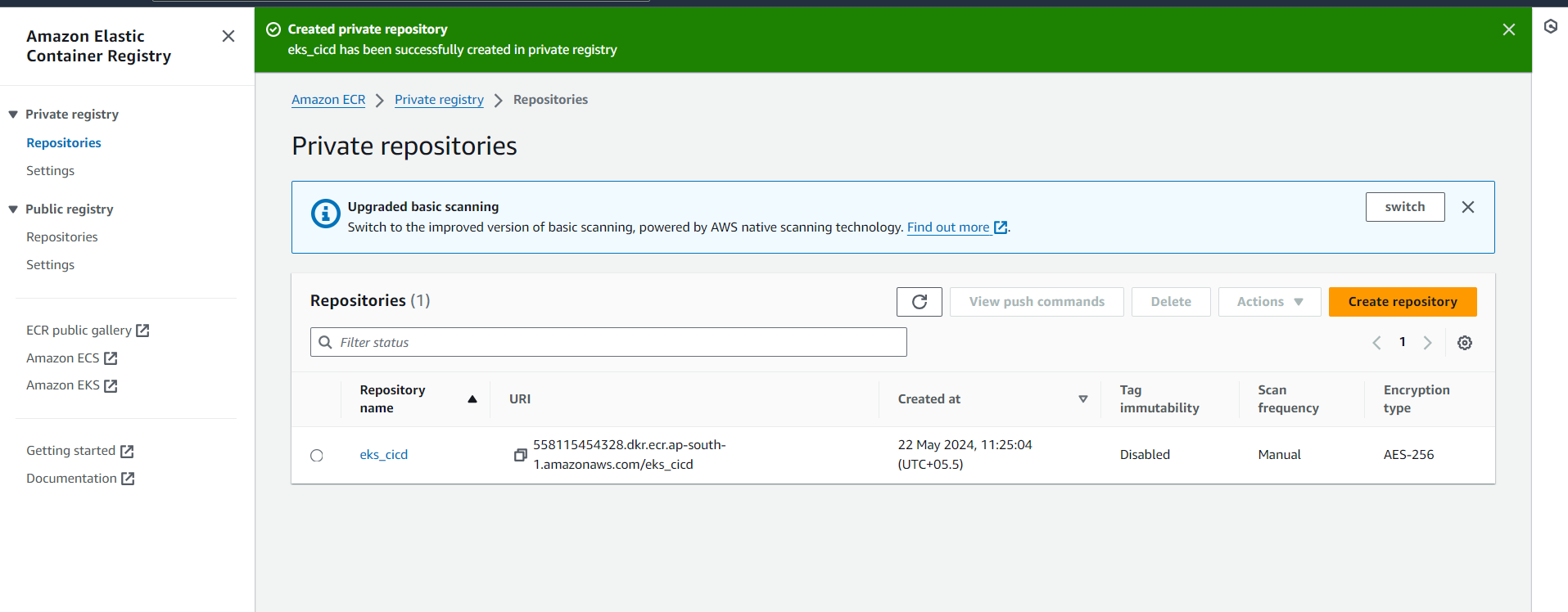
**Create Maven3 variable under Global tool configuration in Jenkins**

**Make sure you create Maven3 variable under Global tool configuration.**



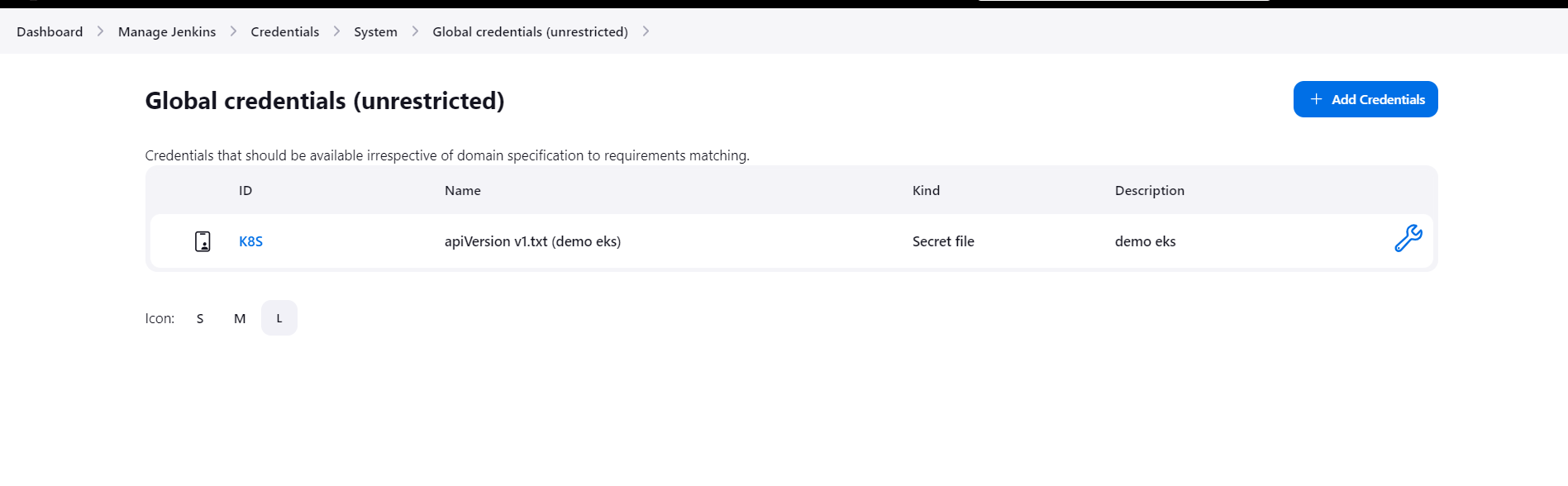
[**Create ECR repo in AWS**](https://www.cidevops.com/2020/05/how-to-setup-elastic-container-registry.html)

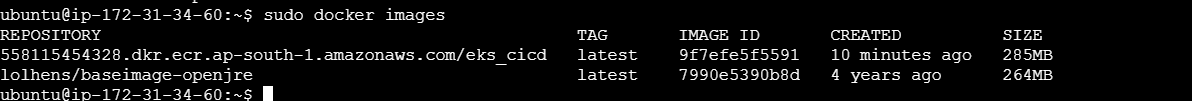




**Create Credentials for connecting to Kubernetes Cluster using kubeconfig**

**Click on Add Credentials, use Kubernetes configuration from drop down.**





**Create a pipeline in Jenkins**

**Pipeline script:**

**‘’’**

**pipeline {**

**tools {**

**maven 'Maven3'**

**}**

**agent any**

**environment {**

**registry = "account\_id.dkr.ecr.us-east-2.amazonaws.com/my-docker-repo"**

**}**

**stages {**

**stage('Cloning Git') {**

**steps {**

**checkout([$class: 'GitSCM', branches: [[name: '\*/main']], doGenerateSubmoduleConfigurations: false, extensions: [], submoduleCfg: [], userRemoteConfigs: [[credentialsId: '', url: 'https://github.com/yasminjeelani/EKS-CICD.git']]])**

**}**

**}**

**stage ('Build') {**

**steps {**

**sh 'mvn clean install'**

**}**

**}**

**// Building Docker images**

**stage('Building image') {**

**steps{**

**script {**

**dockerImage = docker.build registry**

**}**

**}**

**}**

**// Uploading Docker images into AWS ECR**

**stage('Pushing to ECR') {**

**steps{**

**script {**

**sh 'aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin account\_id.dkr.ecr.us-east-2.amazonaws.com'**

**sh 'docker push account\_id.dkr.ecr.us-east-2.amazonaws.com/my-docker-repo:latest'**

**}**

**}**

**}**

**stage('K8S Deploy') {**

**steps{**

**script {**

**sh ('kubectl apply -f eks-deploy-k8s.yaml')**

**}**

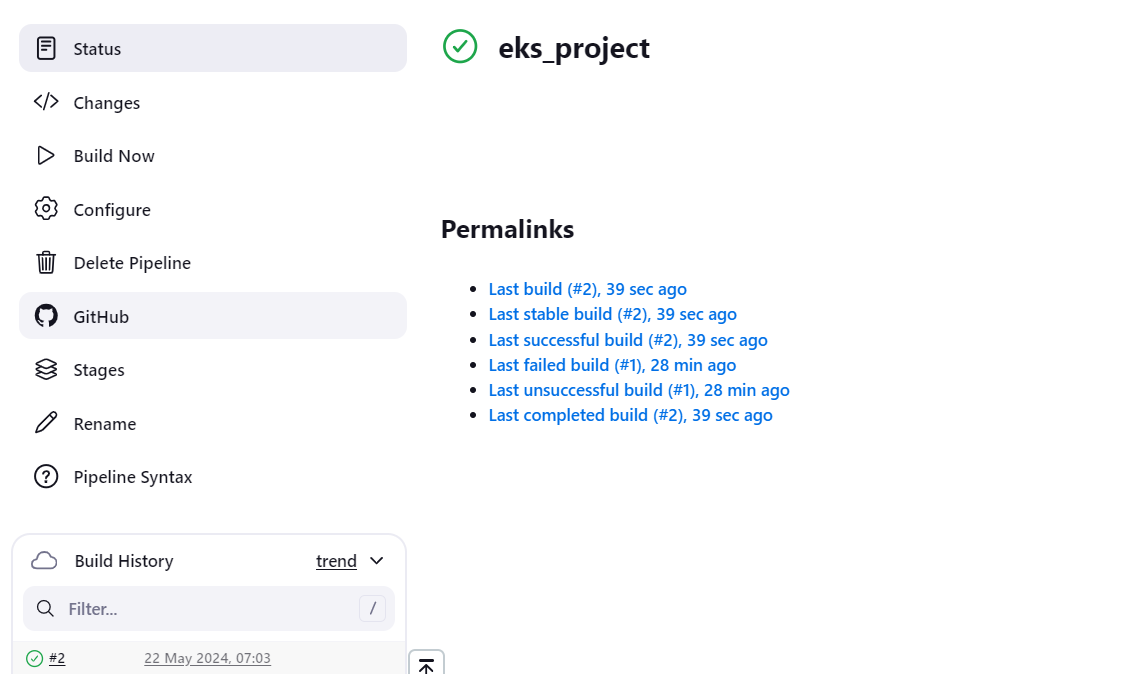
**}**

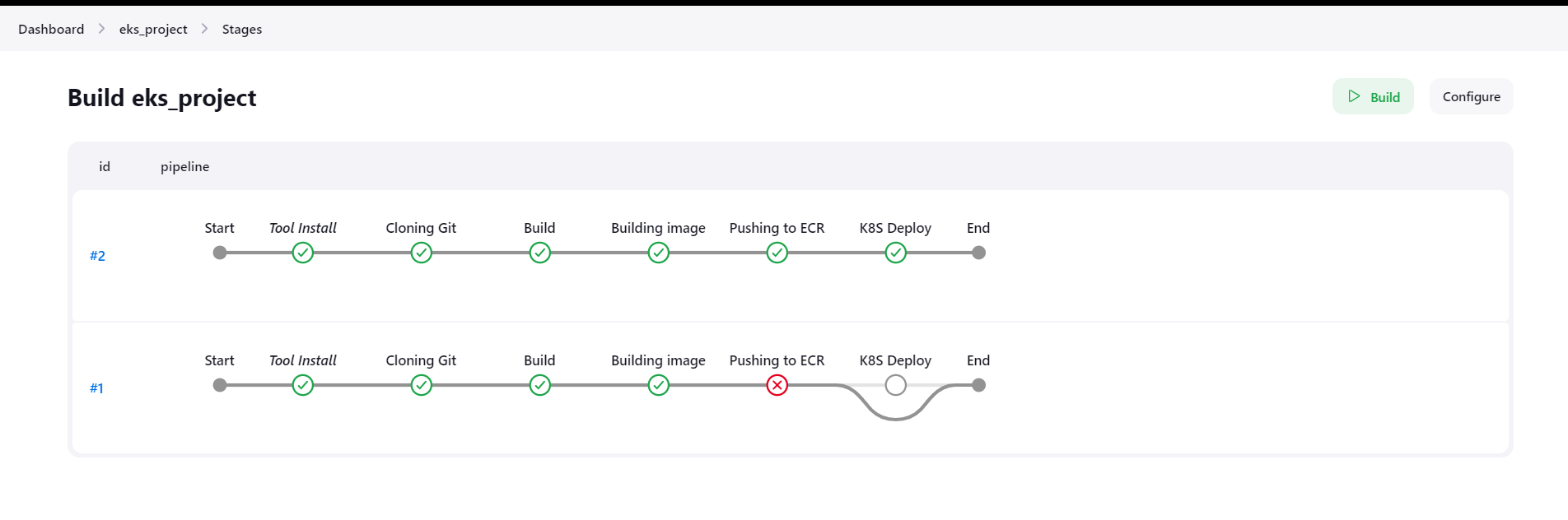
**}**

**}**

**}**

**Red color-should change according to your configuration details**



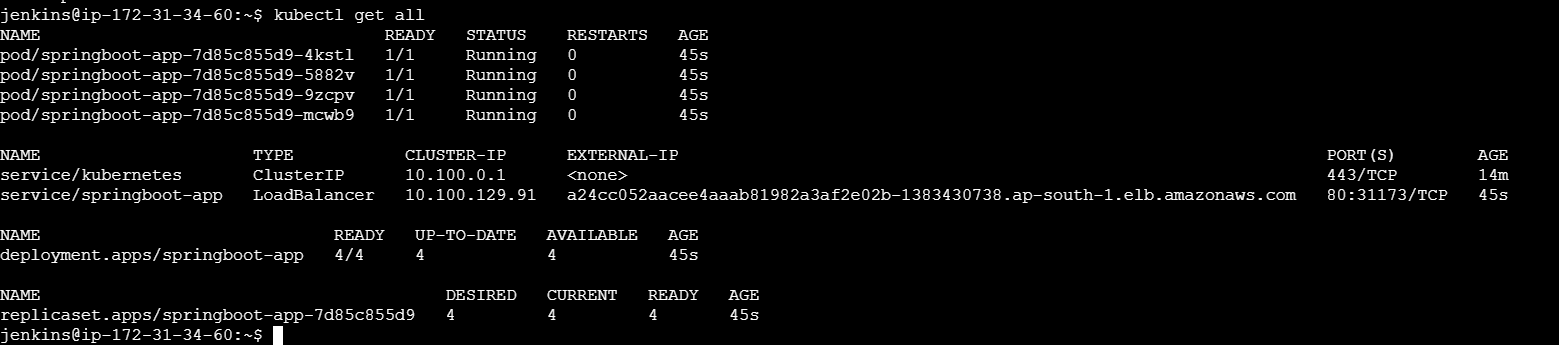


**execute the below command to login as jenkins user.**

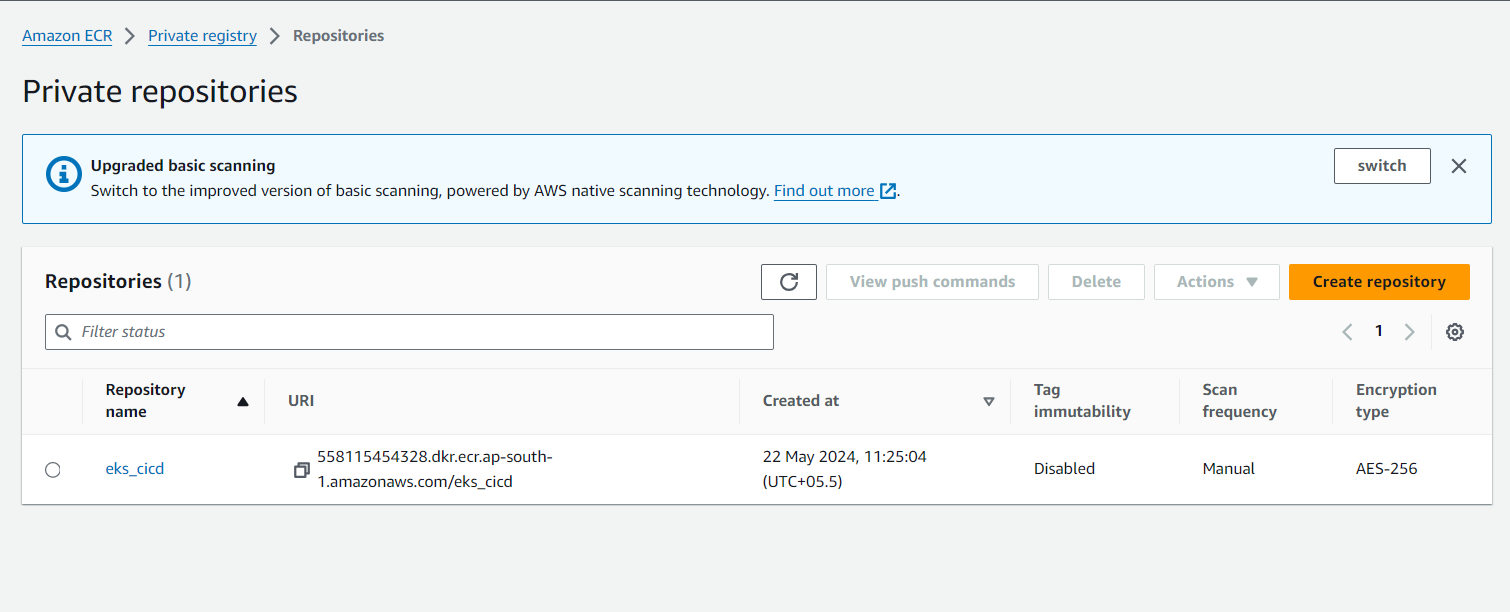
**sudo su - jenkins**

**you should see the nodes running in EKS cluster.**

**kubectl get nodes**



**DOCKER IMAGE PUSHED VIA JENKINS INTO AMAZON ECR**



**ACCESS YOUR APPLICATION USING LOAD BALANCER**

