**Problem Statement**: To create and configure a simple CI/CD pipeline using Jenkins. And the pipeline should be able to get triggered when any new commit come-up to the Version Control System and Web App will be update.

**Implementation of CICD pipeline:**

**What is CI/CD?**

CI/CD stands for continuous integration and continuous delivery. CI/CD refer to the automation. Continuous Integration is the ability to keep integrating code in your application, this means to push new features and/or fixes to your repo every day/week or any other period you want to. And Continuous Delivery is the ability to continuous release new versions of your software safely and quickly.

I used the AWS Cloud to create a CI/CD pipeline and deploy the Web Application.

**I complete the solution of this problem in some staps-**

1. **Firstly, I launched the RedHat EC2 Instance. In this instance, I setup the Docker and Jenkins for create the image and pipeline.**

Setup the docker using below commands –

#yum install -y yum-utils

#yum-config-manager --add-repo <https://download.docker.com/linux/centos/docker-ce.repo>

#yum install docker-ce -y

#systemctl start docker

Setup the jenkins using below commands –

#yum install wget -y

#wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat/jenkins.repo

#rpm --import <https://pkg.jenkins.io/redhat/jenkins.io.key>

#yum install -y jenkins java-11-openjdk-devel

#systemctl start Jenkins

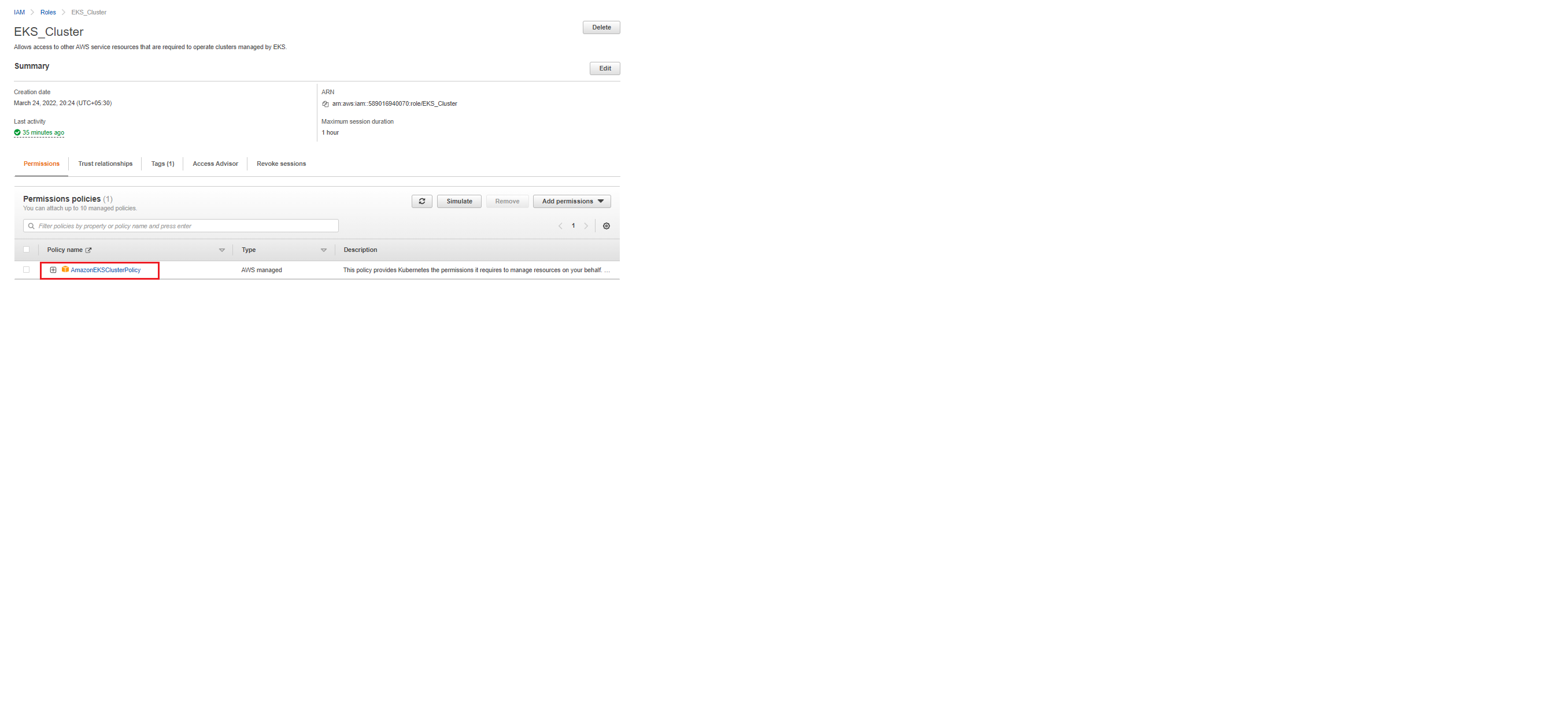
1. **Now, create the EKS Cluster for managing and deploy the application using Kubernetes.**

Steps for creating the EKS Cluster-

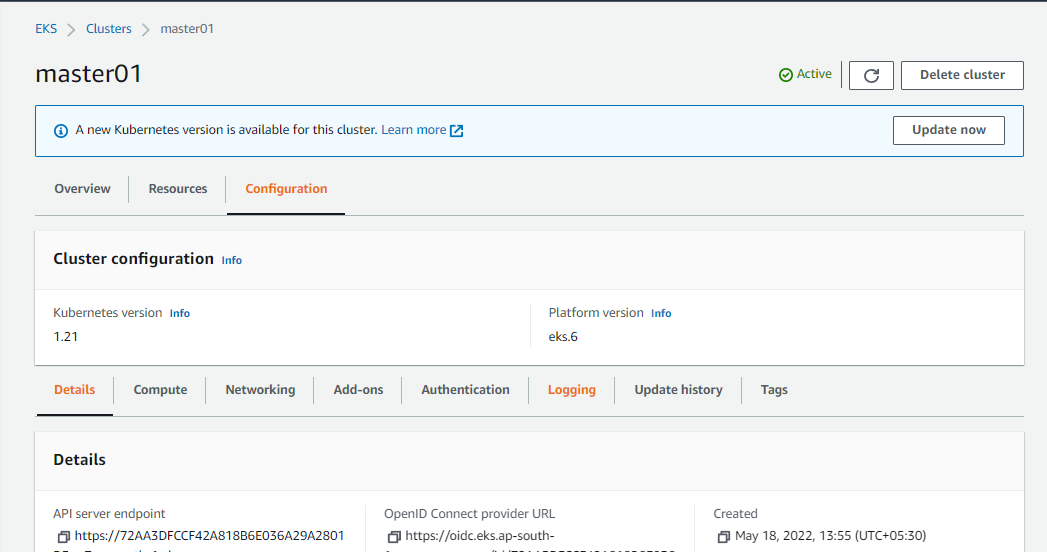
* First, I created the IAM role (EKS\_Cluster) for EKS cluster. And I give the policy permission AmazonEKSClusterPolicy .

**Note: - To Create IAM Role: -**

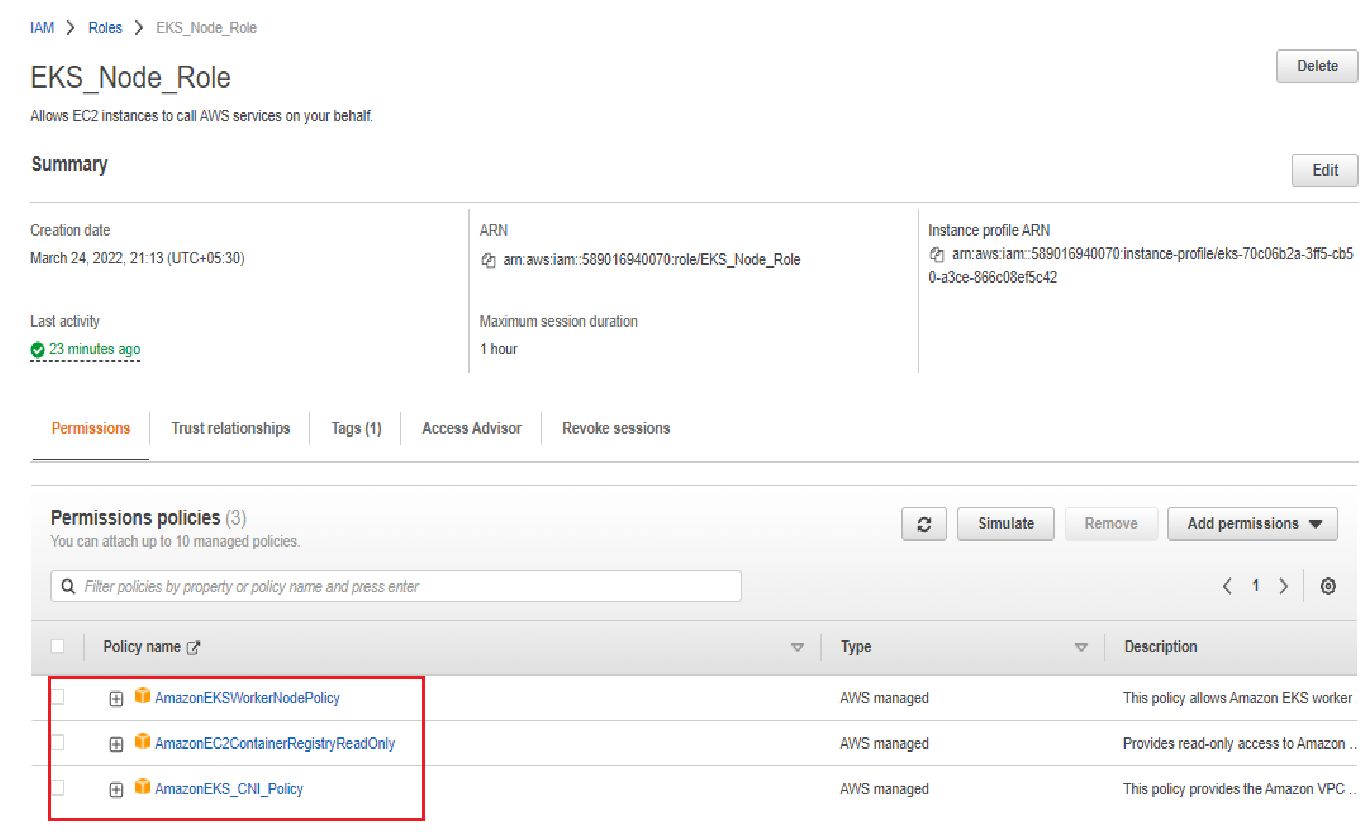
Go to IAM > role > click on create Role > select common Use Case > click next > chose the Policy Permission> click next> Give the Role Name> click create Role



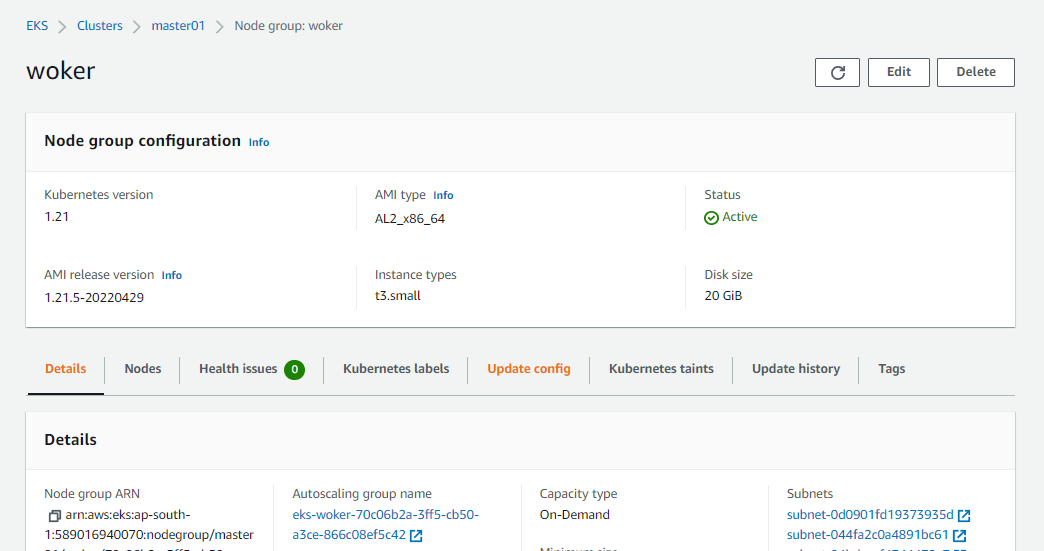
* Now, I created a EKS Cluster name master01. and assigned the IAM EKS\_Cluster role.



* I created again IAM EKS\_Node\_Role role for NodeGroup and give three-policy permission. ( AmazonEKSWorkerNodePolicy , AmazonEKS\_CNI\_Policy, AmazonEC2ContainerRegistryReadOnly)



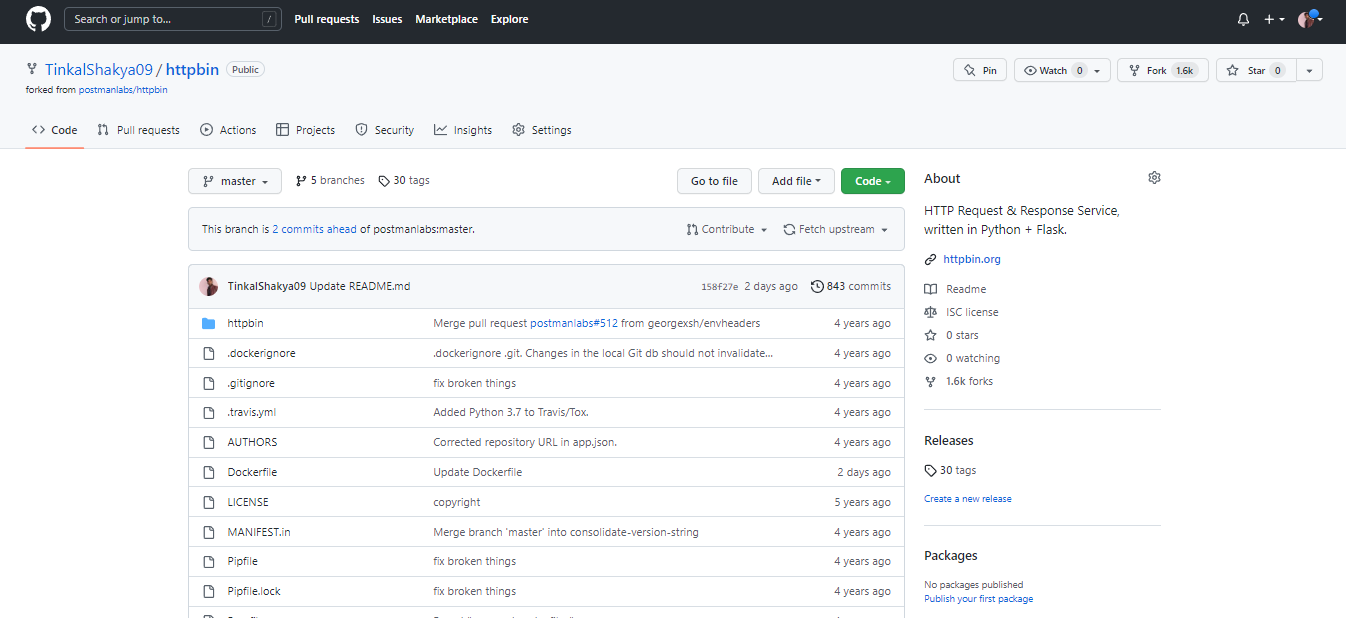
* Then I created a NodeGroup (worker) in the cluster and assign the IAM EKS\_Node\_Role role.

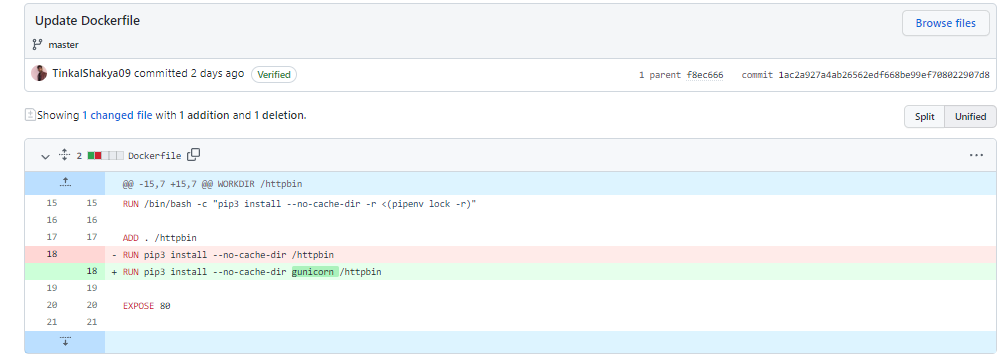


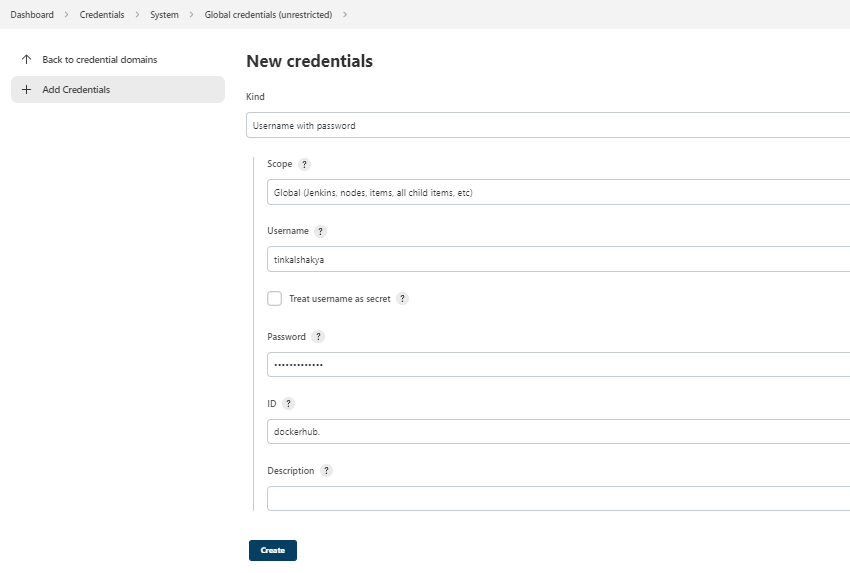
1. **And also install the awscli and kubectl command in the instance for managing the node (**[**https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html**](https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html) **)**
2. **After that connect the cluster to the instance using below command.**

aws eks --region region update-kubeconfig --name cluster\_name

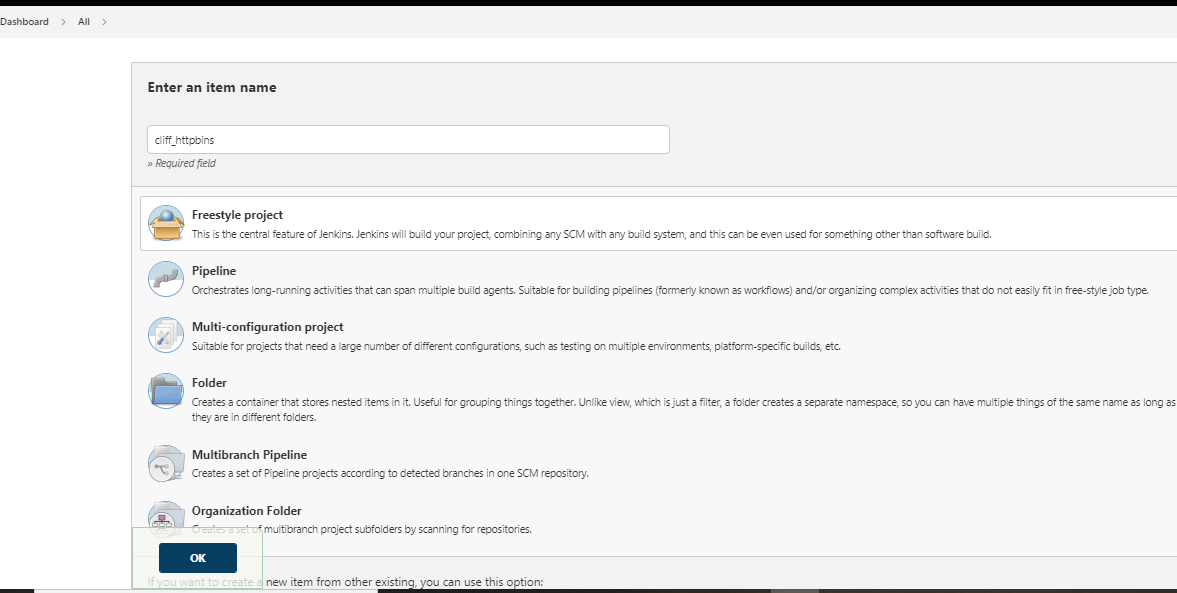
1. **In this step, I have fork the GitHub repository from the given URL in the assessment.**



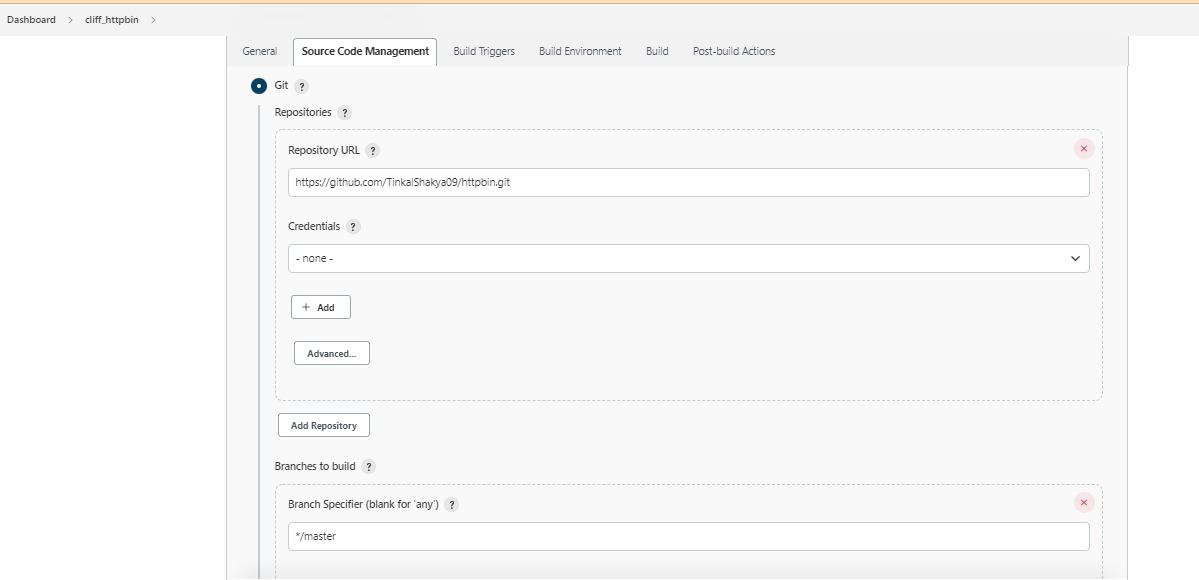
1. **And I had do some changes in the dokerfile.**
2. **In the Jenkins, I installed docker & git plugins. and set the docker credential.**



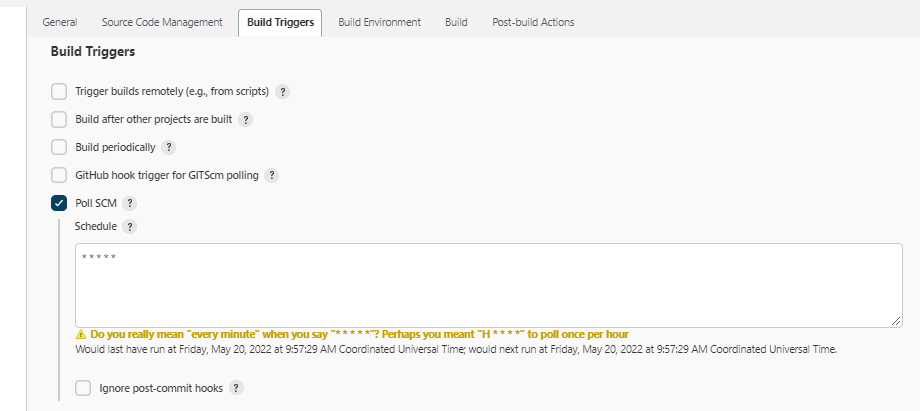
1. **Now, create a job.**



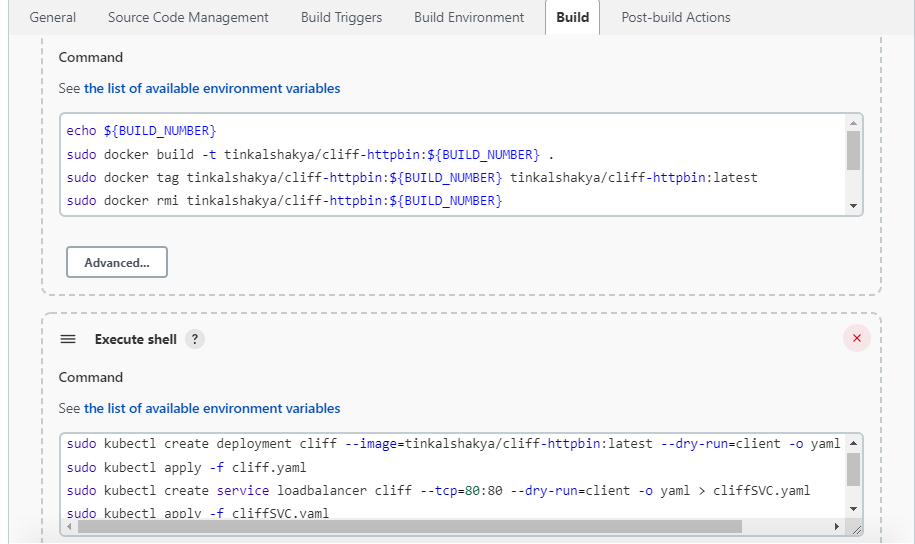
1. **In this job, Go to SCM section and add the git repository URL.**



1. **Go to Build Trigger section and add the Poll SCM for getting the GitHub data on the real time.**



1. **Now, go to the Build section and add Execute Shell and type commands.**



1. **And save the configuration and click on ‘build now’ for testing the pipeline is creating fine or not.**
2. **If pipeline is creating fine then go to the instance and type below command for getting the URL and hit the URL on the Browser.**

**Components used:**

1. Docker
2. Kubernetes
3. AWS
4. Jenkins
5. GitHub

**Jenkins URL:**

<http://13.235.48.119:8080/job/Cliff-httpbin>

**URI of Hosting Web Application:**

<http://aa618c940b0c5459f82662806a6b32a8-1557049052.ap-south-1.elb.amazonaws.com/>

**End Note:** In this document, I have Successfully developed CI/CD pipeline using Jenkins. And the pipeline should be If any changes on the GitHub Repo from the developer side than webapp automatically update on the client-side server.

**Created By- Tinkal Shakya**