How to setup Gitlab (Runner) on K8:

Prerequisites(Make a Script and Run):

# Download and install kubectl

`curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl

chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin/kubectl

​

# Verify the installation of kubectl

kubectl help

​

# Download and install aws-iam-authenticator

curl -o aws-iam-authenticator https://amazon-eks.s3-us-west-2.amazonaws.com/1.13.7/2019-06-11/bin/linux/amd64/aws-iam-authenticator

chmod +x ./aws-iam-authenticator

mkdir -p $HOME/bin && cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$HOME/bin:$PATH

echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc

​

#Verify the installation of aws-iam-authenticator

aws-iam-authenticator help`

​

# Install the awscli

sudo yum install python-pip awscli

​

# Verify the installation

aws --version

​

# Download and configure eksctl

curl --silent --location "https://github.com/weaveworks/eksctl/releases/download/latest\_release/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

​

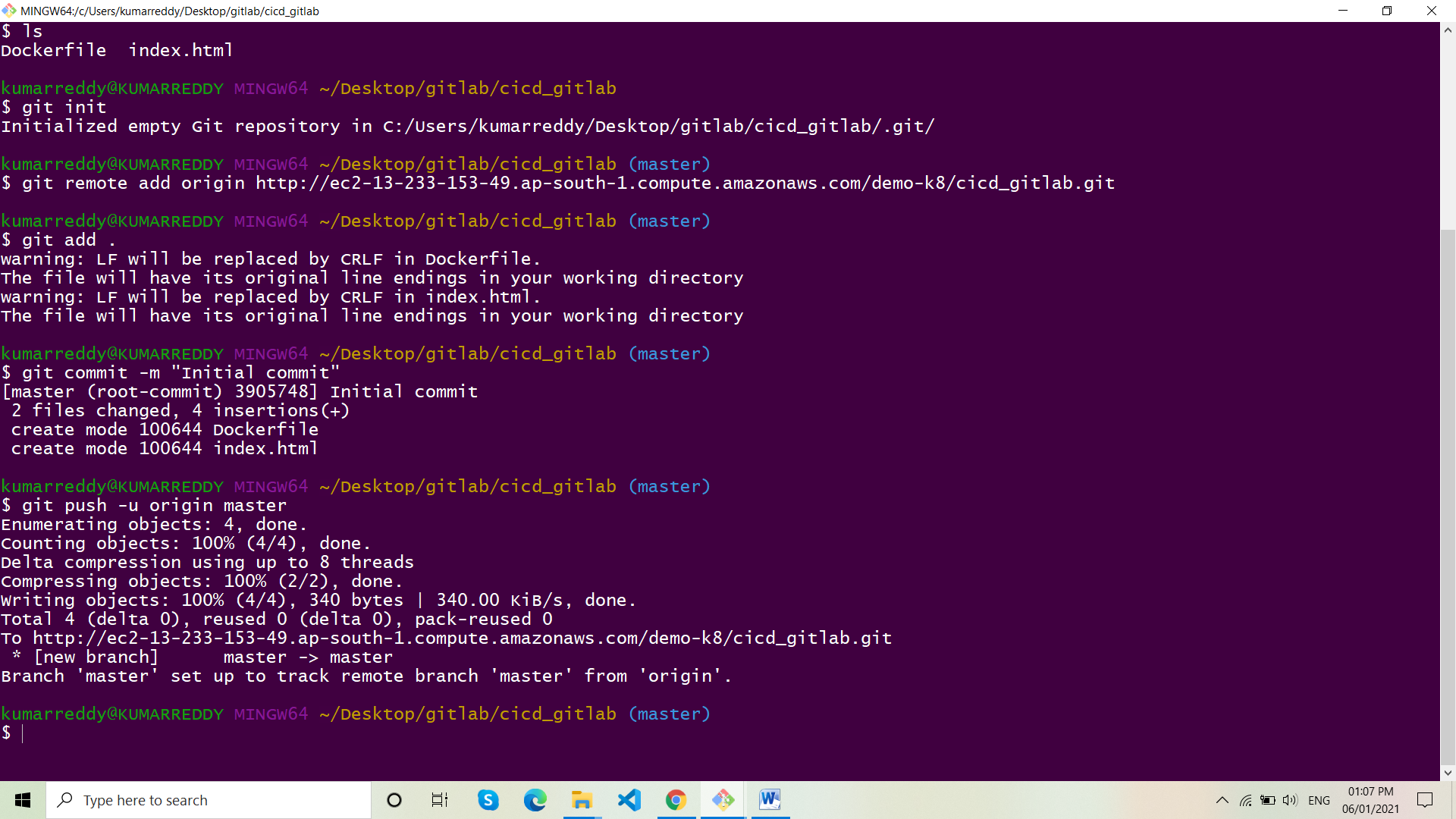
sudo mv /tmp/eksctl /usr/local/bin

​

# Verify the installation

eksctl help

1. Create a Group in Gitlab and Create a Blank Project
2. Write a code in simply index.html + Dockerfile and push the codes to gitlab repo through git commands



1. Follow the Above Screenshot Commands for reference.

git init

git remote add origin <http://ec2-13-233-153-49.ap-south-> 1.compute.amazonaws.com/demo-k8/cicd\_gitlab.git

git add .

git commit -m "Initial commit"

git push -u origin master

NOTE:

We are Integrating gitlab with two protocols named as SSH and HTTP. For Http we don’t required anything. For SSH we need ssh-keygen and add that pub key in gitlab under settings 🡪 SSH keys🡪You add it.

Configure Helm and values.yaml

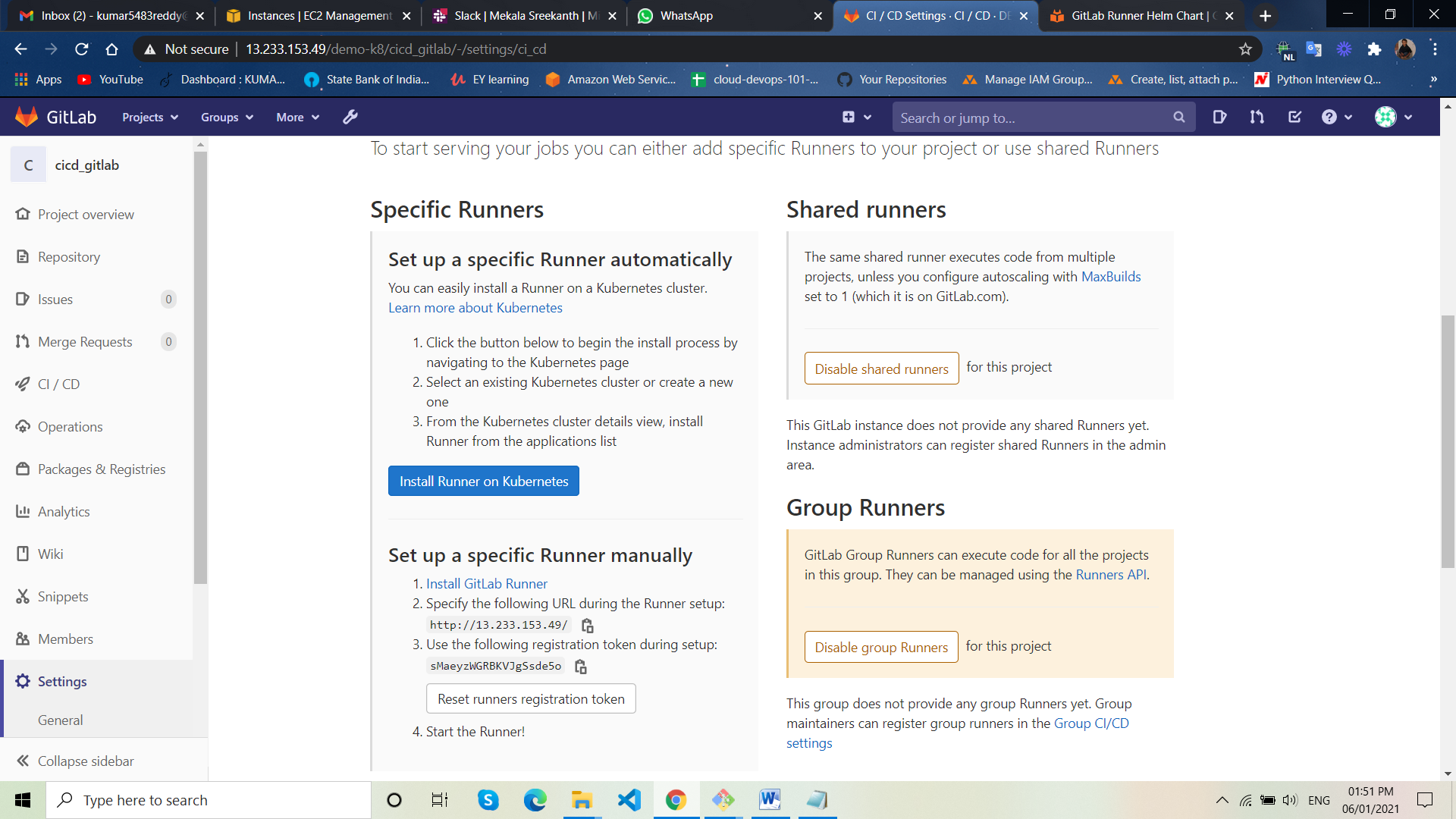
1. wget https://gitlab.com/gitlab-org/charts/gitlab-runner/-/raw/master/values.yaml
2. curl -fsSL -o get\_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3
3. chmod 755 get\_helm.sh
4. ./get\_helm.sh
5. Helm

Edit the values.yaml and configure the below given values

1. enable the gitlab url : <http://gitlab.etechee.com/> or EC2 endpoint.
2. runnerRegistrationToken: "7m63UHzeevmcz1yvnWQ4"

Note: token should get from gitlab console. Follow below steps for token.

1. Goto your project 🡪settings🡪CICD--> specific runners in Runners area
2. Refer the below screenshot



101 rbac:  
102   create: true

108   clusterWideAccess: true

203   privileged: true

211   namespace: gitlab

1. helm repo add gitlab https://charts.gitlab.io
2. helm install --namespace gitlab gitlab-runner -f values.yaml gitlab/gitlab-runner

Create a local .gitlab-ci.yaml file and copy paste the below content and you will make changes.

stages:  
  - build  
  - deploy

image: [349948765351.dkr.ecr.ap-south-1.amazonaws.com/baseimage:latest](http://349948765351.dkr.ecr.ap-south-1.amazonaws.com/baseimage:latest)

variables:  
  DOCKER\_HOST: <tcp://localhost:2375/>  
  DOCKER\_DRIVER: overlay2  
  DOCKER\_TLS\_CERTDIR: ""  
  AWS\_ACCESS\_KEY\_ID: $AWS\_ACCESS\_KEY\_ID  
  AWS\_SECRET\_ACCESS\_KEY: $AWS\_SECRET\_ACCESS\_KEY  
services:  
  - name: docker:19.03.1-dind  
    entrypoint: ["env", "-u", "DOCKER\_HOST"]  
    command: ["dockerd-entrypoint.sh"]

build:  
  stage: build  
  script:  
    - aws ecr get-login-password --region $AWS\_DEFAULT\_REGION | docker login --username AWS --password-stdin $AWS\_REPO\_URL  
    - docker build -t $appname:$CI\_PIPELINE\_ID .  
    - docker tag $appname:$CI\_PIPELINE\_ID $AWS\_REPO\_URL/$appname:$CI\_PIPELINE\_ID  
    - docker push $AWS\_REPO\_URL/$appname:$CI\_PIPELINE\_ID

deploy:  
  stage: deploy  
  script:  
    #Download deployment.yaml  
    #in deployment replace the image tag  
    - aws eks --region $AWS\_DEFAULT\_REGION update-kubeconfig --name dev1  
    - kubectl get nodes

 #kubectl apply