Step 1:

How to setup Gitlab.

<https://repost.aws/questions/QUbvKhK1LQQTO4hzlz0-pP8w/how-to-install-gitlab-on-amazon-linux-2023-x86>

sudo su -

dnf install postfix -y

systemctl enable postfix

systemctl start postfix

curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.rpm.sh | sudo bash

sudo GITLAB\_ROOT\_PASSWORD="Venkat@123" EXTERNAL\_URL="http:// 13.235.103.220" dnf install -y gitlab-ee

gitlab-ctl reconfigure

To run as docker container

Run GitLab Container:

bash

Copy code

docker run -d --name gitlab \

-p 80:80 -p 443:443 -p 22:22 \

--restart always \

--volume /srv/gitlab/config:/etc/gitlab \

--volume /srv/gitlab/logs:/var/log/gitlab \

--volume /srv/gitlab/data:/var/opt/gitlab \

gitlab/gitlab-ce:latest

This command starts a GitLab container, mapping ports 80, 443, and 22, and creating volumes for configuration, logs, and data.

Run GitLab Runner:

bash

Copy code

docker run -d --name gitlab-runner \

--restart always \

--volume /srv/gitlab-runner/config:/etc/gitlab-runner \

--volume /var/run/docker.sock:/var/run/docker.sock \

gitlab/gitlab-runner:latest

This starts a GitLab Runner container, linking it to the Docker socket for handling jobs

Now login to gitlab server.

http://<ip address of instance>

Also you can create gitlab account in <https://gitlab.com>

Step 2:

First we create one project for k8s connection.

New Project -> create project - > give project name, select public -> click on create project.

A screenshot of a computer

Description automatically generated

Step 2.1:

Now we setup connection between K8s cluster and gitlab.

To connect a Kubernetes cluster to GitLab, you must first [install an agent in your cluster](https://docs.gitlab.com/ee/user/clusters/agent/install/index.html).

## Installation steps

To install the agent in your cluster:

1. [Create an agent configuration file](https://docs.gitlab.com/ee/user/clusters/agent/install/index.html#create-an-agent-configuration-file).
2. [Register the agent with GitLab](https://docs.gitlab.com/ee/user/clusters/agent/install/index.html#register-the-agent-with-gitlab).
3. [Install the agent in your cluster](https://docs.gitlab.com/ee/user/clusters/agent/install/index.html#install-the-agent-in-the-cluster).
4. Create an agent config file.

Go inside k8s-connection project we have created in above step and create a “.gitlab/agents/<agent-name>/config.yaml” file.

A screenshot of a computer

Description automatically generated

Give the file name and click on create. You can leave the file blank for now, and [configure it](https://docs.gitlab.com/ee/user/clusters/agent/install/index.html#configure-your-agent) later.

“.gitlab/agents/k8s-connections/config.yaml

A yellow and black label

Description automatically generated

1. Now we can register agent with gitlab.

You must register an agent before you can install the agent in your cluster. To register an agent:

1. On the left sidebar, select **Search or go to** and find your project. If you have an [agent configuration file](https://docs.gitlab.com/ee/user/clusters/agent/install/index.html#create-an-agent-configuration-file), it must be in this project. Your cluster manifest files should also be in this project.
2. Select **Operate > Kubernetes clusters**.
3. Select **Connect a cluster (agent)**.
   * If you want to create a configuration with CI/CD defaults, type a name.
   * If you already have an [agent configuration file](https://docs.gitlab.com/ee/user/clusters/agent/install/index.html#create-an-agent-configuration-file), select it from the list.
4. Select **Register an agent**.
5. GitLab generates an access token for the agent. You need this token to install the agent in your cluster.
6. Copy the command under **Recommended installation method**. You need it when you use the one-liner installation method to install the agent in your cluster.

Go to operator -> Kubernetes cluster -> connect a cluster

A screenshot of a computer

Description automatically generated

From dropdown select the repo we have created and click on register.

A screenshot of a computer

Description automatically generated

We need to follow the below steps in our cluster server.

A screenshot of a computer

Description automatically generated

Login to our Kubernetes cluster server and install the connections.

A screen shot of a computer

Description automatically generated

After installing the connection we can see the connection is active.

A screenshot of a computer

Description automatically generated

Step 3:

We need one registry to move our image. We can use github, ECR, gitlab registry. We will use gitlab registry. We need to create one in our gitlab.

Below is manual steps. We can do the same using CI/CD job.

Gitlab account -> Go inside our repo -> Deploy -> Container registry

A screenshot of a computer

Description automatically generated

A white line on a white background

Description automatically generated

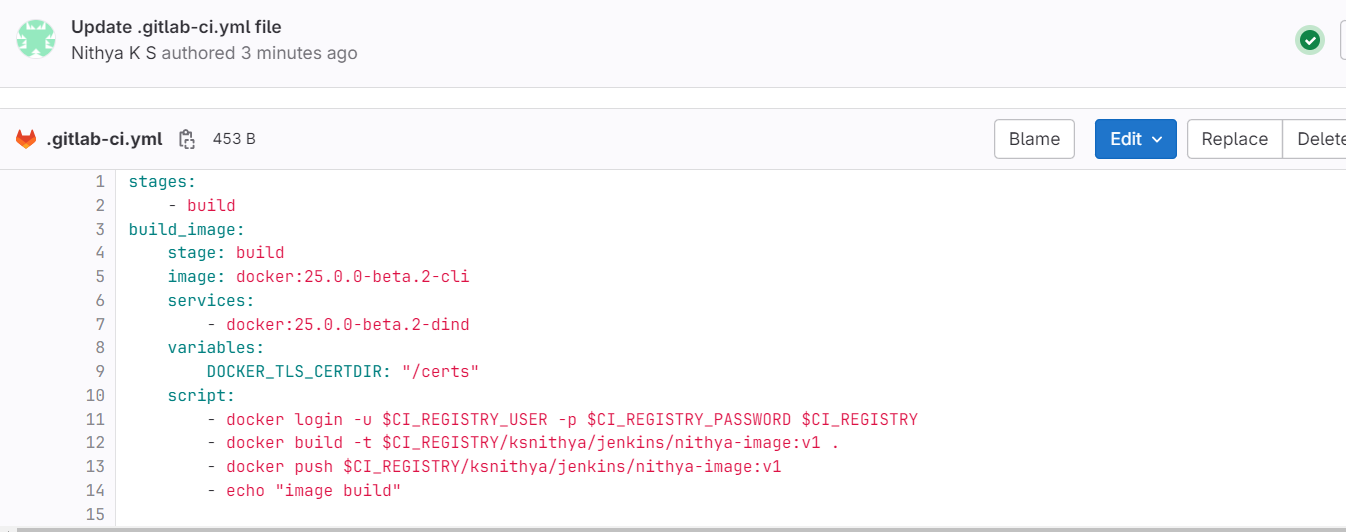
A screenshot of a computer

Description automatically generated

Step 4:

First we need to create .gitlab-ci.yml in our repository. This file is indicate we ci/cd pipeline is there. Below code will create image and push to gotlab registry.

We are using gitlab registry so we can use predefined variable to call username, password and registry URL.

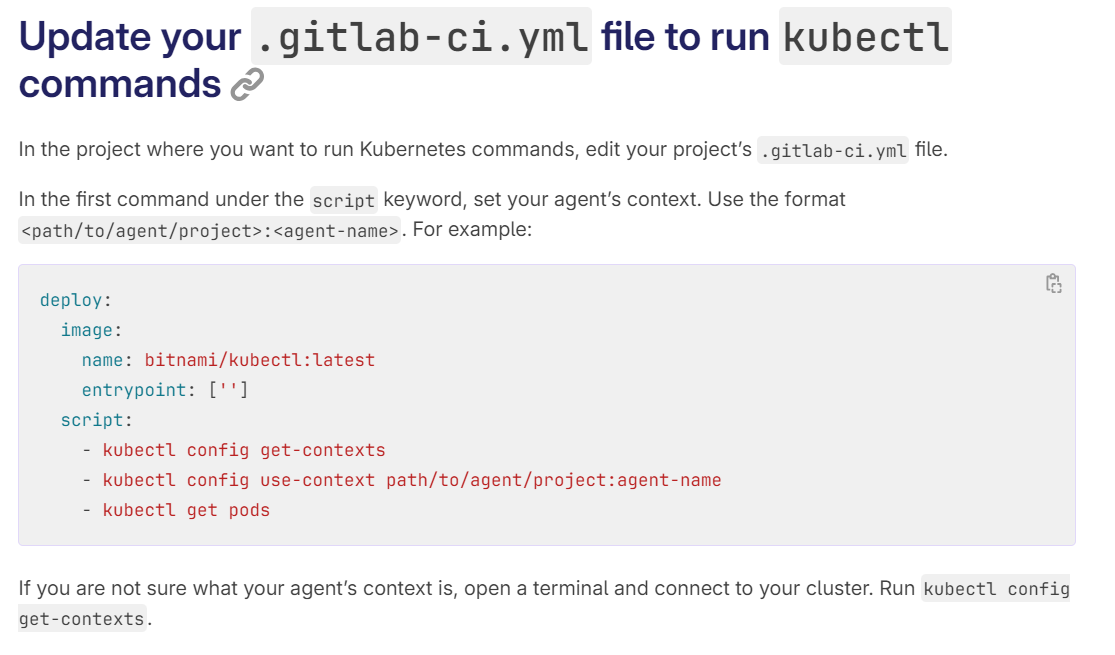
 Step 5:

Now we can setup gitlab to use Kubernetes cluster.

Use below link for steps:

<https://docs.gitlab.com/ee/user/clusters/agent/ci_cd_workflow.html>

Use the sample code in below and we modify as per our requirements.



Below is our modified code.

A screenshot of a computer

Description automatically generated

Step 6:

Now we need to authenticate k8s-connections to our repo where our code exists.

This can be done by adding the permission in k8s-connections repo.

.gitlab/agents/k8s-connections/config.yaml

Steps available in below link:

<https://docs.gitlab.com/ee/user/clusters/agent/ci_cd_workflow.html>

A screenshot of a computer

Description automatically generated

A white rectangular object with red text

Description automatically generated

Now save the file.

Step 7:

Now goto our repo where our code exist and update as below and save it.

Below is our full updated code. Kubernetes code is available in same repo under eks directory.

A screenshot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

Full code available in below repo.

<https://gitlab.com/ksnithya/jenkins.git>