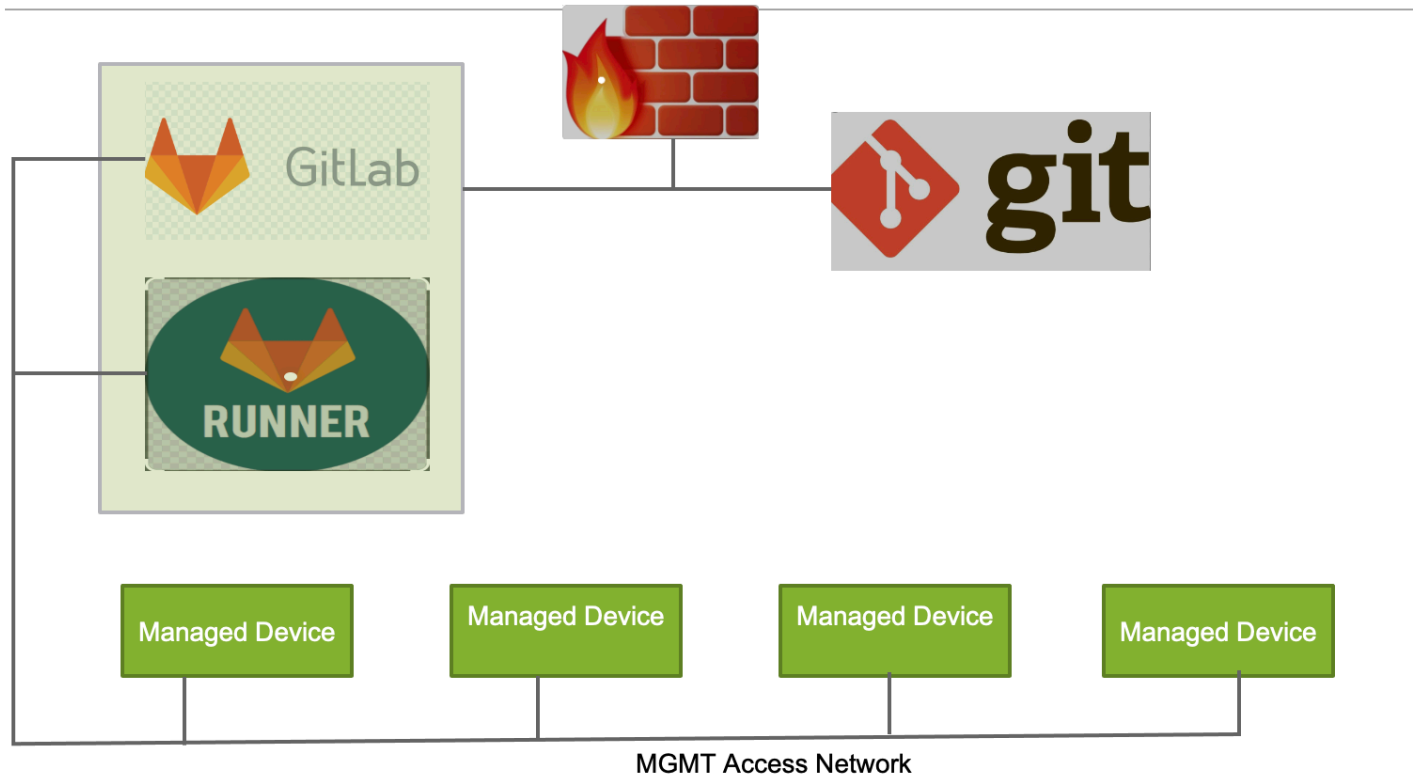
Gitlab Local Instance Bring UP, it's Integration with gitlab-runner and setting up docker-executor

- This document is intended for readers who have working knowledge of git and some know how of gitlab CI/CD concepts.
- The aim of this document is to demonstrate , Gitlab Local Instance Bring UP, its integration with gitlab-runner and setting up docker-executor.

Lab Diagram



Gitlab VM Bring UP (executed from KVM Host)

```
node_name=gitlab
vcpus=4
vram=8096
node_suffix=knawaz.lab.jnpr
root_password=gitlab123
gitlab_password=gitlab123
export LIBGUESTFS_BACKEND=direct
cloud_image=/var/lib/libvirt/images/CentOS-7-x86_64-GenericCloud.qcow2
qemu-img create -f qcow2 /var/lib/libvirt/images/${node_name}.qcow2 100G
virt-resize --expand /dev/sda1 ${cloud_image} /var/lib/libvirt/images/${node_name}.qcow2

virt-customize  -a /var/lib/libvirt/images/${node_name}.qcow2   --run-command 'xfs_growfs /'   --root-password password:${root_passw

virt-install --name ${node_name}   --disk /var/lib/libvirt/images/${node_name}.qcow2   --vcpus=${vcpus}   --ram=${vram}   --network
```

Environment Specific Files

```
[gitlab@gitlab ~]$ cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
PEERDNS="yes"
IPV6INIT="no"
IPADDR=192.168.3.20
NETMASK=255.255.255.0
GATEWAY=192.168.3.1

[gitlab@gitlab ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
```

```
::1         localhost localhost.localdomain localhost6 localhost6.localdomain

192.168.3.20 gitlab.knawaz.lab.jnpr gitlab
192.168.3.21 gitlab-runner.knawaz.lab.jnpr gitlab-runner

[gitlab@gitlab ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 1.1.1.1
```

Installing Required Packages in Gitlab VM

```
yum -y install curl policycoreutils openssh-server openssh-clients postfix vim firewalld lynx mutt
systemctl start postfix && systemctl enable postfix && systemctl start firewalld && systemctl enable firewalld
curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh | bash
yum -y install gitlab-ce
```

Adding Firewall Rules

```
firewall-cmd --permanent --add-service http && firewall-cmd --permanent --add-service https
firewall-cmd --reload && firewall-cmd --list-all
```

Setting UP TLS

```
openssl genrsa -out ca.key 2048
openssl req -new -sha256 -key ca.key -subj "/CN=GITLAB-KNAWAZ-CA" -out ca.csr
openssl req -x509 -sha256 -days 365 -key ca.key -in ca.csr -out ca.pem
openssl genrsa -out gitlab.knawaz.lab.jnpr.key 2048
openssl req -new -sha256 -key gitlab.knawaz.lab.jnpr.key -subj "/CN=gitlab.knawaz.lab.jnpr" -out gitlab.knawaz.lab.jnpr.csr
openssl x509 -req -in gitlab.knawaz.lab.jnpr.csr -CAcreateserial -CAserial ca.seq -sha256 -days 365 -CA ca.pem -CAkey ca.key -out gi
```

Creating Chain Certificate

```
cat gitlab.knawaz.lab.jnpr.pem ca.pem > /etc/gitlab/ssl/gitlab.knawaz.lab.jnpr.pem
cp gitlab.key /etc/gitlab/ssl/
cp ca.key  ca.pem /etc/gitlab/ssl/
chmod 600 /etc/gitlab/ssl/*
```

Updating Config File

- Update / edit following lines in /etc/gitlab/gitlab.rb as per your setup

```
vim /etc/gitlab/gitlab.rb
external_url 'https://gitlab.knawaz.lab.jnpr'
nginx['redirect_http_to_https'] = true
nginx['ssl_certificate'] = "/etc/gitlab/ssl/gitlab.knawaz.lab.jnpr.pem"
nginx['ssl_certificate_key'] = "/etc/gitlab/ssl/gitlab.knawaz.lab.jnpr.key"
```

Adding Config to Setup Local Docker Registry

```
################################################################################
## Container Registry settings
##! Docs: https://docs.gitlab.com/ee/administration/container_registry.html
################################################################################

registry_external_url 'https://gitlab.knawaz.lab.jnpr:5050'
registry_nginx['ssl_certificate'] = "/etc/gitlab/ssl/gitlab.knawaz.lab.jnpr.pem"
registry_nginx['ssl_certificate_key'] = "/etc/gitlab/ssl/gitlab.knawaz.lab.jnpr.key"
firewall-cmd --add-port=5050/tcp --permanent
firewall-cmd --reload
```

Re-Configure Gitlab

- Be patient, as this step will take some time

```
gitlab-ctl reconfigure
```

Access Gitlab GUI

- Get root user password

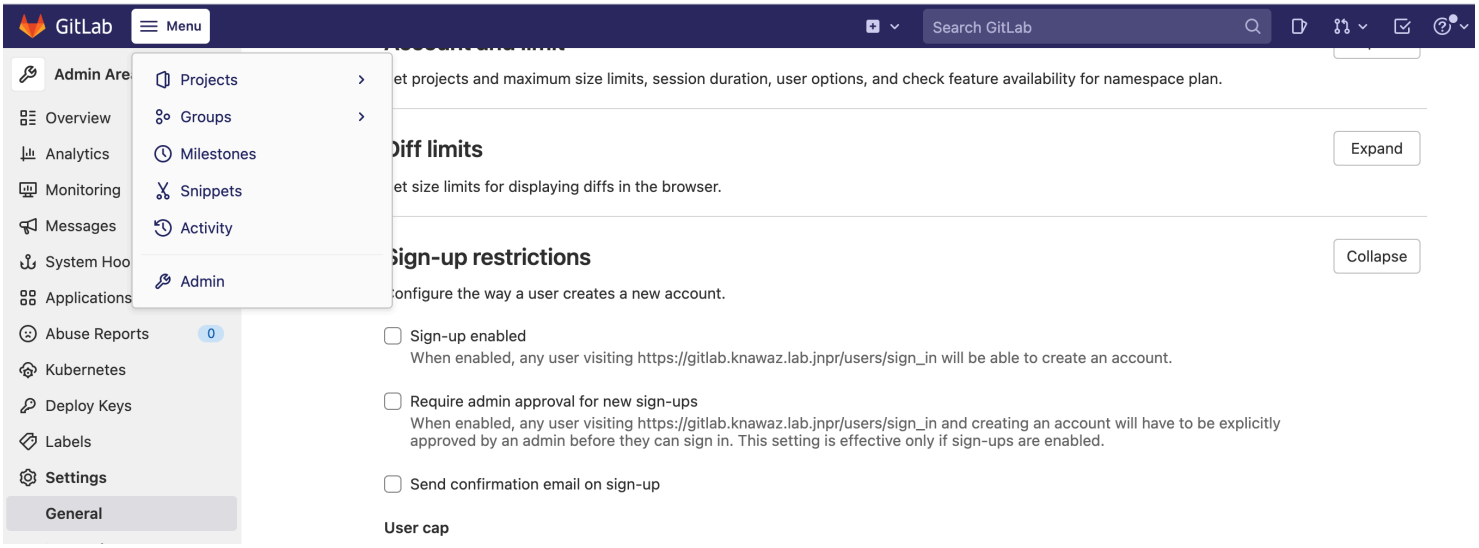
```
cat /etc/gitlab/initial_root_password
```

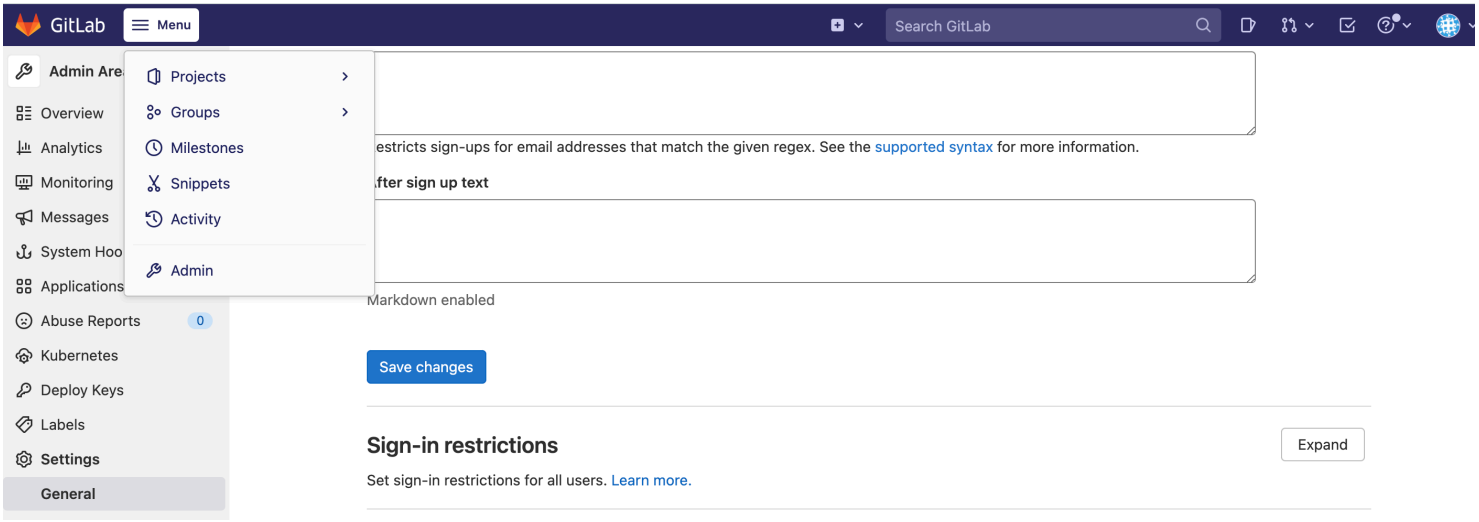
- Access the Gitlab GUI via your favourite browser, but don't forget to add the self-signed certificate as trusted in your browser.
- Login via root user.

```
https://gitlab-URL-AS-PER-YOUR-DNS-ENTRY
```

Disabling the Singup Option

- Disabling the signup option is an important step.





Creating Users, Groups and Projects

- For the sake of brevity, I am avoiding adding screenshots from my setup.
- Visit the following link to view the relationship between groups, users, and projects.

```
https://docs.gitlab.com/ee/user/group/
```

Gitlab-runner VM Bring Up (executed from KVM host)

```
node_name=gitlab-runner
vcpus=8
vram=16192
node_suffix=knawaz.lab
root_password=gitlab123
gitlab_password=gitlab123
export LIBGUESTFS_BACKEND=direct
cloud_image=/var/lib/libvirt/images/CentOS-7-x86_64-GenericCloud.qcow2
qemu-img create -f qcow2 /var/lib/libvirt/images/${node_name}.qcow2 150G
virt-resize --expand /dev/sda1 ${cloud_image} /var/lib/libvirt/images/${node_name}.qcow2
virt-customize  -a /var/lib/libvirt/images/${node_name}.qcow2   --run-command 'xfs_growfs /'   --root-password password:${root_passw
brctl show
virt-install --name ${node_name}   --disk /var/lib/libvirt/images/${node_name}.qcow2   --vcpus=${vcpus}   --ram=${vram}   --network
```

Environment Specific Files

```
[gitlab@gitlab-runner ~]$ cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="static"
ONBOOT="yes"
```

```
TYPE="Ethernet"
USERCTL="yes"
PEERDNS="yes"
IPV6INIT="no"
IPADDR=192.168.3.21
NETMASK=255.255.255.0
GATEWAY=192.168.3.1

[gitlab@gitlab-runner ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.3.20 gitlab.knawaz.lab.jnpr gitlab
192.168.3.21 gitlab-runner.knawaz.lab.jnpr gitlab-runner

[gitlab@gitlab-runner ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 1.1.1.1
```

Installing Docker

```
curl -fsSL https://get.docker.com/ | sh
systemctl status docker
systemctl enable docker
sudo systemctl start docker
```

Installing Gitlab-runner

```
curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.rpm.sh" | sudo bas
sudo yum install gitlab-runner -y

mkdir -p /etc/gitlab-runner/certs
```

Getting TLS Chain Certificate from Gitlab Server

```
scp gitlab@192.168.3.20:/etc/gitlab/ssl/gitlab.knawaz.lab.jnpr.pem /etc/gitlab-runner/certs/
```

Update CA-TRUST in Gitlab-Runner VM

CA-TRUST-UPDATE

```
  openssl s_client -connect gitlab.knawaz.lab.jnpr:443 <<<'' | openssl x509 -out /etc/pki/ca-trust/source/anchors/gitlab.knawaz.lab.
  update-ca-trust enable
  update-ca-trust extract
  chmod u+w $(readlink /etc/pki/tls/certs/ca-bundle.crt)
  echo >> $(readlink /etc/pki/tls/certs/ca-bundle.crt)
  echo "gitlab.knawaz.lab.jnpr" >> $(readlink /etc/pki/tls/certs/ca-bundle.crt)
  cat /etc/pki/ca-trust/source/anchors/gitlab.knawaz.lab.jnpr.crt >> $(readlink /etc/pki/tls/certs/ca-bundle.crt)
  chmod u-w $(readlink /etc/pki/tls/certs/ca-bundle.crt)
  systemctl restart gitlab-runner
  systemctl restart docker
```

Verfying Connection to Gitlab Docker Registry

gitlab Container Registry

```
docker login gitlab.knawaz.lab.jnpr:5050 -u knawaz -p Zvoy8TG9LsoRsBXncjbv
``
## Verfying Connection to Gitlab Server


echo | openssl s_client -CAfile /etc/gitlab-runner/certs/gitlab.knawaz.lab.jnpr.pem -connect gitlab.knawaz.lab.jnpr:443


## Registering Gitlab-runner with Gitlab

* The assumption is that a group is created, a user is mapped to the group, and a project is created under the group.
* In my case, group name is labs and project name is test.
* Click on the Gitlab icon on the left to see a list of your projects.
![project_list](./Images/project_list.png)
* Go to the relevant project where you want to set up CI/CD and click on Setting > CI/CD.
![project_ci_cd_settings](./Images/project_cicd_setting.png)
* In the left window, click on the Expand button against Runners.
![runner_url_token](./Images/runner_token.png)
```

gitlab-runner register --url https://gitlab.knawaz.lab.jnpr/ --registration-token $REGISTRATION_TOKEN --tls-ca-file /etc/gitlab-runner/certs/gitlab.knawaz.lab.jnpr.pem

```
* With the above command, the URL is already provided as a parameter, so just press enter at the URL prompt.
* With the above command, the registration-token is already provided as a parameter, so just press enter at the registration-token p
* Specify an appropriate tag on the prompt (tags are important so that CI/CD jobs can be linked to the registered runner).
* At the prompt, specify "docker" as the executor type.
* You must specify the default docker image for CI/CD (I provided alpine: latest).

## Edit Gitlab-runner Config
* You may need to mount /etc/hosts to allow Docker containers to resolve gitlab urls if your /etc/resolve.conf is unable to do so.
* It is also necessary to mount the /etc/gitlab-runner/certs directory so that the docker container can access the full chain certif
```

vim /etc/gitlab-runner/config.toml [runners.docker] volumes = ["/cache", "/etc/gitlab-runner/certs/gitlab.knawaz.lab.jnpr.pem:/etc/gitlab-runner/certs/gitlab.knawaz.lab.jnpr.pem:ro"] volumes = ["/cache", "/etc/hosts:/etc/hosts:ro"]

```
## Running the Simple Pipeline

### Create .gitlab-ci.yml
```

---

image: "alpine:latest" stages:

- build build: stage: build script:
  - echo "Hello world" tags:
  - ci

```
### Execution
```

git add .gitlab-ci.yml git commit -m'added gitlab-ci' git push -u origin main

```
### Check Pipline Status
* On the Gitlab UI, select your project , CI/CD > Pipelines.
* If the pipeline status is "Passed" , say hurray, else someone else needs to look into your setup.


## Ref

(https://gitlab.com/gitlab-org/gitlab-runner/-/issues/2659)
(https://gitlab.com/gitlab-org/gitlab-runner/-/issues/2950)
(https://gitlab.com/gitlab-org/gitlab-runner/-/issues/27062)
(https://docs.gitlab.com/runner/configuration/tls-self-signed.html#supported-options-for-self-signed-certificates-targeting-the-gitl
(https://docs.gitlab.com/runner/configuration/tls-self-signed.html)
(https://docs.gitlab.com/omnibus/settings/nginx.html#update-the-ssl-certificates)
(https://docs.gitlab.com/omnibus/settings/ssl.html)
(https://docs.gitlab.com/runner/executors/)
(https://www.howtoforge.com/tutorial/how-to-install-and-configure-gitlab-ce-on-centos-7/)
```