

Gitlab Runner with K3d

Table of Contents

Requirements	1
Gitlab preparations	1
Gitlab Group	1
Runner	2
K3s cluster preparations	2
Prepare to use gitlab runner helm chart	3
Install runner with a helm chart	3

Requirements

Plan is to have gitlab runner working on your laptop/machine connected to your personal group in the Gitlab. This document is valid for most Linux based systems and MacOS.

For this Workshop you need those tools to be installed:

- Docker
- Brew
- kubectl
- kubectx
- helm
- k3d

You can find how to install required tools for this part of workshops here: <https://local-workshops-requirements-guide-b9c7de.gitlab.io>

Gitlab preparations

I'm assuming that you will be using official Gitlab which is available here: <https://gitlab.com> but you can use any on premise Gitlab instance. You will just need to modify some examples below if your on premise instance will be used.

Gitlab Group

Please create group which will be used for this example, you can also reuse existing one. For a purpose of this manual I'll use group name in format of `username-workshops` which in my case is `devops-training-info`.

Next step will be to disable instance runners for this group. In my case I can do this just by going here: https://gitlab.com/groups/devops-training-info/-/settings/ci_cd expanding **Runners** section and disabling it here.

Another step is to take **Registration token** which in my place can be accessed here: <https://gitlab.com/groups/devops-training-info/-/runners> by clicking on **Create group runner** button. You should have at least one runner for untagged jobs - please add a checkbox for **Run untagged jobs**, put more details if you need and click **Create runner** submit button.

Please stay in the result page or in a secure way store a runner authentication token which starts with: **glrt-** at the moment of writing this documentation.

Runner

K3s cluster preparations

First we need to prepare our K3s cluster which will be used for our Gitlab group and we will be using to install our Gitlab Runner^[1].

```
k3d cluster create gitlabRunnerCluster
```

The result of it should look like this:

```
INFO[0000] Prep: Network
INFO[0000] Created network 'k3d-gitlabRunnerCluster'
INFO[0000] Created image volume k3d-gitlabRunnerCluster-images
INFO[0000] Starting new tools node...
INFO[0001] Creating node 'k3d-gitlabRunnerCluster-server-0'
INFO[0001] Pulling image 'ghcr.io/k3d-io/k3d-tools:5.4.9'
INFO[0003] Pulling image 'docker.io/rancher/k3s:v1.25.7-k3s1'
INFO[0006] Starting Node 'k3d-gitlabRunnerCluster-tools'
INFO[0014] Creating LoadBalancer 'k3d-gitlabRunnerCluster-serverlb'
INFO[0016] Pulling image 'ghcr.io/k3d-io/k3d-proxy:5.4.9'
INFO[0024] Using the k3d-tools node to gather environment information
INFO[0025] Starting new tools node...
INFO[0025] Starting Node 'k3d-gitlabRunnerCluster-tools'
INFO[0027] Starting cluster 'gitlabRunnerCluster'
INFO[0027] Starting servers...
INFO[0028] Starting Node 'k3d-gitlabRunnerCluster-server-0'
INFO[0035] All agents already running.
INFO[0035] Starting helpers...
INFO[0036] Starting Node 'k3d-gitlabRunnerCluster-serverlb'
INFO[0042] Injecting records for hostAliases (incl. host.k3d.internal) and for 3
network members into CoreDNS configmap...
INFO[0045] Cluster 'gitlabRunnerCluster' created successfully!
INFO[0045] You can now use it like this:
```

```
kubectl cluster-info
```

After successfully created cluster we need to

```
kubectl create namespace gitlab-runner
```

Prepare to use gitlab runner helm chart

There is full documentation connected to this process available here: <https://docs.gitlab.com/runner/install/> but we will be using only part connected with K8s installation available here: <https://docs.gitlab.com/runner/install/kubernetes.html>

Please prepare to use helm chart based on the upper documentation but **DO NOT** install/upgrade it

In basics the most importing steps are currently looking like this:

Add the GitLab Helm repository:

```
helm repo add gitlab https://charts.gitlab.io
```

Update your repo to be sure you are using the latest one:

```
helm repo update gitlab
```

Install runner with a helm chart

For this purpose we will prepare `gitlab-runner-values.yaml` file which will be looking like this:

```
image:
  registry: registry.gitlab.com
  image: gitlab-org/gitlab-runner

imagePullPolicy: IfNotPresent

## How many runner pods to launch.
replicas: 1

## The GitLab Server URL (with protocol) that want to register the runner against
## ref: https://docs.gitlab.com/runner/commands/README.html#gitlab-runner-register
##
gitlabUrl: https://gitlab.com/

## The Runner Token for adding new Runners to the GitLab Server. This must
## be retrieved from your GitLab instance. It is the token of an already registered
runner.
```

```

## ref: (we don't have docs for that yet, but we want to use an existing token)
##
runnerToken: "<put-your-token-here>"
#

## Configure the maximum number of concurrent jobs
## ref: https://docs.gitlab.com/runner/configuration/advanced-configuration.html#the-
global-section
##
concurrent: 2

## Defines in seconds how often to check GitLab for a new builds
## ref: https://docs.gitlab.com/runner/configuration/advanced-configuration.html#the-
global-section
##
checkInterval: 30

## For RBAC support:
rbac:
  create: true

## Configure ServiceAccount
##
serviceAccount:
  ## Specifies whether a ServiceAccount should be created
  ##
  ## TODO: Set default to `false`
  create: true

```

There is much more parameters we can set and which can be check just by analyzing default values available here: <https://gitlab.com/gitlab-org/charts/gitlab-runner/-/blob/main/values.yaml>

Please remember to change `<put-your-token-here>` to yours **runner authentication token**.

```

helm upgrade -i --namespace gitlab-runner gitlab-runner -f gitlab-runner-values.yaml
--set runnerToken=$RUNNER_TOKEN gitlab/gitlab-runner

```

Output should look like this:

```

Release "gitlab-runner" does not exist. Installing it now.
NAME: gitlab-runner
LAST DEPLOYED: Mon May 20 18:48:29 2024
NAMESPACE: gitlab-runner
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Your GitLab Runner should now be registered against the GitLab instance reachable at:
"https://gitlab.com/"

```

Runner namespace "gitlab-runner" was found in runners.config template.

You can check if all is OK by first checking if pod is OK

```
kubectl get po --namespace gitlab-runner
```

NAME	READY	STATUS	RESTARTS	AGE
gitlab-runner-869cff885f-r5g9j	1/1	Running	0	3m38s

and then checking if runner is registered here: <https://gitlab.com/groups/devops-training-info/-/runners>

You can now start to use your own runners - CONGRATULATIONS!



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

[1] Gitlab Runner <https://docs.gitlab.com/runner/>