

GitLab Cache

➤ Installing **Gitlab-Runner**

Follow the Step to install **Gitlab-Runner**

Link: <https://artifacthub.io/packages/helm/gitlab/gitlab-runner/0.45.0>

- ❖ Open Gitlab-Runner Helm-Chart and change Values.yaml as shown below.
- ❖ Installing the Chart use following command:

- `helm repo add gitlab http://charts.gitlab.io/`
- `helm repo update`
- `helm upgrade --install [RELEASE_NAME] gitlab/gitlab-runner -n gitlab-runner -f values.yaml`

Changes made in **Gitlab-Runner** Helm Chart are listed Below.

1. Comment Image and registry part in Values.yaml

```
# image:
#   registry: registry.gitlab.com
#   image: gitlab-org/gitlab-runner
#   tag: alpine-v11.6.0
```

2. Add GitlabUrl in Values.yaml

```
gitlabUrl: https://gitlab.com
```

3. Add RunnerRegistrationToken in Values.yaml

```
runnerRegistrationToken: "GR1348*****PmEnikza"
```

4. Change rbac.create=true in Values.yaml

```
rbac:
  create: true
```

5. Change rbac.clusterWideAccess=true in Values.yaml

```
clusterWideAccess: true
```

6. Change service.enabled=true in Values.yaml

```
## prometheus-operator serviceMonitor
service:
  enabled: true
```

7. Add runners.tags in Values.yaml

```
tags: "cache-demos"
```

8. Add runners.privileged=true in Values.yaml

```
privileged: true
```

9. Add runners.cache in Values.yaml

Note: Create Secret Named s3access and add data accesskey & secretkey

```
cache:
  cacheType: s3
  cacheShared: true
  s3ServerAddress: s3.amazonaws.com
  s3BucketName: gitlab-cache-crictracker
  s3BucketLocation: ap-south-1
  s3CacheInsecure: false


  ## GCS settings
  secretName: s3access
  ## Use this line for access using gcs-access-id and gcs-private-key
  # secretName: gcsaccess
```

10. Adding Node-Selector in Values.yaml

```
nodeSelector:
  gitlab-runner: cache
```

➤ After Deploying **Gitlab-Runner** Now configure **.gitlab-ci.yml** to create/use Cache.

1. Add cache path in .gitlab-ci.yml

```
 .gitlab-ci.yml 10.35 KiB
```

```
1  # Caches
2  .node_modules-cache: &node_modules-cache
3    key:
4      files:
5        - package-lock.json
6    paths:
7      - ./node_modules
8    policy: pull-push
9
```

2. Add Variables like Below to fast DOWNLOAD and UPLOAD of Cache from S3.

```
9
10 variables:
11   FF_USE_FASTZIP: "true"
12   ARTIFACT_COMPRESSION_LEVEL: "fast"
13   CACHE_COMPRESSION_LEVEL: "fast"
14   TRANSFER_METER_FREQUENCY: 5s
15
```

3. Add Tags to Run job on Specific Runner.

```
26
27 lint-check:
28   stage: lint-check
29   tags:
30     - cache-demos #demo
31
```

4. Using Cache in specific Jobs with Policy

```
88     cache:
89     # - key:
90     #     files:
91     #         - package-lock.json
92     #     paths:
93     #         - ./node_modules
94     - <<: *node_modules-cache
95     policy: pull
```

5. Add this 2 Variables in Jobs will SpeedUp docker Build

→ `DOCKER_TLS_CERTDIR: ""`

→ `DOCKER_BUILDKIT: '1'`

```
100     services:
101     #- docker:18.09-dind
102     - name: docker:20.10.6-dind
103       command: ["--tls=false"]
104     variables:
105       DOCKER_HOST: tcp://localhost:2375
106       DOCKER_DRIVER: overlay2
107       DOCKER_TLS_CERTDIR: ""
108       DOCKER_BUILDKIT: '1'
```

6. Add “--build-arg BUILDKIT_INLINE_CACHE=1” in docker Build CMD

```
157     script:
158     #deploy.
159     - time docker build --build-arg BUILDKIT_INLINE_CACHE=1 --build-arg NODE_ENV=development --build-arg REACT_APP_API_URL=
160     - docker tag crictracker-website:dev_${CI_COMMIT_SHA} 693583001687.dkr.ecr.ap-south-1.amazonaws.com/crictracker-website
```