【GitLab】runner 使用 k8s 流程說明

一、前置條件

- 已有可用的 **Kubernetes 叢集**（kubectl/helm 可連通）
- GitLab 可連到 Runner（你的 GitLab URL 例：`http://10.2.11.139`）
- 在 GitLab 介面建立 Runner（Project/Group/Instance → CI/CD → Runners →**New runner**），取得 **Authentication Token（`glrt-...`）**

二、建立命名空間

```
kubectl create namespace gitlab-runner     # 放 Runner 本體
kubectl create namespace ci-jobs           # job 跑的位置（也可與上面同一 ns）
```

三、用 Helm 裝 Runner（K8s executor）

1. 加入 chart

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

2. 建一份 `values.yaml`

```
gitlabUrl: "http://10.2.11.139/"
# 新流程建議用 runnerToken（glrt-...）
runnerToken: "glrt-xxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

rbac:
  create: true

serviceAccount:
  create: true
  name: gitlab-runner

runners:
  executor: kubernetes
  # job 要跑在哪個 ns；不填則跟 Runner 同 ns
  namespace: ci-jobs
  # 預設 job 容器用什麼 image（可被 .gitlab-ci.yml 的 image 覆蓋）
  image: alpine:3.20
  pollTimeout: 180

  # 是否允許特權容器（多數情況不需要，建議先關）
  privileged: false

  # 幫 job Pod 加上標籤（可選）
  podLabels:
    app: gitlab-ci-job

  # 預設資源限制（可選）
  resources:
    limits:
```

```
      cpu: "1"
      memory: "1Gi"
    requests:
      cpu: "200m"
      memory: "256Mi"

  # Cache（可選：接 S3/MinIO；此段示意，若未啟用請刪除）
  cache:
    type: s3
    path: "gitlab-runner"
    s3ServerAddress: "minio.minio.svc.cluster.local:9000"
    s3BucketName: "gitlab-ci-cache"
    s3AccessKey: "minio-access"
    s3SecretKey: "minio-secret"
    s3Secure: false
```

> 新流程下：**tags / run-untagged / locked** 等屬性建議在 **GitLab UI** 的 Runner 設定頁調整，不再用 CLI 參數寫入。

3. 安裝

```
helm upgrade --install gitlab-runner gitlab/gitlab-runner \
  -n gitlab-runner -f values.yaml
```

4. 驗證

```
kubectl -n gitlab-runner get pods
# 應看到 gitlab-runner 的 Pod（或 deployment/rs）
```

四、最小 `.gitlab-ci.yml`（驗證 K8s executor 正常）

```
stages: [ping]

hello:
  stage: ping
  image: alpine:3.20
  script:
    - echo "Runner: $CI_RUNNER_DESCRIPTION"
    - echo "Executor: Kubernetes OK"
    - cat /etc/os-release
```

跑起來時的 log 開頭應出現：**Preparing the "kubernetes" executor**
K8s 裡 `ci-jobs` 命名空間會看到對應的 **Pod**（生命周期：Running → Succeeded）。

五、在 K8s executor 內建 Docker 映像（推薦用 Kaniko）

K8s executor 不建議 DinD。常見做法是用 **Kaniko**（rootless，無需 Docker daemon）。

1. 在 GitLab 專案 CI Variables 設定：

- `CI_REGISTRY`（你的 registry 位址）
- `CI_REGISTRY_IMAGE`（目標 repo）
- `CI_REGISTRY_USER` / `CI_REGISTRY_PASSWORD`（登入憑證）

2. Kaniko Job 範例

```
stages: [build]

docker-build:
  stage: build
  image:
    name: gcr.io/kaniko-project/executor:latest
    entrypoint: [""]
  variables:
    # registry 登入（若用 GitLab Container Registry，可改為 CI 提供的 token）
    DOCKER_CONFIG: /kaniko/.docker
  script:
    - >
      mkdir -p /kaniko/.docker &&
      cat > /kaniko/.docker/config.json <<EOF
      {
        "auths": {
          "${CI_REGISTRY}": {
            "auth": "$(printf "%s:%s" "$CI_REGISTRY_USER" "$CI_REGISTRY_PASSWORD" | base64 -w0)"
          }
        }
      }
      EOF
    - echo -e 'FROM alpine:3.20\nCMD ["echo","kaniko build OK"]' > Dockerfile
    - >
      /kaniko/executor
      --context "${CI_PROJECT_DIR}"
      --dockerfile "${CI_PROJECT_DIR}/Dockerfile"
      --destination "${CI_REGISTRY_IMAGE}:$CI_COMMIT_SHA"
      --destination "${CI_REGISTRY_IMAGE}:latest"
```

> 這個 job 會在 K8s 中跑 Kaniko，把映像打好推到 Registry，**不需要** Docker daemon、也不需要 privileged。

六、（可選）Sidecar 服務示例（K8s 會幫你生多容器 Pod）

```
stages: [test]

pg-test:
  stage: test
  image: postgres:16-alpine
  services:
    - name: postgres:16-alpine
      alias: db
  variables:
    POSTGRES_PASSWORD: secret
  script:
    - psql -h db -U postgres -c "SELECT version();"
```

> 在 K8s executor 下，`services:` 會轉成 **同一個 Pod 的 sidecar 容器**，網路互通用 `alias` 連線。

七、常見疑難排解

- **Pipeline 卡 Pending**
  - 確認 Runner 在 GitLab UI 顯示*Online*、tags 是否匹配（或允許 untagged）。
- **Pod 起不來 / CrashLoopBackOff**
  - `kubectl -n ci-jobs describe pod <pod>` 看 event / 拉不到 image / 權限問題。
- **需要特權容器**
  - 在 `values.yaml` 的 `runners.privileged: true`（謹慎評估安全）。
- **Cache/S3 連線**
  - 先用最小流程跑通，再逐一加上 cache 設定；MinIO/GCS/S3 憑證與位址要正確。

八、如果不用 Helm（純手工方式，了解即可）

- 部署一個 `gitlab/gitlab-runner` 的 Deployment
- 用 ConfigMap/Secret 提供 `config.toml`（`executor = "kubernetes"`、`[runners.kubernetes]` 參數）
- 綁一個 ServiceAccount + RBAC（允許建立/刪除 Pod）

> **但維運上 Helm chart 會簡單很多（升級/回滾/佈署差異）。**

需要我幫你把 values.yaml 客製到你的環境（例如 MinIO 位址、特定節點選擇器、Pod 安全性設定、預設資源）嗎？我可以直接產一份可用的版本給你。

修訂版本 #2
★由 treeman 建立於 14 K 2025 11:45:11
由 treeman 更新於 14 K 2025 11:46:17