DEVOPS CHEAT SHEET

GitLab Runner hardening checklist

dominik-dev.pl
full article →

Own VPS, Docker executor, glrt- token, protected deploy runner

TL;DR Warm caches, no minutes, private deploys. It runs pipeline code with your secrets: own machine, `docker` executor, `privileged = false`, deploys on a **protected** runner.

REQUIREMENTS

- `glrt-` token from the UI. Registration tokens: off since GitLab 17.0, removal scheduled for 20.0.
- Runner `major.minor` = GitLab's. Here: `alpine-v19.5.3` (2026-11-10).
- Linux + Docker Engine, sized for `concurrent` x heaviest job.
- Outbound HTTPS only, **no inbound port**. Not on the GitLab or production app host.

BUILDING IMAGES

- **DinD:** `privileged = true`, `/certs/client`, `DOCKER_TLS_CERTDIR: "/certs"`. Breakout risk.
- **Socket binding:** jobs drive the host daemon = root on the host.
- **Buildah:** daemonless. **Kaniko:** archived 3 June 2025, check upstream first.
- Privileged builds: own protected runner, separate VM.

WHEN NOT TO Low build volume. Nobody patches the box. Public projects or forks. Burst concurrency or many architectures.

REGISTER

```
docker compose run --rm gitlab-runner register \
  --non-interactive \
  --url "https://gitlab.com" \
  --token "$RUNNER_TOKEN" \
  --executor docker \
  --docker-image alpine:3.22 \
  --description "vps-runner-1"
```

Then `chmod 600 config/config.toml`, `docker compose up -d`. Tags, protected, timeout: in the UI.

CONFIG.TOML

```
concurrent = 2
listen_address = ":9252"
[[runners]]
  executor = "docker"
  limit = 2
  request_concurrency = 2
  [runners.docker]
    privileged = false
    volumes = ["/cache"]
    pull_policy = ["always", "if-not-present"]
    memory = "4g"
    cpus = "2"
```

Area	Check
Executor	`docker`, not `shell`; `privileged = false` (else root on the host); `memory`, `cpus`
Images	`allowed_images`, `allowed_services`, `allowed_pull_policies`; no bare `if-not-present` if shared
Deploy	Runner tagged `deploy`, **Protected**; `tags: [deploy]`; `main`, `develop`, variables protected
Tags	Runner needs **all** job tags; no run untagged = untagged jobs stuck
Host	Socket mount only where nothing else runs; `config.toml` `chmod 600`, out of git
Cache	Local `/cache`; S3 from runner two; measure `queued_duration` vs `duration`
Metrics	`/metrics` on `127.0.0.1:9252` (no auth); alert on target down
Disk	Weekly `clear-docker-cache`; `docker system prune` for images; next run is cold
Upgrades	Pause in UI, bump tag, `docker compose pull && docker compose up -d`