DEVOPS CHEAT SHEET

GitHub runner hardening checklist

dominik-dev.pl
full article →

Ephemeral JIT, no Docker socket, separate deploy runner

TL;DR The runner sandboxes nothing. One job per fresh container (JIT, `--rm`), no Docker socket, no `sudo`, OIDC, deploys on their own runner, never on a public repo.

REQUIREMENTS

- Linux (Ubuntu 20.04+, Debian 10+, RHEL 8+...), Windows, macOS 11+; x64, ARM64.
- Outbound HTTPS 443 only, **no inbound port**: `github.com`, `api.github.com`, `*.actions.githubusercontent.com` + cache/update hosts. `./run.sh --check`.
- `--disableupdate`: update within **63 days** of a release or jobs stop.

COMPOSE: EPHEMERAL GITHUB RUNNER

```
github-runner:
  image: ghcr.io/actions/actions-runner:2.338.2
  profiles: [ephemeral]
  command: ["bash", "-c", "exec ./run.sh --
jitconfig \"$$JIT_CONFIG\""]
  environment:
    - JIT_CONFIG
  security_opt:
    - no-new-privileges:true
  cap_drop: [ALL]
```

REGISTER AND JIT

- `./config.sh --url <repo> --token <token> --labels vps-trusted --ephemeral --unattended`, then `sudo ./svc.sh install runner`. Token expires after 1 h.
- Per job, a systemd loop runs `docker compose run --rm -T github-runner`. The image (built for ARC) has passwordless `sudo`: keep `no-new-privileges`.
- JIT: `POST .../actions/runners/generate-jitconfig`, then `./run.sh --jitconfig`. Repo **Administration: write** or org **Self-hosted runners: write**; token stays on the host.

Checklist

Area	Do	Why
Repos	Private repos only; no `pull_request_target`	Fork PRs run code on your host
Lifetime	`--ephemeral` or JIT, new container per job	Reused runners stay compromised
Docker	No socket, no `privileged`, no `docker` group	`docker` group = root-level
Routing	One label per trust level; small runner groups	`runs-on` needs **all** labels
Token	`contents: read`; OIDC `id-token: write`	No long-lived cloud keys
Deploy	`environment:` + separate runner	Rules do not isolate a shared runner
Reviewers	Up to 6, no self-approval	Free/Pro/Team: public repos only
Cache	BuildKit `type=registry`; host dir per trust level	`actions/cache`: GitHub, 10 GB, 7 days
Scale	ARC Helm charts, pod per job	`dind` = privileged pods

TRUST RULE One runner per trust level; the boundary is the machine. A runner with the Docker socket is root on its host and can read every other runner's token there. **When not to:** public repos, low build volume, nobody owns updates, many parallel clean builds (ARC or hosted runners).