



链滴

使用 docker 容器运行 gitlab-runner

作者: [expoli](#)

原文链接: <https://ld246.com/article/1577506005531>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



使用 docker 容器运行 gitlab-runner

1. 拉取镜像

```
docker pull gitlab/gitlab-runner
```

2. 注册 runner

GitLab runner 如果没有指定相应的参数的话、那就是可以在一台机器上注册多个 runner、如果你有台服务器、那么你就可以在性能比较好的机器上多注册几个 runner、提高 GitLab 在进行任务分配的候高性能服务器分配到任务的概率。

2.1 手动注册

```
docker run --rm -t -i -v /srv/gitlab-runner/config:/etc/gitlab-runner gitlab/gitlab-runner register
```

然后根据提示输入相应的信息

2.2 运行时注册

通过运行下面的命令、gitlab-runner 会在宿主机相应的挂载目录写入 runner 的注册信息、可以在 `/srv/gitlab-runner/config` 文件夹下面看到相应的信息、下面的命令每运行一次会注册一个 runner 可以在 GitLab 的 admin 面板看到、但是你会看到 runner 的最后联系时间为 never、这也就代表着现在的些共享 runner 还不能立刻使用、现在只是将相应的信息注册到 GitLab 上而已。

```
docker run -v /srv/gitlab-runner/config:/etc/gitlab-runner --rm --name=gitlab-runner gitlab/gitlab-runner register \
```

```
--non-interactive --executor "docker" \
--docker-image alpine:latest \      #
--url "http://git.shuangzu.top/" \   # gitlab runners 管理界面看到的注册 链接
--registration-token "g*****" \    # gitlab runners 管理界面看到的注册 token
--description "docker-runner" \
--tag-list "docker,106" \
--run-untagged="true" \
--locked="false" \
--access-level="not_protected"
```

4. 启动 GitLab runner 工作容器

下面的这条命令是启动一个 GitLab runner 的正常工作容器、配置文件使用我们原来通过容器注册后下来的配置文件 (-v /srv/gitlab-runner/config:/etc/gitlab-runner)，挂载 docker 的 sock 文件、证容器能够与宿主机的 docker 守护进程进行通信、然后创建 CI/CD 运行时需要的容器 (-v /var/run/docker.sock:/var/run/docker.sock)；如果不加这一条、在进行 CI/CD 构建的时候会遇到无法与 docker 进程通信的报错。

```
docker run \
-v /srv/gitlab-runner/config:/etc/gitlab-runner \
-v /var/run/docker.sock:/var/run/docker.sock \
--name=gitlab-runner \
-d --restart=always \
gitlab/gitlab-runner
```

5. 验证 runner 是否正常工作

```
docker exec -it gitlab-runner /bin/bash
```

```
root@4f6791a5e7bd:/# gitlab-runner
Runtime platform           arch=amd64 os=linux pid=88 revision=ac8e767a vers
on=12.6.0
NAME:
  gitlab-runner - a GitLab Runner

USAGE:
  gitlab-runner [global options] command [command options] [arguments...]

VERSION:
  12.6.0 (ac8e767a)

AUTHOR:
  GitLab Inc. <support@gitlab.com>

COMMANDS:
  exec      execute a build locally
  list      List all configured runners
  run       run multi runner service
  register  register a new runner
  install   install service
  uninstall uninstall service
  start     start service
  stop     stop service
```

```
restart          restart service
status          get status of a service
run-single      start single runner
unregister      unregister specific runner
verify         verify all registered runners
artifacts-downloader download and extract build artifacts (internal)
artifacts-uploader create and upload build artifacts (internal)
cache-archiver  create and upload cache artifacts (internal)
cache-extractor download and extract cache artifacts (internal)
cache-init      changed permissions for cache paths (internal)
health-check    check health for a specific address
help, h        Shows a list of commands or help for one command
```

GLOBAL OPTIONS:

```
--cpuprofile value    write cpu profile to file [CPU_PROFILE]
--debug              debug mode [DEBUG]
--log-format value    Choose log format (options: runner, text, json) [LOG_FORMAT]
--log-level value, -l value Log level (options: debug, info, warn, error, fatal, panic) [LOG_LEVEL]
--help, -h          show help
--version, -v       print the version
```

```
#####
```

```
root@4f6791a5e7bd:/# gitlab-runner verify
```

```
Runtime platform          arch=amd64 os=linux pid=100 revision=ac8e767a version=12.6.0
```

```
Running in system-mode.
```

```
# 可以看到runner工作正常、也可以触发一下CI/CD进行测试。
```

```
Verifying runner... is alive
```

```
runner=TZi33hJ6
```

```
Verifying runner... is alive
```

```
runner=oMzfmS1J
```

```
Verifying runner... is alive
```

```
runner=3nnLNxsQ
```

```
Verifying runner... is alive
```

```
runner=eyudsagy
```