DevOps 笔记

adamanteye

1. CI/CD

1.1. GitHub Security

1.1.1. Dependabot

- 配置 Dependabot 版本更新 - GitHub 文档
- Dependabot 选项参考 - GitHub 文档

1.2. GitHub Actions

- GitHub Actions 的工作流语法 - GitHub 文档

1.2.1. 代码检出

默认的检出方式是浅克隆,即只包含最新提交,如果需要根据提交历史生成修改变更等,应当禁用浅克隆:

```yaml
- uses: actions/checkout@v4
  with:
    fetch-depth: 0
```

1.2.2. Self-hosted Runner

`systemd` 系统服务例子:

```
# /etc/systemd/system/github-actions-runner.service
[Unit]
Description=GitHub Actions Runner Service
After=network-online.target

[Service]
ExecStart=/home/ci-user/actions-runner/run.sh
WorkingDirectory=/home/ci-user/actions-runner
Restart=on-failure
RestartSec=5
User=ci-user
Group=ci-user
Environment="PATH=/home/ci-user/.local/bin:/usr/local/bin:/usr/bin:/bin"
Environment="all_proxy=http://[::1]:10801"

[Install]
WantedBy=multi-user.target
```

允许在 Self-hosted Runner 上运行容器内的作业,前提是容器内安装有对应的软件.例如 cloudflare/wrangler-action 需要 `npm` 以及 `node`.

1.2.3. 复用工作流

- 复用工作流 - GitHub 文档

1.2.4. 其他参考

- The Pain That is Github Actions

2. 自动化测试

2.1. 单元测试

原则

- Protection against regressions
- Resistance to refactoring
- Fast feedback
- Maintainbility

3. 网络基础设施

3.1. Cloudflare

3.1.1. Tunnel

CF Tunnel 可以方便地在中国地区搭建服务,例如:

```yaml
services:
  nginx:
    image: nginx:1.27.5-alpine
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf:ro
      - /srv/:/srv/:ro
    restart: unless-stopped
    networks:
      - tunnel-net

  cloudflared:
    image: cloudflare/cloudflared:latest
    restart: unless-stopped
    command: [ "tunnel", "--no-autoupdate", "run", "--token", "your-token" ]
    depends_on:
      - nginx
    networks:
```

```
      - tunnel-net

networks:
  tunnel-net:
```

注意在面板中为服务 `http://nginx:80` 配置相应的域名.

3.1.2. Workers and Wrangler

更新 wrangler 版本时注意查看 Compatibility flags 以了解是否出现兼容性问题.

4. 监控与日志

Grafana 是开源的数据可视化平台,用于创建仪表盘,展示来自各种数据源的指标.结合数据收集代理 Telegraf 与高性能的时序数据库 InfluxDB,可以完成数据的采集,存储,查询展示的流程.

首先配置 Telegraf 采集数据,例如采集 CPU 负载:

```toml
[agent]
collection_jitter = "20s"
debug = false
hostname = "heloise"
interval = "20s"

# Read metrics about cpu usage
[[inputs.cpu]]
## Whether to report per-cpu stats or not
percpu = false
## Whether to report total system cpu stats or not
totalcpu = true
## Comment this line if you want the raw CPU time metrics
fielddrop = ["time_*"]
```

连接到 InfluxDB:

```toml
[[outputs.influxdb_v2]]
## The URLs of the InfluxDB cluster nodes.
##
## Multiple URLs can be specified for a single cluster, only ONE of the
## urls will be written to each interval.
##   ex: urls = ["https://us-west-2-1.aws.cloud2.influxdata.com"]
urls = ["http://influxdb2:8086"]

## Local address to bind when connecting to the server
## If empty or not set, the local address is automatically chosen.
# local_address = ""

## Token for authentication.
token = ""

## Organization is the name of the organization you wish to write to.
organization = ""

## Destination bucket to write into.
bucket = ""
```

最终在 Grafana 中添加数据源,配置仪表盘.

5. 证书签发

5.1. acme.sh

通过 Cloudflare 提供的 API 创建 TXT 记录以签发证书:

```shell
./acme.sh --issue --dns dns_cf -d thudep.com -d '*.thudep.com'
```

完成签发后,安装到指定位置:

```shell
./acme.sh --install-cert -d 'thudep.com' --fullchain-file /srv/cert/all.thudep.com.pem --key-file /srv/cert/all.thudep.com.key
```

这里拿到的证书是同样适用于二级域名的:

```
X509v3 Subject Alternative Name:
  DNS:thudep.com, DNS:*.thudep.com
```

可以通过查看 PEM 文件验证之:

```shell
openssl x509 -in <cert> -text -noout
```

此外,注意检查 `acme.sh` 是否创建了 crontab.

6. Docker

6.1. 托管镜像

托管镜像站时注意,如果开启了 htpasswd 鉴权,那么试图通过该镜像站拉取镜像将不再成功,因为

`docker pull` 并不会使用 `docker login https://docker.thudep.com` 时保存的凭据.相反,需要指定源:

```shell
docker pull docker.thudep.com/library/nginx:alpine
```

而如果没有任何鉴权,那么在 `daemon.json` 里面配置的镜像站就可以在默认拉取 DockerHub 时生效了.