1 Introduction

The aim of this document is to provide an overview of the structure of the *CI/CD* pipeline that is currently deployed on my machine.

2 Docker Components

The pipeline consists of the following components, each one of which is running in a designated docker container:

1. Gitea
2. PostgreSQL *(Part of Gitea)*
3. Jenkins
4. Reposilite

All of these components are instantiated form a single docker compose file. Every component is connected to the same network ('`cicd`').

2.1 Jenkins

Although the setup for the Jenkins container is not particularly sophisticated, is is the most complex one of the bunch. Since the container needs to access the Docker engine of the host machine, it requires both the Docker socket, as well as a docker client to be avilable. The former is solved by mounting the Unix socket as usual:

```
volumes:
    ...
    - /var/run/docker.sock:/var/run/docker.sock
```

To approach the latter, I opted to build a custom Jenkins image that derives from the original `jenkins/jenkins`, whilst including the docker client from the `docker:dind` image. The aforementioned concept maps into the following Dockerfile:

```
FROM jenkins/jenkins
USER root

# Grab only the docker client from the DinD image
COPY --from=docker:dind /usr/local/bin/docker /usr/local/bin/

USER jenkins
```

Said image is then built and used as the container image for the Jenkins service, i.e.:

```
docker build . -t jenkins-with-docker
```

```
...
jenkins:
    container_name: cicd-jenkins
    networks:
        - cicd
    image: jenkins-with-docker
    privileged: true
    ...
```

Also note that the Jenkins container is created as a **privileged** container. This is to allow the container to use the mounted socket for communicating with the host device.

3 Gitea Config