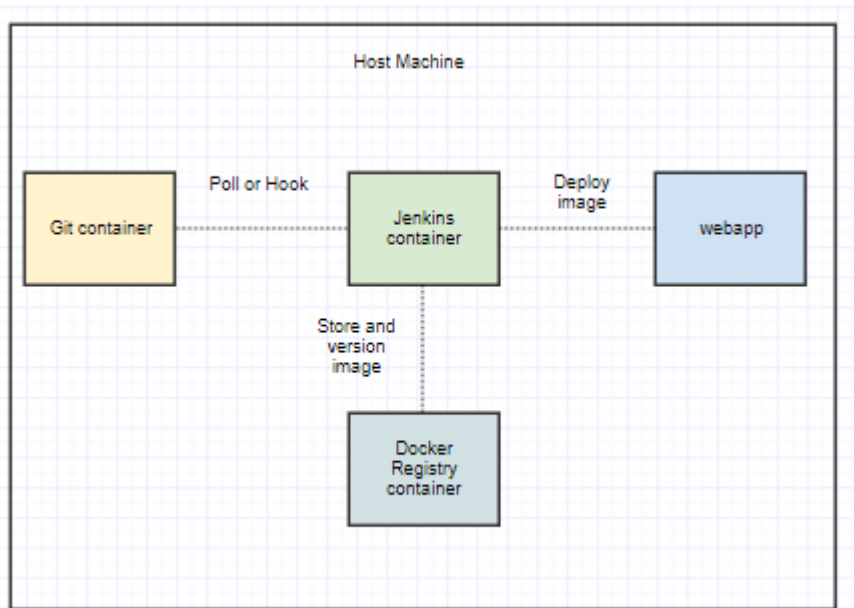


# DevOps Project

*This project is done on MacOS (Using Docker for Mac)*

## 1. Design Document

- Overview



- CI/CD Pipeline View

- This is just a simple demo of a CI/CD pipeline done on the master branch only (no branching model covered here)
- Testing strategies are also not covered in the pipeline as there are no tests in webapp project.
- Deployment is done on the host-machine
- Jenkins will expect an input on the Deploy stage. Go to console log to input the specific version of image that you want to deploy.

### Stage View

	Check out SCM	Unit Test	Build Image	Acceptance Test	Push Image	Deploy	Clean up
Average stage times: (Average full run time: ~38s)	2s	118ms	4s	271ms	1s	14s	975ms
#26 Aug 07 23:59 No Changes	2s	118ms	4s	271ms	1s	14s (paused for 12s)	975ms

## 2. Initial Set up

- Unzip the file and run:
  - Docker-compose up -d
- Run `docker ps -a` and verify all the containers are started (git-server, docker-registry, jenkins-server and webapp)
- Copy your public key into `./git-server/keys` directory
- Go into Jenkins container as root user by running
  - `Docker exec -ti --user root jenkins-server bash`

- Change ownership of docker.sock to jenkins user to enable jenkins-server container to run docker commands from host's machine by running
  - i. chown Jenkins /var/run/docker.sock
  - ii. exit after you are done

### 3. Test

- Cd into webapp folder and change any line of code
- Run automate\_git.sh (this will do git add, git commit and git push)
- There will be a git hook that triggers a build on the master Jenkins job (I did not do jenkins git polling method because it's an expensive and an inefficient operation on the server.)
- At the deploy stage, it will prompt user's input at console log. You can specify the version number of the image you want to deploy. Default value will be the latest image that was just pushed into the registry. Before that, make sure you are logged in as User: admin PW: admin.
- Verify deployment by running docker ps -a on host machine.

### 4. Areas to improve on

- Set up a feedback loop, such that it notifies developers of success/fail builds
- Set up a branching model strategy
- Write code on pipeline shared library instead of Jenkinsfile, so that codes are reusable for every project
- Set up testing strategies
- Set up controls and user permissions on who can trigger deployment
- Removing the manual step of going into Jenkins container to chown docker.sock (Let me know if you have a better way of doing it. Thanks!)