1) Build a docker image from the Dockerfile in https://github.com/STAMP-project/tecor/tree/master/testframework/jenkins

```
docker build -t jenkins-config_testing:latest .
```

2) Run a docker container form the just built image

```
docker run --restart always --name jenkins-config_testing -d -p 8090:8080 -
e JAVA_OPTS="\"-Dhudson.model.DirectoryBrowserSupport.CSP=script-src *
'unsafe-inline' 'unsafe-eval'; img-src *; style-src * 'unsafe-inline';\"" -
e MASTER_SSH_PORT=22 -e MASTER_SLAVE_USER=jenkins -e
MASTER_SLAVE_PWD=jenkins -e DOCKER_GID=993 -v
/var/run/docker.sock:/var/run/docker.sock -v
~/config_testing/jenkins_volumes/jenkins_home:/var/jenkins_home -v
~/config_testing/jenkins_volumes/jenkins_mvn_repo:/var/jenkins_mvn/repo
jenkins-config_testing:latest
```

Notes:

The testing framework performs testing in the containers. We run these containers alongside with our `jenkins-config_testing` container. Therefore, we share docker.sock.

`DOCKER_GID` specifies the groupid of the group, which owns docker.sock. Run `ls –la /var/run/docker.sock` and `cat /etc/group` to find out id.

We also share the home directory of Jenkins. Thus, we do not lose data every time the container restarts (`~/config_testing/jenkins_volumes/jenkins_home`).

Maven repository is also persisted on host OS in `~/config_testing /jenkins_volumes/jenkins_mvn_repo`.

`JAVA_OPTS` specifies parameters for Jenkins. This allows us to execute inline JS and CSS in report which is generated by the tool

3) Go to http://<ip_address>:8090 and follow on-screen instructions to install Jenkins.
   a. Unlock Jenkins
   b. Install all suggested plugins
   c. Create First Admin User
   d. Install the htmlpublisher plugin. Go to Manage Jenkins > Manage Plugins > Available > Search htmlpublisher > Install without restart > Check Restart option.

4) Go to Jenkins > New Item
   a. Create GitHub Organization with name 'My organization'
   b. Change an owner of the GitHub organization. If your repository at https://github.com/STAMP-project/tecor than the owner is STAMP-project.
   c. Enter scan credentials. Should be someone who have access to repository. (Use Jenkins Credentials Provider: Jenkins)
   d. Specify Repository pattern. We want to scan only 'tecor' repository, therefore, 'tecor' is entered.
   e. Save
   f. Jenkins should scan the repository and start the first build
   g. Wait for the build to complete (This may take a while because of the size of the repository etc)

5) To set up automatic build one has to set up a webhook. For example
    a. Go to https://github.com/STAMP-project/tecor
    b. Settings > Webhooks > Add webhook
    c. Enter <host_name or global ip address>/github-webhook/
    d. Choose application/x-www-form-urleccoded content type
    e. Check 'Let me select individual event'
        i. Pull request
        ii. Push
        iii. Repository

6) We are ready to go. Jenkins should perform testing on every new change to the config-testing repository.