# Solution M5: Jenkins Advanced

This is one possible and fully automated solution of the tasks included in the homework

All steps that follow assume that we decided to base our solution on **Debian 11** box

Please note, that this solution is intentionally far from being an optimal one

Here, the emphasis is put on readability and not on optimization or speed

You must adjust some values like IP addresses, image (or box) names, repository names, credentials, etc. to match your use case

Default pairs user and passwords in use are – for **Jenkins** (**admin**/**admin**) and for **Gitea** (**vagrant**/**vagrant**)

## Solution

A pack of files is provided. It contains all the necessary files. They are as follows:

.

├── provision-scripts -> contains all provision/configuration scripts

│   ├── add\_hosts.sh -> used to set the /etc/hosts file of the machines

│   ├── install\_docker.sh -> install and configure Docker

│   ├── install\_jenkins.sh -> install Jenkins

│   ├── setup\_gitea.sh -> deploys Gitea (server + db) with Docker Compose

│   └── setup\_jenkins.sh -> setups Jenkins - credentials, slave, plugins, job, etc.

├── shared-files -> all files that are mounted and visible to the machines

│   ├── docker -> files for the Docker machine

│   │   ├── docker-compose-build.yaml -> used to build the test deployment of the app

│   │   ├── docker-compose-deploy.yaml -> used to deploy the production deployment of the app

│   │   └── docker-compose.yml -> used to deploy Gitea

│   └── jenkins -> files for the Jenkins machine

│   ├── 0-url.groovy -> configure the URL

│   ├── 1-disable-setup.groovy -> configure the Initial Setup

│   ├── 2-admin-user.groovy -> configure Admin user (admin/admin)

│   ├── add-jenkins-credentials.sh -> used to add credentials (referred by setup\_jenkins.sh)

│   ├── add-jenkins-job.sh -> used to add the job (referred by setup\_jenkins.sh)

│   ├── add-jenkins-slave.sh -> used to add Docker machine as slave (referred by setup\_jenkins.sh)

│   ├── job.xml -> the actual job/pipeline

│   └── plugins.txt -> list of plugins to be installed (referred by setup\_jenkins.sh)

└── Vagrantfile -> Vagrantfile that assembles it all. Starts Docker and then Jenkins

You should extract the contents of the archive and navigate to the extracted folder

You must adjust at least the following files:

* **provision-scripts/setup\_jenkins.sh**
* **shared-files/docker/docker-compose-build.yaml**
* **shared-files/docker/docker-compose-deploy.yaml**
* **shared-files/jenkins/2-admin-user.groovy**
* **shared-files/jenkins/job.xml**

Check and change at least **<docker-hub-account-name>** and **<docker-hub-token>** to match yours

You should check the rest as well

Then, after the necessary adjustments, you must execute just this:

**vagrant up**

And after around 10 minutes the solution will be up

The infrastructure services are available here:

* **Jenkins** – <http://192.168.99.101:8080>
* **Gitea** – <http://192.168.99.102:3000>

No build will be run until you commit a change in the repository or trigger the build manually

After a successful build and deployment, the application will be available at <http://192.168.99.102>