1. **Docker Lab 02**
2. **Pull jenkins/jenkins:latest docker image.**

docker pull jenkins/jenkins:latest

1. **Using "docker run command" run a docker container:**
   1. name: “jenkins-container”
   2. image: jenkins/jenkins:latest
   3. in daemon mode
   4. port forwarding between port 8080 on container to port 8099 on host, then open [http://localhost:8099](http://localhost:8099/)

- docker run -d -p 8099:8080 --name=jenkins-container jenkins/jenkins:latest

1. **Access "jenkins-container" container interactively with bash and check the uid of jenkins app runner.**

- docker exec -it jenkins-container bash

- id -u

1. **Check docker-compose version.**

- docker-compose --version

1. **Create an empty project directory "wordpress-compose" which contains a docker-compose.yml file which is complete in itself for a good starter wordpress project. (define and run two containers "wordpress" and "mysql db")  
   Hint: for wordpress container use wordpress:latest image and mysql:5.7 image for mysql container.**

- mkdir ~/wordpress-compose && cd ~/wordpress-compose

- touch docker-compose.yml

- docker-compose up

\* docker-compose.yml content:

version: '3'

services:

mysql-instance:

image: mysql:5.7

ports:

- "8030:3306"

environment:

MYSQL\_DATABASE: wordpressdb

MYSQL\_USER: wordpressuser

MYSQL\_PASSWORD: normal-user-pass

MYSQL\_ROOT\_PASSWORD: super-secret-pass

volumes:

- "./db\_data:/var/lib/mysql"

wordpress-instance:

image: wordpress:latest

ports:

- "8080:80"

depends\_on:

- mysql-instance

environment:

WORDPRESS\_DB\_HOST: mysql-instance

WORDPRESS\_DB\_USER: wordpressuser

WORDPRESS\_DB\_PASSWORD: normal-user-pass

WORDPRESS\_DB\_NAME: wordpressdb

WORDPRESS\_TABLE\_PREFIX: wp\_

volumes:

- "./wordpress:/var/www/html"