

# **Final project:**

## **Project description and goal**

The project will use latest technologies for creating a CI pipeline.

Source control management will be used for team collaboration, backup and versions control.

Code will be deployed in servers using containerization methods.

The product will demonstrate a web server getting requests and keeping track in a database.

A functional and GUI test will be performed on each successful deployment.

## **General:**

Development: Code will be developed in Python using PyCharm IDE.

Third-Party usage: Jenkins, GIT, Docker, Redis, Flask, Selenium WebDriver.

Distribution type: Private.

Networking type: Local.

## **Project guidelines:**

1. Jenkins will run and schedule pipeline.
2. Git will be used as source control management tool.
3. Docker will be used as a containerization tool.
4. The pipeline will be deployed locally, but with the ability to run on cloud.

## Manual steps

1. Clone the following repository into a new Github repository  
<https://github.com/Dgotlieb/DockerRedisPython.git>
2. Write a selenium test which will navigate to localhost:5000 and print the site text (*Hello World! I have been seen...*).
3. Upload Selenium test script to your Github repository.
4. Write Jenkins pipeline as described below.

## Jenkins pipeline configurations:

1. Jenkins pipeline will use Poll SCM mechanism every 30 minutes.
2. Pipeline will be marked as a **Github Project** with the Github project name (URL excluding .git)

## Jenkins pipeline steps:

1. Code will be pulled from your new Github repository holding the cloned project (DockerRedisPython) and your Selenium test script using **git** command.
2. Run docker-compose up to run the web app
3. Run Selenium test script

## Test machine prerequisites:

1. **Jenkins** installed on machine
2. Jenkins **Github** and **Pipeline** plugins are installed
3. **Docker toolbox** is installed and the “**default**” machine is running.

## What to send?

1. Jenkins pipeline **config.xml** content which can be found at the following path:  
**Windows:** C:\Users\**<your user name>**\.jenkins\jobs\**<job name>**  
**OS X:** /Users/**<your user name>**/.jenkins\jobs\**<job name>**
2. Github project URL including:
  - Cloned **DockerRedisPython** project.
  - Selenium test script

## Extras:

1. Run pipeline on a slave machine.
2. Run the web service on a mounted volume
3. Remove the word “World” from printing
4. Send an email when failure occurred in Jenkins pipeline