Advanced Software Engineering (Lab)

Stefano Forti

`name.surname@di.unipi.it`

Department of Computer Science @ University of Pisa

What will I do?

- Orchestrate building and testing a python application with PyInstaller.
- Get familiar with CI/CD with Jenkins.
- Write a `Jenkinsfile` for tic-tac-toe.

Create a bridge network

- It will be shared between Jenkins containers to communicate:

```
docker network create jenkins
```

Download and run docker:dind

- It will permit Jenkins to run docker containers (using **jenkins** network)

```
docker run \
  --name jenkins-docker \
  --rm \
  --detach \
  --privileged \
  --network jenkins \
  --network-alias docker \
  --env DOCKER_TLS_CERTDIR=/certs \
  --volume jenkins-docker-certs:/certs/client \
  --volume jenkins-data:/var/jenkins_home \
  docker:dind \
  --storage-driver overlay2
```

Create Dockerfile

- Customise the official Jenkins Docker image:

```
FROM jenkins/jenkins:2.361.4-jdk11
USER root
RUN apt-get update && apt-get install -y lsb-release
RUN curl -fsSLo /usr/share/keyrings/docker-archive-keyring.asc \
  https://download.docker.com/linux/debian/gpg
RUN echo "deb [arch=$(dpkg --print-architecture) \
  signed-by=/usr/share/keyrings/docker-archive-keyring.asc] \
  https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list
RUN apt-get update && apt-get install -y docker-ce-cli
USER jenkins
RUN jenkins-plugin-cli --plugins "blueocean:1.25.8 docker-workflow:521.v1a_a_dd2073b_2e"
```

Build it

- Build the image and tag it:

```
docker build -t myjenkins-blueocean:2.361.4-1 .
```

Run it!

```
docker run \
  --name jenkins-blueocean \
  --detach \
  --network jenkins \
  --env DOCKER_HOST=tcp://docker:2376 \
  --env DOCKER_CERT_PATH=/certs/client \
  --env DOCKER_TLS_VERIFY=1 \
  --publish 8080:8080 \
  --publish 50000:50000 \
  --volume jenkins-data:/var/jenkins_home \
  --volume jenkins-docker-certs:/certs/client:ro \
  --volume "$HOME":/home \
  --restart=on-failure \
  --env JAVA_OPTS="-Dhudson.plugins.git.GitSCM.ALLOW_LOCAL_CHECKOUT=true" \
  myjenkins-blueocean:2.361.4-1
```

For Windows (**without WSL**) you must use this command instead:

```
docker run --name jenkins-blueocean --detach ^
  --network jenkins --env DOCKER_HOST=tcp://docker:2376 ^
  --env DOCKER_CERT_PATH=/certs/client --env DOCKER_TLS_VERIFY=1 ^
  --volume jenkins-data:/var/jenkins_home ^
  --volume jenkins-docker-certs:/certs/client:ro ^
  --volume "%HOMEDRIVE%%HOMEPATH%":/home ^
  --restart=on-failure ^
  --env JAVA_OPTS="-Dhudson.plugins.git.GitSCM.ALLOW_LOCAL_CHECKOUT=true" ^
  --publish 8080:8080 --publish 50000:50000 myjenkins-blueocean:2.361.4-1
```

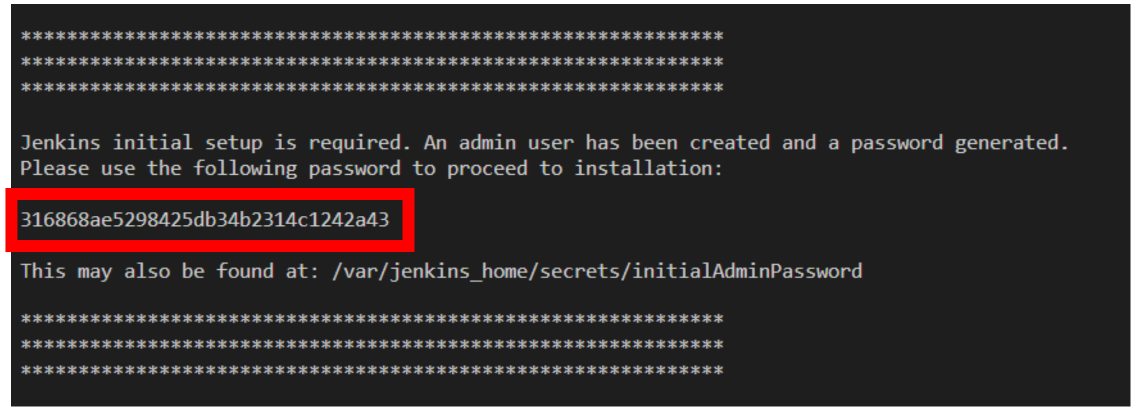
Access Jenkins

- Connect to:

```
http://localhost:8080
```

- Retrieve password through issuing the command:

```
docker logs jenkins-blueocean
```



```
*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

316868ae5298425db34b2314c1242a43

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*************************************************************
*************************************************************
*************************************************************
```

- Click “**Install suggested plugins.**”
- Create first Admin User and continue until “**Start using Jenkins**”

Create and setup the github project

- Clone the test application;

```
git clone https://github.com/jenkins-docs/simple-python-pyinstaller-app.git
```

- Note the folder where you cloned:
  - For macOS - `/Users/<your-username>/Documents/GitHub/`
  - For Linux/WSL - `/home/ase/simple-python-pyinstaller-app`
  - For Windows - `C:\Users\<your-username>\Documents\ase\simple-python-pyinstaller-app`
- Your home directory, e.g. **`username/`**… will be mapped to **`home/`**…

Create pipeline project in Jenkins

- **New Element -> Pipeline**
- **Choose the definition "Pipeline script from SCM"**
- From the **SCM** field, choose **Git**.
- In the Repository URL field, specify the directory path of your locally cloned repository above, which is from your user account/home directory on your host machine, mapped to the /home directory of the Jenkins container - i.e.
  - For macOS - /home/Documents/ase/simple-python-pyinstaller-app
  - For Linux/WSL - /home/ase/simple-python-pyinstaller-app
  - For Windows - /home/Documents/ase/simple-python-pyinstaller-app
- Click **Save**

Create Jenkinsfile

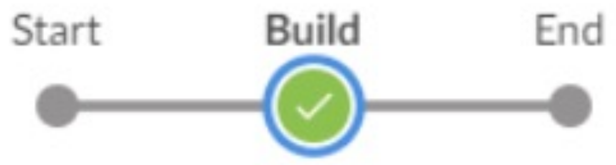
```
pipeline {
    agent none
    stages {
        stage('Build') {
            agent {
                docker {
                    image 'python:2-alpine'
                }
            }
            steps {
                sh 'python -m py_compile sources/add2vals.py sources/calc.py'
                stash(name: 'compiled-results', includes: 'sources/*.py*')
            }
        }
    }
}
```

- Then commit:

```
git add .
git commit -m "Add initial Jenkinsfile"
```

Run the job in Jenkins

- Open Blue Ocean with the button **Open Blue Ocean**
- In the **This job has not been run** message box, click **Run**
    - Or "Build Now" inside the pipeline dashboard (outside Blue Ocean)



Add Test stage

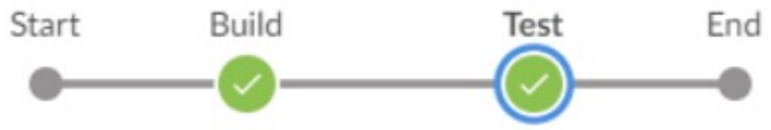
```
        stage('Test') {
            agent {
                docker {
                    image 'qnib/pytest'
                }
            }
            steps {
                sh 'py.test --junit-xml test-reports/results.xml sources/test_calc.py'
            }
            post {
                always {
                    junit 'test-reports/results.xml'
                }
            }
        }
```

- Then commit:

```
git add .
git commit –m "Add ‘Test’ stage"
```

Re-run the job in Jenkins

- Open Blue Ocean with the button **Open Blue Ocean**
- In the **This job has not been run** message box, click **Run**
  - Or "Build Now" inside the pipeline dashboard (outside Blue Ocean)



Add final Deliver stage

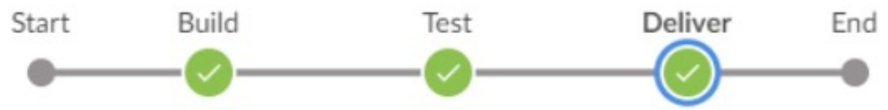
```
        stage('Deliver') {
            agent any
            environment {
                VOLUME = '$(pwd)/sources:/src'
                IMAGE = 'cdrx/pyinstaller-linux:python2'
            }
            steps {
                dir(path: env.BUILD_ID) {
                    unstash(name: 'compiled-results')
                    sh "docker run --rm -v ${VOLUME} ${IMAGE} 'pyinstaller -F add2vals.py'"
                }
            }
            post {
                success {
                    archiveArtifacts "${env.BUILD_ID}/sources/dist/add2vals"
                    sh "docker run --rm -v ${VOLUME} ${IMAGE} 'rm -rf build dist'"
                }
            }
        }
```

- Then commit:

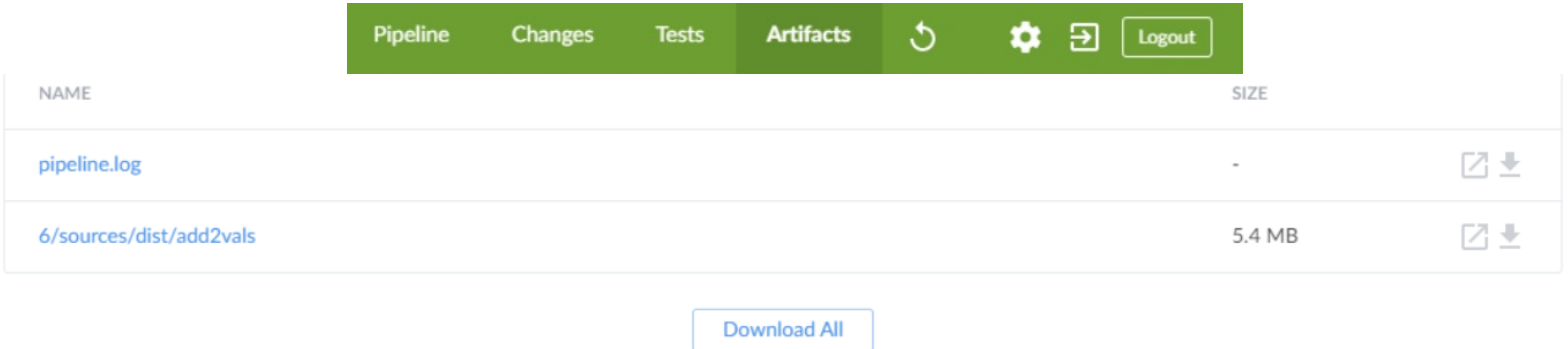
```
git add .
git commit –m "Add ‘Deliver’ stage"
```

Re-run the job in Jenkins

- Open Blue Ocean with the button **Open Blue Ocean**
- In the **This job has not been run** message box, click **Run**
  - Or "Build Now" inside the pipeline dashboard (outside Blue Ocean)



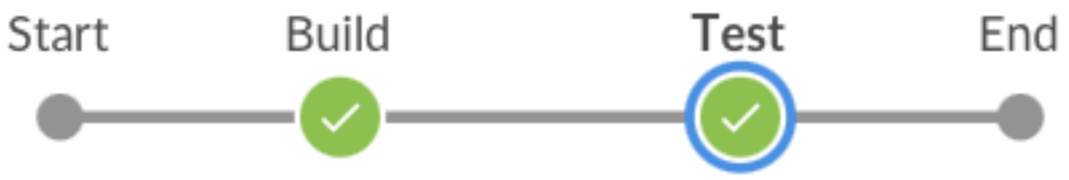
- Now you can download the executable generated inside Artifacts



NAME	SIZE
pipeline.log	-
6/sources/dist/add2vals	5.4 MB

Exercise

- Clone the repo at https://github.com/teto1992/tic-tac-toe
- Write a Jenkins pipeline to build `tic-tac-toe` into two stages:



Hint! – Choose Docker images (for building and testing) featuring Python3.