**Dockerisation of an application**

Goals:

• Dockerise the application from Sprint 1 of the project manually.

• Run your application in a container on a remote VM

Tasks:

• Install Docker on a fresh VM

*Docker already installed so didn’t need to do this*

• Create a Dockerfile for the application from Sprint 1 of the project.

*git clone* [*https://github.com/TMughalGitHub/LBG-Python.git*](https://github.com/TMughalGitHub/LBG-Python.git)

*sudo usermod -aG docker $USER*

*newgrp docker*

*sudo usermod -aG docker jenkins*

*sudo systemctl restart jenkins*

*Using the LBG-Python project in GitHub. Already an empty Dockerfile in there … add the below into it:-*

FROM python:3.10  
MAINTAINER Tahira Mughal

#install wget for maven

RUN apt-get install -y wget

WORKDIR /LBG-Python

COPY . .

RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 8080

CMD [ "python", "./lbg.py" ]

• Build a Docker image of your application.

*docker build -t tm110/tmughal:1 .*

• Run your container from the image using Docker to verify that your Dockerfile is correct.

*docker run -d -P tm110/tmughal:1*

to run it:-

*docker run -it tm110/tmughal:1 /bin/bash*

Had to make another change to lbg.py file to replace localhost with 0.0.0.0

HOST = '0.0.0.0'

Use the below to check that the app will launch;-

curl <http://localhost:8080/index.html>

Issue seen in my VM when running docker ps

WARNING: Error loading config file: /home/mughalt110/.docker/config.json: open /home/mughalt110/.docker/config.json: permission denied

$ sudo chown "$USER":"$USER" /home/"$USER"/.docker -R

$ sudo chmod g+rwx "/home/$USER/.docker" -R

**2. Creating build scripts in Bash**

Goals:

• Demonstrate how to create a Bash script which automates the build process and dockerisation of your application.

• Understand how to make your application build repeatable.

Tasks:

• Create a bash script which perform the following tasks: o Build your application.

o Build a Docker image of your application.

• Modify your application.

• Generate new Docker image of the application using your script.

• Run the docker image as a container.

Shell script

*rm -rf ~/LBG-Python*

*git clone https://github.com/TMughalGitHub/LBG-Python.git*

*cd LBG-Python*

*docker build -t tm110/tmughal:1 .*

*docker run -d -p 8081:8081 tm110/tmughal:1*

to run it . ./Dockerbuild.sh

Use docker ps to see that it’s successful.

**3. Configuration of an application**

Goals:

• Know how to configure your application according to the 12 Factor configuration approach.

• Take advantage of immutable deployments.

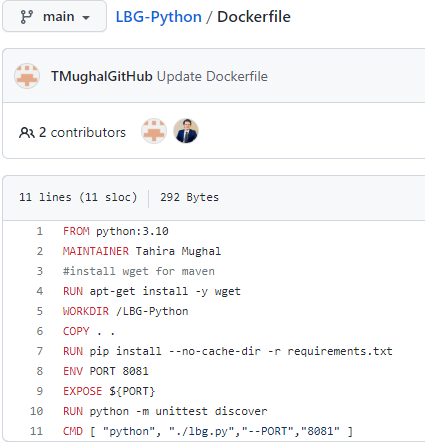
• Understand how environment variables can be used to configure your application in an immutable deployment manner.

Tasks:

• Notice that the port that the application listens on is defined by an environment variable PORT, and if no environment variable exists then it defaults to port 8080.

• Run your application as a Docker container but use a **Docker environment variable –** research this to specify the port (for example run the application listening to the port 8081).

Amend Dockerfile as below

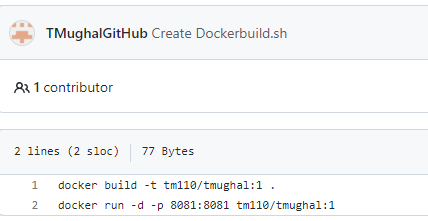


Rerun Dockerbuild.sh and then run below to check app working ok.

curl http://localhost:8081/index.html

use http://[34.22.147.216:8081/index.html](http://34.22.147.216:8081/index.html) in browser

Add dockerbuild.sh to GitHub



**Setting up Jenkins CI pipeline**

Goals:

• Demonstrate how to install Jenkins server.

• Understand how to set up a CI pipeline using Jenkins.

• Know how to configure your CI pipeline to verify that your project can be successfully built on every Git commit.

Tasks:

• Use and configure a Jenkins server.

• Initialize a basic Jenkins configuration.

• Create a GitHub Webhook polling triggering changes from your GitHub server.

• Configure your job to run your application's tests on every Git commit.

• Introduce a documentation change to your project and push that change to the Git repository.

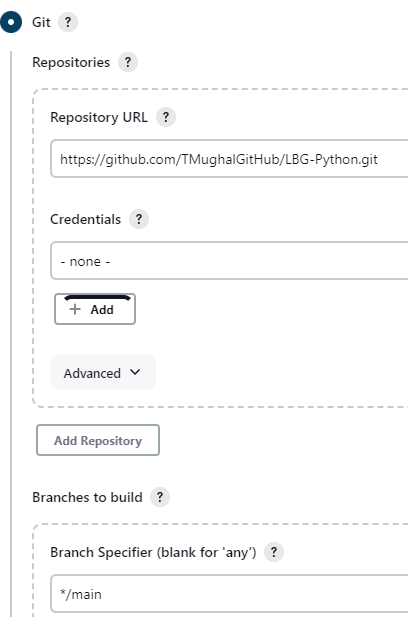
• Verify that Jenkins successfully pulled the changes and executed your job as expected.

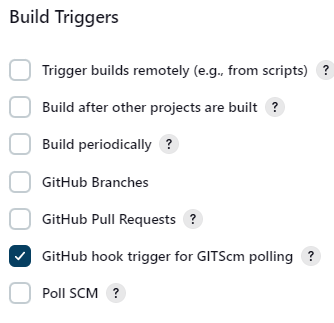
Login to Jenkins using

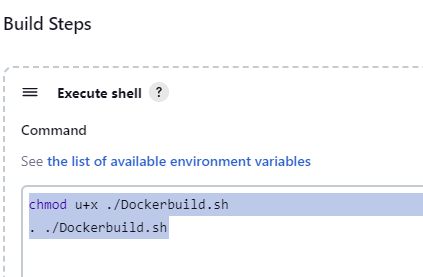
Http://<ip>:8080

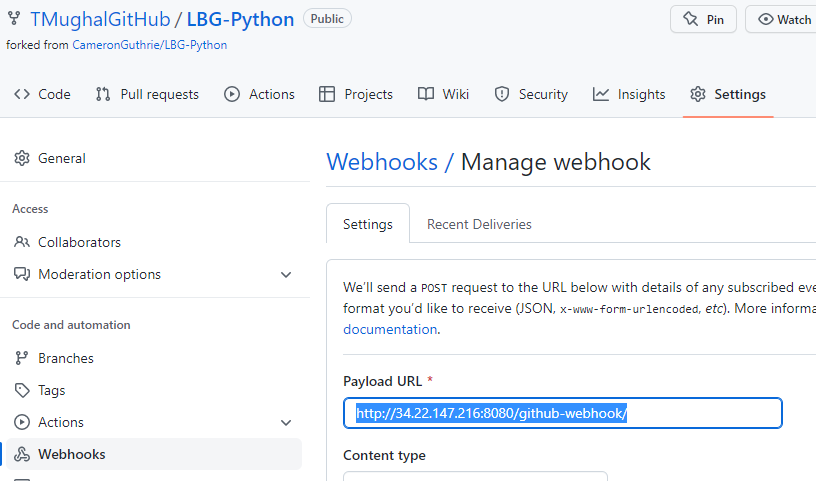
User/pass admin/Pa$$w0rd

Create New Item (Freestyle project)









Build the job. Should run ok.