**Devops project – summary subjects**

**Jenkins, Docker (with compose) and GitHub (with desktop tool)**

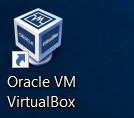
**By Shlomi Kaduri**

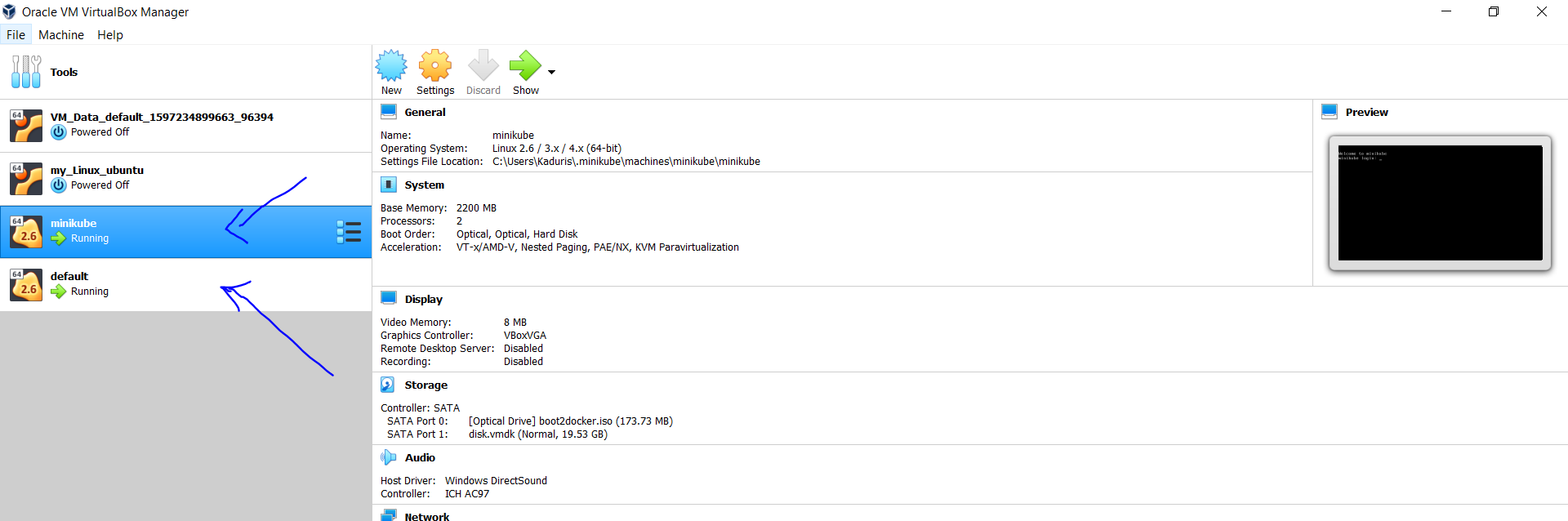
[**https://github.com/shlomikaduri/ProjectGames.git**](https://github.com/shlomikaduri/ProjectGames.git)

This project includes flask web-server that know to read txt file and present it to web-server.

What we did in this project is to take current project that called MainScores and run it via docker as container.

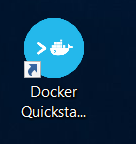
We bring up 2 virtual machines for this project

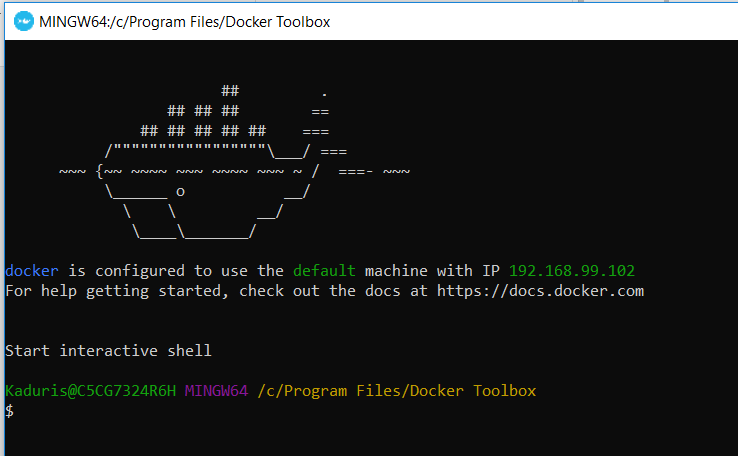




That created automatically after docker and minikube setup installation

The ip address of our docker was 192.168.99.102 – see below



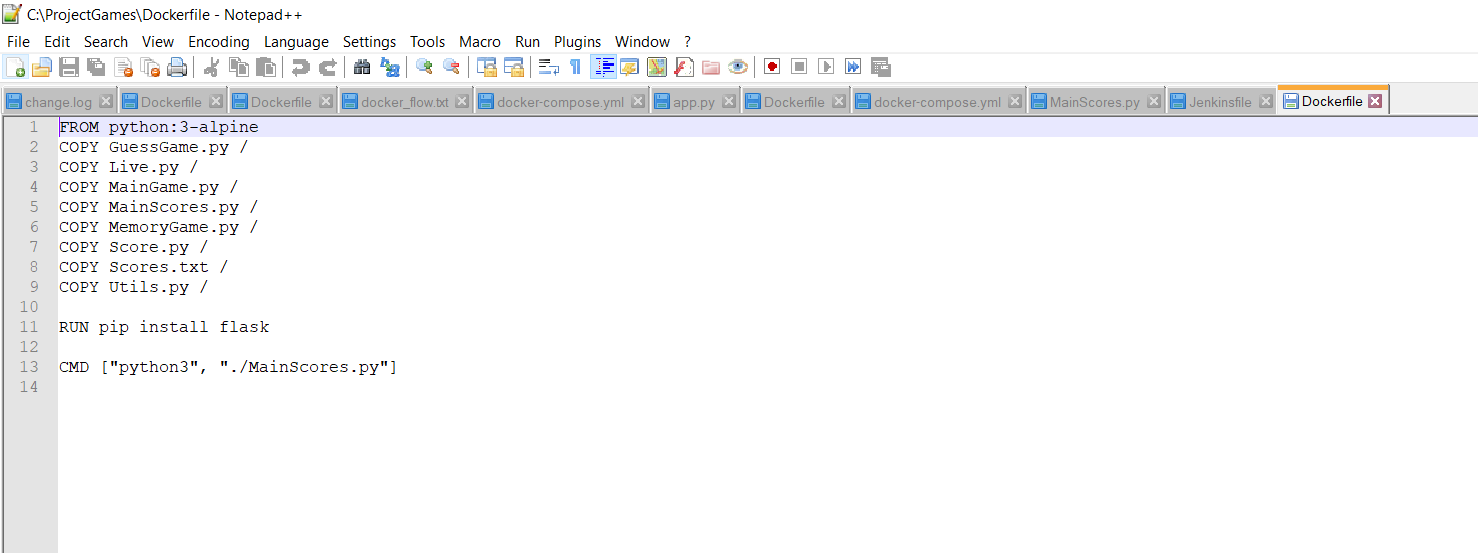


From this point – we continue to work via CMD.

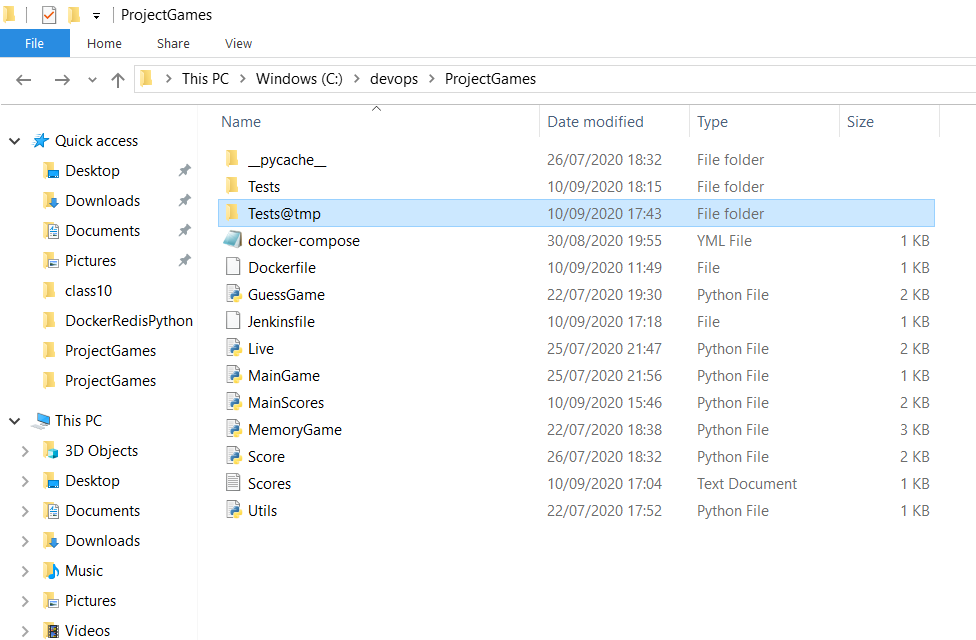
Before we continue to Jenkins and GitHub we will explain regarding the process of building project to docker 🡪 image 🡪 container

Main Steps:

1. Develop the tests , python files, etc.
2. Create dockerfile - should be located in project folder



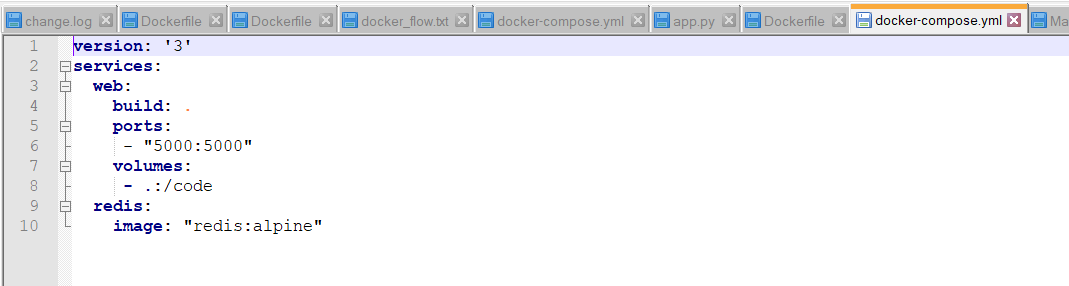
1. Prepare docker-compose (should be located in project folder).



1. Run it via script or Jenkins pipeline, you need to run

*C:\ProjectGames>****docker-compose up --build -d*** that release he command line.

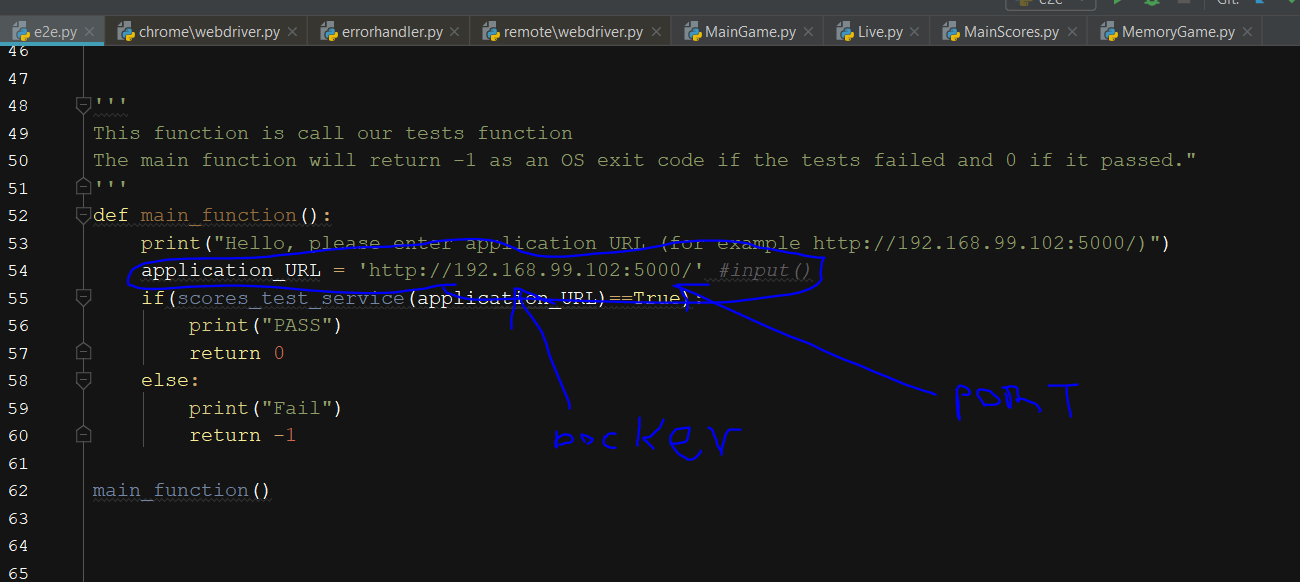
*Docker-compose is yml file – 2 services (web and redis)*

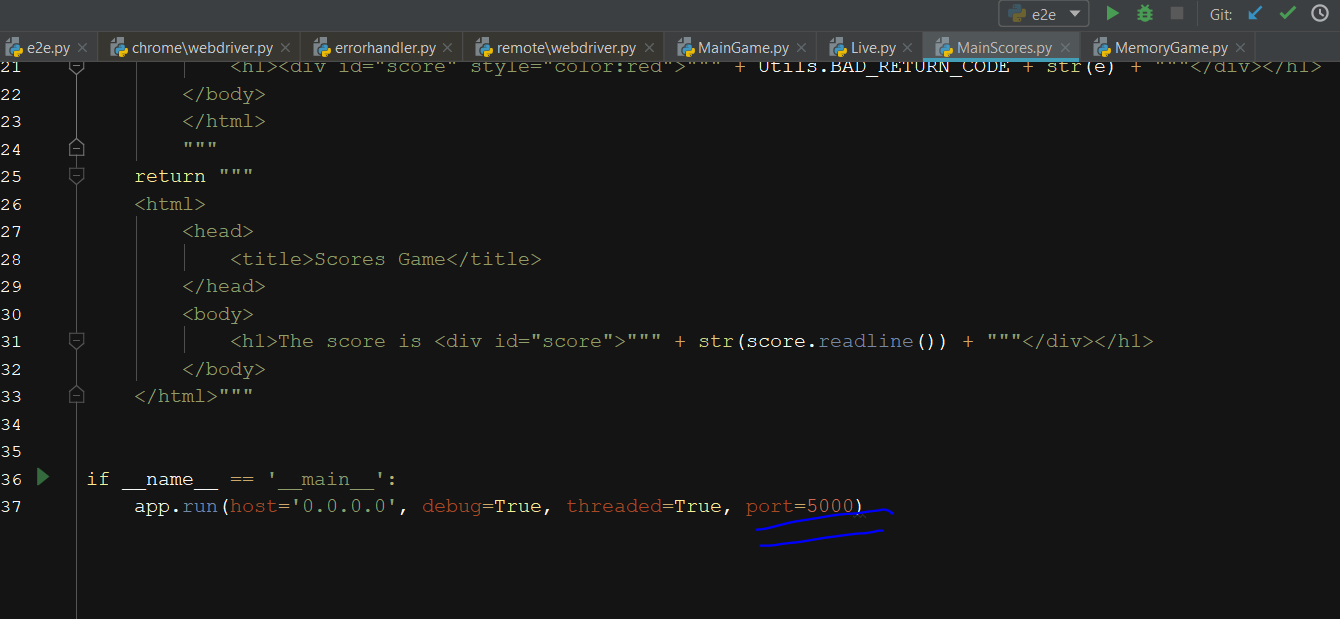


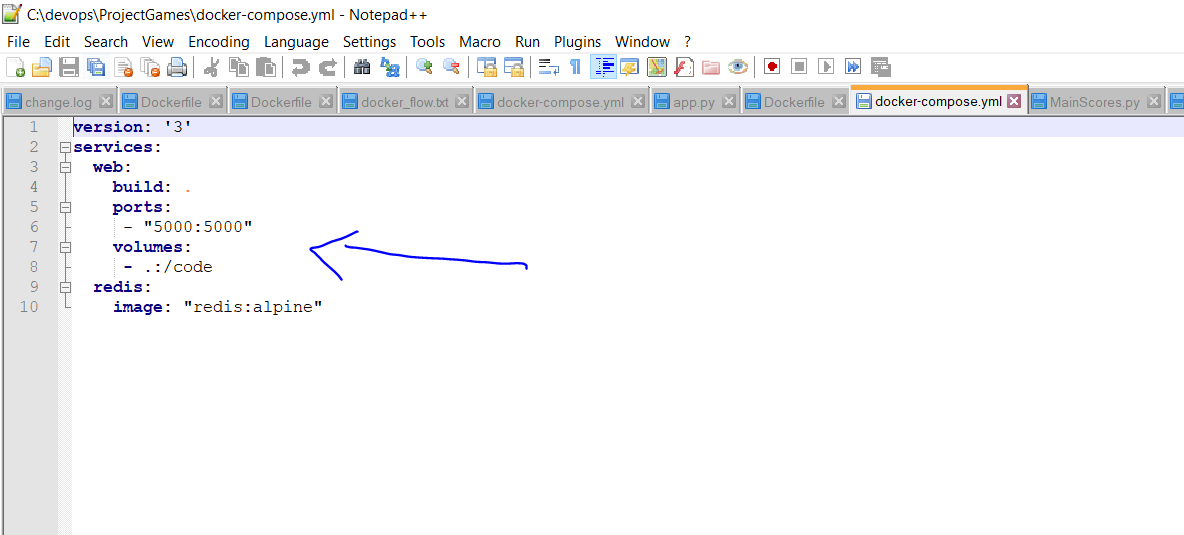
1. From this point the container is run as isolated environment, and we can see it via (docker/container) ps.
2. When we finish to use the container, we need to stop and remove the containers by

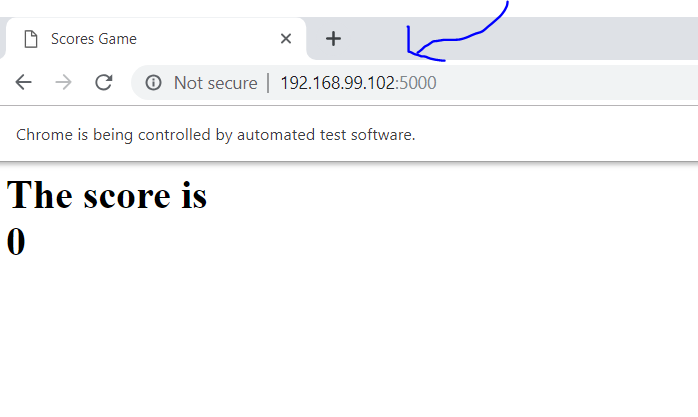
*C:\ProjectGames>docker-compose down.*

1. **The way to access to the server inside docker is**









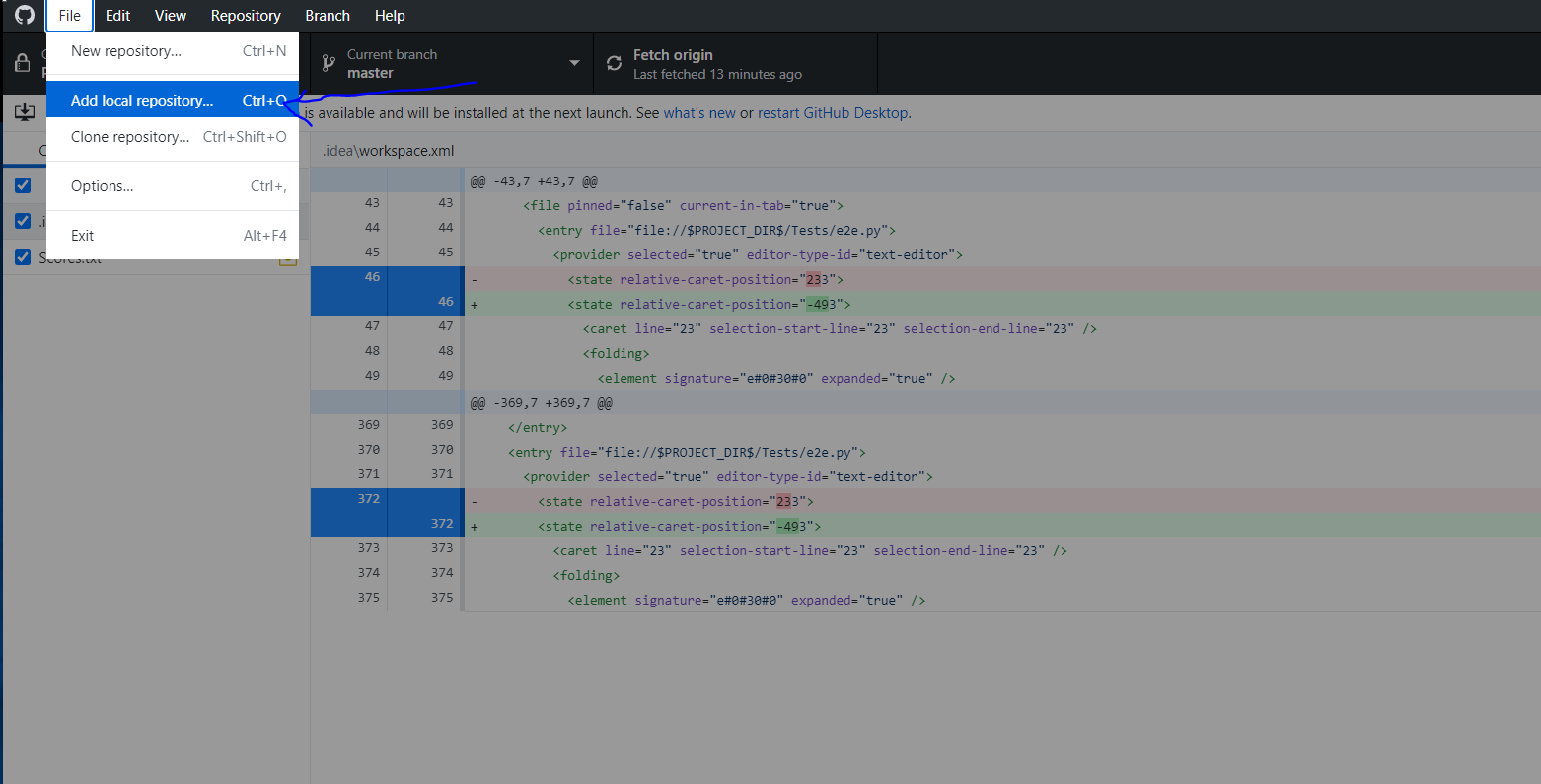
**So acutely this web, run from our temporary docker**

**And if we will change the score.txt local, it will not be reflected to the docker.**

1. **Jenkins and Githut**

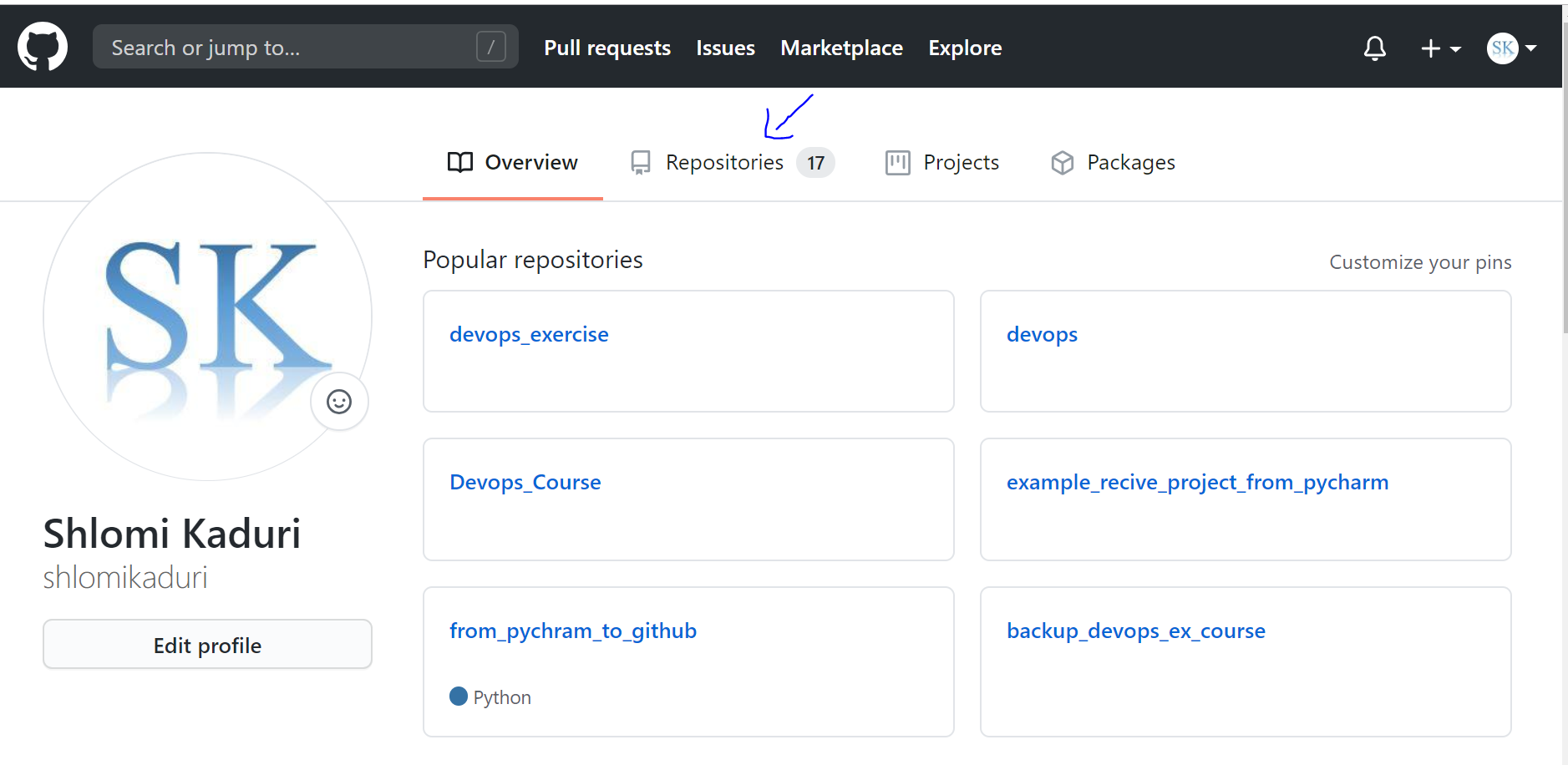
In order to mange and create project in GitHub – we already connected our git desktop with GitHub

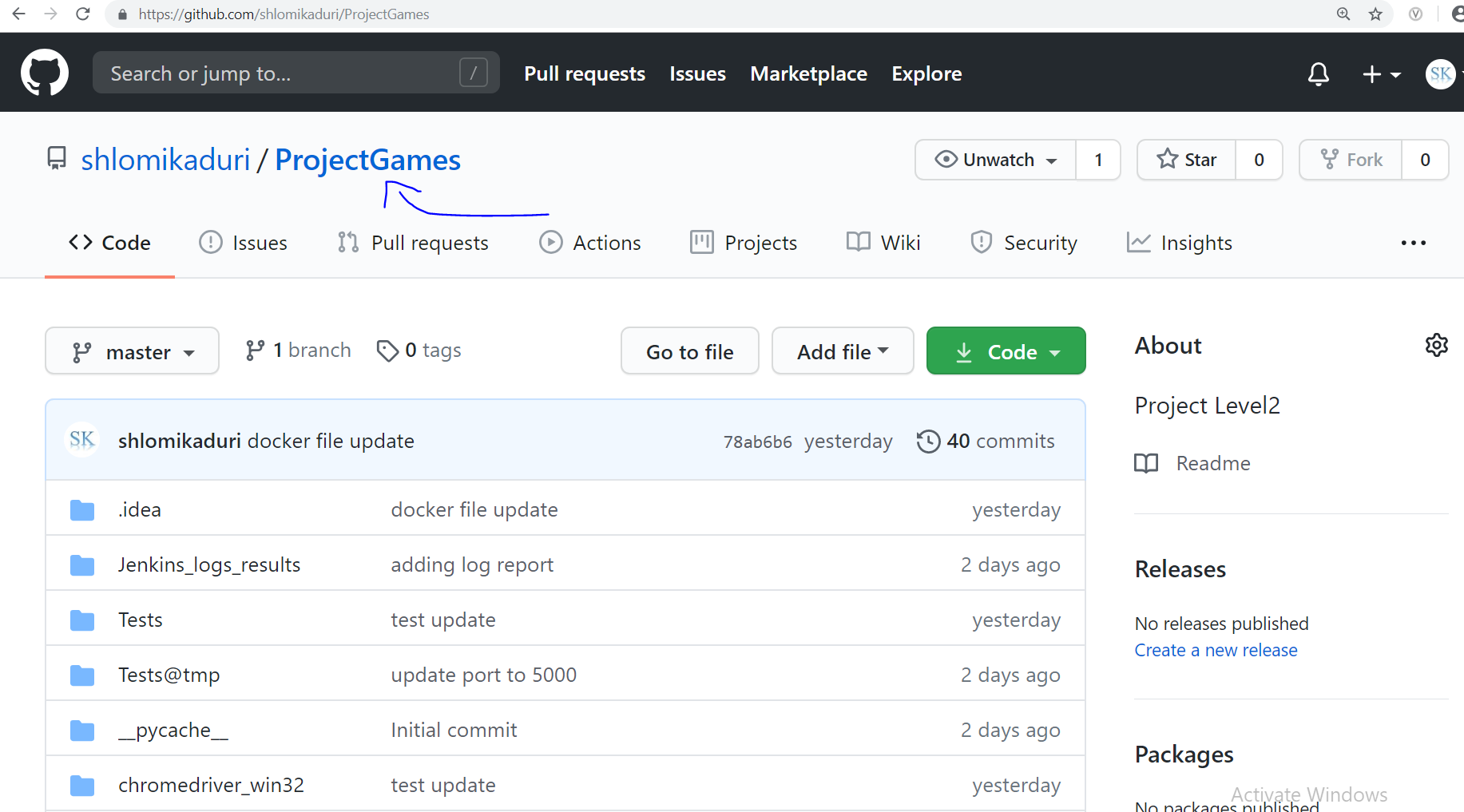
Steps:



Create new local repository, then init commit to all files and do fetch.

Then, go to your GitHub account and fount id under repositories tub





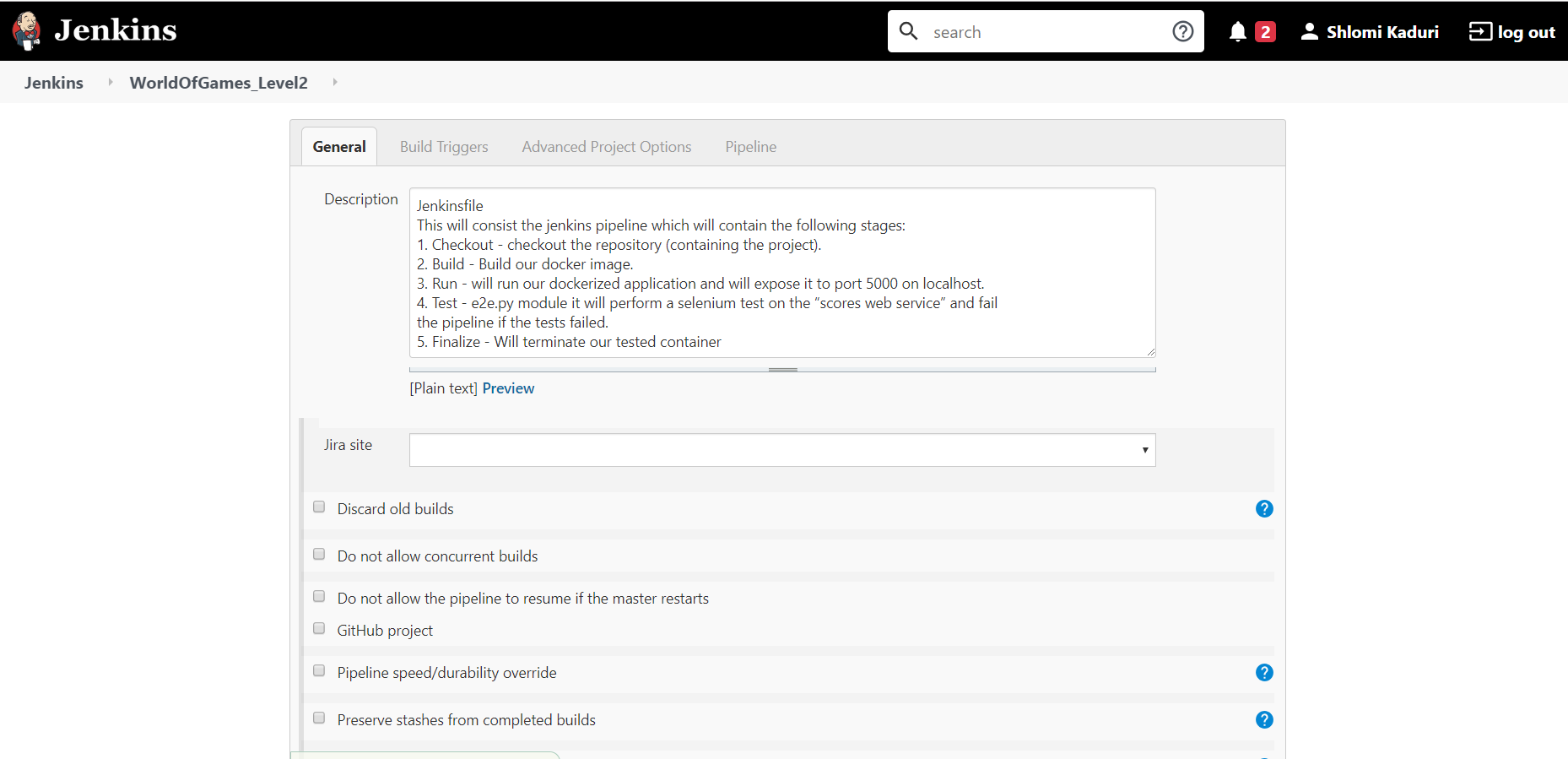
From this point, this is our source, and we will use this data to do any next devops actions like pipeline with Jenkins.

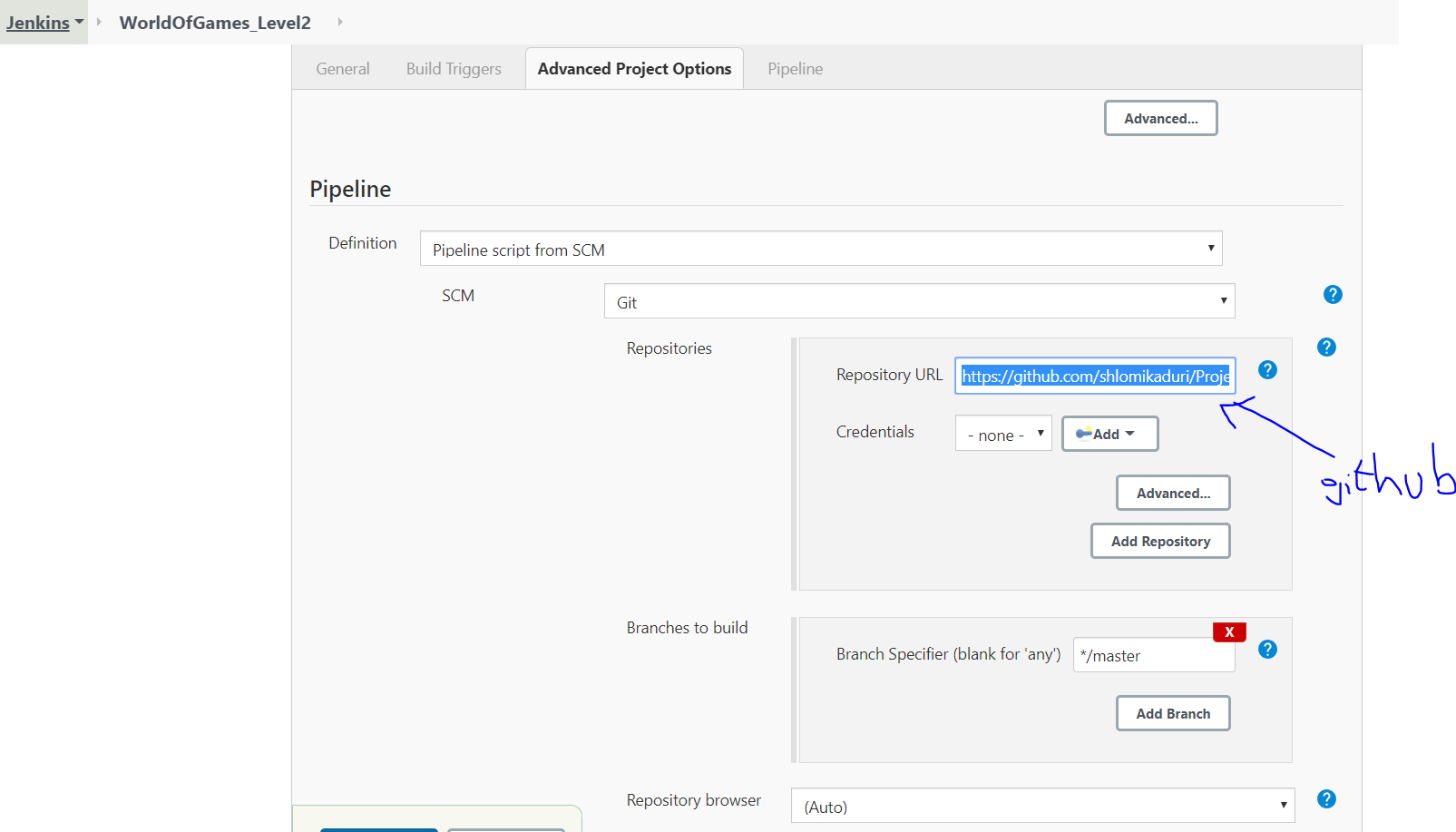
Jenkins,

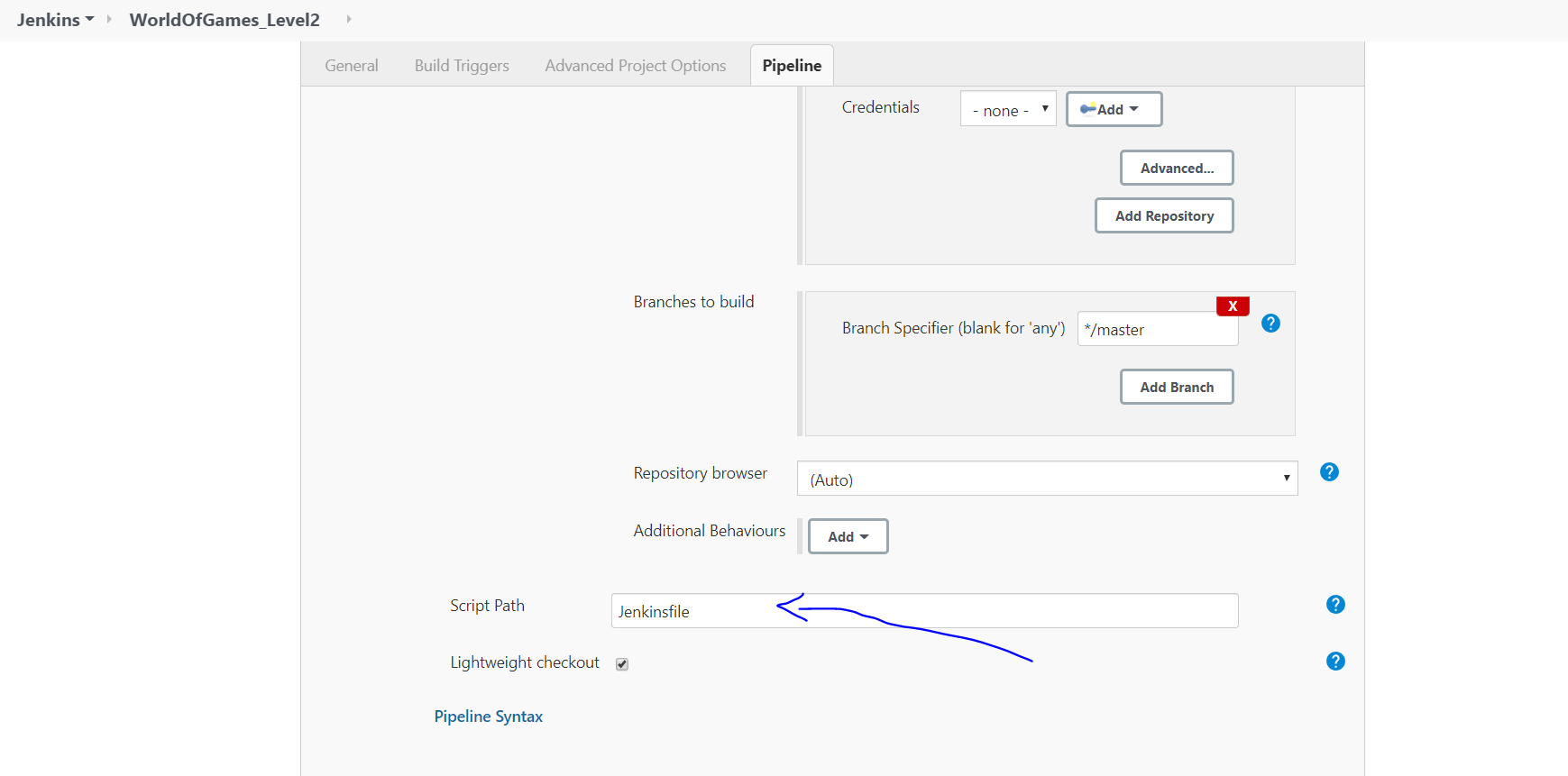
First, don’t forget to start Jenkins engine by (9090 is free port in my pc)

*java -jar jenkins.war --httpPort=9090*

Enter Jenkins and Create pipeline config

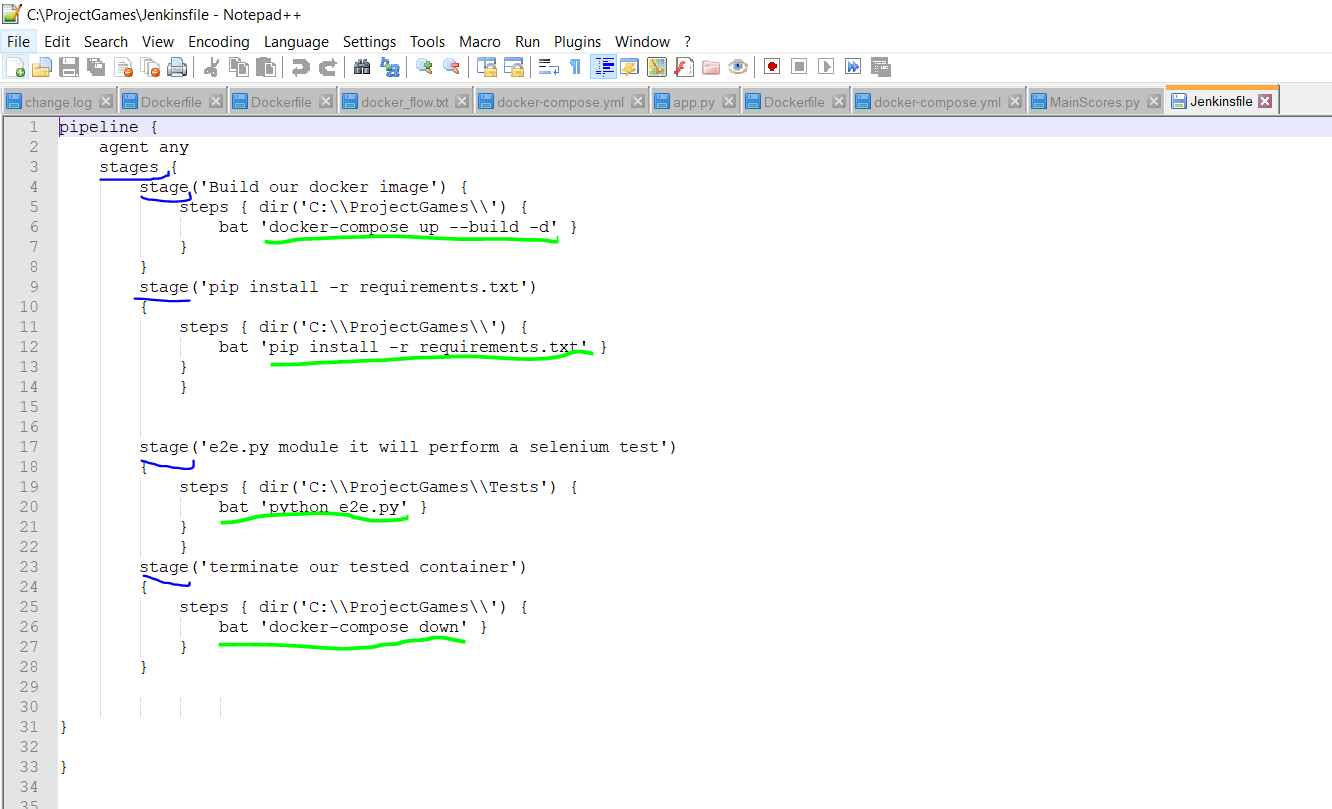


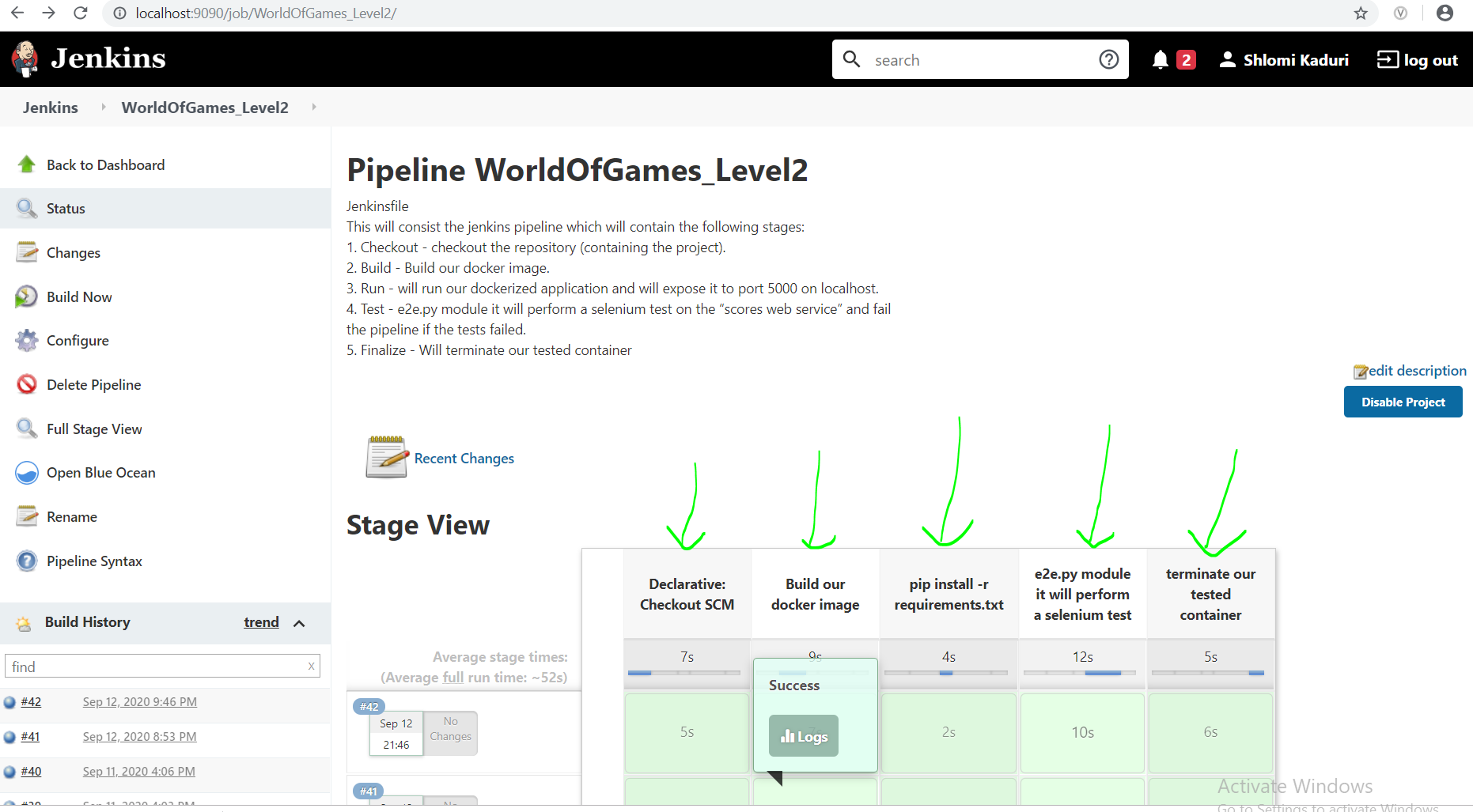




This is very impotent to use jenkinsfile for our pipeline steps

Jenkinsfile is written in Grovy and this is our jenkinsfile





In order to run – we do – build now

And this is our output console

**SuccessConsole Output**

Started by user [Shlomi Kaduri](http://localhost:9090/user/admin)

Obtained Jenkinsfile from git <https://github.com/shlomikaduri/ProjectGames.git>

Running in Durability level: MAX\_SURVIVABILITY

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [Jenkins](http://localhost:9090/computer/(master)/) in C:\Users\Kaduris\.jenkins\workspace\WorldOfGames\_Level2

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

No credentials specified

> git.exe rev-parse --is-inside-work-tree # timeout=10

Fetching changes from the remote Git repository

> git.exe config remote.origin.url <https://github.com/shlomikaduri/ProjectGames.git> # timeout=10

Fetching upstream changes from <https://github.com/shlomikaduri/ProjectGames.git>

> git.exe --version # timeout=10

> git.exe fetch --tags --force --progress -- <https://github.com/shlomikaduri/ProjectGames.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10

> git.exe rev-parse "refs/remotes/origin/origin/master^{commit}" # timeout=10

Checking out Revision 78ab6b684455489583214e82de48530d846fa3f2 (refs/remotes/origin/master)

> git.exe config core.sparsecheckout # timeout=10

> git.exe checkout -f 78ab6b684455489583214e82de48530d846fa3f2 # timeout=10

Commit message: "docker file update"

> git.exe rev-list --no-walk 78ab6b684455489583214e82de48530d846fa3f2 # timeout=10

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Build our docker image)

[Pipeline] dir

Running in C:\ProjectGames

[Pipeline] {

[Pipeline] bat

C:\ProjectGames>docker-compose up --build -d

Building web

Step 1/11 : FROM python:3-alpine

---> 0f03316d4a27

Step 2/11 : COPY GuessGame.py /

---> Using cache

---> 01a3994c4091

Step 3/11 : COPY Live.py /

---> Using cache

---> 98fad5683e07

Step 4/11 : COPY MainGame.py /

---> Using cache

---> 218abe1f1fa9

Step 5/11 : COPY MainScores.py /

---> Using cache

---> 499532e40fe1

Step 6/11 : COPY MemoryGame.py /

---> Using cache

---> 26f71f15d0eb

Step 7/11 : COPY Score.py /

---> Using cache

---> 7d2ce28c435a

Step 8/11 : COPY Scores.txt /

---> 509d3b7bc92c

Step 9/11 : COPY Utils.py /

---> 1399f8598e5a

Step 10/11 : RUN pip install flask

---> Running in 5e65e697f589

Collecting flask

Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)

Collecting click>=5.1

Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)

Collecting Jinja2>=2.10.1

Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)

Collecting itsdangerous>=0.24

Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)

Collecting Werkzeug>=0.15

Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)

Collecting MarkupSafe>=0.23

Downloading MarkupSafe-1.1.1.tar.gz (19 kB)

Building wheels for collected packages: MarkupSafe

Building wheel for MarkupSafe (setup.py): started

Building wheel for MarkupSafe (setup.py): finished with status 'done'

Created wheel for MarkupSafe: filename=MarkupSafe-1.1.1-py3-none-any.whl size=12627 sha256=b5d002f6787513c63500b7138f82d2e822fac8a9b06725fdc4295819ea8c6ad2

Stored in directory: /root/.cache/pip/wheels/0c/61/d6/4db4f4c28254856e82305fdb1f752ed7f8482e54c384d8cb0e

Successfully built MarkupSafe

Installing collected packages: click, MarkupSafe, Jinja2, itsdangerous, Werkzeug, flask

Successfully installed Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 flask-1.1.2 itsdangerous-1.1.0

Removing intermediate container 5e65e697f589

---> 961bb3d7489c

Step 11/11 : CMD ["python3", "./MainScores.py"]

---> Running in a16960c8543c

Removing intermediate container a16960c8543c

---> 4bb2a4f0a4db

Successfully built 4bb2a4f0a4db

Successfully tagged projectgames\_web:latest

projectgames\_redis\_1 is up-to-date

Recreating projectgames\_web\_1 ...

[1A[2K

Recreating projectgames\_web\_1 ... [32mdone[0m

[1B

[Pipeline] }

[Pipeline] // dir

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (pip install -r requirements.txt)

[Pipeline] dir

Running in C:\ProjectGames

[Pipeline] {

[Pipeline] bat

C:\ProjectGames>pip install -r requirements.txt

Requirement already satisfied: Flask==1.1.2 in c:\users\kaduris\appdata\local\programs\python\python37-32\lib\site-packages (from -r requirements.txt (line 1)) (1.1.2)

Requirement already satisfied: Werkzeug>=0.15 in c:\users\kaduris\appdata\local\programs\python\python37-32\lib\site-packages (from Flask==1.1.2->-r requirements.txt (line 1)) (1.0.1)

Requirement already satisfied: Jinja2>=2.10.1 in c:\users\kaduris\appdata\local\programs\python\python37-32\lib\site-packages (from Flask==1.1.2->-r requirements.txt (line 1)) (2.11.2)

Requirement already satisfied: itsdangerous>=0.24 in c:\users\kaduris\appdata\local\programs\python\python37-32\lib\site-packages (from Flask==1.1.2->-r requirements.txt (line 1)) (1.1.0)

Requirement already satisfied: click>=5.1 in c:\users\kaduris\appdata\local\programs\python\python37-32\lib\site-packages (from Flask==1.1.2->-r requirements.txt (line 1)) (7.1.2)

Requirement already satisfied: MarkupSafe>=0.23 in c:\users\kaduris\appdata\local\programs\python\python37-32\lib\site-packages (from Jinja2>=2.10.1->Flask==1.1.2->-r requirements.txt (line 1)) (1.1.1)

You are using pip version 10.0.1, however version 20.2.3 is available.

You should consider upgrading via the 'python -m pip install --upgrade pip' command.

[Pipeline] }

[Pipeline] // dir

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (e2e.py module it will perform a selenium test)

[Pipeline] dir

Running in C:\ProjectGames\Tests

[Pipeline] {

[Pipeline] bat

C:\ProjectGames\Tests>python e2e.py

Hello, please enter application URL (for example <http://192.168.99.102:5000/>)

Scores Game

10

the score in the good range

PASS

[Pipeline] }

[Pipeline] // dir

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (terminate our tested container)

[Pipeline] dir

Running in C:\ProjectGames

[Pipeline] {

[Pipeline] bat

C:\ProjectGames>docker-compose down

Stopping projectgames\_web\_1 ...

Stopping projectgames\_redis\_1 ...

[1A[2K

Stopping projectgames\_redis\_1 ... [32mdone[0m

[1B[2A[2K

Stopping projectgames\_web\_1 ... [32mdone[0m

[2BRemoving projectgames\_web\_1 ...

Removing projectgames\_redis\_1 ...

[1A[2K

Removing projectgames\_redis\_1 ... [32mdone[0m

[1B[2A[2K

Removing projectgames\_web\_1 ... [32mdone[0m

[2BRemoving network projectgames\_default

[Pipeline] }

[Pipeline] // dir

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

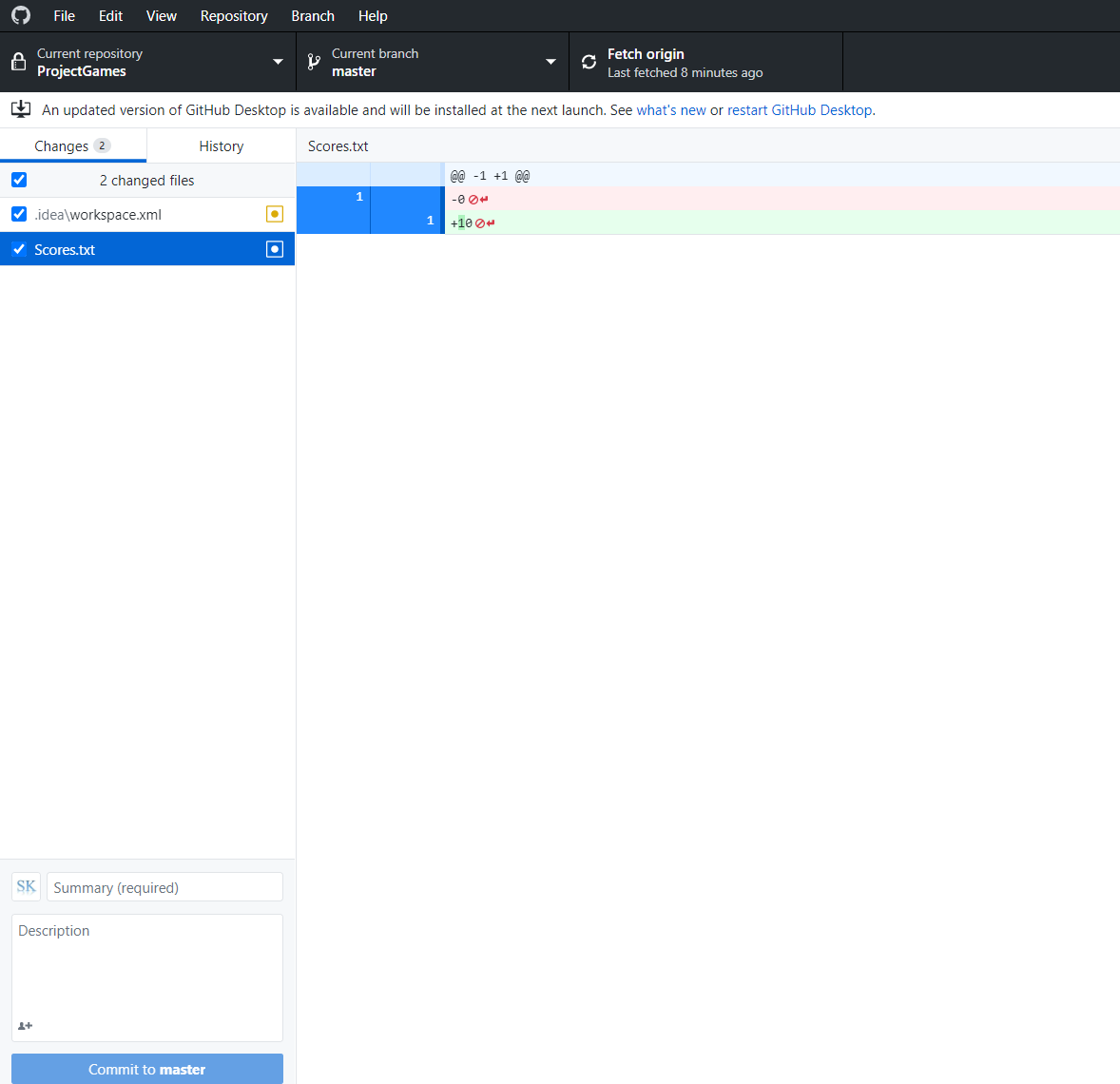
Finished: SUCCESS

In this project we use also “requirements.txt” in order to automatically install all required pip

We also use RUN PIP from Jenkins for try this also

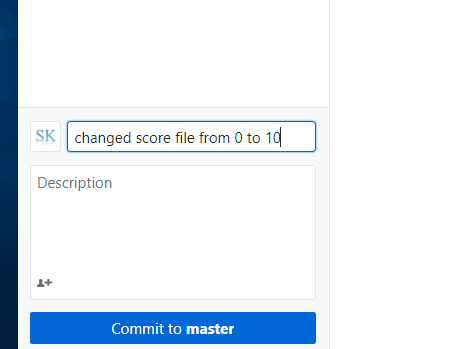
Git and GitHub changes, commits and fetch, push during the development – example

Change the score.txt

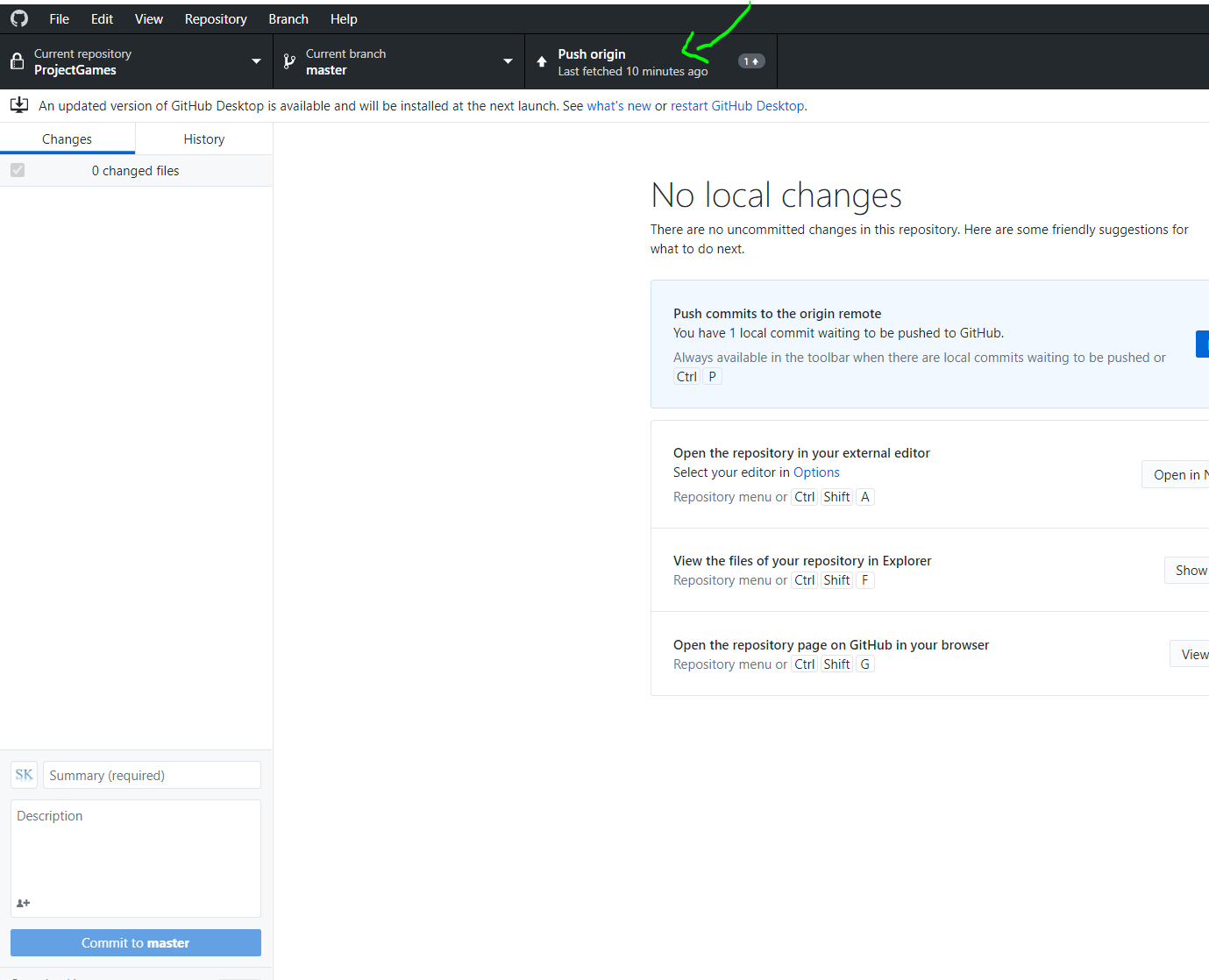


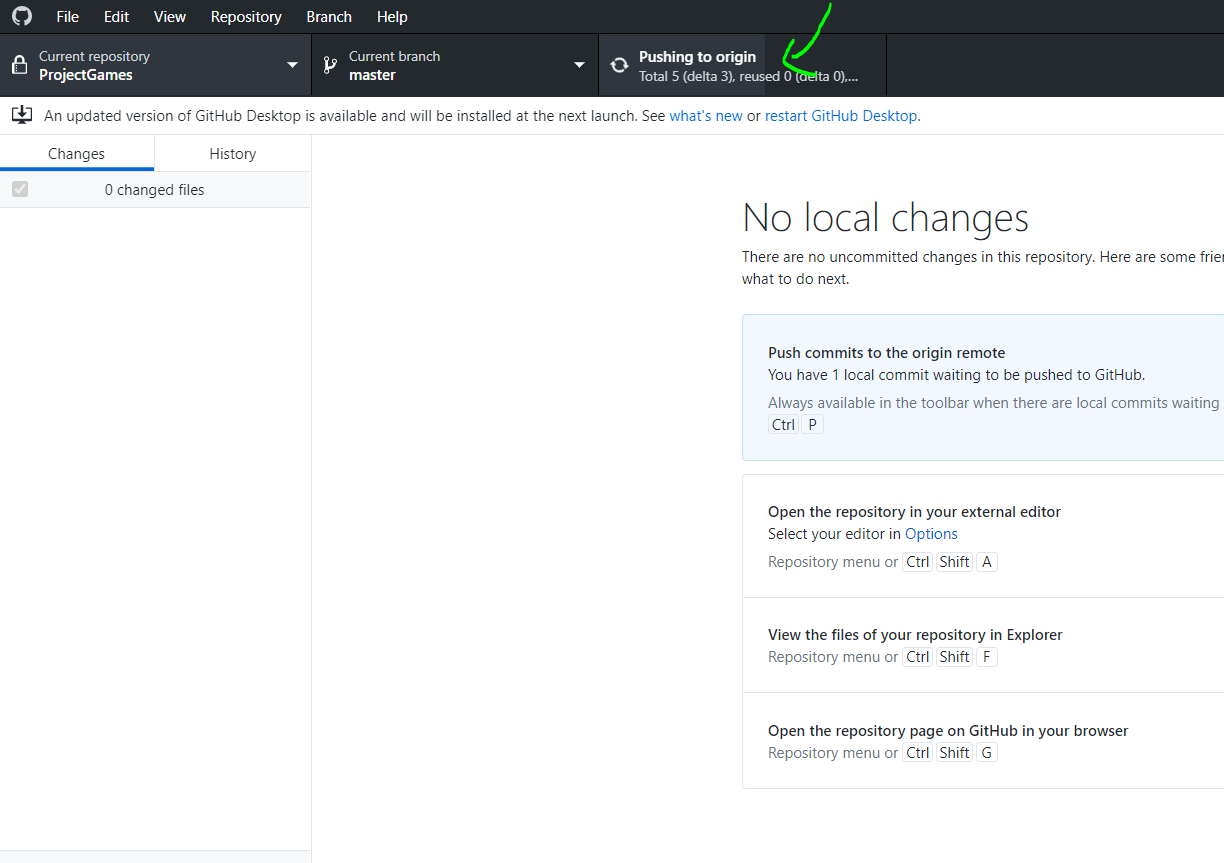
Automatedly detected that was change in score file.

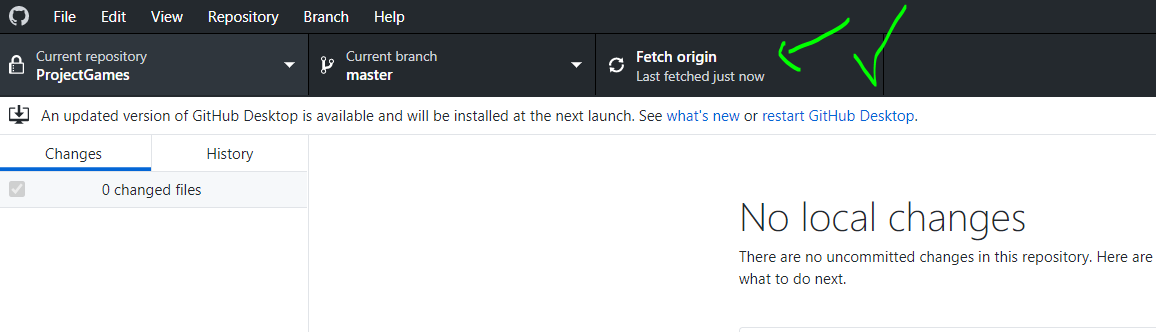
Here we need to add commit message



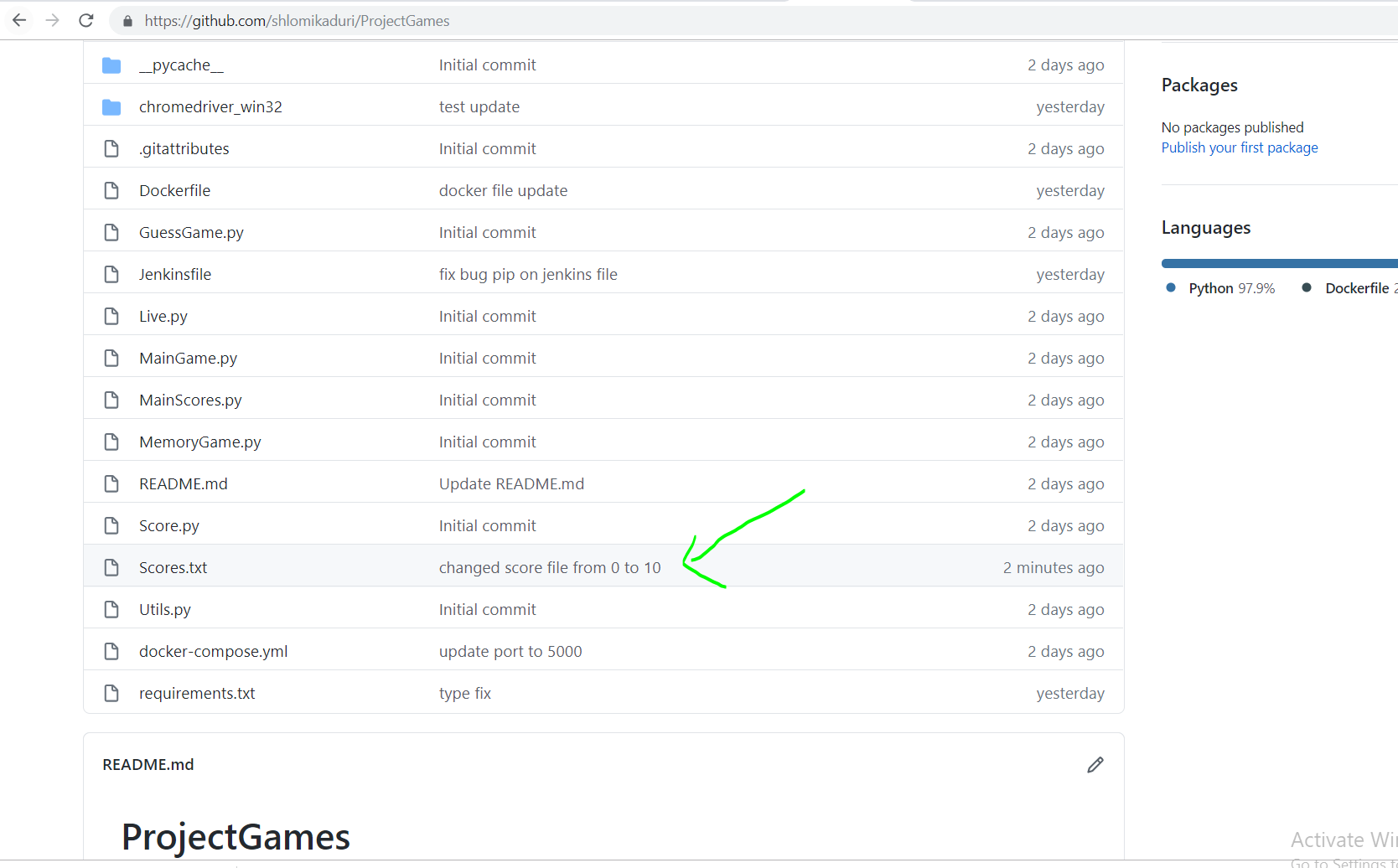
Now, we need to updated it to our GitHub by **push**





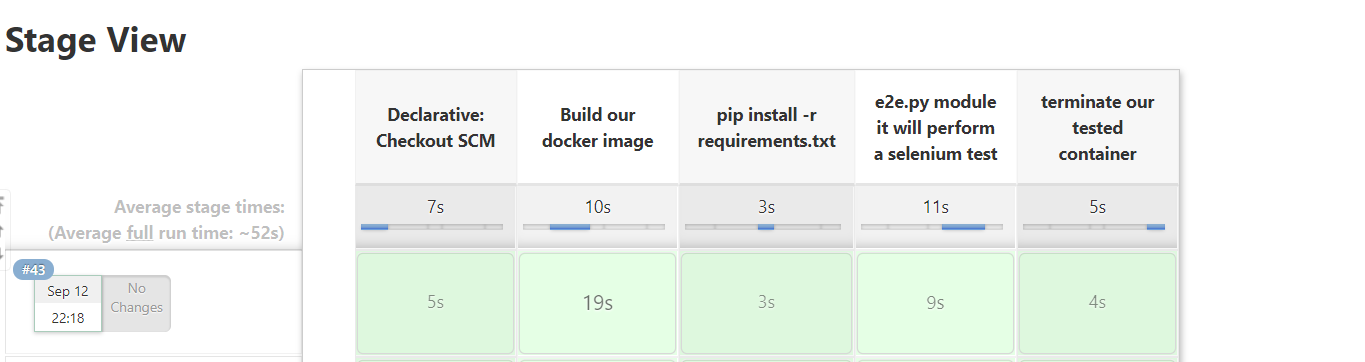


Now, we can see that GitHub (after refersh) was updated with our commit message



From this point, our Jenkins with pipeline can run and will

Fetches changes from the remote Git repository and run the updated files from rep to our projects and then to crate image and then container with these files.



The nice think here that all project including configuration is located in Github, and our Jenkins is connected to GitHub and run according to the configuration of Jenkins that come from GitHub, that is mean the Jenkins pipeline config and build on line according to the latest GitHub version.