ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING
SOFTWARE ENGINEERING II
CONTINUOUS INTEGRATION WORKSHOP - I TERM 2021

Objectives:

- Automatically run tests after uploading new code to an existing repository using Jenkins
- Obtain a report with the results of the tests.

Requirements:

- Jenkins >= 2.222.1
- Ngrok
- Eclipse with gradle tool
- Git

Introduction:

Continuous integration is a software development practice where a new code is joined to an existing code that is hosted in a repository. The union of codes comes with automatic execution of tests to ensure its correct operation[1].

The goal of continuous integration is to quickly detect and locate bugs, so reduce the time to fix and improve the software, and that updates are available as soon as possible.

To carry out this workshop, you must use Jenkins as a continuous integration tool. Jenkins is Open Source and runs as a server on the host machine; In addition, it has numerous plugins that increase its capabilities.

Additionally, you must use Ngrok, which is a tool that allows exposing a local url to a public url.

For this workshop, Jenkins will run locally on your computer, by uploading the changes to our remote repository hosted on Github, it will be able to send a response to our server, so it is necessary to have a public url to expose our local server to the internet. Here Ngrok intervenes that will act as a communication bridge between Github and Jenkins.

Activities

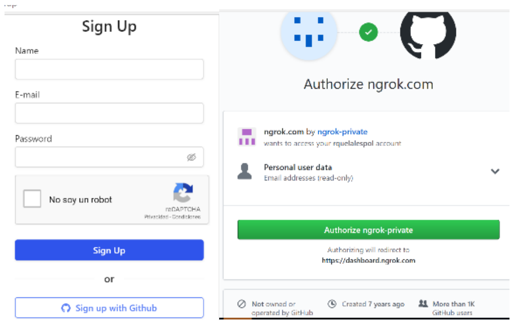
There is a small program used to compare the height and width of an object with the allowed dimensions. The relational operations implemented are: greater than and less than. When the object has the right dimensions, it is accepted. Otherwise, an error is displayed.

Step 1: Install Requirements

1. Install Jenkins
   a. Go to the following https://jenkins.io/download/
   b. Download Jenkins (LTS o Weekly) for your OS.
   c. Follow the steps shown in the following tutorial:

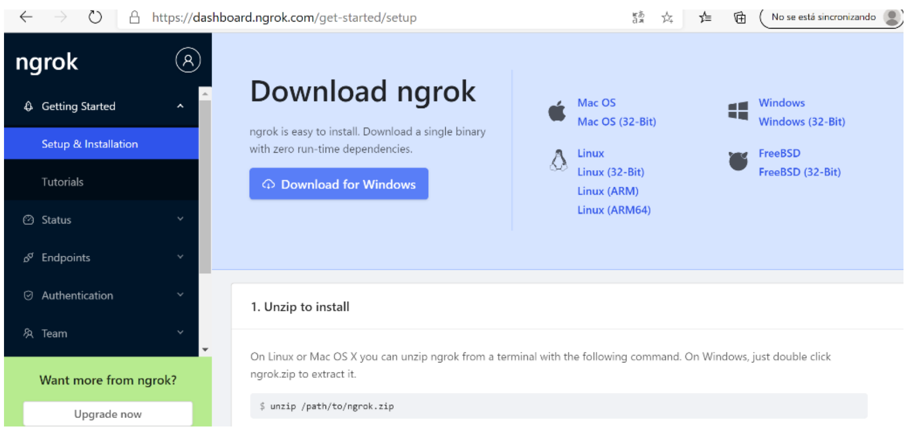
      Windows [2] o ubuntu[3]

2. Install ngrok
   a. Go to the following https://dashboard.ngrok.com/signup
   b. Create an account. Github is recommended.



Sign Up
Name
E-mail
Password
No soy un robot
reCAPTCHA
Privacidad - Condiciones
Sign Up
or
Sign up with Github
Authorize ngrok.com
ngrok.com by ngrok-private
wants to access your rquelalespol account
Personal user data
Email addresses (read-only)
Authorize ngrok-private
Authorizing will redirect to
https://dashboard.ngrok.com
Not owned or operated by GitHub
Created 7 years ago
More than 1K GitHub users

   c. Download the file.



ngrok is easy to install. Download a single binary with zero run-time dependencies.

On Linux or Mac OS X you can unzip ngrok from a terminal with the following command. On Windows, just double click ngrok.zip to extract it.

```
$ unzip /path/to/ngrok.zip
```

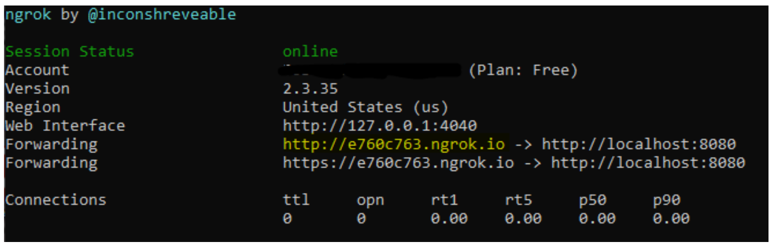
   d. Unzip the file.

```
//Replace <your authtoken> with the token given in the page
./ngrok authtoken <your authtoken>
```

   e. Open the program and enter the following command
      Note: The authtoken is in the 2nd item of the ngrok page.

   f. To test the operation, write the following command.

```
./ngrok http 8080
```



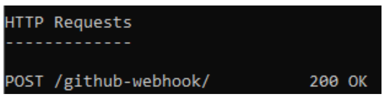
```
ngrok by @inconshreveable

Session Status                online
Account                       [redacted] (Plan: Free)
Version                       2.3.35
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://e760c763.ngrok.io -> http://localhost:8080
Forwarding                    https://e760c763.ngrok.io -> http://localhost:8080

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

g. Take note of the highlighted URL and do not close the console.

Note: ngrok will give you a random URL every time the command is entered. Do not close.

Step 2: Create a Repository

1. Create a Github repository
2. Set up your Jenkins connection:
   a. Go to Settings > Webhook > Add Webhook
   b. In Payload URL, place your url as follow:
   <your url>/github-webhook/.
   For example: http://e760c763.ngrok.io/github-webhook/
   c. In Content type, select application/json
   d. Leave the rest of the options as they are and click on Add WebHook
   e. Check the console again and verify the connection



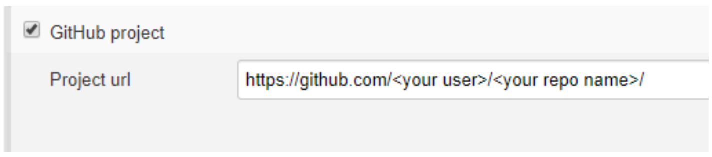
```
HTTP Requests
-------------

POST /github-webhook/          200 OK
```

Step 3: Prepare Jenkins

1. Select option “Manage Jenkins” > Manage Plugins
2. Select the Available tab and search “test-results-analyzer”
3. Mark the check Install y click “Install without restart”

Paso 4: Create a project in Jenkins

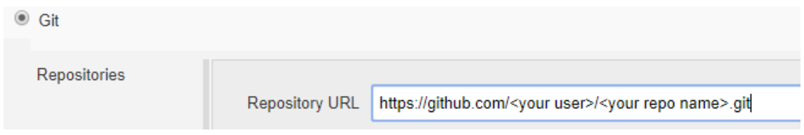
1. Select the option “New Item” in the left menu
2. Enter a name, select “Freestyle Project” and click Ok
3. In the tab “General”, mark Github Project and write your repository url.



☑ GitHub project

Project url: https://github.com/<your user>/<your repo name>/

4. In the tab “Source Code Management”, select git and enter the url again.



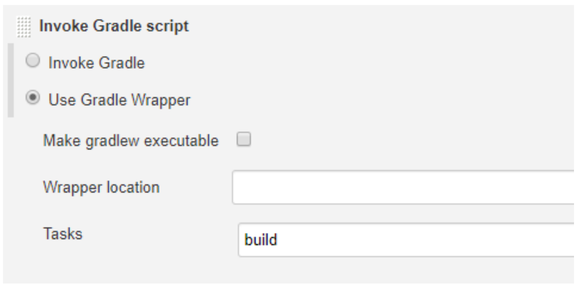
● Git

Repositories

Repository URL: https://github.com/<your user>/<your repo name>.git

5. In the tab “Build Trigger”, select the option “GitHub hook trigger for GITScm polling”
6. In the tab “Build”
   a. Select “Invoke Gradle script”

b. Enter data as follow


Invoke Gradle script

Invoke Gradle

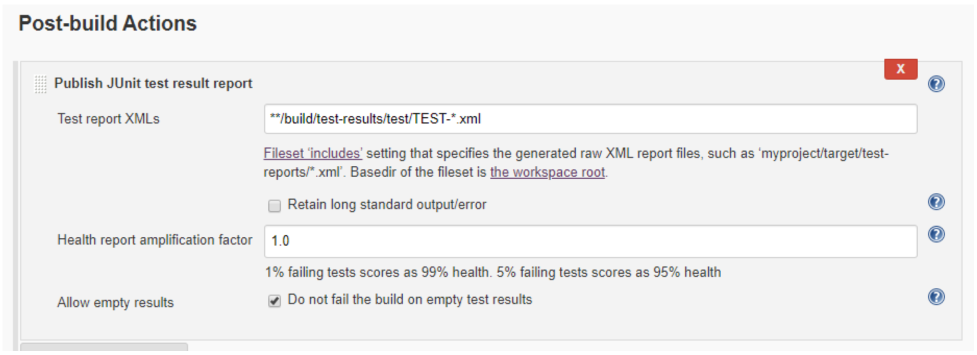
[✓ Use Gradle Wrapper]

Make gradlew executable ☐

Wrapper location

Tasks: build

7. In the tab “Post-build Actions”, enter data as follow.


Post-build Actions

Publish JUnit test result report

Test report XMLs: **/build/test-results/test/TEST-*.xml

Fileset 'includes' setting that specifies the generated raw XML report files, such as 'myproject/target/test-reports/*.xml'. Basedir of the fileset is the workspace root.

☐ Retain long standard output/error

Health report amplification factor: 1.0

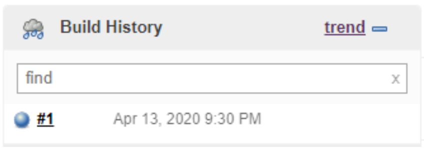
1% failing tests scores as 99% health. 5% failing tests scores as 95% health

Allow empty results: ☑ Do not fail the build on empty test results

8. Click Save.

Step 5: Check communication between Github and Jenkins

1. Clone this repository:
   https://github.com/leortyz/ContinuousIntegration
2. Open the project with Eclipse and verify the program and the test are running correctly.
3. Place the content in the repository created at step 1
4. Push to the repository
5. Review Jenkins and check a new build is running (lower left corner). The first build always takes more time, wait to the end and get the results.
6. When complete successfully, it will be marked in blue next to the build number. If it is marked in red, there are errors.


Build History trend

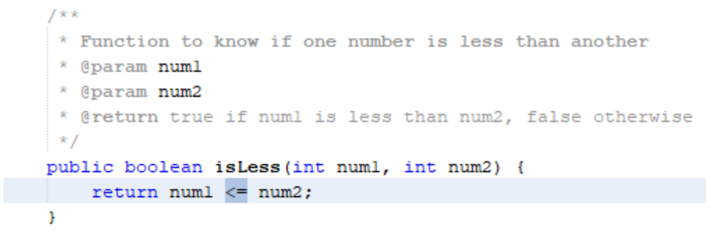
find

● #1 Apr 13, 2020 9:30 PM

7. From the side menu, select "Test Results Analyzer” to get charts about the tests performed and save those charts.

Development

After reviewing the code with the development team, a bug was found in one of the methods, although it does not show compilation errors and passes written tests, it is necessary to correct it to avoid further problems in the development process.



```
/**
 * Function to know if one number is less than another
 * @param num1
 * @param num2
 * @return true if num1 is less than num2, false otherwise
 */
public boolean isLess(int num1, int num2) {
    return num1 <= num2;
}
```

What you have to do:

1. Correct the error in isLess method in the correct relational operator, it should be only <
2. Upload changes to repository
3. Review the report obtained in Jenkins (save the graphics)

After review again, the error was corrected, but the Jenkins report indicates that not all tests have been passed successfully.

In addition, it was identified that missing test cases that have not been included and are necessary.

You should:

- Correct the necessary tests
- Add additional tests. There should be 6 tests in total, 3 for each method.
- Upload changes to repository
- Review Jenkins report (save charts)

Members by team: 3-5

Deliverables:

1. Practice report with at least: cover, introduction, development, conclusions and recommendations, and references.
2. Answer these questions in the report:
    a. What is continuous integration?
    b. Why is it important and almost necessary?
    c. In what type of projects is it recommended to apply it?
    d. What would be the result of not doing the continuous integration of a project?

    Note: Do not include the questions, these are only a guide.

3. In the development of the report, include screens of the practice process. 6 graphs should be about the tests process (pie and bars for each build, the line graph is not necessary)
4. Include in the report the url of the repository where you did the practice.

Rubric

Description	Value
Testing code (in the repository)	
Jenkins configurations	10
Correction of tests	20
Addition of tests	20
Jenkins reports	10
Practice report	
Completions of requested sections in the report	16
Quality screens of the process	12
Quality answers to questions	12
Total	100
Penalty per hour or fraction delay	-30
Penalty per deliverables that missing or not uploaded	-30

Referencias

[1] Amazon, «What is Continuous Integration?,» [En línea]. Available: https://aws.amazon.com/devops/continuous-integration/.

[2] P. Nguyen, «How to Install Jenkins on Windows,» 14 Julio 2018. [En línea]. Available: https://dzone.com/articles/how-to-install-jenkins-on-windows.

[3] E. Evans, «Installing Jenkins on Ubuntu 16.04,» 1 Agosto 2019. [En línea]. Available: https://www.liquidweb.com/kb/installing-jenkins-on-ubuntu-16-04/.

[4] «Workshop Specification Online», Available: https://github.com/leortyz/softwareEngineeringResources/wiki/Continuous-Integration