

**Muuk
Test**

Integrating MuukTest with CircleCi - Guidelines

Intro

MuukTest provides a set of tools that makes test creation easier and faster. Once the tests are created in the platform, it relies on the business needs on how to get them integrated into the continuous integration cycle. These tests can be executed directly on the MuukTest platform, on-premise or using any of the existing Continuous Integration (CI) providers.

This guideline covers the changes needed to execute the available MuukTest test cases under CircleCI.

Prerequisites

- You need a valid GitHub and CircleCI account and the projects need to be linked to each other. If you have not linked projects, follow the **CircleCI [steps](#)** to allow access to your Github repository.
- Create a MuukTest account with a minimum of one user in the organization.
- Use the content of the `config.yml` available in this document. This is a reference of the basic configurations needed.
 - Note: The provided `config.yml` works as reference, more complex configurations can be combined (eg. multiple docker support). However that is out of the scope of this document.
- MuukTest should have tests created. You can use hashtags to reference which tests you want to execute using CircleCi or provide a list of Test IDs.

MuukTest provides a CLI which handles the communication to the platform. Such a CLI is part of the GitHub project which will be downloaded (cloned) as a step of the CircleCI process.

Guidelines

Steps to execute MuukTest generated tests on CircleCI

1. Under your code project, include the file **config.yml** provided in the appendix to your `.circleci` folder. It is likely that you have another `config.yml` file already, so just add the sections that are important for your use case.

2. Download the execution key (*key.pub*) file. Refer to the MuukTest tutorial - [Tutorial](#))

User Information

Name: Renan 

Email: renan@muuklabs.com

Organization: RenanMuuklabs

Password

Change your account password

[Change Password](#)

Key

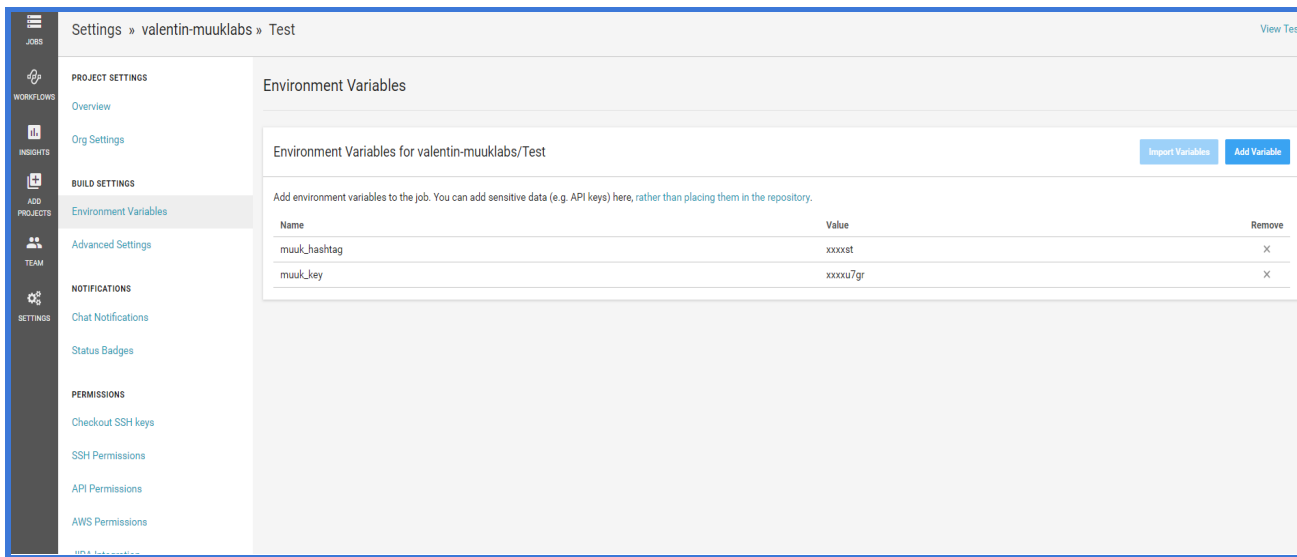
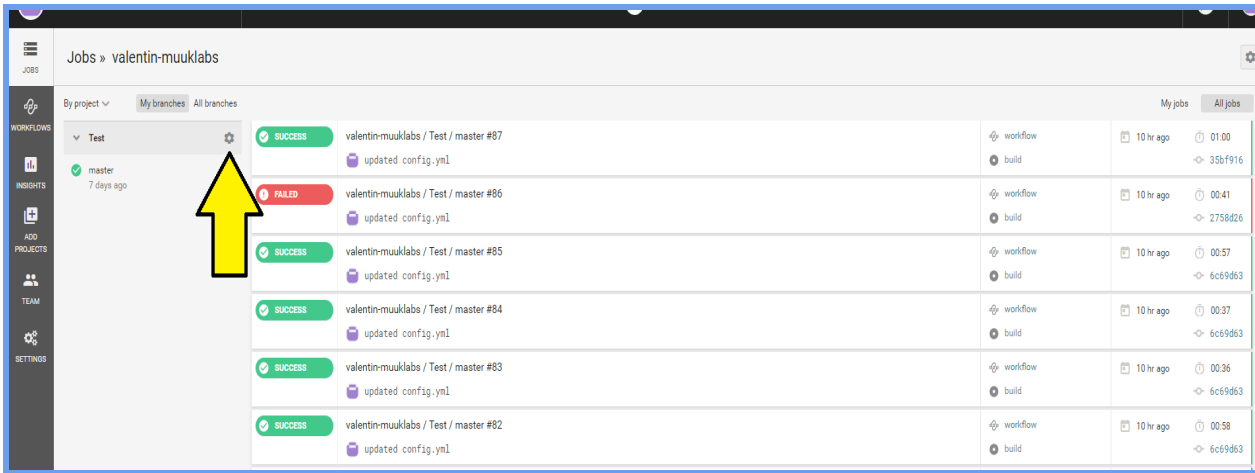
Download or refresh your key

 [Download](#)

 [Refresh](#)

3. Under CircleCi settings, create the environment variables needed for the execution. Adding an environmental variable is part of the settings section for each project. Add the following environment variables:

Name	Value
muuk_key	Copy the content of the previously downloaded executor key (key.pub). This key allows the communication between the server and MuukTest.
muuk_hashtag	Name of the tests grouped by hashtags that you want the ci to execute.
muuk_testId	Test Id of the test to execute. This is an individual test and it is optional.



Note: Updating your execution key in the MuukTest portal will invalidate your previous key. This means that you need to update the CircleCI environmental variable as well. Skipping this step would cause not being able to generate and download the scripts by the CI. In addition, if the user misses to add the muuk_hashtag variable, the default value to use would be #autotest

- After all these steps are implemented, CircleCI will execute the tests using the files on your repository each time It detects changes on your git repository. The results will be displayed on the MuukTest portal as regular tests under the *Executions tab*.

About the config.yml file

The *config.yml* file is a text file required by **CircleCI** to execute automatic testing. It contains the instructions to execute the tests in the form of [YAML](#) commands.

Once the file is part of your git repository, **CircleCI** looks for this file under the */.circleci* folder and executes the steps found. In addition to the commands to execute the tests, it also contains instructions to prepare the environment provided to execute the tests.

Further information about how syntax, configuration and expected commands in the [Configuration Introduction Guide](#)

The config.yml file suggested in the following section contains the following steps :

1. Use a Docker image provided by MuukTest (**muuk/muuktest_v1**) . This image comprises tools, libraries running on a Ubuntu 18.04 image. You can find the image hosted in [docker hub](#).
2. Prepare the environment to run the executor project.
3. Enable video recording.
4. Update the proper selenium driver for the browser.
5. Using the provided environmental variables (created in the step 3 of this document), it selects which tests cases to execute.
6. Run the executor project which downloads and runs your **MuukTest** tests. You can specify the testID of a single test or execute the tests using hashtags.

CircleCI - config.yml File content

```
version: 2
jobs:
  build:
    docker:
      # specify the version you desire here
      - image: muuk/muuktest_v1:0.1
    environment:
      GRADLE_OPTS: -Xmx1024m -XX:MaxMetaspaceSize=256m
      GRADLE_USER_HOME: .gradle-home

      # Specify service dependencies here if necessary
      # CircleCI maintains a library of pre-built images
      # documented at https://circleci.com/docs/2.0/circleci-images/

    steps:
      - run:
          command: |
            Xvfb :99 -screen 0 1366x768x16 -nolisten tcp -fbdir /var/run >
/dev/null 2>&1 &
            export DISPLAY=:99
            git clone https://github.com/muuklabs/executor.git
            cd executor/
            git checkout videoEnabled
            printf $muuk_key > key.pub
            cat key.pub
            chmod 755 gradlew
            chmod 755 key.pub
            sed -i 's/79.0.3945.36/90.0.4430.24/g' build.gradle
            cat build.gradle
            if [ -z ${muuk_hashtag+z} ]; then
            python3.7 mkcli.py -p hashtag -t $muuk_hashtag
            else
            python3.7 mkcli.py -p hashtag -t $muuk_testId
            fi;
```

Reference

Any comments related with this document please contact renan@muuklabs.com or jesus.cabada@muuklabs.com