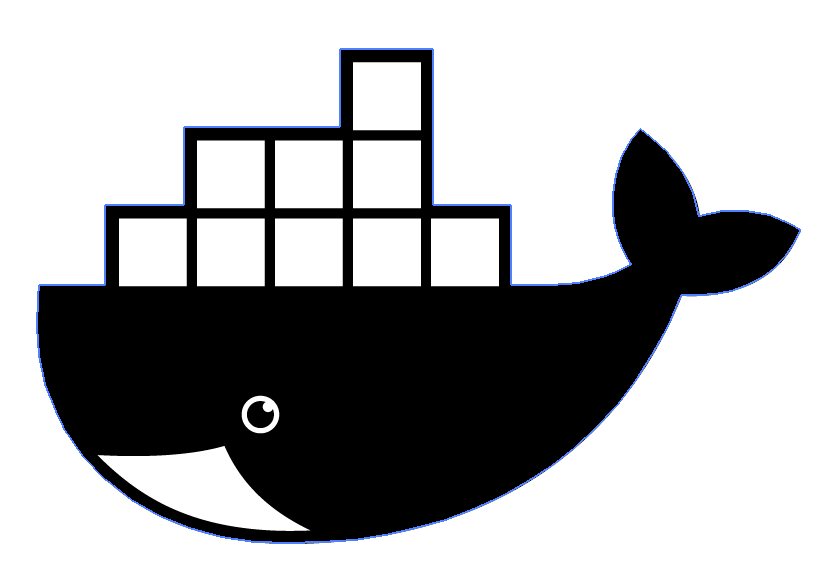
# Installing and running Docker and Che

## Step 1: Download and Install Docker

* Docker is the software container for this project.
* Install Docker toolbox on Windows (versions before Windows 10, Windows 10 Home):  
  <https://www.docker.com/products/docker-toolbox>
* Install Docker on Windows (Windows 10 Professional or Enterprise 64-bit):  
  [https://store.docker.com/editions/community/docker-ce-desktop-windows](https://store.docker.com/editions/community/docker-ce-desktop-windows?tab=description)
* Install Docker toolbox on Mac (versions before Mac OS Yosemite 10.10.3):  
  <https://www.docker.com/products/docker-toolbox>
* Install Docker on Mac (Mac OS Yosemite 10.10.3 or above):   
  <https://www.docker.com/docker-mac>
* *Note*: Installing this will require Admin privileges on your computer
* *Note:* Known Docker issue on Windows 7.

## Step 1.5: Run Docker

* Run Docker from Applications.
* On a Mac: The ‘Docker Whale’ will show up in the application menu bar while it is running.

## Optional Step: Download and install KitematicScreen Shot 2017-03-02 at 10.25.17 AM.png

* Kitematic provides GUI for Docker. It is available for Mac OS X 10.8+ and Windows 7+ (64-bit).
* Kitematic comes in the Docker toolbox - you do not need to download if you had to install Docker Toolbox in a previous step.
* On a Mac, download Kitematic from the Docker application. Click on the ‘Docker Whale’ in the application menu bar, and select “Kitematic” from the dropdown (see screenshot).

## Step 2: Start Che

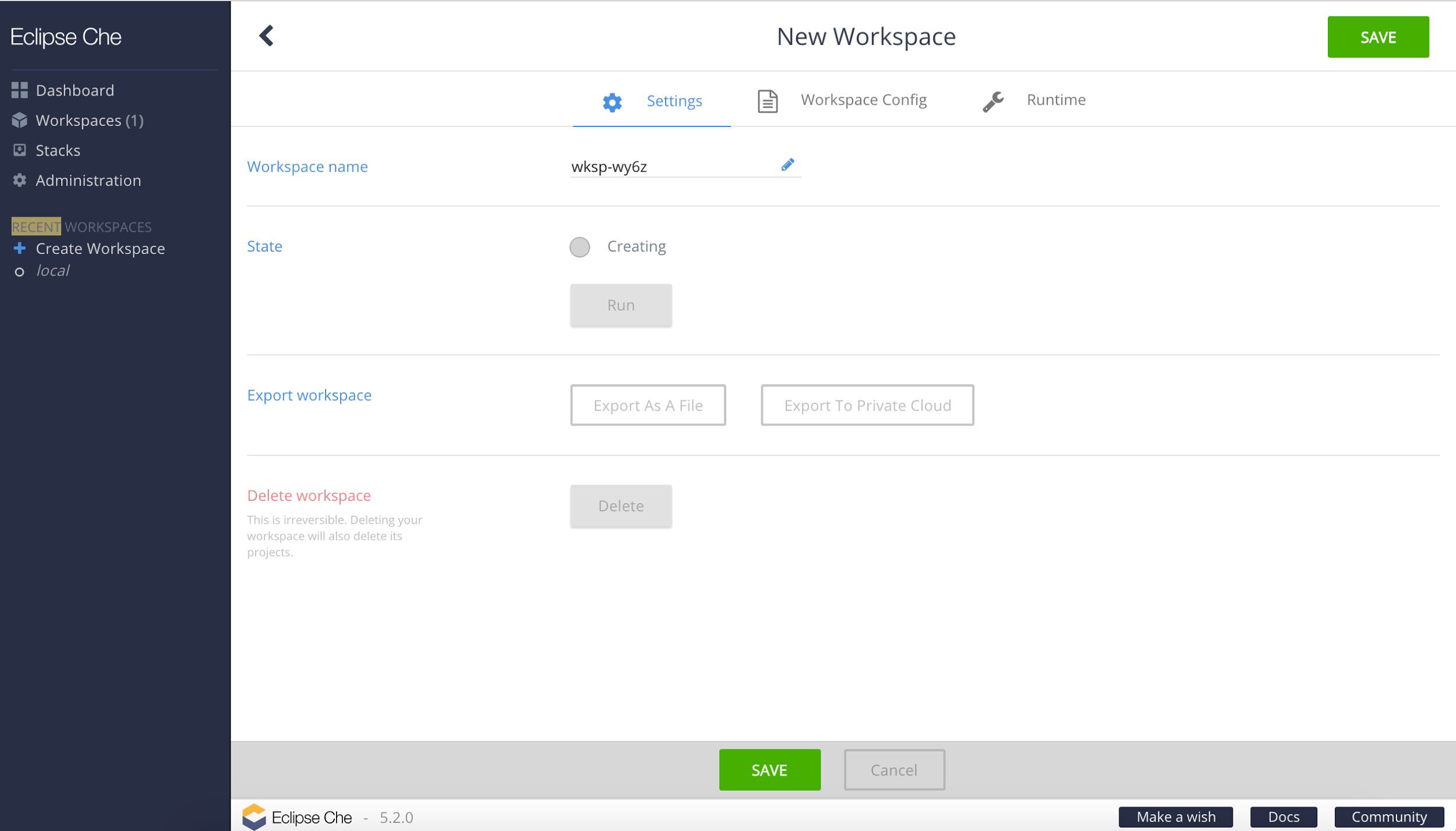
* Open Terminal (Mac) or Powershell (Windows).   
  *Note*: For Windows, the path name did properly link in Mingw. Use /D/<path> instead of D:<path>
* If you have a Mac:
  + cd to the folder in which you want to run Che (you may want to create a new empty folder for this project)
    - cd <folder name>
  + Run the code found in file “che\_start\_mac.sh”
    - docker run -it --rm -e CHE\_PORT=7000 -v /va r/run/docker.sock:/var/run/docker.sock -v $PWD/data:/data eclipse/che:5.2.0 dir up
  + View Che at: localhost:7000/
* If you have a Windows:
  + Run the code found at <http://www.eclipse.org/che/docs/setup/getting-started/>
    - docker run -it --rm -v /var/run/docker.sock:/var/run/docker.sock -v <path>:/data eclipse/che start
    - <path> should be the path to the folder in which you want to run Che (you may want to create an empty folder for this project)
  + View Che at: localhost:8080/

## Step 3: Ensure you have Github and Docker hub account

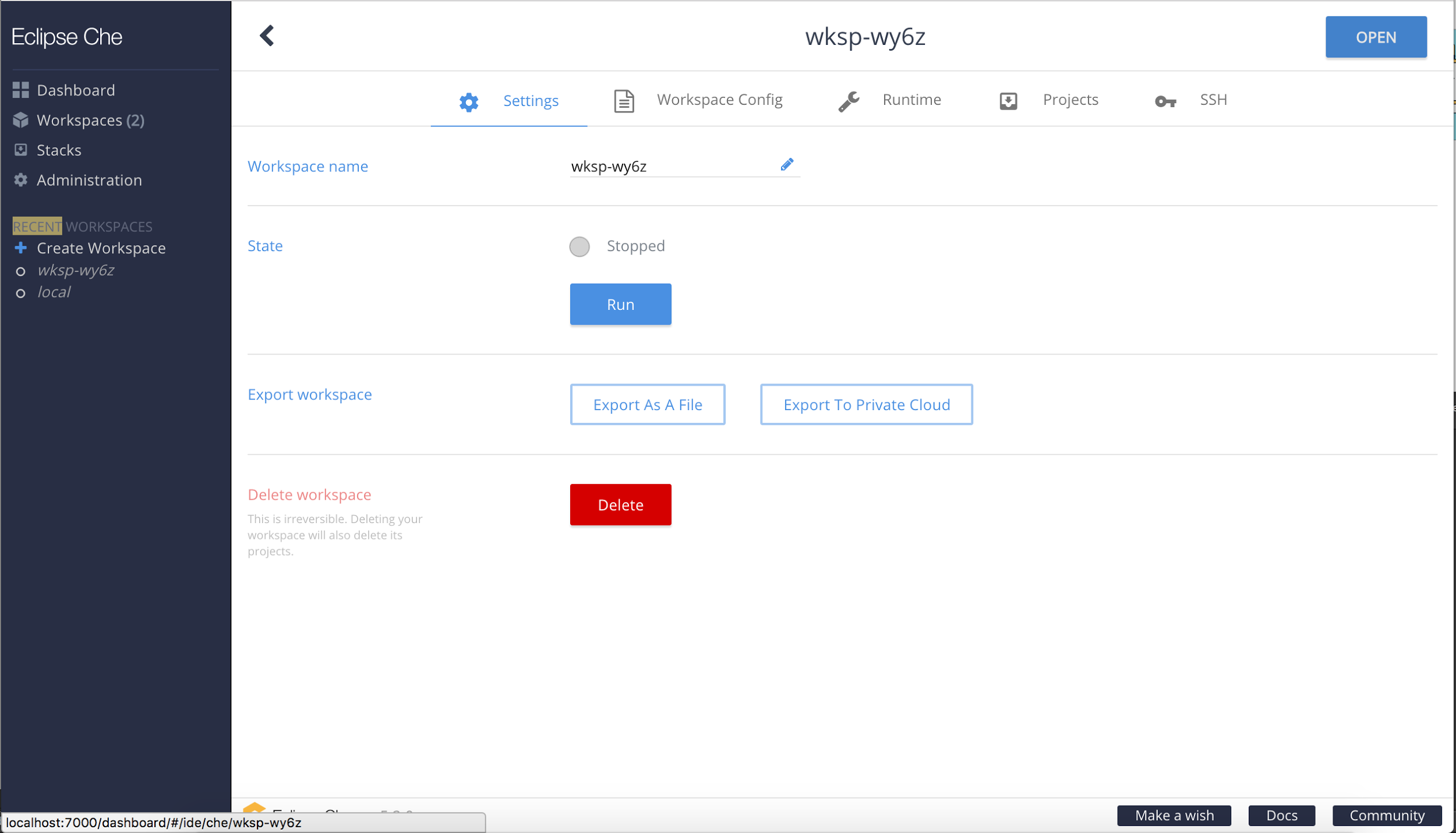
* Create a github account here: [http://www.github.com](http://www.github.com/)
* Create a docker hub account here: [http://hub.docker.com](http://hub.docker.com/)
* DevEnvy’s Github and Docker hub
  + <https://github.com/DevEnvy-HackWithFriends>
  + <https://hub.docker.com/u/devenvy17/>

# Create a new workspace

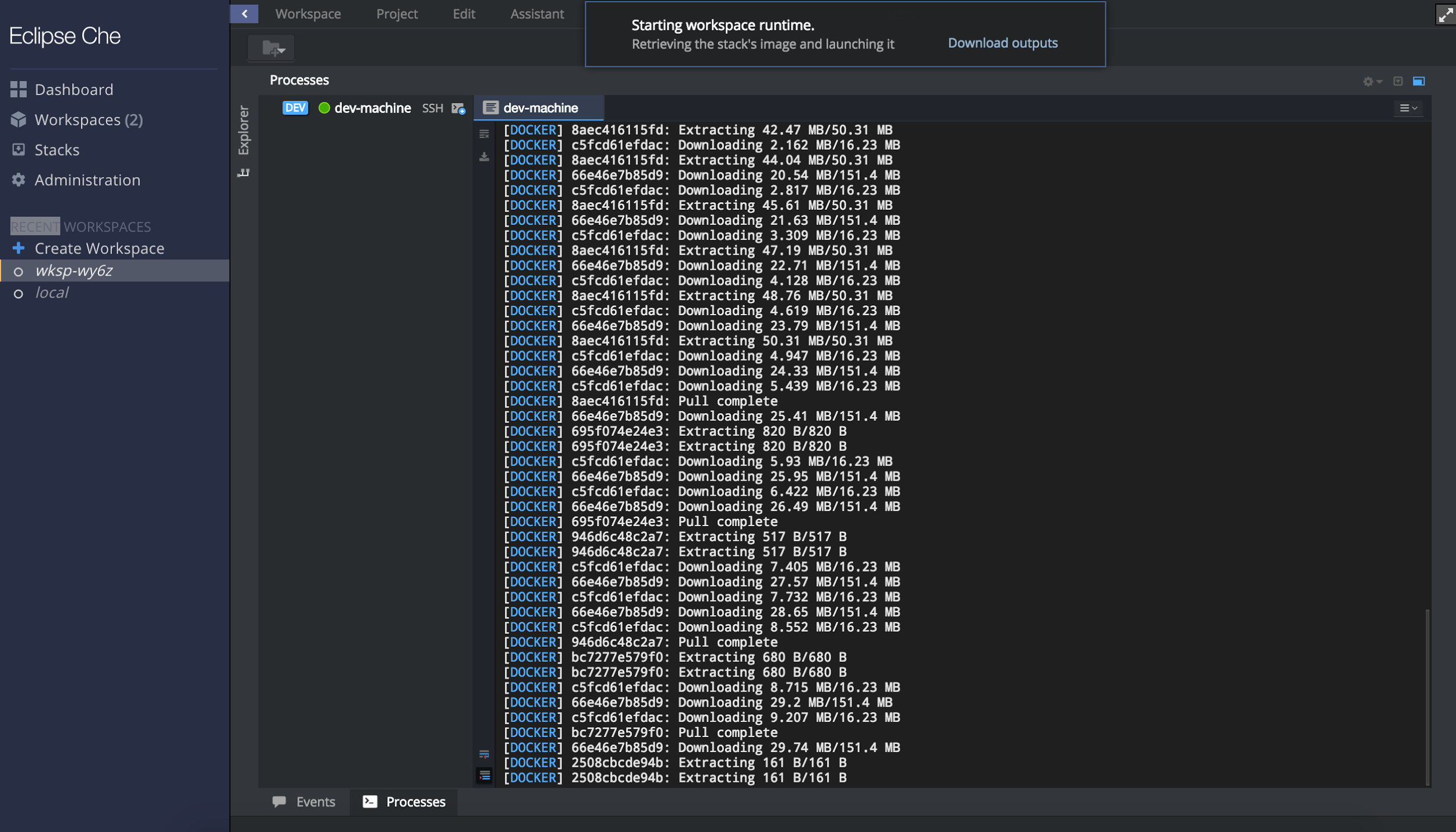
In Che, go into “Create Workspace”



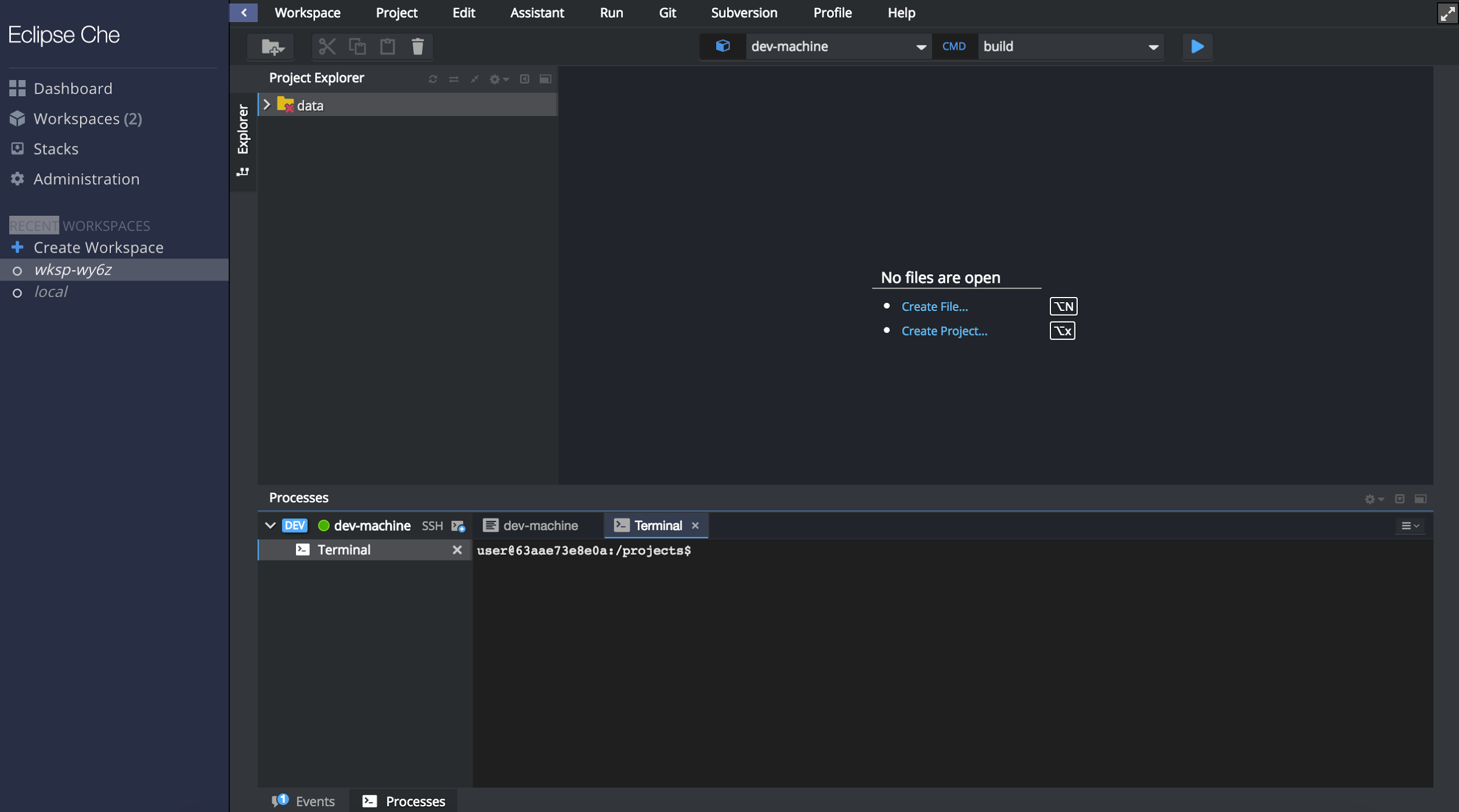
We did not change any of the configurations. Click “Save”



From the workspace, click “Open” (top right corner).



When you click “Open”, Che will start up the dev-machine in the Che workspace.



Here you can see the open Che workspace.

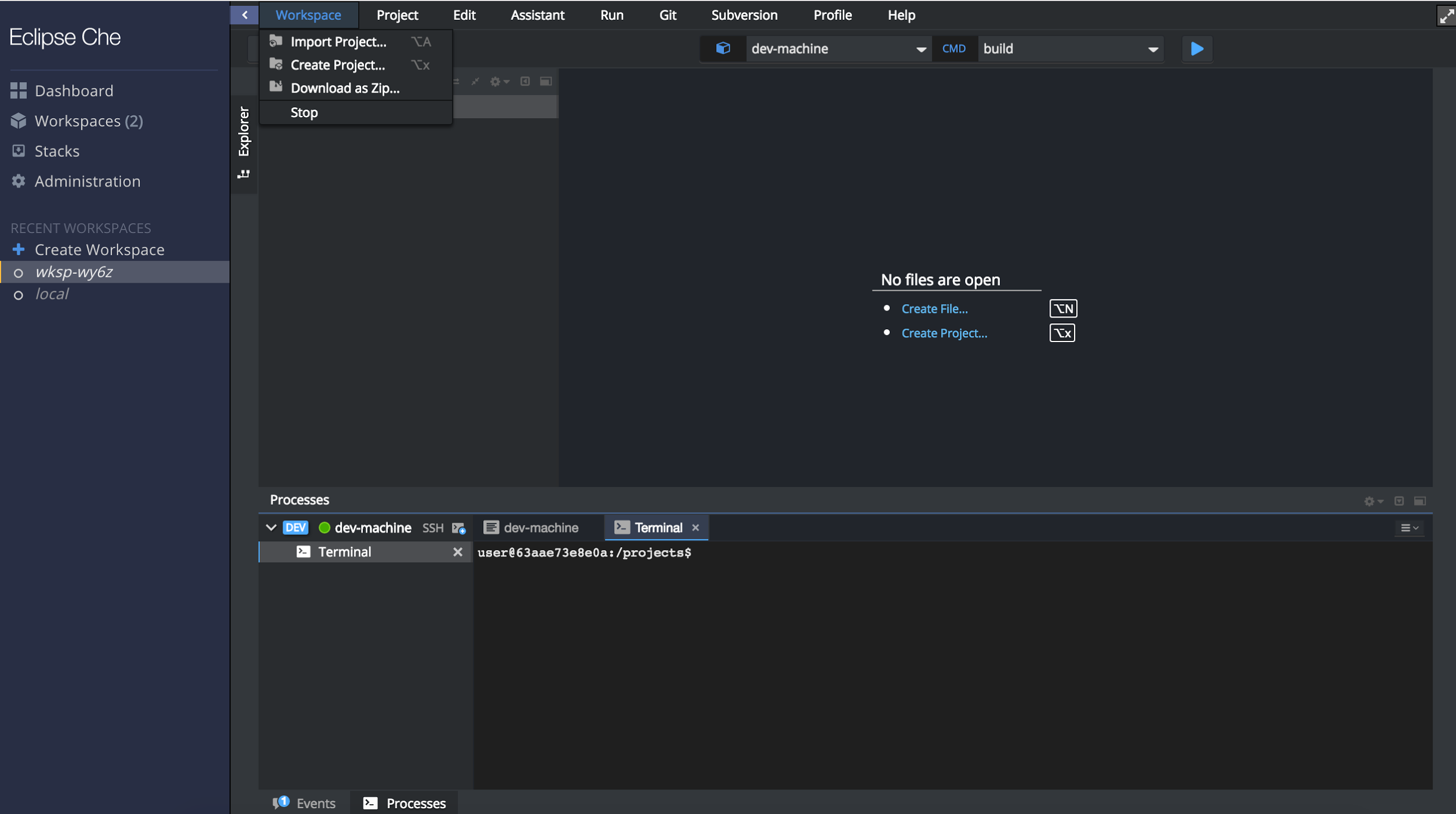
# Import project from GitHub (Java Springboot)

## Step 1: Find project on Github

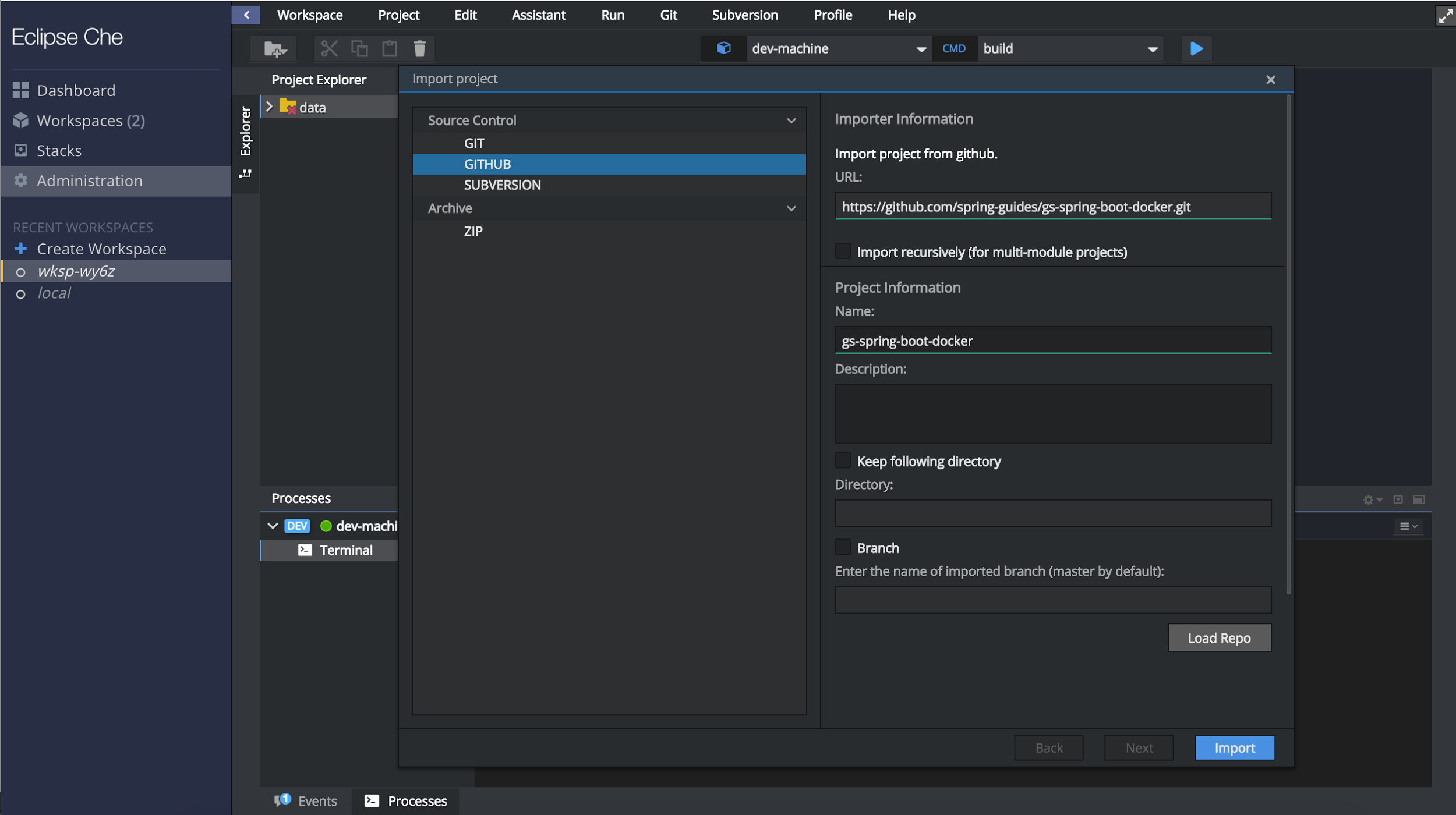
* For Java Springboot project, copy url at: [https://spring.io /guides/gs/spring-boot-docker/](https://spring.io/guides/gs/spring-boot-docker/)
* This project already has Docker file.

## Step 2: Import GitHub project

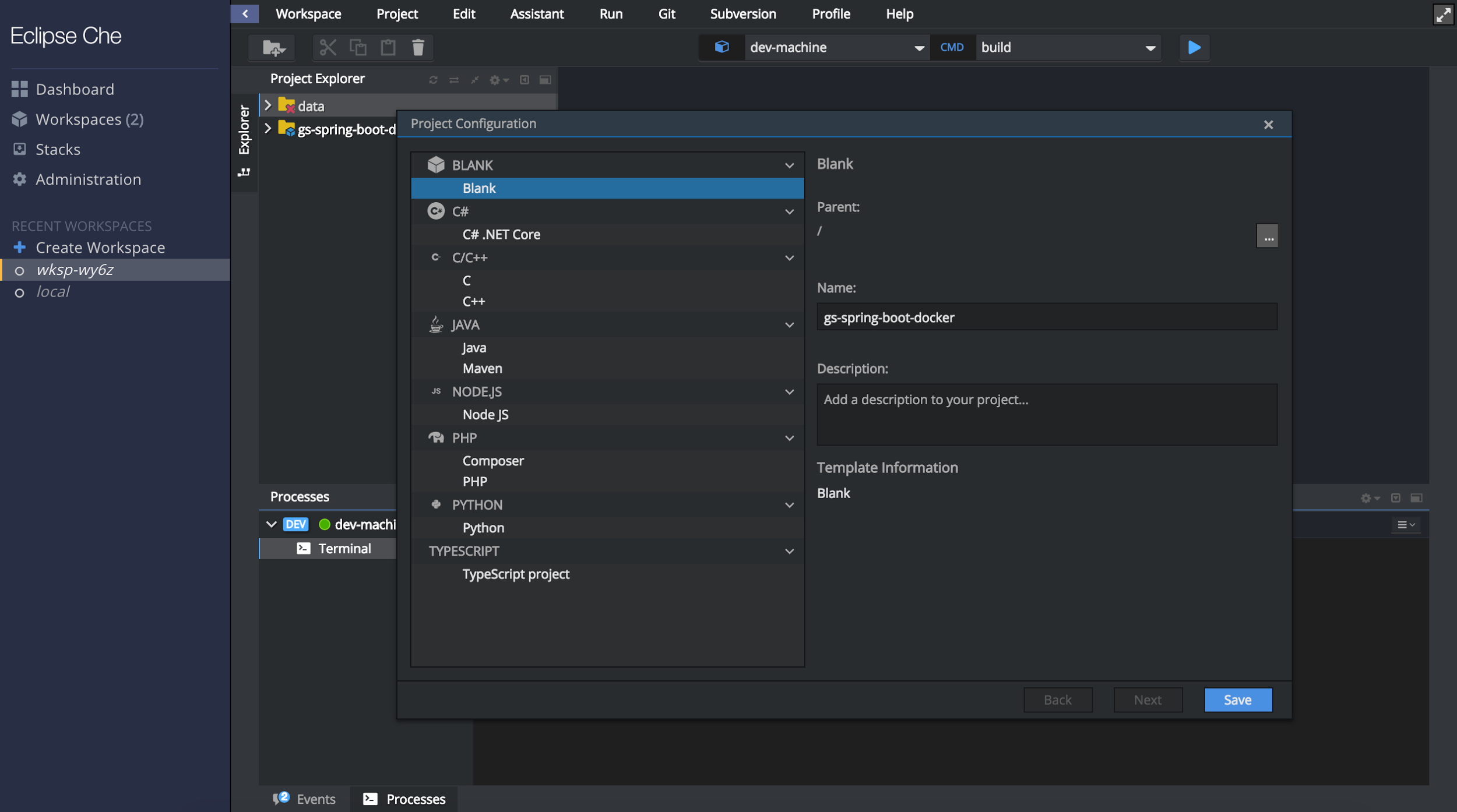
Create a new Workspace.



Import project by selecting “Workspace” < “Import Project…” from the Che menu bar.

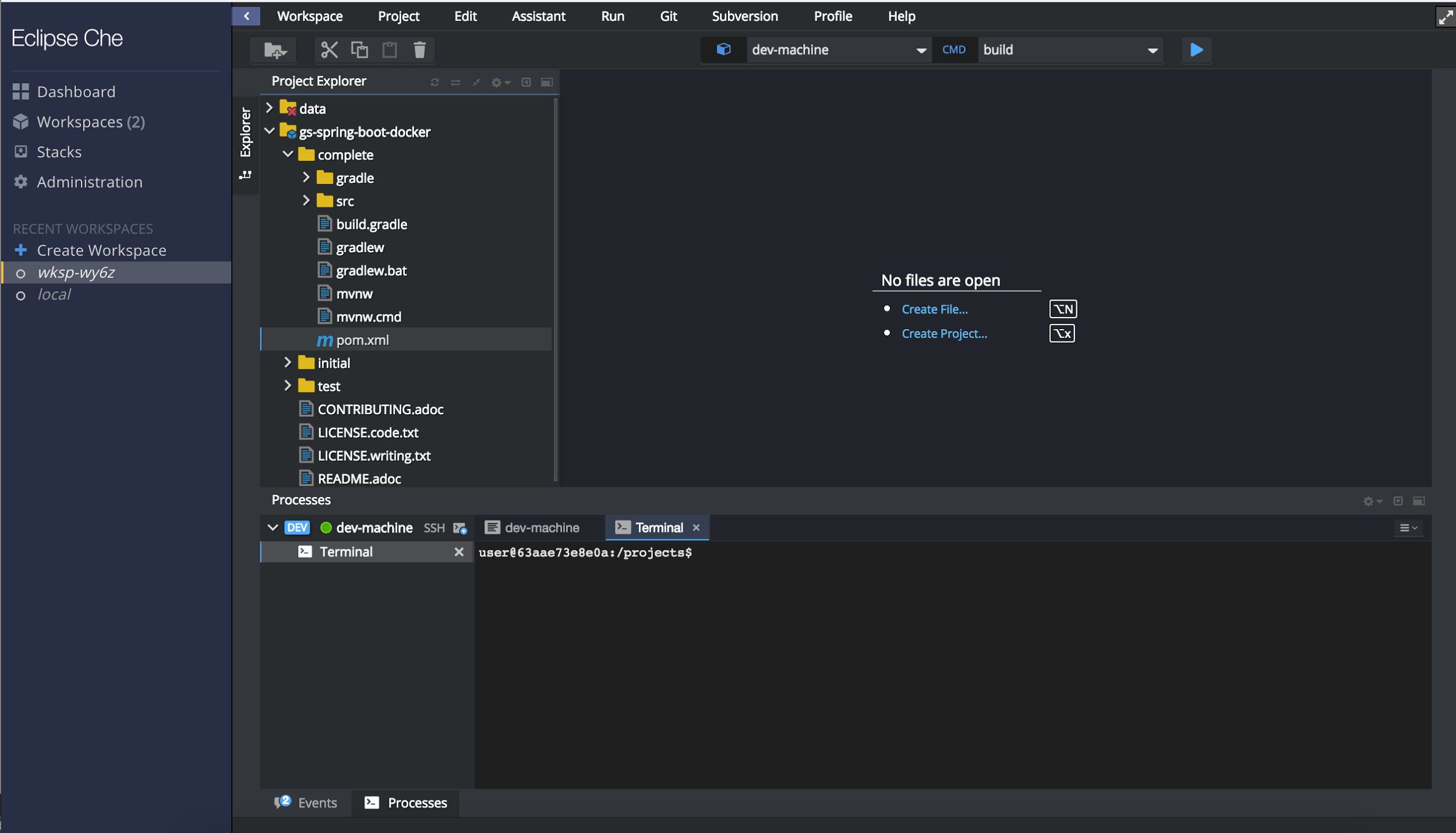


Select, “Github” and past in the copied url from Github (step 1)

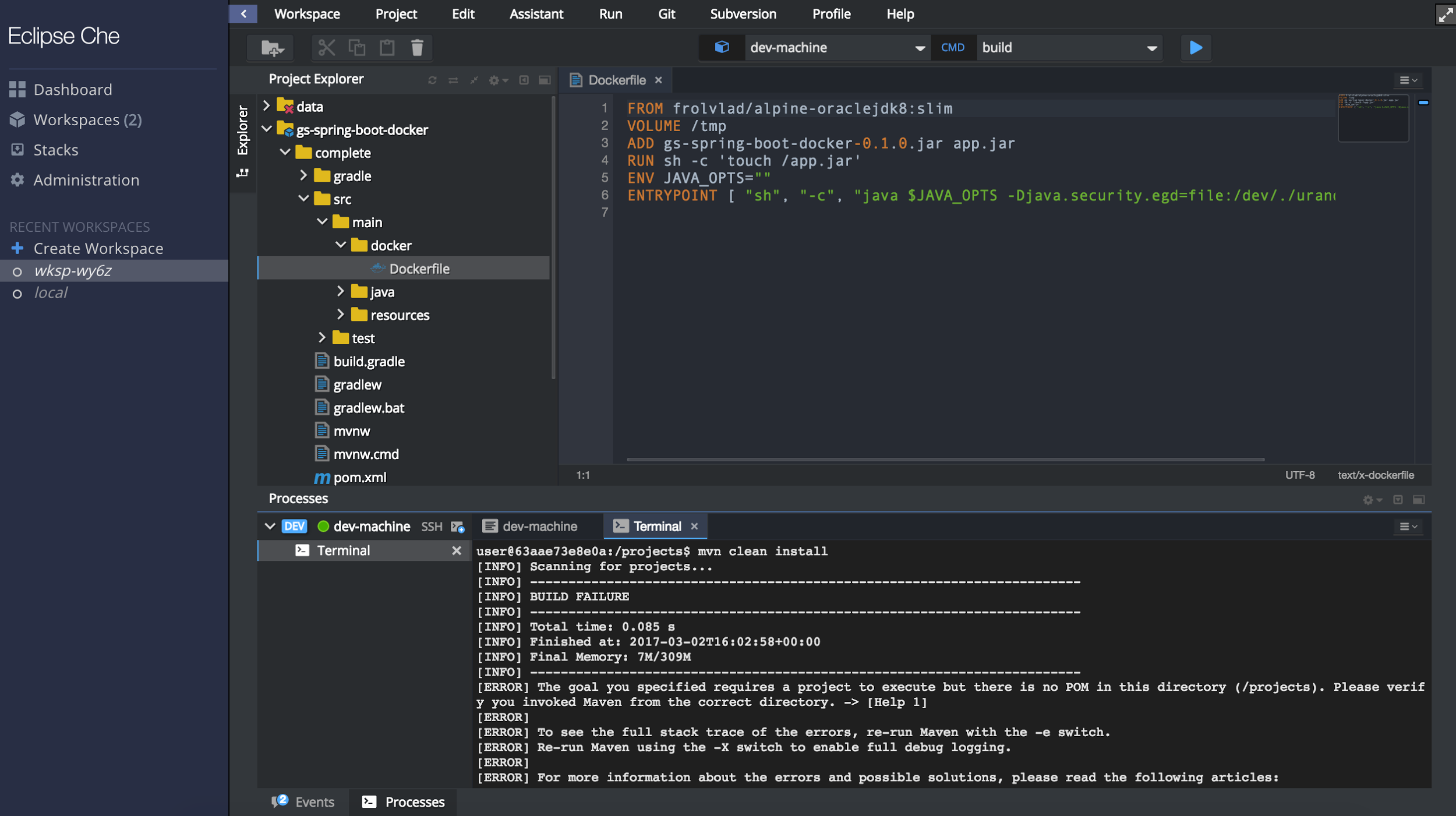


You can see the project has imported. We did not change any of the configurations for this project. Click “Save”

## Optional step: Explore the imported project



In this screenshot we can see the pom.xml file within the imported project.



In this screenshot we can see the Dockerfile, which helps you build the docker image. See <https://spring.io/guides/gs/spring-boot-docker/#_containerize_it> for more details (about halfway down the page, see “Containerize It”)

## Step 3: Build Project

* For this example we built our project in Maven.
* Cd to folder with pom.xml in terminal within Che
  + cd gd-spring-boot-docker/complete
* Run
  + mvn clean build

## Step 4: Run Project

* Test that project is running at localhost

# Import project from Docker hub (Python)

## Step 1: Find project on Github

# Troubleshooting: Windows

* If you are having trouble installing Docker on a Windows 10, ensure it is using Windows 10 Professional. Windows 10 Home requires install of the Docker Toolbox.
* If you experience trouble starting Che, ensure you are using Powershell instead of Mingw. Use syntax /D/<path> instead of D:<path>
* If you experience trouble starting the Docker VM, start Kitematic and choose “Use Virtualbox” instead of the default virtual environment.