# Mise en place d’une plate-forme d’Integration Continue

## Installer Jenkins avec Docker

* Pour éviter d’installer Jenkins sur notre ordinateur, c’est possible d’utiliser un conteneur Docker
* Faudra que le plugin Docker soit incluse dans les outils Jenkins, pour faire ça utiliser le suivant Dockerfile:

FROM jenkins/jenkins:lts

USER root

RUN apt-get update

RUN curl -sSL https://get.docker.com/ | sh

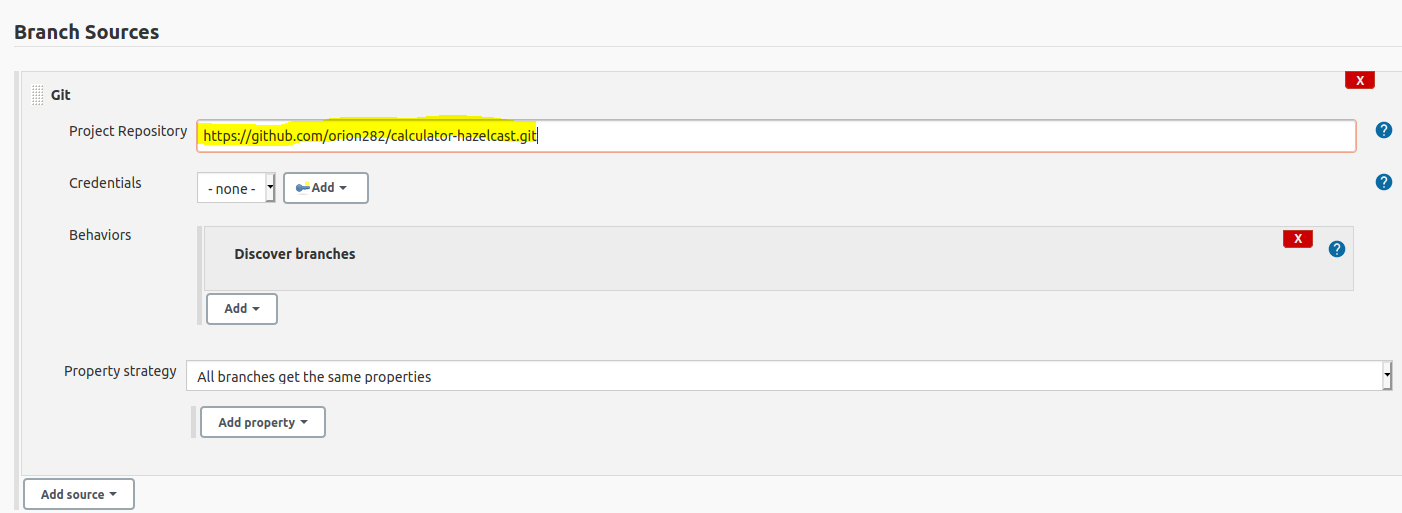
* Maintenant est possible créer l’image Docker:
  + sudo docker build –t username/jenkins (“orion282/jenkins” dans ce cas)
* ...et lancer le conteneur Jenkins:
  + sudo docker run -d -p8080:8080 -v /var/run/docker.sock:/var/run/docker.sock --name jenkins\_manu orion282/jenkins

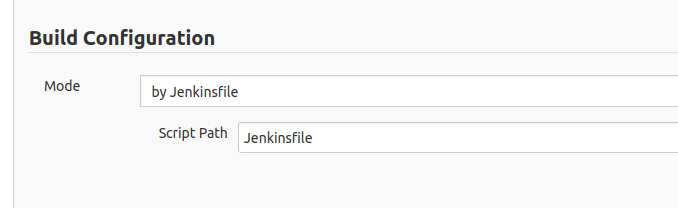
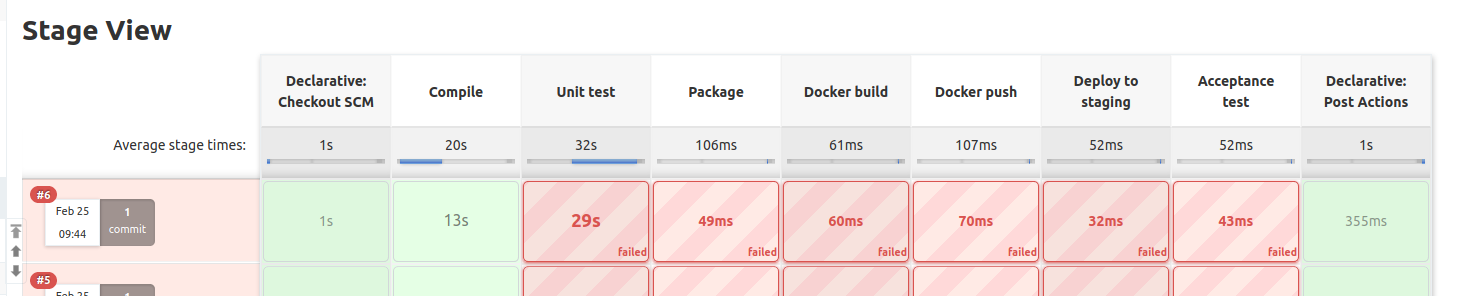
## Configurer Jenkins

* Maintenant faut ouvrir jenkins et le configurer: localhost:8080
* A la fenêtre qui s'ouvre, faut mettre la mot de passe d'administrateur. Pour récuperer ce mdp faut ouvrir le log du conteneur jenkins:
  + docker logs jenkins\_manu
  + Copier la mdp:
  + Jenkins SeCup is requüed. An admin user has been created and a 
    Please "se the following password to proceed to installation: 
    dc9b1481d5f343d28384fdb08d2e414b 
    password generated. 
    ivar/jenkins_home/secrets/tnittaudmtnpassword 
  + Coller la mdp et cliquer sur "Continue"
  + Pour configurer Jenkins, cliquer sur "Install suggested plugins" :
  + Getting Started 
    Customize Jenkins 
    Plugins extend Jenkins with additional features to support many different needs. 
    Install suggested 
    plugins 
    e Jenkins 
    community finds most 
    useful. 
    Select plugins to 
    install 
    Select and install plugins 
    most suitable for your needs. 
  + Attendre que l'installation des outils soit complet et créer propre compte, à utiliser pour les prochains logins:
  + Getting Started 
    Create First Admin User 
    username: 
    manu@mail 
    Password: 
    manusamp 
    Confirm password: 
    Full name: 
    E-mail address: 
    Jenkins 22638 
    Skip and continue as admin 
    Save and Continue 
  + Maintenant on peut utiliser Jenkins.

## Creation Pipeline

* + Dashboard [Jenkins] 
    Jenkins 
    Dashboard 
    New Item 
    People 
    Build History 
    Manage Jenkins 
    My Views 
    Lockable Resources 
    New View 
    Build Queue 
    NO builds in queue. 
    localhost:gogo 
    q 
    Welcome to Jenkins! 
    searcn 
    O 
    manusamp log out 
    ['add description 
    This page is where your Jenkins jobs will be displayed. To get started, you can set up 
    distributed builds or start building a software project. 
    Start building your software project 
    Create a Job 
    Set up a distributed build 
    Set up an agent 
    Configure a cloud 
    Learn more about distributed builds 
  + Cliquer sur "New Item" et créer une multi-branch pipeline nommé "calculator-hazelcast":
  + Enter an item name 
    calculator-branches 
    Required field 
    Freestyle project 
    This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build. 
    pipeline 
    Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly kncpwn as workflows) and/or organizing complex activities that do not easily fit in free-style job 
    type. 
    Multi-configuration project 
    Suitable for projects that need a large number Of different configurations, such as testing on multiple environments. platform-specific builds, etc. 
    Folder 
    Creates a container that Stores nested items in it. Useful for grouping things together. unlike view, which is just a filter, a folder creates a Separate namespace, So you can have multiple things of the same name 
    as long as they are in different folders. 
    GitHub Organization 
    Scans a GitHub organization (or user account) for all repositories matching some defined markers. 
    GitHub Organization 
    Scans a GitHubor anization (or user account) for all repositories matching some defined markers. 
    Multibranch Pipeline 
    Createsa setofPi eli 
    according to detected branches in one SCM repository. 
  + Dans "Branch Sources", cliquer sur "Add source", choisir "Git" et mettre l'adresse du repository GitHub:



* In "Build Configuration" choisir "by Jenkinsfile" (le Jenkinsfile avec la pipeline est inclus dans cet repository):
* 
* Vers le bas, cliquer sur "Periodically if not otherwise run" et mettre 1 minute comme Interval:
* Scan Multibranch pipeline Triggers 
  riodically if not otherwise run 
  Interval 
  1 minute 
* Cliquer sur "Save"
* De cette façon, Jenkins vérifie toutes les minutes s'il y a des changements sur le repository et dans ce cas la pipeline sera exécuté.
* **ATTENTION: dans cette phase c’est normal que la pipeline échoue, car le build de l’application fait par Gradle donne une erreur. Cela est dû au fait que le service de caching Hazel n’a pas encore mis en place:**
* 

## Docker registry

En sein de la phase correspondant au stage “Docker push”, l’image Docker de l’application cible de cette pipeline est déposé sur le Docker registry, qu’est une application serveur permettant de stocker des images (push) et de les récupérer (pull). C'est similaire à Docker Hub, mais plus facilement accessible.

Pour cette plateforme nous utilisons un conteneur Docker qui lance l’image du Docker registry dans le localhost.

Pour démarrer cet conteneur, utiliser cet commande:

docker run -d -p 5000:5000 --restart=always --name registry registry:2

Le conteneur “registry” est démarré sur notre machine. L’option “restart=always” assure qu’il soit tout le temps fonctionnante pour permettre la pipeline d’être accompli.