2.19.4

## Repo github

https://github.com/LearningCitizen/Jenkins-ece.git

## Plugins suggéré

* ant
* antisamy-markup-formatter
* build-timeout
* cloudbees-folder
* credentials-binding
* email-ext
* git
* github-organization-folder
* gradle
* ldap
* matrix-auth
* mailer
* pam-auth
* pipeline-stage-view
* ssh-slaves
* subversion
* timestamper
* workflow-aggregator
* ws-cleanup

## Key CI/CD/Jenkins concepts

### Continuous Delivery/Continuous Integration Concepts

**Continuous integration** : software development practice where members of a team integrate their work frequently. Then there is an automated build (including test) in an integration machine (in dev environment). (git push + automatic build). A build is self-testing if a test fail makes it fail.

**Continuous delivery** : continuous integration + the possibility of easily deploy any version to any environment (especially in a production-like env). All the steps of the delivery process should be automatated (deployment pipeline).

**Continuous deployment** : continuous delivery + every change goes throught the pipeline and automatically gets put into production.

**Differences between CI and CD** : Continuous delivery requires continuous integration. Continuous integration + the possibility of easily deploy any version to any environment (especially in a production-like env).

**Stages of CI and CD :   
CI :** integrate code with a VCS + automated the build (with self-testing code, unit test) + automated tests on the executable   
**CD :** CI + all the steps of the deployment are automated (roll back should be easy)

**Continuous delivery vs continuous deployment** : In continuous deployment every changes are automatically puts in production. Whereas in continuous delivery, you can do frequent deployment but can choose to not do it. Continous deployment requires continuous delivery.

### Jobs

**What are jobs (project) in Jenkins?** : The jobs are a particular steps in the build process or in the deployment process. It can be a way of compiling, testing, packaging and deploying the project, but also measuring code quality/code coverage, generate documentation.

Job is a deprecated term. A project is a user-configured description of work which Jenkins should perfom, such as building a piece of software.

**Types of jobs**

* Freestyle software project: general-purpose build jobs. Jenkins will build your project, combining any scm with any build system. It can be used for something other than software build.
* Pipeline : Jenkins Pipeline is a combination of plugins that support the integration and implementation of continuous delivery pipelines using Jenkins. running activities
* Multiconfiguration project : job with many different configurations. Can be useful to test the application in many different environments. It runs the job with all the different. combinations of axis parameterers.
* Gitub Organization : Scans a Github organization for all matching some defined markers.

Multibranch pipeline : Create sets of pipelines according to detected branches in one scm repository.

* (Maven project : build a maven project taking advantage of POM files.)
* (External job : this type of job record the execution of a process run outside of Jenkins, even on a remote Jenkins.)

**Scope of jobs** : you can have jobs with same name in different folders.

### Builds

**What are builds in Jenkins ?**

Result of a single execution of a Project/job.

**What are build steps, triggers, artifacts, and repositories ?**

A build step is a single task during a build.

A build trigger is a criteria for triggering a new pipeline run or Build (time, scm polling, etc..).

An artifact is an immutable file generated during a build or pipeline run which is **archived** onto the Jenkins master for later retrieval by users.

A git repository is a data structure (folder .git in a project) used by git to store the set of files as well as history of changes made to those.

An artifact repository is a source for artifacts needed for build ans a arget to deploy artifacts generated in the build process.

**Build tools configuration**

In manage Jenkins -> systems configuration -> set the location of the tools used in build as ant, jdk, maven, node js python, etc..

### Source

**What are source code management systems and how are they used ?**

It is a software wich allows user to keep track of the changes in a project and enable them to collaborate.

**Cloud-based SCMs**

Cloud-based or distributed scms are a type of scms in which all the developer’s computer contains the complete code base and the full history.

Ex: Github.

**Jenkins changelogs**

It is a functionality of Jenkins which shows changes tracked by the scm between each revision.

Lists commits since last build.

**Incremental updates v clean check out**

Incremental update -> faster

Clean check out -> guarantees no extra or changed local files

**Checking in code**

Means when code is uploaded in the main branch repository, to review the code before update the project version. At least daily with CI.

**Infrastructure-as-Code**

Branch and Merge Strategies

### Testing

**◦ Benefits of testing with Jenkins**

**◦ Define unit test, smoke test, acceptance test, automated verification/functional tests**

### Notifications

**◦ Types of notifications in Jenkins**

**◦ Importance of notifications**

### Distributed Builds

**◦ What are distributed builds?**

**◦ Functions of masters and agents**

### Plugins

**◦ What are plugins?**

**◦ What is the plugin manager?**

### Jenkins Rest API

**◦ How to interact with it**

**◦ Why use it?**

### Security

**◦ Authentication versus authorization**

**◦ Matrix security**

**◦ Definition of auditing, credentials, and other key security concepts**

### Fingerprints

**◦ What are fingerprints?**

**◦ How do fingerprints work?**

### Artifacts

**◦ How to use artifacts in Jenkins**

**◦ Storing artifacts**

### Using 3rd party tools

**◦ How to use 3rd party tools**

**• Installation Wizard**

**◦ What is the Jenkins Installation Wizard?**

**◦ How to use the Wizard?**

**◦ Which configurations are covered by the Installation Wizard?**

Last link: https://jenkins.io

## Jenkins Usage

## Building Continuous Delivery (CD) Pipelines

## CD-as-code best practices