# JENKINS CHEAT SHEET

## WHAT IS JENKINS AND WHY DO WE NEED A CONTINUOUS INTEGRATION SYSTEM?

**Jenkins** is an open source automation tool written in Java with plugins built for Continuous Integration purposes.

Continuous integration is a practice of automating the integration of code changes from multiple developers onto a single software project. It allows the Developers to frequently merge code changes into a central repository where builds test and run.

## JENKINS JOB TYPES

JENKINS PIPELINES
Runs the entire software development workflow as code, Instead of creating several jobs for each stage of software development, you can now run the entire workflow as one code.
Learn more >

FREESTYLE
Provides maximum flexibility, with a general-purpose build job. It can be used for any type of project.
Learn more >

MULTICONFIGURATION
Provides the ability to run the same build job on different environments, usually we will use it for testing an application on different environments.
Learn more >

FOLDER
Creates a container that stores nested items in it. It is useful for grouping things together.
Learn more >

MAVEN PROJECT
Builds a Maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
Learn more >

GITHUB ORGANIZATION
Scans a GitHub organization (or user account) for all repositories matching some defined markers.
Learn more >

## HOW TO INSTALL JENKINS

Jenkins is distributed as WAR files, native packages, installers, and Docker images. There are different installation types, depending on your OS system.

Use the following commands to install Jenkins with Docker:

```
$ docker pull jenkins/jenkins

$ docker run -p 8080:8080 -p 50000:50000
--name jenkins jenkins/jenkins:lts
```

Once installed, you'll be able to access the Jenkins server on port 8080 and create your first build.

Naming the container "--name jenkins" allows us to easily stop / restart it with the following command:

```
$ docker stop/restart command
```

Artifactory places no limitations and lets you set up any number of Docker registries, through the use of local, remote and virtual Docker repositories.

## JENKINS PIPELINE CONCEPTS

When using the Jenkins Pipeline you can run the entire workflow as a single code, instead of running multiple jobs for different stages of software development.

The code is stored in the Jenkins file and it is written using the Groovy DLS programming language.
It is based on the following two syntaxes: Scripted Pipeline and Declarative Pipeline.

In the Scripted Pipeline, the code is written using the Jenkins UI instance and enclosed within the node block such as the following.

```
node {
scripted pipeline code
}
```

For the declarative pipeline, the code is enclosed within the pipeline block and it is written locally within a file.

```
Pipeline {
declarative pipeline code
}
```

## BASIC JENKINS PIPELINE EXAMPLE

```
node {
state ('SCM checkout') {
//checkout from your SCM (Source control
Management)
//for ex: Git Checkout
}
State('Build') {
//compile code
//install dependencies
//perform Unit Test, Integration Test
}
state ('Test') {
//resolve test server dependencies
//perform UAT
}
Stage ('deploy') {
//deploy code to prod server
//solve dependency issues
}
```

## JENKINS PLUGINS

The Jenkins capabilities can easily be extended by installing the Jenkins plugins inside the Jenkins Dashboard --> Manage jenkins --> manage plugins.

Top 5 Jenkins plugins for DevOps in 2021:

1. Job DSL
2. Job Generator Plugin
3. Performance Plugin
4. GitHub/GitLab Pull Request Builder
5. JIRA Plugin

## JENKINS ARTIFACTORY PLUGIN

The Jenkins Artifactory plugin is an open source project that allows your build jobs to deploy artifacts and resolve dependencies to and from JFrog Artifactory. The build info is created and linked to the build job.

The plugin includes a vast collection of features, including a rich pipeline API library and release management for Maven and Gradle builds with Staging and Promotion.

Refer to this Github repository for some Artifactory & Jenkins DSL and pipelines examples.