



Jenkins Assignment

September 20, 2016

Table of Contents

1. Objective.....	2
2. Software Required for the assignment.....	2
3. Deliverables for the assignment.....	2
4. Assignment	3

1. Objective

The objective of this assignment is to implement a Continuous Integration (CI) process and mimic or implement Continuous Delivery (CD) process (Bonus credit) using Jenkins pipeline job (aka Workflow job). In this assignment, we will assess an individual's ability to

- a. Create a Jenkins pipeline using the best practices
- b. Knowledge of CI/CD process
- c. Coding skills in groovy DSL
- d. Ability to implement tasks under a strict deadline

2. Software Required for the assignment

The following software may be required for implementing this assignment. Unless specified as mandatory one may prefer to use alternative tools based on the comfort level and experience. However, if alternative tools are used, please mention that in a README document, when turning in the assignment.

- a. MANDATORY - Jenkins 2.0+ (Pipeline Plugin + any other plugin that is required for the pipeline job and assignment. A list of plugins that are associated with the pipeline plugin are available at <https://jenkins.io/doc/pipeline/steps/>)
- b. GIT (Any GIT tool that you are familiar. Pick a tool that has web-hooks feature available. GitHub and GitLab support web-hooks)
- c. Groovy SDK (If you prefer coding on an editor, such as IntelliJ or Eclipse)
- d. AWS EC2 instance (if you are eligible for a free-tier, pick one from the free-tier ex: - t2-micro), with tomcat installed on this instance. This may be required only if you are planning to demonstrate CD process in your assignment
- e. Apache Maven 3.x

3. Deliverables for the assignment

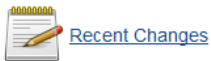
The following deliverables are expected as part of this assignment

- a. Pipeline groovy code
- b. README.md file that explains about the pipeline and how the CI process and CD (if implemented) process is implemented.
- c. The assumptions, if any, that were made to arrive at this solution

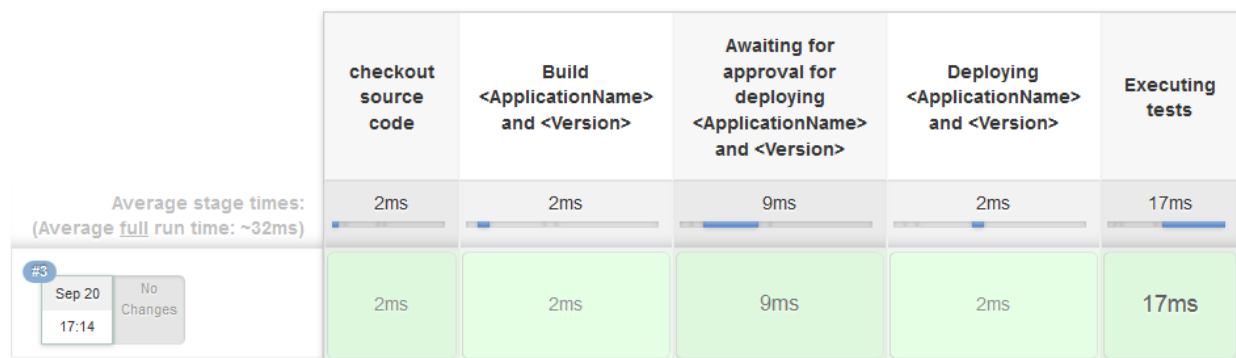
4. Assignment

As mentioned above, the crux of this assignment is to showcase your CI/CD implementation using Jenkins pipeline job. The pipeline job should consist of 5 stages as shown in the image below

Pipeline jenkins-assignment-pipeline



Stage View



Permalinks

- [Last build \(#2\), 13 min ago](#)
- [Last stable build \(#2\), 13 min ago](#)
- [Last successful build \(#2\), 13 min ago](#)
- [Last completed build \(#2\), 13 min ago](#)

1. Stage 1 – In this stage, you should check in the attached code (sent along with the email) into a GIT repository of your choice. Due to security reasons, the file is named as JenkinsAssignment.txt. Once you download this file, rename the file as .zip and extract the code. Once the source code is checked into the git repository, checkout the code from the pipeline. The stage must be named as “Checkout source code” as shown in the pipeline stage view above
2. Stage 2 – The application that is shared is a Java maven application, you must create a stage to build the application. Before the application is build, you must read the pom.xml and extract the value in <artifactId> and <version> and pass it to the stage name. The stage must be named “Build <ApplicationName> and <Version>”. In this the <ApplicationName> value must be replaced with the artifactId and <Version> must be replaced with <version> from the pom.xml. Save the extracted values, so these can be repurposed across the pipeline. You must also figure out the command to build the maven application
3. Stage 3 – Once the application is built and is successful, you must create an input step wrapped around a timeout seeking an approval to deploy the application. The timeout should be set to 5 minutes (in

the interest of this assignment), inside the timeout function, create an input step that requests for approval. The input message must be “Awaiting approval for deploying <ApplicationName> and <Version>. Are you ready to deploy? The <ApplicationName> and <Version> must be replaced with the values that were extracted in the pom.xml in Stage 2. If you approve the request, only then the pipeline must go to the next stage. Otherwise, the pipeline must be aborted.

4. Stage 4 – This is the stage you have an option for Bonus credit. The artifact that was built in Stage 2, is a .war file, which can be deployed to an EC2 instance that has tomcat installed. If you are unable to deploy the application to a real environment, then you can skip this stage. However, just print the stage name and don’t have any steps under this stage. The stage name must be “Deploying <ApplicationName> and <Version>”. If you deploy this to an EC2 instance, then implement the next stage
5. Stage 5 – Once the application is deployed to an instance, run a simple curl command and echo the output of the curl command. The curl command must be executed against <http://localhost:<port#>/JenkinsAssignment/jenkinsapp>. The stage name must be “Executing tests”