Jenkins lecture session by Guarav:

What is Jenkins: it is a software application that allows continuous integration and continuous delivery of projects.

Jenkins is an open source automation server which enables developers around the world to reliably build, test, and deploy their software.

The architecture looks like:

1. Developer pushes their code to the repository, central.

2. The Jenkins update its change in its repository and start to build the preliminary project called snapshot version. This creates the war or jar file and save it in its internal repository.

3. The Jenkins initiates the unit test every time after the initial build

4. Last the maven will come into picture and get the final build but still save it in the Jenkins repo

5. The Conan will then get the final artiractId stated at the xxx-dd folder and find that war or ear folder from the Jenkins repo. It then build and save it at the central server repo.

Developers send their code to local and bitbucket. when changes is made at the repo, Jenkins—- produce war or jar files and it store it in its repository file.

Conan job or automated job will deploy the final project or product war file on the server what repo to be deployed or built is depend on the pom.xml in the repo name with the xxx-dd name at the end.

10/18/2016

Discussion with Gaurav

It was really great and interesting discussion and learned a lot.

I will now explain what I have learned and what we discussed:-

1. Jenkins and its architectures and its application
   1. Developer pushes their code to the repository,
   2. Jenkins started to build the source, into a snapshot version and save it in the Jenkins repository.
   3. you will get a link where you can get the built source code and you can do the testing afterwards
   4. During the build and testing there is always a notifications regarding to the progress made.
   5. once the test is passed the Jenkins allows for the final build by the Maven and can
   6. Then the Conan job will build the final release and save it at the server.
   7. the final release will come out from the Maven and still get saved in the Jenkins repository
   8. The final package is saved or build in war or ear form
   9. Last Conan will build the final and deploy it to the server.
2. How does the Jenkins read or start buildings or building projects
   1. The first file Jenkins will look at is the POM.xml file (the parent)
   2. The Jenkins will see the module blocks in the POM.xml and will start building accordingly
   3. Within each child POM.xml files of the specific source src, it will get the information how it will be built. It will build and get all the dependencies.
   4. There is a last module that always have to exist with -dd to the end of the project name. This one has only a POM.xml file and it will tell which one should be the ear must be built from. It will find it under the artifact block.
3. You can always find the built link so that you can save, make the test and save the final built ear file in server.
4. What does the parent POM.xml file contain? The major blocks are
   1. The module blocks which contains what projects or sources are to be built
   2. The Connection which will let the Jenkins what SSH is the source on in the repository (Github or Bitbucket). They both should have same SSH during the configurations. In the Jenkins and Bitbucket configurations
   3. The repositories for both the snapshot and the final version build location to be stored
   4. Some dependencies with a lot of other tags
5. How to give access other to the repository
6. Difference between master and branch in repositories
7. how to get the SSH from the repository
8. There is nexus, Conan

New reading from Jenkins:

Jenkins can stand by itself as it has server application embedded in it called jetty whinstone

1. Install it in mac using terminal and type this text java -jar Jenkin .war
2. when it is installing you will get a password which helps you to start or unlock your Jenkins in your local browser. you can access it in the browser using http://localhost:8080
3. You will be asked to install plugins and you can do that using either by selecting yourself or follow the recommendations. The Jenkins directory could be hidden sometimes and you should be able to make it visible in the terminal using some line of codes.

if we have tomcat server installed and the default port will be taken by it and we have to assign another port separately for Jenkins during installations.

operating your work on Jenkins, using command line or Terminal is good choose as it is easier, after.

I was looking at manipulating the github well with pulling and pushing codes to the branches, testing, checking for conflicting and commit and push to masters for code reviews … etc

conan only reads the dd file to check the artifacts needs to be deployed

10/21/2016

Discussion with Gaurav Kumar today:

1. About scripts/Codes more of the DB
   1. Incremental
      1. It can only be run and build by the Conan. If we try to run it twice, Conan will fail.
      2. If we want to make change and run it again we can do it manually by removing its history from the DB and Conan can run safely and successfully.
   2. PostProcessing
      1. contains configuring files
   3. Repeatable
      1. Can run more than many times and Conan can run successfully
2. One Conan job 🡪 one script 🡪 one repo
3. During our scheduling our build or release in Conan we select Dry run only for scripts
4. Scripts 🡪 DB
5. If we get some defects in our previous release and if we already started in new spring task how do we fix the defects
   1. Branch repo comes into picture. Since we already started working with next spring and the master repo is now full of both the previous and current spring, it is hard to make change in the previous master spring. We go back to the branches and make all the necessary changes. We then merge it in to the master.
6. How do we request for ticket for the final release to the final maven

The child pom.xml contain the artifactId to be deployed and the version how it should be build. Jar or war

Last module in the dd file contains the pom which tells the final content to be deployed finally and the version with its type.

10/26/2016

Inside the pom.xml:

There is

groupdId🡪 location of the build in the Jenkins

artifactId🡪 repository name or folder name

version 🡪 kind id unique to everyone

10/31/2016

To create a project and build it all the way toward deployment or building a projects in Jenkins

* + - 1. We create a project in a given folder.
      2. We create a project or repo in our git
      3. We set up all the setting like ssh key and enable and also give project permission so that Jenkins can access it freely
      4. Create a new job in Jenkins and setup will all the information about our remote GitHub information like GitHub URL and branch info.
      5. You need to setup a lot of the info like number of builds of same project to save and periodically of building the project etc
      6. You can build the project and follow the building in the console log page
      7. Once it gives you the status of the project you can follow up accordingly
         1. If it is failed to build you will need to see what happened and solve the issues accordingly
         2. If it is succeeded, you can carry on.
      8. Error could be from the pom.xml set up or permission or something else.