**JENKINS**

**Overview**

* **It is an open source automation server, which automates all sorts of work like build, test and deploy.**
* **Built in java.**
* **It implements continuous integration and continuous delivery concepts.**
* **Basically it is a platform for SDLC (software development life cycle).**
* **It is light weight, does not use too much of resources.**

**Installation:**

* **It can be installed on any standalone machine with JRE installed**
* **It can be installed through native system packages and Docker.**

**Basic Terms**

**Continuous Integration: It is a software development practice, where developers regularly merge their code to a shared repository after which the automated builds and tests are run.**

**Continuous Delivery: It is a software development approach where the software/application is automatically built and prepared for a release to production.**

**Enables organization to deliver new features to the customers as fast as possible.**

**Pipeline: Pipeline breaks the project into different stages. Each stage is aimed at verifying new features.**

**There is no standard pipeline. But a typical pipeline includes**

* **Build automation and continuous integration**
* **Test automation**
* **Deploy automation.**

**Build automation and Continuous Integration**

**This pipeline starts by building the binaries that create the deliverables. The new features are integrated to the source repository on CI basis, from where the project is built and unit tested. This stage is very helpful for the developer to know the health of the application.**

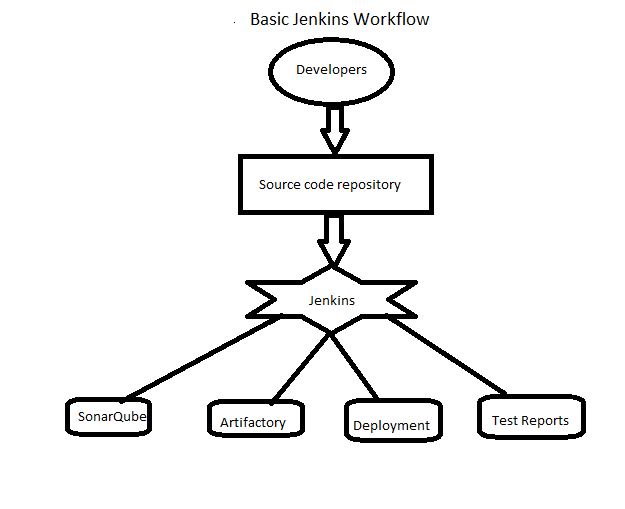
**Test automation**

**Here the new versions of the application are tested to ensure whether it has met all the desired qualities. All the qualities include functionality, security, performance and compliance. It involves automated and manual activities.**

**Deploy automation**

**The deployment is automated, which ensures the delivery of application to the users within minutes.**

**Basic Workflow**

****

1. **Get code from SCM (like Git).**
2. **Executes build/compile.**
3. **Executes unit tests and gets the code coverage.**
4. **Executes the automated functional tests.**
5. **Executes any other tests if necessary.**
6. **Executes static code analysis (sonarqube).**
7. **Deploy artifact/build to a build repo (nexus/Jfrog).**
8. **Deliver the artifact/build to a test machine.**

**BUILD+PACKAGE+SONAR+JFROG**

**Installing and Configuring Jenkins**

* **Jenkins requires appropriate JDK to be installed.**

**Java Link:**

**http://download.oracle.com/otnpub/java/jdk/8u131b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-windows-x64.exe?AuthParam=1498746397\_efd1410fbaa7848022b25a667102d3e4**

* **It can be installed as standalone application or running on a webserver (like tomcat).**

**Standalone Application**

**Windows:**

* **Download Jenkins.War file.**

**(http://ftp-nyc.osuosl.org/pub/jenkins/war-stable/2.60.1/jenkins.war)**

* **Place it in C:/programfiles.**
* **Open Git bash where the Jenkins.War file is present.**
* **Then run the command java –jar jenkins.war**
* **By default Jenkins run on port 8080**
* **To give a specific port number then run java –jar jenkins.war –httpPort=<port number>**
* **Browse localhost:8080 (port number)**

**Ubuntu:**

* **Install Jenkins by following the steps in the link**

[**https://vexxhost.com/resources/tutorials/how-to-install-configure-and-use-jenkins-on-ubuntu-14-04/**](https://vexxhost.com/resources/tutorials/how-to-install-configure-and-use-jenkins-on-ubuntu-14-04/)

* **Browse ipaddress:8080 (port number)**
* **Initial admin password is present in /var/lib/jenkins/secrets/initialAdminPassword**

**Cat /var/lib/jenkins/secrets/initialAdminPassword**

* **Unlock Jenkins using this password.**
* **Install suggested plugins.**
* **Create user.**

**Configuring Jenkins**

**Plugins:**

* **They add functionality to Jenkins.**
* **Git plugin: manage Git repo**
* **Maven: handles build and package of java applications**
* **Junit: handles unit test reports.**

**Procedure to install plugin**

* **Open jenkins dashboard**
* **Manage jenkins- manage plugins- click on available tab- install**
* **If the plugin is not present in the available tab, then download it from the respective site and upload.**

**Jenkins Dashboard**

* **New Item: This is where new projects, folders and pipelines are added**
* **People: This is where we can see a list of users and their latest activities.**
* **Build History: Display overview of build history for all projects in graphical form.**
* **Manage Jenkins : Where Jenkins is managed**
* **My Views: For configuring custom views for projects for the logged in user.**
* **Credentials: Lists credentials that have been configured for Jenkins.**
* **Build Queue: Jobs waiting for an executor are listed here.**
* **Build Executor Status: The status of projects associated with Jenkins executors.**
  1. **An executor runs projects dictated by Jenkins.**
  2. **They can run in parallel.**
  3. **The default number of executors on the master is 2.**
* **Build History: Shows the history of builds.**

**Blue: Build is stable.**

**Red: Build is broken.**

**Jenkins on tomcat server**

**Why?**

**Jenkins standalone container jetty/winstone (standalone)**

**Running Jenkins on tomcat servlet container advantage is that we can start all applications on a single server.**

**We can deploy jenkins on**

* **Glassfish**
* **Tomact**
* **Jboss**
* **IBM Websphere**
* **Jetty**
* **Jonas**
* **WebLogic**
* **Apache Geronimo 3.0**
* **Liberty profile**

**PreReq: Tomcat 5 or above**

**Java 7 or above**

**Step1: Download tomcat**

**Step2: unzip and place tomcat folder at any location on your system.**

**Step3: copy and place jenkins.war file inside tomcat webapps folder.**

**Step4: Goto cmd (windows) ---- Goto tomcat bin directory----- make all file executable: chmod +x \*.sh**

**Step5: start tomcat server --- starup.sh**

**Shutdown-------- shutdown.sh**

**To ensure whether jenkins is running--- tomcat runs on port 8080**

**Browse localhost: 8080/jenkins**

**How to change home directory in Jenkins**

* **By default it will be the user location of the system (.jenkins folder)**

**Why to change home directory?**

* **As it is installed in user profile we have to move to a location where we have more user space**
* **Requirement of the project**

**Step1: Check your current home directory**

**Step2: start Jenkins. Create a new folder which will be a new home directory**

**Step3: Manage Jenkins --- configure system --- we can see home directory**

**Step4: copy all from old directory to new directory**

**Step5: change environment variable, set it to new home directory (JENKINS\_HOME)**

**Step6: restart Jenkins**

**How to use CLI for Jenkins**

**Why?**

**Easy**

**Fast**

**Consumes less memory than UI**

**Step1: Start Jenkins**

**Goto manage Jenkins – configure global security – check enable security -- apply**

**Goto --- localhost:8080/cli/**

**Download Jenkins-cli jar and place at any location**

**To test— go where jenkins cli jat is present**