**Build Part**

Whenever Developer commits the code to mainline trunk or master in case of git and push the code to the github, Jenkins will check out the code using poll scm and it will kick off the maven scripts and maven will do the compile, test, package, install and deploy to the artifactory here we use nexus in my current project. Here I configured nexus with maven whenever we do mvn deploy, Artifacts are deployed into the nexus repository. There are again snapshot and release versions, For Continuous integration part we keep on using snapshot version, Whenever developer thinks that the development is done and says like we are good to go for the release. Then there is another build which will be kicked off called release build where it will checkout the latest code builds the code and deploys the artifacts to the nexus release repository. Till here build part is over.

Of course we will run code quality checks unit test cases, unit tests, integration test and if every thing is good we are going to publish into the nexus repository.

**Deployment Part**

Coming to deployment we need to create different environments like QA, UAT, PROD

For the deployment part also we will have a job called deploy job. Deploy job will have some parameters like environment, Component, Branch and version. Depending on the environment Jenkins will kick of the CFT templates from GIT repository and CFT will spin off the instances in AWS and

Integrated chef with CFT where chef will take care of provisioning of nodes where we kind of install and configure different packages.

Downloads Chef-Client package,

installs chef-client,

also download required keys like user.pem and validator.pem and configuration files for authenticating node to the chef server into /etc/chef/ directory

and bootstrap the node to register node into the chef server, (Then after we Assign the role consisting of the runlists which are having required cookbooks to configure the node for particular cookbook) and runs the chef-client on the node and deploys the artifact into the newly created environment. For example if I give the parameter as QA, QA environment is created and Deploy the artifacts into QA environment. Now we will give the QA env for the testing purposes, If the testing is done and every thing is good we will promote the same code into different other environments like UAT, Stage and production.

We have deployment cookbook to pull the artifacts from artifactory and deploy to

Weblogic

**Continuous Delivery**

Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time.

You achieve continuous delivery by continuously integrating the software done by the development team, building executables, and running automated tests on those executables to detect problems. Furthermore you push the executables into increasingly production-like environments to ensure the software will work in production.

The principal benefits of continuous delivery are:

* **Reduced Deployment Risk:** since you are deploying smaller changes, there's less to go wrong and it's easier to fix should a problem appear.
* **Believable Progress:** many folks track progress by tracking work done. If "done" means "developers declare it to be done" that's much less believable than if it's deployed into a production (or production-like) environment.
* **User Feedback:** the biggest risk to any software effort is that you end up building something that isn't useful. The earlier and more frequently you get working software in front of real users, the quicker you get feedback to find out how valuable it really is (particularly if you use [ObservedRequirements](https://martinfowler.com/bliki/ObservedRequirement.html)).