**CI Implementation:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

We got 3 different type of requests from Application team

1. Pull Requests
2. Push Requests
3. Release Branches

As per application team, it must be something like INT, DEV and PROD environments.

So far, we are working to implementing CI model in INT environment.

1. Pull requests:
2. Application team will try to update any branch in one repository and wants to execute the Jenkins Job (which will contain the Tests to be run) on every pull request they create.
3. So Once initiated from Stash 🡪 to Jenkins, after executing the job in Jenkins successfully, the pull request in stash will update the status about the Build (like SUCCESS or FAIL)
4. Upon Success message in stash, the person who supposed to merge the code will review and Merge the code.
5. Upon Failure message in Stash, the developer supposed to update the code and create a new Pull request. And This cycle will go on.
6. PUSH/Merge requests:
   1. When ever Application team pushes the code to stash repository, a new Jenkins Job should be triggered.
   2. For this we are using Post Hooks and configured that in one repository. This Jenkins job should contain an option in the Build Triggers “Build when a change is pushed to BitBucket”.
   3. This option should be enabled, and the Branch should be hard coded in Source Code management. Because this is complete Test, so it has to pull the proper code to its Jenkins workspace, run some integration tests, Build a war file and Deploy that war file to Tomcat application server.
   4. This “post hook” is only the thing that triggering the Jenkins Job by polling its own repository for any new Push/Merges.
7. Release Branches:
   1. For every Iteration, application team will create a new Release Branch just to make sure it incorporates all the required changes in that release.
   2. So there should be some Jenkins Job, and the Branch name should be parameterized.
   3. Means when ever we trigger this Jenkins Job, it should take some input like Enter Branch name. At this point of time, application team will enter the Branch name and will execute the build.
   4. For this we have used “GIT PARAMETER” Plugin.
   5. This still has to be tested.