**CI :** Continuous integration : When we write / update our code, we push it to GIT repository. As soon as we push it, Jenkins build and run it OR it executes some test cases on it. So once we commit our code , then the Jenkins will run automated test cases for it to make sure it is working as expected. This is called CI.

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early. By integrating regularly, you can detect errors quickly, and locate them more easily.

**CD  :**

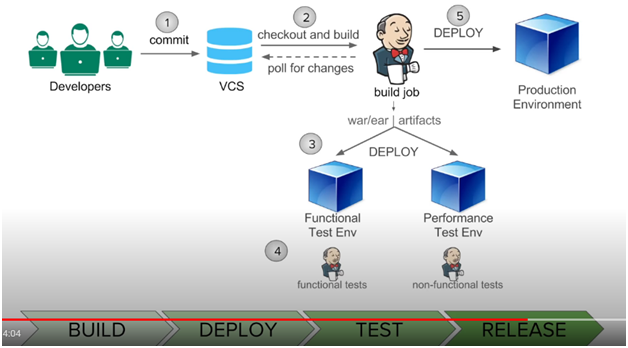
1.       Continuous delivery

2.       Continuous deployment

**Continuous delivery** :  Deploying your project, that is ready till today, on a mock server. It is a production like server where the current project will run. It gives us the confidence that the things are working. So, we are keeping it ready for the deployment on production server.

**Continuous deployment :** Code goes to production server.

Every companies requirements are different so every company chooses either only CI or CI and continuous delivery or some chooses CI CD.



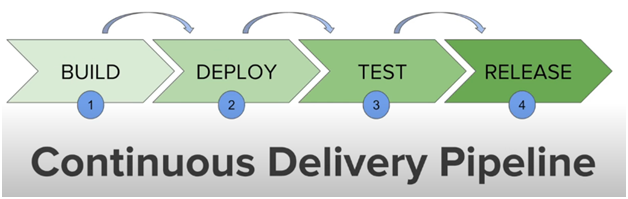
 NOTE : As a QA we just perform CI because we never deliver our selenium project to client.

Once a developer push changes to GIT, it triggers a build in Jenkins and after finishing a build(after finishing a build, it generates .war OR .ear file. If it is a web application then it generates .war file and if it is Enterprise Application, then it generates .ear file in **workspace** of your build project), it executes some test cases on it (if it is mentioned in Jenkins job, under post build actions section).  In Jenkins, in Post Build Actions section, there is option (deploy WAR/EAR to a container- this option only available after installing ‘deploy to container plugin’ in Jenkins) so after successful build, it triggers the deployment job. It deploys the .war / .ear file on the server. Further after successful deployment, it runs a few tests on it and after success it automatically triggers the deployment to Production server. This is called Continuous integration and continuous deliver.

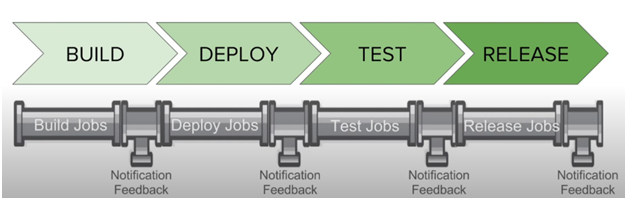
**EAR** (Enterprise Application aRchive) is a file format used by Java EE for packaging one or more modules into a single archive so that the deployment of the various modules onto an application server happens simultaneously

**WAR** (Web application ARchive ) JAR files are the files that have Java class files, associated metadata and resources aggregated into a single file to execute a Java application. It executes our project

**What is pipeline in Jenkins** :



Here every stage has some jobs associated with it and they are integrated with each other and all these jobs work in a sequence and we call it as a continuous delivery Pipeline.



How to setup delivery pipeline in Jenkins : [https://www.youtube.com/watch?v=ndLbn24OwQg&list=PLhW3qG5bs-L\_ZCOA4zNPSoGbnVQ-rp\_dG&index=15](https://eur02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DndLbn24OwQg%26list%3DPLhW3qG5bs-L_ZCOA4zNPSoGbnVQ-rp_dG%26index%3D15&data=04%7C01%7Cpratik.toke%40fisglobal.com%7Ce772e1d76f55448bc27e08d97993fc8d%7Ce3ff91d834c84b15a0b418910a6ac575%7C0%7C0%7C637674498787184184%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=EM9tGFazj5Sp7mgDSZx5%2BsKGRYx4FTFZkE8o7OgnvBQ%3D&reserved=0)

How to create a Jenkins pipeline  :

1.       Create new free style projects in Jenkins with name : sample build job, sample deploy job, sample test job

2.       Now, we have to chain our job. Open sample build job  > configure (in post build section, we can see option : build other projects, we can use this option. OR under build trigger section, there is an option : build after other projects are build. ) for the first job we will not add any step.

3.       Open sample deploy job.  Configure > build triggers : build after other projects are build > ‘sample build job  ‘  > select option : Trigger only if build is stable.  i.e. this project will build after sample build job  will build.

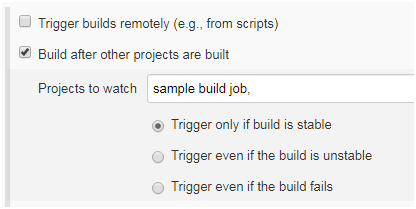
4.       Open sample test job.  Configure > build triggers : build after other projects are build >  ‘sample deploy job’ > select option : Trigger only if build is stable.  i.e. this project will build after sample deploy job.

Now, these 3 jobs are chained.

Having these jobs created in Jenkins :



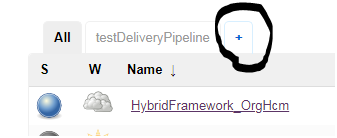
Below are the settings :



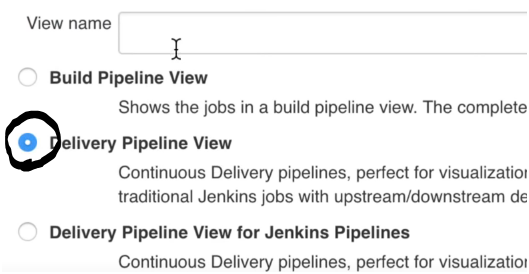
So we will have to run build job and it will automatically triggers the next jobs and it will be executed one after another.

We can install a plugin : delivery pipeline plugin and then we can see the following view of our pipeline :

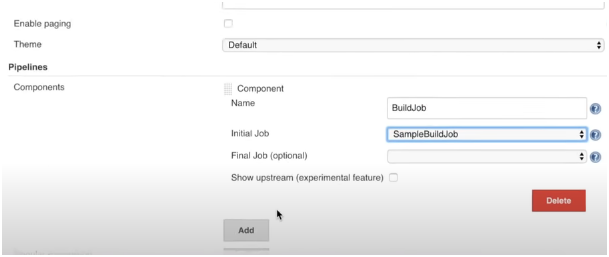
Click here on the + button :



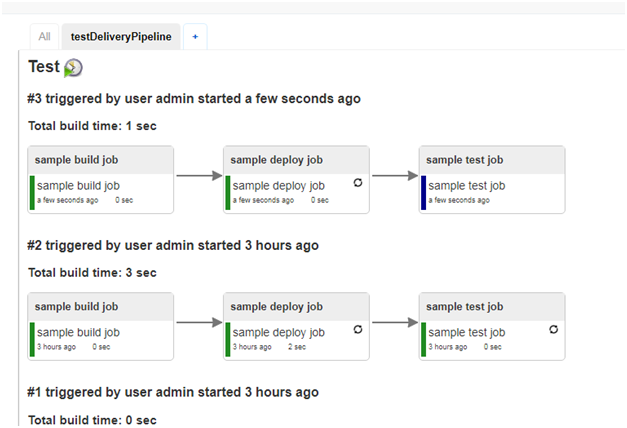
 Then click on the highlighted option :



 Fill the details and click on OK button



 Then we get the following view :



To learn more :

[https://www.youtube.com/watch?v=j5D8SLxn6YA&list=PLhW3qG5bs-L\_ZCOA4zNPSoGbnVQ-rp\_dG&index=12](https://eur02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3Dj5D8SLxn6YA%26list%3DPLhW3qG5bs-L_ZCOA4zNPSoGbnVQ-rp_dG%26index%3D12&data=04%7C01%7Cpratik.toke%40fisglobal.com%7Ce772e1d76f55448bc27e08d97993fc8d%7Ce3ff91d834c84b15a0b418910a6ac575%7C0%7C0%7C637674498787184184%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=%2BhuTKBIk9hVI2%2F%2FkKpa%2Fa05cwYB9CRSUN63cDh%2F%2BCzY%3D&reserved=0)