**Jenkins Notes**

**Jenkins:**

Jenkins isan open source automation tool written in Java programming language that allows continuous integration. Jenkins builds and tests our software projects which continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

**DevOps:**

The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to testing, deployment, and operations. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.

**Setting up Jenkins Job:**

**Step 1** − Go to the Jenkins dashboard and Click on New Item.

**Step 2** − In the next screen, enter the Item name. Choose the ‘Freestyle project option’.

**Step 3** − We need to specify the location of files which need to be built like git repository.

**Step 4** − Now go to the Build section and click on Add build step.

**Step 5** − Once the build is scheduled, it will run. The following Build history section shows that a build is in progress.

**Step 6** − Once the build is completed, a status of the build will show if the build was successful or not.

**Step 7** − Click on the Console Output link to see the details of the build.

**Pipeline:**

In Jenkins, a pipeline is a collection of events or jobs which are interlinked with one another in a sequence.

Setting up a Jenkins Pipeline:

**Step 1** – Click New Item on your Jenkins home page, enter a name for your (pipeline) job, select Pipeline, and click OK.

**Step 2** – Select the job that need to be run first.

**Step 3** – Select the one or multiple jobs which need to be run after the first job.

**Step4** – Click save and run the pipeline.

**Step 5** – Check the console for the output.

**Continuous Integration:**

Continuous integration is a software development process where developers integrate the new code they've written more frequently throughout the development cycle, adding it to the code base at least once a day.

**Continuous Deployment:**

 Continuous deployment is a strategy for software releases wherein any code commit that passes the automated testing phase is automatically released into the production environment, making changes that are visible to the software's users.