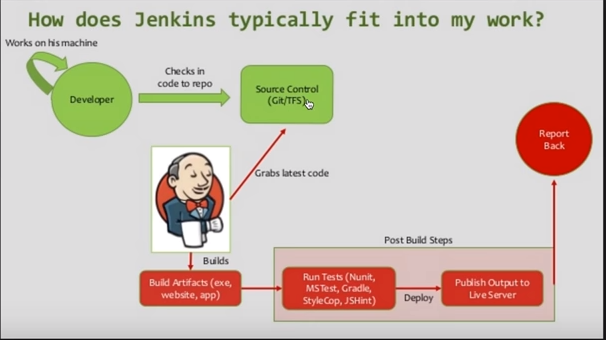
Jenkins is Java application since it is a java application it is platorm independent.

It is used for continous integration and Continous delivery.

When we work in team we have different developers who develop the code and keep on working on changing or enhancing the code and as the developer walks on his own machine and he changes the code he will check in the code into shared repository which can be Git or bit bucker or TFS . Now what happens lets suppose through the entire day there are many developers who are changing the code and checking in the code in the repository at the end of the day when you have a build lets suppose there is some bug introduced in any of the code and now the build failed. Now it would be very difficult for us to identify what exact code failed the build and at what point was this bug was introduced in the code and then you have to go back and check in you know all the code that was created that day and there will be a lot of confusion. So here come jenkins do is as soon as the developer commits the code in the shared repository Jenkins will take that latest code and trigger a build and the build notification will be sent out so in case there is problem in the build or there is an error you will be get and notified as soon the build trigger builds and completes. So we can check that if there is any issue due to any commits we can reverses that and we will not waste a lot of time in finding out what code caused the break . The other thing is let’s suppose the build is succesfull we can also integrate our unit test or acceptance test or performance test along with build as a post build actions in jenkins and it will be automated as soon as the build gets deployed jenkins will trigger some test cases or some testing which will be automated and it will send out the report back to us.Lets suppose the build was successful and there was no exceptions but due to changes in the code there was something which broke in the application so we will come to know instantly that there is some break due to the code and we can troubleshoot it. You can see how efficient and powerful the entire system can be and this what is called as continuos integration we are taking the code and doing a build as soon as the code gets commit and then we are also triggering the test and we are sending back the reports. So as soon as there is any change in the code this entire cycle gets triggered and we come to know if there is any issue with the application so this is what the continous delivery is all about and this is how jenkins fit into the picture



Step 1 : Download Jenkins war file - [https://jenkins.io/](https://www.youtube.com/redirect?q=https%3A%2F%2Fjenkins.io%2F&v=89yWXXIOisk&event=video_description&redir_token=Dmr5fVrwyGBcrzDD6TV247TLaJJ8MTU0MTY0ODE1OEAxNTQxNTYxNzU4)

Step 2 : Place the war file into any location on your system

Step 3 : goto command prompt (windows) | terminal (mac) - goto folder where jenkins.war is - java -jar jenkins.war

Step 4 : goto browser - [http://localhost:8080](https://www.youtube.com/redirect?q=http%3A%2F%2Flocalhost%3A8080&v=89yWXXIOisk&event=video_description&redir_token=Dmr5fVrwyGBcrzDD6TV247TLaJJ8MTU0MTY0ODE1OEAxNTQxNTYxNzU4) (Jenkins window should show up)

Step 5 : install required plugins. Installed plugins will be available in the folder C:\Users\Narayana\.jenkins\plugins

Step 6 : get started with Jenkins

**How to setup jenkins on Tomcat?**