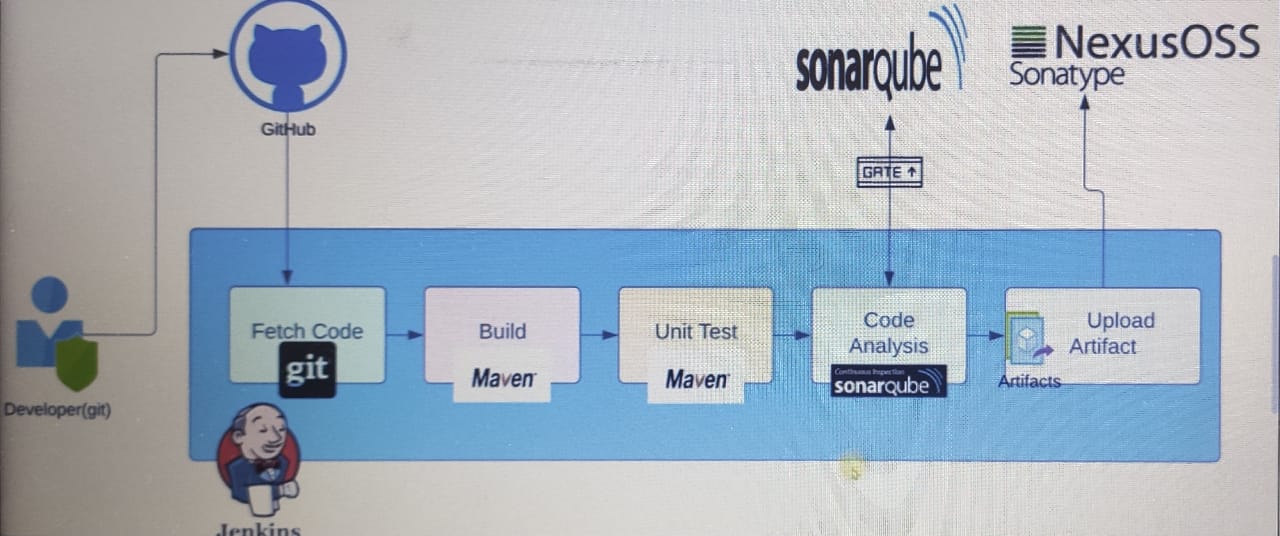
We will talk about the **continuous integration** pipeline that we are going to set up.



We will understand the flow. We are understanding the flow by using these tools

primarily.

Jenkins Then we have git, Maven, sonarqube, nexus.

So these are the tools that we will be taking as example.

But in your project, you may have different tools.

So focus on the flow.

And practice it with these tools that we are using.

And once you have hands on, you can do the same setup by using other tools.

So let's see the flow.

So we have the developer.

The developer's job is to write the code and developer will write the code, make the changes to the

code test it locally.

If they are good with the changes, they will push it to a centralized repository like GitHub.

So developers will have a tool which will integrate with GitHub repository and the code will be committed

to GitHub repository as soon as there is a code change.

Jenkins will detect a change and fetch the code by using git tool.

So Jenkins will have git tool and git plugin.

Which will help accomplish this task to fetch the code.

Whenever there is a change after that in the pipeline, the code will be bulid.

We will be using MAVEN to build the code because we have Java code and our code can be built with MAVEN

tool, but it could be any other source code and other build tools as well.

Once the build completes, it will generate artifacts.

Next we will conduct unit test again by using Maven.

Maven will have some unit testing framework that developer will use.

Unit testing will be part of your source code.

So being a DevOps, you don't need to do much here.

You just need to execute some steps that will run this test and generate reports mostly in xml format.

Once you have the reports ready, we will conduct another kind of test called us code analysis. Now unit

test checks whether the unit of the code works or not.

Code analysis checks if the code has any vulnerability.

Are you following the best practices?

Do you have any bug in the code?

And there are many other parameters on which code analysis will judge your code.

We will be using sonar cube scanner to scan the code.

Also, we will be using checks.

style

So there are many code analysis tools available in the market.

We are using sonar cube scanner and checkstyle to scan the code and this will generate reports in

a similar format.

These reports will be uploaded to Sonar qube server.

In sonarqube.

You can have proper graph, charts and you can see what are the bugs, vulnerabilities

and many other things in your code.

We can also set a quality gate and we can say, if my code does not follow these practices, then fail

the build.

And if it fails, the pipeline will stop.

If it passes.

We have then a verified copy of the artifact.

So we build the code, we test the code, we analyze the code.

And now you we can distribute the artifact to be deployed on the servers.

But before deploying it to the server, these artifacts will be versioned and will be uploaded to Nexus

sona type repository.

Now all this pipeline will be happening in Jenkins.

As I said previously, you may have a different tool in later like you may have GitLab circle CI Bamboo

there are many CI tools, but the process will be almost same.

Fetch the code, build the code, test it, analyze it and then publish the artifact.

Whatever CI tool you're using, you have to integrate it with other tools like GitHub, Sonar, Cube

Nexus or any other tool.

In the next video, we will see the steps that we will take to set up this entire continuous integration

pipeline.