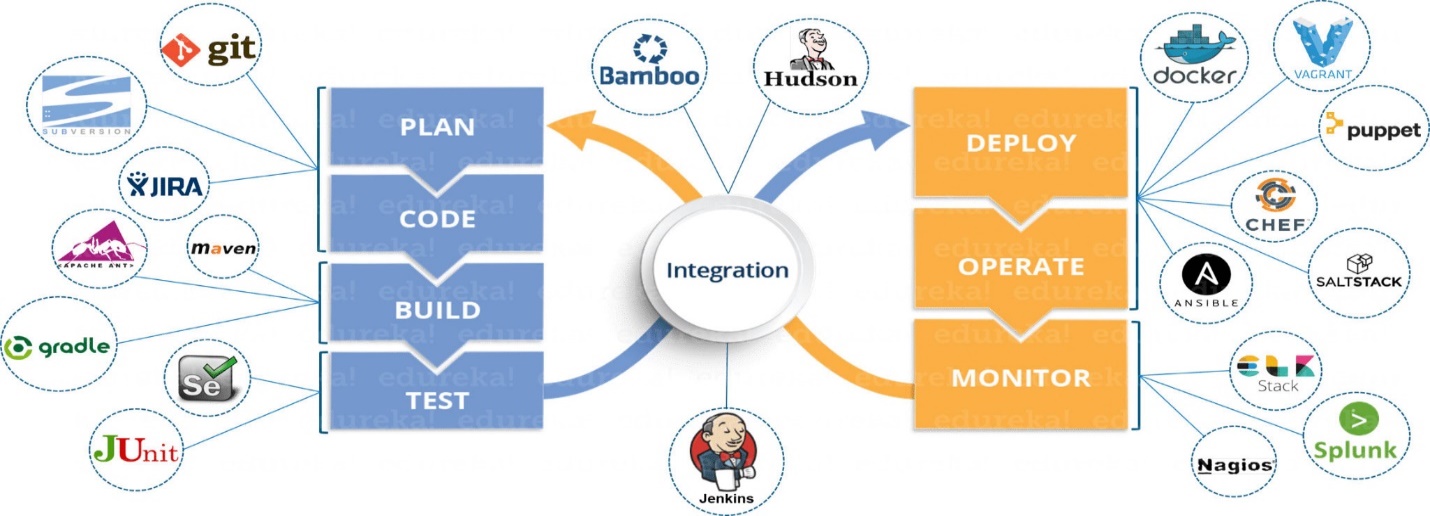
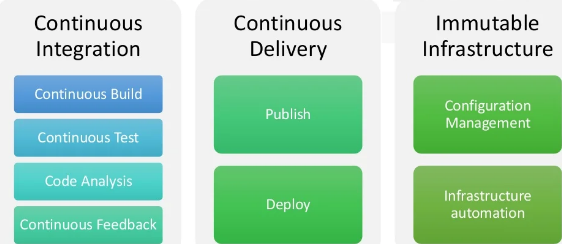
**DevOps: Jenkins integration with GitHub using Webhooks and Pipelines to ensure CI/CD (Continuous Integration and Deployment)**

**DevOps: Introduction**

Software developers are often blamed for the time it takes to develop and release new features. But there’s another group of people who deserve some responsibility in this process too: **Operations Professionals**, DevOps combines their skills with those from development, testing & QA teams so that apps can be delivered faster than ever before-without sacrificing quality or customer satisfaction. The goal of DevOps is to create an environment where **Developers** and **IT professionals** can work more closely with each other, as well as incorporate tools from across the industry. This will allow them deliver solutions faster than ever before, while also lowering costs through increased collaboration between teams responsible for deployment procedures. This is a process that continuously integrates and delivers value to end users. In simple words, it is a process that continuously integrates and delivers value to end users.



**DevOps: Component**



**DevOps: Benefits**

1. **Speed:**

This is a powerful collaboration of people, process and working product that enables continuous integration (CI) and delivery to end users.

1. **Rapid Delivery:**

By increasing the pace of releases, you can improve your product faster and build competitive advantage.

1. **Improved collaboration:**

DevOps is a revolutionary approach to software development, which involves developers and operations teams collaborating closely. The collaboration reduces inefficiencies while saving time for both parties involved.

1. **Promote agility:**

Businesses of all sizes can now enjoy the agility and scalability needed for success.

1. **Continuous delivery of software:**

Continuous delivery is the new way to ensure that your software always works. DevOps helps with Agile, and you can't have one without the other.

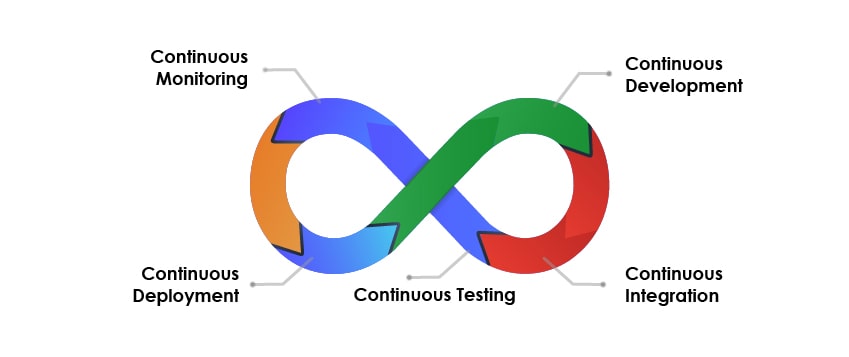
1. **Faster Bug Identification:**

With DevOps processes, fixes are identified and resolved at the earliest with faster identification of defects.

**DevOps: Lifecycle phases**

The lifecycle of DevOps works involves various stages such as:

**Continuous Development, Continuous Integration, Continuous Testing, and Continuous Deployment.**



1. **Continuous Integration (CI):**

The continuous integration stage is where the developer frequently commits changes to their source code and new features are continuously integrated with older ones. This way, developers can get rid of waiting for long periods between committing or updating specific parts in order reach a lot more quickly than if they were doing individual tasks individually instead - which helps them achieve this faster as well. Famous CI tools are **Jenkins, Maven, Selenium** etc.

1. **Continuous Development:**

The project plan is finalized and the application code begins to take shape. This phase focuses on strategy, planning for coding languages in order that no restrictions will be placed upon what can ultimately end up being used by developers when creating the applications. Famous CD tools are **Jenkins, GitLab, CircleCI, Bamboo** etc.

1. **Continuous Testing:**

This is the final stage of development, where software undergoes thorough testing to ensure no features or functionality are broken. To do this efficiently developers use Test automation tools which automate parts test cases for them so they don't have repeat work and can focus on new additions instead. Famous Continuous testing tools are **Selenium, Appium, TestSigma, Katalon Studio** etc.

1. **Continuous Deployment:**

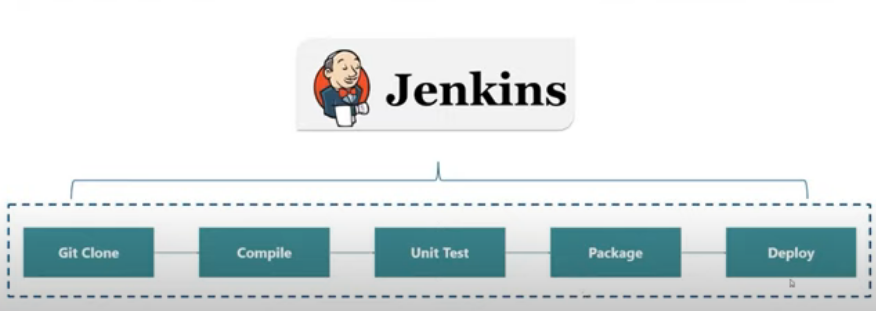
In this phase, Configuration management tools and/or containerization technologies are used in this stage to deploy apps. Configuration Management helps establish the app's functional requirements while also keeping track of performance data for each environment with confidence that it will be consistent across all environments, which saves time. Famous Continuous Deployment (CD) tools are **Docker, Kubernetes, Puppet, Ansible etc**.

1. **Continuous Monitoring:**

This is the final stage in which performance of software applications are monitored. Errors related to functionality, low memory and app availability can be detected with continuous monitoring tools that use both internal data sources as well external providers such infrastructure monitoring offerings from Cloud Custodians or other service companies who offer these types services. The tool that can be used in continuous monitoring is **Nagios** or many more.

**Continuous Integration & Continuous Delivery Pipelines with Jenkins**

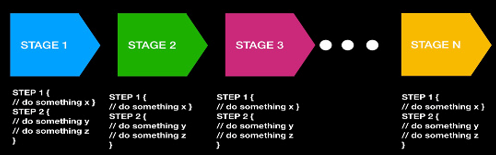
With the rise of CI/CD, more companies are turning to Jenkins for their DevOps needs. Jenkins is an open source [continuous integration](https://searchsoftwarequality.techtarget.com/definition/continuous-integration)/continuous delivery and deployment (CI/CD) automation software DevOps tool written in the [Java](https://www.theserverside.com/definition/Java) programming language. It is used to implement CI/CD workflows, called **Pipelines.**



**Visual Representation of Pipeline**

A pipeline is a sequence of events or jobs that can be executed. Below are the two-familiar concept:

1. **Stage:** A block that contains a series of steps. It can be named anything; it is used to visualize the pipeline process
2. **Step:** A task that says what to do, they are defined inside a stage block.



**How to Build Jenkins Pipeline**

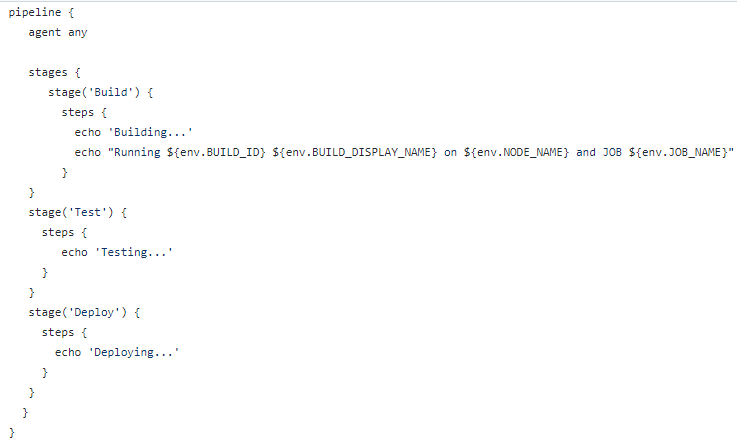
1. **Prerequisites**

Install Java Development kit (JDK) and add it to the environment path.

1. **Basic Steps**

* **Step 1**: Download Jenkins
* **Step 2:** Execute Jenkins as a Java binary (java -jar ./jenkins.war)
* **Step 3:** Create a new Jenkins job (by navigation to localhost:8080)
* **Step 4:** Create a pipeline job
* **Step 5:** Configure and execute a pipeline job through direct script

The format of the direct script is as follows:



**Step 6:** To start the process to build the pipeline click **“Build Now”** on **Jenkins**.

**Integrate with GitHub (Webhook): build after each commit**

**Jenkins Steps:**

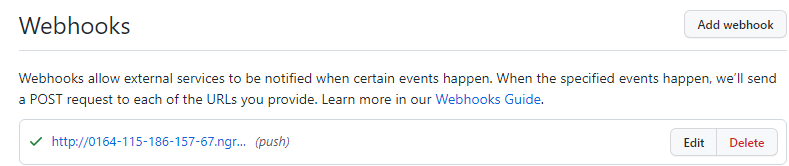
* **Step 1:** Download the plugin for Git **“GitHub Integration Plugin”.**
* **Step 1:** Create new job of Free Style Project.
* **Step 2:** Under Source Code Management check “**Git”** and give the Repository URL (of GitHub).
* **Step 3:** Under Source Code Management give the name of branch to build.
* **Step 4:** Under Build Triggercheck **“GitHub hook trigger for GITScm polling”.**
* **Step 5:** Save the changes**.**

**GitHub Steps:**

* **Step 1:** Go to the repository which you want to integrate with Jenkins. Repository should contain any file in my case there is a **readme file**.



* **Step 2:** Go to Setting Tabs.
* **Step 3:** Go to WebHooks (Webhook allows external services to be notified when certain events happen).
* **Step 4:** Add the WebHooks.
* **Step 5:** Under **Payload URL** give the Jenkins environment URL, i.e. **(**<https://address:port/github-webhooks/>)
* **Step 6:** In **Content Type** select **“application/json”,** leave the **secret field** empty.
* **Step 7:** If the Step 5 won’t work download **ngrok**.
* **Step 8:** Open cmd and type **ngrok.exe - -version.**
* **Step 9:** Type **ngrok.exe http 8080**.
* **Step 10:** Copy the URL from ngrok and give to the **Payload URL**, and click on the update Webhook button.
* **Step 11:** Once you click **“Update Webhook”** button, the Github server will send a request to your Jenkins server and if successful it will show a green check mark as shown below:



* **Step 12:** Go to **“code”** button in GitHub and make some changes to the file and save them.
* **Step 13:** Once you push to the origin on Git it should trigger a build within Jenkins. You will see automatically running of test in Jenkins.

**References:**

1. <https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.fiverr.com%2Faselasg%2Fconfigure-your-devops-tools&psig=AOvVaw2aHsCogwFG-3B_TaKzRN1W&ust=1646394367209000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCNip6I7vqfYCFQAAAAAdAAAAABAJ>
2. <https://www.google.com/url?sa=i&url=https%3A%2F%2Fopensource.com%2Farticle%2F19%2F9%2Fintro-building-cicd-pipelines-jenkins&psig=AOvVaw1xGyBhKUqNN5fvtkEr8T5H&ust=1646394471274000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCMCDi8DvqfYCFQAAAAAdAAAAABAD>
3. <https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.testingxperts.com%2Fblog%2Fwhat-is-devops&psig=AOvVaw3asiD4RAzhdNqtIsyiCBcH&ust=1646394759269000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCIjMmtfwqfYCFQAAAAAdAAAAABAD>