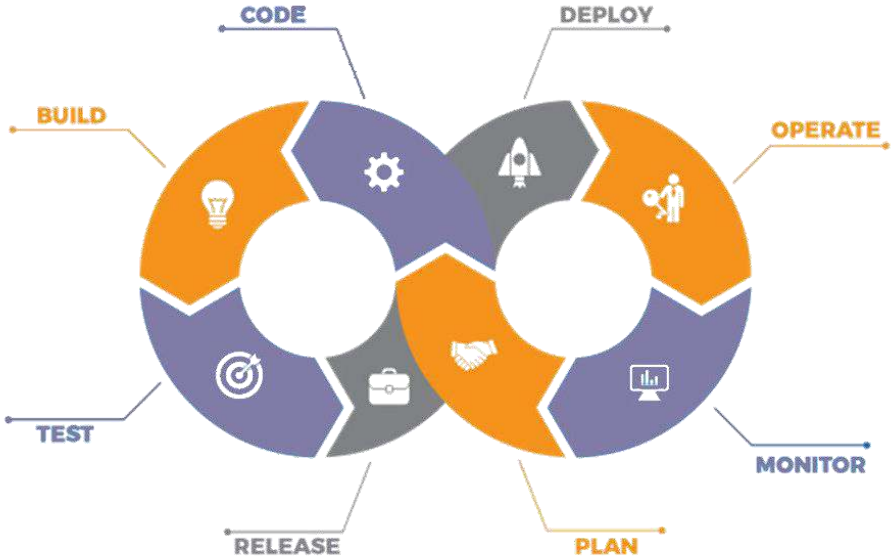
IntelliPaat

Continuous Integration using Jenkins



Agenda

01 INTRODUCTION TO CONTINUOUS INTEGRATION

02 WHAT IS JENKINS?

03 INSTALLING JENKINS ON AWS

04 JENKINS ARCHITECURE

05 MANAGING NODES ON JENKINS

06 JENKINS INTEGRATION WITH DEVOPS TOOLS

07 UNDERSTANDING CI/CD PIPELINES

08 CREATING AN END TO END AUTOMATED PIPELINE IN AWS

Why Continuous Integration?

Before Continuous Integration


Version 1


Developer 1


Version 1


Developer 2

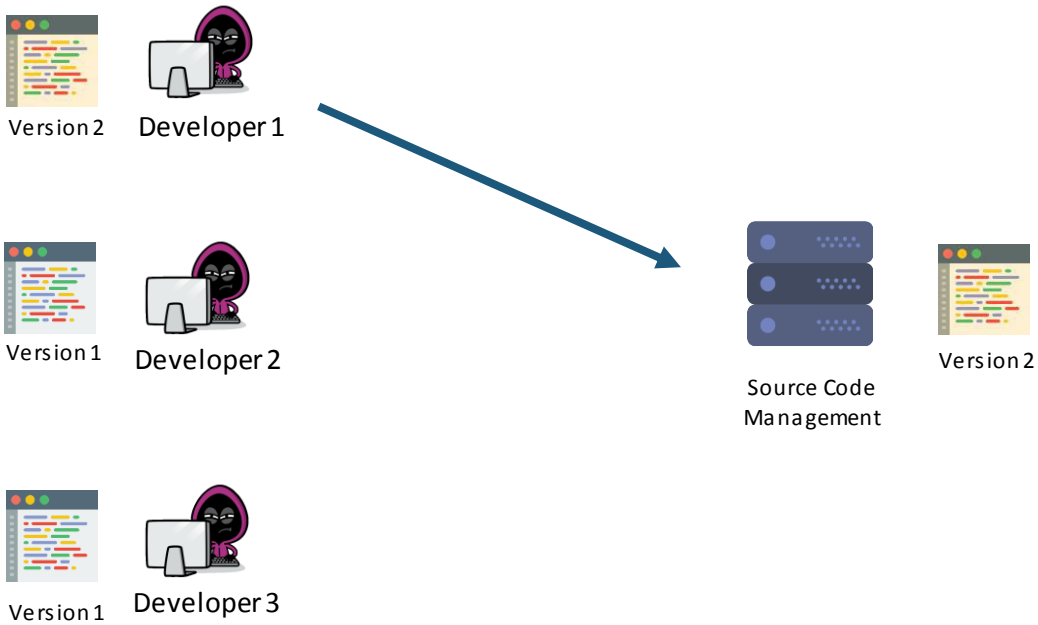

Version 1


Developer 3

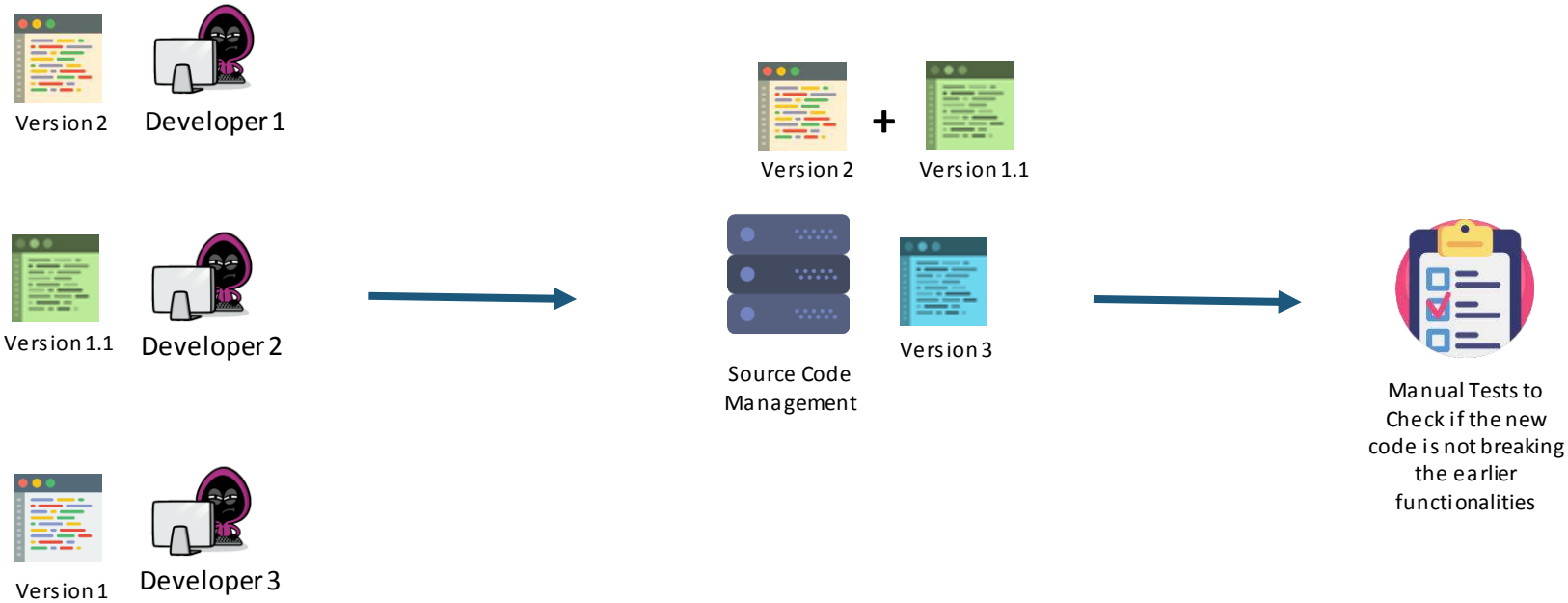

Source Code Management


Version 1

Before Continuous Integration



Before Continuous Integration



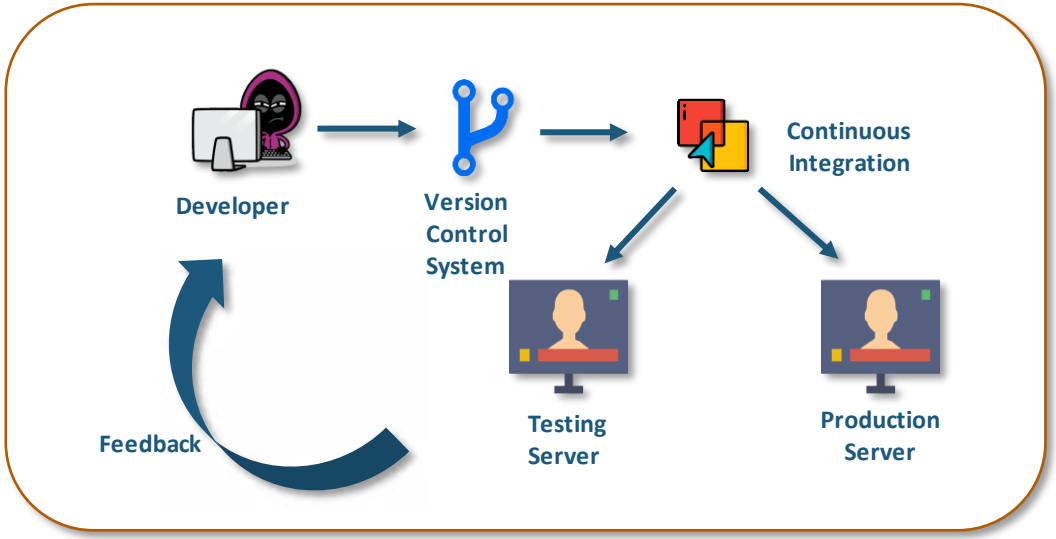
Problems before Continuous Integration

- Infrequent Commits led to bigger releases in one go leading to complex integration
- Manual Testing took a lot of time
- Feedback took a lot of time to reach Developer
- High risk and uncertainty

What is Continuous Integration?

What is Continuous Integration?

The process of having shorter release cycles(sometimes serveral times a day) i.e creating small features and integrating them to the source code, and employing automated build and test processes for quicker feedback is called Continuous Integration



Advantages of Continuous Integration

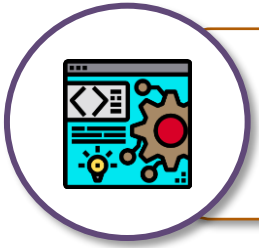
- Frequent Commits hence small feature release
- Automated Build and Testing
- Instant Feedback to Developer
- Low Risk and Faster Delivery

What is Jenkins?

What is Jenkins?

Jenkins is an open source automation server written in Java. Jenkins helps to automate the non-human part of the software development process, with continuous integration and facilitating technical aspects of continuous delivery

Features of Jenkins



Adoption: Jenkins is extremely popular among the open-source community, hence there are more than 147,000 active installations throughout out the world and 1 million people are using it

Plugins Support: With an extremely active open-source community, Jenkins has around 1000 plugins that allow it to integrate with most of the development, testing and deployment tools

Advantages of Jenkins

Before Jenkins

- Locating and fixing bugs in the event of build and test failure was difficult and time consuming
- Tests were triggered Manually
- No Central Place for triggering jobs on remote systems

After Jenkins

- Smaller and Automated continuous build and testing makes the task accurate faster
- Developers have to just commit code in remote repository, build, testing happens automatically
- All the builds or tests on multiple remote systems can be controlled from one place

Installing Jenkins on AWS

Installing Jenkins on AWS

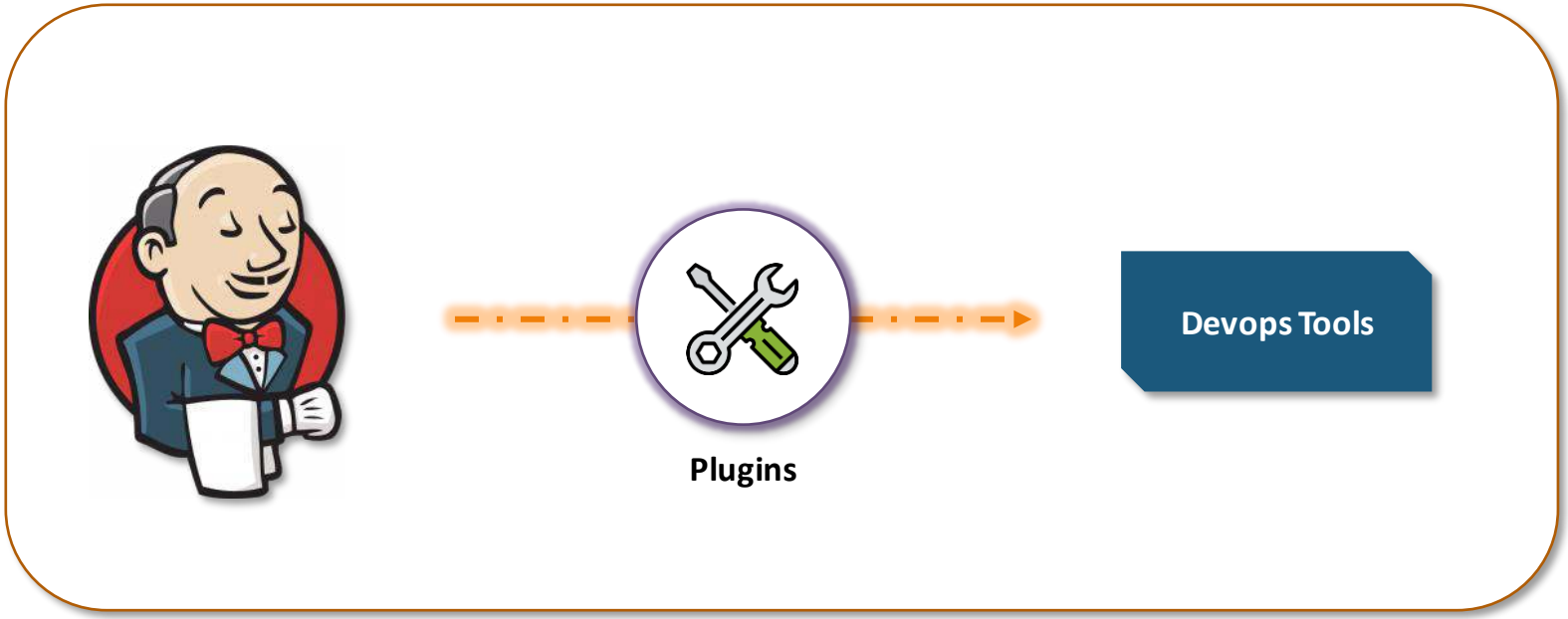
1. Launch an AWS Instance
2. Connect through SSH
3. Execute the following commands:

Jenkins Installation:

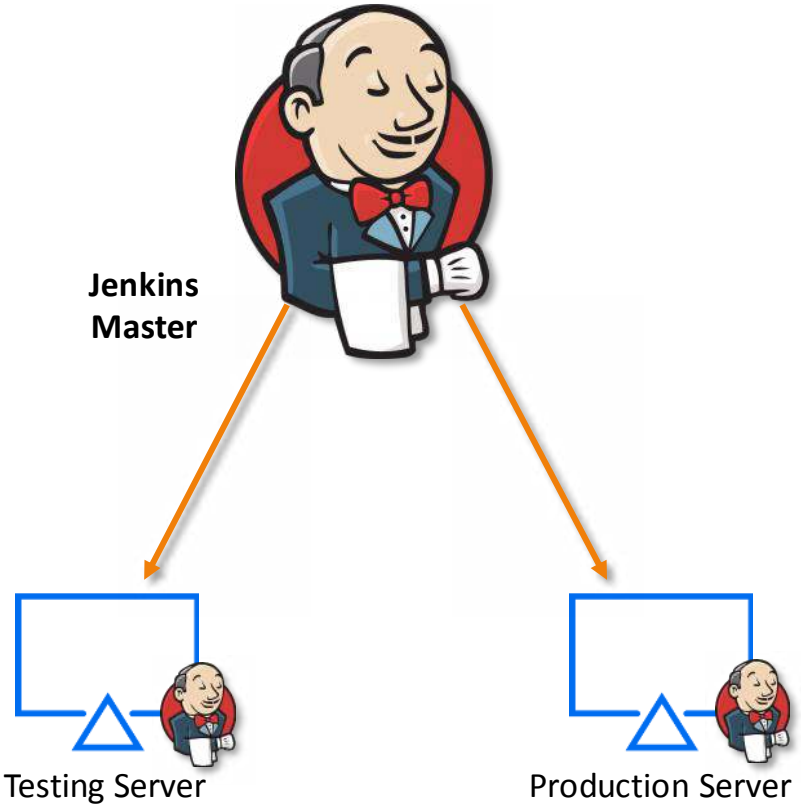
```
$> sudo apt-get update
$> sudo apt install openjdk-8-jdk
$> wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key  | sudo apt-key add -
$> sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable  binary/ >
/etc/apt/sources.list.d/jenkins.list'
$> sudo apt update
$> sudo apt install jenkins
```

Jenkins Architecture

Jenkins Architecture



Jenkins Architecture



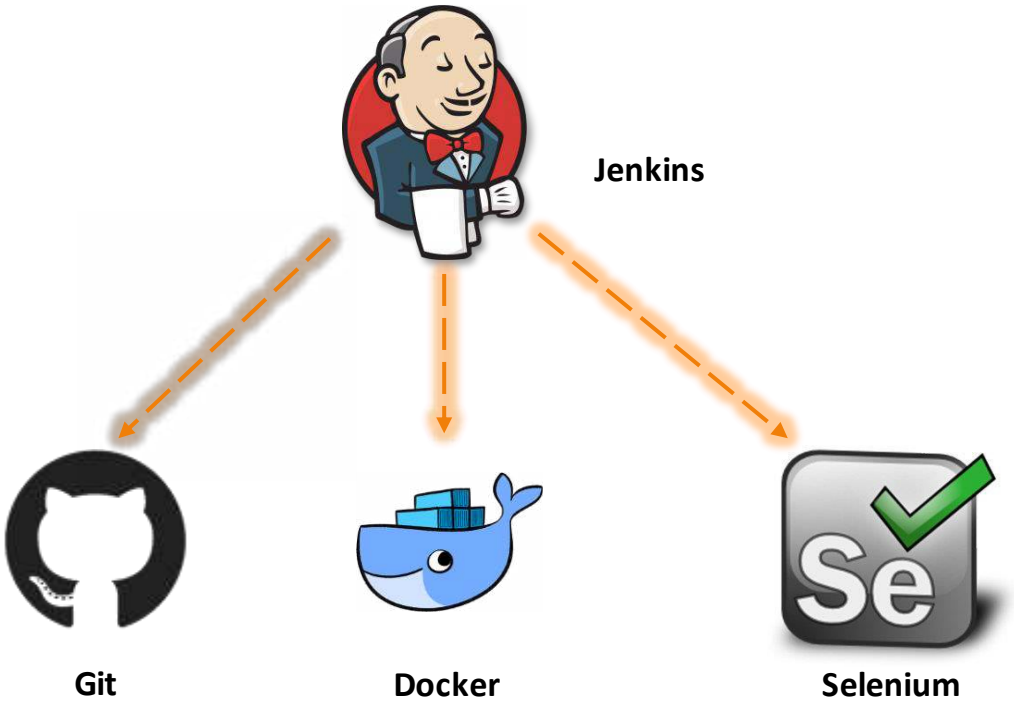
Managing Nodes on Jenkins

Managing Nodes on Jenkins

Add a slave node to Jenkins, using JNLP connection

Jenkins Integration with Devops Tools

Jenkins Integration with Devops Tools



Jenkins Integration with Devops Tools

- Git
- Docker
- Selenium

Copy a Git repository to the slave's filesystem from Jenkins master

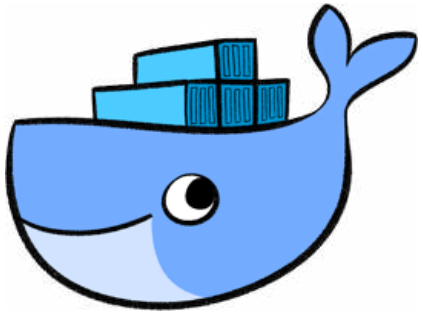
Jenkins Integration with Devops Tools

Git

Docker

Selenium

Containerize the website in previous step to a Docker Container using Jenkins



Jenkins Integration with Devops Tools

Git

Docker

Selenium

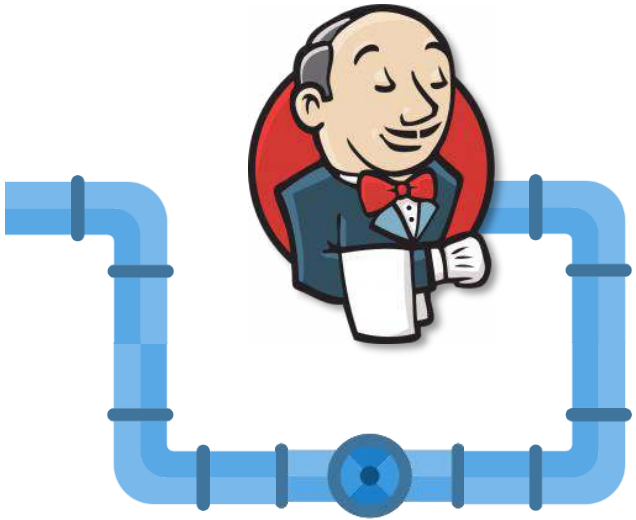
Create a test case for the website in the previous step, and execute the test on the slave using Jenkins

Understanding CI/CD Pipelines

What are CI/CD Pipelines?

CICD Pipelines i.e Continuous Integration, Continuous Delivery and Deployment pipelines, are a way of running Jenkins jobs in a sequence, which resembles a pipeline view.

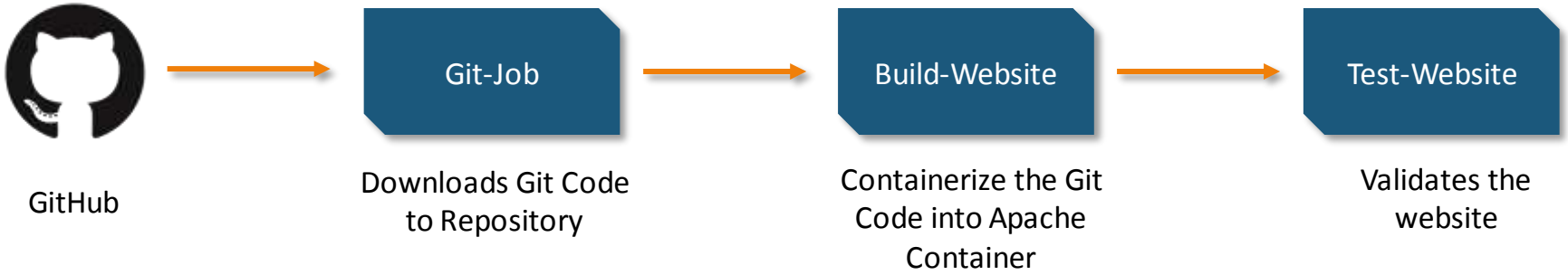
Finished Application



What are CI/CD Pipelines?

CICD Pipelines i.e Continuous Integration, Continuous Delivery and Deployment pipelines, are a way of running Jenkins jobs in a sequence, which resembles a pipeline view.

For Example:



Creating an Automated CI/CD Pipeline

Creating an Automated CI/CD Pipeline

1. Initiate a Git Webhook for the Jenkin's git-job repository
2. Trigger the jobs after completion of previous jobs with the following map: Git-Job → Build-Website → Website-Test
3. Install the plugin for Pipeline View
4. Make changes to the website and commit the job to see the changes

India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24X7 Chat with our Course Advisor