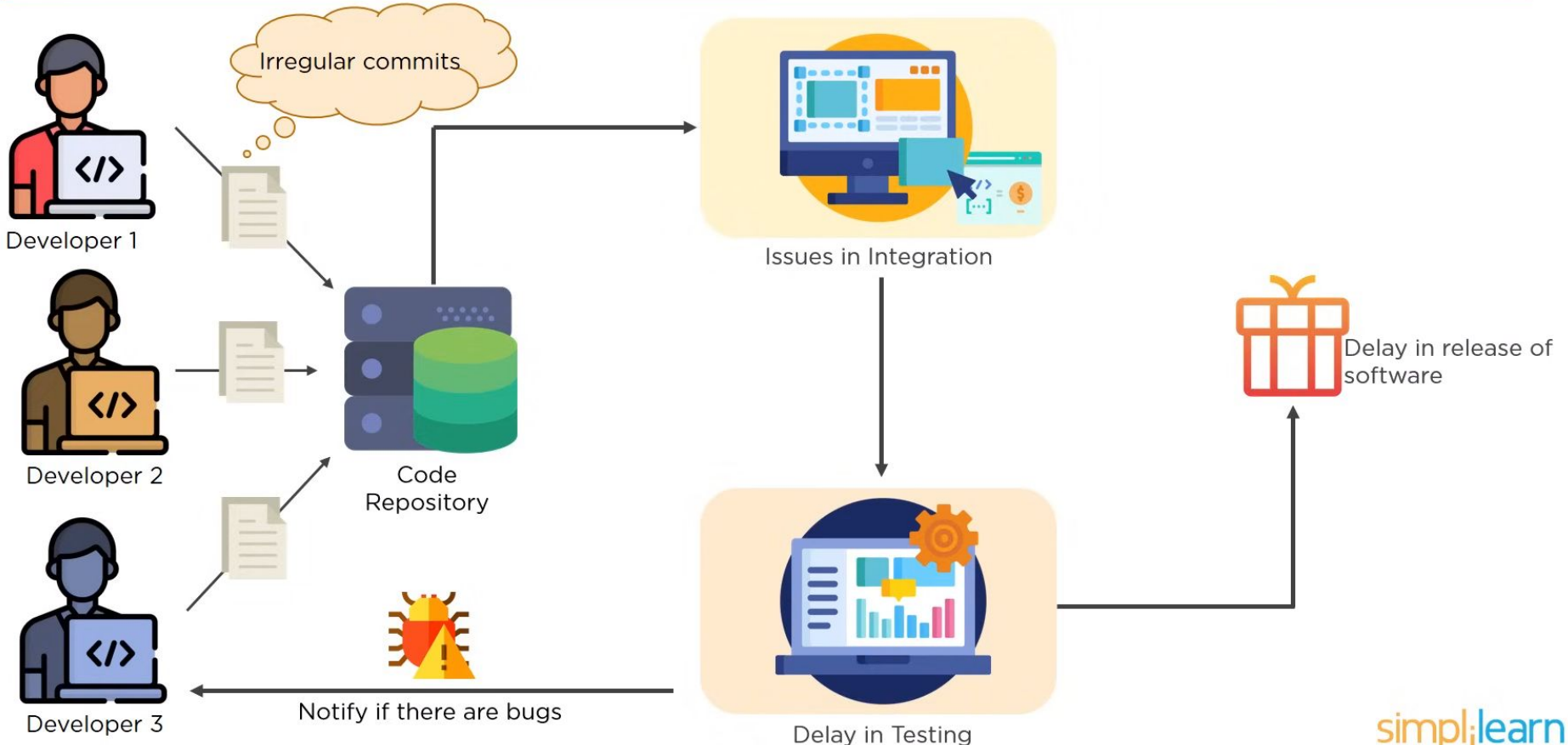
Jenkins

Agenda

- What is Jenkins ? Use ?
- What is Continuous Integration?
- What makes Jenkins more powerful ?
- Benefits of using CI to software development
- What is Continuous Delivery?
- What is Continuous Deployment?
- Jenkins Architecture Types
- Jenkinsfile & Pipeline syntax types
- Installing and Configuring Jenkins.

Before Jenkins



Before Jenkins

Developer 1

Developer 2

Developer 3

Developers had to wait till the entire software code was built and tested to check for errors

simplilearn

What is Continuous Integration ?

- It is a **development practice** in which developers are required to commit changes to source code in a shared repository several times a day.
- Every commit in the repository is then build and this allows the teams to detect problems early.

Continuous Integration (CI) Stage :

1) Developers upload their code on version control system like git.
2) Build the code (If we used java, built it using maven, If we used docker, built it using dockerfile)
3) Push the image on any Artifact repository like Dockerhub.
4) Run unit test on code if exists.

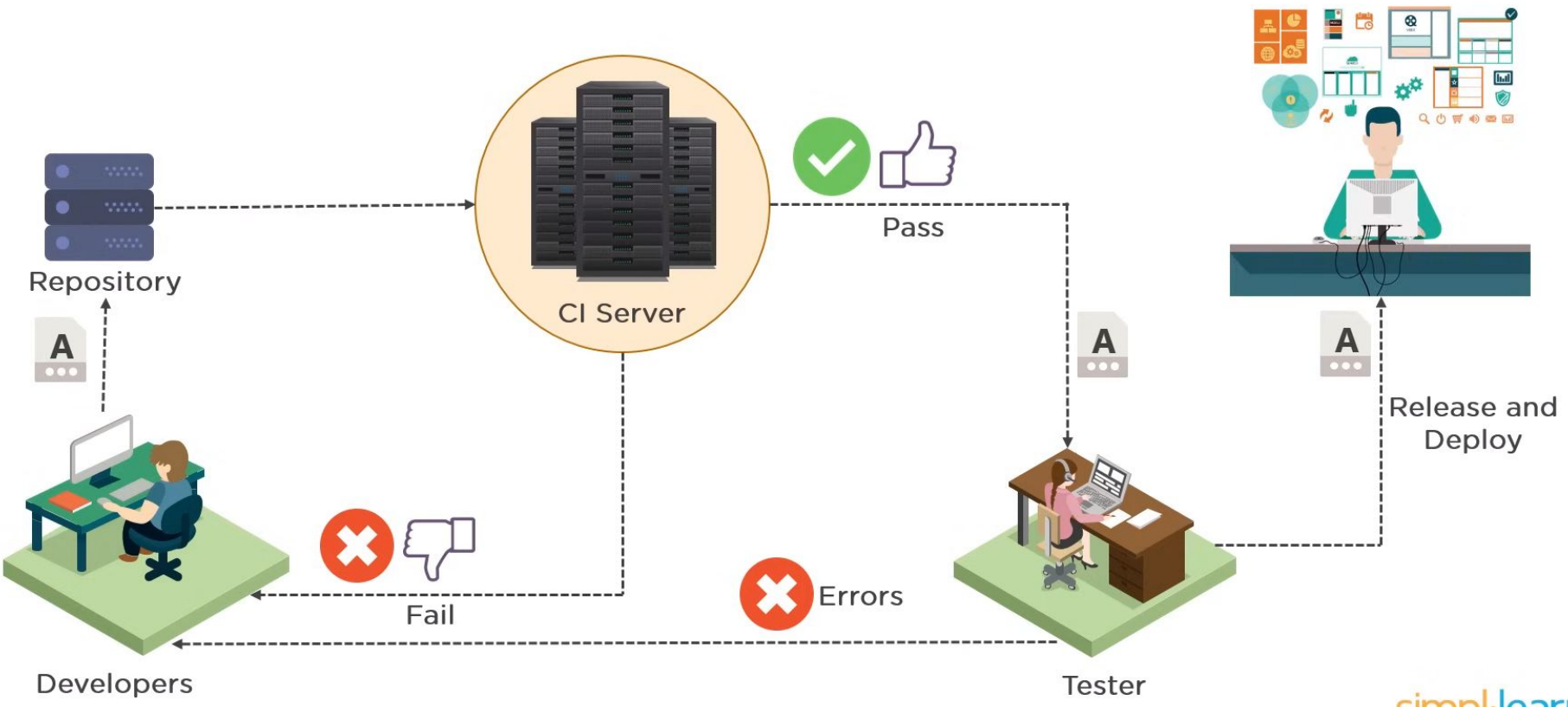
What is Continuous Delivery?

- Continuous delivery refers to the approach of ensuring that software can be released to production at any time.
- Continuous delivery involves automating the entire software release process, including testing and deployment to staging environments, but the actual release to production is still triggered **manually**.

What is Continuous Deployment?

- Continuous deployment takes continuous delivery one step further. It refers to the practice of automatically deploying every build that passes the automated tests to production.
- Continuous deployment involves **fully automating** the release process, where successful builds are automatically deployed to production **without any manual intervention**.

What is Continuous Integration?



Benefits of using CI to software development

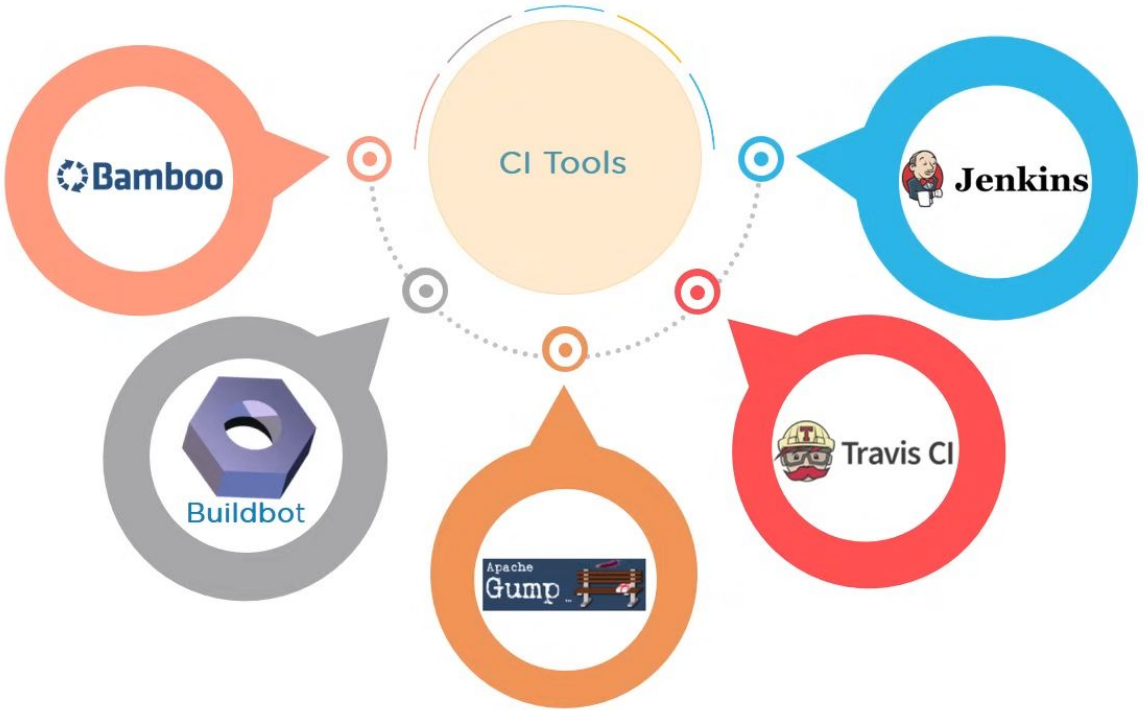
1)Catch issues fast and help to fix problems earlier.
2)Everyone can see what's happening.
3)Automate the build.
4)Continuous Integration leads to Continuous Deployment allowing us to deliver software more rapidly.

Requirements of Continuous Integration Practices

There are some hard requirements for a Continuous Integration workflow to take place:

- A Version Control System tool
- A Build Tool
- An Artifacts Repository Manager

Continuous Integration Tools

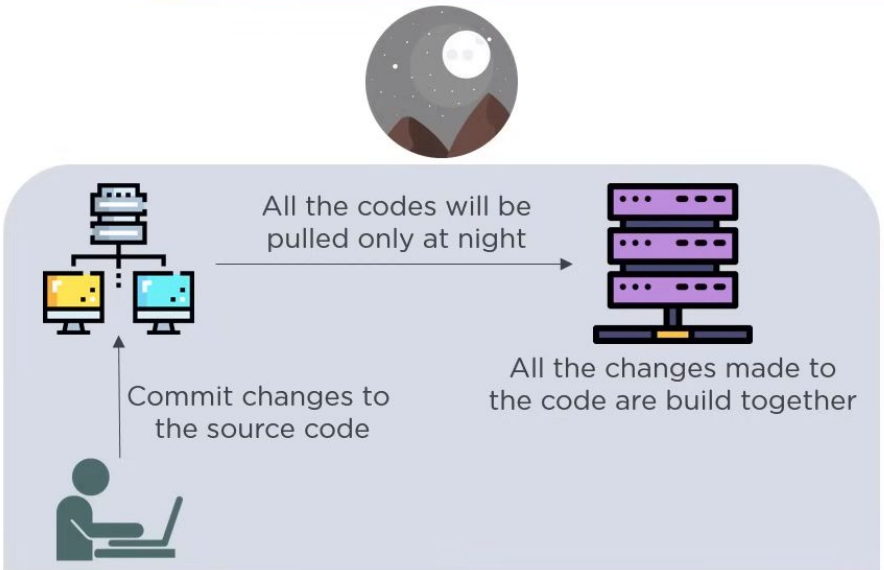


What is Jenkins ? Use ?

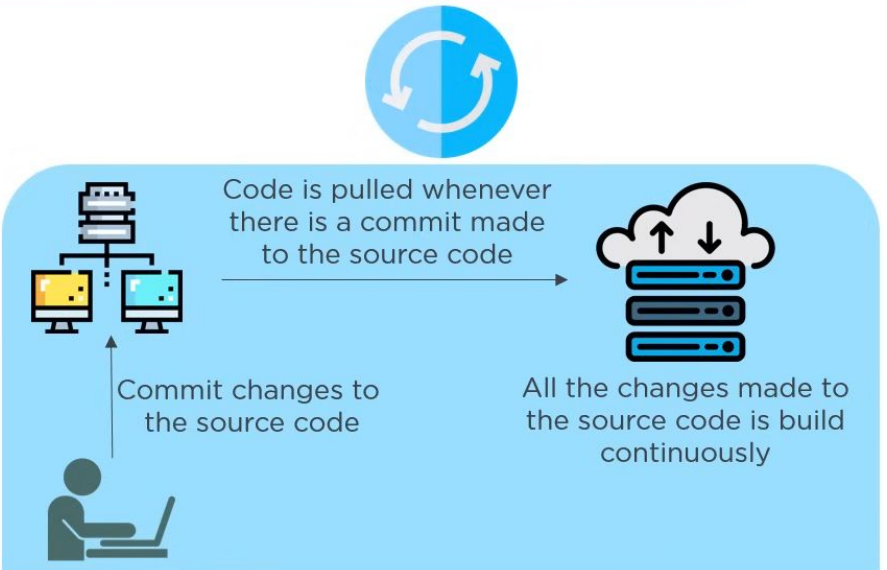
- Jenkins is an **open source** continuous integration/continuous delivery and deployment (CI/CD) **automation** software DevOps tool written in the Java programming language.
- It is used to implement CI/CD workflows, called **pipelines**.
- Jenkins is extremely powerful with vast amount of **plug-in supported**.
- Jenkins server is a **web dashboard** which is nothing but powered from a war file, default run on 8080 ports.
- Jenkins is a popular open-source automation server **used for building, testing, and deploying software.** It facilitates **continuous integration and continuous delivery (CI/CD) processes** by automating the steps involved in the software development lifecycle.

What is Jenkins?

Jenkins is a Continuous Integration tool that allows continuous development, test and deployment of newly created codes



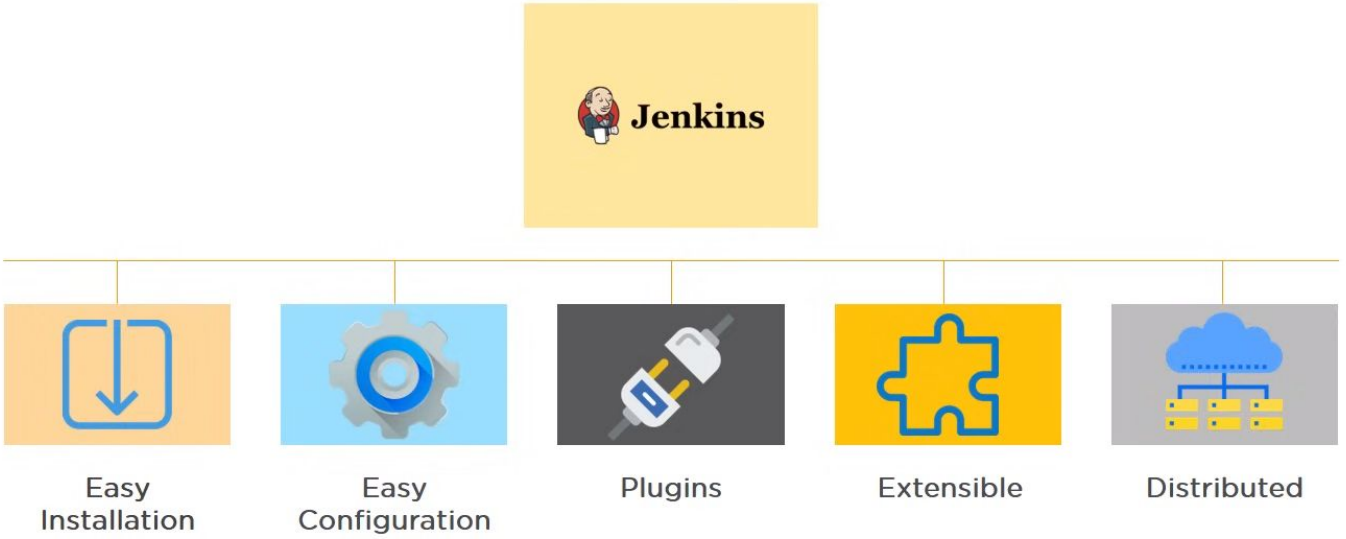
Nightly build and integration



Continuous build and Integration

simplilearn

Features of Jenkins

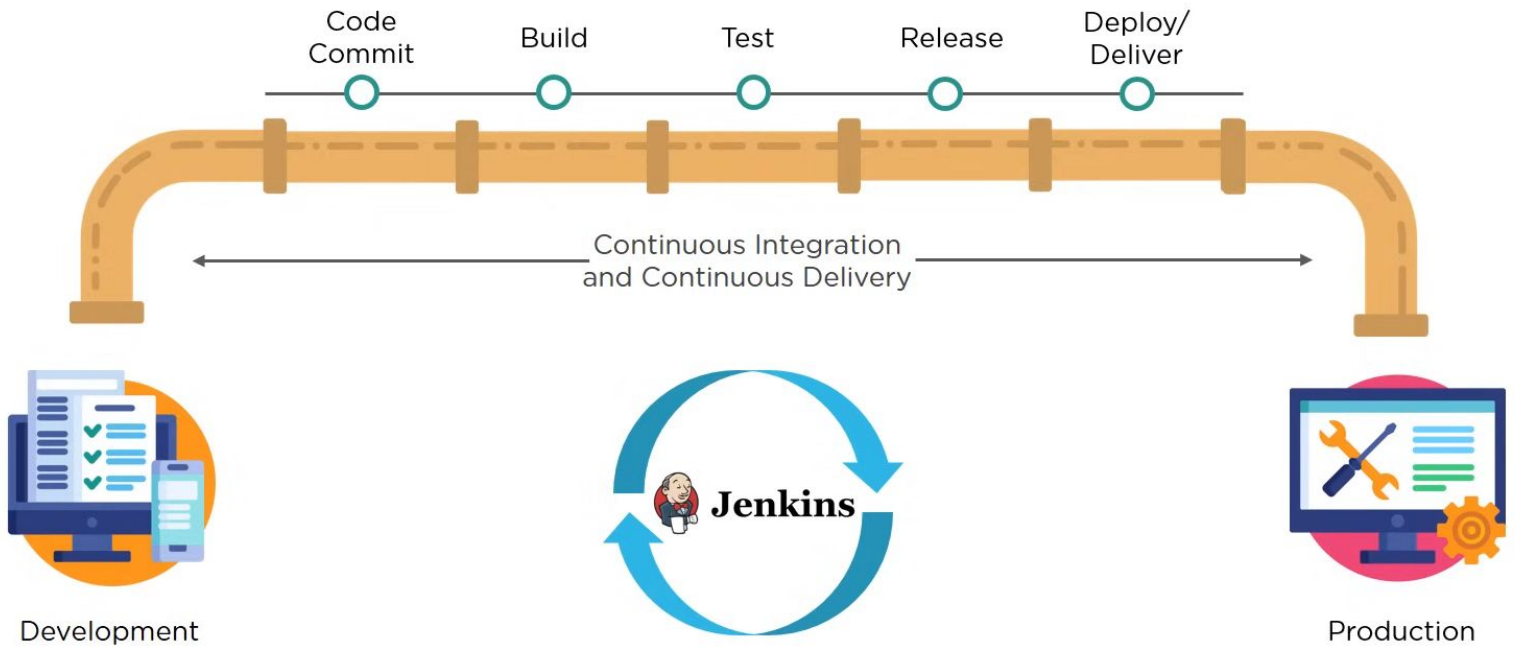


Jenkins is a self contained Java-based program, ready to run with packages for Windows, Mac OS X and Unix-like OS

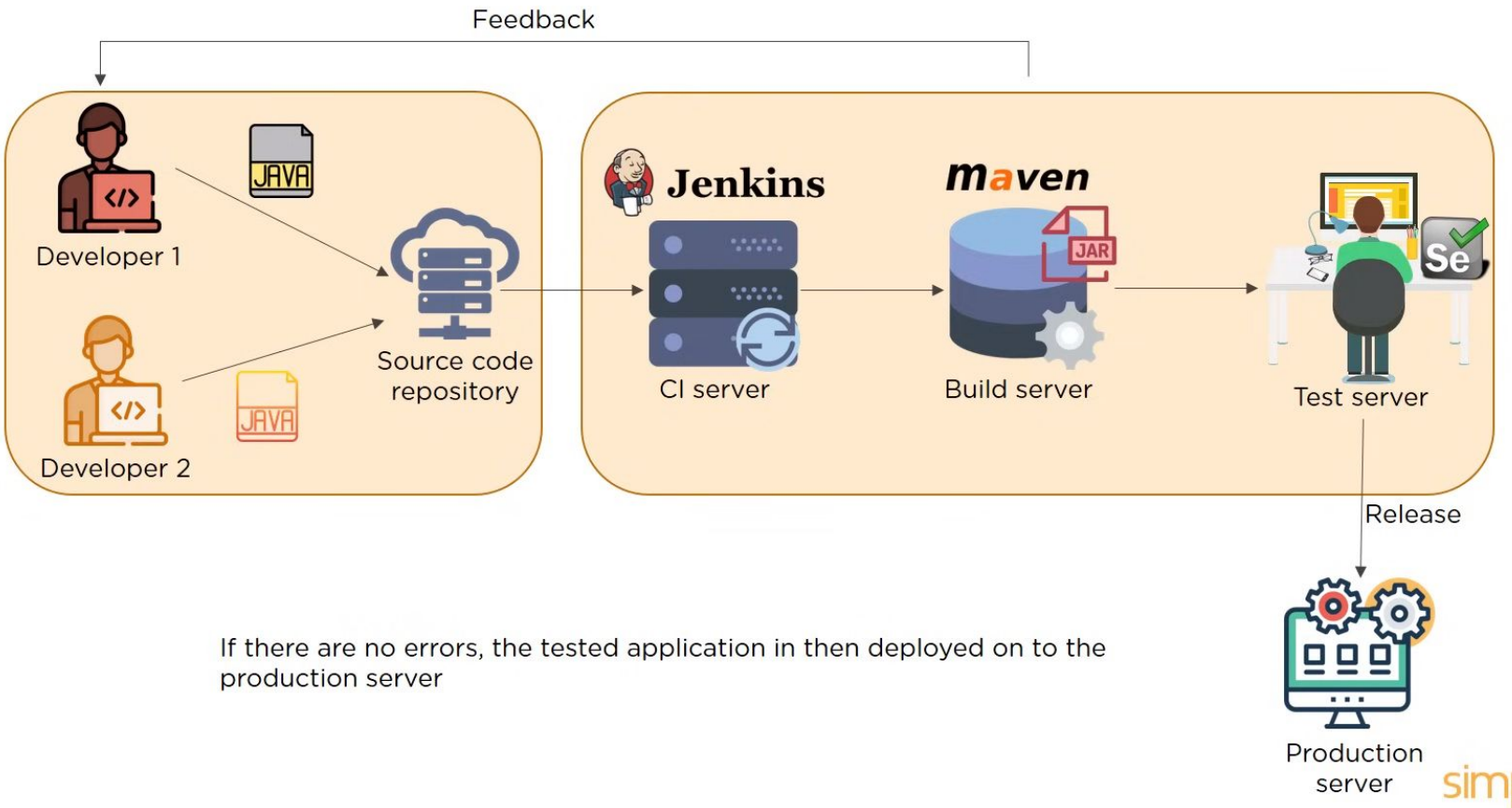
What makes Jenkins more powerful ?

- A plugin is an enhancement to the Jenkins system.
- They help extend Jenkins capabilities and integrated Jenkins with other software.
- Plugins can be downloaded from the online Jenkins Plugin repository and loaded using the Jenkins Web UI or CLI.
- You can also customize your plugin and share it with community.

Jenkins Pipeline

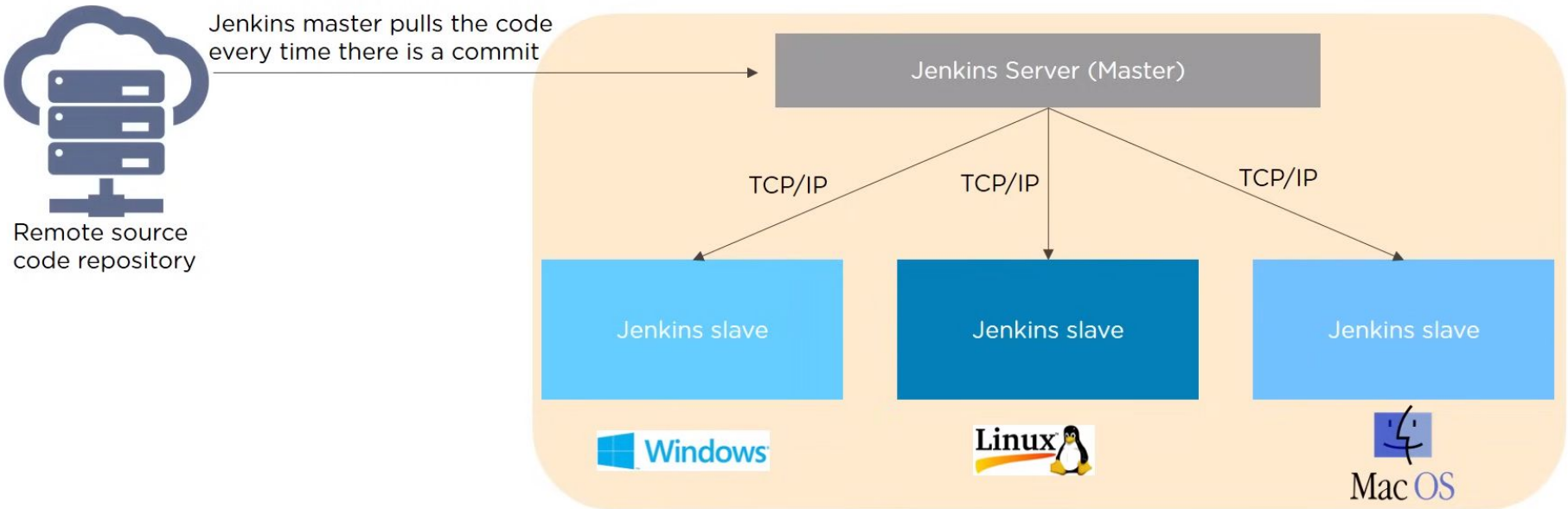


Jenkins Architecture



If there are no errors, the tested application in then deployed on to the production server

Jenkins Master-Slave Architecture



- Jenkins master distributes its workload to all the slaves
- On request from Jenkins master, the slaves carry out builds and tests and produce test reports

Jenkins Architecture

Jenkins Master	Jenkins Slave
Your main Jenkins server is the Master. And Master's job is to handle scheduling build jobs.	It hears requests from the Jenkins Master instance.
Dispatching builds to the slaves for the actual execution.	Slaves can run on a variety of operating systems.
Monitor the slaves (possibly taking them online and offline as required).	The job of a Slave is to do as they are told to, which involves executing build jobs dispatched by the Master.
A Master instance of Jenkins can also execute build jobs directly.	You can configure a project to always run on a particular Slave machine or a particular type of Slave machine, or simply let Jenkins pick the next available Slave.

Task

Task 1 (3 Grades)

1- Install jenkins (optional: using docker image)

2- Complete configuration

3- Create admin user

4- Create a pipeline that executes and prints “Hello World”

Task 2 (7 Grades)

1- Create a bash script file that executes the “ls” command

2- Push the bash file into a newly created repo

3- Create a new pipeline item on jenkins

4- Create a CI/CD for this by configuring jenkins to pull the repo and execute the bash file