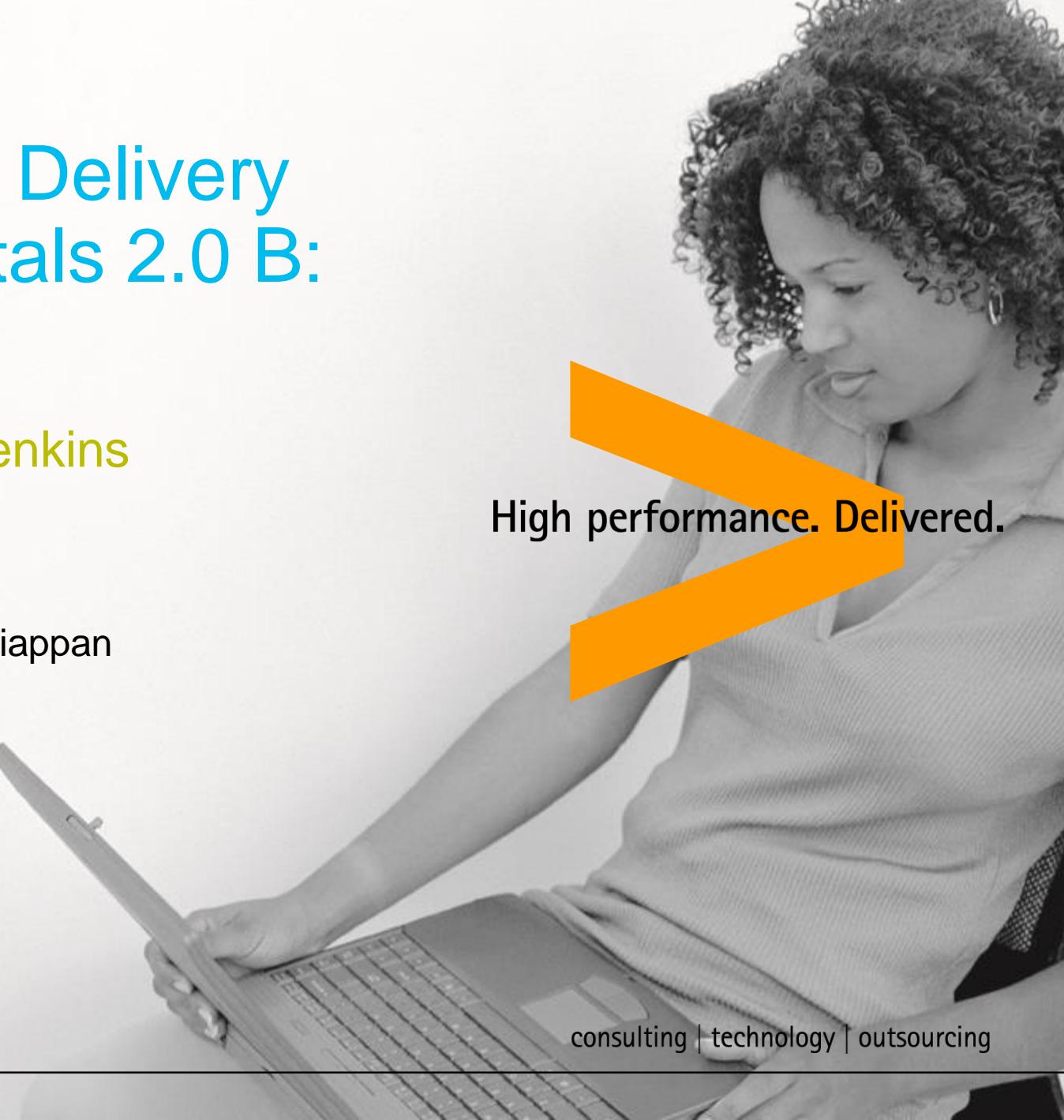


# Application Delivery Fundamentals 2.0 B: Java

Introduction to Jenkins

Parameswari Ettiappan

A black and white photograph of a woman with curly hair, wearing a light-colored shirt, sitting at a desk and working on a laptop. She is looking down at the screen. The background is plain.

High performance. Delivered.

# Goals

---

- Devops
- CI/CD Pipeline
- Maven Build
- Docker Build
- Docker Run
- Jenkins Pipeline
- Sonar Qube

# Software Requirements

---

- Minimum java 8
- STS 4
- maven 3.6
- jenkins 2 or more
- sonar qube 6.6
- github/gitlab
- Docker desktop wsl2

# Water Fall Model vs Agile

---

## Waterfall

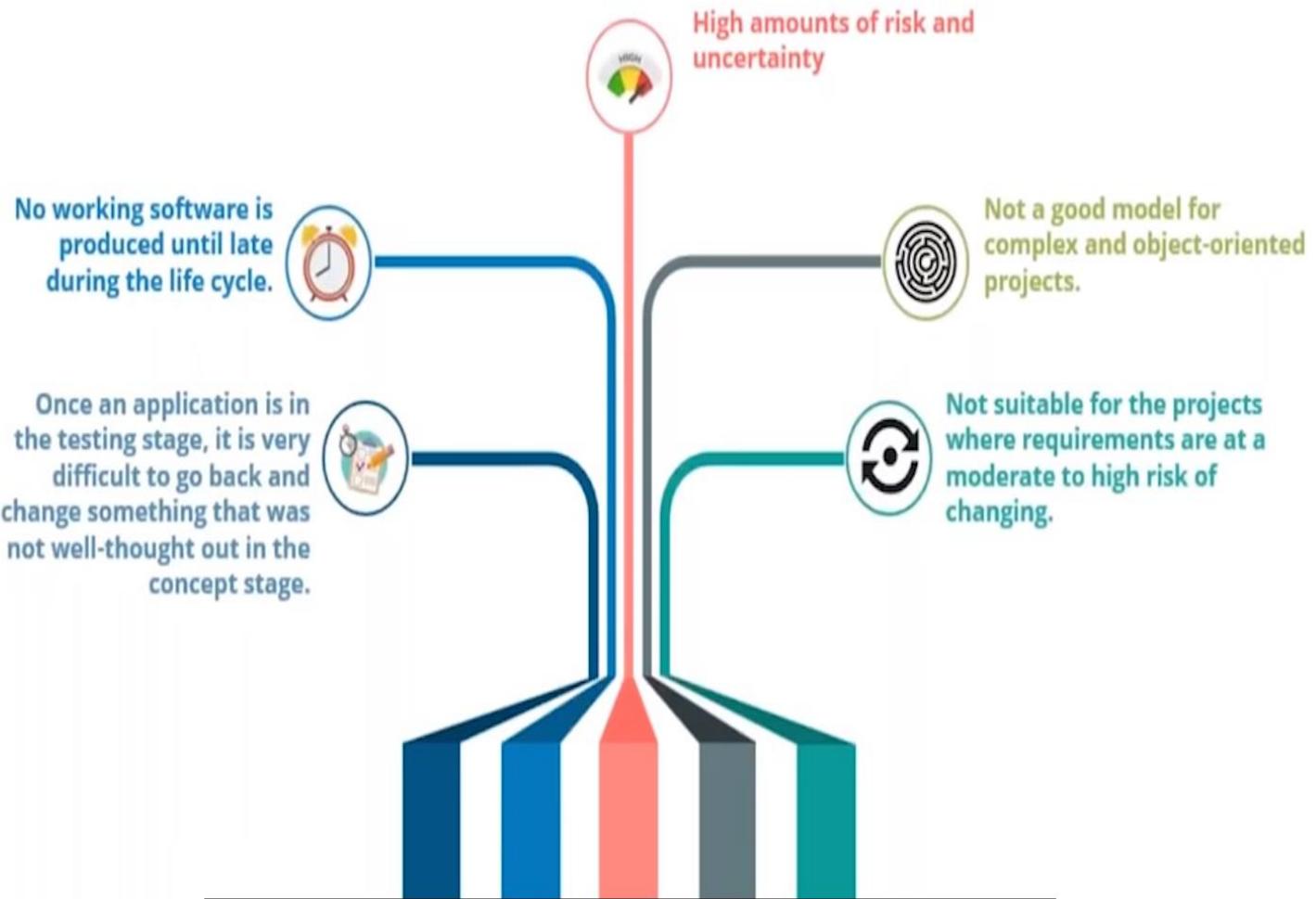


## Agile

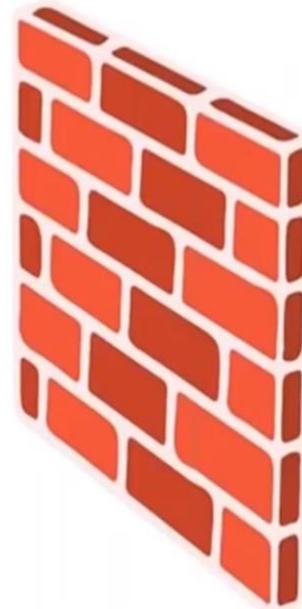


# Limitations of Waterfall Model

---



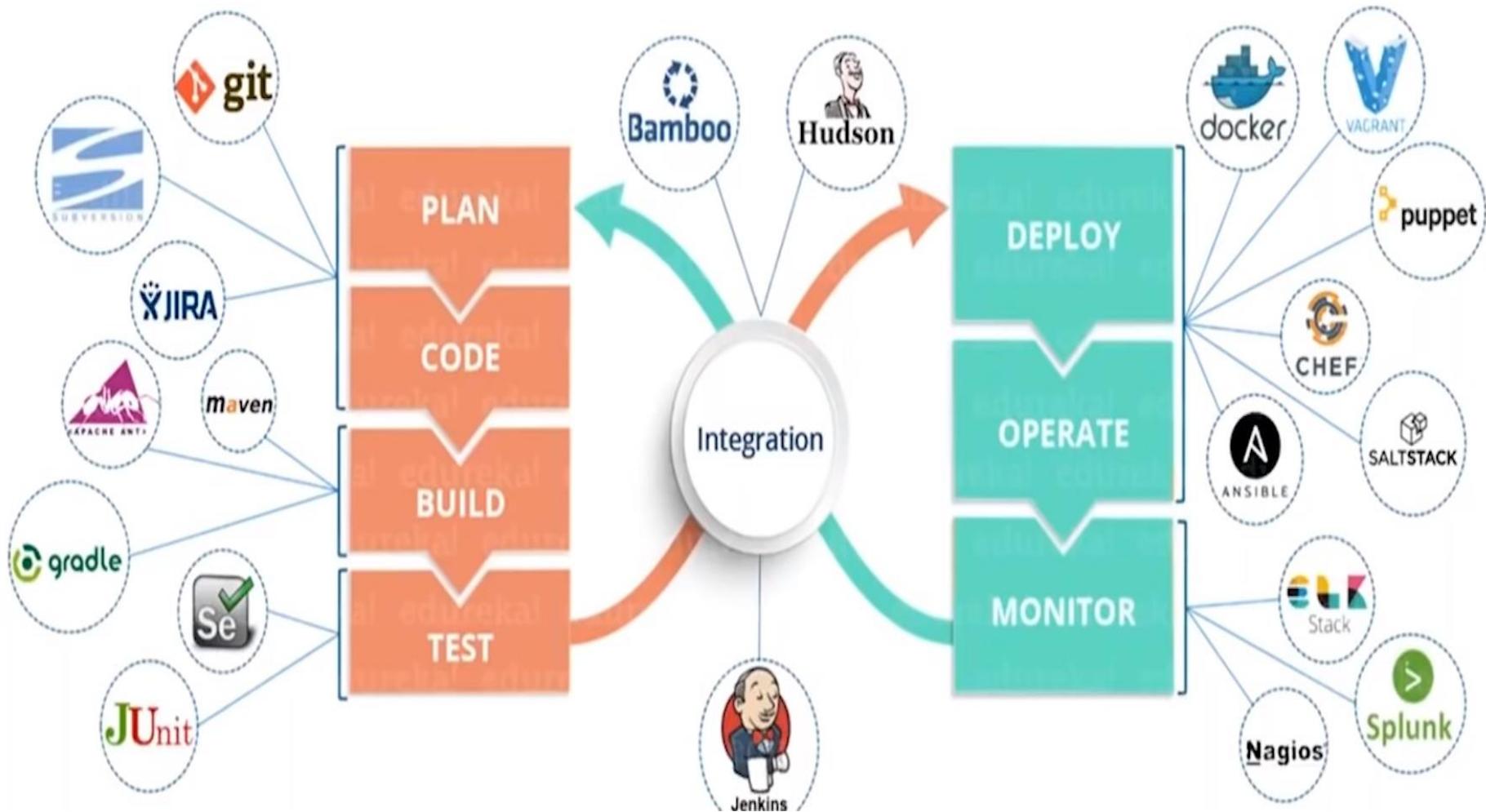
# Limitations of Agile



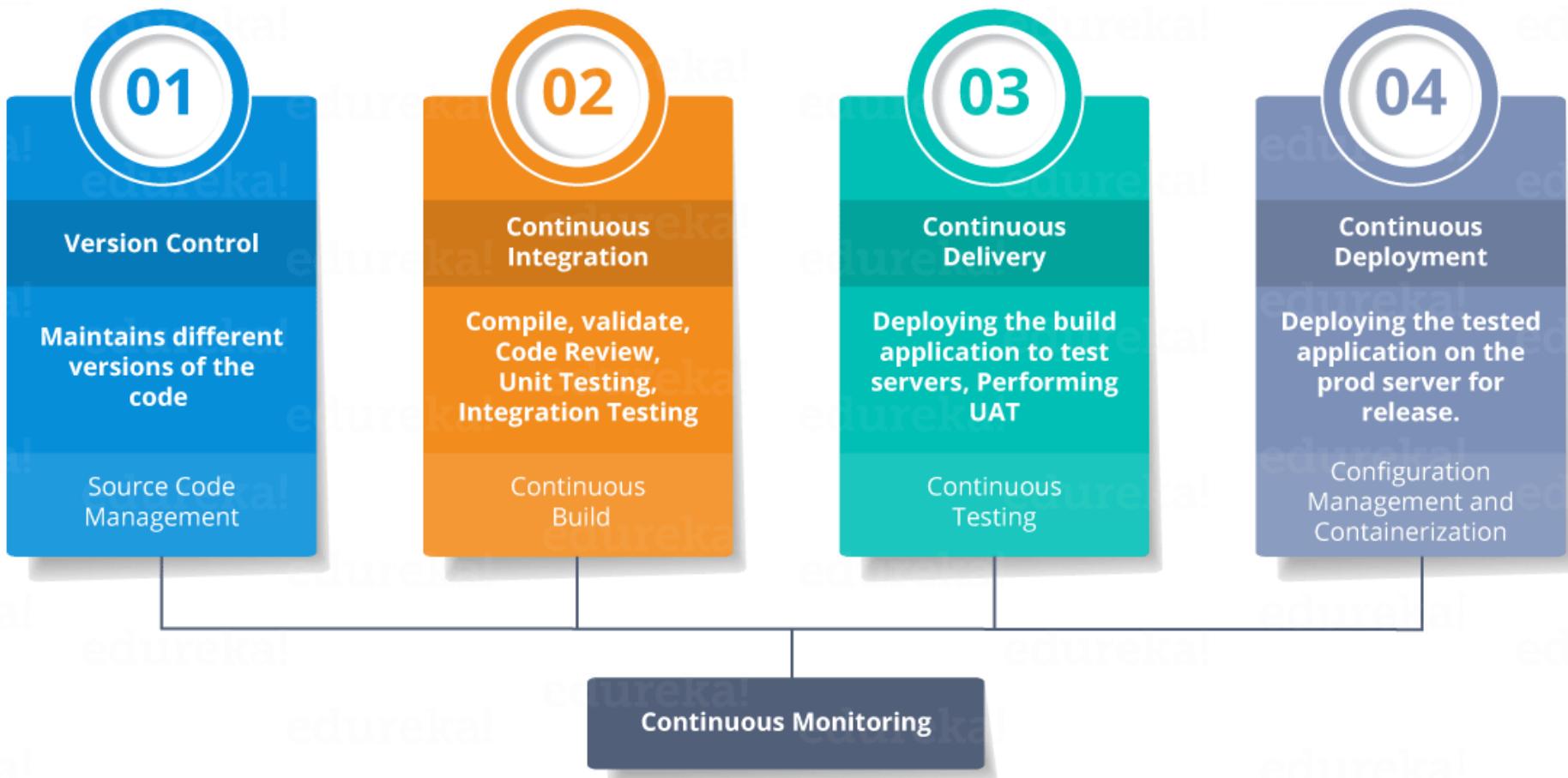
Wants Change

Wants Stability

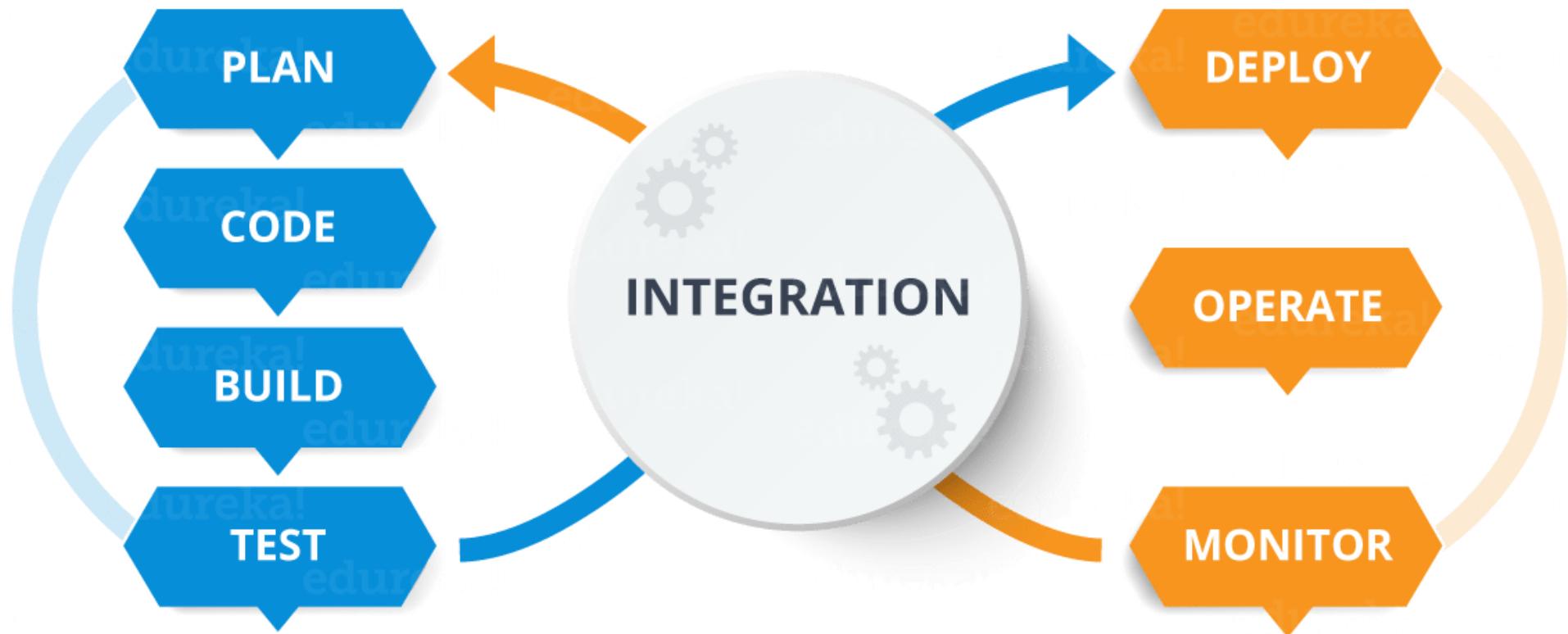
# What is Devops



# CI/CD Pipeline



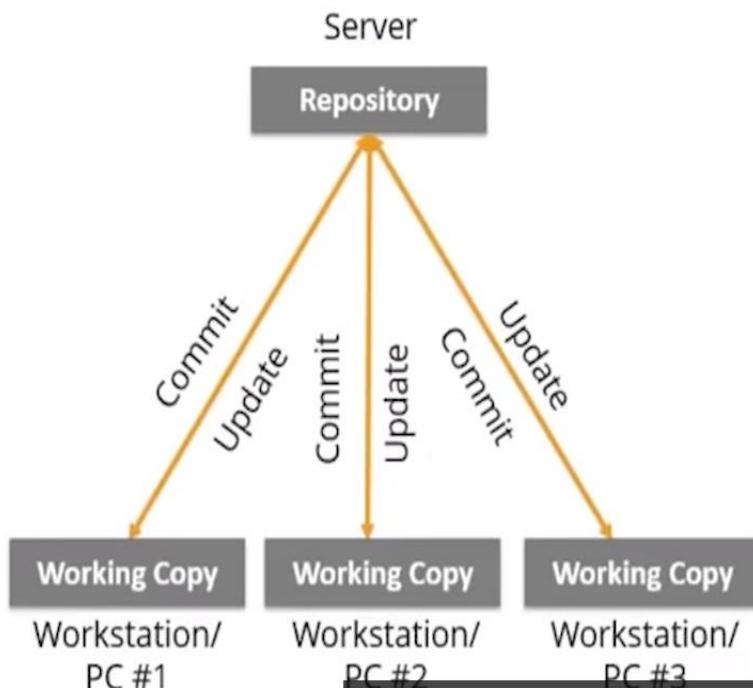
# CI/CD Pipeline



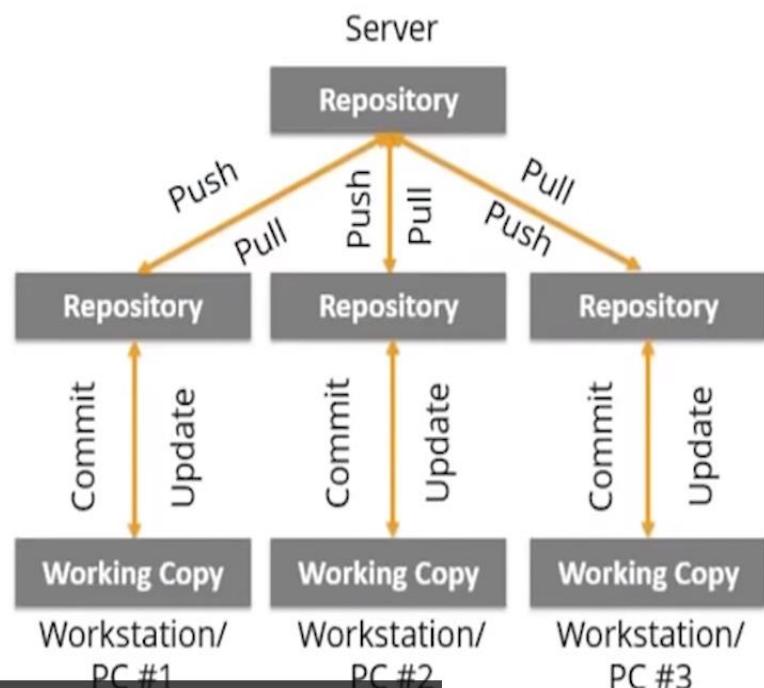
# Source Code Management

The management of changes to documents, computer programs, large websites and other collection of information

## Centralized Version Control System



## Distributed Version Control System

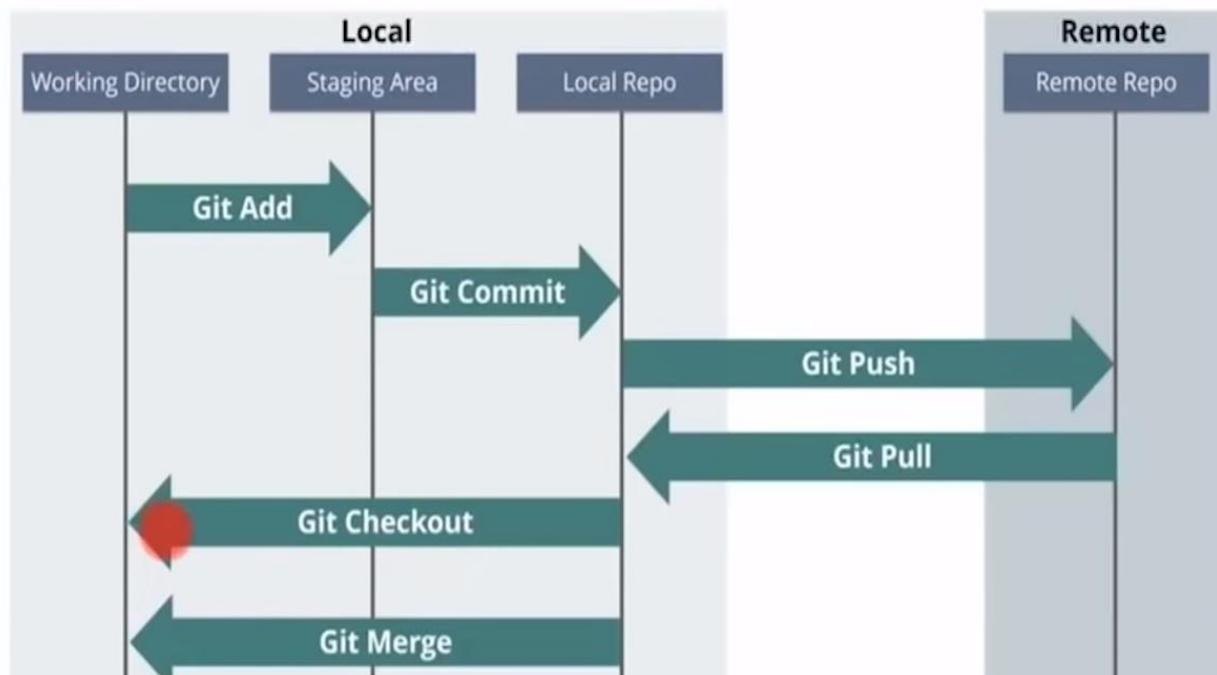


# Source Code Management

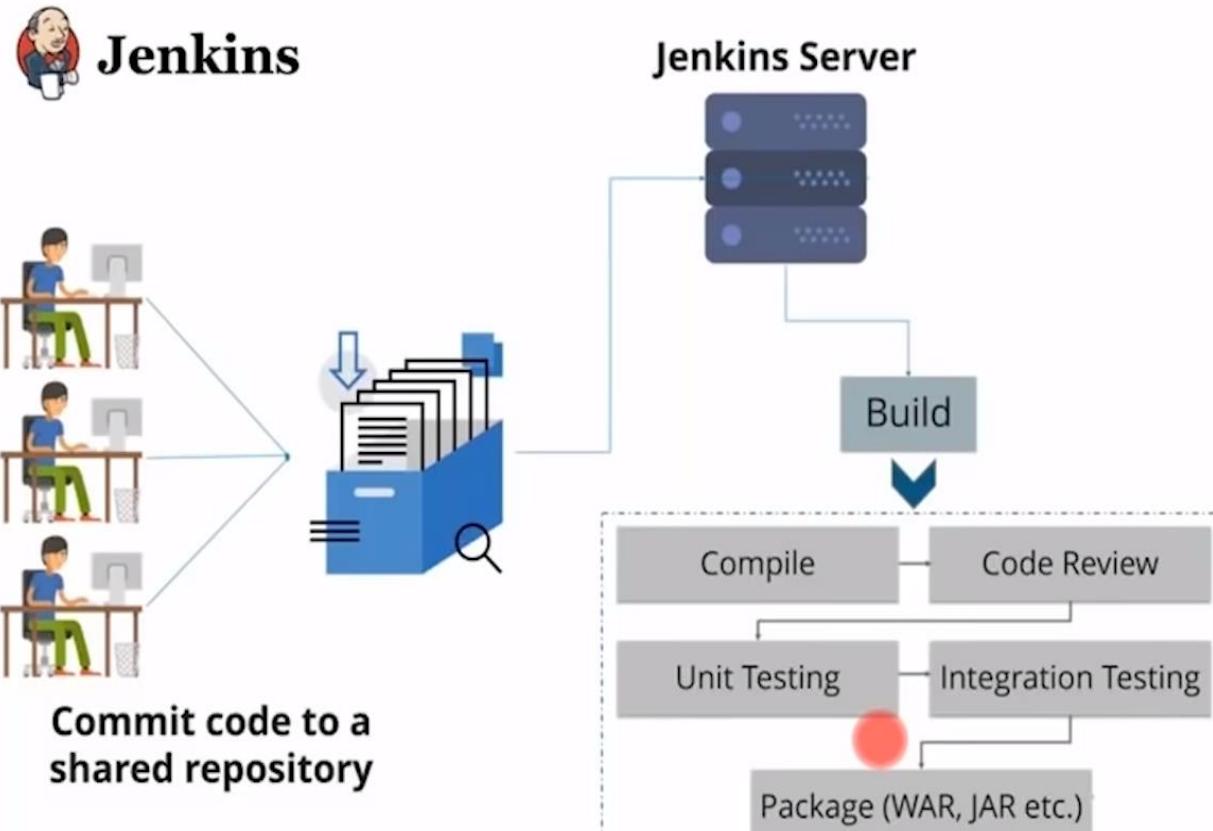


git

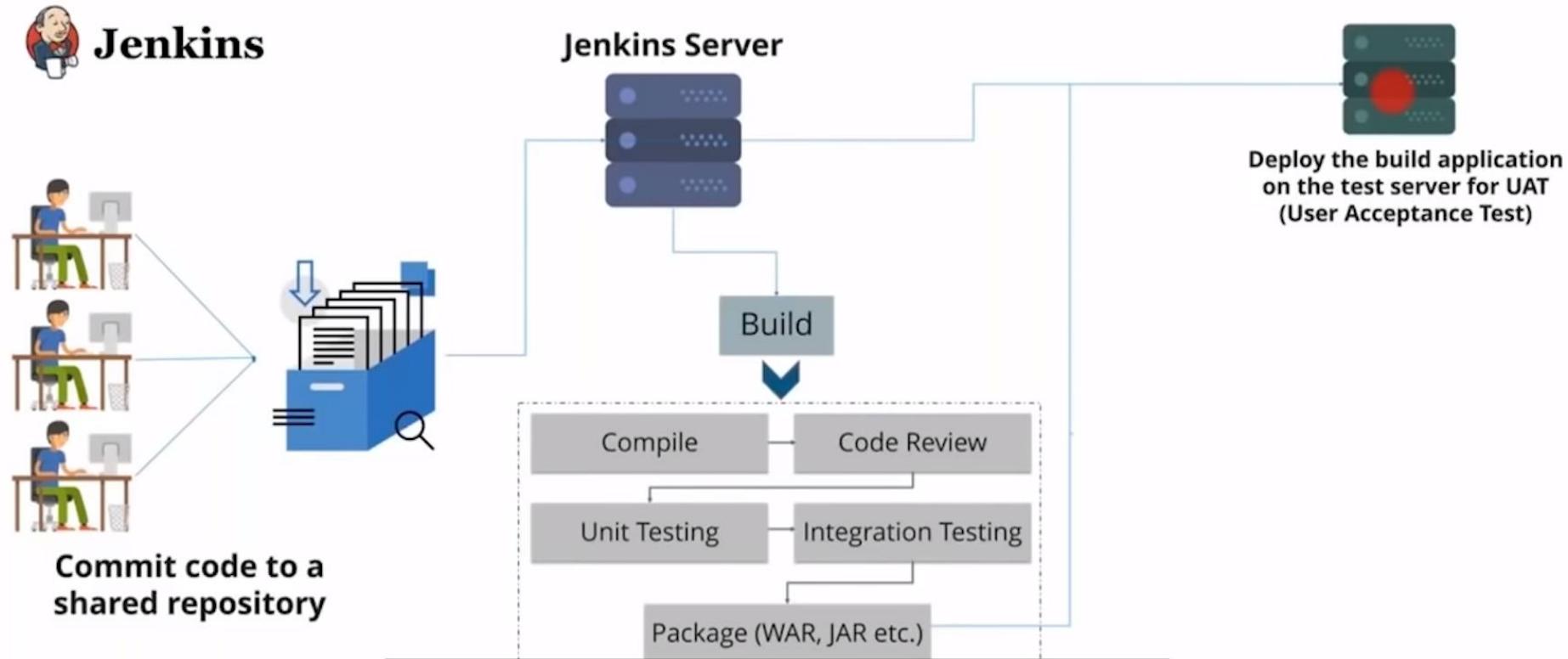
Git is a Distributed Version Control tool that supports distributed non-linear workflows by providing data assurance for developing quality software



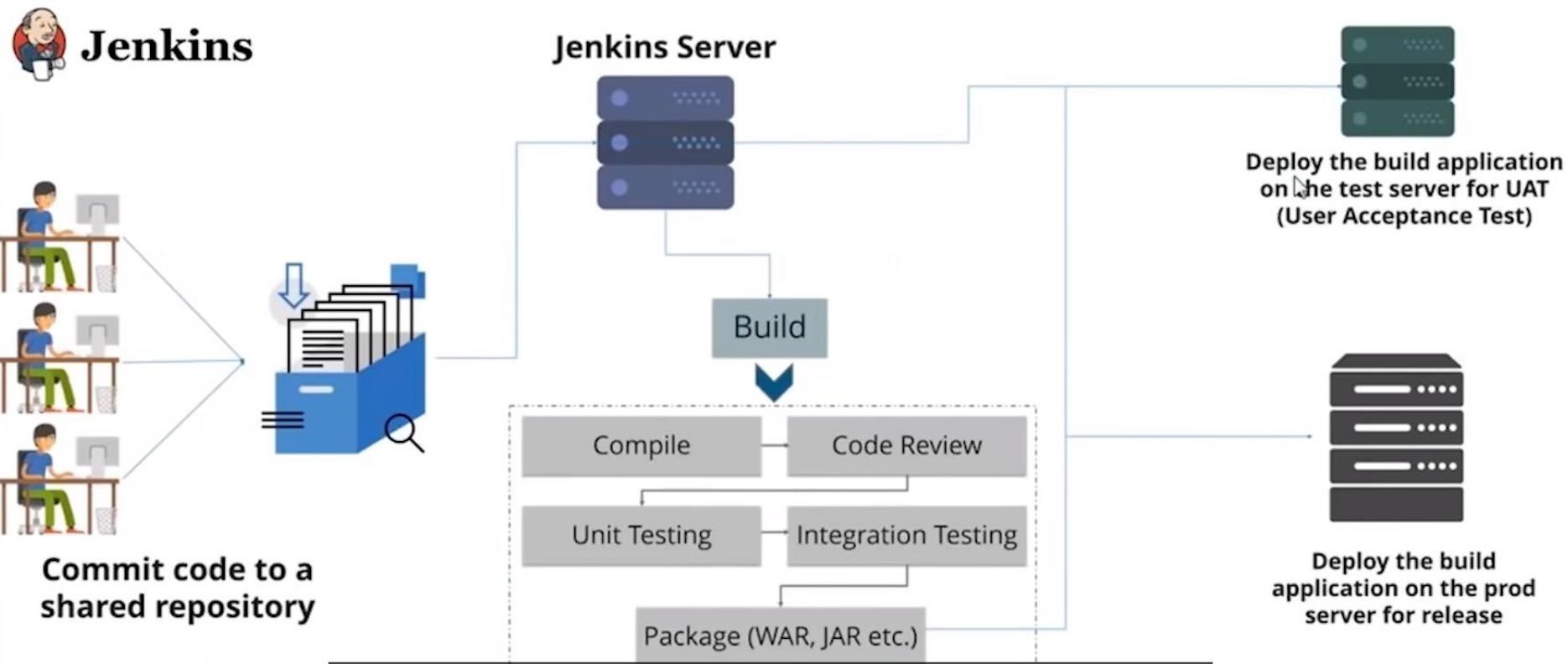
# Continuous Integration



# Continuous Delivery



# Continuous Deployment

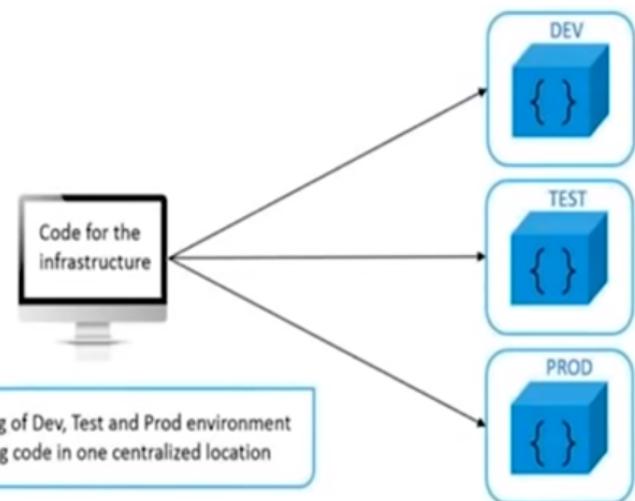


# Configuration Management

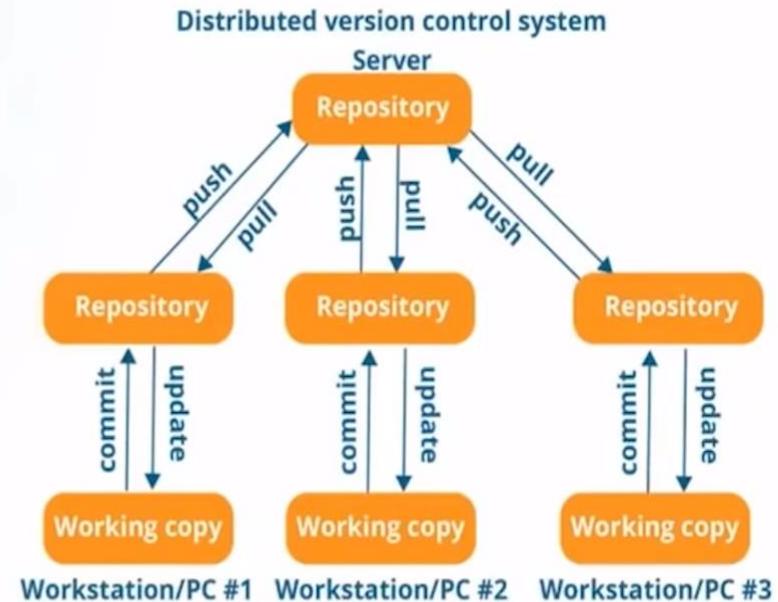
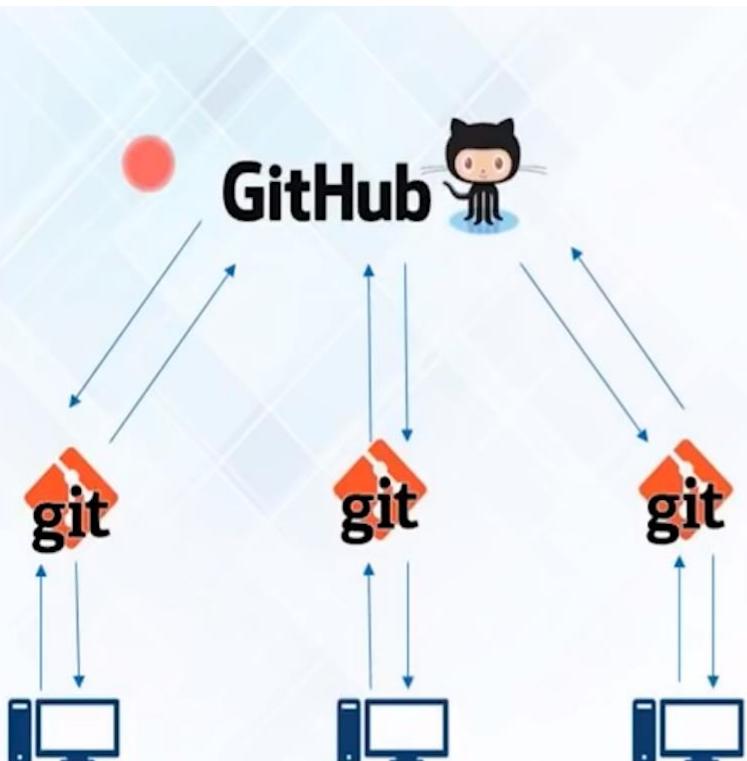
- Configuration Management is the practice of handling changes systematically so that a system maintains its integrity over time

- Configuration Management (CM) ensures that the current design and build state of the system is known, good & trusted

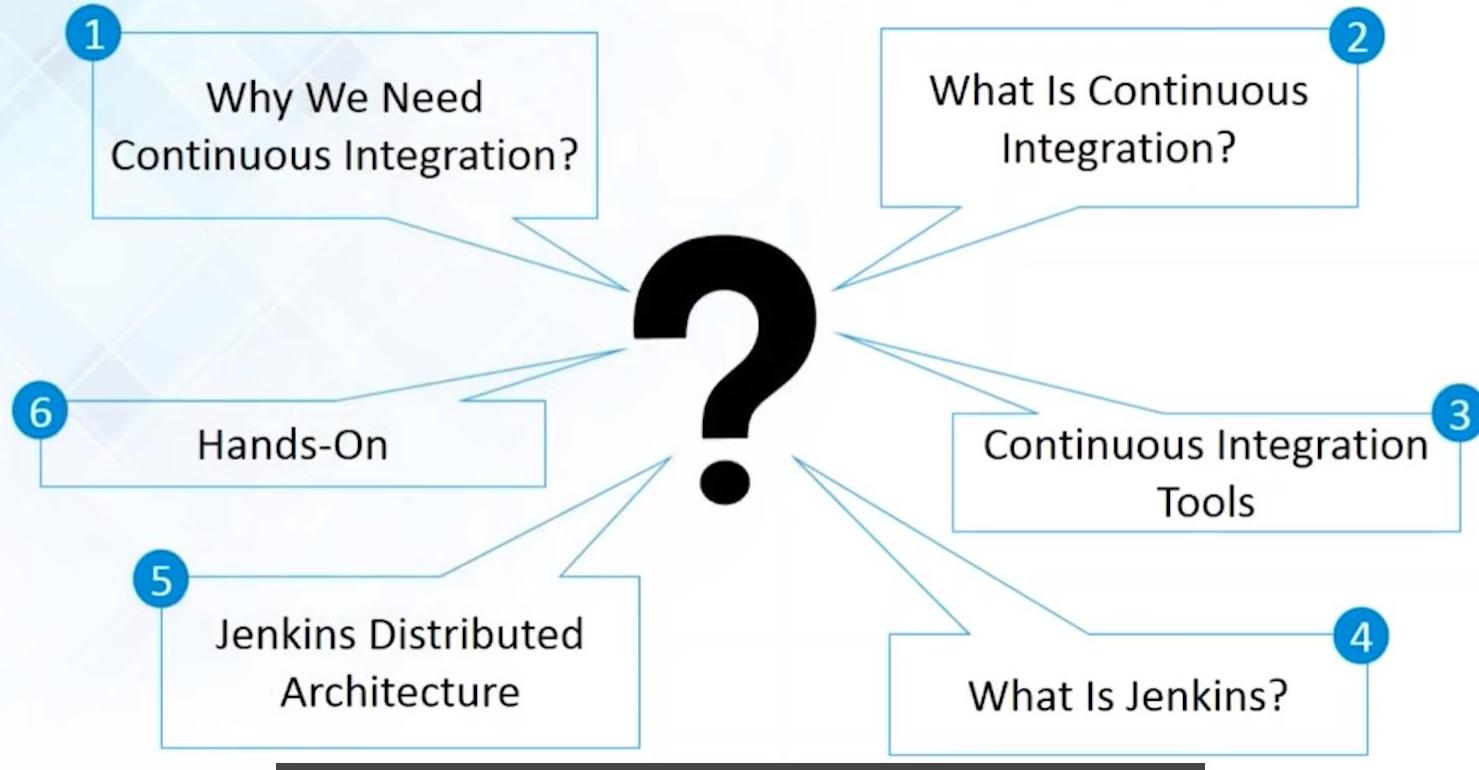
- It doesn't rely on the tacit knowledge of the development team



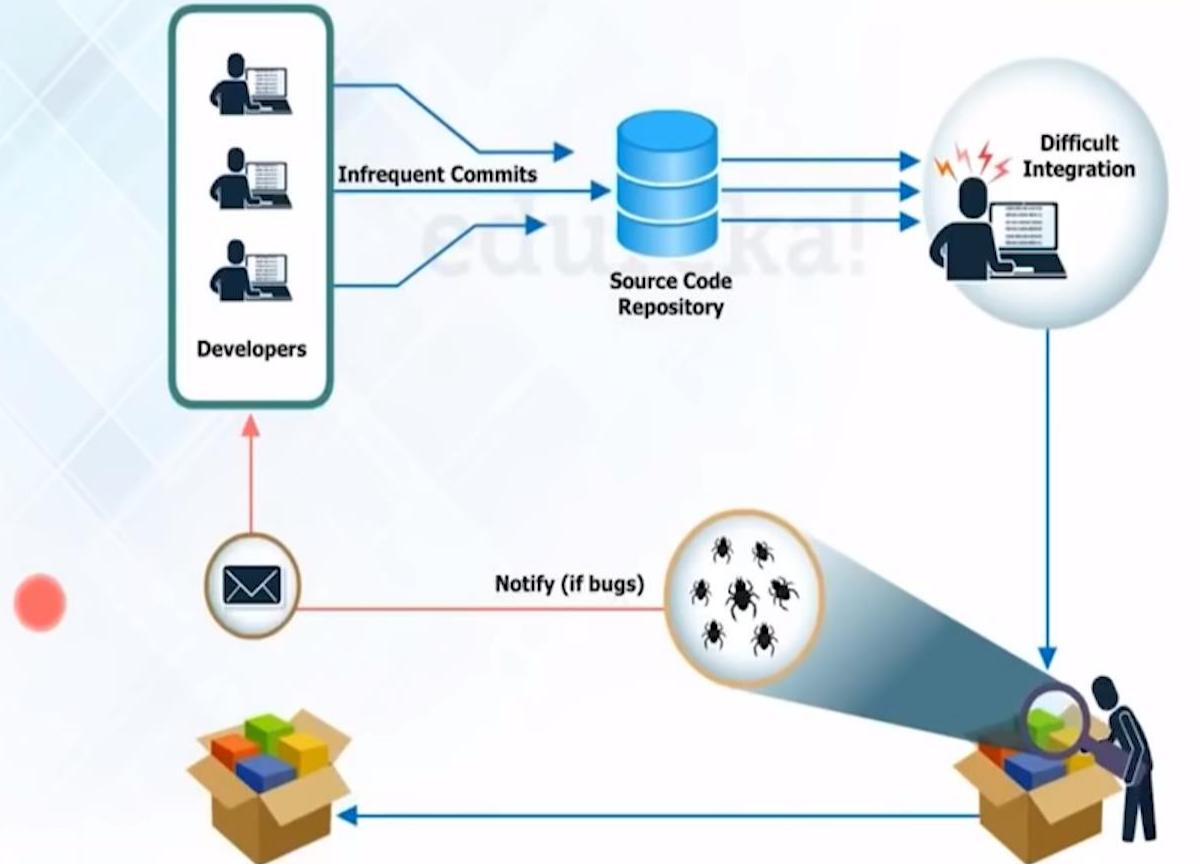
# Git vs Github



# Why Continuous Integration?



# Process Before Continuous Integration?



# Process Before Continuous Integration?

Developers have to wait till the complete software is developed for the test results.

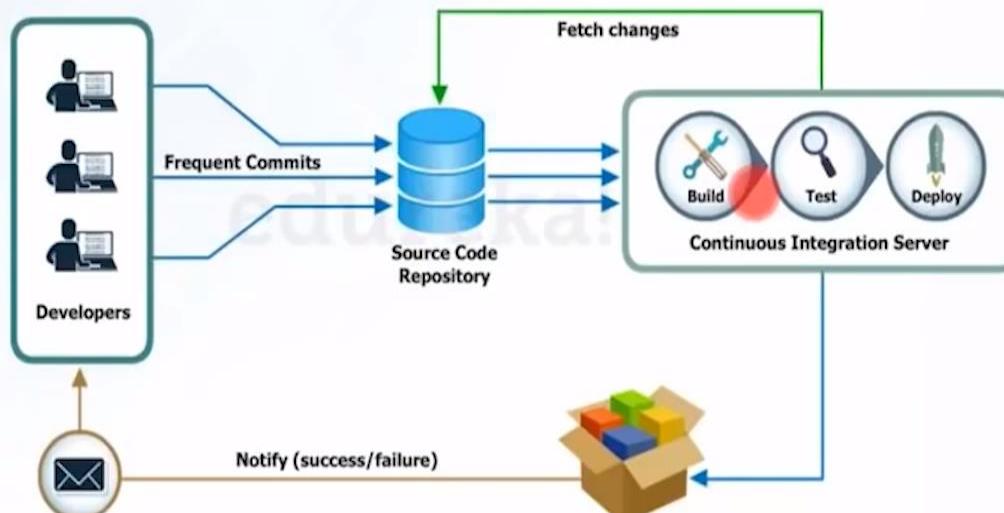


If the test fails then locating and fixing bugs is very difficult. Developers have to check the entire source code of the software.



# Continuous Integration To the Rescue

- ❑ Since after every commit to the source code an auto build is triggered and then it is automatically deployed on the test server
- ❑ If the test results shows that there is a bug in the code then the developers only have to check the last commit made to the source code
- ❑ This also increases the frequency of new software releases
- ❑ The concerned teams are always provided with the relevant feedback



# Continuous Integration To the Rescue

## Before Continuous Integration

The entire source code was built and then tested.

Developers have to wait for test results

No Feedback

## After Continuous Integration

Every commit made in the source code is built and tested.

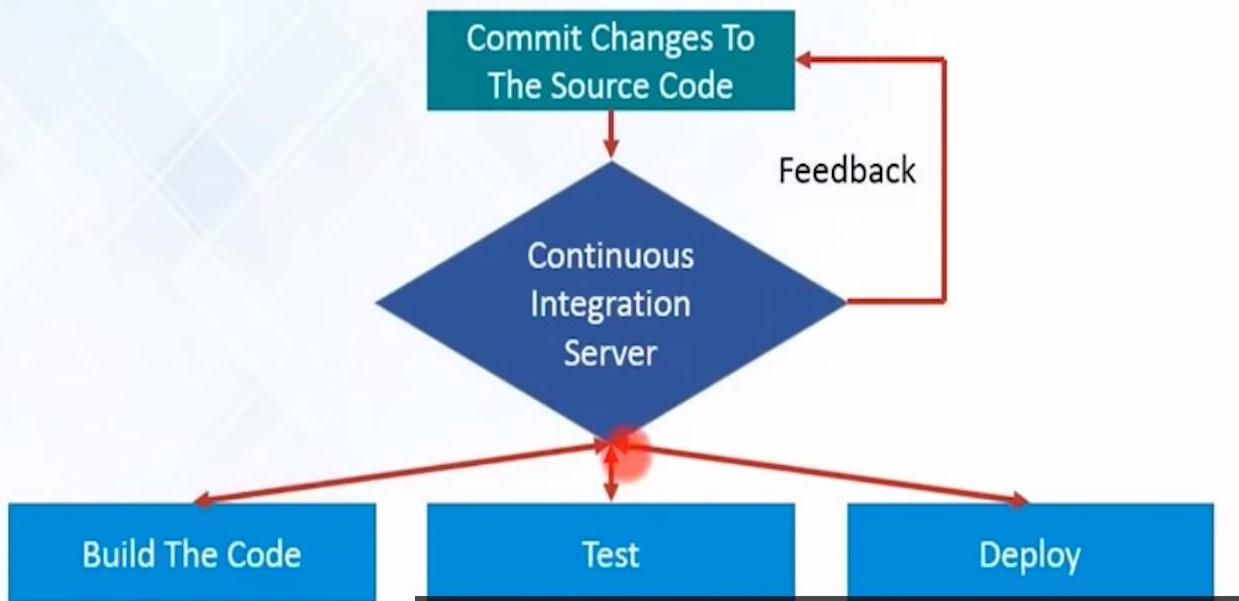
Developers know the test result of every commit made in the source code on the run

Feedback is present



# What is Continuous Integration

- ❑ Continuous Integration is a development practice in which the developers are required to commit changes to the source code in a shared repository several times a day or more frequently.
- ❑ Every commit made in the repository is then built. This allows the teams to detect the problems early.

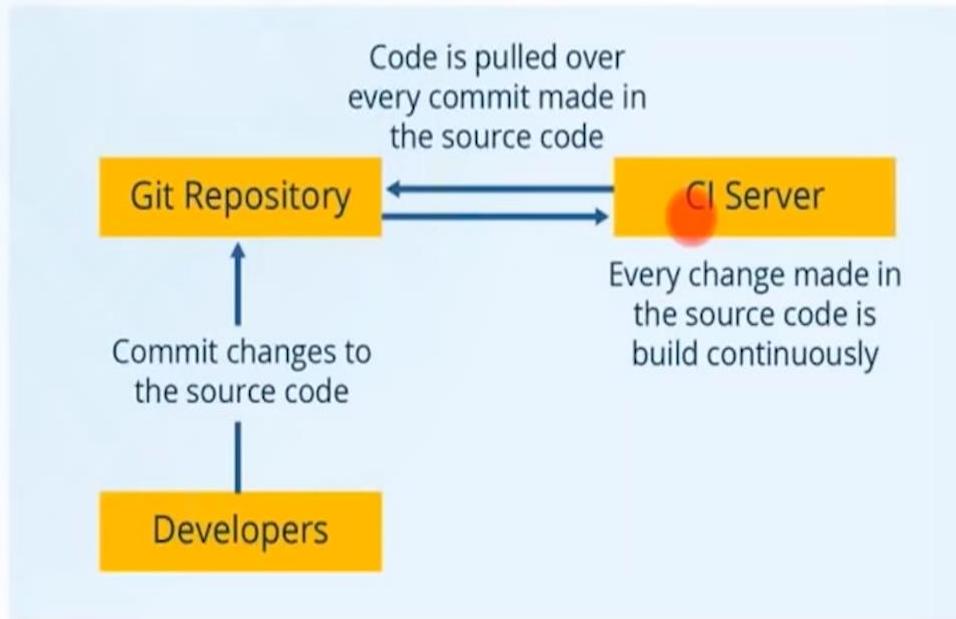


# What is Continuous Integration Case Study : Nokia

Solution:

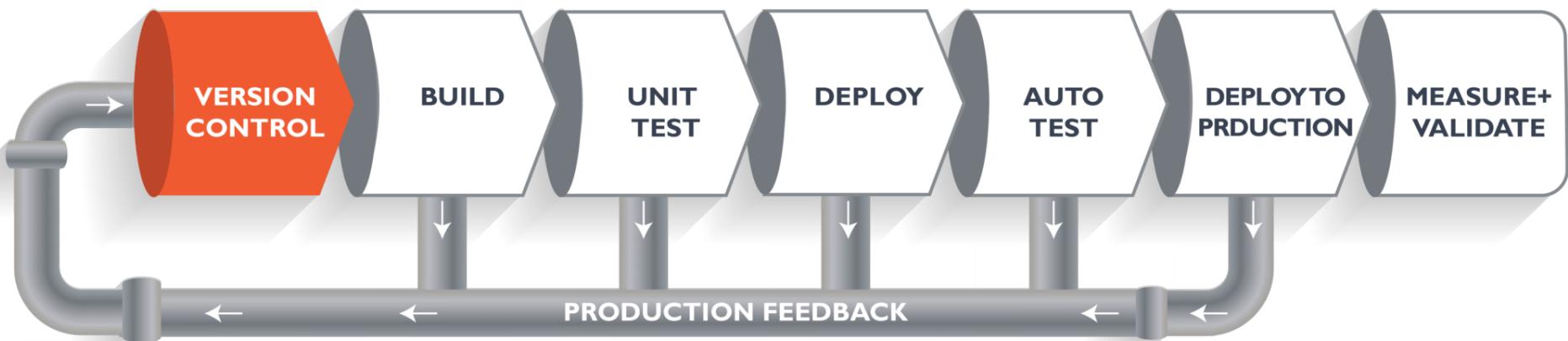


## Continuous Integration

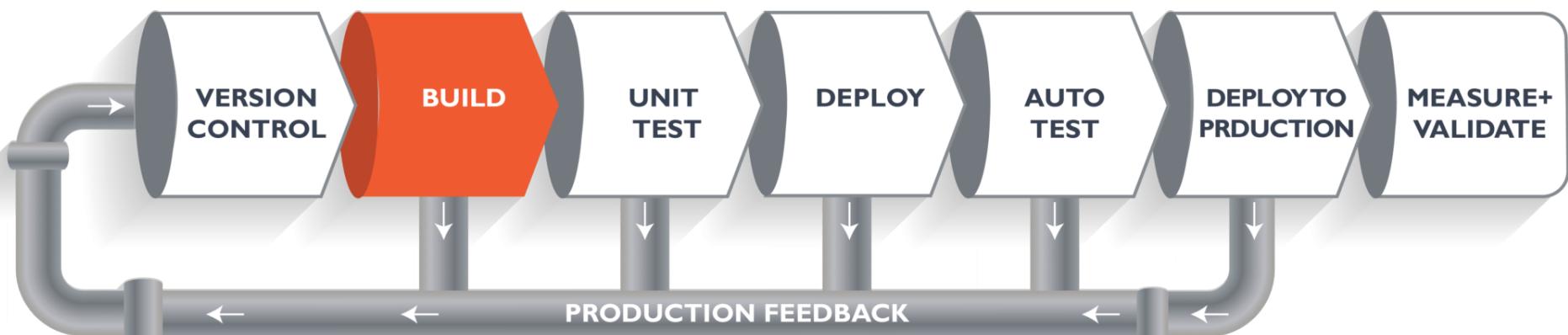


# CI/CD Pipeline

---

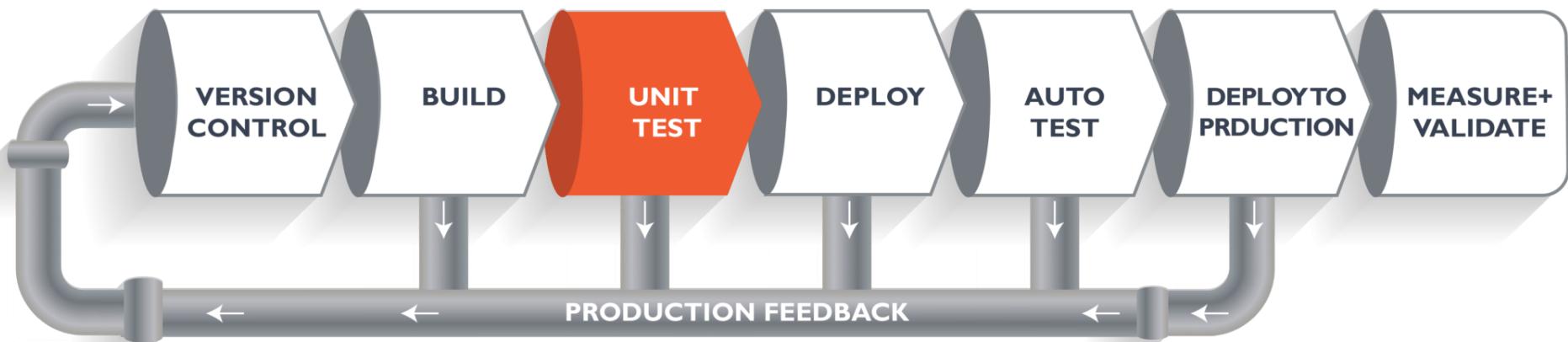


# CI/CD Pipeline

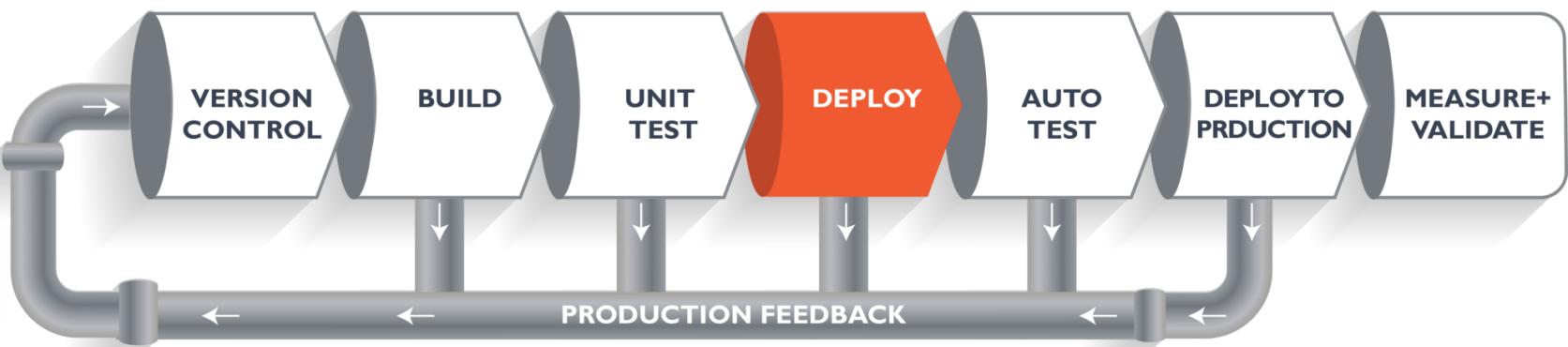


# CI/CD Pipeline

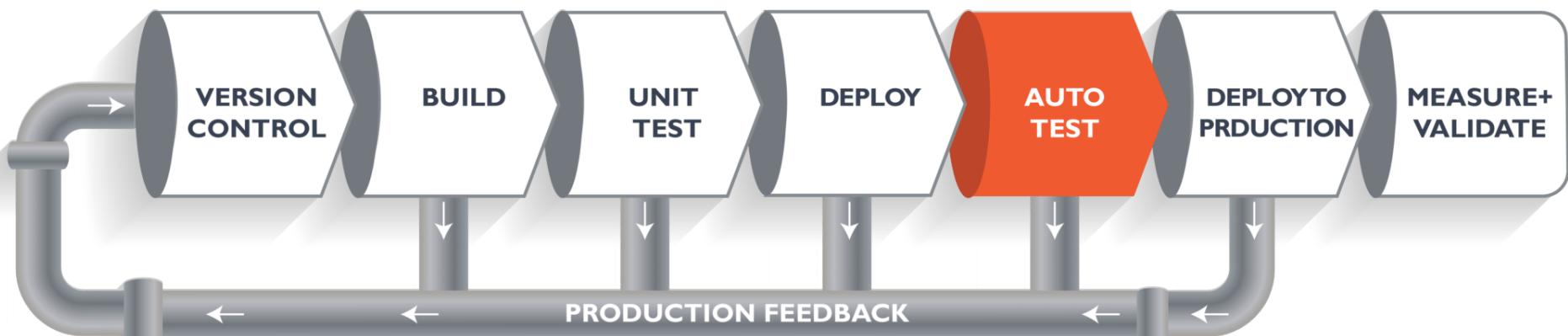
---



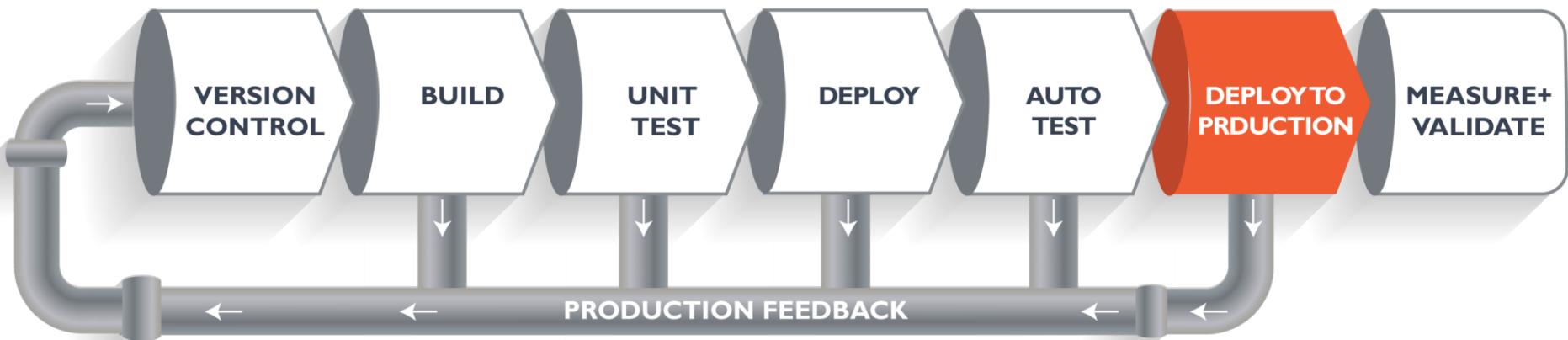
# CI/CD Pipeline



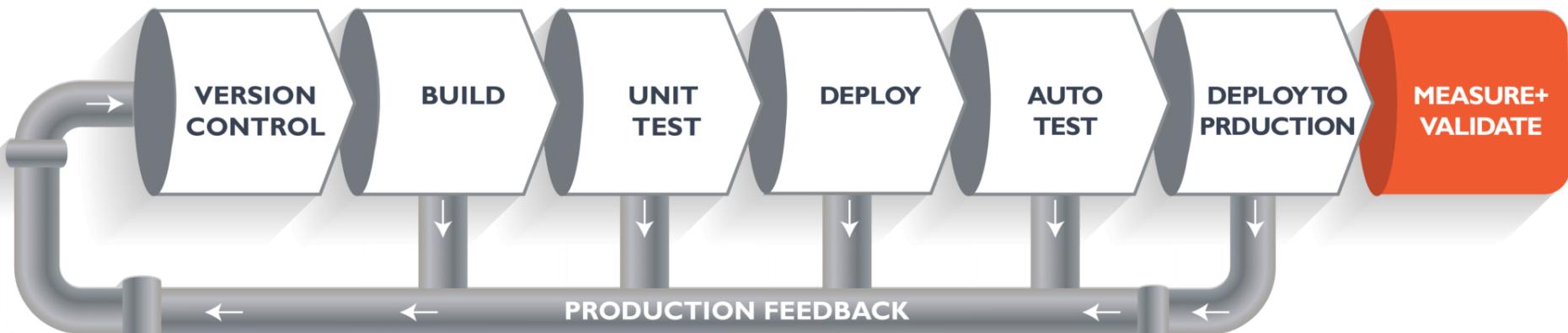
# CI/CD Pipeline



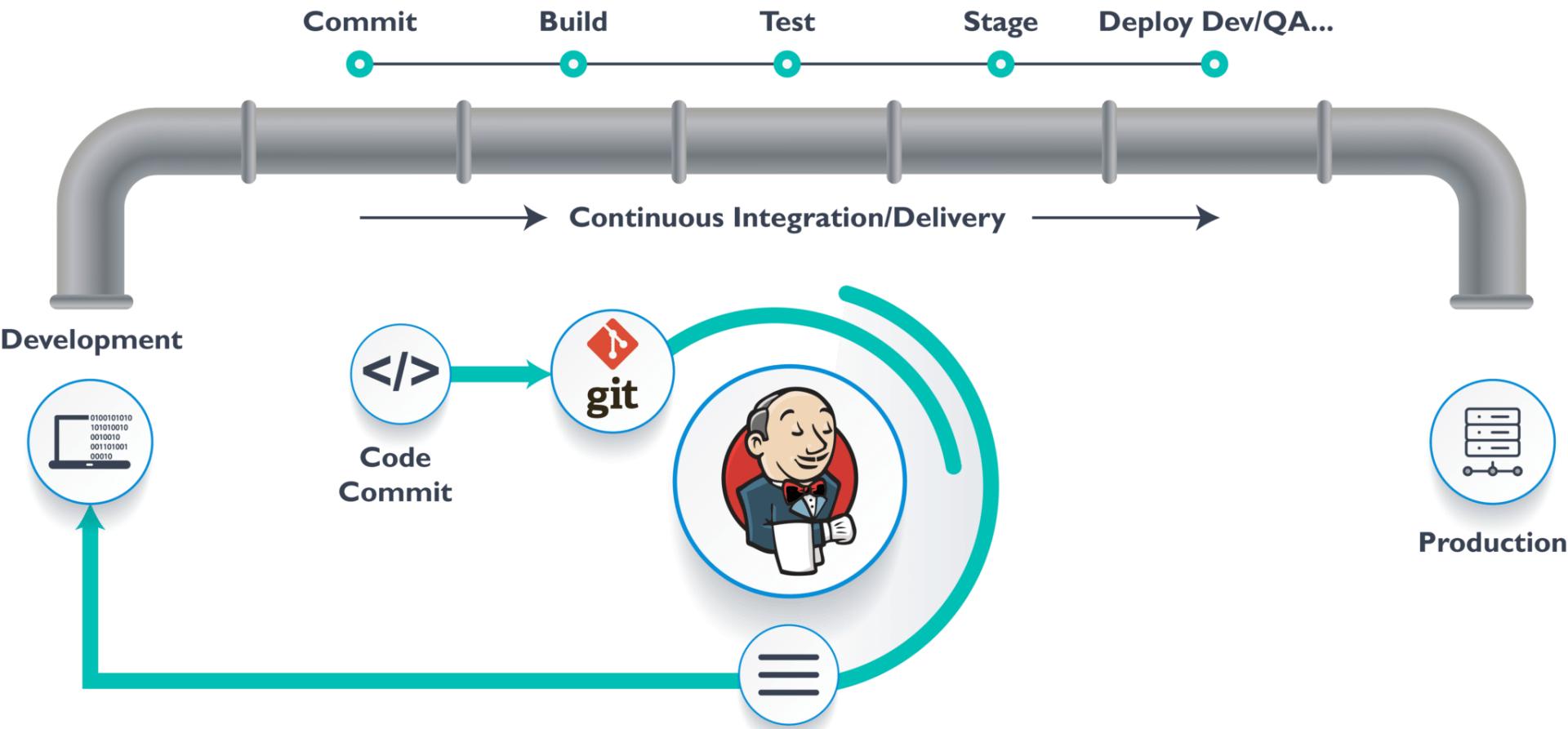
# CI/CD Pipeline



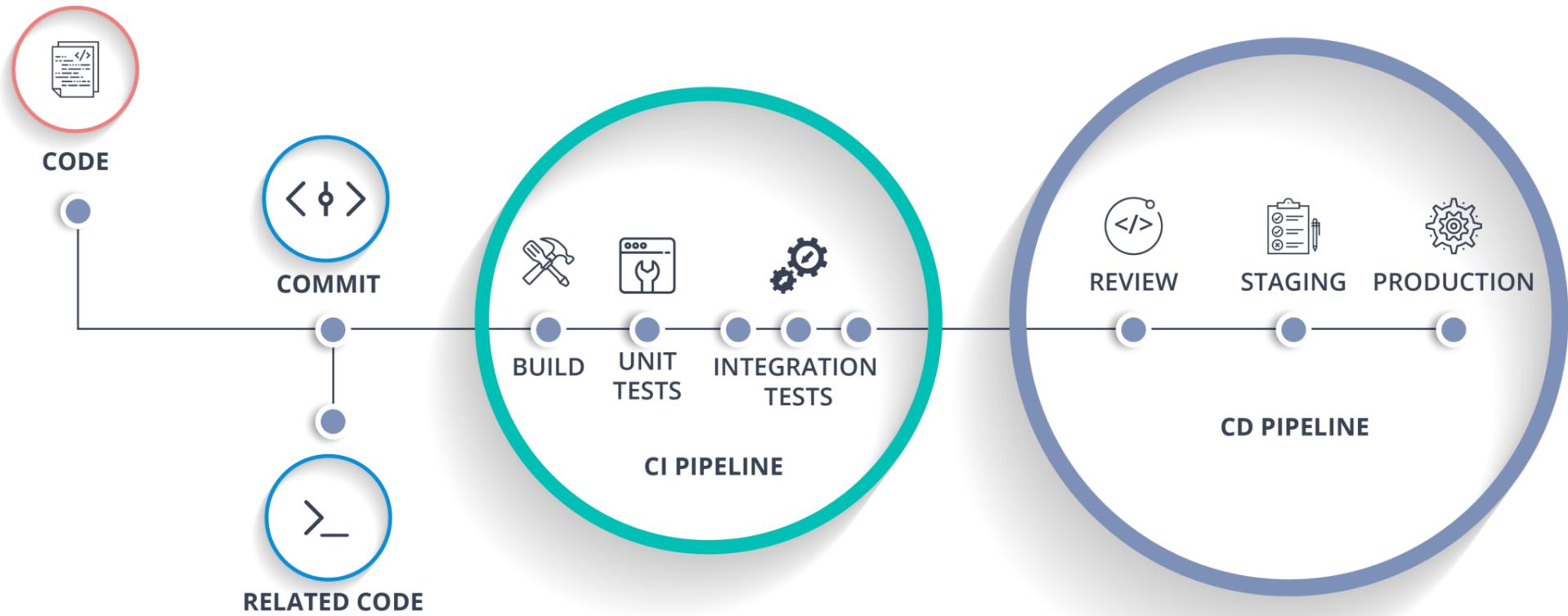
# CI/CD Pipeline



# CI/CD Pipeline



# CI/CD Pipeline



# Continuous Integration Tools

---



**Jenkins**



**Buildbot**

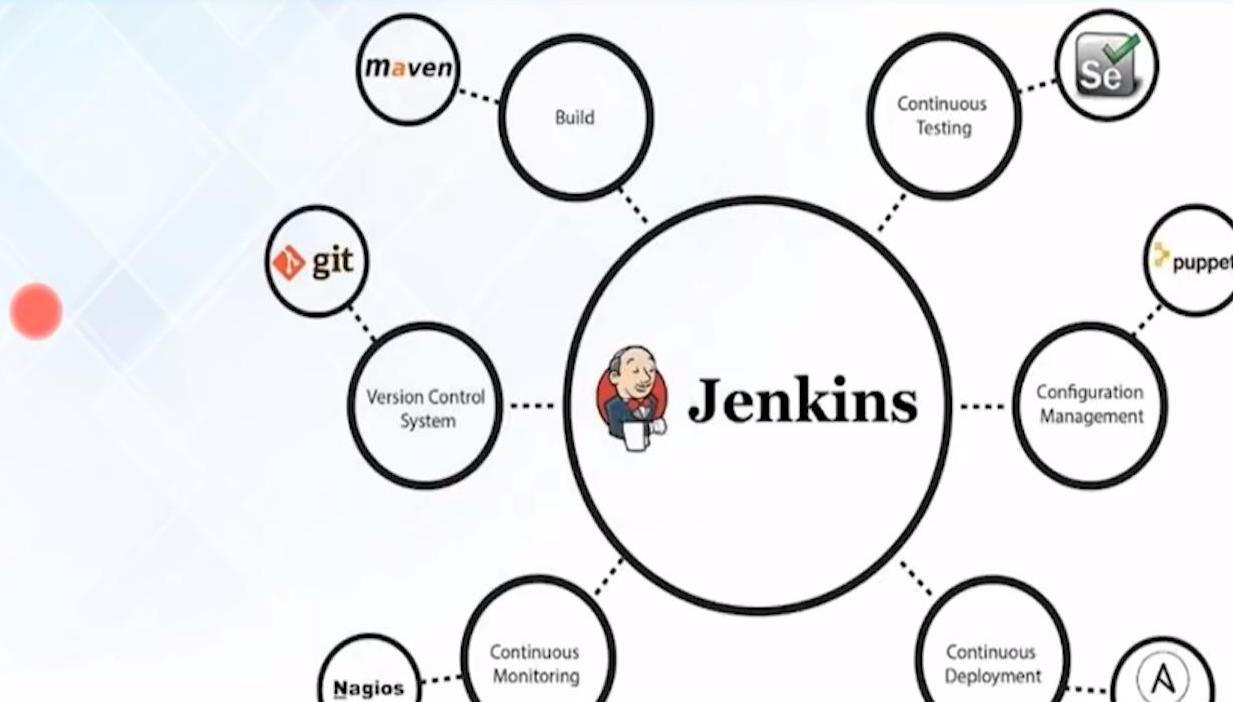


**Travis CI**



# What is Jenkins

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Plugins allows integration of various DevOps stages.



# Shortcomings of Single Jenkins Server



If you need to run web tests using Internet Explorer, you need to use a Windows machine.

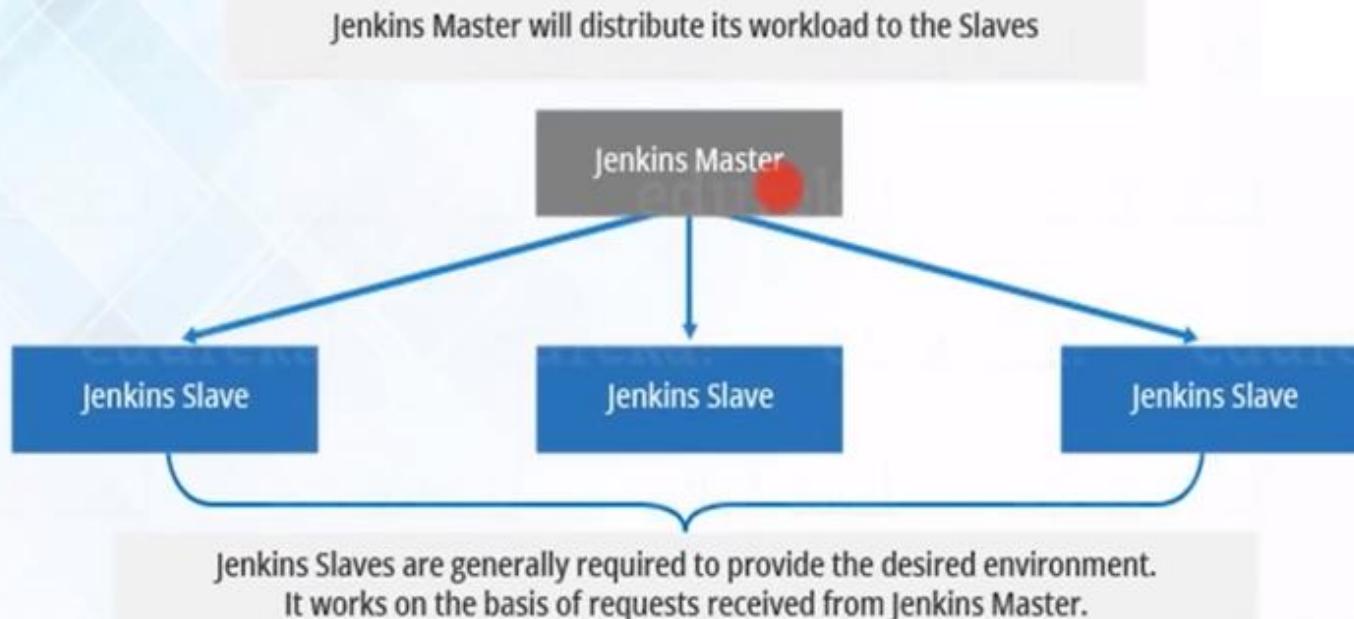


Another build job may require Linux box.

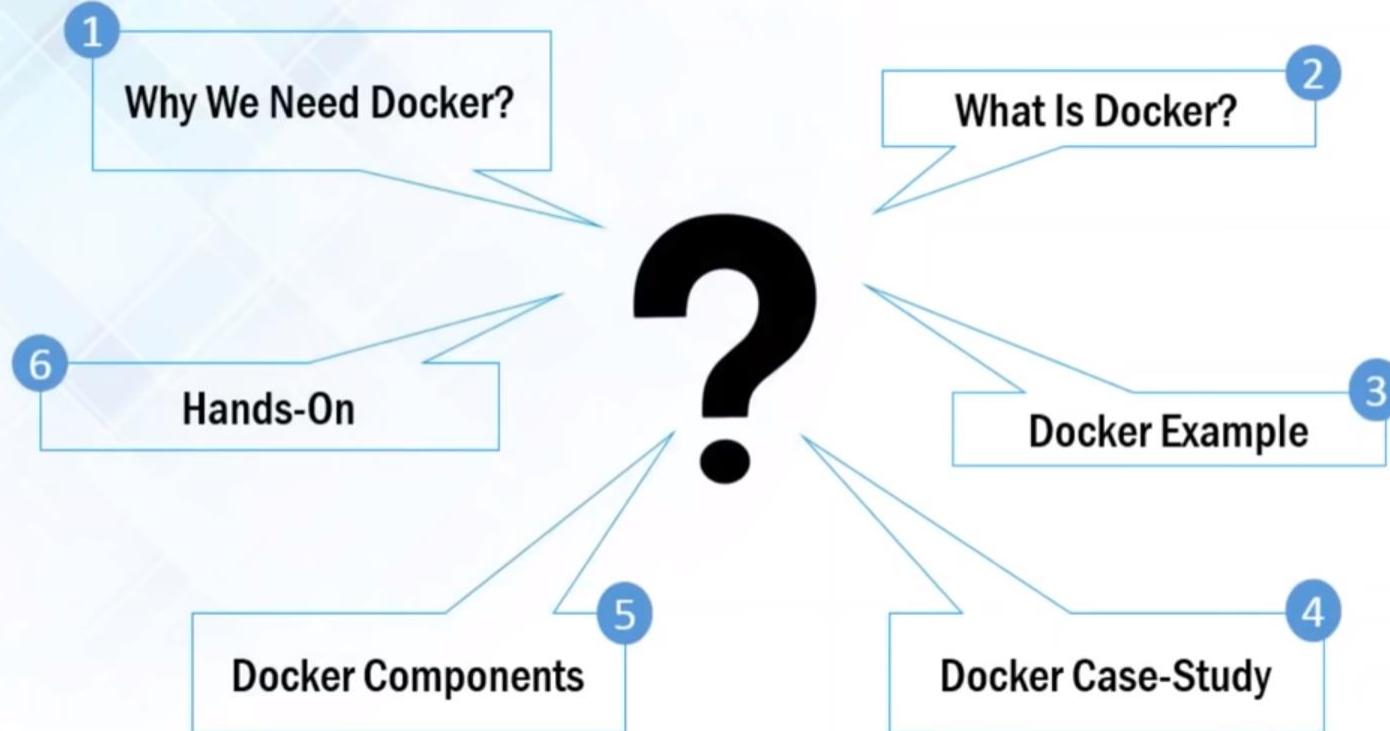


How to manage spikes in build activity

# Jenkins Distributed Architecture

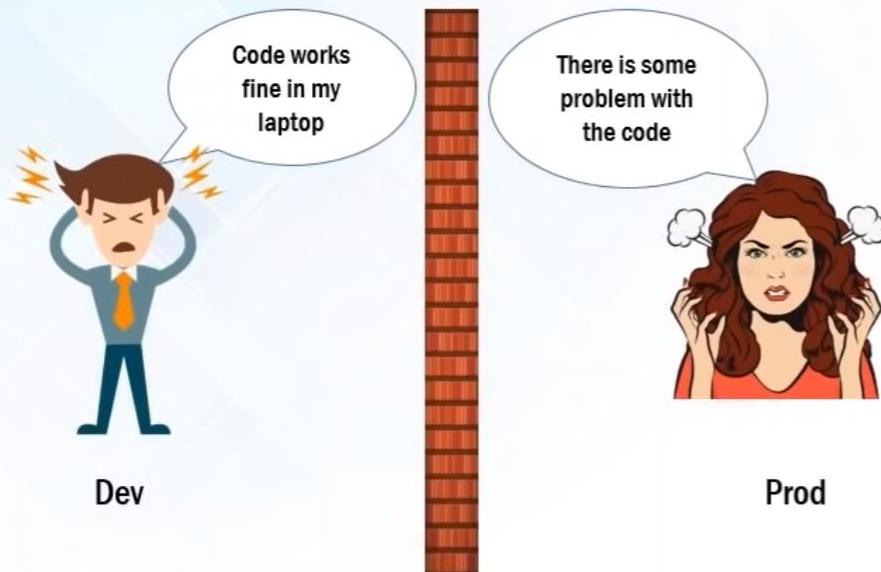


# Do you know the following?



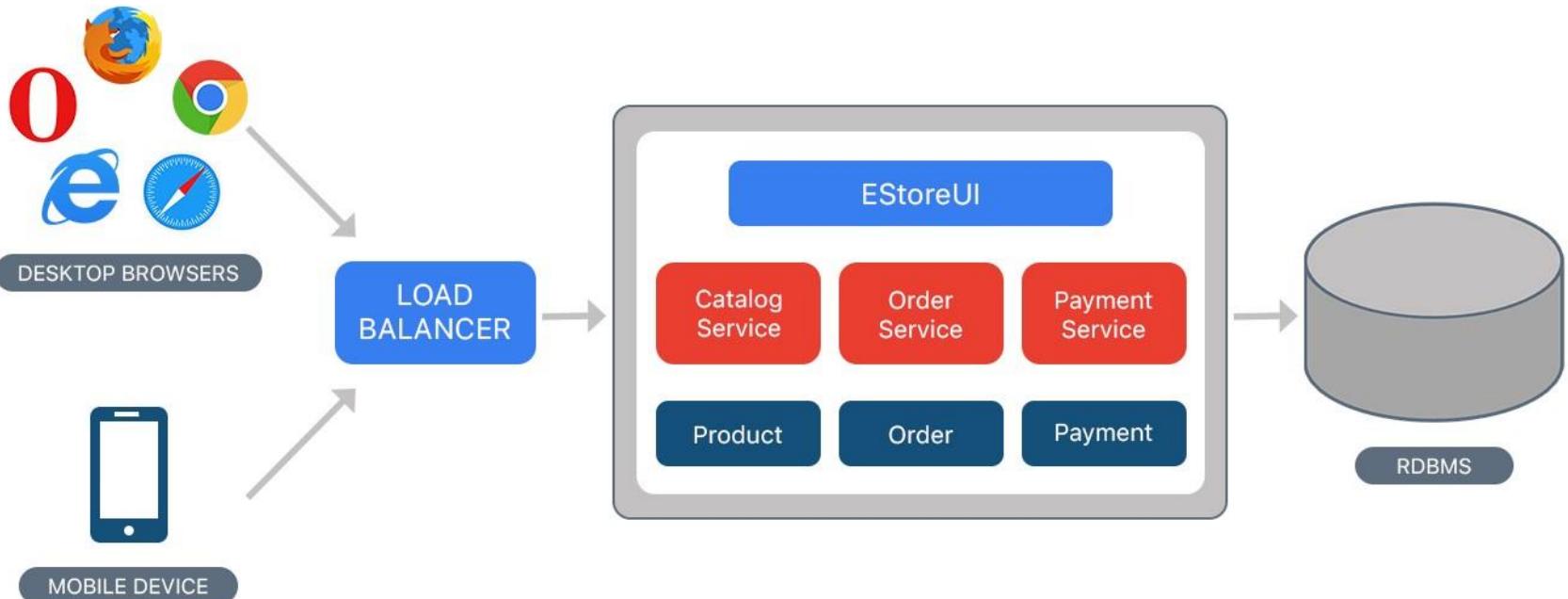
# Problems Before Docker

An application works in developer's laptop but not in testing or production. This is due to difference in computing environment between Dev, Test and Prod.

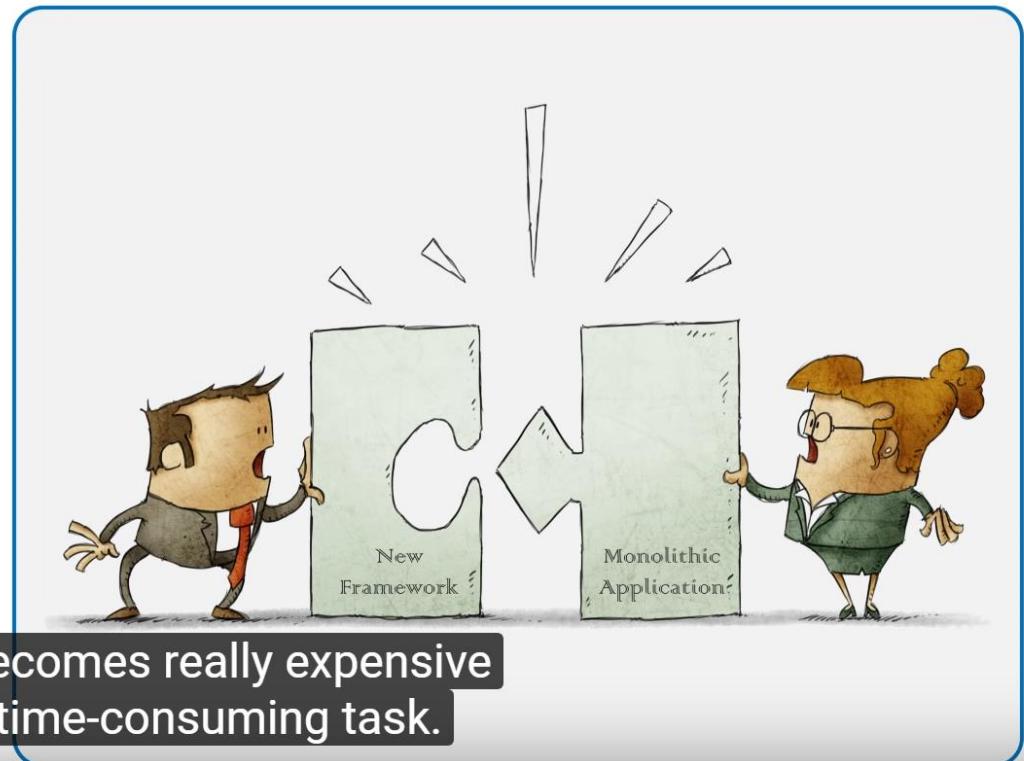


In Dev there can be a software that is upgraded and in Prod the old version of software might be present

# Example of Monolithic Approach

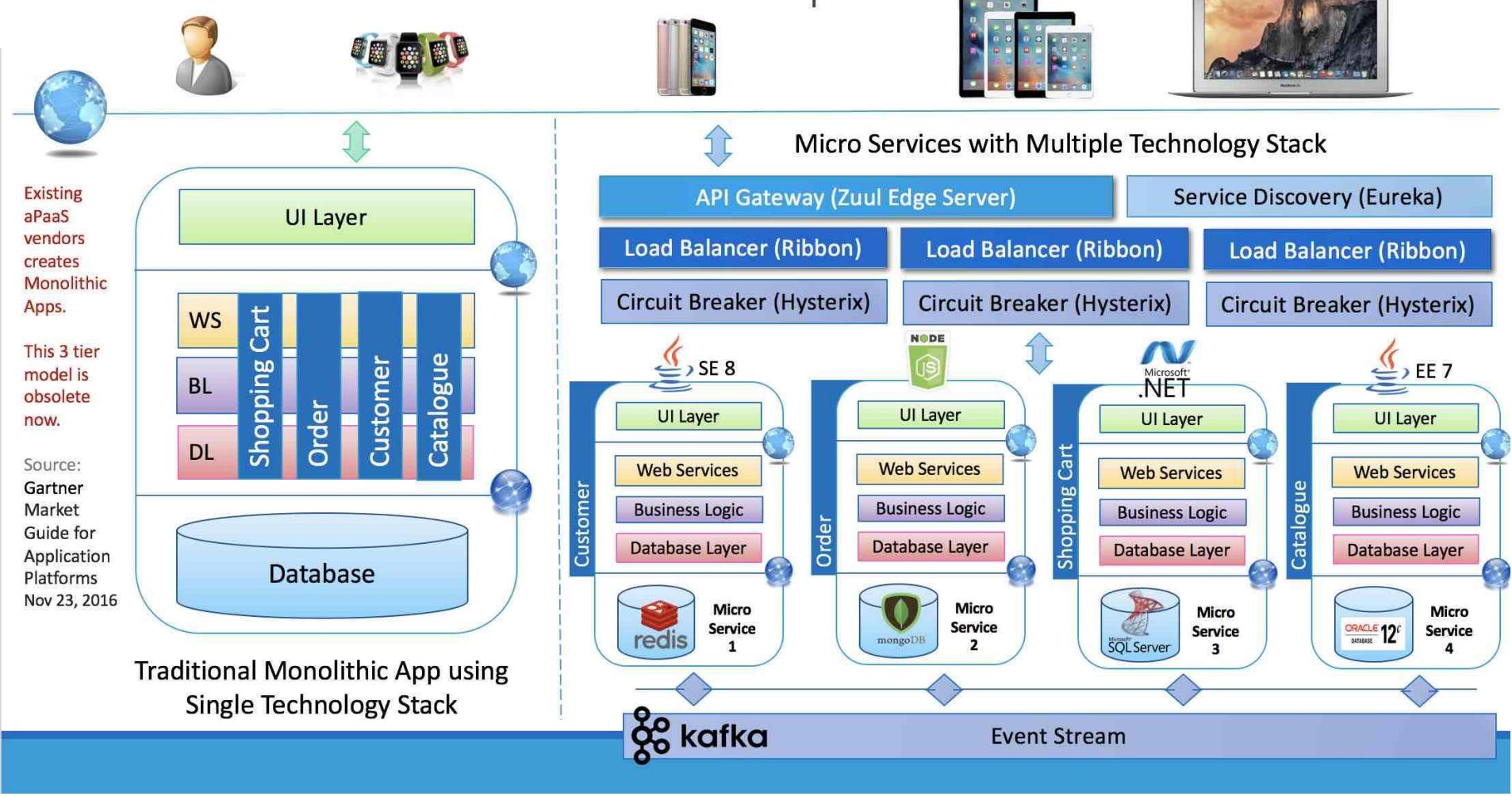


# Monolithic Architecture



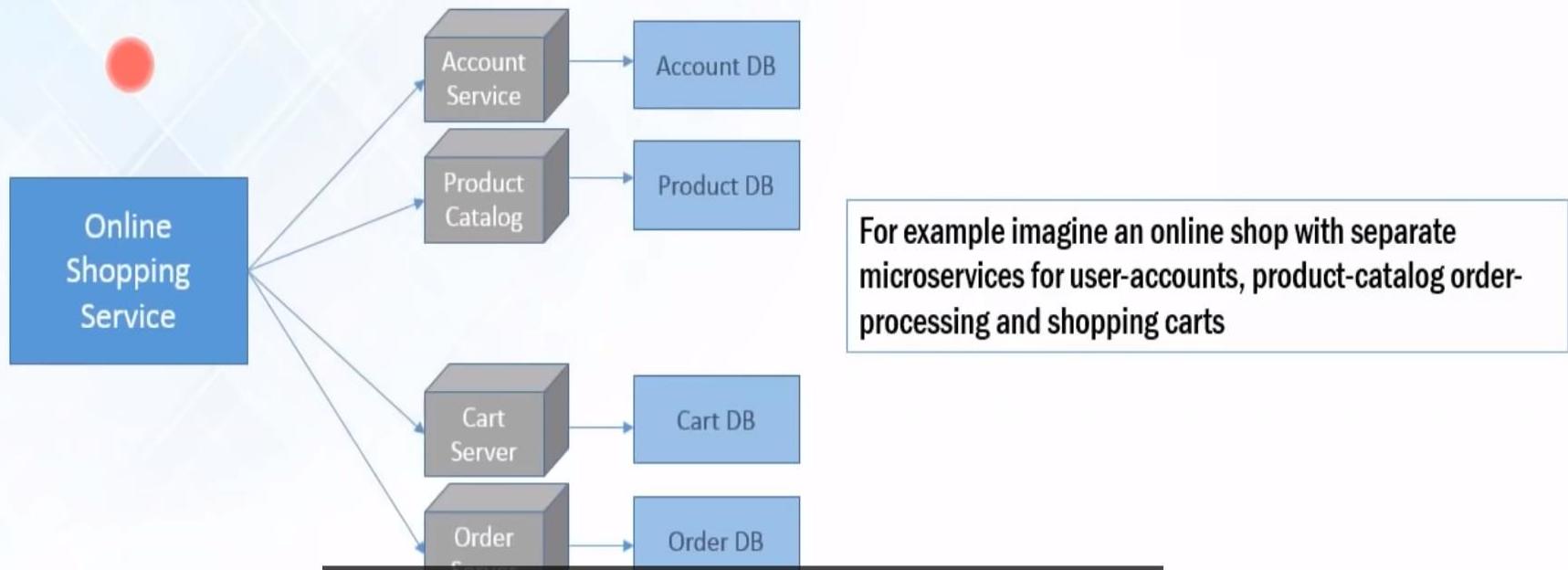
So it becomes really expensive  
and time-consuming task.

# Monolithic vs. Micro Services Example



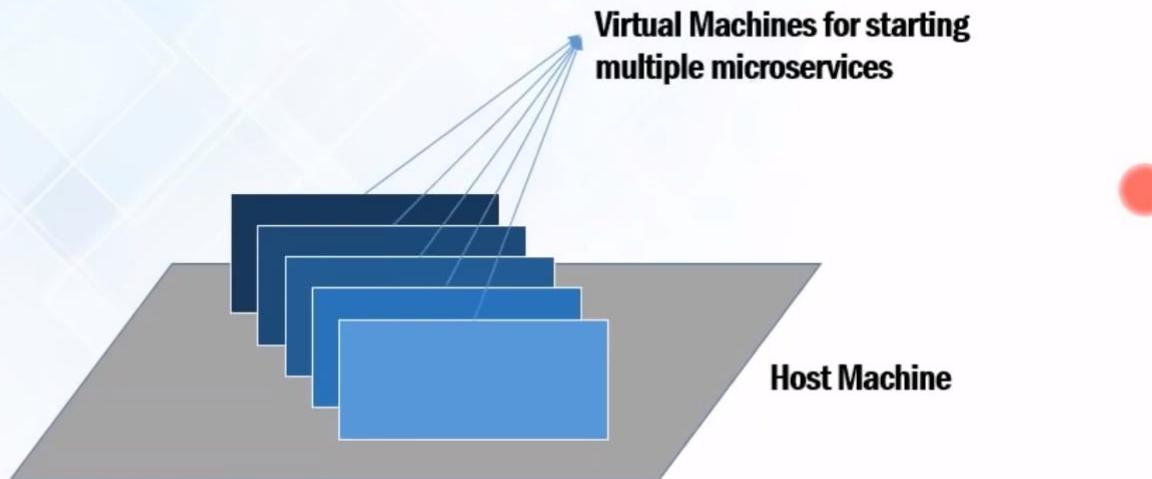
# Problems Before Docker

The idea behind microservices is that some types of applications become easier to build and maintain when they are broken down into smaller, composable pieces which work together. Each component is developed separately, and the application is then simply the sum of its constituent components.



# Problems Before Docker

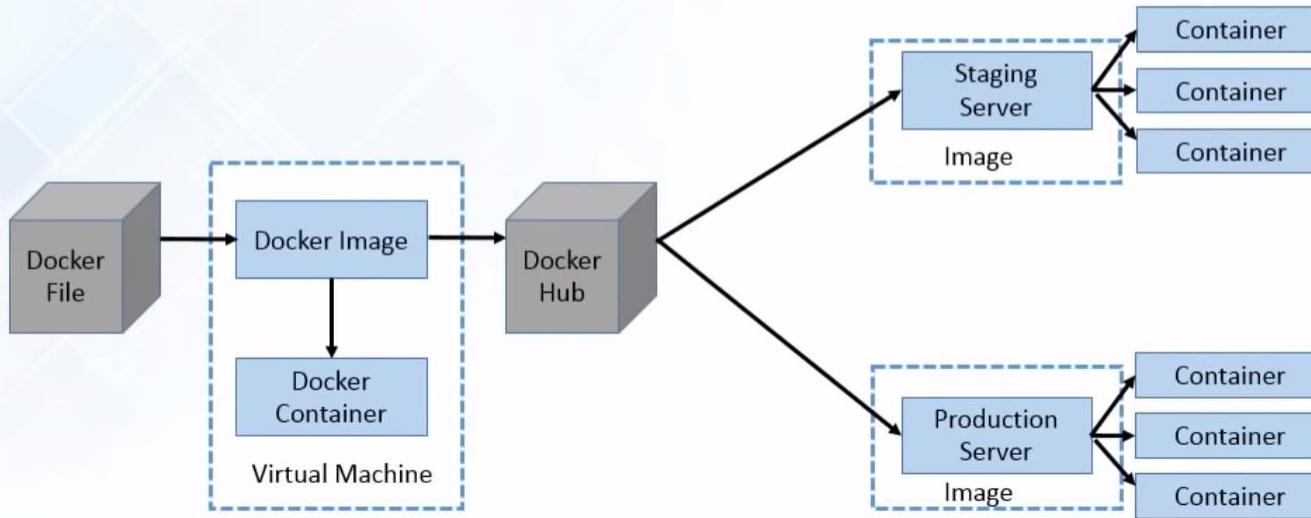
Developing an application requires starting several of microservices in one machine. So if you are starting five of those services you require five VMs on that machine.



# Docker in a Nutshell

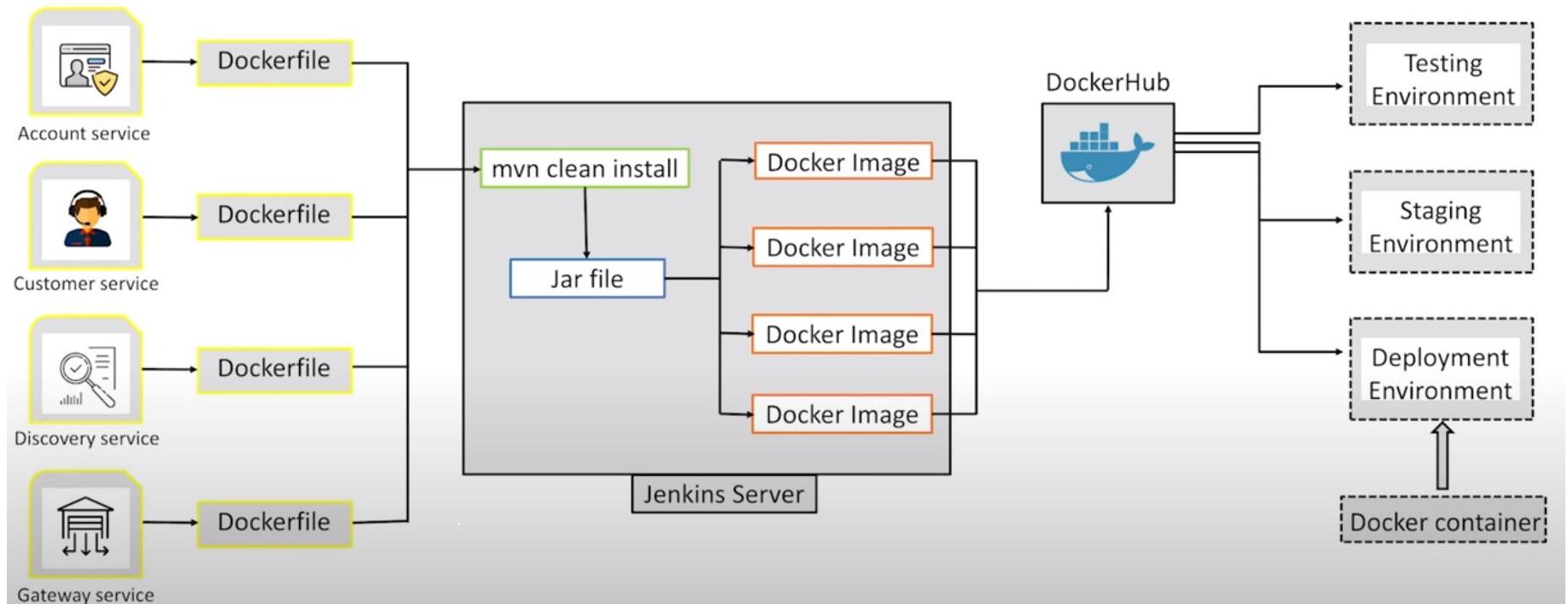


- Docker file builds a Docker image and that image contains all the project's code
- You can run that image to create as many Docker containers as you want
- Then this Image can be uploaded on Docker hub, from Docker hub any one can pull the image and build a container



# Continuous delivery through DevOps Integration

## Solution



## Docker jenkins

---

- docker jenkins
- docker pull jenkins/jenkins
- docker run -p 8080:8080 --name=jenkins-master jenkins/jenkins
- docker stop jenkins-master
- docker rm jenkins-master
- docker run -p 8080:8080 --name=jenkins-master -d --env JAVA\_OPTS="-Xmx8192m" --env JENKINS\_OPTS="--handlerCountMax=300" jenkins/jenkins

# Docker jenkins

```
C:\WINDOWS\system32>docker run -p 8080:8080 --name=jenkins-master jenkins/jenkins
docker: Error response from daemon: Conflict. The container name "/jenkins-master" is already in use by container "f43c2222247e094a75a6aca8b1a9a4544baf7145dc299d49ca0d6f2e55361ef5". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
```

```
C:\WINDOWS\system32>docker container ls -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
f43c2222247e        jenkins/jenkins      "/sbin/tini -- /usr..."   18 minutes ago   Up 18 minutes      jenkins-master
e6f0ed404d77        barathece91/spring-cloud-sidecar-polygot-wstore-node-service:v2.3.0   "docker-entrypoint.s..."  27 hours ago    Exited (0) 26 hours ago
54dee9e0e76b        barathece91/spring-cloud-sidecar-polygot-zuul-proxy:v2.3.0      "java -jar app.jar"     27 hours ago    Exited (143) 26 hours ago
a85871df1cb4        barathece91/spring-cloud-sidecar-polygot-sidecar:v2.3.0       "java -jar app.jar"     27 hours ago    Exited (143) 26 hours ago
fa7736867ee7        barathece91/spring-cloud-sidecar-polygot-csstore-microservice:v2.3.0  "java -jar app.jar"     27 hours ago    Exited (143) 26 hours ago
5490e027ce4d        barathece91/spring-cloud-sidecar-polygot-msstore-microservice:v2.3.0  "java -jar app.jar"     27 hours ago    Exited (143) 26 hours ago
cd4d202cdf59        barathece91/spring-cloud-sidecar-polygot-eureka-server:v2.3.0     "java -jar app.jar"     27 hours ago    Exited (143) 26 hours ago
5b129b2c9196        barathece91/spring-cloud-sidecar-polygot-config-server:v2.3.0     "java -jar app.jar"     27 hours ago    Exited (143) 26 hours ago
e3e7a9df79a4        wurstmeister/zookeeper                                "/bin/sh -c '/usr/sb..."  9 days ago     Exited (137) 8 days ago
55265d9dbc70        wurstmeister/kafka                                  "start-kafka.sh"       9 days ago     Exited (1) 8 days ago
```

```
C:\WINDOWS\system32>docker container start f43c2222247e
f43c2222247e
```

```
C:\WINDOWS\system32>
```

# Jenkins using war

---

```
D:\Program Files\Jenkins>java -jar jenkins.war
Running from: D:\Program Files\Jenkins\Jenkins.war
webroot: $user.home/.jenkins
2020-11-08 13:02:57.113+0000 [id=1]      INFO  org.eclipse.jetty.util.log.Log#initialized: Logging initialized @1481ms to org.eclipse.jetty.util.log.JavaUtilLog
2020-11-08 13:02:57.419+0000 [id=1]      INFO  winstone.Logger#logInternal: Beginning extraction from war file
2020-11-08 13:03:05.177+0000 [id=1]      WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2020-11-08 13:03:05.306+0000 [id=1]      INFO  org.eclipse.jetty.server.Server#doStart: jetty-9.4.30.v20200611; built: 2020-06-11T12:34:51.929Z; git: 271836e4c1f4612f12b7bb13ef5a92a927634b0d; jvm 1.8.0_261-b12
2020-11-08 13:03:06.749+0000 [id=1]      INFO  o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
2020-11-08 13:03:06.845+0000 [id=1]      INFO  o.e.j.s.s.DefaultSessionIdManager#doStart: DefaultSessionIdManager workerName=node0
2020-11-08 13:03:06.846+0000 [id=1]      INFO  o.e.j.s.s.DefaultSessionIdManager#doStart: No SessionScavenger set, using defaults
2020-11-08 13:03:06.859+0000 [id=1]      INFO  o.e.j.server.session.HouseKeeper#startScavenging: node0 Scavenging every 600000ms
2020-11-08 13:03:07.499+0000 [id=1]      INFO  hudson.WebAppMain#contextInitialized: Jenkins home directory: C:\Users\Balasubramaniam\.jenkins found at: $user.home/.jenkins
2020-11-08 13:03:07.943+0000 [id=1]      INFO  o.e.j.s.handler.ContextHandler#doStart: Started w.@2c5d601e{Jenkins v2.249.1,/,file:///C:/Users/Balasubramaniam/.jenkins/war/,AVAILABLE}{C:\Users\Balasubramaniam\.jenkins\war}
2020-11-08 13:03:08.090+0000 [id=1]      INFO  o.e.j.server.AbstractConnector#doStart: Started ServerConnector@3e0e1046{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
2020-11-08 13:03:08.091+0000 [id=1]      INFO  org.eclipse.jetty.server.Server#doStart: Started @12462ms
2020-11-08 13:03:08.099+0000 [id=23]     INFO  winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2020-11-08 13:03:10.092+0000 [id=29]     INFO  jenkins.InitReactorRunner$1#onAttained: Started initialization
2020-11-08 13:04:02.247+0000 [id=28]     INFO  jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2020-11-08 13:04:15.389+0000 [id=37]     INFO  jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2020-11-08 13:04:15.404+0000 [id=32]     INFO  jenkins.InitReactorRunner$1#onAttained: Started all plugins
2020-11-08 13:04:19.082+0000 [id=40]     INFO  jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2020-11-08 13:04:19.155+0000 [id=43]     INFO  jenkins.InitReactorRunner$1#onAttained: System config loaded
2020-11-08 13:04:19.158+0000 [id=43]     INFO  jenkins.InitReactorRunner$1#onAttained: System config adapted
2020-11-08 13:04:19.698+0000 [id=42]     INFO  jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2020-11-08 13:04:19.699+0000 [id=32]     INFO  jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2020-11-08 13:04:19.791+0000 [id=62]     INFO  hudson.model.AsyncPeriodicWork$lambda$doRun$0: Started Download metadata
```

# Jenkins

Dashboard [Jenkins] x +

localhost:8080

search

?

1

Parameswari log out

## Jenkins

New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Lockable Resources

New View

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration
		NetabankingMavenProject	1 mo 8 days - #14	1 mo 9 days - #11	52 sec

Icon: S M L

Legend Atom feed for all Atom feed for failures Atom feed for just latest builds

Build Queue ^

# SonarQube

The screenshot shows a Windows File Explorer window titled "SonarQube" with the path "windows-x86-64" selected. The window displays a log file with the following content:

```
jvm 1 | 2020.09.30 09:04:16 INFO app[] [o.s.a.AppFileSystem] Cleaning or creating temp directory E:\software\A08\file\sonarqube-6.6\temp
jvm 1 | 2020.09.30 09:04:16 INFO app[] [o.s.a.es.EsSettings] Elasticsearch listening on /127.0.0.1:9001
jvm 1 | 2020.09.30 09:04:16 INFO app[] [o.s.a.p.ProcessLauncherImpl] Launch process[[key='es', ipcIndex=1, logFilenamePrefix=es]] from [E:\software\A08\file\sonarqube-6.6\elasticsearch]: C:\Program Files\Java\jdk1.8.0_152\jre\bin\java -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+AlwaysPreTouch -server -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -Djdk.io.permissionsUseCanonicalPath=true -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Dlog4j.skipJansi=true -Xms512m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -Delasticsearch -Des.path.home=E:\software\A08\file\sonarqube-6.6\elasticsearch -cp lib/* org.elasticsearch.bootstrap.Elasticsearch -Epath.conf=E:\software\A08\file\sonarqube-6.6\temp\conf\es
jvm 1 | 2020.09.30 09:04:16 INFO app[] [o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
jvm 1 | 2020.09.30 09:04:17 INFO app[] [o.e.p.PluginsService] no modules loaded
jvm 1 | 2020.09.30 09:04:17 INFO app[] [o.e.p.PluginsService] loaded plugin [org.elasticsearch.transport.Netty4Plugin]
jvm 1 | 2020.09.30 09:04:37 INFO app[] [o.s.a.SchedulerImpl] Process[es] is up
jvm 1 | 2020.09.30 09:04:37 INFO app[] [o.s.a.p.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2, logFilenamePrefix=web]] from [E:\software\A08\file\sonarqube-6.6]: C:\Program Files\Java\jdk1.8.0_152\jre\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=E:\software\A08\file\sonarqube-6.6\temp -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -cp ./lib/common/*;./lib/server/*;E:\software\A08\file\sonarqube-6.6\lib\jdbc\h2\h2-1.3.176.jar org.sonar.server.app.WebServer E:\software\A08\file\sonarqube-6.6\temp\sq-process7163985333166906569properties
jvm 1 | 2020.09.30 09:04:59 INFO app[] [o.s.a.SchedulerImpl] Process[web] is up
jvm 1 | 2020.09.30 09:04:59 INFO app[] [o.s.a.p.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFilenamePrefix=ce]] from [E:\software\A08\file\sonarqube-6.6]: C:\Program Files\Java\jdk1.8.0_152\jre\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=E:\software\A08\file\sonarqube-6.6\temp -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -cp ./lib/common/*;./lib/server/*;./lib/ce/*;E:\software\A08\file\sonarqube-6.6\lib\jdbc\h2\h2-1.3.176.jar org.sonar.ce.app.CeServer E:\software\A08\file\sonarqube-6.6\temp\sq-process2633312515576990903properties
jvm 1 | 2020.09.30 09:05:06 INFO app[] [o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | 2020.09.30 09:05:06 INFO app[] [o.s.a.SchedulerImpl] SonarQube is up
```

The log shows the startup of SonarQube components: Elasticsearch, Web Server, and Ce Server, along with the initialization of the SonarQube application itself.

# SonarQube Maven

sonar:sonar -  
Dsonar.host.url=http://localhost:9000 -  
Dsonar.login=b56042da35208eb303feb4b1ab  
afe76f9ae572ac

hsbcfresher2020 - netbanking/pom.xml - Spring Tool Suite 4

File Edit Navigate Search Project Run Design Window Help

Project Explorer    Overview Dependencies Dependency Hierarchy Effective POM pom.xml

netbanking/pom.xml

```
<!-- activation -->
<properties>
    <db.driverClassName>oracle.jdbc.driver.OracleDriver</db.driverClassName>
    <db.url>jdbc:oracle:thin:@localhost:1521:XE</db.url>
    <db.username>system</db.username>
    <db.password>vignesh</db.password>
```

Markers Properties Servers Data Source Explorer Snippets Console Git Repositories Git Staging

<terminated> netbanking (2) [Maven Build] C:\Program Files\Java\jdk1.8.0\_152\bin\javaw.exe (30-Sep-2020, 9:40:15 am)

```
[INFO] Sensor JaCoCoSensor [java]
[INFO] Sensor JaCoCoSensor [java] (done) | time=1ms
[INFO] Sensor SonarJavaXmlFileSensor [java]
[INFO] 1 source files to be analyzed
[INFO] Sensor SonarJavaXmlFileSensor [java] (done) | time=1539ms
[INFO] 1/1 source files have been analyzed
[INFO] Sensor XML Sensor [xml]
[INFO] Sensor XML Sensor [xml] (done) | time=1687ms
[INFO] Sensor Analyzer for "php.ini" files [php]
[INFO] Sensor Analyzer for "php.ini" files [php] (done) | time=4ms
[INFO] Sensor Zero Coverage Sensor
[INFO] Sensor Zero Coverage Sensor (done) | time=19ms
[INFO] Sensor CPD Block Indexer
[INFO] Sensor CPD Block Indexer (done) | time=17ms
[INFO] SCM provider for this project is: git
[INFO] 3 files to be analyzed
[INFO] 2/3 files analyzed
[WARNING] Missing blame information for the following files:
[WARNING] * pom.xml
[WARNING] This may lead to missing/broken features in SonarQube
[INFO] 1 file had no CPD blocks
[INFO] Calculating CPD for 0 files
[INFO] CPD calculation finished
```

522M of 919M

Type here to search

09:40 30/09/2020

# SonarQube

Jenkins Installation In Windows | Jenkins download and deployment | NetabankingMavenProject [Jenkins] | Integrating SonarQube and Jenkins | Projects

localhost:9000/projects?sort=-analysis\_date

Projects Issues Rules Quality Profiles Quality Gates Search for projects, sub-projects and files... Log in

**sonarqube** Projects Issues Rules Quality Profiles Quality Gates Search for projects, sub-projects and files... Log in

Perspective: Overall Status Sort by: Last analysis date 1 projects

**Filters**

**Quality Gate**

Status	Count
Passed	1
Warning	0
Failed	0

Last analysis: October 3, 2019, 3:09 PM

**day14springmvcjpasshopping** Passed

1 C Bugs 0 A Vulnerabilities 50 A Code Smells 0.0% Coverage 0.0% Duplications

XS 867 Java, XML

1 of 1 shown

**Reliability (Bug)**

Grade	Count
A	0
B and worse	1
C and worse	1
D and worse	0
E	0

**Security (Vulnerabilities)**

Grade	Count
A	1
B and worse	0
C and worse	0
D and worse	0
E	0

**Maintainability (Code Smells)**

Grade	Count
A	1
B and worse	0

Embedded database should be used for evaluation purpose only  
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA  
Version 6.6 (build 32724) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About

Type here to search

9:06 30/09/2020

# SonarQube Properties

---

- sonar.projectKey=netbanking
- sonar.projectName=C:/Users/Balasubramaniam/.jenkins/workspace/NetbankingMavenProject
- sonar.projectVersion=1.0
- sonar.login=admin
- sonar.password=admin
- sonar.exclusions=vendor/\*\*, storage/\*\*, resources/\*\*
- sonar.language=java
- sonar.sources=C:/Users/Balasubramaniam/.jenkins/workspace/\$JOB\_NAME
- sonar.java.binaries=C:/Users/Balasubramaniam/.jenkins/workspace/\$JOB\_NAME/netbanking/target/classes
- sonar.sourceEncoding=UTF-8

# Jenkins Build

localhost:8989/job/NetabankingMavenProject/1/console

Paused

Apps Empire New Tab How to use Assert... Browser Automatio... node.js - How can I... Freelancer-dev-810... Courses New Tab Airtel 4G Hotspot nt8F83

Jenkins > NetabankingMavenProject > #1

```
[INFO]
[JENKINS] Recording test results
[INFO]
[INFO] --- maven-jar-plugin:3.2.0:jar (default-jar) @ netbanking ---
[INFO] Building jar: C:\Users\Balasubramaniam\.jenkins\workspace\NetabankingMavenProject\Netbanking\target\netbanking-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.3.4.RELEASE:repackage (repackage) @ netbanking ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ netbanking ---
[INFO] Installing C:\Users\Balasubramaniam\.jenkins\workspace\NetabankingMavenProject\Netbanking\target\netbanking-0.0.1-SNAPSHOT.jar to C:\Users\Balasubramaniam\.m2\repository\com\hsbc\banking\netbanking\0.0.1-SNAPSHOT\netbanking-0.0.1-SNAPSHOT.jar
[INFO] Installing C:\Users\Balasubramaniam\.jenkins\workspace\NetabankingMavenProject\Netbanking\pom.xml to C:\Users\Balasubramaniam\.m2\repository\com\hsbc\banking\netbanking\0.0.1-SNAPSHOT\netbanking-0.0.1-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 22.561 s
[INFO] Finished at: 2020-09-30T08:48:32+05:30
[INFO] -----
[WARNING] The requested profile "test" could not be activated because it does not exist.
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving C:\Users\Balasubramaniam\.jenkins\workspace\NetabankingMavenProject\Netbanking\pom.xml to com.hsbc.bankng/netbanking/0.0.1-SNAPSHOT/netbanking-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving C:\Users\Balasubramaniam\.jenkins\workspace\NetabankingMavenProject\Netbanking\target\netbanking-0.0.1-SNAPSHOT.jar to com.hsbc.bankng/netbanking/0.0.1-SNAPSHOT/netbanking-0.0.1-SNAPSHOT.jar
channel stopped
Finished: SUCCESS
```

# Questions



# Module Summary

---

- In this module we discussed
  - Overview of Maven
  - Maven archetypes
  - Maven life cycle phases
  - The pom.xml file
  - Creation of Java projects using Maven
  - Creation of war files

