

# Application Delivery Fundamentals 2.0 B: Java

Introduction to Artifactory Devops  
and Jenkins

Parameswari Ettiappan



High performance. Delivered.

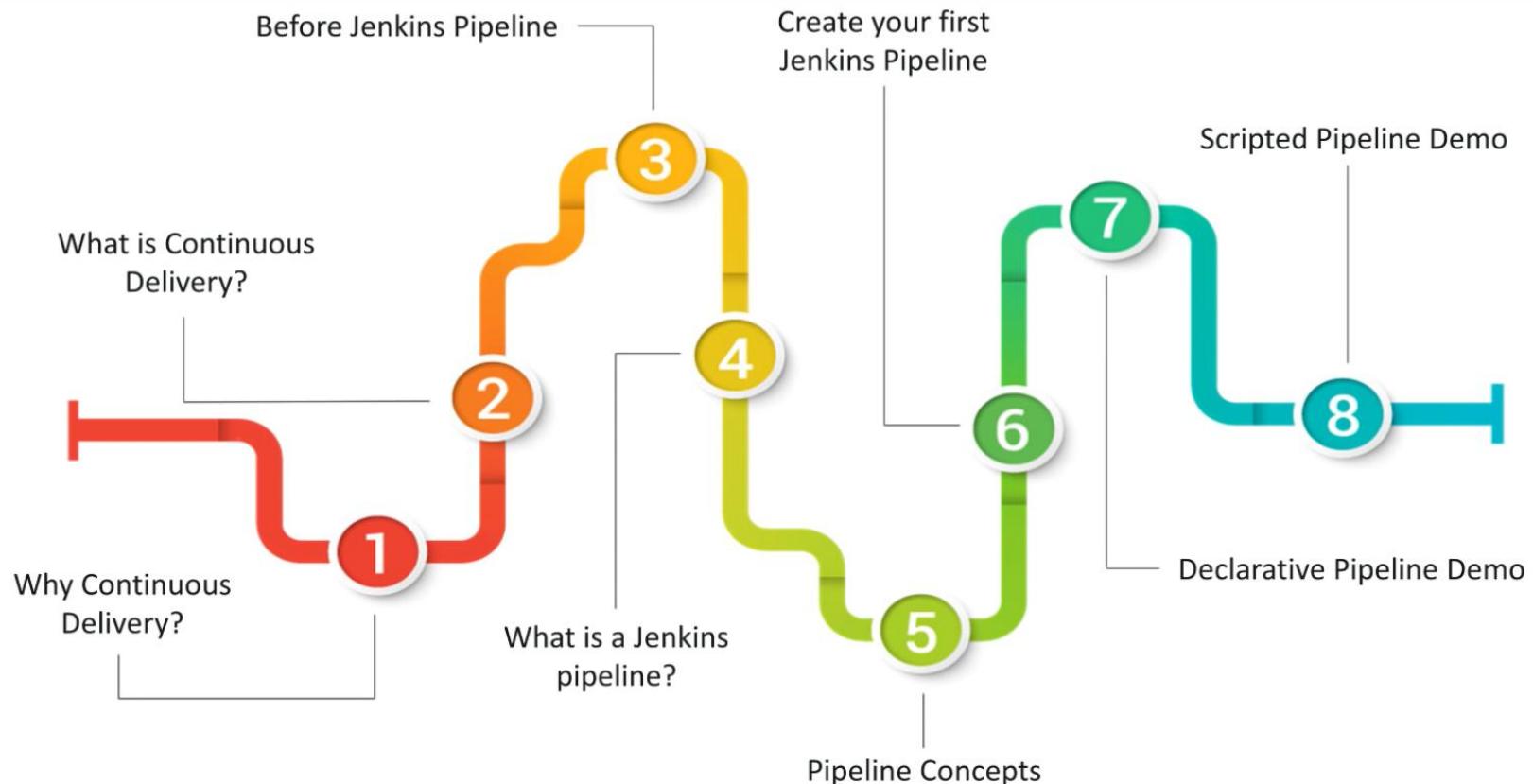
# Goals

---

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

# Goals

## What Will I Learn Today?



# Software Requirements

---

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

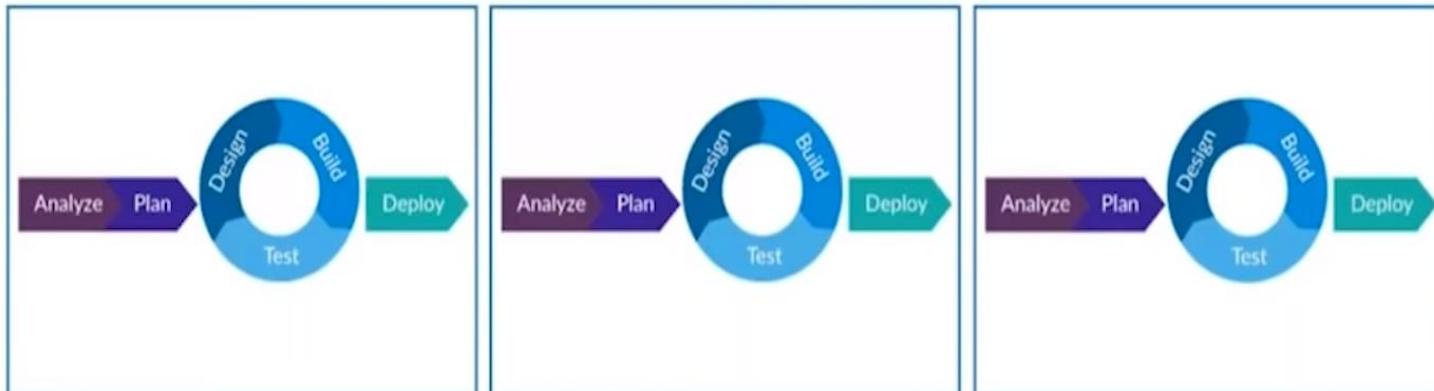
# Water Fall Model vs Agile

---

## Waterfall

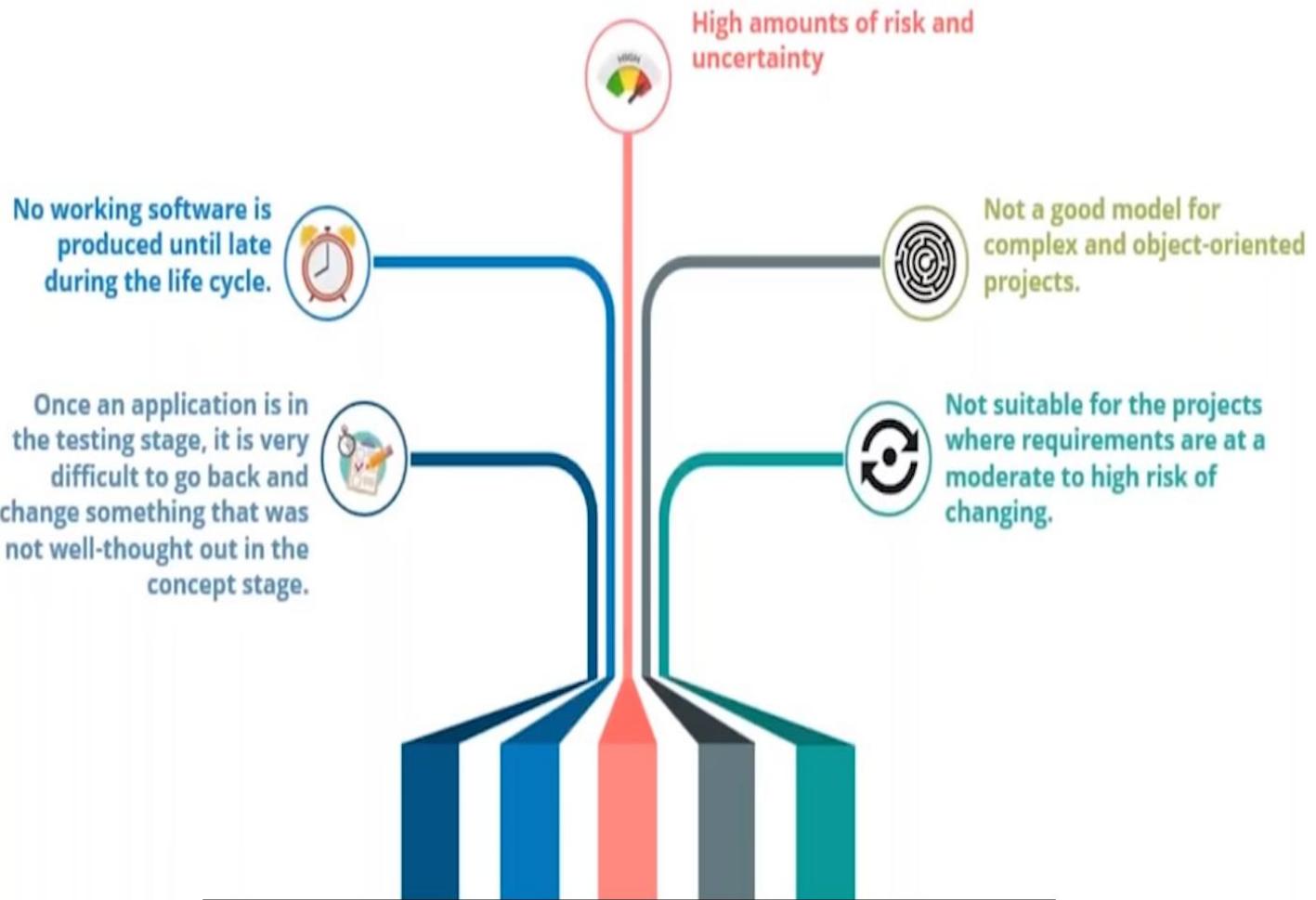


## Agile

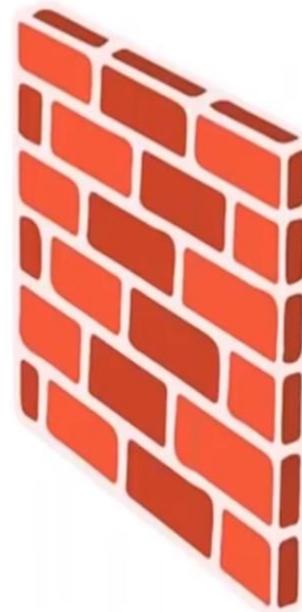


# Limitations of Waterfall Model

---



# Limitations of Agile



Wants Change

Wants Stability

# HP case study

---



# HP case study

*How did they achieve this?*

They implemented a CD pipeline  
with two key features



- Practice CI
- Automation at every step

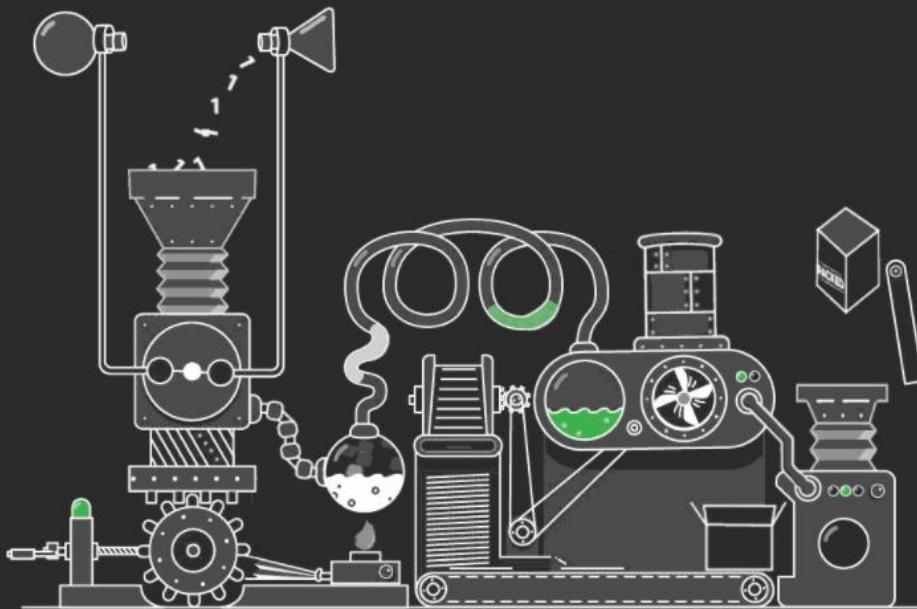


- Overall development cost reduced ~ 40%
- Programs under development increased ~ 140%
- Development cost per program went down ~ 78%

# Artifactory

---

JFrog Artifactory is getting ready to work...





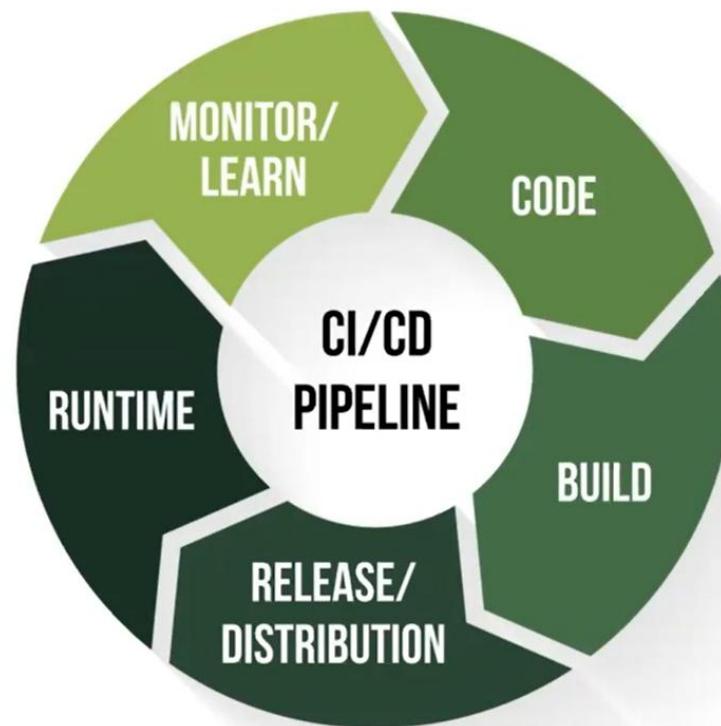
JFrog  
ARTIFACTORY

ARTIFACTORY ESSENTIALS

---

## WHAT IS ARTIFACTORY

## From Code to Release



## From Code to Release

```
1 package hello;
2
3 public class HelloWorld {
4
5     public static void main(String[] args) {
6         System.out.println("Hello World!");
7     }
8
9 }
```

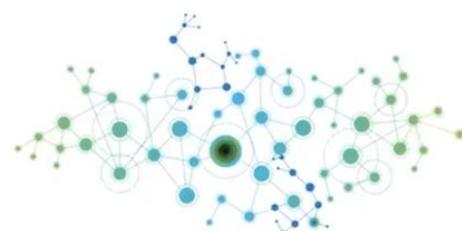


Source Code

Dependencies

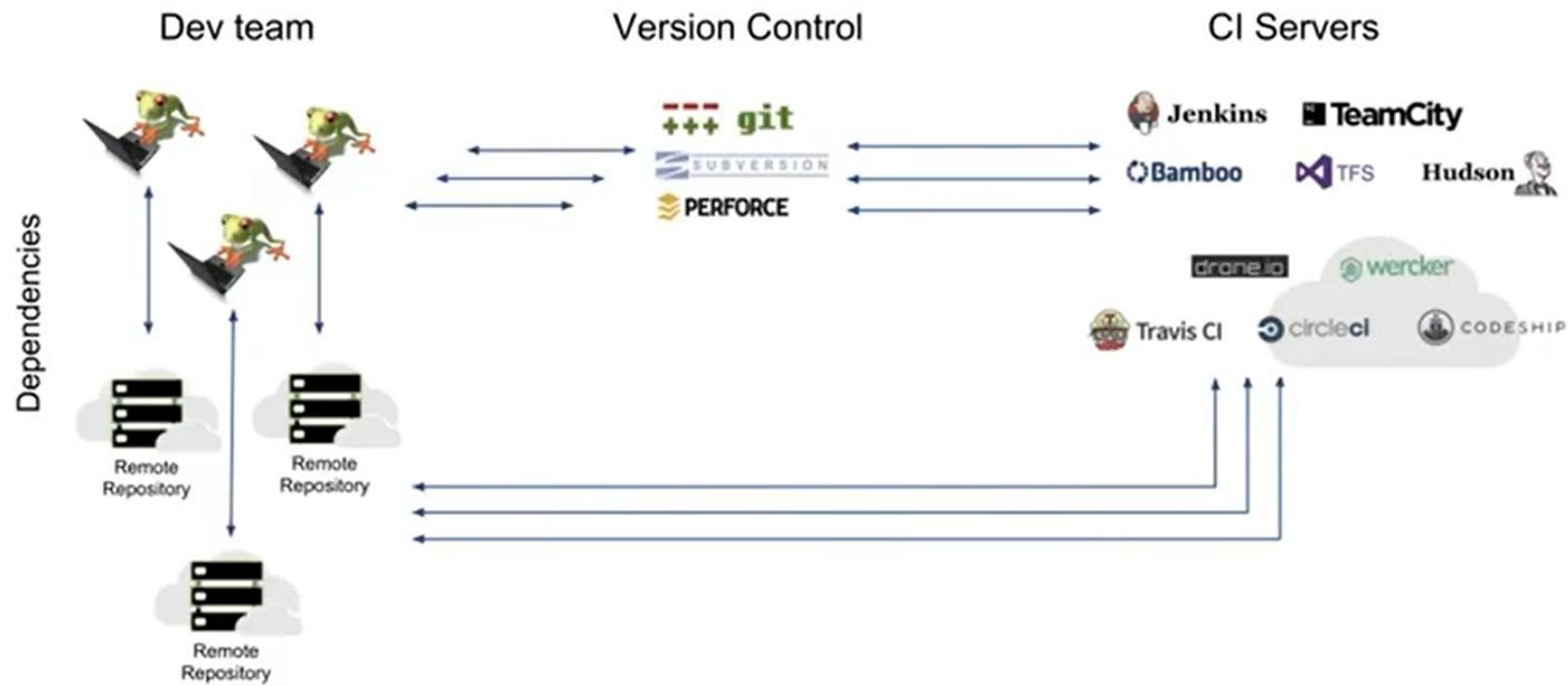
Compile

Artifact



# Artifactory

## From Code to Release



# Artifactory



ARTIFACTORY ESSENTIALS | SYSTEM OF RECORD

# Artifactory (System of Records)

## OVERVIEW OF RECORDS



Authoritative  
Source of  
Information

### DEVELOPERS GENERATE

Artifacts  
Binaries  
Builds  
Packages  
Bundles  
Other resources

### ARTIFACTORY RECORDS EVERYTHING

Timestamps  
Dependencies  
License details  
And much more

## Can I use my VCS (version control system) for binary management?

<u>Source</u>	<u>Binary</u>
Text	Blob
Diffable	Not diffable
Versioned by content	Versioned by name
Mutable	Immutable

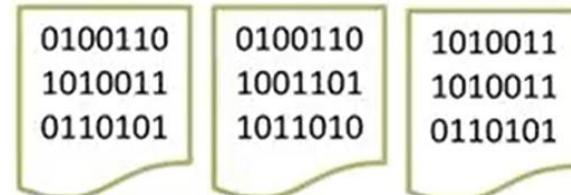
Version Control System

Binary Repository Manager

## Versioning by content vs. Versioning by name



VS.



MyBook-1.0.zip

MyBook-1.1.zip

MyBook-2.0.zip

What Changed?

## Can I use my FS (file server) for binary management?

- Support package specific protocols (maven, docker, npm, etc)
- Proxy other repositories
- Search by name, context and content
- Manage artifact lifecycle
- Cleanup SNAPSHOTS and GC strategies
- Optimize storage space
- Enforce Module Security
- Expose REST API / CLI for Automation
- Replication to remote sites
- Etc, etc, etc...

## Challenges Without a Binary Repository Manager

- Availability
  - Geographic spread
  - HA
  - Caching
- Functionality
  - Storage space and performance
  - Different technologies - different repositories
  - Manage versions and updates
- Security
  - Authentication and authorization
  - Integrity & Traceability

## What is a Binary Repository Manager?

A collection of binary software artifacts

and metadata ,

stored in a way which can be used by

clients, package managers, and CI servers

to retrieve / store binaries

during the development

and build process



## Are You Ready?

- CI/CD Pipeline
- From Code to Release
- Alternatives (like VCS or FS)
- Challenges while working without Binary Repository Manager
- Binary Repository Manager

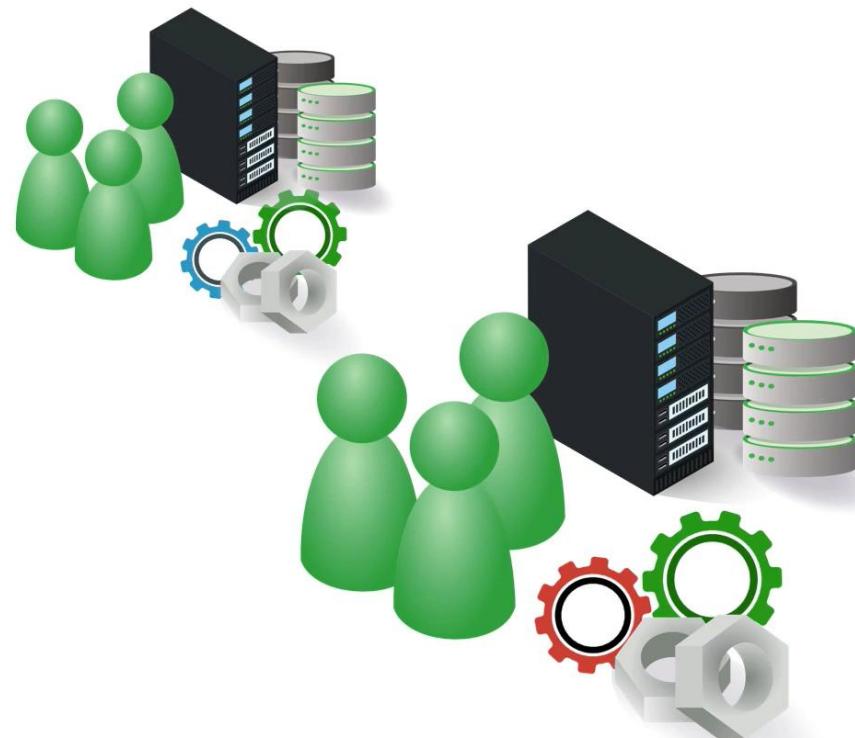
## Universal Binary Repository Manager

Integrates with package managers and CI tools



# Artifactory (Distributed Environment)

---



# Artifactory (Distributed Environment)

---

## Single Source of Truth for Binaries

Single access point that organizes all the resources



© 2010 JFrog Ltd. All Rights Reserved



# Artifactory (Distributed Environment)

---



# Artifactory (Distributed Environment)

---

## Single Source of Truth for Binaries

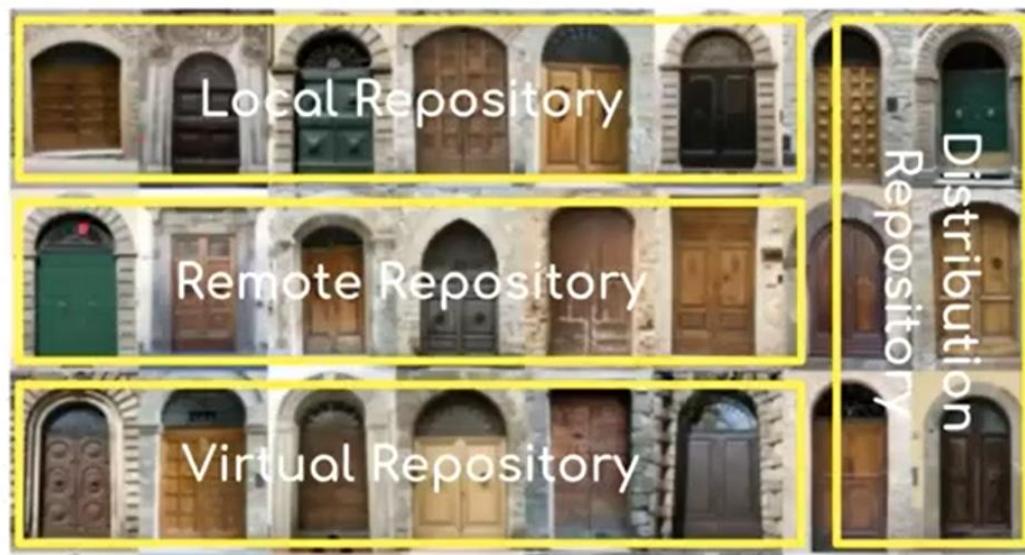
Single access point that organizes all the resources



# Artifactory (Distributed Environment)

## Single Source of Truth for Binaries

Single access point that organizes all the resources

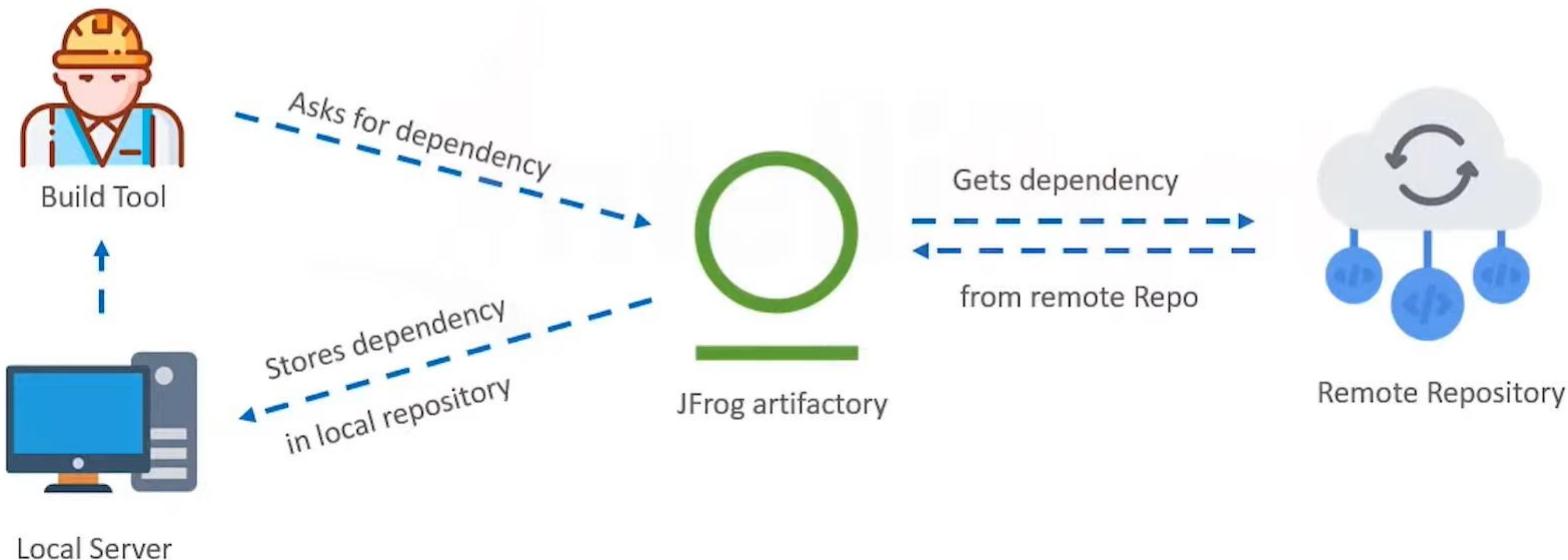


# Artifactory (Distributed Environment)

## JFrog Artifactory responsibilities?



Helps in downloading and managing dependencies



# Artifactory (Distributed Environment)

## why should we use JFrog for Devops?



System stability and reliability with artifactory high availability

Managing binaries across different environments

Security, access control and traceability

Full support for docker

Quickly replicating remote repos in local area

It has support for chef

The Binaries needed are always available on JFrog once they are cached with it and there is a certain assurance for their availability provided by the company, But the same can't be said about an open source software which doesn't provide the same assurance.



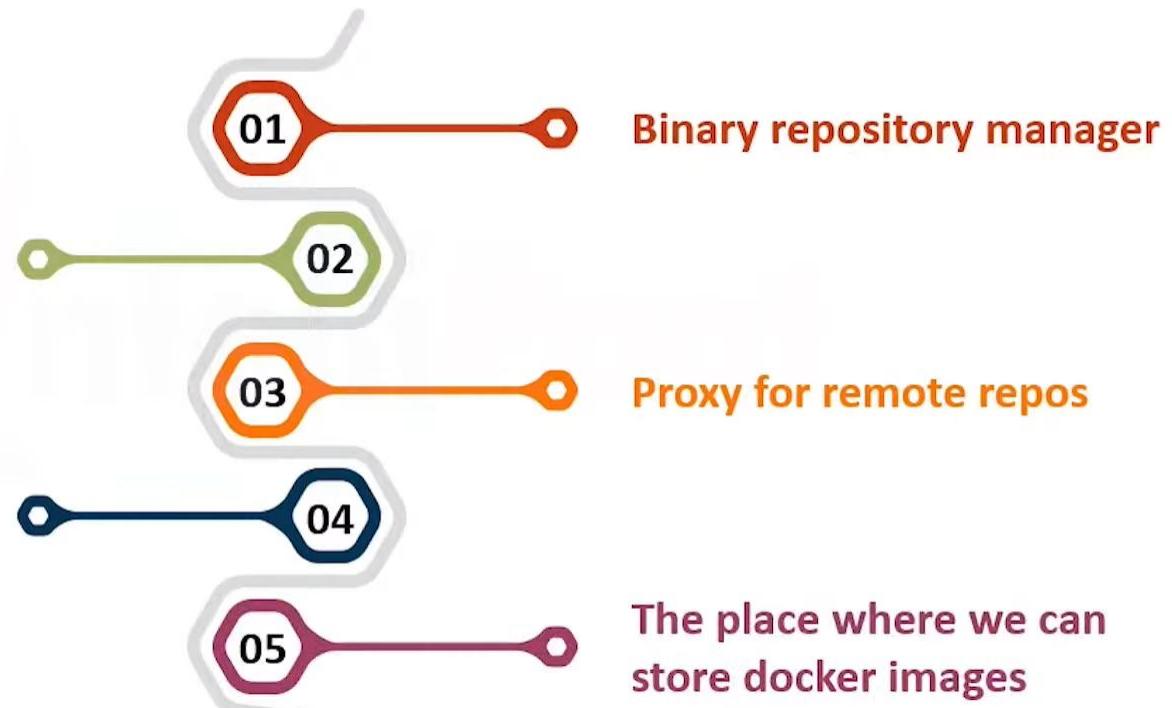
# Artifactory (Distributed Environment)

## Use Cases



The place where you can store all your artifacts

Dependency manager works against it



# Artifactory (Distributed Environment)

## Use Cases



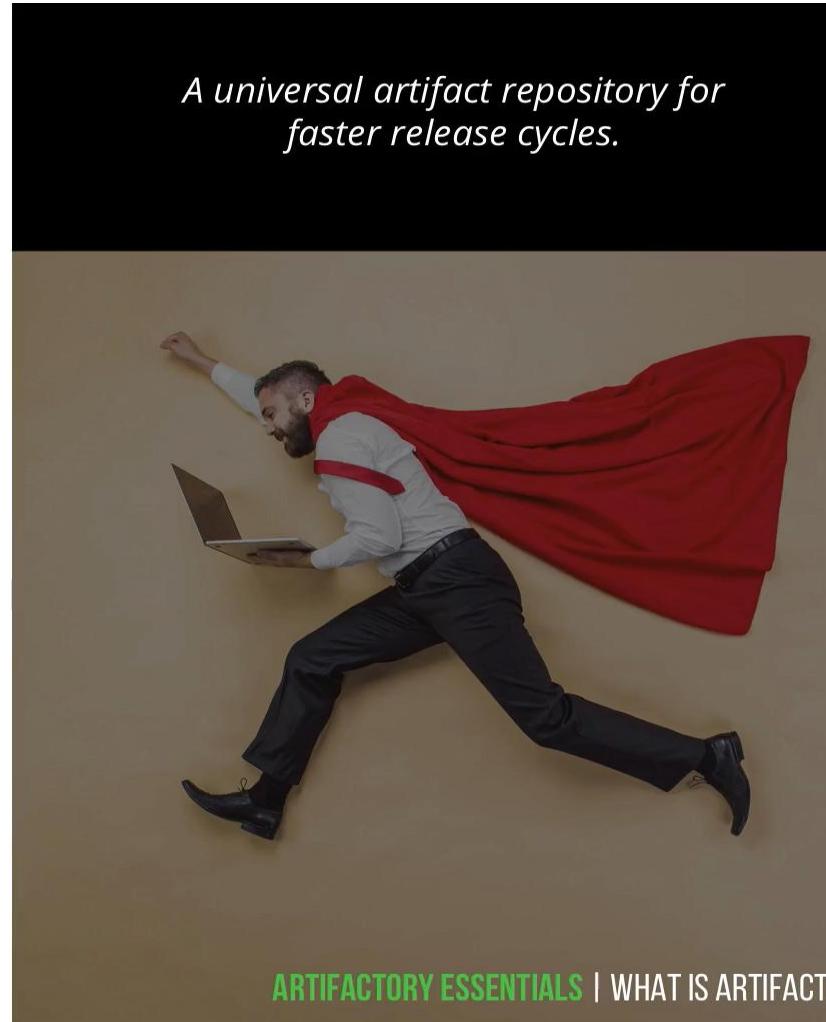
# Artifactory (Distributed Environment)

---

## WHAT IS ARTIFACTORY

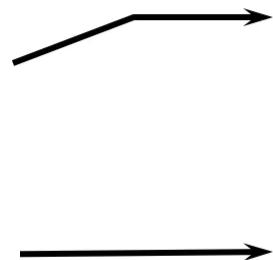
- 1 Overview
- 2 Repositories
- 3 Controls and Advantages
- 4 Supported Services

*A universal artifact repository for faster release cycles.*



# Artifactory (Distributed Environment)

## OVERVIEW



Binary repository manager with a common location to store all your build and local artifacts

Proxy remote management tool for libraries, and configuration files

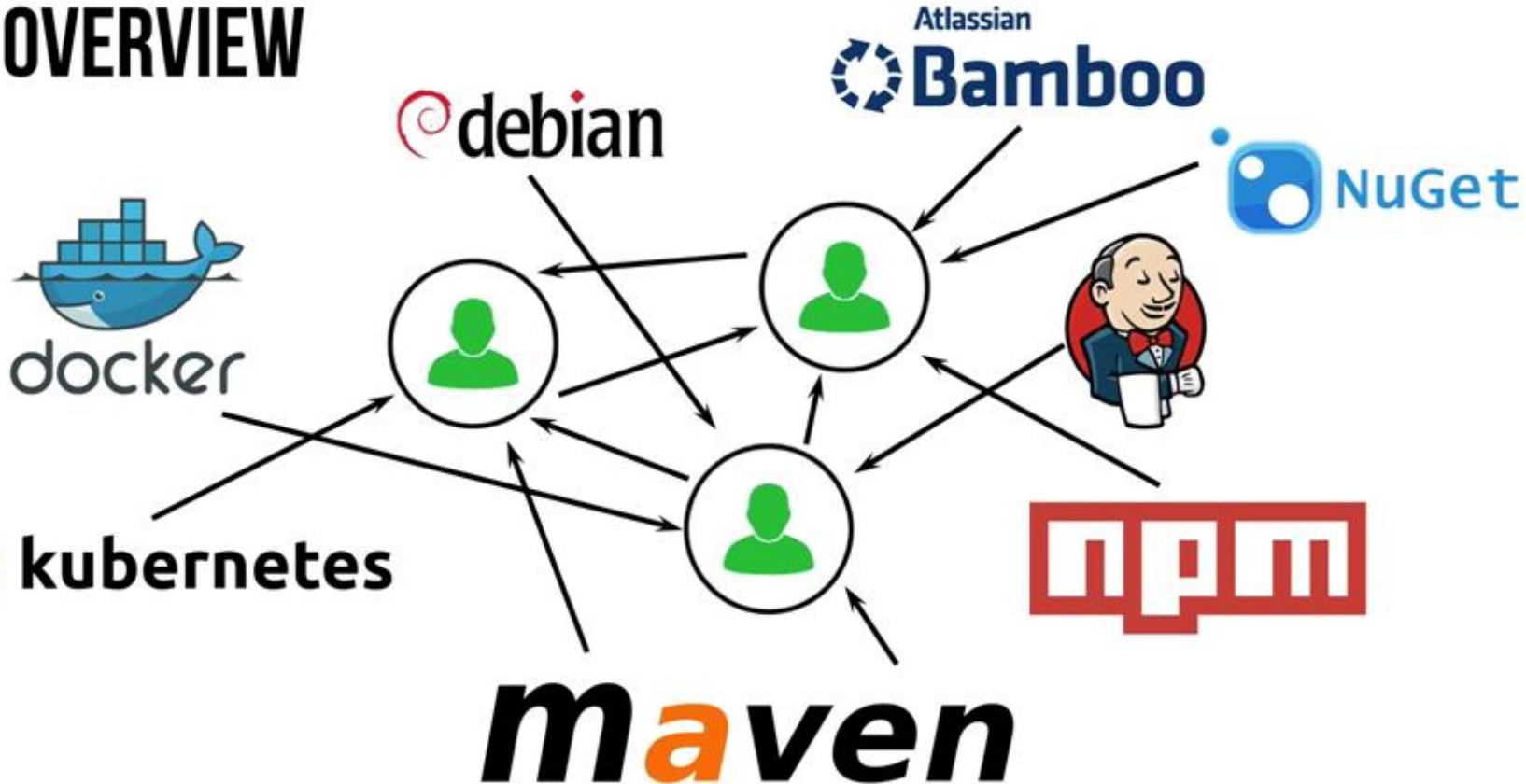
# Artifactory (Distributed Environment)

## OVERVIEW



# Artifactory (Distributed Environment)

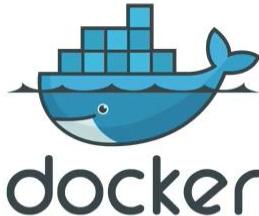
## OVERVIEW



# Artifactory (Distributed Environment)

---

## OVERVIEW



kubernetes

The Kubernetes logo is a blue hexagon containing a white steering wheel icon. The word "kubernetes" is written in a bold, black, sans-serif font.

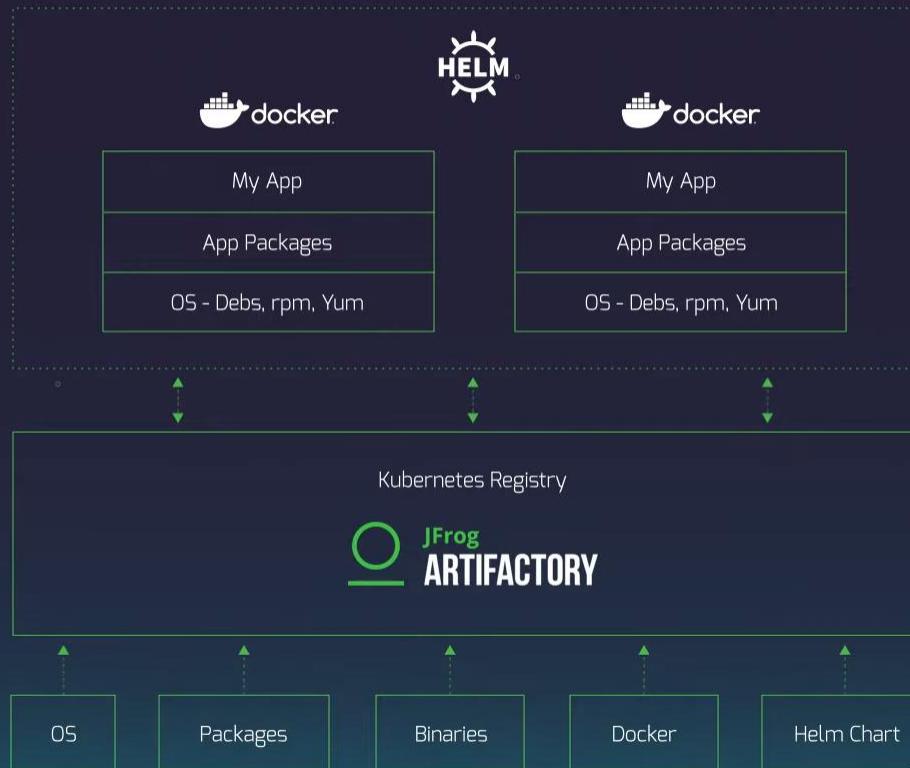
*maven*

The word "maven" is written in a large, bold, italicized black font, with the letter "a" colored orange.

# Artifactory (Distributed Environment)



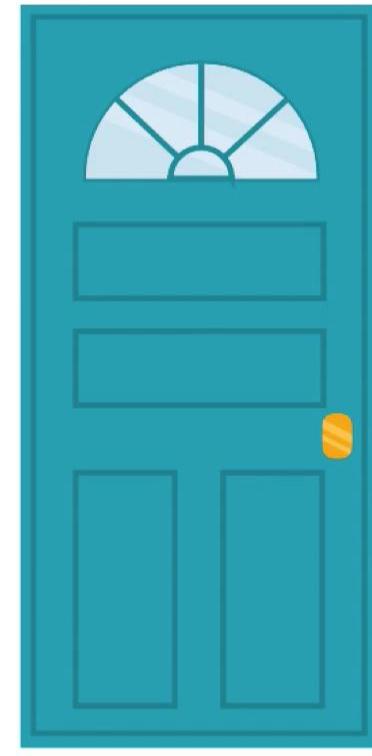
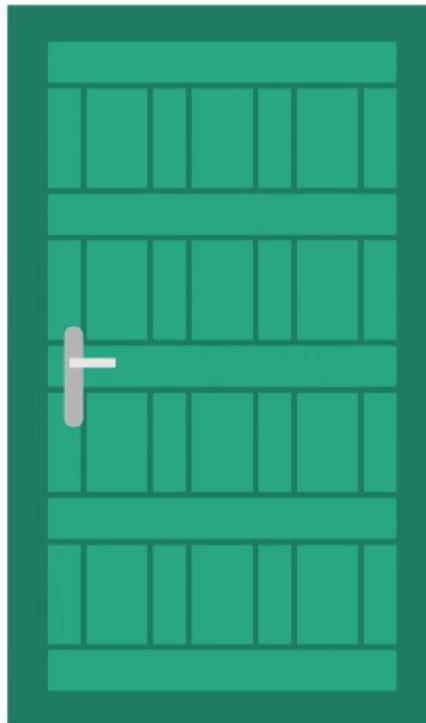
# Artifactory (Distributed Environment)



# Artifactory (Distributed Environment)

---

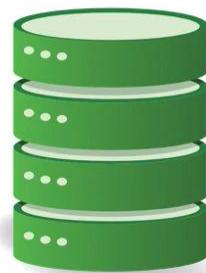
## REPOSITORIES



# Artifactory (Distributed Environment)

---

## REPOSITORIES



Local



Remote



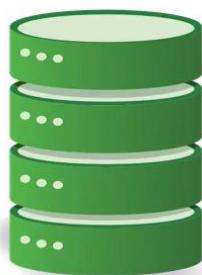
Virtual



# Artifactory (Distributed Environment)

---

## REPOSITORIES



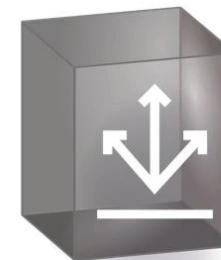
Local



Remote



Virtual



Distribution

# Artifactory (Distributed Environment)

## ADVANTAGES



Properties

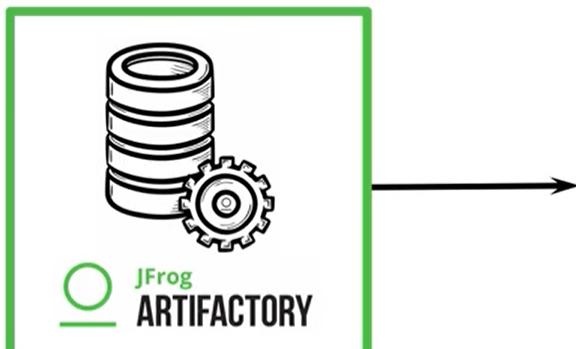
Traceability

Repository filters

Local access

# Artifactory (Distributed Environment)

## SUPPORTED SERVICES



Universal artifact repository

### Services

- Build tools
- Dependency managers
- CI servers
- Operational tools
- And more



# Artifactory (Distributed Environment)

## SUPPORTED SERVICES



- Artifacts
- 3<sup>rd</sup> party libraries
  - Internally-developed artifacts
  - Build results
  - In-house plug-ins
  - And more



# Artifactory (Distributed Environment)

## SUPPORTED SERVICES



Universal artifact repository

### Metadata

- Build environment
- Access permission
- Licenses
- Signatures
- Completely customizable

# Artifactory (Distributed Environment)

---

## Metadata

Artifactory allows you to associate metadata with content and thus perform complex queries and metadata driven deployment and resolution



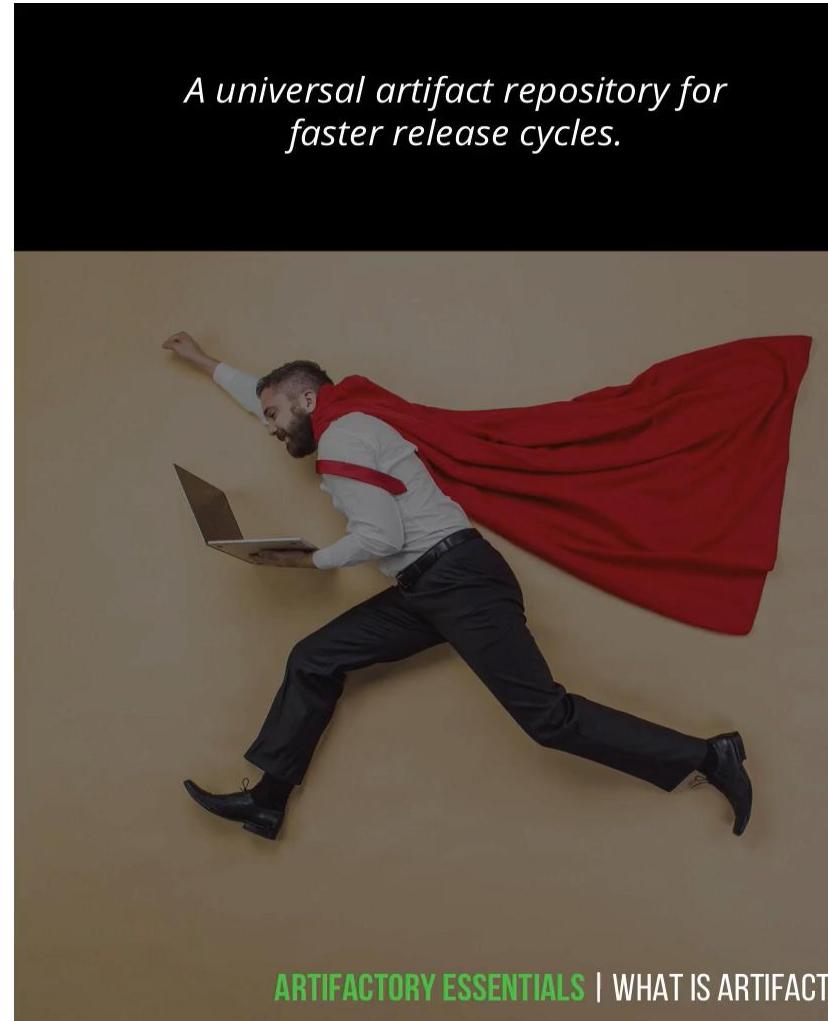
# Artifactory (Distributed Environment)

---

## WHAT IS ARTIFACTORY

- 1 Overview
- 2 Repositories
- 3 Controls and Advantages
- 4 Supported Services

*A universal artifact repository for faster release cycles.*

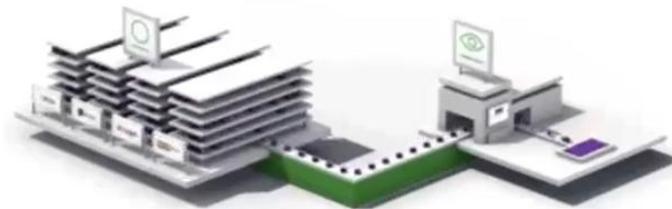


# Artifactory (Distributed Environment)

---

## Automation Driven Tool

Artifactory fits seamlessly with how your team works with a powerful API for automating processes



# Artifactory (Distributed Environment)

---

## What is JFrog Artifactory ?

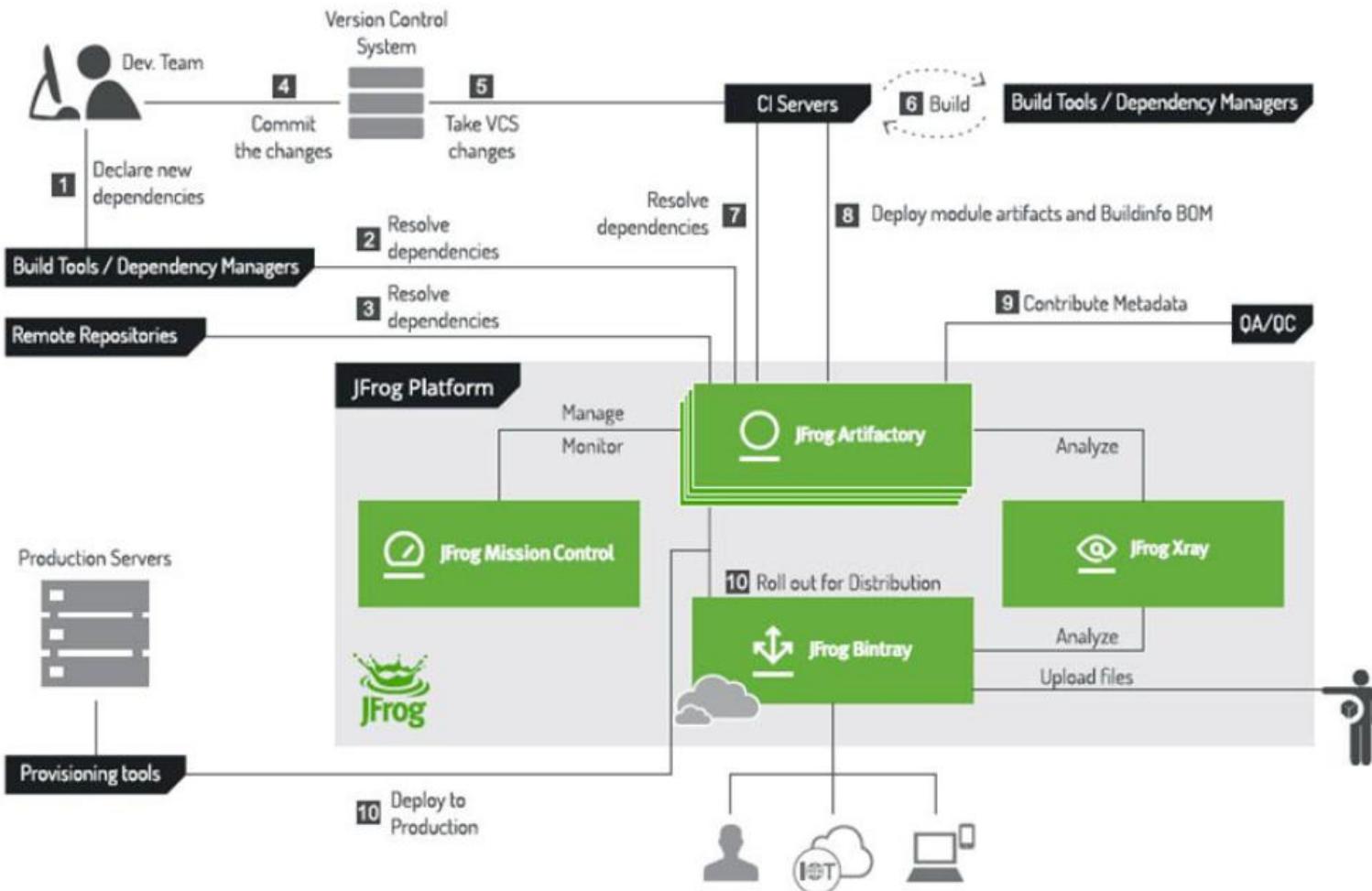
- A universal Binary Repository Manager
- The place where you can store all your artifacts and/or dependencies
- Proxy for remote repositories

## End-to-End Artifact Management

---

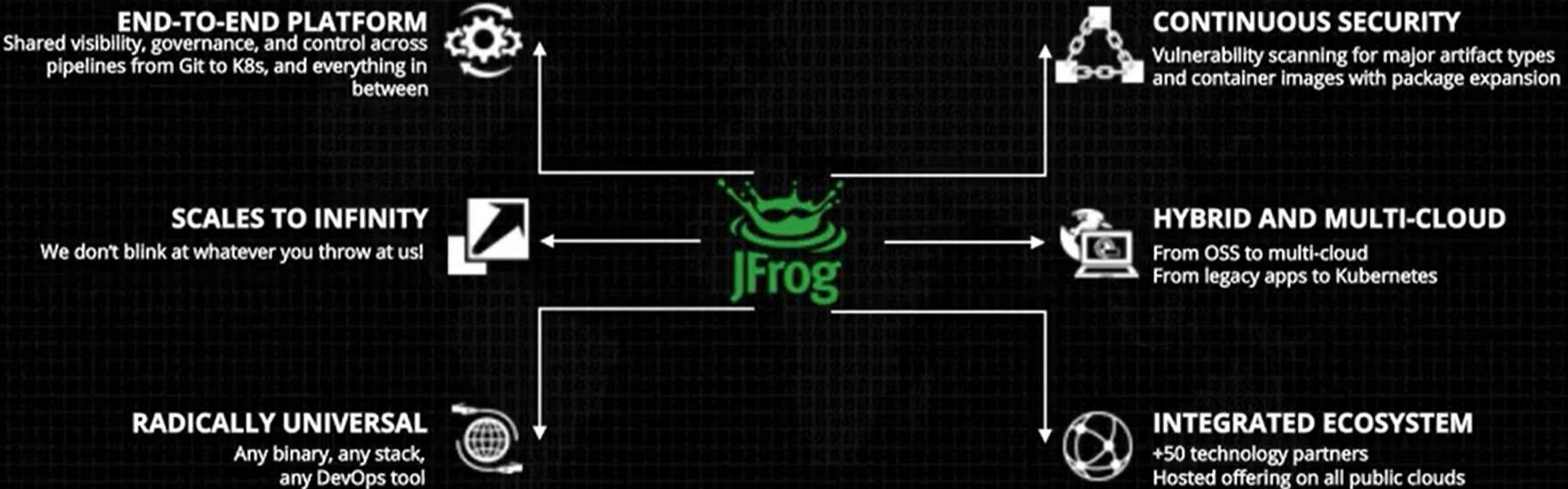
- JFrog Artifactory is just one component of a complete and tightly integrated end-to-end solution for artifact management available through JFrog's suite of products.
- As a complete solution to an organization's software delivery pipeline, Artifactory works seamlessly with the other products in the suite.

# End-to-End Artifact Management



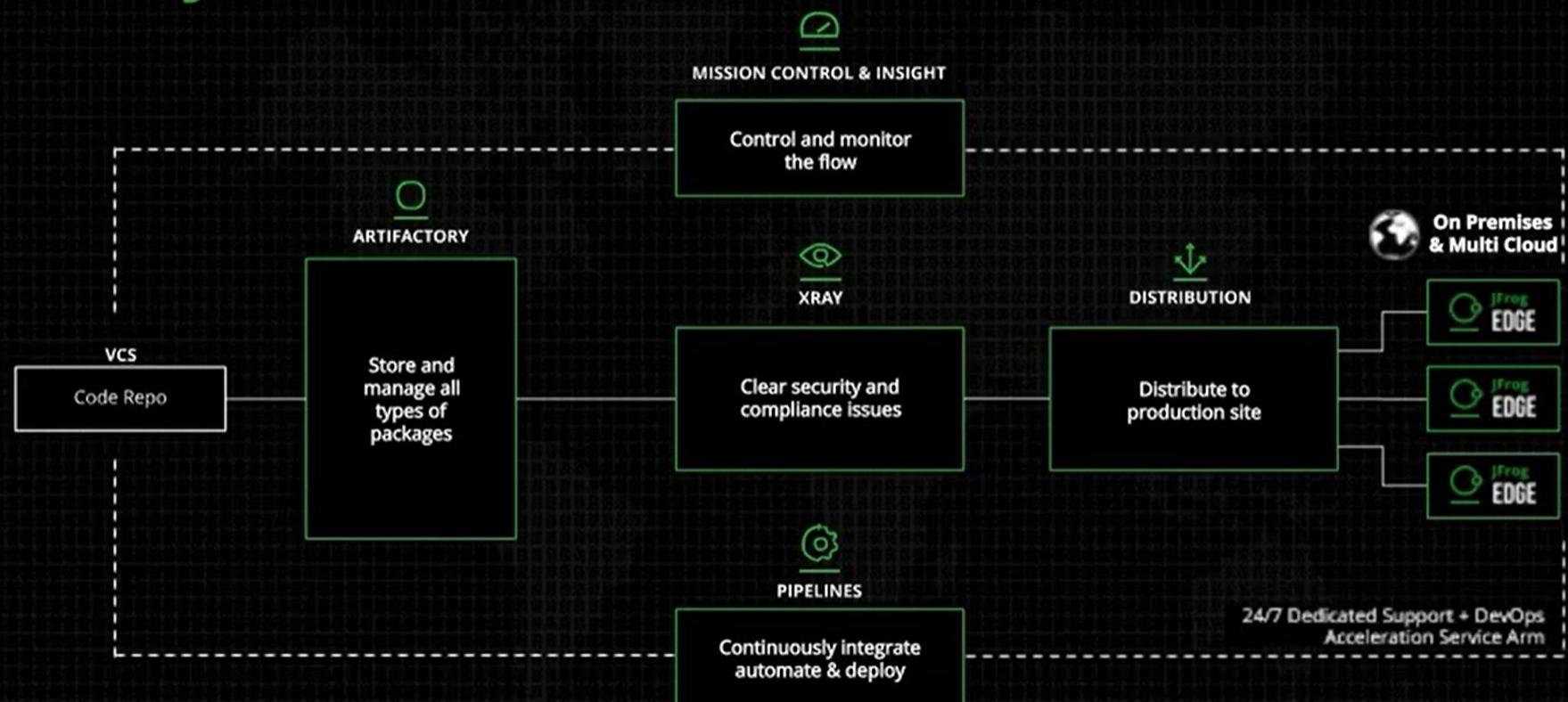
# End-to-End Artifact Management

## JFROG'S UNIFIED APPROACH

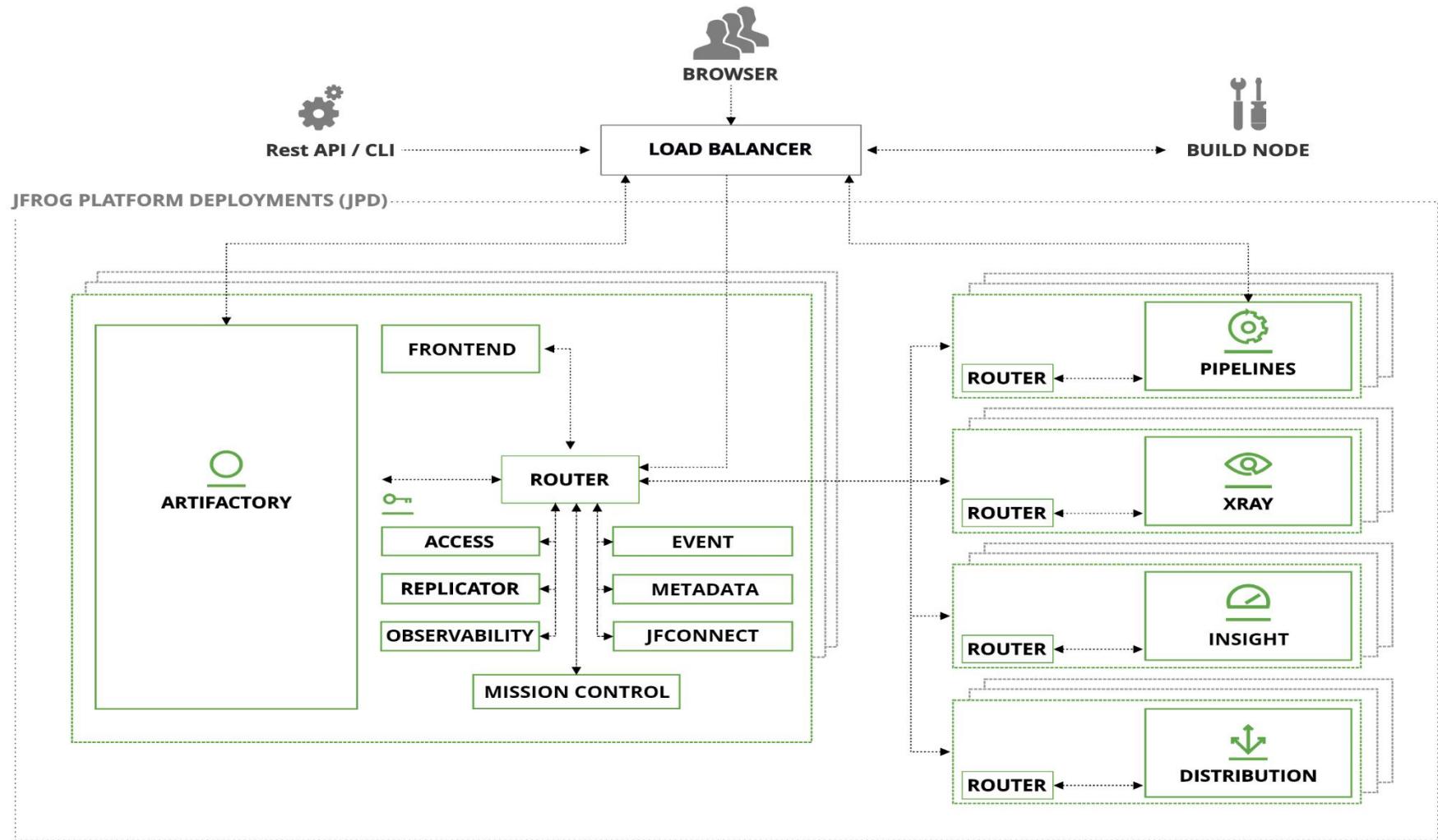


# End-to-End Artifact Management

## THE JFROG PLATFORM



# Jfrog Artifactory System Architecture



# Jfrog Microservice Components

---

- Artifactory - The application server
- Router - The central hub for communication between all the microservices, and cross-product.
- Access - The authentication server and service registry. Used to manage Users, Groups, Permissions and Access Tokens for all products in the JPD
- Event - The events distribution mechanism for JFrog products. Distributes Webhook events to external systems.
- Frontend - The application user interface (UI for the entire JPD)
- JFConnect - The microservice that fetches the entitlements from the subscription into a JPD
- Metadata - The component's metadata server. Serves the Packages screen in the JPD UI.
- Mission Control - The Mission Control microservice (available with an Enterprise+ license)
- Observability - The microservice responsible for consolidating logs and metrics in the service.
- Replicator - The replication engine (available with an Enterprise+ license)

# Jfrog X-Ray Components

---

- JFrog Xray Service
  - The Xray service contains the following microservices:
  - Indexer - Responsible for the indexing process, including:
    - Recursively extracting artifacts and builds
    - Collecting artifact metadata from accompanying files
    - Building an artifact components graph representation
  - Persist - Responsibilities include:
    - Matching the given components graph with the public component information
    - Completing component naming
    - Storing graph data and component metadata in PostgreSQL
  - Analysis - Responsible for enriching component metadata such as vulnerabilities, licenses and versions.
  - Server - Responsibilities include:
    - Generating violations by matching analysis data with watches and policies
    - Hosting the API and UI endpoints
    - Running scheduled jobs such as the database synchronization process
  - Router - Responsible for communication between all the microservices and cross-product

# JFrog Insight Service

---

- The Insight service contains the following microservices:
  - Insight - Handles insight related data management and API functions
  - Insight-Scheduler - Manages scheduling for different Insight tasks
  - Router - Responsible for communication between all the microservices and cross-product.

# JFrog Distribution Service

---

- The Distribution service contains the following microservices:
  - Distribution - Responsibilities include:
    - Hosting the API and UI endpoints
    - Sending distribution tasks via Redis to the Distributor
    - Syncing the other Distribution nodes with changes
    - Processing distribution tasks
    - Communicate with Artifactory edge nodes
    - Communicate with the source Artifactory
  - Router - Responsible for communication between all the microservices and cross-product.

# JFrog Pipelines Service

---

- The Pipelines service contains the following microservices:
  - API - Responsible for hosting the API endpoints.
  - Cron - Responsible for running internal jobs on intervals, including timing out steps and calculating statistics.
  - HookHandler - Processes resource webhooks received by API.
  - Logup - Transfers console logs from steps from the database to Artifactory as steps complete.
  - Marshaller - Responsible for starting build nodes in dynamic node pools.
  - Nexec - Responsible for running scripts on auto-initialized build nodes.
  - PipelineSync - Reads the YML files in the pipeline source and updates the pipelines and resources with any changes.
  - RunTrigger - Responsible for creating runs when a run is triggered manually or through a resource update.
  - StepTrigger - Determines which waiting steps are able to run and queues them to be picked up by the build nodes.
  - WWW - Hosts Pipelines UI pages that are supplemental to the JFrog Platform, including Pipeline History and Run Logs.
  - Router - Responsible for communication between all the microservices and cross-product..

# Artifactory (Distributed Environment)

## Universal Solution

### Supports all major package formats

Artifactory offers a universal solution supporting all major package formats including Maven, Gradle, Docker, Vagrant, Debian, YUM, P2, Ivy, NuGet, PHP, NPM, RubyGems, PyPI, Bower, CocoaPods, GitLFS, Opkg, SBT and more.

### Build integration

Through a set of plugins, Artifactory is tightly integrated with all major build tools and CI servers including Maven, Gradle, Ivy/Ant Jenkins, Team City, Bamboo and TFS.

### Metadata

Artifactory provides full metadata for all major package formats for both artifacts and folders. These include metadata that originates with the package itself, custom metadata added by users such as searchable properties and metadata that is automatically generated by tools such as build information and more.

# Artifactory (Distributed Environment)

---

## End-to-End Artifact Management

JFrog's complementary suite of products are an end-to-end solution covering the full lifecycle of software artifact management, control, analysis and distribution:

### End-to-End

**JFrog Artifactory: Universal Artifact Repository**

**JFrog Xray: Universal Artifact Analysis**

**JFrog Mission Control: Centralized Repository Management**

**JFrog Bintray: Universal Distribution Platform**

# Artifactory (Distributed Environment)

---

## Enterprise Ready

### High Availability

Full active/active HA solution with live failover and non-disruptive production upgrades.

### Replication

Supports any network topology, over multiple, geographically distant sites and data centers, through different options for replication including pull, push, event-based and **multi-push** replication. Multiple replication options facilitate disaster recovery with instant failover from Master site to a DR site located anywhere in the world.

### Storage

Supports a variety of enterprise-scale storage capabilities including S3 Object Storage, Google Cloud Storage and filestore sharding providing unlimited scalability, disaster recovery, and unmatched stability and reliability.

### Massively scalable

### Users, build servers and Interactions

Accommodates large load bursts with no compromise to performance. Increase capacity to any degree with horizontal server scalability to serve any number of concurrent users, build servers and interactions.

# Artifactory (Distributed Environment)

---

## Security, Authentication and Permissions

Secure deploy and download. Use a detailed permission schema per user or group (read, deploy, delete, annotate, manage and admin) to exercise fine-grained access control over repositories. Authenticate with industry standards like LDAP, HTTP SSO, Crowd, SAML, OAuth and more.

## Superior performance

Artifactory uniquely uses **checksum-based storage** where all metadata is stored in the Artifactory database. This means that any operation on an artifact is implemented through a quick database transaction resulting in up to 5 times better performance than competing products.

## Hybrid solution, on-prem or SaaS

Artifactory offers several options for on-premise installation (Zip, Debian, RPM, Docker image and Homebrew), as well as a SaaS solution hosted on either AWS or GCP.

## Licensing model

The Artifactory license is per server and allows an unlimited number of users and repositories.

## Unbeatable support

24/7 SLA-based support for any time zone with unbeatable response times.

# Artifactory (Distributed Environment)

## Full Automation

### REST API

Artifactory exposes an extensive REST API that provides access to its features anywhere in the development cycle effectively letting you automate any action you could do through the UI.

### JFrog CLI

A compact and smart client that greatly simplifies and optimizes automation scripts making them more efficient, more readable and easier to maintain across all JFrog products

## Full Support for Docker

### Multiple registries per instance

Artifactory lets you create any number of Docker registries per instance. Use local repositories as secure private Docker registries to distribute and share Docker images across your organization with fine-grained access control. Proxy and cache remote Docker registries with remote repositories, and aggregate local and remote Docker registries under a single virtual Docker registry to set up a Docker promotion pipeline.

### Detailed Metadata

Annotate Docker tags and images with searchable properties.

# Artifactory (Distributed Environment)

## Unique Advanced Capabilities

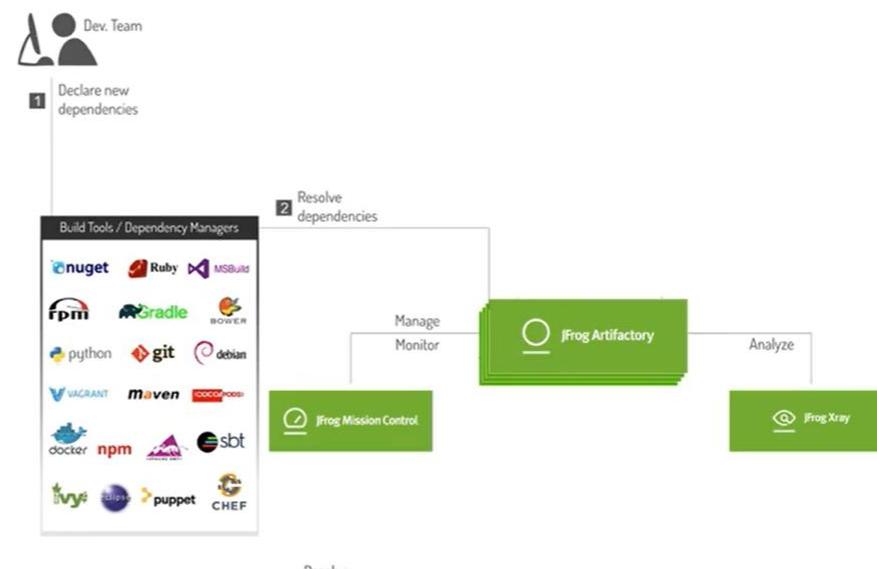
<b>Distribution</b>	Use distribution repositories to directly upload packages to JFrog Bintray for distribution.
<b>Quick resolution of production issues</b>	Artifactory stores exhaustive metadata for all build artifacts deployed to it. With this “bill of materials” it is easy to <b>faithfully reproduce a build</b> , and trace the cause of issues to reduce time to resolution (TTR), even if the build is already in production.
<b>Accommodate any custom life-cycle policy</b>	Using <b>custom properties</b> and <b>user plugins</b> , Artifactory can support custom behavior, triggered by virtually any action on an artifact, as required by an organization, to accommodate any custom lifecycle policy.
<b>Advanced search with Artifactory Query Language</b>	AQL gives you unprecedented flexibility in how you search for artifacts. It offers a simple way to formulate complex queries that specify any number of search criteria, filters, sorting options and output fields.
<b>Virtual Repositories</b>	Encapsulate any number of repositories making them seamlessly accessible from a single URL. Control which repositories can be accessed by different developers and set up a promotion pipeline.

# End-to-End Artifact Management

The screenshot shows the JFrog Platform Admin interface. The top navigation bar includes tabs for 'Verify your email' (Action Required), 'Artifactory Differentiators', 'New Messages!', and a '+' button. Below the bar, the URL is `vrepo.jfrog.io/ui/admin/projects/list`. The main header says 'JFrog Platform' and 'Administration'. A search bar says 'Search admin resources' and a green 'UPGRADE' button is visible. A yellow banner at the top right states: 'Your email address has not been verified yet and your subscription might get cancelled. Resend email verification'. On the left sidebar, under 'Administration', there are links for 'Projects', 'Repositories', 'Identity and Access', 'Users', 'Groups', 'Global Roles', and 'Permissions'. Under 'Learning Center', there is a link to 'ArtifactoryDifferen...pdf'. The main content area is titled 'Projects' and shows tabs for 'All Projects' (selected) and 'Unassigned'. It features a 'Get Started with your first Project!' message and a '+ Create New' button. A floating chat window says 'Hi there, How can we assist?' with a green frog icon. The bottom status bar shows system icons and the date/time: '20:14 30/01/2022'.

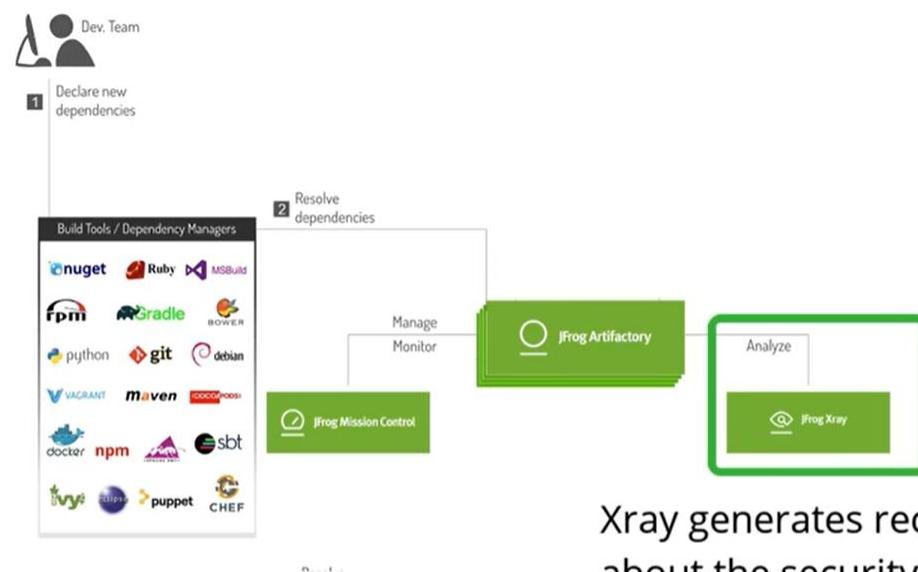
# End-to-End Artifact Management

## THE DEVELOPMENT PIPELINE



# End-to-End Artifact Management

## THE DEVELOPMENT PIPELINE



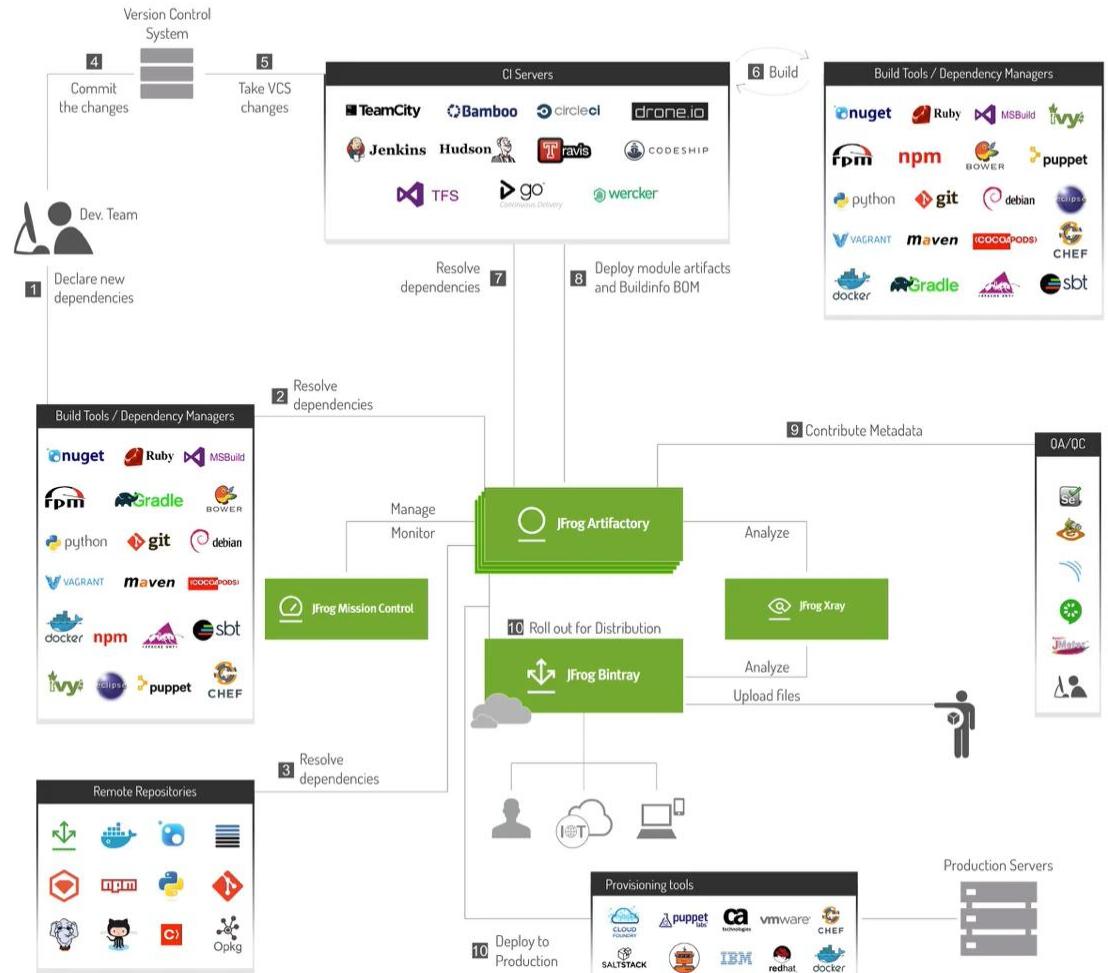
Xray generates records about the security of components in your build

# End-to-End Artifact Management

## THE DEVELOPMENT PIPELINE



2



# End-to-End Artifact Management

---

## SUMMARY



Summary



ARTIFACTORY ESSENTIALS | SYSTEM OF RECORD

# End-to-End Artifact Management

---

- **JFrog Bintray - Universal Distribution Platform**
  - Bintray is JFrog's software distribution platform that natively supports all major package formats allowing you to work seamlessly with industry standard development, build and deployment tools.

## End-to-End Artifact Management

---

- **JFrog Mission Control** - Universal Repository Management Mission Control offers centralized control, management and monitoring for all your enterprise artifact assets globally.
- By providing a clear and instant picture of the relationships and flow between your different development organizations, Mission Control provides your IT and Ops leaders real-time visibility into your worldwide development, distribution, and consumption of software packages.

# End-to-End Artifact Management

---

- **JFrog Xray - Universal Artifact Analysis**
- Xray works with JFrog Artifactory to analyze software artifacts and reveal a variety of issues and vulnerabilities at any stage of the software application lifecycle.
- By scanning binary components and their metadata, recursively going through dependencies at any level, JFrog Xray provides unprecedented visibility into vulnerable components lurking anywhere in your organization.

# End-to-End Artifact Management

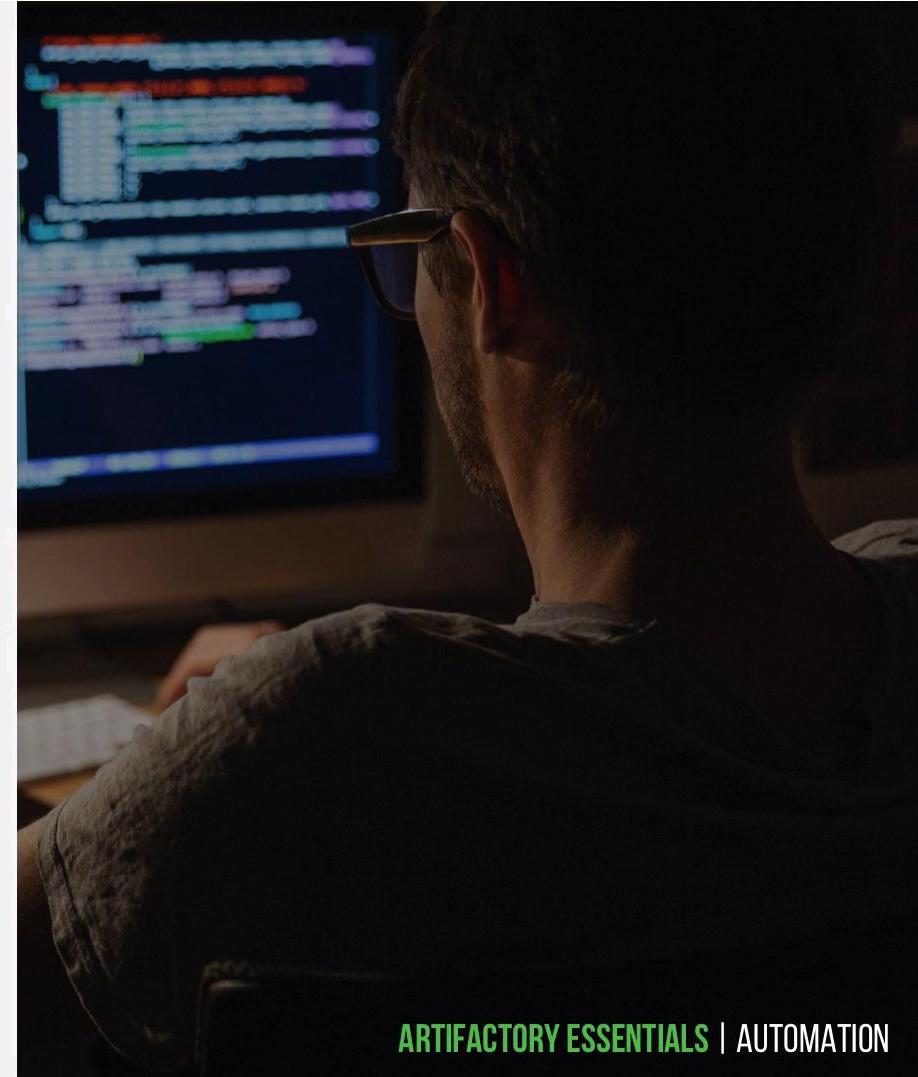


Built with automation in mind

# End-to-End Artifact Management

## AUTOMATION

- 1 Rest API
- 2 JFrog CLI
- 3 Plug-in Framework
- 4 Built-in Integration



# End-to-End Artifact Management

## REST API

> JFrog operates on these same REST APIs.

### USE TO INITIATE ARTIFACTORY TASKS

build automation  
artifact deployment  
plug-in endpoints

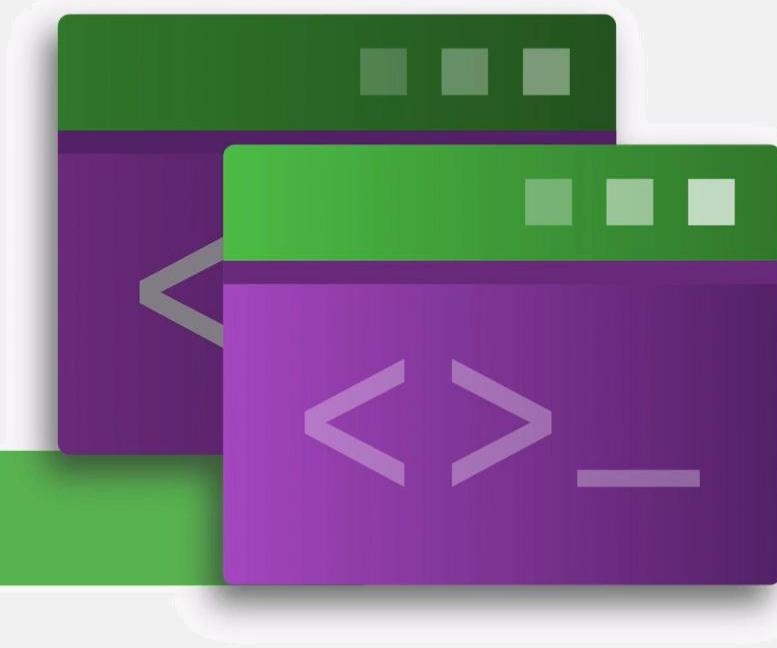
### INTEGRATES WITH ANY CI/CD AUTOMATION TOOL

# End-to-End Artifact Management

## JFROG CLI

- > interfaces with JFrog services through their REST APIs
- > client application
- > written in GO
- > deployed as an executable file

COMPACT, SIMPLE, AND OPTIMIZED

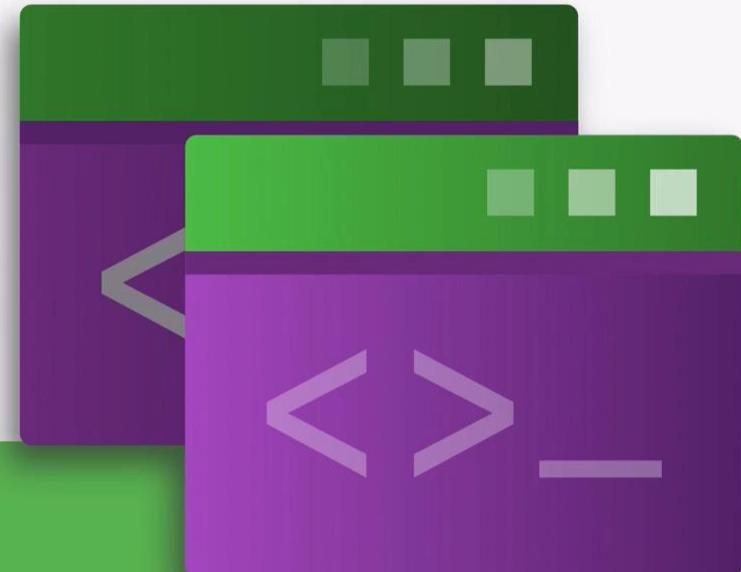


# End-to-End Artifact Management

## JFROG CLI

- > uses REST APIs to automate communication with Artifactory
- > optimizes tasks
- > checksum performance
- > simplified command structure

COMPACT, SIMPLE, AND OPTIMIZED



# End-to-End Artifact Management

---

## PLUG-IN FRAMEWORK



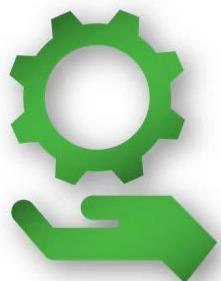
*Plug-ins manage or respond to any event and integrate with tools to create systems that work together without intervention.*



# End-to-End Artifact Management

---

## PLUG-IN FRAMEWORK



### BUILD TRIGGERS AND HANDLERS TO AUTOMATE:

Schedule tasks

Manipulate downloads & upload

Create new jobs

Extend security controls

# End-to-End Artifact Management

---

## PLUG-IN FRAMEWORK



JFrog offers sample plug-ins written in Groovy  
Download:  
<https://github.com/jfrog/artifactory-user-plugins>

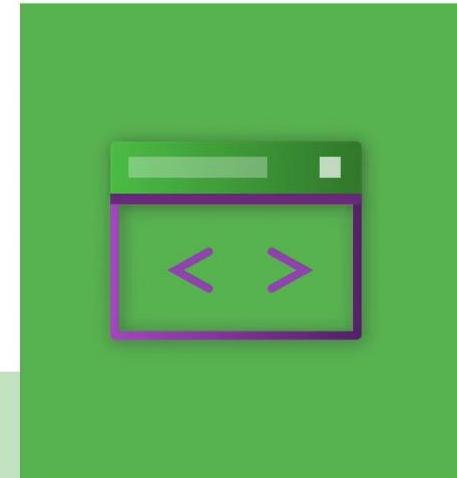
# End-to-End Artifact Management

---

## BUILT-IN INTEGRATION

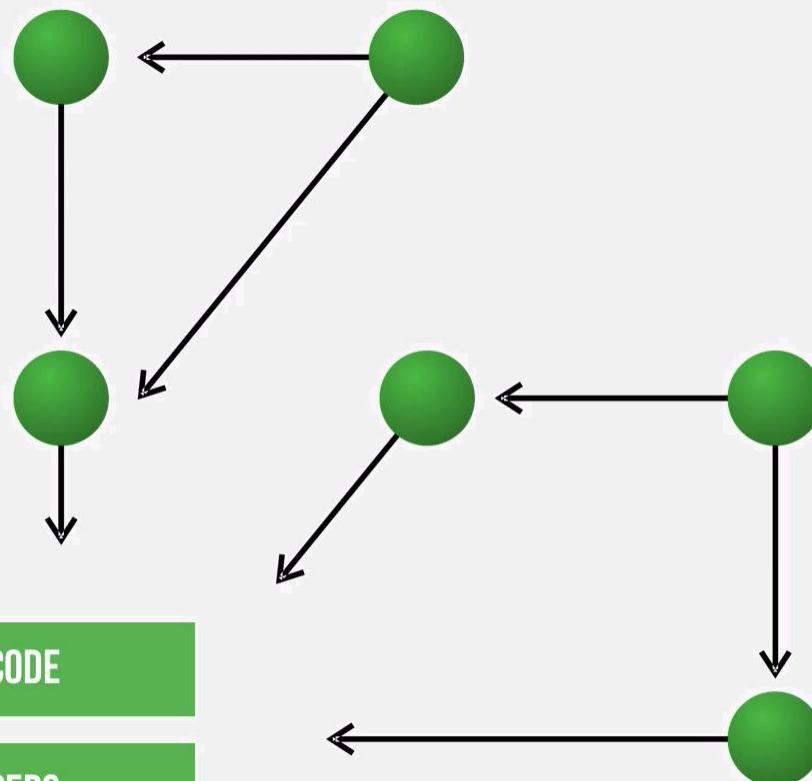
- > all builds
- > artifacts
- > dependencies
- > build environment
- > build operations
- > dependency relationships

CI/CD build integration



# End-to-End Artifact Management

## AUTOMATION



# Internal Architecture

---

JFrog  
ACADEMY



 JFrog  
ARTIFACTORY

**ARTIFACTORY ESSENTIALS**

**INTERNAL ARCHITECTURE**

# Jfrog System Directories

---

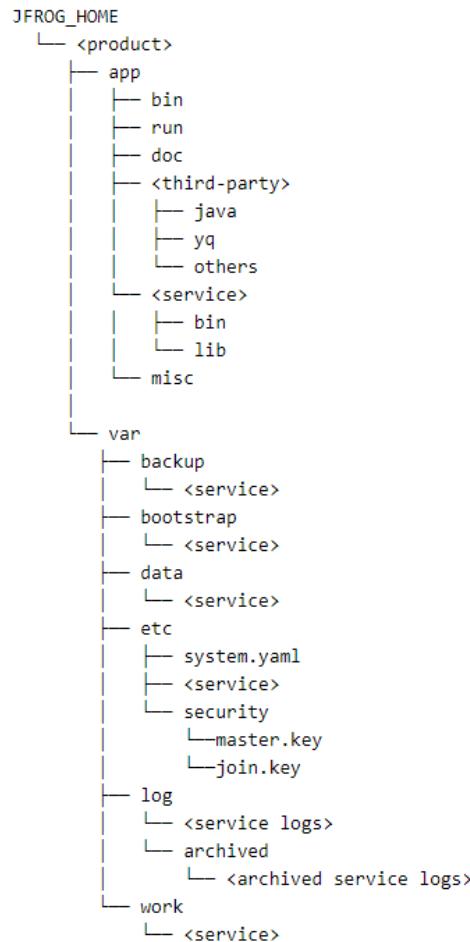
## JFrog Product Directory Structure

Product	Environment Variable	Linux/Windows Archive	RPM/Debian	Docker Container	Docker Host
JFrog Home	JFROG_HOME	<extracted directory>	/opt/jfrog	/opt/jfrog	ROOT_DATA_DIR * <b>Note:</b> Defined and customized in the .env file <b>Default value:</b> ~/jfrog
Data	JFROG_HOME/<product>/var	<extracted directory>/<product>/var	/var/opt/jfrog/<product>/var * 1. <b>Note:</b> symbolic link	/var/opt/jfrog/<product>/var	ROOT_DATA_DIR/var * <b>Default value:</b> ~/jfrog/<product>/var
Application	JFROG_HOME/<product>/app	<extracted directory>/<product>/app	/opt/jfrog/<product>/app	/opt/jfrog/<product>/app	-

# Jfrog System Directories

## Generic Directory Layout

This is a generic directory layout for all JFrog products.



# System Requirements

## Requirements Matrix

The following tables describe what is required for each single-service node. In a High Availability configuration,

### Supported Platforms:

Product	Debian	Centos	RHEL	Ubuntu	Windows Server	Helm Charts	SLES
Artifactory	8.x, 9.x, 10.x	7.x, 8.x	7.x, 8.x	16.04, 18.04, 20.04	2008 R2, 2016 or 2019	2.x, 3.x	12 SP 5
Insight	8.x, 9.x, 10.x	7.x, 8.x	6.x, 7.x, 8.x	16.04, 18.04, 20.04	✖	2.x, 3.x	12 SP 5
Mission Control	8.x, 9.x, 10.x	7.x, 8.x	7.x, 8.x	16.04, 18.04, 20.04	✖	2.x, 3.x	12 SP 5
Xray	8.x, 9.x, 10.x	7.x, 8.x	7.x, 8.x	16.04, 18.04, 20.04	✖	2.x, 3.x	✖
Distribution	10.x	7.x, 8.x	7.x, 8.x	16.04, 18.04, 20.04	✖	2.x, 3.x	✖
Pipelines	✖	7.x, 8.x	7.x, 8.x	16.04, 18.04, 20.04	Build nodes only	2.x, 3.x	✖

# System Requirements

---

<https://www.jfrog.com/confluence/display/JFROG/System+Requirements>

## Common Resources

---

<https://www.jfrog.com/confluence/display/JFROG/Common+Resources>

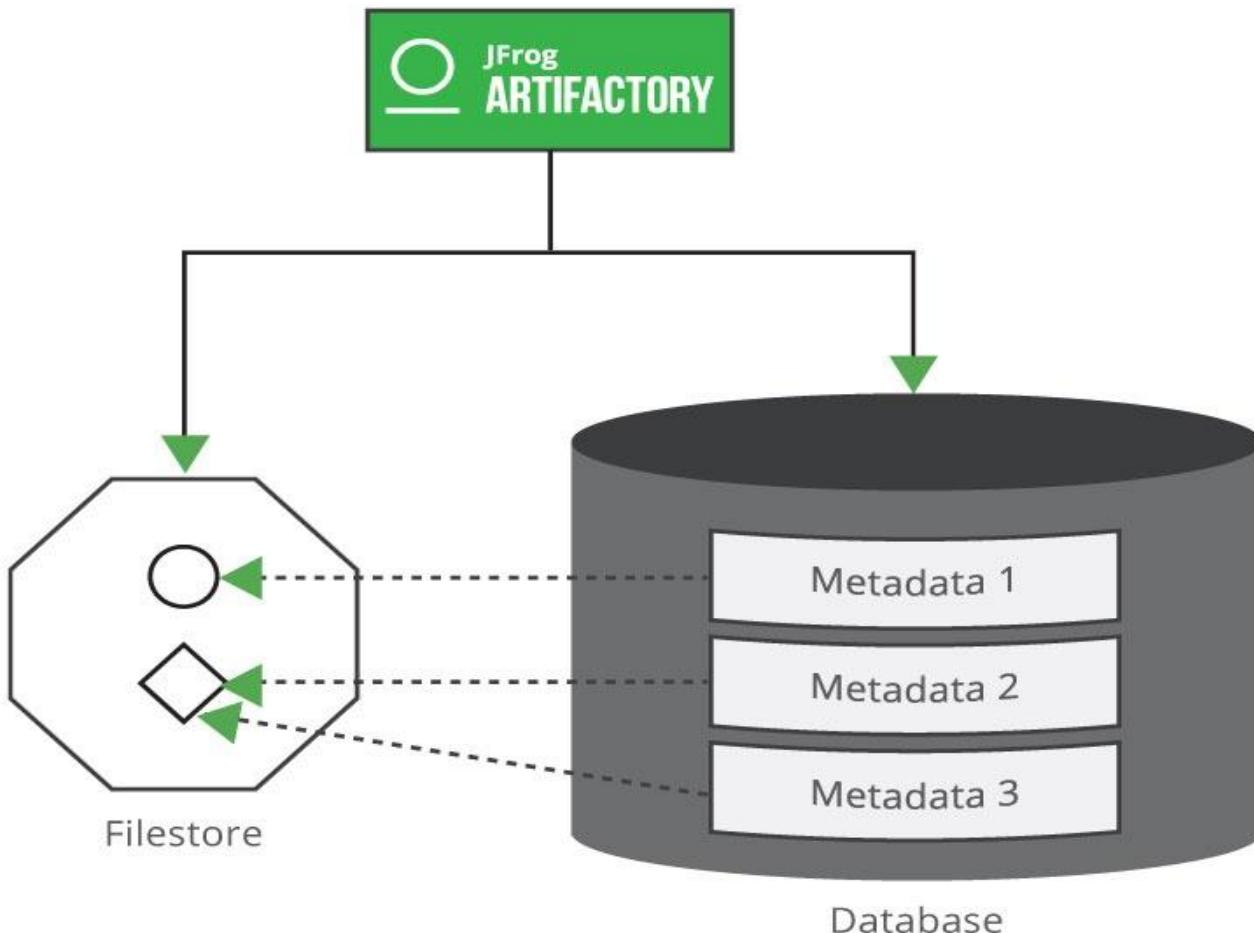
# Checksum-Based Storage

---

- Artifactory uniquely stores artifacts using checksum-based storage.
- A file that is uploaded to Artifactory, first has its SHA1 checksum calculated, and is then renamed to its checksum.
- It is then hosted in the configured filestore in a directory structure made up of the first two characters of the checksum.
- For example, a file whose checksum is "ac3f5e56..." would be stored in directory "ac"; a file whose checksum is "dfe12a4b..." would be stored in directory "df" and so forth.
- The example below shows the "d4" directory that contains two files whose checksum begins with "d4".

# Checksum-Based Storage

---



# Checksum-Based Storage Implementation

---

- Deduplication
  - Artifactory stores any binary file only once.
  - This is what we call "once and once only storage".
  - First time a file is uploaded, Artifactory runs the required checksum calculations when storing the file.
  - If the file is uploaded again (to a different location, for example), the upload is implemented as a simple database transaction that creates another record mapping the file's checksum to its new location.
  - There is no need to store the file again in storage. No matter how many times a file is uploaded, the file store only hosts a single copy of the file.

# Checksum-Based Storage Implementation

---

- Copying and Moving Files
  - Copying and moving a file is implemented by simply adding and removing database references and, correspondingly, performance of these actions is that of a database transaction.
- Deleting Files
  - Deleting a file is also a simple database transaction in which the corresponding database record is deleted.
  - The file itself is not directly deleted, even if the last database entry pointing to it is removed.
  - So-called "orphaned" files are removed in the background by Artifactory's garbage collection processes.

# Checksum-Based Storage Implementation

---

- Upload, Download, and Replication
  - Before moving files from one location to another, Artifactory sends checksum headers. If the files already exist in the destination, they are not transferred even if they exist under a different path.
- Filesystem Performance
  - Filesystem performance is greatly improved because actions on the file store are implemented as database transactions, so there is never any need to do a write-lock on the filesystem.

# Internal Architecture

---



# Internal Architecture

---

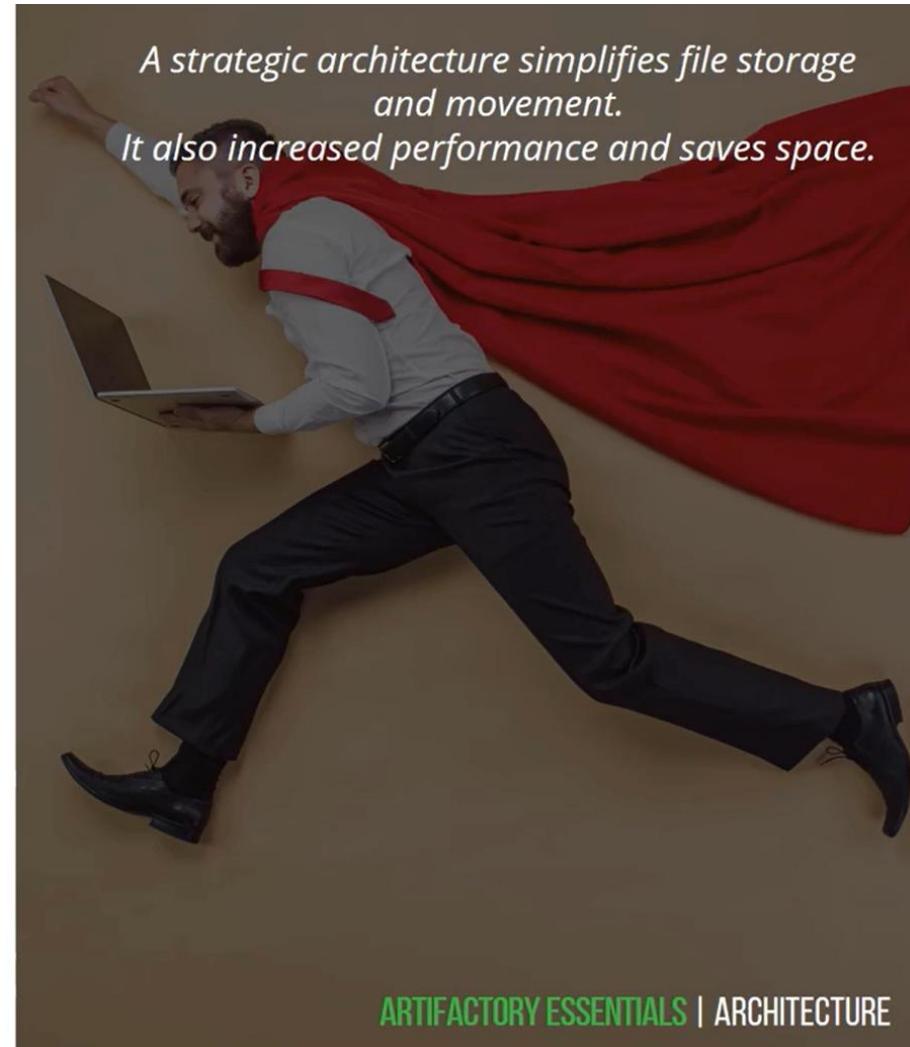
## ARCHITECTURE

1 Under the Hood

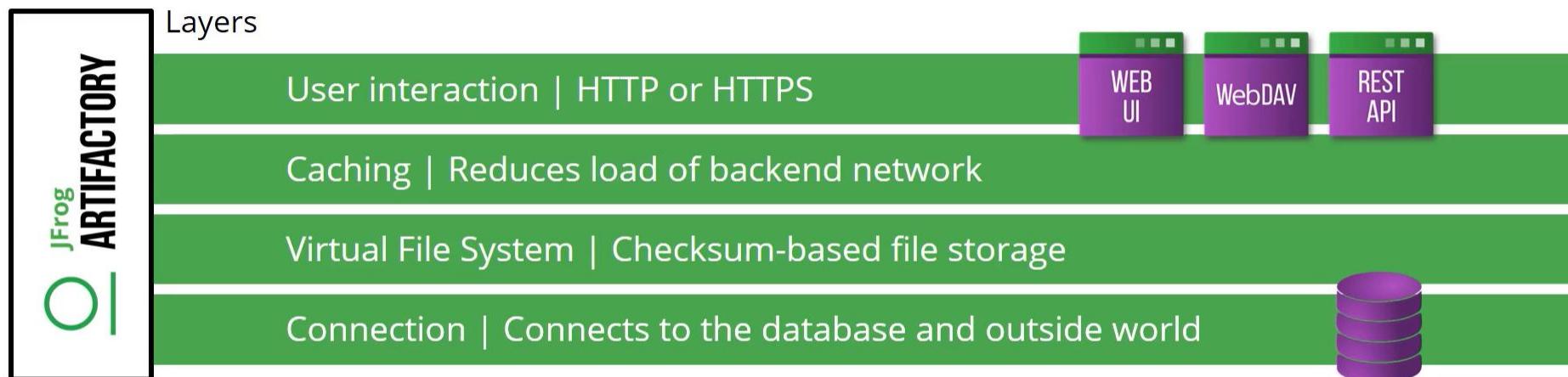
2 Microservice Architecture

3 Checksum Storage

4 Checksum-based Storage  
Advantages



# Internal Architecture



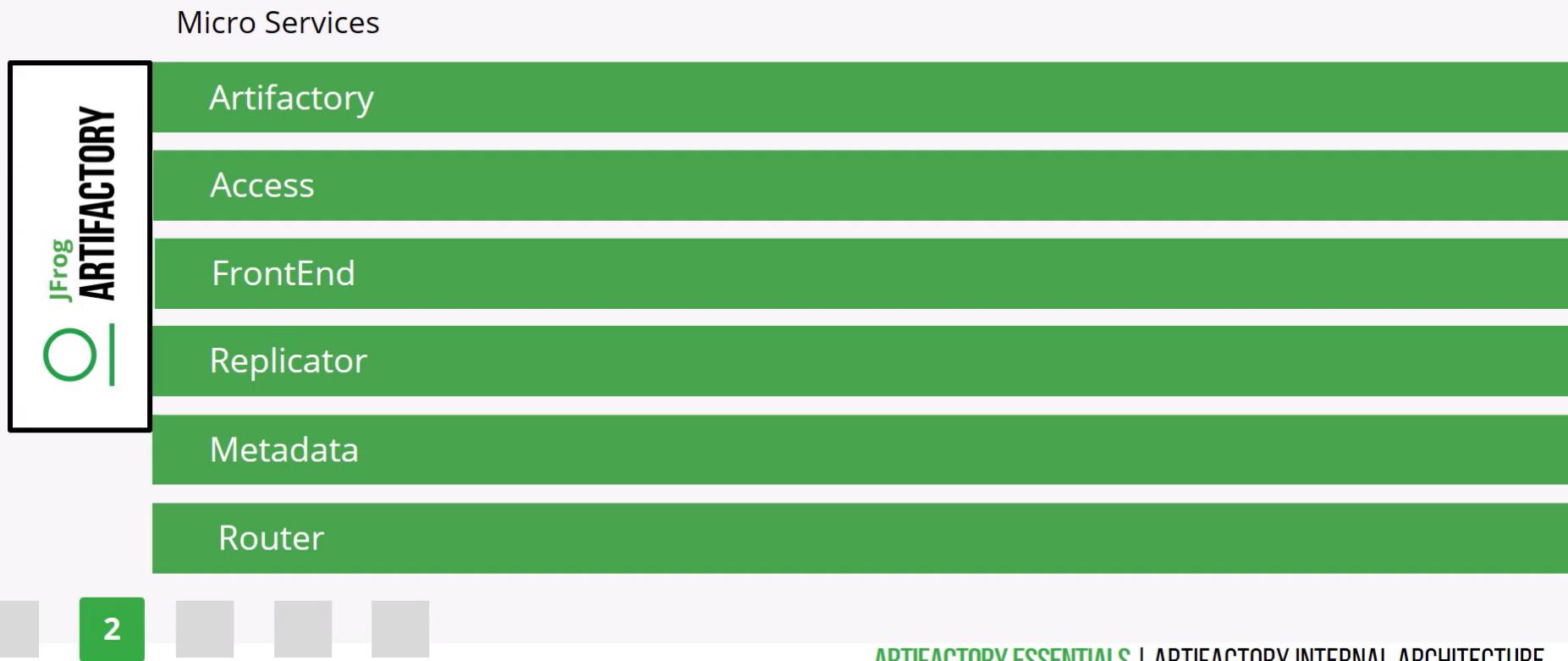
# Internal Architecture

---



# Internal Architecture(Micro Service)

---



# Internal Architecture (Check Sum Storage)

---



## Layers

### Virtual File System | Checksum-based file storage

Transactional, consistently healthy, corruption-protected storage layer

Improved performance and disk space savings

Database references move; not the file itself

# Internal Architecture (Check Sum Storage)



Layers

Virtual File System | Checksum-based file storage



An artifact's unique file ID is derived from a calculated checksum.

An artifact is stored only once, and any additional upload will result in only metadata updates.



ARTIFACTORY ESSENTIALS | ARTIFACTORY INTERNAL ARCHITECTURE

# Internal Architecture (Check Sum Storage)



## CHECKSUM-BASED STORAGE ADVANTAGES

# BENEFITS

Deduplication

Reduced file system activity

Improved performance

No write-locks required

Improved search stability

Customized metadata



# Internal Architecture (Check Sum Storage)

---



# Installation

---



ARTIFACTORY ESSENTIALS

## INSTALLATION AND SETUP

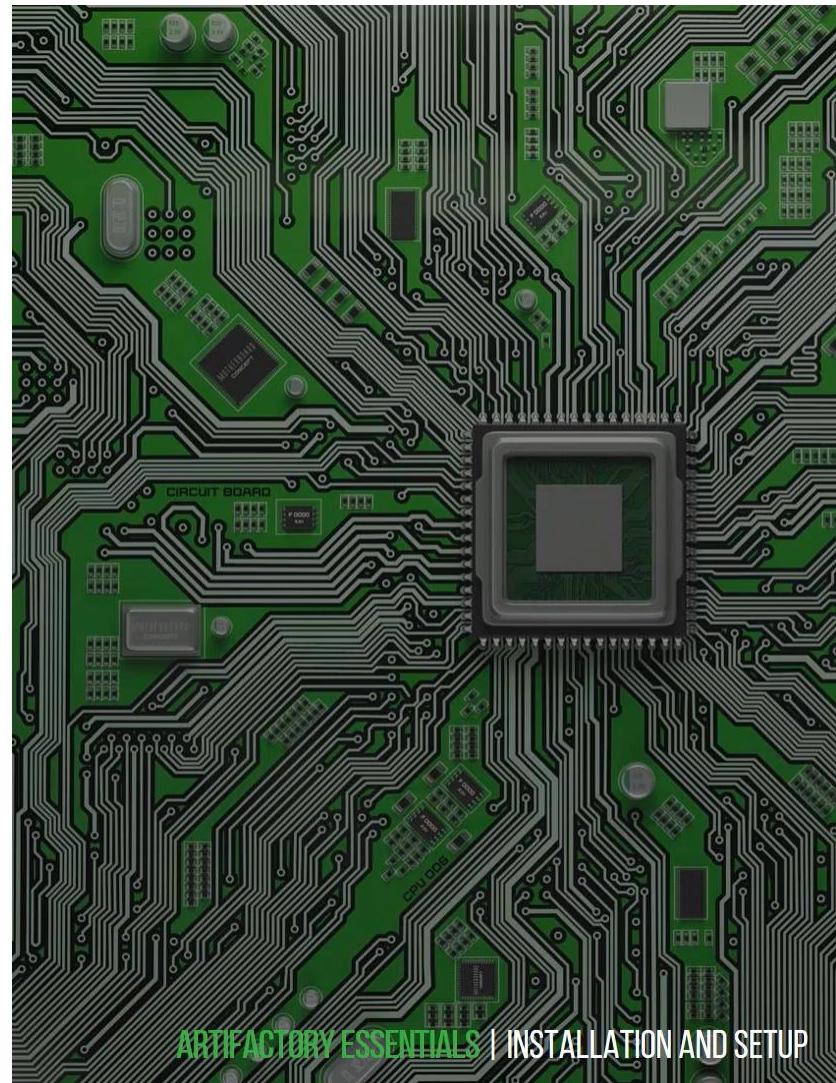
# Installation

---



## INSTALLATION AND SETUP

- 1 Installation Requirements**
- 2 Installation**
- 3 Network Configuration**
- 4 JVM Fine-Tuning**
- 5 Database Considerations**
- 6 S3 Installation**



# Installation



## INSTALLATION REQUIREMENTS

Framework

Network

Hardware

# Installation



## INSTALLATION REQUIREMENTS

Framework	Software	Heap size
	JDK 11  Point \$JAVA_HOME to Installation	4 GB  8 GB (small-medium)



# Installation



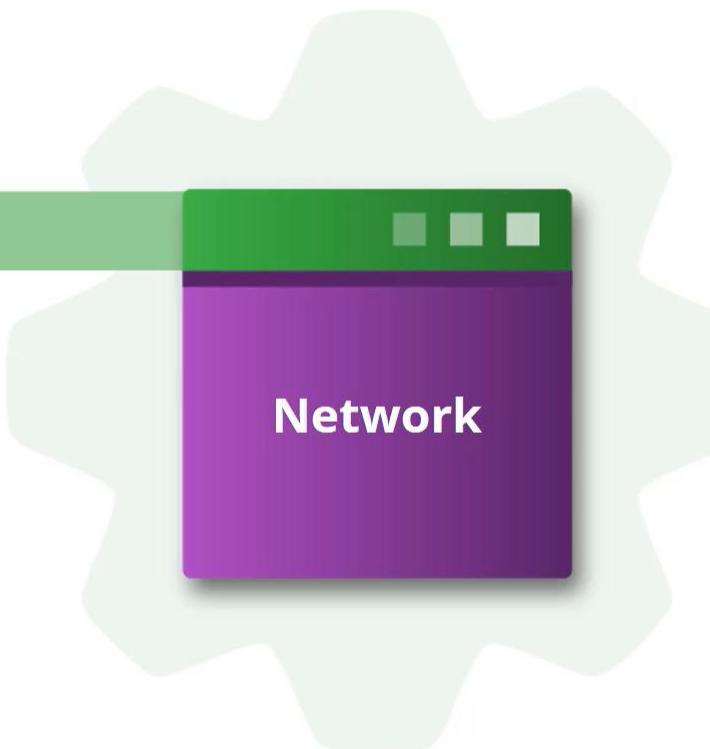
## INSTALLATION REQUIREMENTS

### Network

1 GB NIC  
*up to 32 GB Heap*  
10 GB NIC  
*up to 128 GB Heap*

### HTTP/HTTPS

Reverse Proxy for SSL  
HTTP for Apache ←→ Tomcat  
*not AJP*  
DB/FS at 100 GB (min)



A large, semi-transparent light-green cloud shape surrounds a central purple rectangular box. The box has a green header bar with three small squares. The word "Network" is written in white in the center of the purple area. To the left of the main table, there is a smaller, solid green box containing the number "1".

Network

# Installation



## INSTALLATION REQUIREMENTS

### Hardware

#### Hardware

Storage, fast disk with 3x space of total artifacts

OS/JVM, 64 bit

Processor, 4 cores (sm-med)

Processor, 16 cores (large)

# Installation



## INSTALLATION REQUIREMENTS

### Hardware Table

Storage	Memory (RAM)	Processor	OS/JVM	Num of Builds	Num of Developers
Fast disk with free space = X3 the total size of artifacts stored	4 GB	4 Cores	64 bit	4	1-20
Fast disk with free space = X3 the total size of artifacts stored	8 GB	4 Cores	64 bit	20	20-100
Fast disk with free space = X3 the total size of artifacts stored	12 GB	8 Cores (16 recommended)	64 bit	40	100-200

# Installation



## INSTALLATION



### Installation Distribution

RPM

Debian

.zip

Docker Container

Helm for Kubernetes

# Installation

---



## INSTALLATION

### Execution

Setup Wizard

Alternate Silent Installation

JFrog ARTIFACTORY



2

# Installation

---

- <https://www.jfrog.com/confluence/display/RTF6X/Installing+with+Docker>
- Running Artifactory Pro
- `docker run --name artifactory -d -p 8081:8081 docker.bintray.io/jfrog/artifactory-pro:latest`

# Installation

---

- Setting Java Memory Parameters
- ```
docker run --name artifactory -d -p 8081:8081 -e EXTRA_JAVA_OPTIONS='-Xms512m -Xmx2g -Xss256k -XX:+UseG1GC' docker.bintray.io/jfrog/artifactory-pro:latest
```

## Jfrog pro\* Installation

---

- <https://jfrog.com/download-jfrog-platform/#xray>
- Download docker-compose file
- Try this, Check mail for link
- <https://jfrog.com/xray/install/>

# Installation

Minimum System Requirements: 8 CPU, 16 GB Memory, 300 GB Fast Disk (3000+IOPS). External Ports: 8081, 8082.  
Requires Root access.

## Docker Compose

Requires Docker Compose on Linux

### 1. Download the installer

```
wget -O jfrog-compose-installer.tar.gz "https://releases.jfrog.io/artifactory/
```

### 2. Extract it

```
tar -xvf jfrog-compose-installer.tar.gz
```

### 3. CD into directory

```
cd jfrog-platform-trial-peo*
```

### 4. Run the config script

```
sudo ./config.sh
```

### 5. Start RabbitMQ

```
sudo docker-compose -p trial-peo-rabbitmq -f docker-compose-rabbitmq.yaml up
```

### 6. Start JFrog Artifactory and Xray

```
sudo docker-compose -p trial-peo up -d
```

## + RPM for CentOS (7, 8) / RHEL (7, 8)

## + Debian for Debian (9, 10) / Ubuntu (16.04, 18.04)

# Installation

```
eswaribala@WINDOWS-BVQNF6J: ~
/home/eswaribala/.hushlogin file.
eswaribala@WINDOWS-BVQNF6J:~$ wget -O jfrog-compose-installer.tar.gz "https://releases.jfrog.io/artifactory/jfrog-prox/
ng/artifactory/pro/docker/jfrog-platform-trial-prox/[RELEASE]/jfrog-platform-trial-prox-[RELEASE]-compose.tar.gz"
Warning: wildcards not supported in HTTP.
--2022-01-30 21:27:50-- https://releases.jfrog.io/artifactory/jfrog-prox/org/artifactory/pro/docker/jfrog-platform-tri-
l-prox/[RELEASE]/jfrog-platform-trial-prox-[RELEASE]-compose.tar.gz
Resolving releases.jfrog.io (releases.jfrog.io)... 52.6.179.45, 34.238.197.127, 52.200.29.158, ...
Connecting to releases.jfrog.io (releases.jfrog.io)|52.6.179.45|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://releases-cdn.jfrog.io/filestore/3a/3a3f59bf097fcfa7467545c8446e03c55fff16c?response-content-type=app-
ication/x-gzip&response-content-disposition=attachment%3Bfilename%3D%22jfrog-platform-trial-prox-7.31.13-compose.tar.gz
22&x-jf-traceId=c2e107e4b72da593&X-Artifactory-repositoryKey=jfrog-prox&X-Artifactory-projectKey=default&Expires=164355-
331&Signature=0A4kw1gd1l0T1GusgclU8K~tT8AHk5-LikEHx6xYw09xysgQZnIZeY7S6GsUGi0ReAe-oswqjXXrZCuFb4JtHtGtG5NHFc-Kc900KE55-
ZetloihNQo90B4iKv8zPe2v2XTivmYvoB31iGGlfvs200BoUS2mEbbD8ljACpOPQ70jzbjTAE5YemaqRNfh3fNt3LNVYOs8Ju5CptN3s7MCPSK851Pc-~bk-
bsB0d-P25AcCpmzhI7D0kr4Czs5Alt1Zqguf6jsvVh~CKSjjtmk1d1HS-jJ6CBwnHdwAdwdJg-JoXaIs72Y7AZU2NjktivK~p7JN2KKgwJusg2ShKurQ_&
ey-Pair-Id=APKAJ6NHFWMVU3M6DPBA [following]
--2022-01-30 21:27:52-- https://releases-cdn.jfrog.io/filestore/3a/3a3f59bf097fcfa7467545c8446e03c55fff16c?response-c-
ntent-type=application/x-gzip&response-content-disposition=attachment%3Bfilename%3D%22jfrog-platform-trial-prox-7.31.13-
compose.tar.gz%22&x-jf-traceId=c2e107e4b72da593&X-Artifactory-repositoryKey=jfrog-prox&X-Artifactory-projectKey=default-
Expires=1643558331&Signature=0A4kw1gd1l0T1GusgclU8K~tT8AHk5-LikEHx6xYw09xysgQZnIZeY7S6GsUGi0ReAe-oswqjXXrZCuFb4JtHtGtG-
NHFc-Kc900KE55kZetloihNQo90B4iKv8zPe2v2XTivmYvoB31iGGlfvs200BoUS2mEbbD8ljACpOPQ70jzbjTAE5YemaqRNfh3fNt3LNVYOs8Ju5CptN3s-
MCPSK851Pc-~bk4bsB0d-P25AcCpmzhI7D0kr4Czs5Alt1Zqguf6jsvVh~CKSjjtmk1d1HS-jJ6CBwnHdwAdwdJg-JoXaIs72Y7AZU2NjktivK~p7JN2KKg-
Jusg2ShKurQ_&Key-Pair-Id=APKAJ6NHFWMVU3M6DPBA
Resolving releases-cdn.jfrog.io (releases-cdn.jfrog.io)... 54.230.90.2, 54.230.90.30, 54.230.90.56, ...
Connecting to releases-cdn.jfrog.io (releases-cdn.jfrog.io)|54.230.90.2|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8599308 (8.2M) [application/x-gzip]
Saving to: 'jfrog-compose-installer.tar.gz'

jfrog-compose-installer.tar.g 100%[=====] 8.20M 16.6MB/s in 0.5s
```

# Installation

```
eswaribala@WINDOWS-BVQNF6J: ~ - □ X

2022-01-30 21:27:52 (16.6 MB/s) - 'jfrog-compose-installer.tar.gz' saved [8599308/8599308]

eswaribala@WINDOWS-BVQNF6J:~$ tar -xvzf jfrog-compose-installer.tar.gz
jfrog-platform-trial-prox-7.31.13-compose/third-party/logrotate/logrotate
jfrog-platform-trial-prox-7.31.13-compose/third-party/yq/yq_linux
jfrog-platform-trial-prox-7.31.13-compose/third-party/yq/LICENSE
jfrog-platform-trial-prox-7.31.13-compose/third-party/yq/yq_darwin
jfrog-platform-trial-prox-7.31.13-compose/bin/dockerComposeHelper.sh
jfrog-platform-trial-prox-7.31.13-compose/templates/
jfrog-platform-trial-prox-7.31.13-compose/templates/docker-compose-rabbitmq.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/docker-compose-all.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/docker-compose-postgres.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/docker-compose.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/xray/system.basic-template.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/xray/system.full-template.yaml
jfrog-platform-trial-prox-7.31.13-compose/third-party/rabbitmq/
jfrog-platform-trial-prox-7.31.13-compose/third-party/rabbitmq/rabbitmq.conf
jfrog-platform-trial-prox-7.31.13-compose/third-party/rabbitmq/setRabbitCluster.sh
jfrog-platform-trial-prox-7.31.13-compose/templates/artifactory/system.basic-template.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/artifactory/system.full-template.yaml
jfrog-platform-trial-prox-7.31.13-compose/README.md
jfrog-platform-trial-prox-7.31.13-compose/config.sh
jfrog-platform-trial-prox-7.31.13-compose/third-party/postgresql/createPostgresUsers.sh
jfrog-platform-trial-prox-7.31.13-compose/bin/systemYamlHelper.sh
jfrog-platform-trial-prox-7.31.13-compose/.env
eswaribala@WINDOWS-BVQNF6J:~$
```

# Installation

```
eswaribala@WINDOWS-BVQNF6J: ~/jfrog-platform-trial-prox-7.31.13-compose
jfrog-platform-trial-prox-7.31.13-compose/templates/docker-compose-all.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/docker-compose-postgres.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/docker-compose.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/xray/system.basic-template.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/xray/system.full-template.yaml
jfrog-platform-trial-prox-7.31.13-compose/third-party/rabbitmq/
jfrog-platform-trial-prox-7.31.13-compose/third-party/rabbitmq/rabbitmq.conf
jfrog-platform-trial-prox-7.31.13-compose/third-party/rabbitmq/setRabbitCluster.sh
jfrog-platform-trial-prox-7.31.13-compose/templates/artifactory/system.basic-template.yaml
jfrog-platform-trial-prox-7.31.13-compose/templates/artifactory/system.full-template.yaml
jfrog-platform-trial-prox-7.31.13-compose/README.md
jfrog-platform-trial-prox-7.31.13-compose/config.sh
jfrog-platform-trial-prox-7.31.13-compose/third-party/postgresql/createPostgresUsers.sh
jfrog-platform-trial-prox-7.31.13-compose/bin/systemYamlHelper.sh
jfrog-platform-trial-prox-7.31.13-compose/.env
eswaribala@WINDOWS-BVQNF6J:~/jfrog-platform-trial-prox-7.31.13-compose$ sudo ./config.sh
```

Beginning JFrog Platform Trial Pro X setup

Validating System requirements

Installation Directory found in .env. Skipping prompt

Creating third party directories (if necessary)

Attempting to seed PostgreSQL. This may take some time.

# Installation

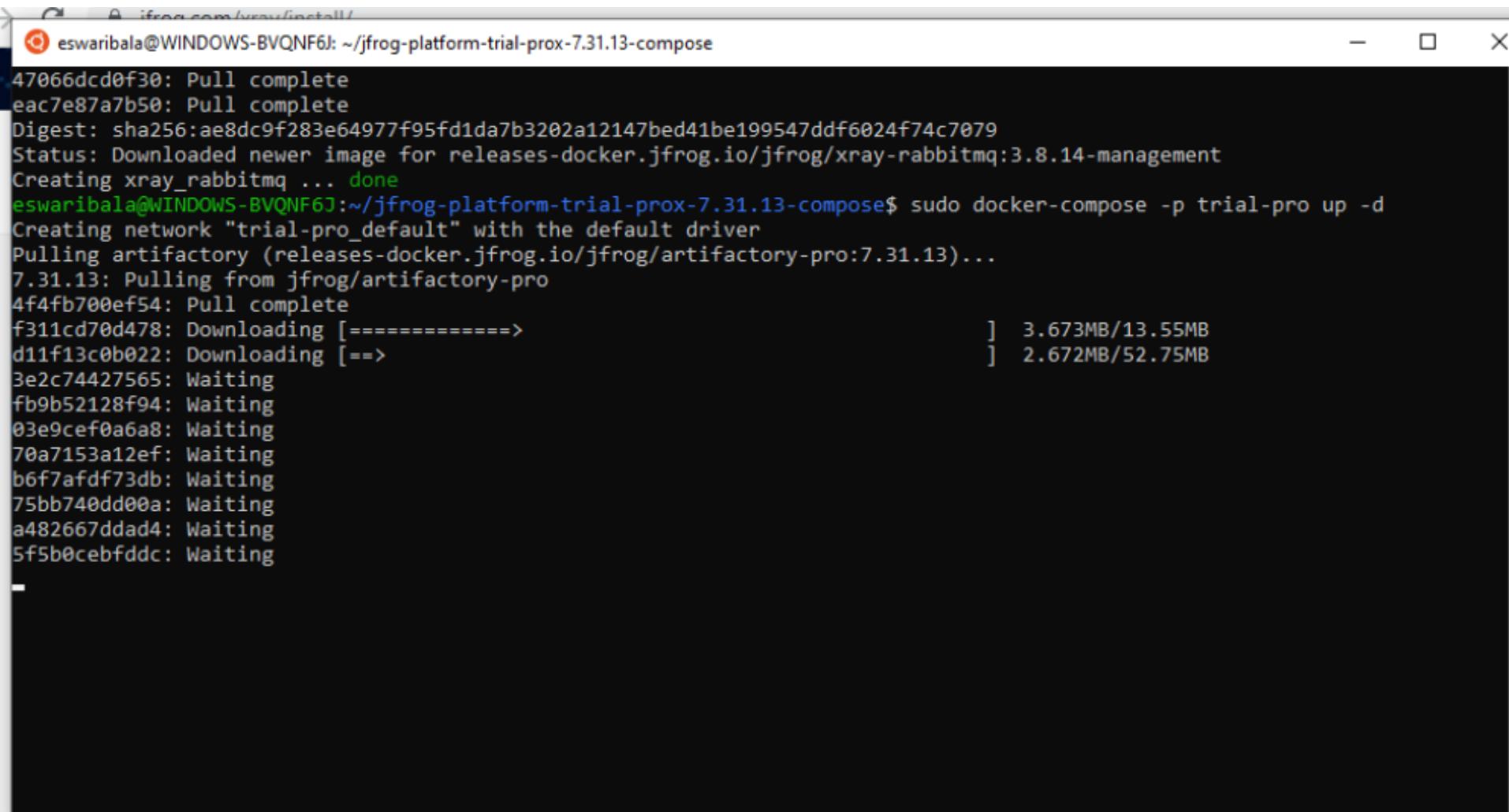
```
eswaribala@WINDOWS-BVQNF6J: ~/jfrog-platform-trial-prox-7.31.13-compose
start: docker-compose -p trial-prox up -d
stop:  docker-compose -p trial-prox down

NOTE: The compose file uses several environment variables from the .env file. Remember to run from within the [/home/eswaribala/jfrog-platform-trial-prox-7.31.13-compose] folder

Done

eswaribala@WINDOWS-BVQNF6J:~/jfrog-platform-trial-prox-7.31.13-compose$ sudo docker-compose -p trial-pro-rabbitmq -f docker-compose-rabbitmq.yaml up -d
Creating network "trial-pro-rabbitmq_default" with the default driver
Pulling rabbitmq (releases-docker.jfrog.io/jfrog/xray-rabbitmq:3.8.14-management)...
3.8.14-management: Pulling from jfrog/xray-rabbitmq
92dc2a97ff99: Pull complete
be13a9d27eb8: Pull complete
c8299583700a: Pull complete
541e793d3e49: Pull complete
e4cea239b885: Pull complete
0737091b6222: Pull complete
2bb945602fa2: Pull complete
1b21b4c227d6: Pull complete
a72f379f6ff9: Pull complete
aa2c51deec02: Pull complete
5d5f03b8bd44: Pull complete
47066dcd0f30: Pull complete
eac7e87a7b50: Pull complete
Digest: sha256:ae8dc9f283e64977f95fd1da7b3202a12147bed41be199547ddf6024f74c7079
Status: Downloaded newer image for releases-docker.jfrog.io/jfrog/xray-rabbitmq:3.8.14-management
Creating xray_rabbitmq ...
```

# Installation



A screenshot of a terminal window titled "ifrog.com (www/install)". The window shows a command-line session for installing JFrog Platform using Docker Compose. The session starts with pulling images from Docker Hub, then creating a network, and finally pulling the Artifactory image. The terminal uses a dark theme with white text and a black background.

```
eswaribala@WINDOWS-BVQNF6J: ~/jfrog-platform-trial-prox-7.31.13-compose
47066dc0f30: Pull complete
eac7e87a7b50: Pull complete
Digest: sha256:ae8dc9f283e64977f95fd1da7b3202a12147bed41be199547ddf6024f74c7079
Status: Downloaded newer image for releases-docker.jfrog.io/jfrog/xray-rabbitmq:3.8.14-management
Creating xray_rabbitmq ... done
eswaribala@WINDOWS-BVQNF6J:~/jfrog-platform-trial-prox-7.31.13-compose$ sudo docker-compose -p trial-pro up -d
Creating network "trial-pro_default" with the default driver
Pulling artifactory (releases-docker.jfrog.io/jfrog/artifactory-pro:7.31.13)...
7.31.13: Pulling from jfrog/artifactory-pro
4f4fb700ef54: Pull complete
f311cd70d478: Downloading [=====] 3.673MB/13.55MB
d11f13c0b022: Downloading [==>] 2.672MB/52.75MB
3e2c74427565: Waiting
fb9b52128f94: Waiting
03e9cef0a6a8: Waiting
70a7153a12ef: Waiting
b6f7afdf73db: Waiting
75bb740dd00a: Waiting
a482667ddad4: Waiting
5f5b0cebfddc: Waiting
```

# Installation

```
eswaribala@WINDOWS-BVQNF6J:~/jfrog-platform-trial-prox-7.31.13-compose
b4036b631195: Already exists
a7eebf021c6f: Already exists
0706d11bd5d4: Pull complete
8d3a4aea578d: Pull complete
c670edb2d784: Pull complete
db29f8c0cac3: Pull complete
7472afe9253c: Pull complete
1fef2c627a39: Pull complete
Digest: sha256:f875939997a198a5b0ae1474514475eac3cedd82e39635389c4050824d080336
Status: Downloaded newer image for releases-docker.jfrog.io/jfrog/xray-indexer:3.37.2
Pulling xray-analysis (releases-docker.jfrog.io/jfrog/xray-analysis:3.37.2)...
3.37.2: Pulling from jfrog/xray-analysis
4f4fb700ef54: Already exists
098a109c8679: Already exists
b4036b631195: Already exists
a7eebf021c6f: Already exists
a536d086d9df: Pull complete
66d61c957112: Pull complete
6fccaa6ae9a21: Pull complete
0f27f4fd705d: Pull complete
0665c978d0e0: Pull complete
945e9fea34bb: Pull complete
Digest: sha256:0b56e68d5141b90c2f39c978bb8e1328872e221c4fb7b4050245814b1d55e262
Status: Downloaded newer image for releases-docker.jfrog.io/jfrog/xray-analysis:3.37.2
Pulling xray-persist (releases-docker.jfrog.io/jfrog/xray-persist:3.37.2)...
3.37.2: Pulling from jfrog/xray-persist
4f4fb700ef54: Already exists
098a109c8679: Already exists
b4036b631195: Already exists
a7eebf021c6f: Already exists
e338513a42f9: Pull complete
833a62c8d916: Pull complete
0ff3b8952b03: Pull complete
214aafeb37fc: Pull complete
bf396ab811e4: Pull complete
26d86364e4d7: Pull complete
Digest: sha256:50b4c9aed791a8093dc51dd037815b77115f282186a7f282a33a48b825d3e465
Status: Downloaded newer image for releases-docker.jfrog.io/jfrog/xray-persist:3.37.2
Creating xray_router ... done
Creating artifactory ... done
Creating xray_analysis ... done
Creating xray_persist ... done
Creating xray_observability ... done
Creating xray_indexer ... done
Creating xray_server ... done
eswaribala@WINDOWS-BVQNF6J:~/jfrog-platform-trial-prox-7.31.13-compose$
```

Activate Windows  
Go to Settings to activate Windows.

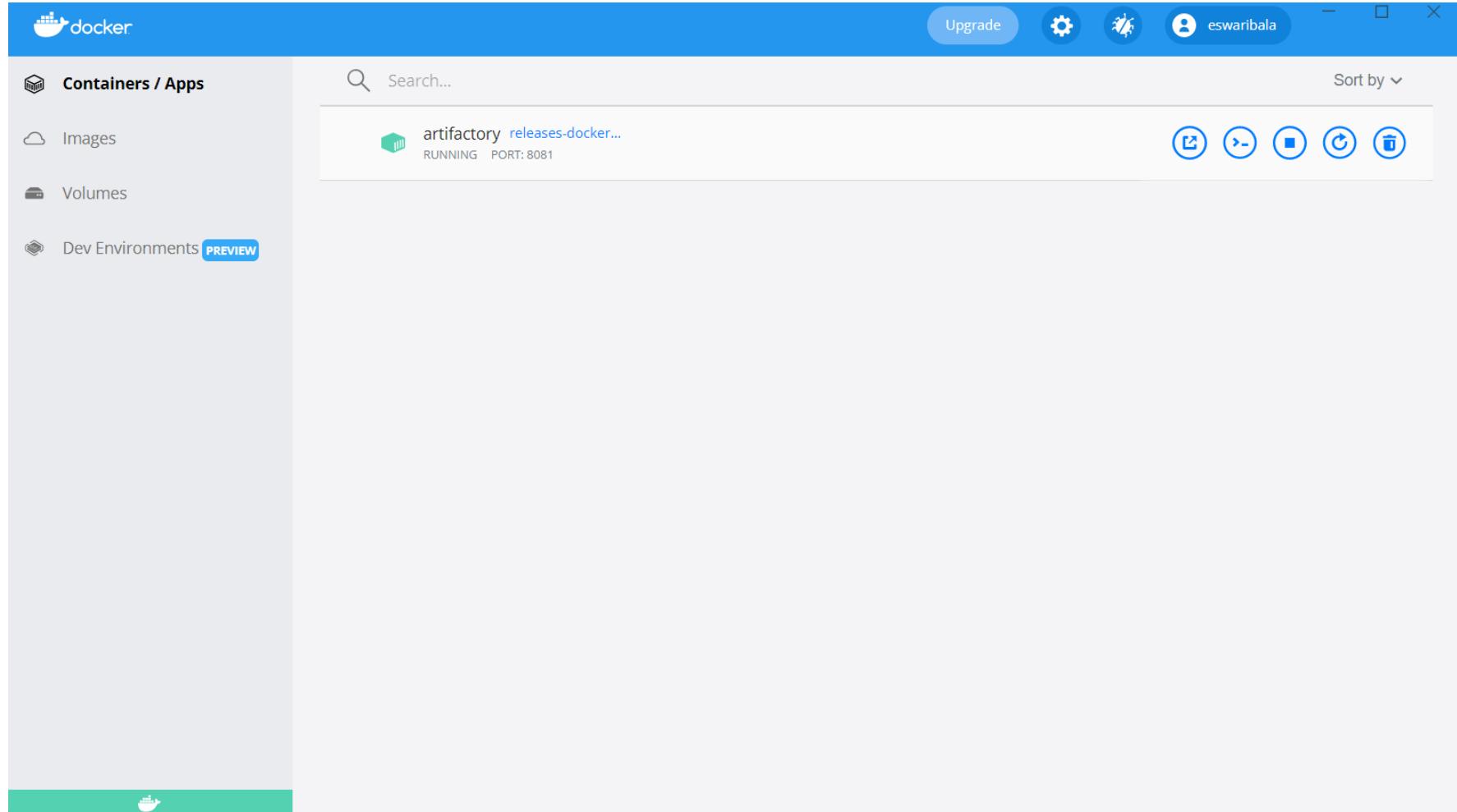
# Installation

The screenshot shows the Docker desktop application interface. The left sidebar has options: Containers / Apps (selected), Images, Volumes, and Dev Environments (PREVIEW). The main area shows a list of running containers:

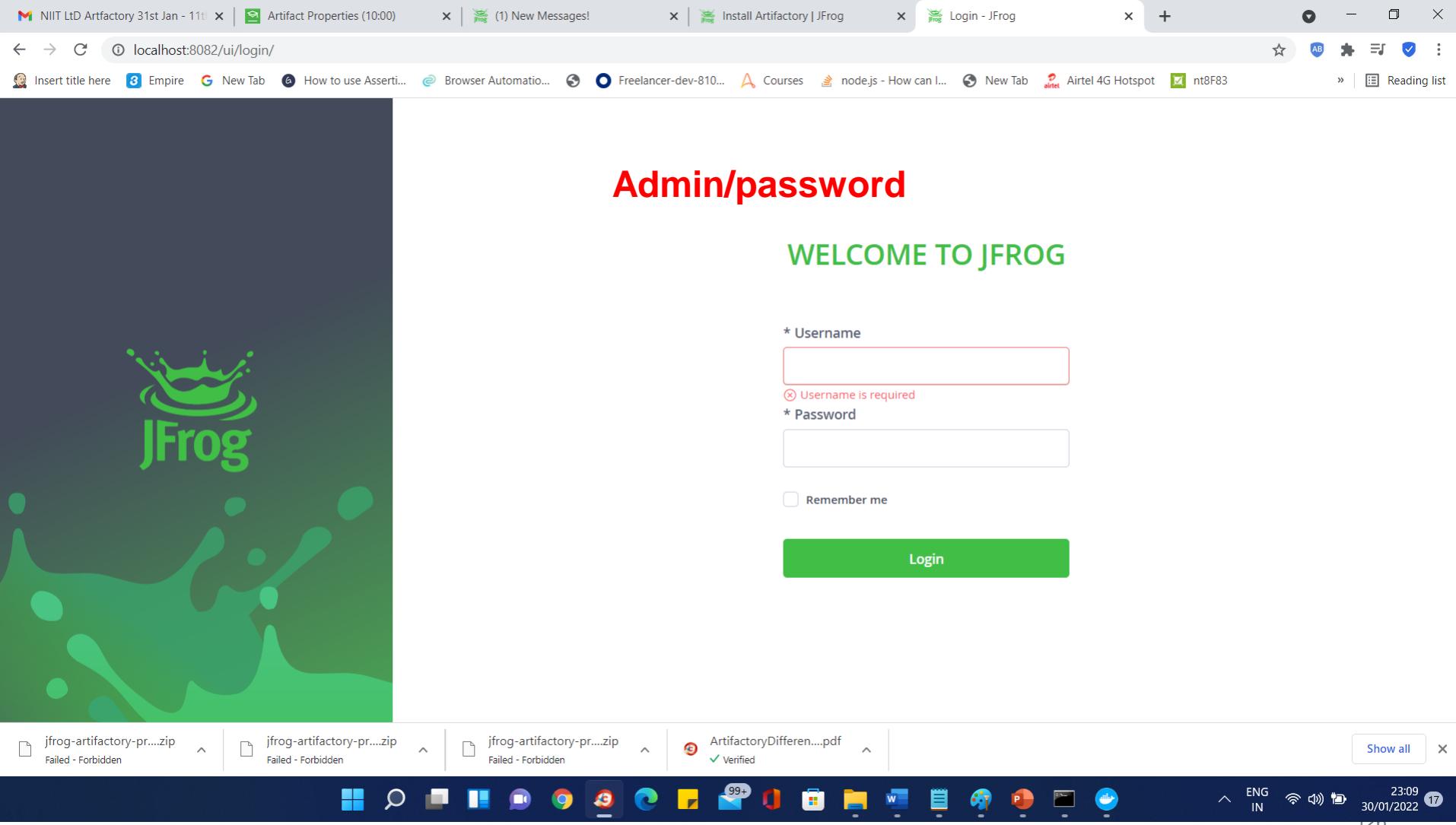
- trial-prox-postgres (RUNNING)
- trial-pro-rabbitmq (RUNNING)
- trial-pro (RUNNING)
  - artifactory (releases-docker...) (RUNNING PORT: 8081)
  - xray\_router (releases-docker...) (RUNNING PORT: 8083)
  - xray\_observability (releases-docker...) (RUNNING)
  - xray\_analysis (releases-docker...) (RUNNING)
  - xray\_indexer (releases-docker...) (RUNNING)
  - xray\_persist (releases-docker...) (RUNNING)
  - xray\_server (releases-docker...) (RUNNING)

At the top right are buttons for Upgrade, Settings, Help, and Sign in. At the bottom right are icons for Stop, Start, Refresh, and Delete. A message at the bottom right says "Activate Windows Go to Settings to activate Windows." The bottom navigation bar includes icons for Home, Images, and Containers.

# Installation using Jfrog artifactory for Windows



# Installation using Jfrog artifactory for Windows



Artifact Properties (10:00) | (1) New Messages! | Install Artifactory | JFrog | Login - JFrog

localhost:8082/ui/login/

Insert title here Empire New Tab How to use Asserti... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83

Reading list

## Admin/password

### WELCOME TO JFROG

\* Username

(\*) Username is required

\* Password

Remember me

Login

jfrog-artifactory-pr....zip Failed - Forbidden jfrog-artifactory-pr....zip Failed - Forbidden jfrog-artifactory-pr....zip Failed - Forbidden ArtifactoryDifferen....pdf Verified

Show all

ENG IN 23:09 30/01/2022 17

# Installation using Jfrog artifactory for Windows

The screenshot shows the JFrog Artifactory administration interface. The left sidebar has a green header 'Administration' with options: Projects, Repositories, Identity and Access, Security, Licenses (selected), and Licenses. Below this is a 'Getting Started' button. The main content area has a search bar 'Search admin resources'. Under 'Licenses', there's a field labeled '\* Enter your license key below or drop a file' containing a very long, complex license key. At the bottom right are 'Reset' and 'Save' buttons. The status bar at the bottom shows system icons and the date '31/01/2022'.

License from official mail

```
chJvZHVjdHM6CiAgYXJ0aWZhY3Rvcnk6CiAgICBwcm9k  
dWN0oiBaWGh3YVhKbGN6b2dNakF5TWkw  
d01pMH1PR1F4T1RvMU5Eb3pNRm9LYVdRNk1EWTFaVEZr  
TURSaExXSXlObVF0TkRaalpDMDVORE5o  
TFRZeFpqVXdNRGs1WldNM05ncHzkMjVsY2pvZ1VsQ1RD  
bkJ5YjNCbGNuUnBaWE02Q21BZ2NtVm5h  
WE4wY21GMGFVOXVTM1Y1Vkc5clpXNDZJR1Y1U2pKYVdF  
bHBUMmxKZVVscGQybGtTR3gzU1dwdmFW  
TnNaR1ZKYVhkcFdWZDRia2xxYjJsVmJFMTVUbFJaYVV4  
RFNuSmhWMUZwVDJss2VrMXFXbEpPVkdo  
MVRWaHdOV1ZxWXpGVFJrSktXV3RXTUdOclNsVk1WMDVP  
VTBjMVVsTnFXa1ZrUmxweFZET1NNM1ZW  
T1c5bFdGRjNTVzR3TG1WNVNteGxTRkZwVDJsS04xaERT  
bmhxhVjJSd116T1N1VmxBzVW5CaU1qVlZZ  
akowYkdKc1VqVmpSMVpqU1dwd1kwbHNRalpWYXpGQ1ZH  
dFdUMVpHZDJsTVJuZHBZMjFXYmlGWVRq
```

# Jfrog Artifactory Cloud Quick Repo Setup

The screenshot shows the Jfrog Artifactory Cloud interface. The main focus is a modal window titled "Set Me Up". Inside the modal, there are three dropdown menus: "Package Type" set to "Docker", "Repository" set to "default-docker-virtual", and "Client" set to "Docker". Below these, there are three tabs: "Configure" (which is selected), "Deploy", and "Resolve". Under the "Configure" tab, there is a "Login" section with instructions: "In your Terminal, Run the following command. when prompted - provide your username(parameswaribala@gmail.com) and API Key". A code editor-like area shows the command: "1 docker login vbrepo.jfrog.io". At the bottom right of the modal is a green "Docker login" button and a link to "~/docker/config.json". The background shows the Artifactory sidebar with options like Application, Artifactory, Quick Setup, Packages, Builds, Artifacts, Pipelines, and Learning Center. The bottom navigation bar shows several file tabs and system status icons.

## Jfrog X-Ray

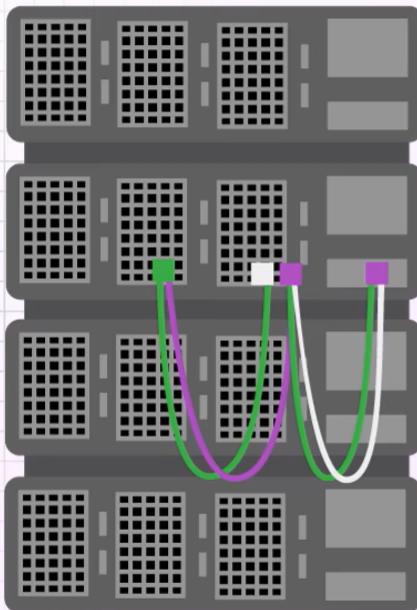
---

- <https://jfrog.com/download-jfrog-platform/#xray>
- Download docker-compose file
- Try this, Check mail for link
- <https://jfrog.com/xray/install/>

# Installation



## NETWORK CONFIGURATION



1. Set server name
2. Include base URL
3. Define outbound proxies
4. Setup reverse proxy (*recommended for Tomcat*)
5. Set notification preferences

# Installation



## JVM TUNING



The Java Virtual Machine is a key component of the Artifactory system process.



# Installation



## JVM TUNING



Four parameters

-Xmx

-Xms

-Xss

-XX:+UseG1GC

# Installation



## JVM TUNING

| Parameter    | Meaning           | Recommendations                      |
|--------------|-------------------|--------------------------------------|
| -Xmx         | Maximum heap size | Should be big enough (e.g. 4 Gb)     |
| -Xms         | Initial heap size | Should at least $\frac{1}{2}$ of Xmx |
| -Xss         | Stack size        | Can be small (e.g. 128 Kb)           |
| -XX:+UseG1GC | GC to use         | Use Garbage One collector            |

# Installation

---



## PLUGGABLE STORAGE IMPLEMENTATION

Last consideration: binary and metadata store

Store binaries in Artifactory's default file system  
and use a database solution to store metadata.

# Installation

---



## PLUGGABLE STORAGE IMPLEMENTATION

Standard DB (JDBC Driver Required)

Derby – default (replace if you plan to scale)

MySQL

Oracle

MySQL Server

PostgreSQL

# Installation



## PLUGGABLE STORAGE IMPLEMENTATION

For binary storage: (3 options)

Standard DB (JDBC Driver Required)

Oracle

MS SQL Server

PostgreSQL (up to 100s GB)

# Installation

---



## PLUGGABLE STORAGE IMPLEMENTATION

For binary storage: (3 options)

S3 (Cloud Filestore)

S3

Google Cloud Storage

Ceph

Swift

Local Domains

Any BLOBstore with S3 API

# Installation



## S3 INSTALLATION

Artifactory SaaS

Artifactory SaaS



ARTIFACTORY ESSENTIALS | INSTALLATION AND SETUP

Amazon S3

Google Cloud Platform

Microsoft Azure

# Application Module

Topology - JFrog Application Module - JFrog - JFrog localhost:8082/ui/admin/mission\_control/topology Insert title here Empire New Tab How to use Assert... Browser Automation... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list Welcome, admin

JFrog Platform All Packages Search packages with wildca ? Welcome, admin

Application Dashboard Topology Trends Artifactory Distribution Pipelines Getting Started

Trial license 7.31.10 rev 73110900 Licensed to RPS © Copyright 2022 JFrog Ltd

## Manage Your Multi Site Topology

Read More >

Topology

Tags Search Show Replication

Number of Platform Deployments nearby

- All of the Services are Offline
- Some of the Services are online
- All of the Services are online
- Unauthorized
- Unavailable

HOME1

active running

Service Status

4.1GB Storage

✓ Replication

ENTERPRISE\_PLUS\_TRIAL License

4 Services

ENG IN 18:46 31/01/2022

# Application Module

---

- Dashboard: The Dashboard provides in-depth insight into the health and performance of your JFrog Platform Deployments (JPDs). Requires an Enterprise+ license.
- Packages: View detailed information about packages and their package versions regarding the number of downloads and vulnerability scan status. Xray scanning requires Artifactory Pro X, Enterprise with Xray, or an Enterprise+ license.
- Builds: View information about your builds including build time, build status and Xray scan status with an option to delete builds if necessary. Xray scanning requires Artifactory Pro X, Enterprise with Xray, or an Enterprise+ license.
- Artifacts: View, deploy, browse and scan your artifacts in two ways either through the Tree Browser or Simple browser. Xray scanning requires Artifactory Pro X, Enterprise with Xray, or an Enterprise+ license.

# Application Module

---

- Distribution: Provision your Release Bundles and their distribution processes, including release content and track your distributed Release Bundles on the Edge instances. Requires an Enterprise+ license.
- Pipelines: Build, test, and deploy software as part of a system of continuous integration and continuous delivery (CI/CD) by creating smart, event-driven workflows across your tools and teams to help you release applications faster. Requires an Enterprise+ license.
- Security and Compliance: Configure policies and assign them to the watches allowing Xray to scan for security vulnerabilities and license compliance of your package, builds and artifacts in the JFrog Platform. Xray scanning requires Artifactory Pro X, Enterprise with Xray, or an Enterprise+ license.

# Jfrog Artifactory Cloud Quick Repo Setup

```
C:\WINDOWS\system32>docker login vebrepo.jfrog.io
Username: parameswaribala@gmail.com
Password:
Login Succeeded

C:\WINDOWS\system32>
```



23:37  
ENG IN 30/01/2022

# Jfrog Artifactory Cloud Quick Repo Setup

The screenshot shows the Jfrog Artifactory Platform interface. On the left, there's a sidebar with 'Application' selected, containing links for Artifactory, Quick Setup, Packages, Builds, Artifacts, and Pipelines. Below that is the 'Learning Center'. At the bottom left, it says 'JFrog Cloud © Copyright 2022 JFrog Ltd'.

The main area has tabs for 'All', 'Packages', and a search bar. A green 'UPGRADE' button is visible. The top navigation bar includes tabs for 'Artifact Properties', 'Install Artifactory | JFrog', 'JFrog', and a video player tab for 'How to set up a Private Docker R...'. The address bar shows the URL 'vebrepo.jfrog.io/ui/quick-setup'.

A central modal window titled 'Set Me Up' is open. It has three tabs: 'Package Type' (set to 'Docker'), 'Repository' (set to 'default-docker-virtual'), and 'Client' (set to 'Docker'). Below these tabs are three buttons: 'Configure' (disabled), 'Deploy' (selected and underlined), and 'Resolve'. The 'Deploy' section contains two numbered steps:

1. Tag your image using the following command (replace <IMAGE ID>, <DOCKER REPOSITORY> and <DOCKER TAG> with your values)  

```
1 docker tag <IMAGE_ID> vebrepo.jfrog.io/default-docker-virtual/<DOCKER_REPOSITORY>:<DOCKER_TAG>
```
2. Push your image using the following command  

```
1 docker push vebrepo.jfrog.io/default-docker-virtual/<DOCKER_REPOSITORY>:<DOCKER_TAG>
```

At the bottom of the modal, there are icons for Alpine, Bower, Cargo, Chef, CocoaPods, and Composer. The taskbar at the bottom shows several file tabs: 'RTF20-Usage-3001....pdf' (Verified), 'RTF20-Usage-3001....pdf' (Verified), 'RTF20-Usage-3001....pdf' (Verified), 'jfrog-artifactory-pr....zip' (Failed - Forbidden), 'jfrog-artifactory-pr....zip' (Failed - Forbidden), and 'jfrog-artifactory-pr....zip' (Failed - Forbidden). The system tray at the bottom right shows the date '30/01/2022', time '23:38', and battery level '17%'. A green speech bubble icon is also present in the bottom right corner.

# Jfrog Identity (Authentication)

The screenshot shows the JFrog Identity (Authentication) interface. The top navigation bar includes tabs for 'Key Features of JFrog Artifactory' and 'JFrog'. The URL in the address bar is 'localhost:8082/ui/admin/management/users'. The main header has a search bar 'Search admin resources' and a 'Welcome, admin' message. On the left, a sidebar under 'Administration' lists 'Projects', 'Repositories', 'Identity and Access' (selected), 'Users' (highlighted in green), 'Groups', and 'Global Roles'. A 'Getting Started' section is also present. The central content area is titled 'Users' and contains a table with columns: Name, Email, Realm, Related..., Admin, Status, and Last Log...'. The table shows three users: 'admin' (Email: internal, Admin checked, Last Log: 31-01-22 ...), 'anonymous' (Email: internal, Admin unchecked, Last Log: X), and 'parmeswari' (Email: parmesw..., Realm: internal, Admin checked, Last Log: 1 | re...). There are 'Delete' and 'Unlock' buttons at the top right of the table. The bottom of the page shows a footer with the JFrog logo, trial license information, and a Windows taskbar.

| Name       | Email      | Realm    | Related... | Admin                               | Status | Last Log...  |
|------------|------------|----------|------------|-------------------------------------|--------|--------------|
| admin      | internal   |          |            | <input checked="" type="checkbox"/> |        | 31-01-22 ... |
| anonymous  | internal   |          |            | <input type="checkbox"/>            |        | X            |
| parmeswari | parmesw... | internal | 1   re...  | <input checked="" type="checkbox"/> |        |              |



# Jfrog Repository Quick Setup (from Top Right)

Key Features of JFrog Artifactory x JFrog

localhost:8082/ui/admin/management/users

Insert title here Empire New Tab How to use Asserti... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

JFrog Platform All Search admin resources Welcome, admin

+ New user

Administration

Projects

Repositories

Identity and Access

Users

Groups

Global Roles

Getting Started

Trial license 7.31.10 rev 73110900 Licensed to RPS  
© Copyright 2022 JFrog Ltd

Create Repositories

Select the package type(s) you want - we'll create the default repositories for you!

Search by package type

Alpine Bower Chef CocoaPods Composer Conan Conda CRAN

Debian Docker Gems Generic GitLfs Go Gradle Helm

Next

Last Logi...  
31-01-22 ...

# Jfrog Repository Quick Setup (from Top Right)

The screenshot shows the JFrog Artifactory administration interface. A modal window titled "Create Repositories" is open in the foreground. The modal has a "Repositories prefix" input field containing "deb" with a character limit of 3/20. Below the input are three repository options: "Debian" (Local Repository), "Remote Repository" (with URL <http://archive.ubuntu.c...>), and "Virtual Repository" (with identifier "deb-debian"). At the bottom of the modal are "Back" and "Create" buttons. The background shows the "Management" section of the admin interface, including a search bar, a user list, and a sidebar with links like "Administration", "Projects", "Repositories", and "Identity and Access".

Key Features of JFrog Artifactory x JFrog

localhost:8082/ui/admin/management/users

Insert title here Empire New Tab How to use Asserti... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

Welcome, admin

+ New user

Delete Unlock

Last Logi...

31-01-22 ...

Getting Started

Trial license 7.31.10 rev 73110900 Licensed to RPS © Copyright 2022 JFrog Ltd

ENG IN 99+ 31/01/2022

# Jfrog Maven Repository)

Spring Boot Application integrate x JFrog

localhost:8082/ui/admin/repositories/local/veb-libs-release-local/edit

Insert title here Empire New Tab How to use Assert... Browser Automation... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83

Reading list

JFrog Platform All Search admin resources Welcome, admin

Administration

Projects

Repositories

Repositories

Layouts

Identity and Access

Security

Licenses

General

Getting Started

Trial license 7.31.10 rev 73110900 Licensed to RPS © Copyright 2022 JFrog Ltd

Edit veb-libs-release-local

Basic Advanced Replications

Maven

\* Repository Key ?  
veb-libs-release-local

General Settings Include / Exclude Patterns

Cancel Save

settings.xml settings.xml Verified Show all

00:44 ENG IN 01/02/2022 147

# Jfrog Maven (clean install deploy –DskipTests)

```
C:\Windows\System32\cmd.exe
Uploading to snapshots: http://localhost:8081/artifactory/veb-libs-snapshot-local/com/eric/ecommerce/0.0.1-SNAPSHOT/ecommerce-0.0.1-20220131.191139-1.pom
Uploaded to snapshots: http://localhost:8081/artifactory/veb-libs-snapshot-local/com/eric/ecommerce/0.0.1-SNAPSHOT/ecommerce-0.0.1-20220131.191139-1.pom (4.6 kB at 20 kB/s)
Downloading from snapshots: http://localhost:8081/artifactory/veb-libs-snapshot-local/com/eric/ecommerce/maven-metadata.xml
Downloaded from snapshots: http://localhost:8081/artifactory/veb-libs-snapshot-local/com/eric/ecommerce/maven-metadata.xml (370 B at 5.4 kB/s)
Uploading to snapshots: http://localhost:8081/artifactory/veb-libs-snapshot-local/com/eric/ecommerce/0.0.1-SNAPSHOT/maven-metadata.xml
Uploaded to snapshots: http://localhost:8081/artifactory/veb-libs-snapshot-local/com/eric/ecommerce/0.0.1-SNAPSHOT/maven-metadata.xml (767 B at 11 kB/s)
Uploading to snapshots: http://localhost:8081/artifactory/veb-libs-snapshot-local/com/eric/ecommerce/maven-metadata.xml
Uploaded to snapshots: http://localhost:8081/artifactory/veb-libs-snapshot-local/com/eric/ecommerce/maven-metadata.xml (313 B at 3.4 kB/s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  18.815 s
[INFO] Finished at: 2022-02-01T00:41:42+05:30
```



00:41  
ENG IN 01/02/2022

# Jfrog Maven (clean install deploy –DskipTests)

The screenshot shows the Jfrog Artifactory interface. The left sidebar has 'Artifactory' selected under 'Artifacts'. The main area displays a tree view of artifacts. On the right, a detailed view of the 'ecommerce' artifact from the 'vеб-libs-snapshot-local/com/eric/ecommerce' repository is shown. The artifact details include:

| General                                              | Effective Permissions                                        | Properties                                                                   | Followers          |
|------------------------------------------------------|--------------------------------------------------------------|------------------------------------------------------------------------------|--------------------|
| Name: ecommerce                                      | Repository Path: vеб-libs-snapshot-local/com/eric/ecommerce/ | URL to file: http://localhost:8082/artifactory/vеб-libs-snapshot-local/co... | Deployed By: admin |
| Artifact Count / Size: Show                          |                                                              |                                                                              |                    |
| Created: 31-01-22 19:11:41 +00:00 (0d 0h 4m 35s ago) |                                                              |                                                                              |                    |

At the bottom, there are tabs for 'settings.xml' (Verified) and 'Show all'.

# Docker Hub

How to set up a Private Docker Registry JFrog

localhost:8082/ui/admin/repositories/remote/veb-docker-remote/edit

Insert title here Empire New Tab How to use Assert... Browser Automation Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

JFrog Platform All Search admin resources Welcome, admin

Administration Projects Repositories

Repositories Layouts Identity and Access Security Licenses General Getting Started

Trial license 7.31.10 rev 73110900 Licensed to RPS © Copyright 2022 JFrog Ltd

Edit veb-docker-remote

Basic Advanced Replications

https://registry-1.docker.io/

User Name eswaribala Password / Access Token ···

SSL / TLS Certificate Select Test

General Docker Settings

Cancel Save

settings.xml Verified settings.xml Verified Show all

## Jfrog Cli

---

- docker run releases-docker.jfrog.io/jfrog/jfrog-cli-full-v2-jf jf -v
- Or
- npm install -g jfrog-cli-v2-jf
- Or
- <https://jfrog.com/getcli/>
  
- Installed using powershell
- <https://www.jfrog.com/confluence/display/CLI/CLI+for+JFrog+Artifactory>

# Jfrog Cli

Administrator: Command Prompt

```
G:\Local disk\JFrog>jf rt u jfrog.txt veb-generic-local
Log path: C:\Users\Balasubramaniam\.jfrog\logs\jfrog-cli.2022-02-02.03-47-07.2580.log
{
  "status": "success",
  "totals": {
    "success": 1,
    "failure": 0
  }
}

G:\Local disk\JFrog>■
```



^ ENG IN 03:47  
02/02/2022

# Jfrog Cli

Administrator: Command Prompt

```
G:\Local disk\JFrog>jf rt dl veb-generic-local
Log path: C:\Users\Balasubramaniam\.jfrog\logs\jfrog-cli.2022-02-02.03-48-44.15684.log
{
  "status": "success",
  "totals": {
    "success": 1,
    "failure": 0
  }
}

G:\Local disk\JFrog>_
```



03:48 02/02/2022  
ENG IN 10

# Jfrog Cli

```
Select Administrator: Command Prompt
HTTP Status 404 - Not Found
und</h1></body></html>
G:\Local disk\JFrog>jf project init
03:55:31 [Info] Repository http://localhost:8081/artifactory/default-maven-local created.
03:55:32 [Info] Repository http://localhost:8081/artifactory/default-maven-remote created.
03:55:32 [Info] Repository http://localhost:8081/artifactory/default-maven-virtual created.
```

This project is initialized!

The project config is stored inside the .jfrog directory.

Audit your project for security vulnerabilities by running  
jf audit

Scan any software package on this machine for security vulnerabilities by running  
jf scan path/to/dir/or/package

If you're using VS Code, IntelliJ IDEA, WebStorm, PyCharm, Android Studio or GoLand

1. Open the IDE
2. Install the JFrog extension or plugin
3. View the JFrog panel

Build the code & deploy the packages by running  
jf mvn install deploy

Publish the build-info to Artifactory  
jf rt bp

Read more using this link:

<https://github.com/jfrog/jfrog-cli/blob/v2/guides/getting-started-with-jfrog-using-the-cli.md>

Your feedback is important.

We'd love to get your feedback and answer any questions you may have.



# Jfrog Jenkins Pipeline

---

- The main steps for creating and running a pipeline are as follows:
  - Create a Node Pool
  - Add Integrations
  - Create Pipeline DSL
  - Add a Pipeline Source
  - Run Pipeline

## Jfrog Jenkins Pipeline

---

- <https://www.jfrog.com/confluence/display/JFROG/Managing+Pipelines+Node+Pools>

# Jfrog Jenkins Pipeline

The screenshot shows the JFrog Platform interface. On the left, there's a sidebar with 'Application' selected, containing links for Dashboard, Artifactory (Packages, Builds, Artifacts), Distribution, Pipelines, and a 'Getting Started' button. The main content area is titled 'Builds > jfrogpipeline'. It features a table with one row of data:

| Build ID | Status                                                                | Build Time              | Xray status              |
|----------|-----------------------------------------------------------------------|-------------------------|--------------------------|
| 22       | <a href="http://localhost:7070/jo...">http://localhost:7070/jo...</a> | 03-02-22 04:14:27 +0530 | Requires an Xray license |

At the bottom, there are three tabs labeled 'settings (2).xml', 'settings (1).xml', and 'settings.xml', all marked as 'Verified'. The status bar at the bottom right shows 'ENG IN' and the date '03/02/2022'.

# Jfrog Jenkins Pipeline

Builds Details - JFrog   jfrogpipeline #22 [Jenkins]   Inbox (3,271) - parameswaran   jfrogpipeline #22 Console   project-examples/Jenkinsfile   eswaribala/usmsbootcamp   +

localhost:8082/ui/builds/jfrogpipeline/22/1643841867072/published?buildRepo=artifactory-build-info

Insert title here Empire New Tab How to use Assert... Browser Automation Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

JFrog Platform All Builds Search Builds ? Welcome, admin

Builds > jfrogpipeline > 22

jfrogpipeline Pipeline Build Agent Jenkins/2.319.1 Agent eswaribala Principal admin Artifactory Principal

Started: 03-02-22 04:14:27 +0530 | Duration: 36.1 seconds

Published Modules Environment Xray Data Issues Diff Release History VCS Build Info JSON

1 Published Module Filter

| Module Id                            | Type  | Number Of Artifacts | Number Of Dependencies |
|--------------------------------------|-------|---------------------|------------------------|
| com.intel:customerapi:0.0.1-SNAPSHOT | maven | 2                   | 99                     |

settings (2).xml settings (1).xml settings.xml Show all

Trial license 7.31.10 rev 73110900 Licensed to VEB Consulting Pvt.Ltd © Copyright 2022 JFrog Ltd

Windows Taskbar: 04:17 03/02/2022 ENG IN

# Jfrog Log Analytics

---

- Install Sumologic
- Integrate Jfrog Artifactory to Sumologic using new connection

# Jfrog Sumologic Log Analytics

The screenshot shows the Sumologic web interface. The left sidebar has a dark theme with the following navigation items:

- Recently Opened By Me
- RECENTLY RUN SEARCHES: recent\_dataTier=All
- App Catalog
- Manage Data
- Collection (selected)
- Logs
- Metrics
- Monitoring
- Administration
- Help
- JFrogArtifactory (GNO5V0SZEVCW...)

The main content area displays the "Collection" tab of the "Collection" page. The search bar shows "Search for collectors and sources by name or sourceCategory". The table lists the following data:

| Name             | Health  | Type   | Status | Source Category | Sources | Last Hour | Messages |
|------------------|---------|--------|--------|-----------------|---------|-----------|----------|
| HTTP             | Healthy | Hosted |        |                 | 1       |           | 249      |
| JFrogArtifactory | Healthy |        |        |                 |         |           |          |
| HTTP             | Healthy |        |        |                 |         |           |          |

Actions available for each row include Add Source, Edit, and Delete. A "Regenerate URL" link is also present for the JFrogArtifactory entry.

At the bottom right of the main content area, there is an "Activate Windows" message: "Activate Windows Go to Settings to activate Windows." The bottom of the screen shows the Windows taskbar with various pinned icons and a search bar.

# Jfrog Sumologic Log Analytics

The screenshot shows a browser window with the URL `cloud.cdp.rpsconsulting.in/console/#/client/MzlzOTcAYwBteXnxbA==`. The page title is "JFrog". The main content is the "Artifactory System Logs" section from the "Administration" menu. The left sidebar lists "Identity and Access", "Security", "Licenses", "General", "Proxies", "Monitoring", "Support Zone", "SERVICES", and "Artifactory". The "Artifactory" section is currently selected. The right pane displays the logs for "artifactory-service.log". The logs show several INFO-level messages from February 3, 2022, at 19:47:28.396Z, indicating configuration reloads, application context changes, and garbage collection activity.

```
2022-02-03T19:47:28.396Z [jfrt] [INFO] [2c46f94ff5e29c9c] [c.CentralConfigServiceImpl:910] [http-nio-8081-exec-8] - Reloading configuration... old revision 8, new revision 9
2022-02-03T19:47:28.596Z [jfrt] [INFO] [2c46f94ff5e29c9c] [c.CentralConfigServiceImpl:538] [http-nio-8081-exec-8] - New configuration with revision 9 saved.
2022-02-03T19:47:28.596Z [jfrt] [INFO] [2c46f94ff5e29c9c] [factoryApplicationContext:564] [http-nio-8081-exec-8] - Artifactory application context set to NOT READY by reload
2022-02-03T19:47:28.696Z [jfrt] [INFO] [2c46f94ff5e29c9c] [factoryApplicationContext:564] [http-nio-8081-exec-8] - Artifactory application context set to READY by reload
2022-02-03T19:47:28.696Z [jfrt] [INFO] [2c46f94ff5e29c9c] [c.CentralConfigServiceImpl:928] [http-nio-8081-exec-8] - Configuration reloaded.
2022-02-03T19:47:42.227Z [jfrt] [INFO] [c3355f87dfca0a80b] [ager$LogbackConfigWatchDog:147] [logback-watchdog] - Reloaded logback config from: /opt/jfrog/artifactory/var/etc/artifactory/logback.xml.
2022-02-03T20:00:00.001Z [jfrt] [INFO] [2e656d7a53126130] [.s.d.b.s.BinaryServiceImpl:778] [art-exec-336] - Triggering Garbage Collection
2022-02-03T20:00:00.009Z [jfrt] [INFO] [2e656d7a53126130] [.s.d.b.s.g.GarbageCollector:66] [art-exec-336] - Starting GC strategy 'TRASH_AND_BINARIES'
2022-02-03T20:00:00.010Z [jfrt] [INFO] [2e656d7a53126130] [.s.d.b.s.g.GarbageCollector:68] [art-exec-336] - Finished GC Strategy 'TRASH_AND_BINARIES'
```

# Jfrog Rest API

---

- [Download CURL](#)
- <https://curl.se/download.html>

## Jfrog Rest API

---

- Updating Artifactory System Configuration Files
- Artifactory provides a flexible way to configure an existing Artifactory instance using a simple YAML configuration file.
- Run single or multiple configuration changes as needed, to create, update, and delete any elements in Artifactory.

# Jfrog Rest API

---

- Base URL
- The Platform REST URL is constructed of:
- <JFrog URL>/<Service Context>/
- For example:
  - # Using your JFrog URL
  - `http://myjfrog.acme.org/artifactory/`
  - # Using your Artifactory server hostname and port
  - `http://ARTIFACTORY_SERVER_HOSTNAME:8081/artifactory/`

## Jfrog Rest API Example - Deploying an Artifact

---

- To deploy the file 'myNewFile.txt', which is located in your Desktop directory, ('~/Desktop/myNewFile.txt')
- You have Artifactory running on your local system, on port 8081
- You wish to deploy the artifact into the 'my-repository' repository, under the 'my/new/artifact/directory/' folder structure, and wish to store the file there as 'file.txt'
- You have configured a user in Artifactory named 'myUser', with password 'myP455w0rd!', and this user has permissions to deploy artifacts
- Your API Key is 'ABcdEF'

## Jfrog Rest API Example - Deploying an Artifact

---

- To deploy the file using your username and password for authentication, you would use the following command:
- Using cURL with the REST API
- `curl -u admin:Vigneshbala@95! -X PUT "http://localhost:8081/artifactory/veb/jfrog.txt" -T ./jfrog.txt`

# Jfrog Rest API Example - Deploying an Artifact

```
Administrator: Command Prompt
G:\Local disk\JFrog>curl -u admin:Vigneshbala@95! -X PUT "http://localhost:8081/artifactory/veb/jfrog.txt" -T ./jfrog.txt
{
  "repo" : "veb",
  "path" : "/jfrog.txt",
  "created" : "2022-02-03T20:39:53.241Z",
  "createdBy" : "admin",
  "downloadUri" : "http://localhost:8081/artifactory/veb/jfrog.txt",
  "mimeType" : "text/plain",
  "size" : "543",
  "checksums" : {
    "sha1" : "3e21745207067abd397051db348f5bfd15b612ac",
    "md5" : "97aa9a29e97627d767b9d9cf2a49b747",
    "sha256" : "0cc689eeb6fec79b5d9bdcb0edb3177caadb425e2932407e8c3324a990524822"
  },
  "originalChecksums" : {
    "sha256" : "0cc689eeb6fec79b5d9bdcb0edb3177caadb425e2932407e8c3324a990524822"
  },
  "uri" : "http://localhost:8081/artifactory/veb/jfrog.txt"
}
G:\Local disk\JFrog>
```



02:10  
ENG IN  
04/02/2022 22

# Jfrog Rest API Example - Deploying an Artifact

The screenshot shows the JFrog Artifactory interface. On the left, the sidebar includes sections for Application, Artifactory, Packages, Builds, Artifacts, Distribution, Pipelines, Security & Compliance, and Getting Started. The Getting Started section is highlighted. The main content area displays a tree view of repositories under the 'Artifacts' tab. The 'jfrog-support-bundle' repository is selected, showing its details:

| jfrog-support-bundle |                                                         |
|----------------------|---------------------------------------------------------|
| General              |                                                         |
| Name:                | jfrog-support-bundle                                    |
| Package Type:        | Support                                                 |
| Repository Path:     | jfrog-support-bundle/                                   |
| URL to file:         | http://localhost:8082/artifactory/jfrog-support-bundle/ |
| Repository Layout:   | simple-default                                          |
| Created:             | 03-02-22 20:32:37 +00:00                                |

At the bottom, there is a trash can icon and a status bar with system icons.



# Jfrog Rest API Example - Deploying an Artifact

---

- Jfrog CLI
- ```
jf rt u jfrog.txt rps-generic-local --url  
http://localhost:8081/artifactory --user=admin --  
password=Vigneshbala@95!
```

# Jfrog Rest API Example - Deploying an Artifact

```
Administrator: Command Prompt
}
}
G:\Local disk\JFrog>jf rt u jfrog.txt rps-generic-local --url http://localhost:8081/artifactory --user=admin --password=Vigneshbala@95!
Log path: C:\Users\Balasubramaniam\.jfrog\logs\jfrog-cli.2022-02-04.02-23-57.11120.log
{
  "status": "success",
  "totals": {
    "success": 1,
    "failure": 0
  }
}
G:\Local disk\JFrog>
```



ENG  
IN 02:24  
04/02/2022 22

# Jfrog Rest API Example - Deploying an Artifact

The screenshot shows the JFrog Artifactory interface. On the left, a sidebar menu includes Application, Dashboard, Artifactory, Packages, Builds, Artifacts (which is selected), Distribution, Pipelines, Security & Compliance, and Getting Started. The main content area displays a tree view of repositories under 'Artifactory'. The 'rps-generic-local' repository is expanded, showing its contents: jfrog.txt and veb. To the right of the tree view is a detailed view of the 'rps-generic-local' repository, including tabs for General, Effective Permissions, Properties, and Followers. The General tab displays information such as Name (rps-generic-local), Package Type (Generic), Repository Path (rps-generic-local/), URL to file (http://localhost:8082/artifactory/rps-generic-local/), Repository Layout (simple-default), and Artifact Count / Size (Show). A search bar at the top right says 'Search Artifacts'.

## **Web-based Distributed Authoring and Versioning**

---

- The WebDAV protocol provides a framework for users to create, change and move documents on a server.
- <https://www.jfrog.com/confluence/display/JFROG/Browsing+Artifacts>

## High Availability

---

- You can configure JFrog services for High Availability with a cluster of two or more active/active nodes on the same Local Area Network (LAN).
- For both high availability and resiliency, an HA cluster with a minimum of three nodes provides you true HA.
- If one node goes down due to a system failure or is even taken down for hardware maintenance, the other two nodes still provide high availability and load balancing.
- A three-node cluster is critical to provide no-downtime upgrade as well.

# High Availability

---

- An HA configuration provides the following benefits:
- Optimal Resilience
  - Maximize your uptime. In case one or more nodes is unavailable or down for upgrade, the load is shared between the remaining nodes, ensuring optimal resilience and uptime.
- Improved Performance with Load Balancing
  - Scale your environment with as many nodes as you need. All cluster nodes in an HA configuration are synchronized, and jointly share and balance the workload between them.
  - When a node becomes unavailable, the cluster will automatically spread the workload across the other remaining node(s).

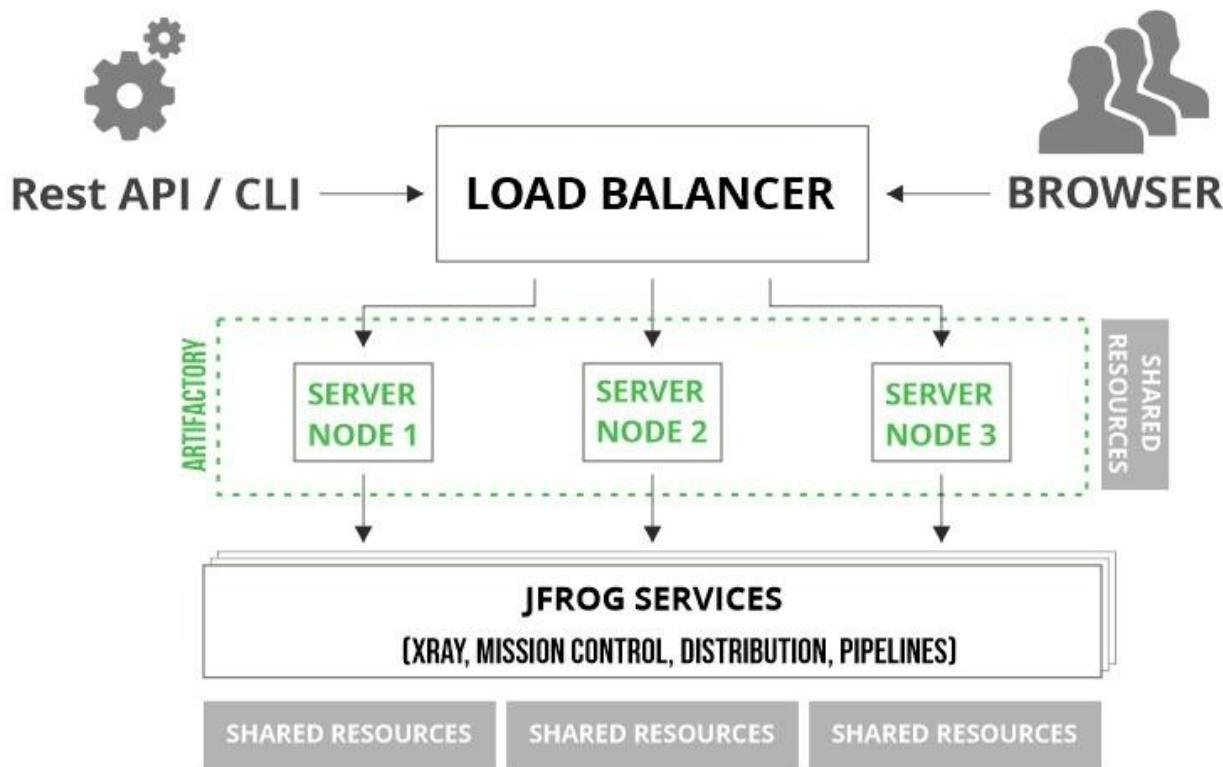
# High Availability

---

- An HA configuration provides the following benefits:
- Managed Heavy Loads
  - Accommodate larger load bursts with no compromise to performance. With horizontal server scalability, you can easily increase your capacity to meet any load requirements as your organization grows.
- Always being Synchronized
  - Seamlessly and instantly synchronize data, configuration, cached objects, and scheduled job changes across all cluster nodes.

# HA Architecture

- The HA architecture consists of 3 building blocks: load balancer, application(s) and common resources.



# Jfrog High Availability

---

- Step1: Install Chocolatey
- Install with cmd (Admin mode)
- @ "%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "[System.Net.ServicePointManager]::SecurityProtocol = 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"

## Jfrog High Availability

---

- Step1: Install Chocolatey
- Install with PS (Admin mode)
- Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol  
= [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object  
System.Net.WebClient).DownloadString('https://comm  
unity.chocolatey.org/install.ps1'))

# Jfrog High Availability

---

- Step1: Install Chocolatey
- Set Chocolatey env variable
- Installed in c:\programdata\chocolatey
- Note: Program Data hidden Item
- Restart cmd and type
- choco

## Install Helm (choco install Kubernetes-helm)

---

- <https://www.jfrog.com/confluence/display/JFROG/Installing+Artifactory#InstallingArtifactory-HelmInstallation>

# Install Helm (choco install Kubernetes-helm)

```
Administrator: Command Prompt

C:\Users\Administrator>choco
Chocolatey v0.12.1
Please run 'choco -?' or 'choco <command> -?' for help menu.

C:\Users\Administrator>choco install kubernetes-helm
Chocolatey v0.12.1
2 validations performed. 1 success(es), 1 warning(s), and 0 error(s).

Validation Warnings:
- A pending system reboot request has been detected, however, this is
  being ignored due to the current Chocolatey configuration. If you
  want to halt when this occurs, then either set the global feature
  using:
    choco feature enable -name=exitOnRebootDetected
  or pass the option --exit-when-reboot-detected.

Installing the following packages:
kubernetes-helm
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-helm 3.7.2... 100%

kubernetes-helm v3.7.2 [Approved]
kubernetes-helm package files install completed. Performing other installation steps.
The package kubernetes-helm wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N)o/[P]rint): y

Downloading kubernetes-helm 64 bit
  from 'https://get.helm.sh/helm-v3.7.2-windows-amd64.zip'
Progress: 100% - Completed download of C:\Users\Administrator\AppData\Local\Temp\chocolatey\kubernetes-helm\3.7.2\helm-v3.7.2-windows-amd64.zip (13.42 MB).
Download of helm-v3.7.2-windows-amd64.zip (13.42 MB) completed.
Hashes match.
Extracting C:\Users\Administrator\AppData\Local\Temp\chocolatey\kubernetes-helm\3.7.2\helm-v3.7.2-windows-amd64.zip to C:\ProgramData\chocolatey\lib\kubernetes-helm\tools...
C:\ProgramData\chocolatey\lib\kubernetes-helm\tools
ShimGen has successfully created a shim for helm.exe
The install of kubernetes-helm was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\kubernetes-helm\tools'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
```

Activate Windows  
Go to Settings to activate Windows.

# Install Helm (choco install Kubernetes-helm)

```
Administrator: Command Prompt

C:\Users\Administrator>choco
Chocolatey v0.12.1
Please run 'choco -?' or 'choco <command> -?' for help menu.

C:\Users\Administrator>helm
The Kubernetes package manager

Common actions for Helm:
- helm search:      search for charts
- helm pull:        download a chart to your local directory to view
- helm install:     upload the chart to Kubernetes
- helm list:        list releases of charts

Environment variables:
| Name           | Description
|-----|
| $HELM_CACHE_HOME | set an alternative location for storing cached files.
| $HELM_CONFIG_HOME | set an alternative location for storing Helm configuration.
| $HELM_DATA_HOME | set an alternative location for storing Helm data.
| $HELM_DEBUG | indicate whether or not Helm is running in Debug mode
| $HELM_DRIVER | set the backend storage driver. Values are: configmap, secret, memory, sql.
| $HELM_DRIVER_SQL_CONNECTION_STRING | set the connection string the SQL storage driver should use.
| $HELM_MAX_HISTORY | set the maximum number of helm release history.
| $HELM_NAMESPACE | set the namespace used for the helm operations.
| $HELM_NO_PLUGINS | disable plugins. Set HELM_NO_PLUGINS=1 to disable plugins.
| $HELM_PLUGINS | set the path to the plugins directory
| $HELM_REGISTRY_CONFIG | set the path to the registry config file.
| $HELM_REPOSITORY_CACHE | set the path to the repository cache directory
| $HELM_REPOSITORY_CONFIG | set the path to the repositories file.
| $KUBECONFIG | set an alternative Kubernetes configuration file (default "~/.kube/config")
| $HELM_KUBEAPISERVER | set the Kubernetes API Server Endpoint for authentication
| $HELM_KUBECFILE | set the Kubernetes certificate authority file.
| $HELM_KUBEASGROUPS | set the Groups to use for impersonation using a comma-separated list.
| $HELM_KUBEASUSER | set the Username to impersonate for the operation.
| $HELM_KUBECONTEXT | set the name of the kubeconfig context.
| $HELM_KUBETOKEN | set the Bearer KubeToken used for authentication.

Helm stores cache, configuration, and data based on the following configuration order:
- If a HELM_*_HOME environment variable is set, it will be used
- Otherwise, on systems supporting the XDG base directory specification, the XDG variables will be used
- When no other location is set a default location will be used based on the operating system
```

Activate Windows  
Go to Settings to activate Windows.

# Install Helm (choco install Kubernetes-helm)

```
Administrator: Command Prompt

C:\Users\Administrator>helm repo add jfrog https://charts.jfrog.io
"jfrog" already exists with the same configuration, skipping

C:\Users\Administrator>helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "jfrog" chart repository
Update Complete. Happy Helming!

C:\Users\Administrator>openssl rand -hex 32
WARNING: can't open config file: /usr/local/ssl/openssl.cnf
f6778f7904047d1220fb034e3ed59ef45bfc3832659d7c92ec0ca29186e1a645

C:\Users\Administrator>set MASTER_KEY=f6778f7904047d1220fb034e3ed59ef45bfc3832659d7c92ec0ca29186e1a645

C:\Users\Administrator>echo ${MASTER_KEY}
${MASTER_KEY}

C:\Users\Administrator>echo $MASTER_KEY
MASTER_KEY

C:\Users\Administrator>echo %MASTER_KEY%
f6778f7904047d1220fb034e3ed59ef45bfc3832659d7c92ec0ca29186e1a645

C:\Users\Administrator>kubectl create secret generic my-masterkey-secret -n artifactory --from-literal=master-key=%MASTER_KEY%
error: failed to create secret secrets "my-masterkey-secret" already exists

C:\Users\Administrator>kubectl delete secret my-masterkey-secret
Error from server (NotFound): secrets "my-masterkey-secret" not found

C:\Users\Administrator>kubectl delete secret -n artifactory my-masterkey-secret
secret "my-masterkey-secret" deleted

C:\Users\Administrator>kubectl create secret generic my-masterkey-secret -n artifactory --from-literal=master-key=%MASTER_KEY%
secret/my-masterkey-secret created

C:\Users\Administrator>openssl rand -hex 32
WARNING: can't open config file: /usr/local/ssl/openssl.cnf
823a909fc979b7f264aa8d727ba5e0801fac9a2453501ab4efb773a8444fa8c3

C:\Users\Administrator>set JOIN_KEY=823a909fc979b7f264aa8d727ba5e0801fac9a2453501ab4efb773a8444fa8c3
Activate Windows
Go to Settings to activate Windows.

C:\Users\Administrator>echo %JOIN_KEY%
823a909fc979b7f264aa8d727ba5e0801fac9a2453501ab4efb773a8444fa8c3

C:\Users\Administrator>kubectl create secret generic my-joinkey-secret -n artifactory --from-literal=join-key=%JOIN_KEY%
```

# Install Helm (choco install Kubernetes-helm)

```
Administrator: Command Prompt
```

```
C:\Users\Administrator>kubectl create secret generic my-joinkey-secret -n artifactory --from-literal=join-key=%JOIN_KEY%
secret/my-joinkey-secret created

C:\Users\Administrator>helm upgrade --install artifactory --set artifactory.masterKey=%MASTER_KEY% --set artifactory.joinKey=%JOIN_KEY% --namespace artifactory jfrog/artifactory
Release "artifactory" does not exist. Installing it now.
NAME: artifactory
LAST DEPLOYED: Sat Feb  5 15:48:34 2022
NAMESPACE: artifactory
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Congratulations. You have just deployed JFrog Artifactory!
```

1. Get the Artifactory URL by running these commands:

NOTE: It may take a few minutes for the LoadBalancer IP to be available.

```
You can watch the status of the service by running 'kubectl get svc --namespace artifactory -w artifactory-artifactory-nginx'
export SERVICE_IP=$(kubectl get svc --namespace artifactory artifactory-artifactory-nginx -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
echo http://$SERVICE_IP/
```

2. Open Artifactory in your browser  
Default credential for Artifactory:  
user: admin  
password: password

```
C:\Users\Administrator>kubectl get svc --namespace artifactory -w artifactory-artifactory-nginx
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP     PORT(S)          AGE
artifactory-artifactory-nginx   LoadBalancer   10.102.107.50  localhost       80:30082/TCP,443:32449/TCP   10s

C:\Users\Administrator>kubectl get svc --namespace artifactory
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP     PORT(S)          AGE
artifactory     ClusterIP   10.104.50.27 <none>         8082/TCP,8081/TCP   4m20s
artifactory-artifactory-nginx   LoadBalancer   10.102.107.50  localhost       80:30082/TCP,443:32449/TCP   4m20s
artifactory-postgresql   ClusterIP   10.106.118.183 <none>         5432/TCP          4m20s
artifactory-postgresql-headless ClusterIP  None            <none>         5432/TCP          4m20s

C:\Users\Administrator>kubectl get pods --namespace artifactory
NAME             READY   STATUS    RESTARTS   AGE
artifactory-0      1/1    Running   0          4m36s
artifactory-artifactory-nginx-bb69bf8db-dszhm 1/1    Running   0          4m37s
artifactory-postgresql-0   1/1    Running   0          4m36s

C:\Users\Administrator>
```

Activate Windows  
Go to Settings to activate Windows.

# Install Helm (choco install Kubernetes-helm)

```
Administrator: Command Prompt - kubectl port-forward svc/artifactory -n artifactory 7070:8082

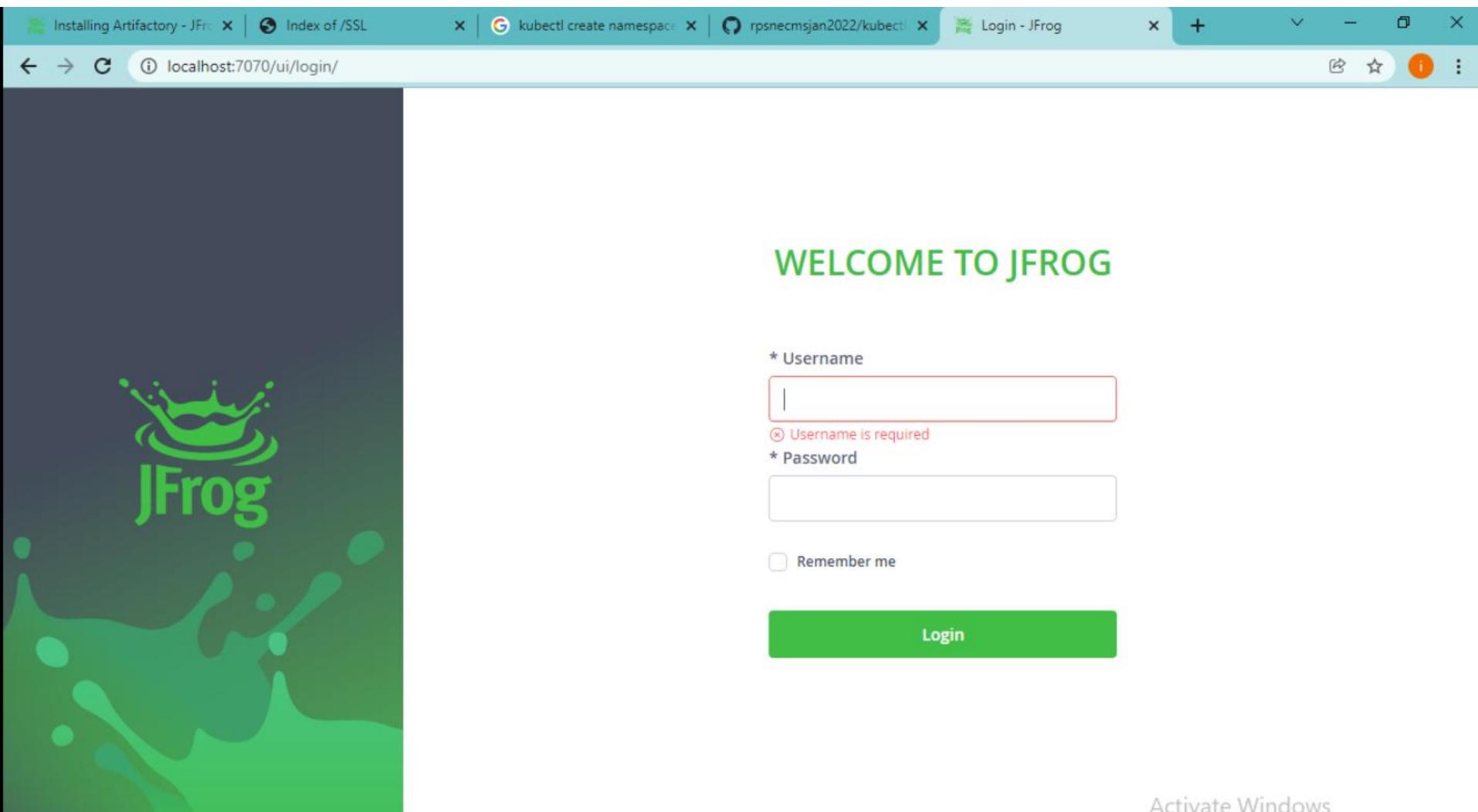
C:\Users\Administrator>kubectl get pods --namespace artifactory
NAME          READY   STATUS    RESTARTS   AGE
artifactory-0   1/1     Running   0          29m
artifactory-artifactory-nginx-bb69bf8db-dszhm  1/1     Running   0          29m
artifactory-postgresql-0  1/1     Running   0          29m

C:\Users\Administrator>kubectl get svc --namespace artifactory
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
artifactory     ClusterIP  10.104.50.27 <none>       8082/TCP,8081/TCP   29m
artifactory-artifactory-nginx  LoadBalancer  10.102.107.50  localhost   80:30082/TCP,443:32449/TCP   29m
artifactory-postgresql   ClusterIP  10.106.118.183 <none>       5432/TCP        29m
artifactory-postgresql-headless  ClusterIP  None         <none>       5432/TCP        29m

C:\Users\Administrator>kubectl port-forward svc/artifactory 7070:8082
Error from server (NotFound): services "artifactory" not found

C:\Users\Administrator>kubectl port-forward svc/artifactory -n artifactory 7070:8082
Forwarding from 127.0.0.1:7070 -> 8082
Forwarding from [::1]:7070 -> 8082
Handling connection for 7070
```

# Install Helm (choco install Kubernetes-helm)



Activate Windows

# Install Helm (choco install Kubernetes-helm)

The screenshot shows the JFrog Platform Administration interface under the Security tab. The main content area is titled "Security Configuration". It includes sections for "Password Encryption Policy" (set to "Supported"), "Multi-factor Authentication Methods" (with an unchecked checkbox for "Enable Google Authenticator"), "Connection details" (with a "Join Key" input field containing a redacted string), and "JFrog URL" (with a placeholder message "Please Configure a Custom Base URL"). A "Text copied" notification is visible at the top right. The left sidebar lists "Administration" and "Settings" sections like "Projects", "Repositories", "Identity and Access", and "LDAP". The bottom left shows a "Getting Started" section with a trial license notice. The bottom right features an "Activate Windows" message.

Installing Xray - JFrog - JFrog Do x Security Settings - JFrog x Gmail x | +

localhost:7070/ui/admin/configuration/security/general

Welcome, admin

JFrog Platform All ?

Administration

Projects

Repositories

Identity and Access

Security

Settings

LDAP

SAML SSO

Getting Started

Trial license 7.33.8 rev  
73308900 Licensed to VEB  
Consulting Pvt.Ltd  
© Copyright 2022 JFrog Ltd

Text copied

Security Configuration

Password Encryption Policy

Supported

Multi-factor Authentication Methods

Enable Google Authenticator

Connection details

Join Key

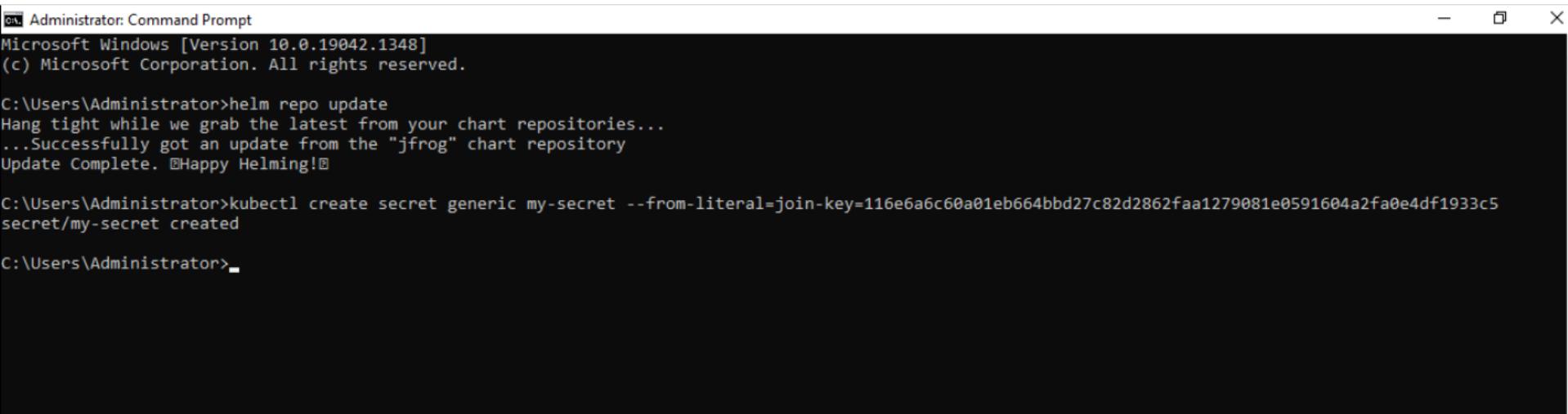
Please Configure a Custom Base URL

Reset Save

Activate Windows  
Go to Settings to activate Windows.  
Show all

openssl-1.0.2q-x64....zip

# Install Helm (choco install Kubernetes-helm)



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The title bar includes standard window controls (minimize, maximize, close) and the text "Administrator: Command Prompt". The main area of the window displays the following command-line session:

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>choco install Kubernetes-helm
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "jfrog" chart repository
Update Complete. Happy Helming!

C:\Users\Administrator>kubectl create secret generic my-secret --from-literal=join-key=116e6a6c60a01eb664bbd27c82d2862faa1279081e0591604a2fa0e4df1933c5
secret/my-secret created

C:\Users\Administrator>
```

# Jfrog JMX and Load Balancer

<https://artifacthub.io/packages/helm/jfrog/artifactory/7.16.2>

The screenshot shows the JFrog Platform Administration interface. The left sidebar has tabs for Administration, Monitoring, Storage, Service Status, and System Logs. The main content area is titled "Storage" and shows the following information:

The data was last refreshed on: 06-02-22 21:26:22 +0530

**Binaries**

Binaries Size:	0 bytes	Binaries Count:	0
Artifacts Size:	0 bytes	Artifacts Count:	0
Optimization:	N/A	Items Count:	0

**Storage**

**File System**

Directory: /opt/jfrog/artifactory/var/data/artifactory/filestore

Used: 5.1 GB / 251.0 GB (2.0%)

**federated-repo**

Used: 5.1 GB / 251.0 GB (2.0%)

The top navigation bar shows several tabs: FW: Congratulations on your JFro..., AMI-AT-Trainer, artifactory 7.16.2 · helm/jfrog, and JFrog. The address bar shows the URL: localhost:7070/ui/admin/monitoring/storage-summary. The bottom taskbar includes icons for various Windows applications like File Explorer, Edge, and Power BI.

# Jfrog JMX and Load Balancer

---

<https://artifacthub.io/packages/helm/jfrog/artifactory/7.16.2>

<https://hub.kubeapps.com/charts/jfrog/artifactory-ha/0.13.2>

# Jfrog JMX and Load Balancer



# Storage Optimization

---

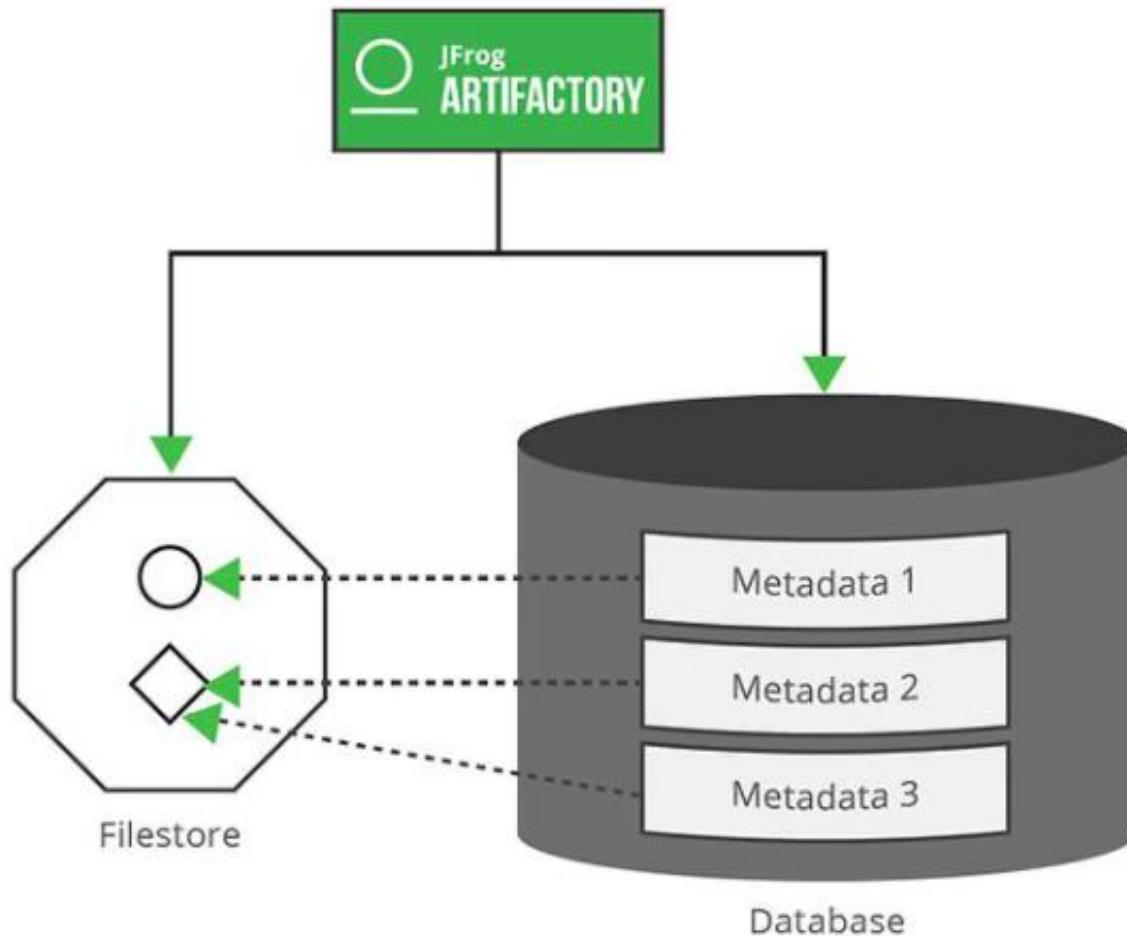
- MANAGING YOUR DATA WITH CHECKSUM-BASED STORAGE
- Artifactory's optimal design utilizes checksum-based storage, which uses two components: filestore and database.
- binary is located in the filestore (local disk by default) and identified by its checksum (sha1), with all the metadata (such as package metadata, artifact names, sizes, creation dates, repo locations, sha256 values, and signatures) saved in the database.

# Storage Optimization

---

- A checksum approach enables unique identification of artifacts, without duplication in the filestore.
- Remote teams can consume the same dependency, that is stored in the filestore, from two different repositories with different metadata referenced from the database.
- This prevents duplication and consumes less storage.

# Storage Optimization



their metadata provides flexibility on how the data is managed.

# Export Repository

S How to migrate artifactory derby x intelartifactoryjan2022/mysql.txt x Install Artifactory | JFrog x JFrog x Congratulations on your JFrog 30 x +

localhost:8082/ui/admin/artifactory/import\_export/repositories

Insert title here Empire New Tab How to use Asserti... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 » Reading list

JFrog Platform All Search admin resources ? Welcome, admin

Administration

- Security
- Licenses
- General
- Proxies
- Monitoring
- Support Zone

SERVICES

- Artifactory

Getting Started

Trial license 7.33.9 rev 7330900 Licensed to VEB Consulting Pvt.Ltd © Copyright 2022 JFrog Ltd

### Repositories Import & Export

#### Export Repository to Path

Target Local Repository: example-repo-local

\* Export Path on Server:  [Browse](#)

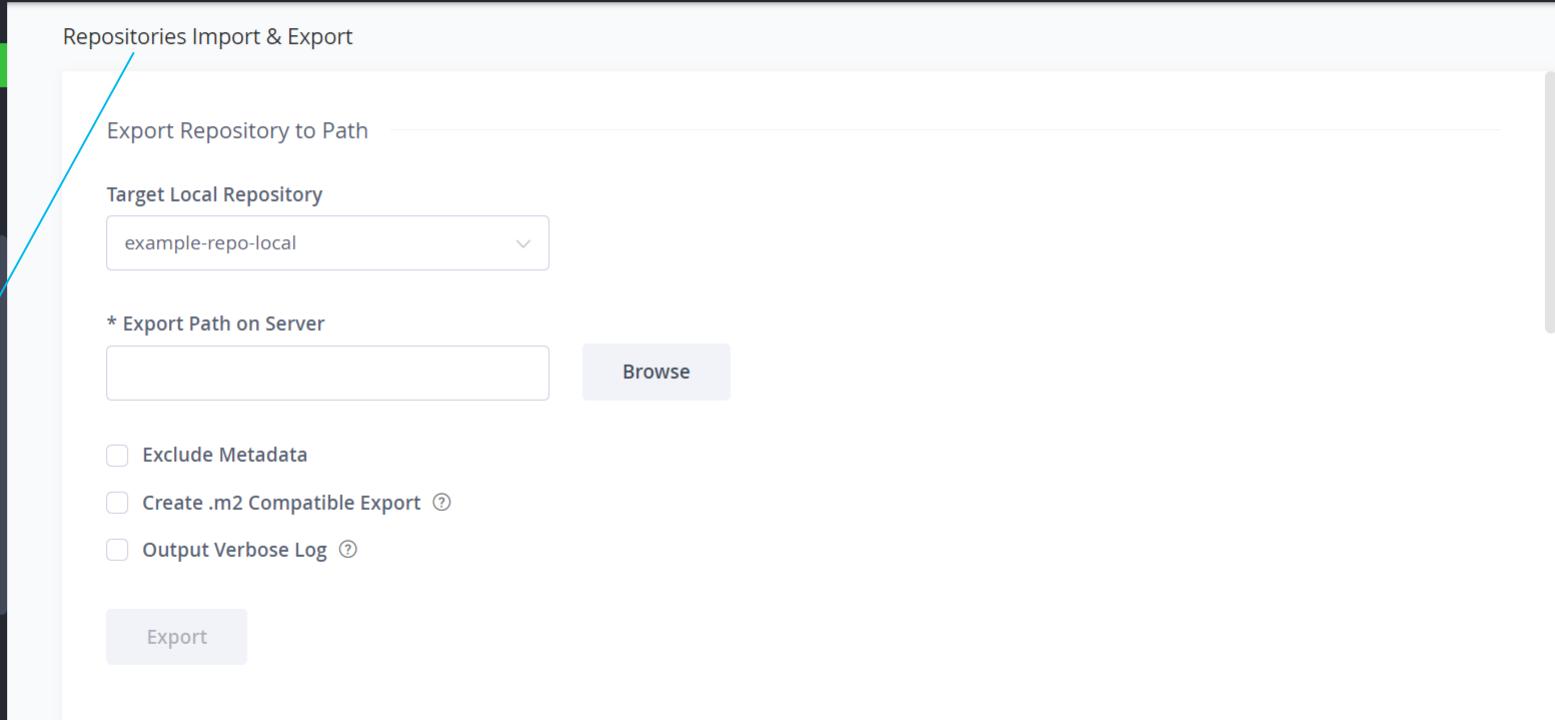
Exclude Metadata

Create .m2 Compatible Export [?](#)

Output Verbose Log [?](#)

[Export](#)

#### Import Repository from Path



# Export Repository

localhost:8082/ui/admin/artifactory/import\_export/repositories

JFrog Platform All Search admin resources Welcome, admin

Administration

- Security
- Licenses
- General
- Proxies
- Monitoring
- Support Zone

Artifactory

Getting Started

Trial license 7.33.9 rev 7330900 Licensed to VEB Consulting Pvt.Ltd © Copyright 2022 JFrog Ltd

Repositories Import Export Repository Target Local Repository \* Export Path or example-repository /boot /dev /docker /etc /home /lib

Mount Point: / Path to export /docker

SERVER FILE SYSTEM BROWSER

boot dev docker etc home lib

Export Cancel Select

Import Repository from Path

00:32 ENG IN 11/02/2022 23

# Access Volume

```
Command Prompt - docker run -it -v artifactory-data:/var/opt/jfrog/artifactory --name testvolume busybox
docker.jfrog.io/jfrog/artifactory-pro:latest
2b38de49ce4b3d8f5304a9d2d58f270fba30d8815e687040ffff7fb6944ab5f0e

G:\Local disk\JFrog>docker run -it -v artifactory-data:/var/opt/jfrog/artifactory --name testvolume busybox
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
009932687766: Pull complete
Digest: sha256:afcc7f1ac1b49db317a7196c902e61c6c3c4607d63599ee1a82d702d249a0ccb
Status: Downloaded newer image for busybox:latest
/ # ls
bin dev etc home proc root sys tmp usr var
/ # cd var
/var # ls
opt spool www
/var # cd opt
/var/opt # ls
jfrog
/var/opt # cd jfrog
/var/opt/jfrog # ls
artifactory
/var/opt/jfrog # cd artifactory
/var/opt/jfrog/artifactory # ls
backup bootstrap data etc log work
/var/opt/jfrog/artifactory #
```



^ ENG  
IN 00:46  
11/02/2022 23

# Access Volume

```
eswaribala@DESKTOP-55AGI0I:/opt$ docker run -it -v artifactory-data:/var/opt/jfrog/artifactory --name testvolume busybox
/ # ls
bin dev etc home proc root sys tmp usr var
/ # cd etc
/etc # ls
group hostname hosts localtime mtab network passwd resolv.conf shadow
/etc # cd ..
/ # cd var
/var # ls
opt spool www
/var # cd opt
/var/opt # ls
jfrog
/var/opt # cd jfrog
/var/opt/jfrog # ls
artifactory
/var/opt/jfrog # cd artifactory
/var/opt/jfrog/artifactory # ls
backup bootstrap data etc log work
/var/opt/jfrog/artifactory # cd etc
/var/opt/jfrog/artifactory/etc # ls
README.md filebeat.yaml observability system.basic-template.yaml
access integration replicator system.full-template.yaml
artifactory logrotate router system.yaml
event metadata security
/var/opt/jfrog/artifactory/etc #
```

# Access Volume

```
Administrator: Command Prompt - docker exec -it mysql8 mysql -u root -p
C:\WINDOWS\system32>docker exec -it mysql8 mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.28 MySQL Community Server - GPL

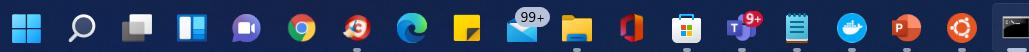
Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| artifactory
| information_schema
| mysql
| performance_schema
| sys
+-----+
5 rows in set (0.07 sec)

mysql> ■
```



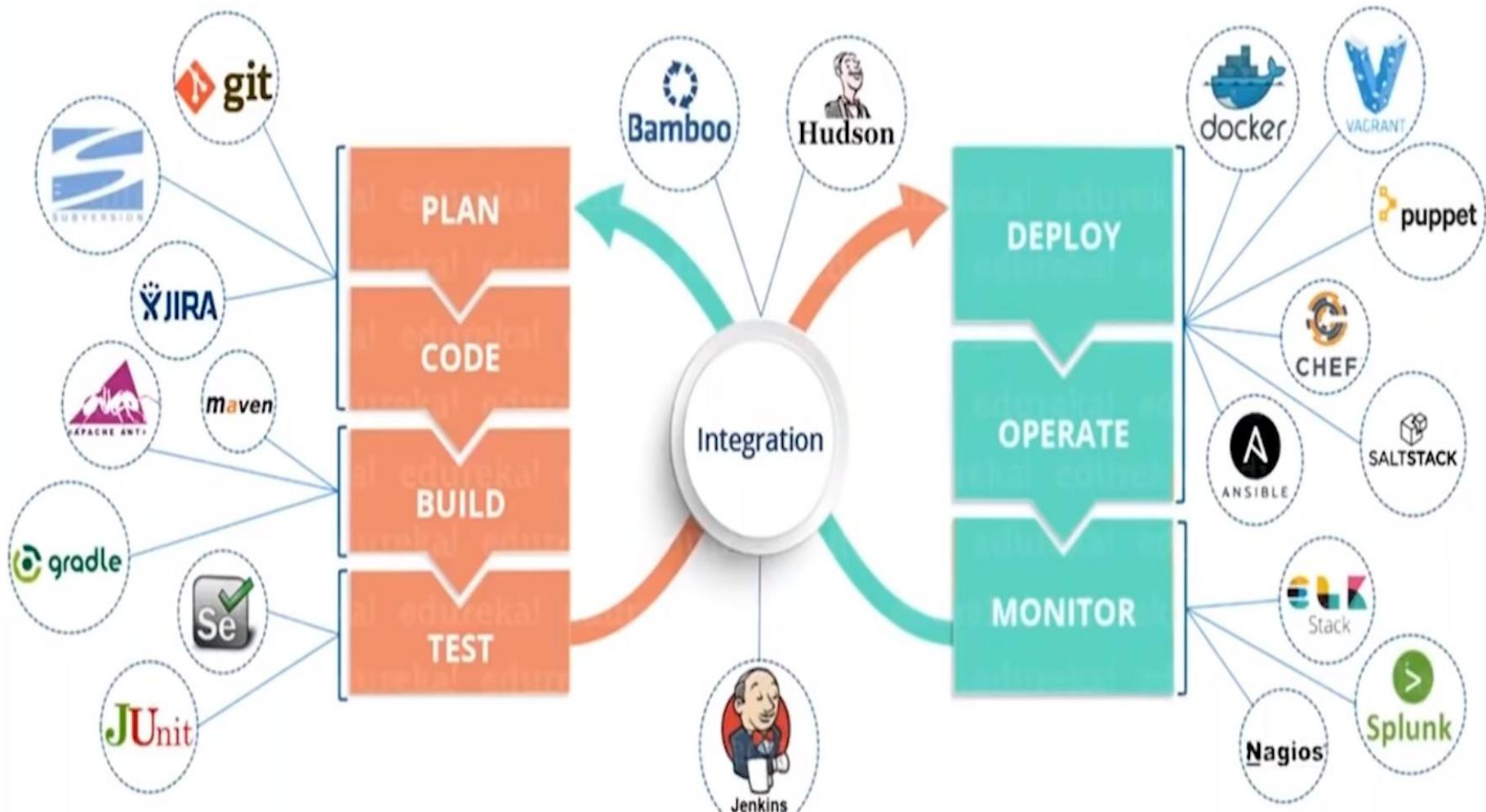
01:41  
ENG  
IN 11/02/2022 23

# Storage Optimization

---

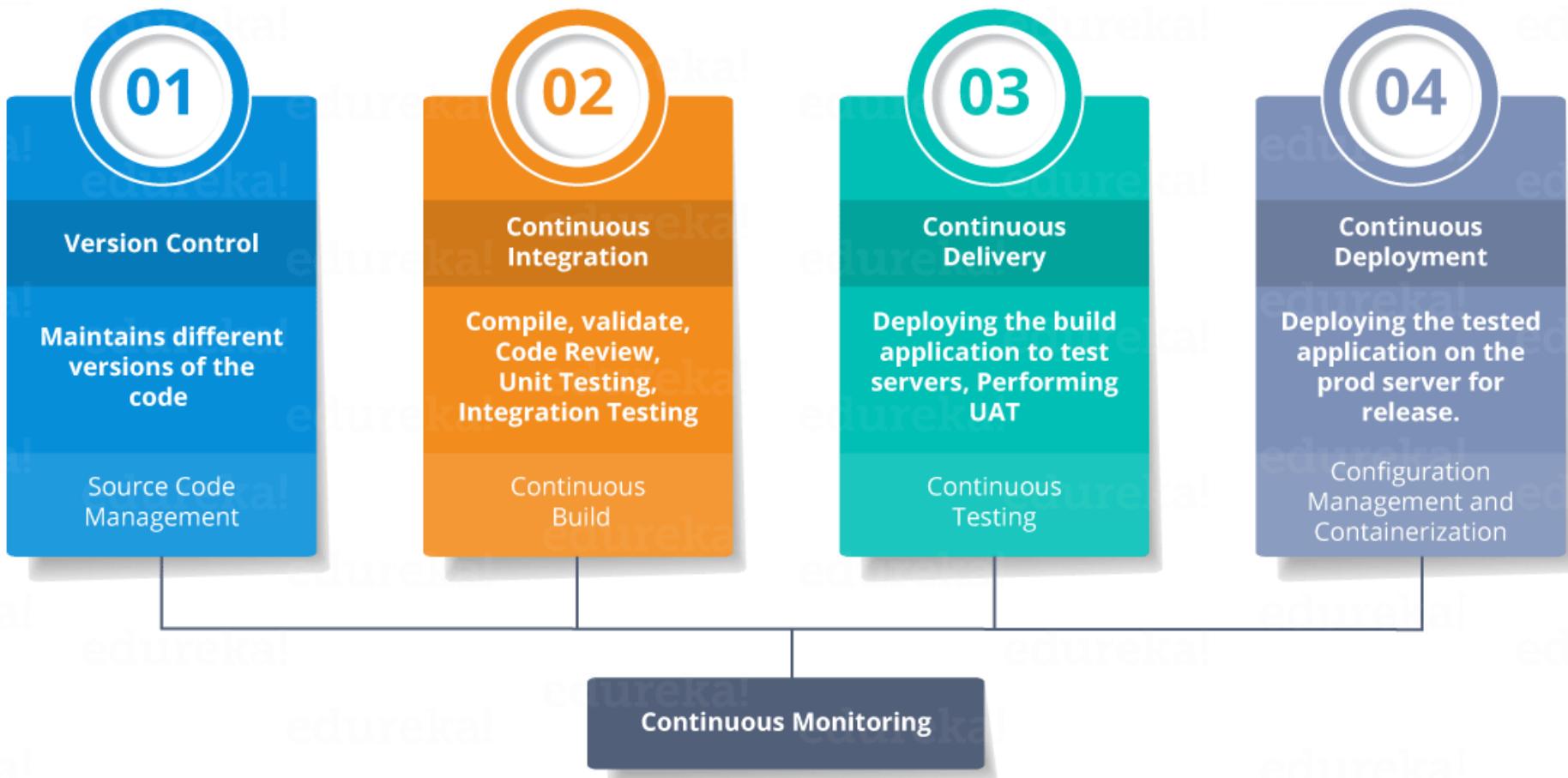
- Separating the binaries from their metadata provides flexibility on how the data is managed.
- Moving or copying artifacts to a different path (for example, during promotion) will only affect the database and the SQL query, and not the filestore.
- This is because this will only change the artifact metadata, which is stored separately in the database.
- The database server may be sized according to the following ratio: 1/100 of the filestore. This is not a strict formula t

# What is Devops

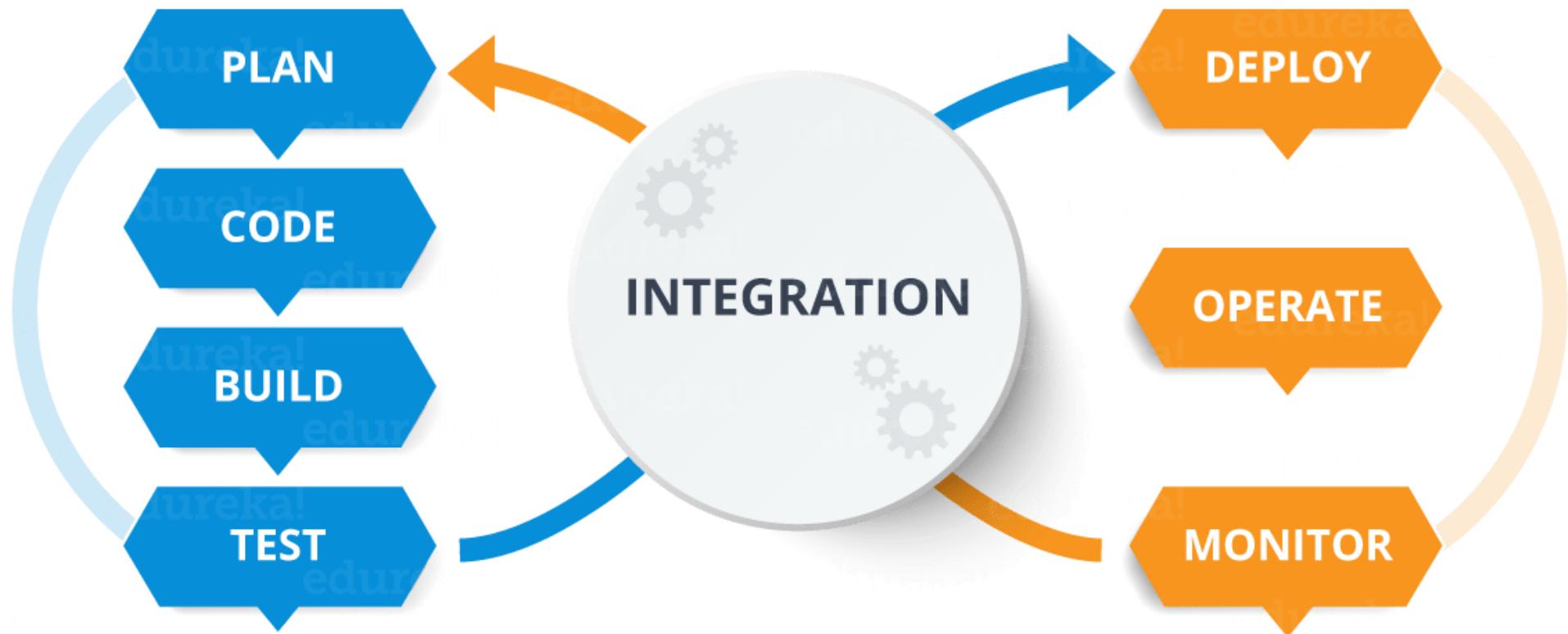


Category	Chef	Puppet	Ansible
Availability (in case of failure)	Backup Server	Alternative Master	Secondary instance
Configuration Language	Ruby DSL	Ruby, Puppet DSL, Embedded Ruby (ERB), DSL	Python, YAML
Architecture	Master-Agent	Master-Agent	Only Master (Agentless)
Installation Process	Time-intensive and complex due to Chef Workstation	Time-intensive due to master-agent certificate signing	Easy
Configuration Management	Pull	Pull	Push and Pull
Scalability	High	High	Very High
Interoperability	Chef Server works only on Linux/Unix; Chef Client and Workstation can work on Windows as well	Puppet Master works only on Linux/Unix; Puppet Agent or Client works on Windows	Ansible Server works on Linux/Unix; Client machines on Windows
Capabilities	<ul style="list-style-type: none"> <li>■ Continuous delivery with automated workflow</li> <li>■ Compliance and security management</li> <li>■ Infrastructure automation</li> </ul>	<ul style="list-style-type: none"> <li>■ Orchestration</li> <li>■ Automated provisioning</li> <li>■ Code and node management</li> <li>■ Configuration automation</li> <li>■ Visualization and reporting</li> <li>■ High transparency</li> <li>■ Role-based access control</li> </ul>	<ul style="list-style-type: none"> <li>■ Simple orchestration</li> <li>■ Streamlined provisioning</li> <li>■ Continuous delivery with automated workflow</li> <li>■ App deployment</li> <li>■ Security and compliance integration into automation</li> </ul>
Pricing	<ul style="list-style-type: none"> <li>■ Standard Hosted Chef – USD 72/year/node</li> <li>■ Chef Automation – USD 137/year/node</li> </ul>	<ul style="list-style-type: none"> <li>■ Puppet Enterprise – USD 120/year/node</li> <li>■ Premium – USD 199/year/node</li> </ul>	<ul style="list-style-type: none"> <li>■ Self-support Package – USD 5000/year</li> <li>■ Premium – USD 14,000/year/100 nodes</li> </ul>

# CI/CD Pipeline



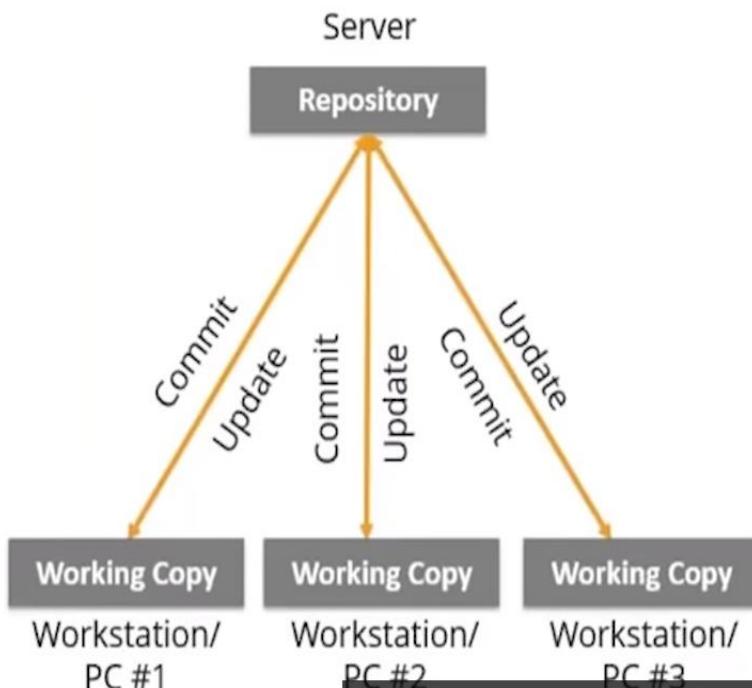
# CI/CD Pipeline



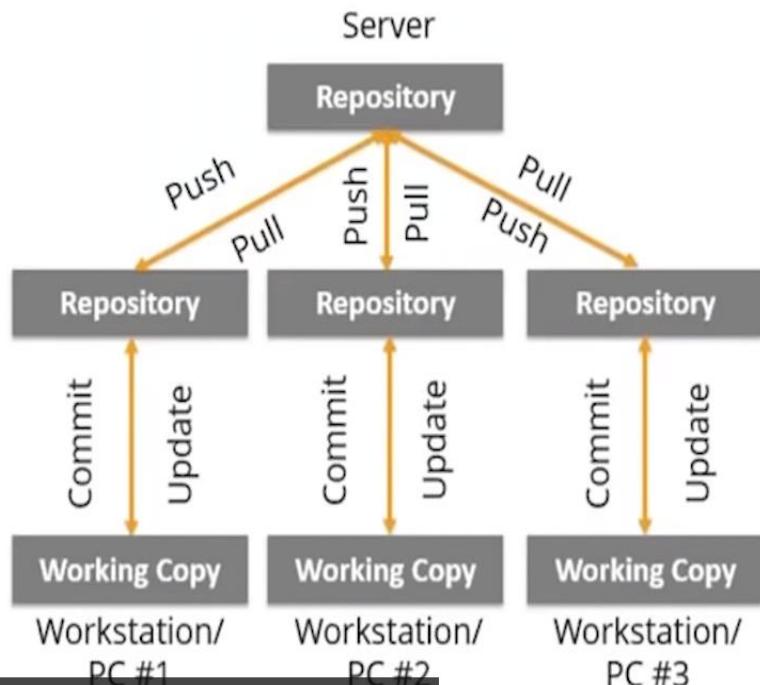
# Source Code Management

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

## Centralized Version Control System



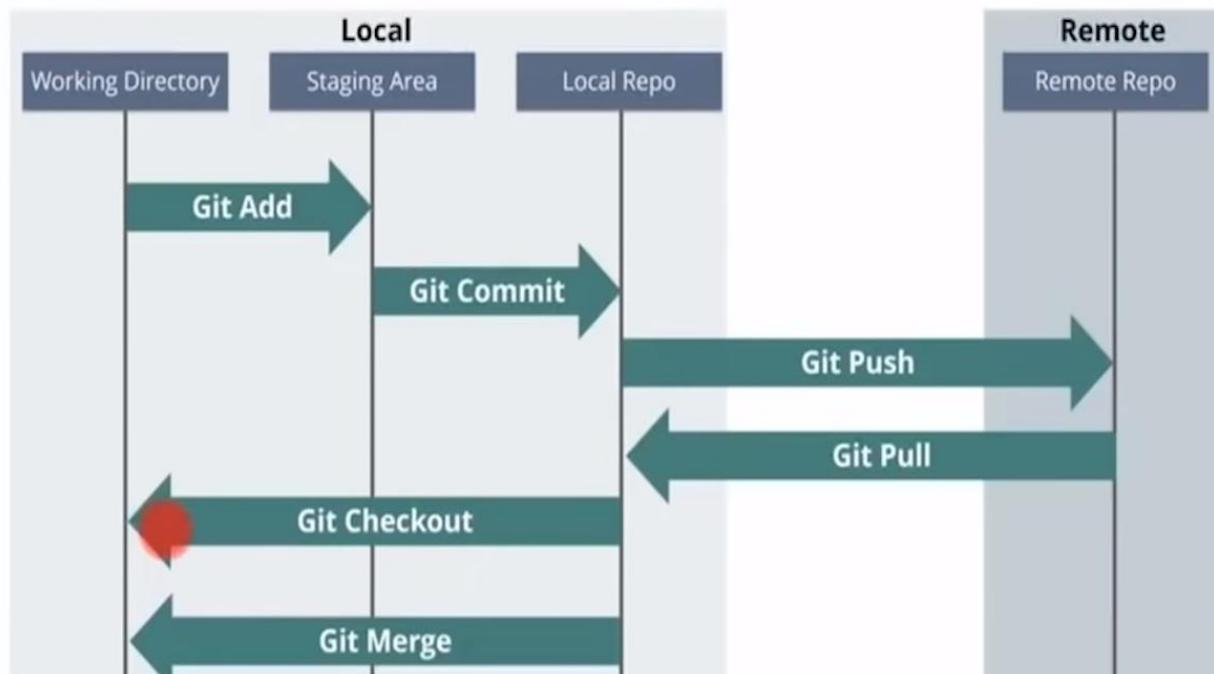
## Distributed Version Control System



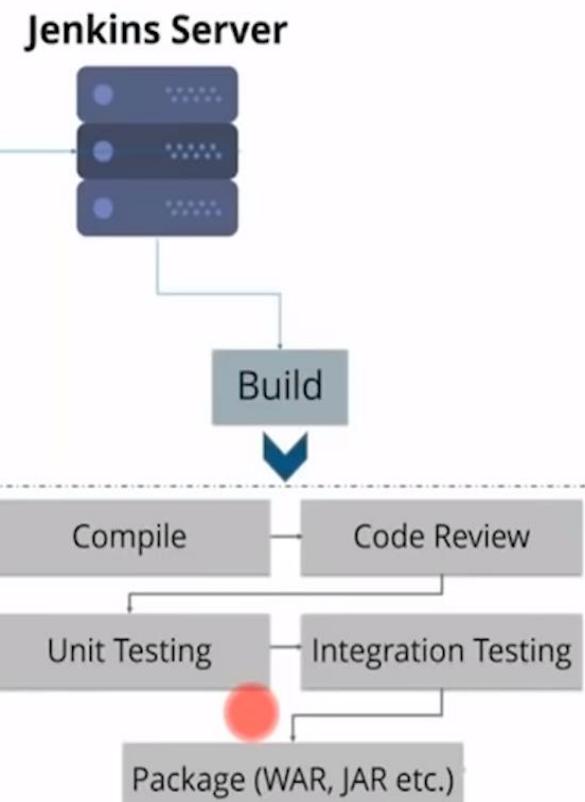
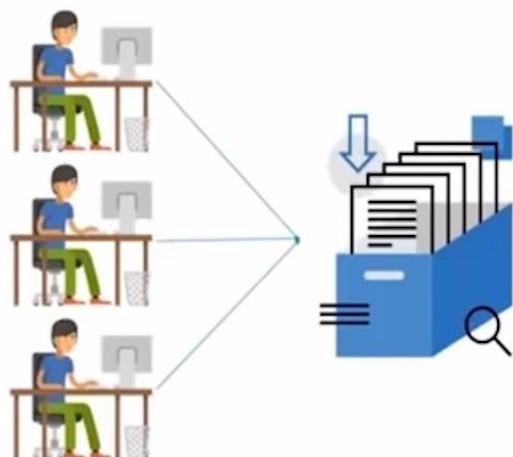
# Source Code Management



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

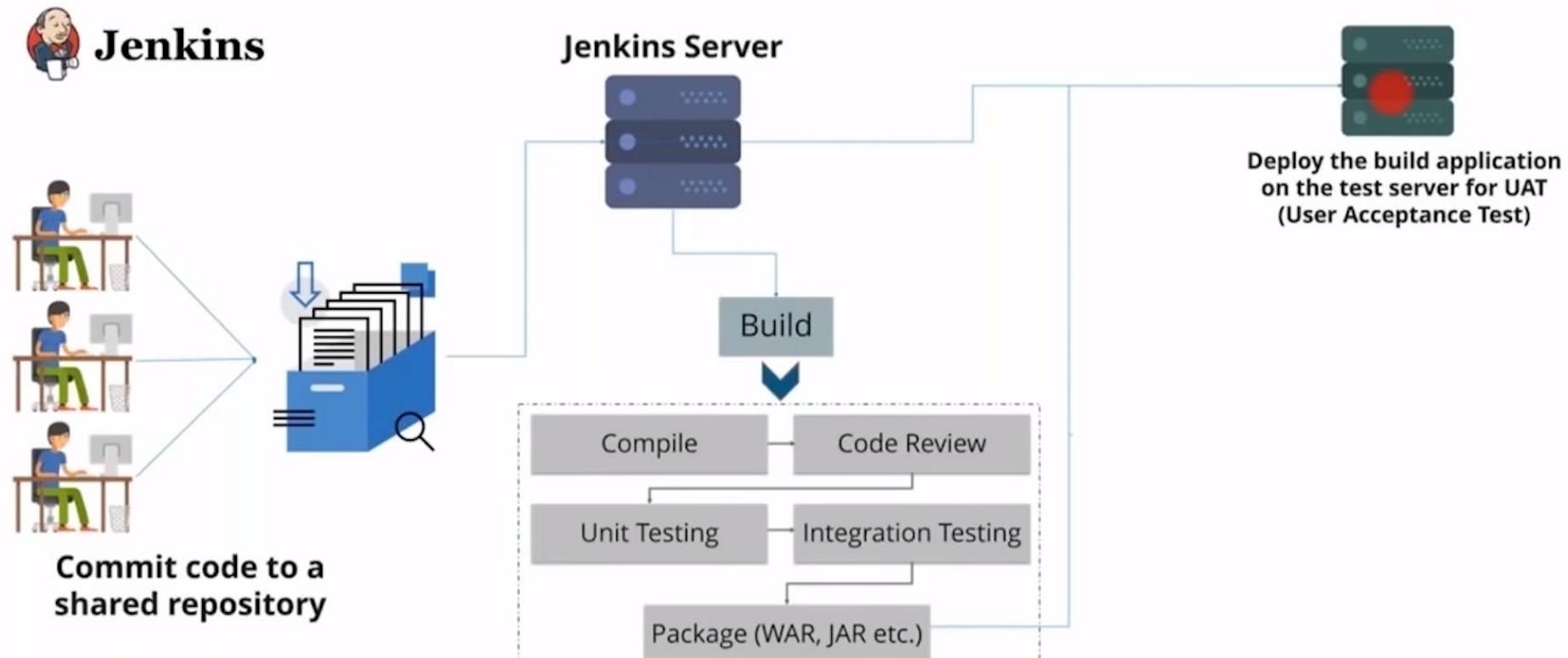


# Continuous Integration

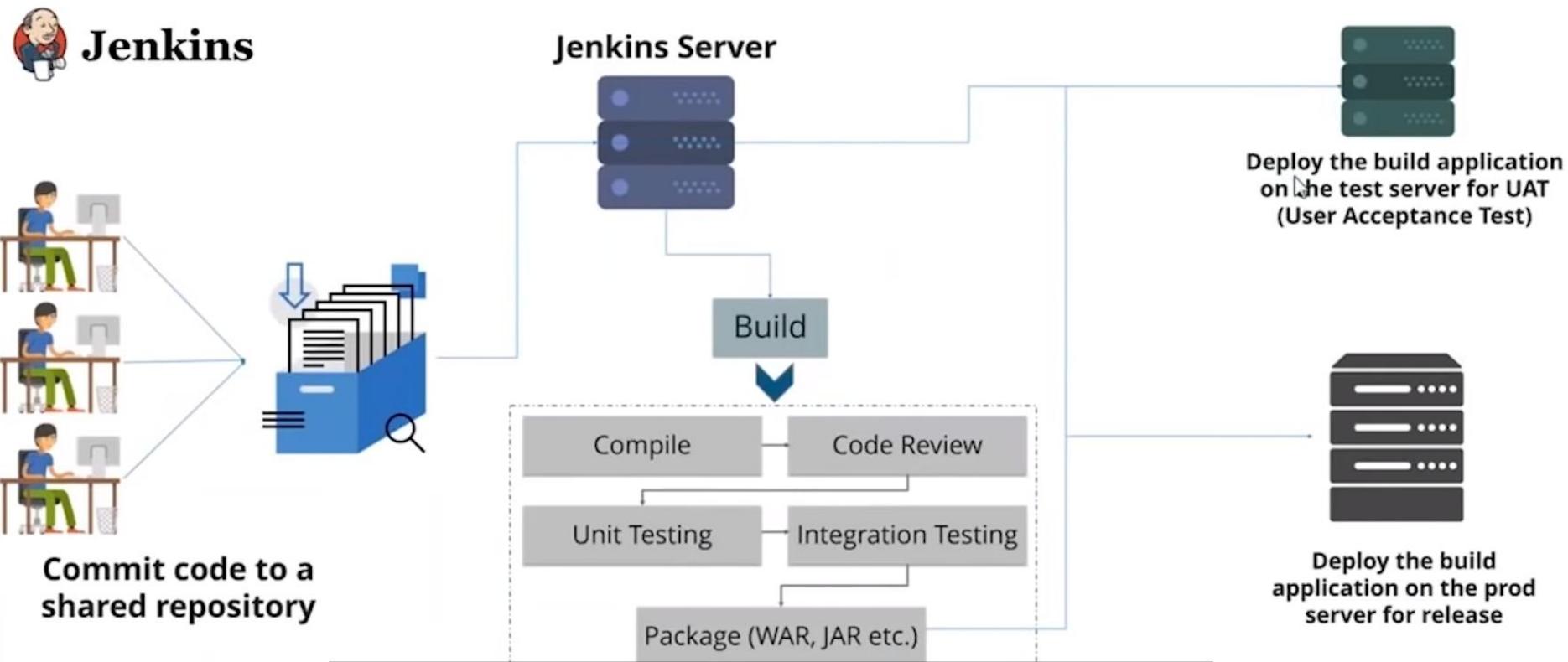


Commit code to a  
shared repository

# Continuous Delivery



# Continuous Deployment

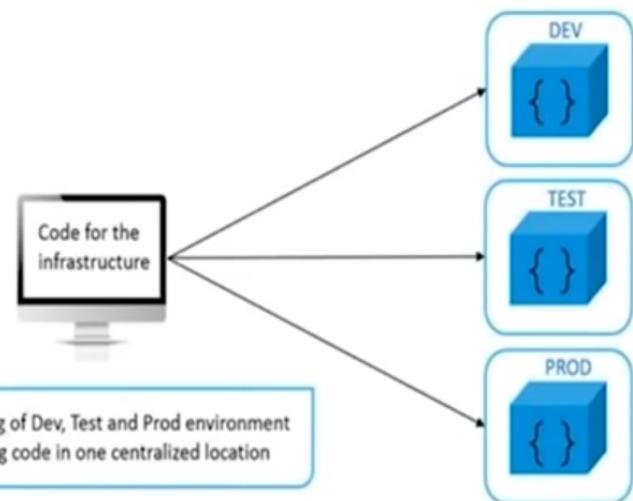


# Configuration Management

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

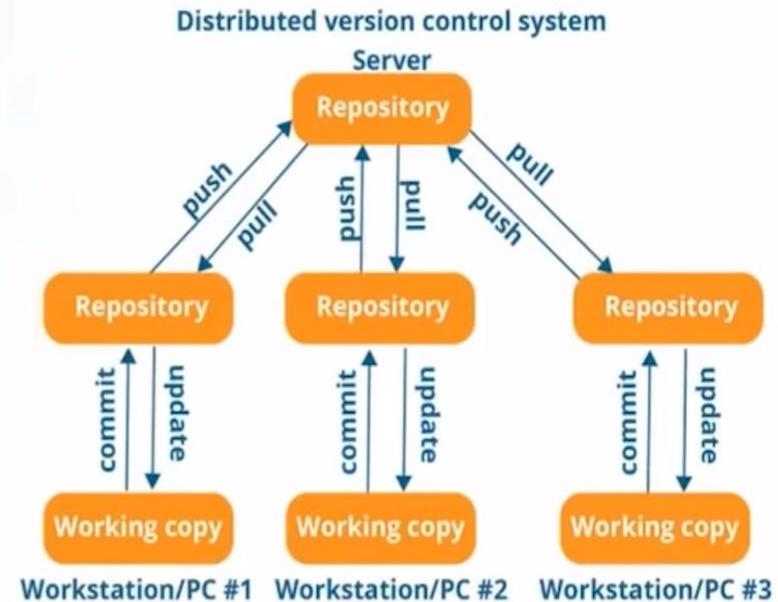
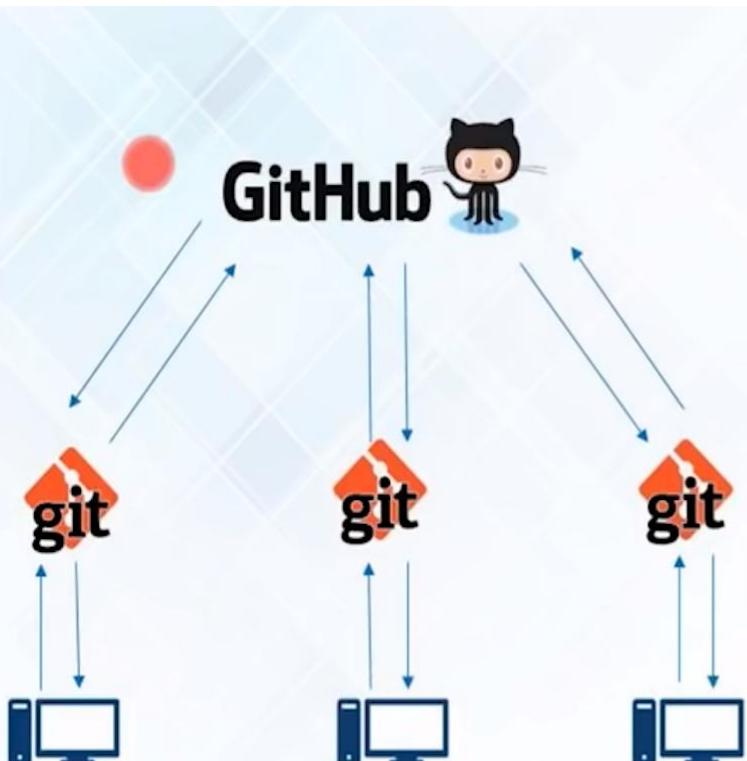
- Configuration Management (CM) ensures that the current design and build state of the system is known, good & trusted

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

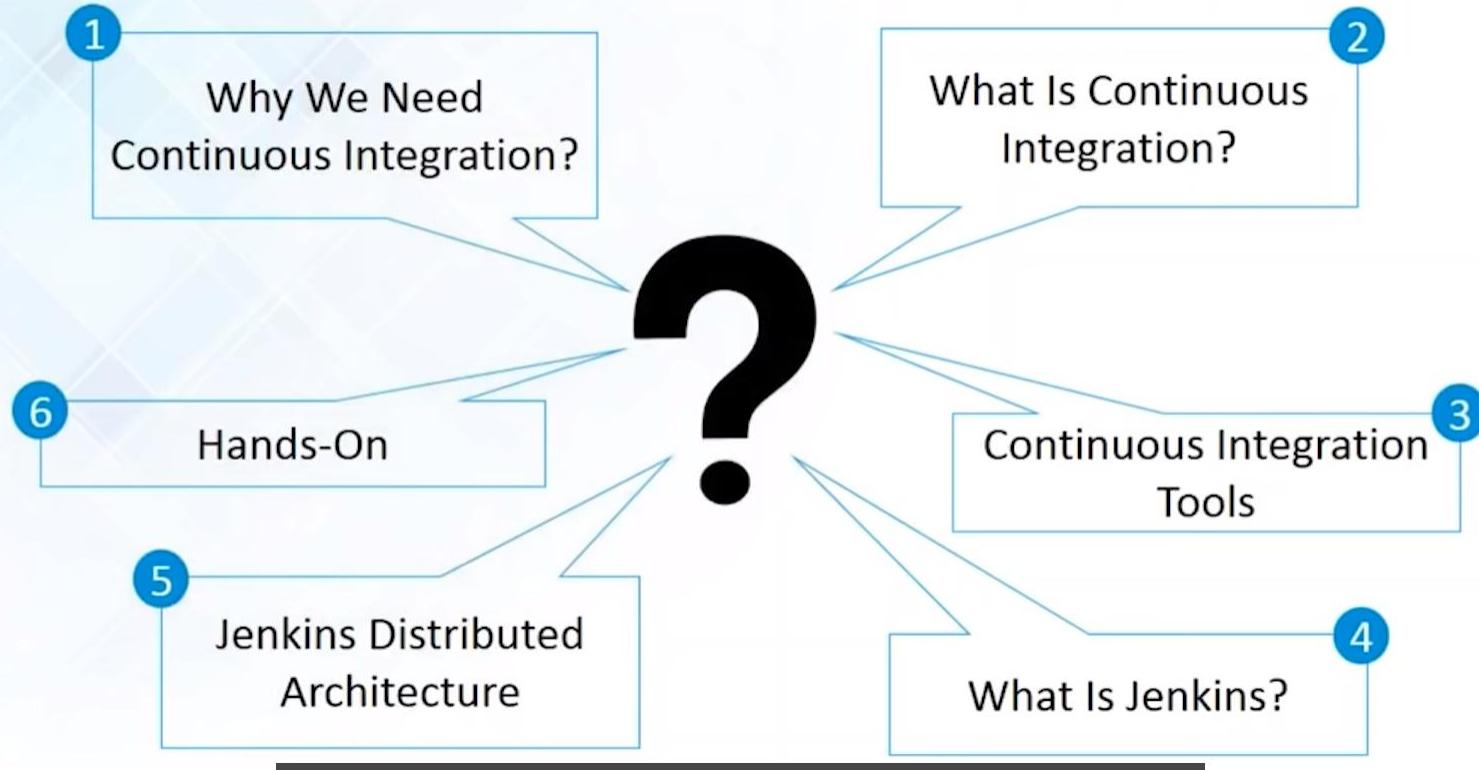


Provisioning of Dev, Test and Prod environment  
by writing code in one centralized location

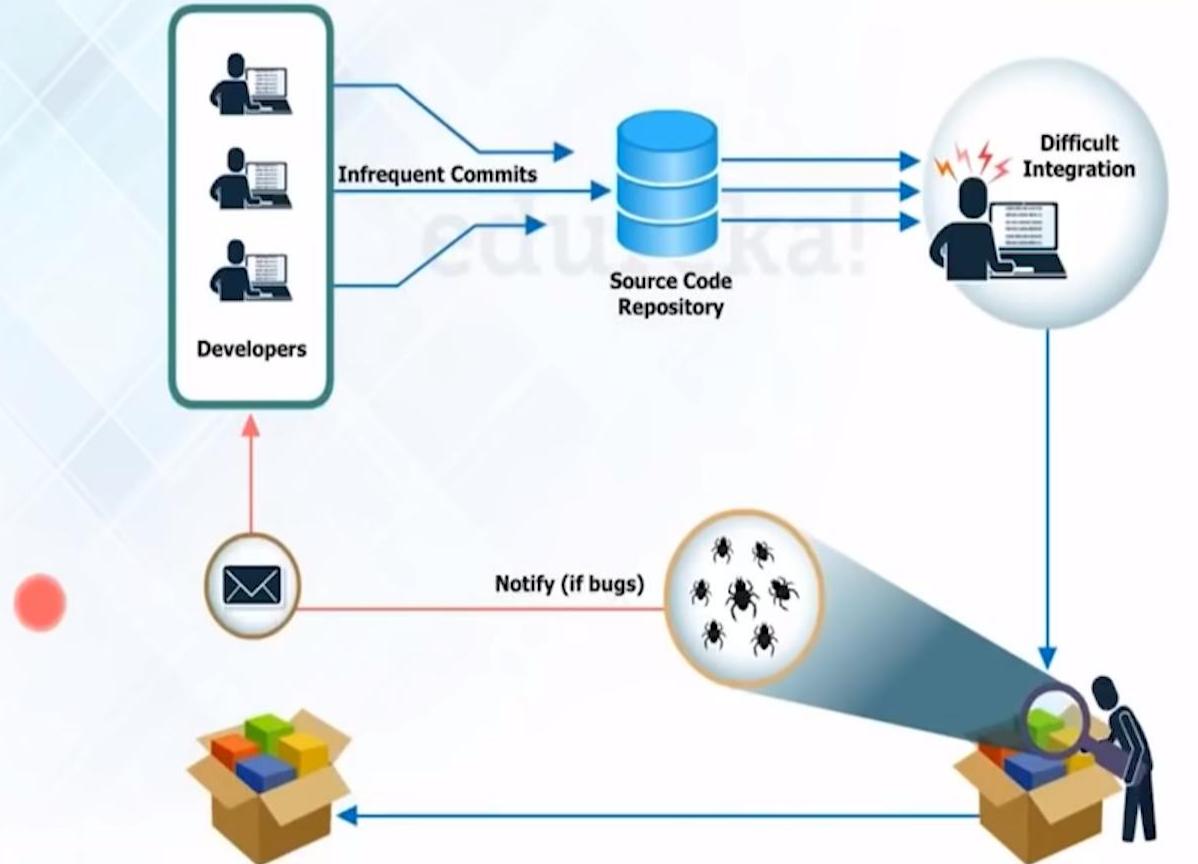
# Git vs Github



# 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.

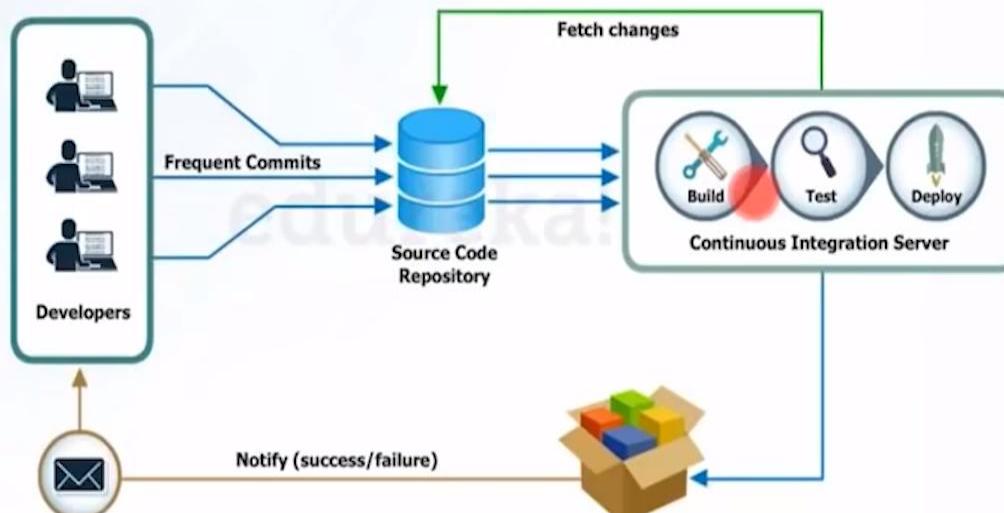


If the test fails then locating and fixing bugs is very difficult. Developers have to check the entire source code of the software.

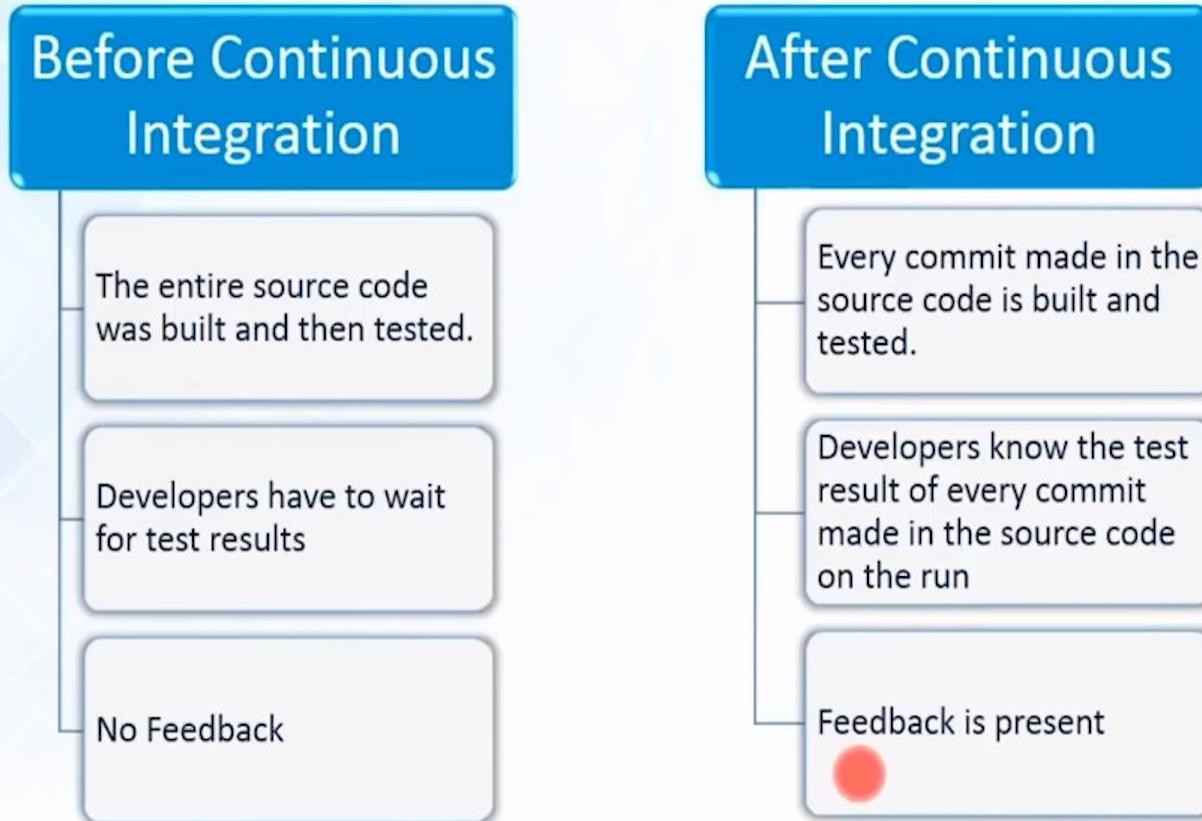


# Continuous Integration To the Rescue

- ❑ 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

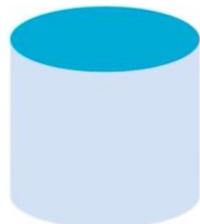
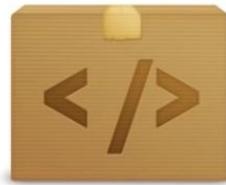


# Why do we need Continuous Delivery?

Developers



Application directly deployed to prod



Code repo



Manual testing  
(UT, IT)

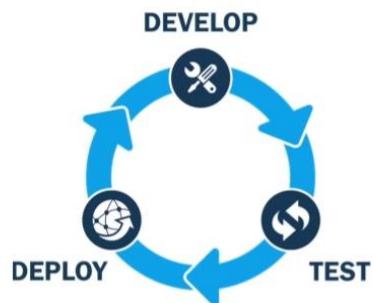
## POSSIBLE REASONS

- Different environments (servers)
- Different libraries & packages (dependencies)
- End user load (traffic)
- App not accessible to intended audience

# Why do we need Continuous Delivery?

---

Automates software release



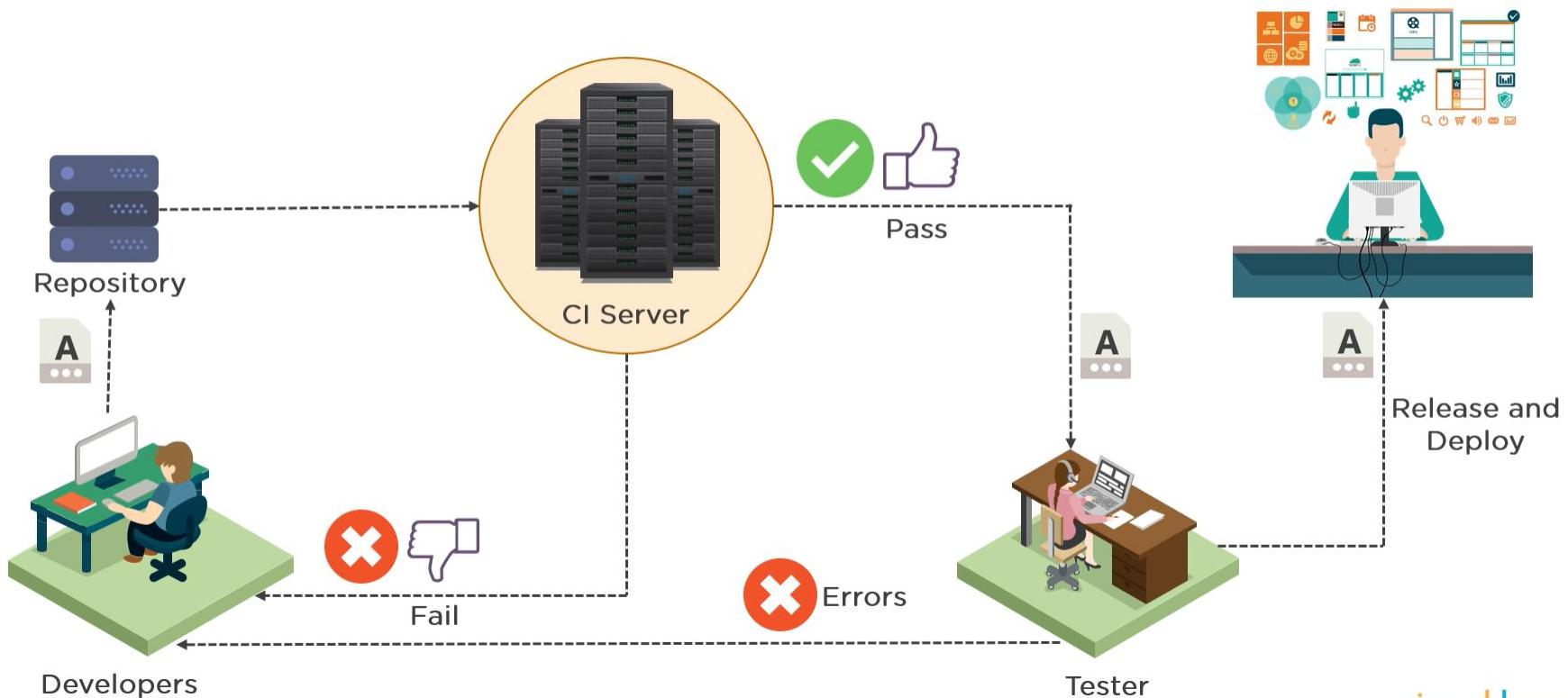
Increases developer productivity



Locates and addresses bugs quicker

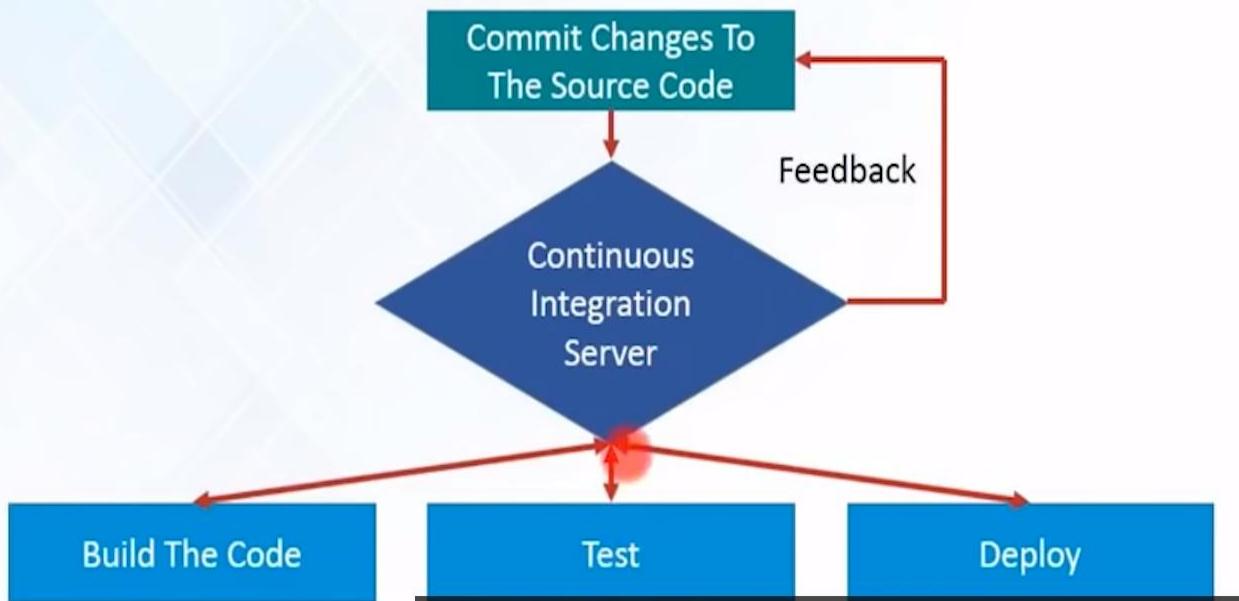


# What is Continuous Integration



# 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.

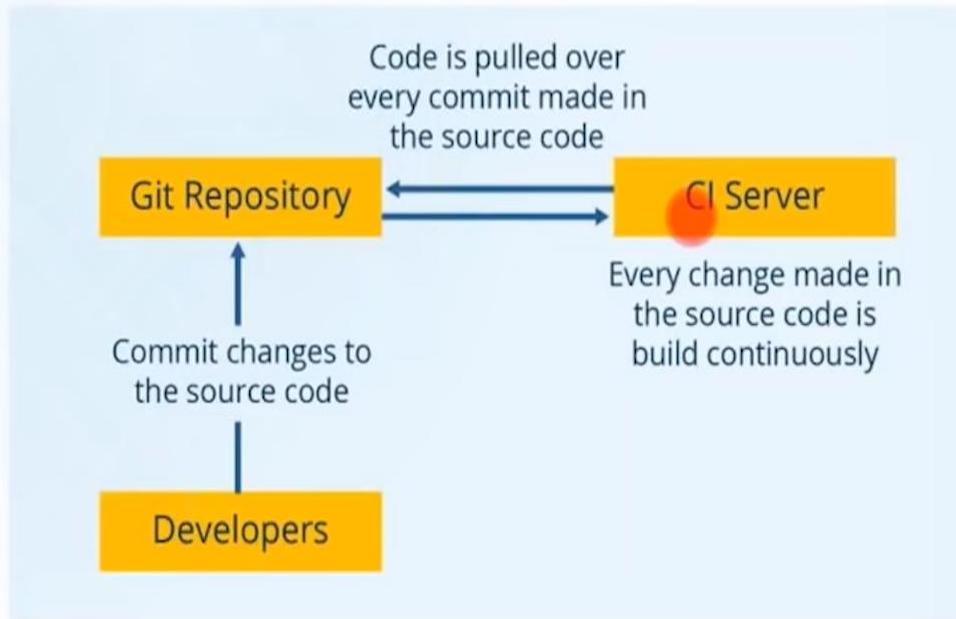


# What is Continuous Integration Case Study : Nokia

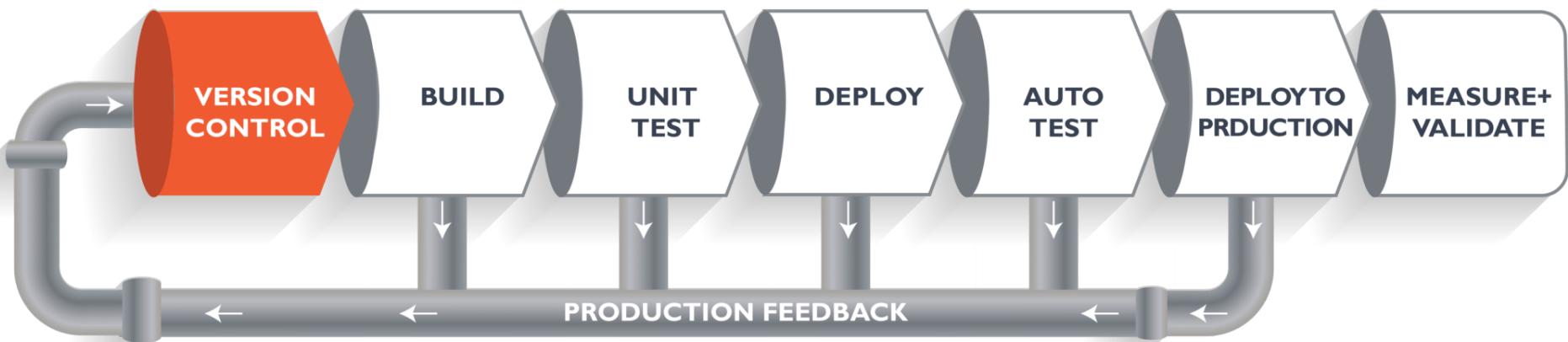
Solution:



## Continuous Integration

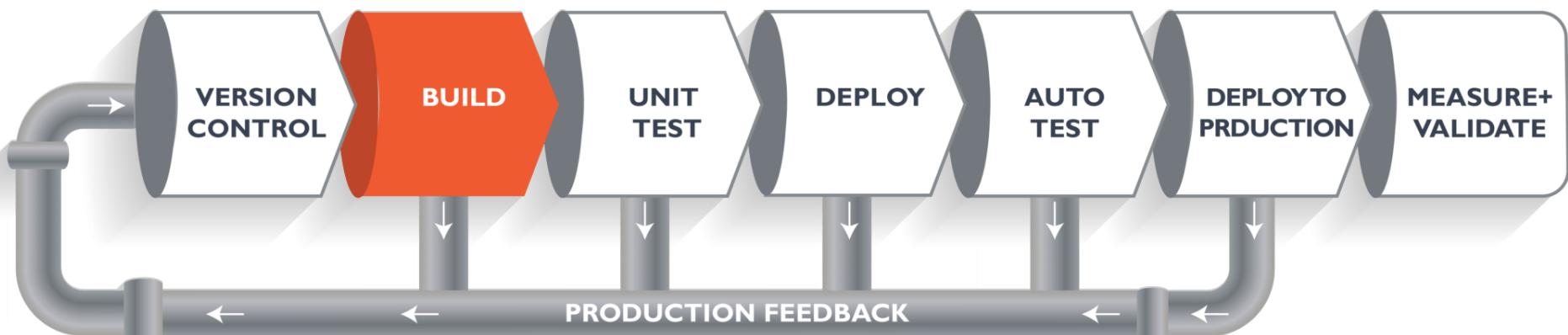


# CI/CD Pipeline



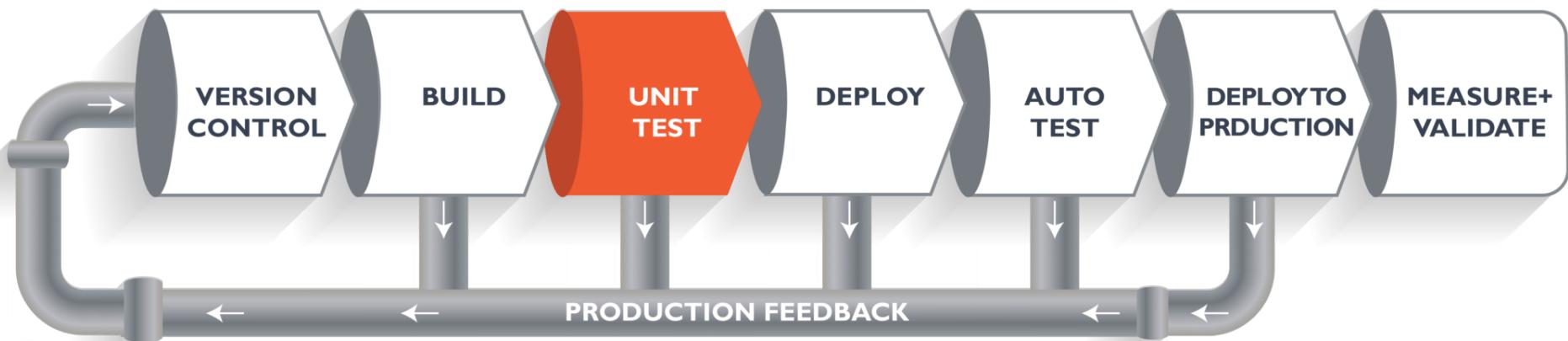
# CI/CD Pipeline

---

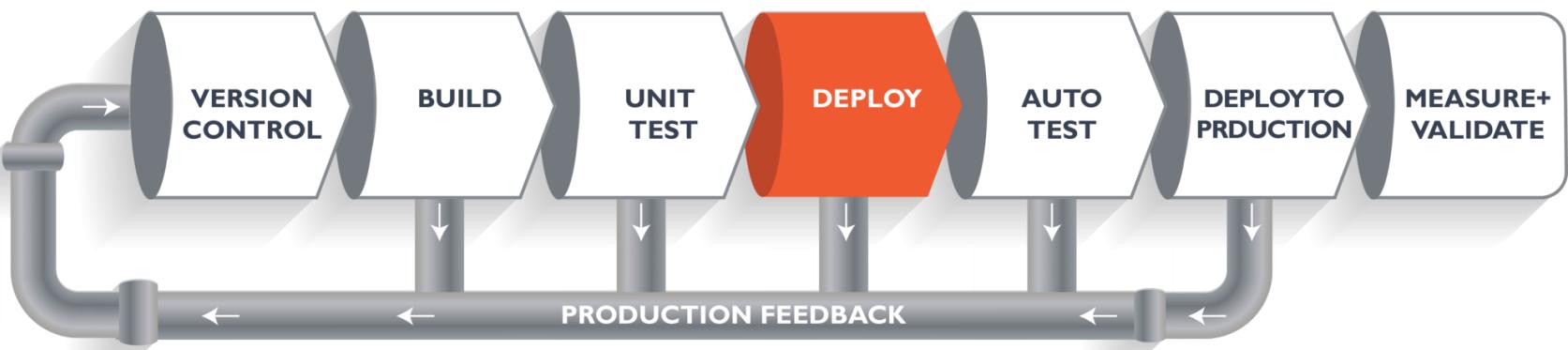


# CI/CD Pipeline

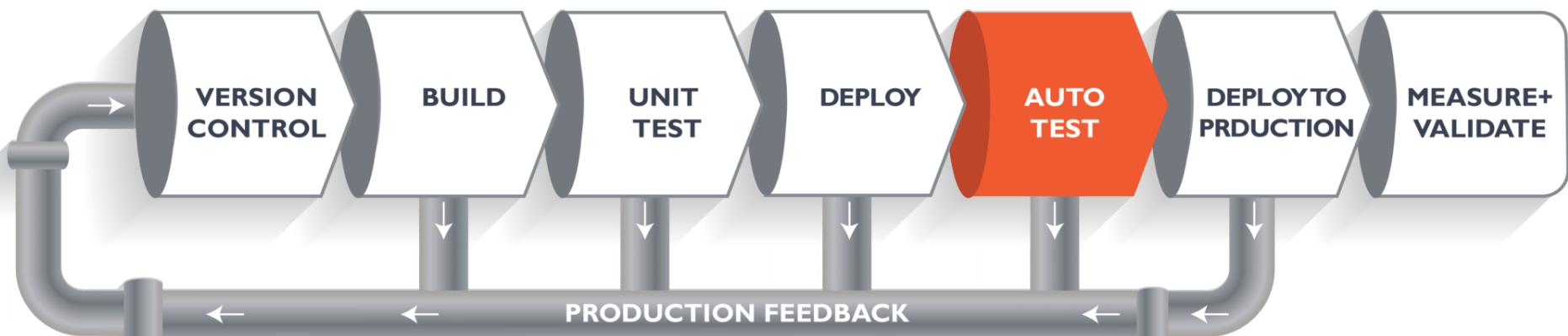
---



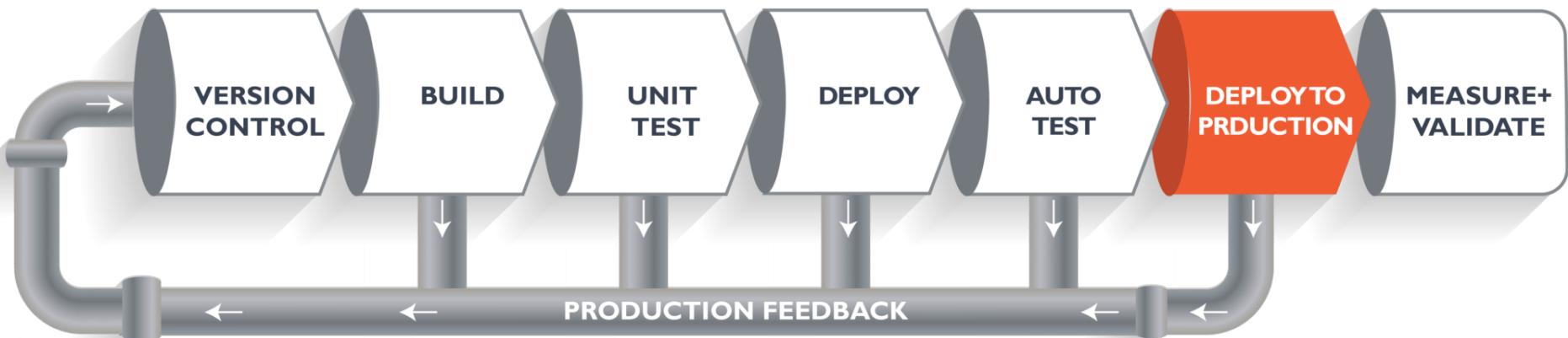
# CI/CD Pipeline



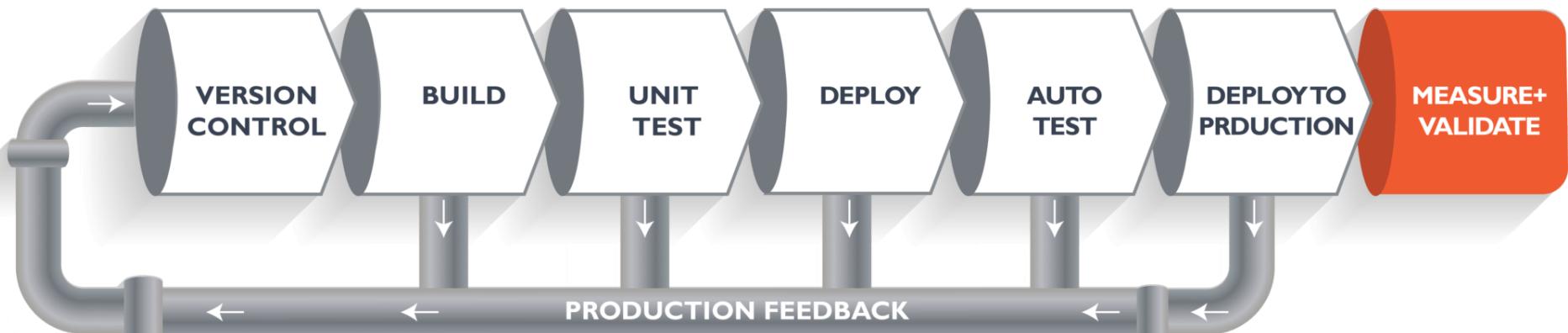
# CI/CD Pipeline



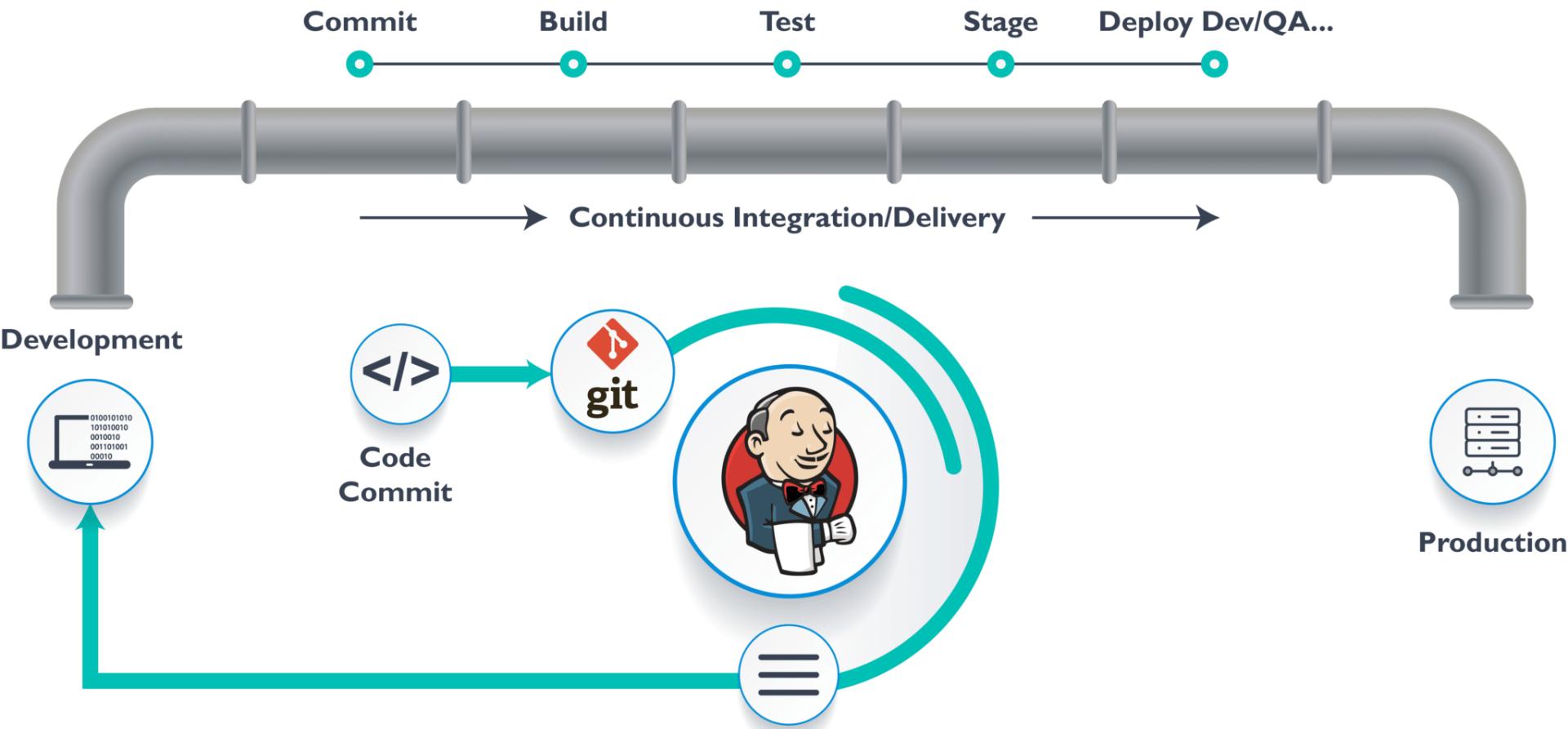
# CI/CD Pipeline



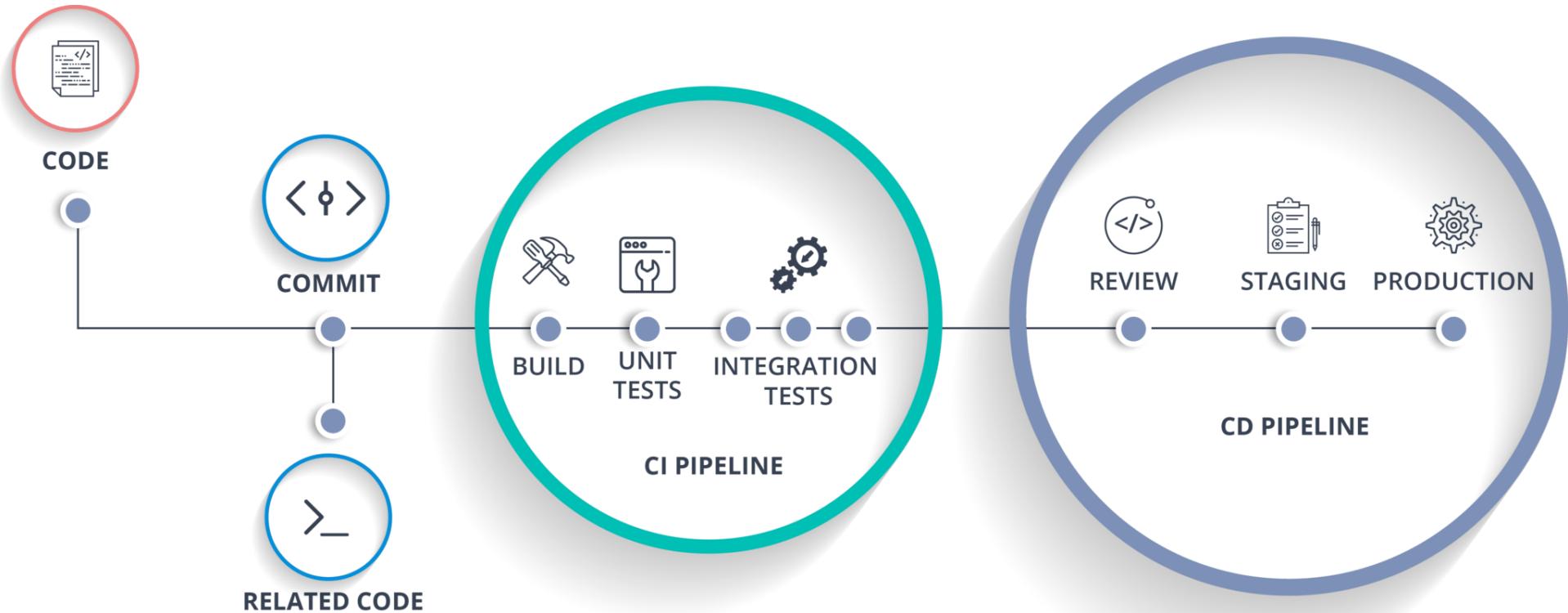
# CI/CD Pipeline



# CI/CD Pipeline

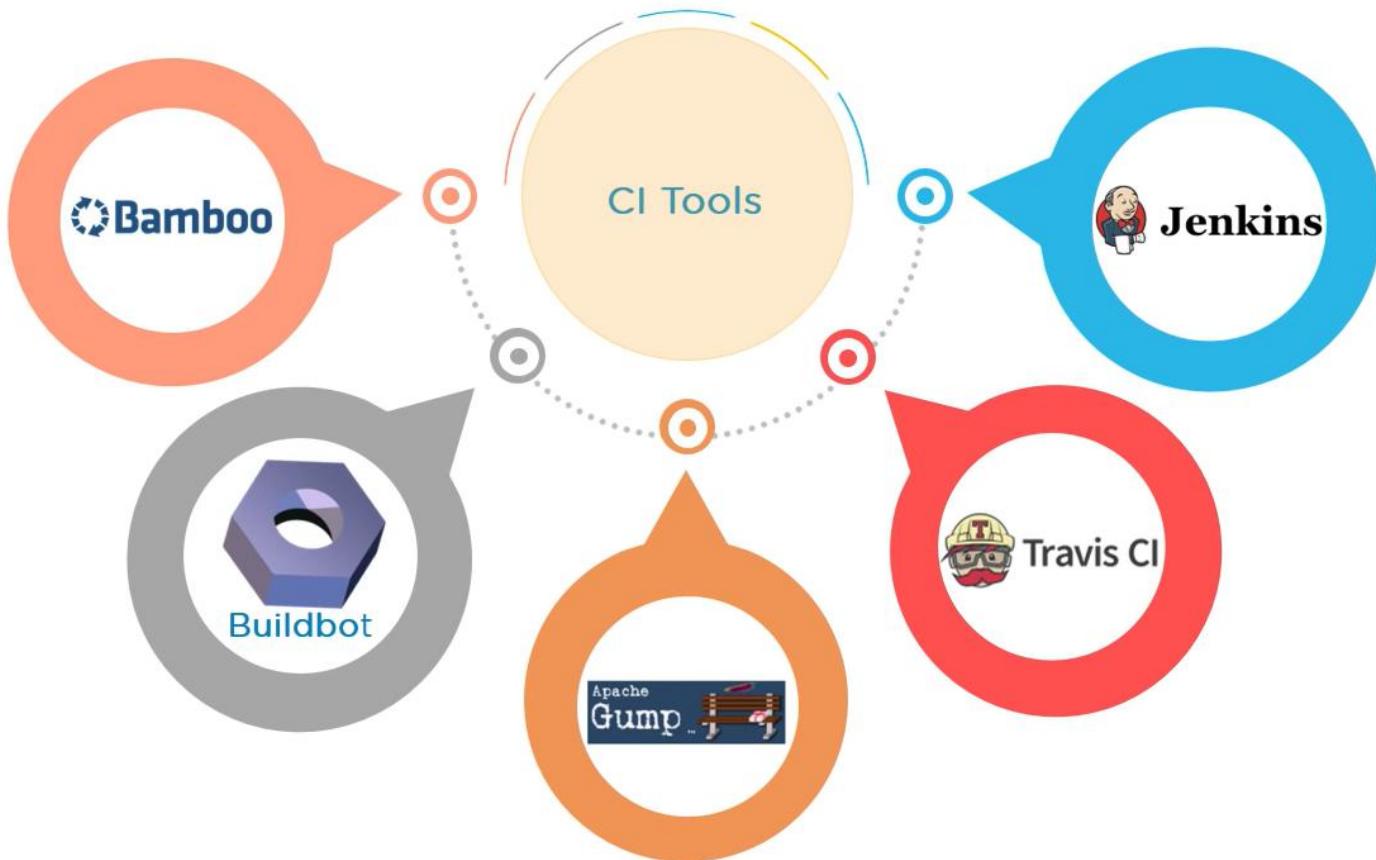


# CI/CD Pipeline



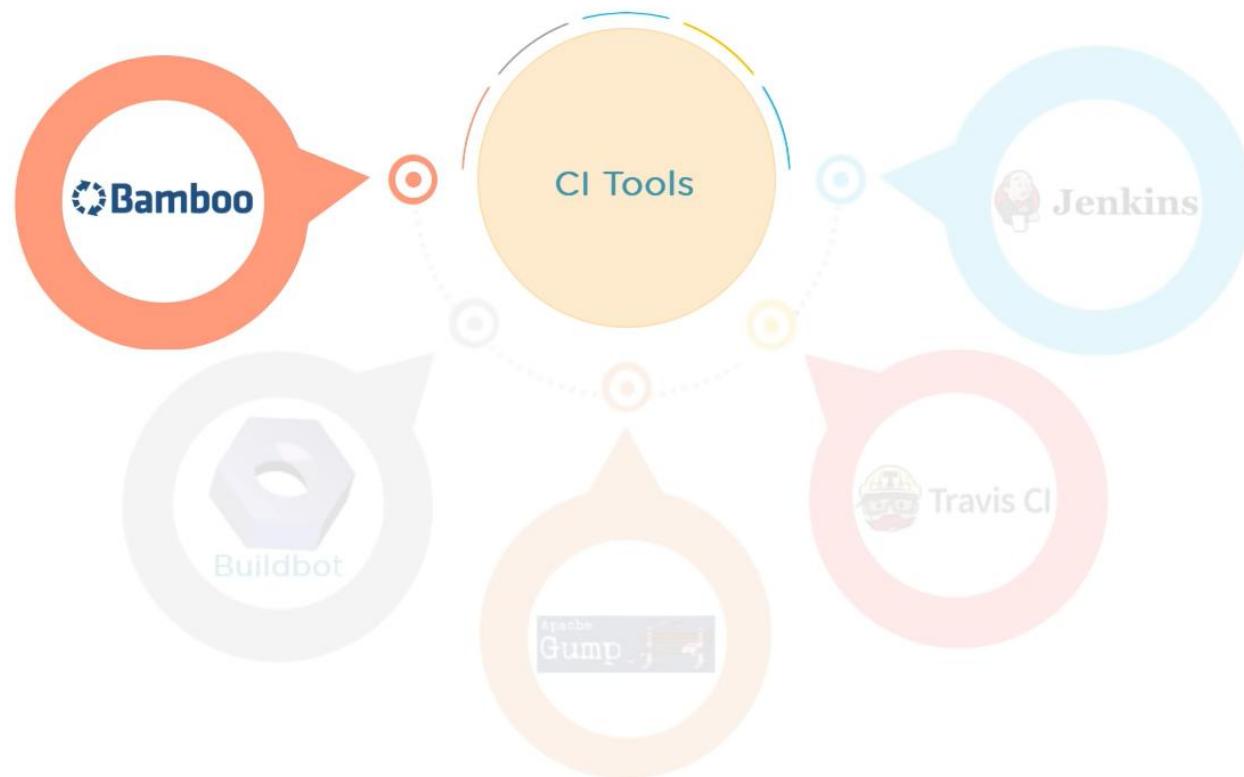
# Continuous Integration Tools

---



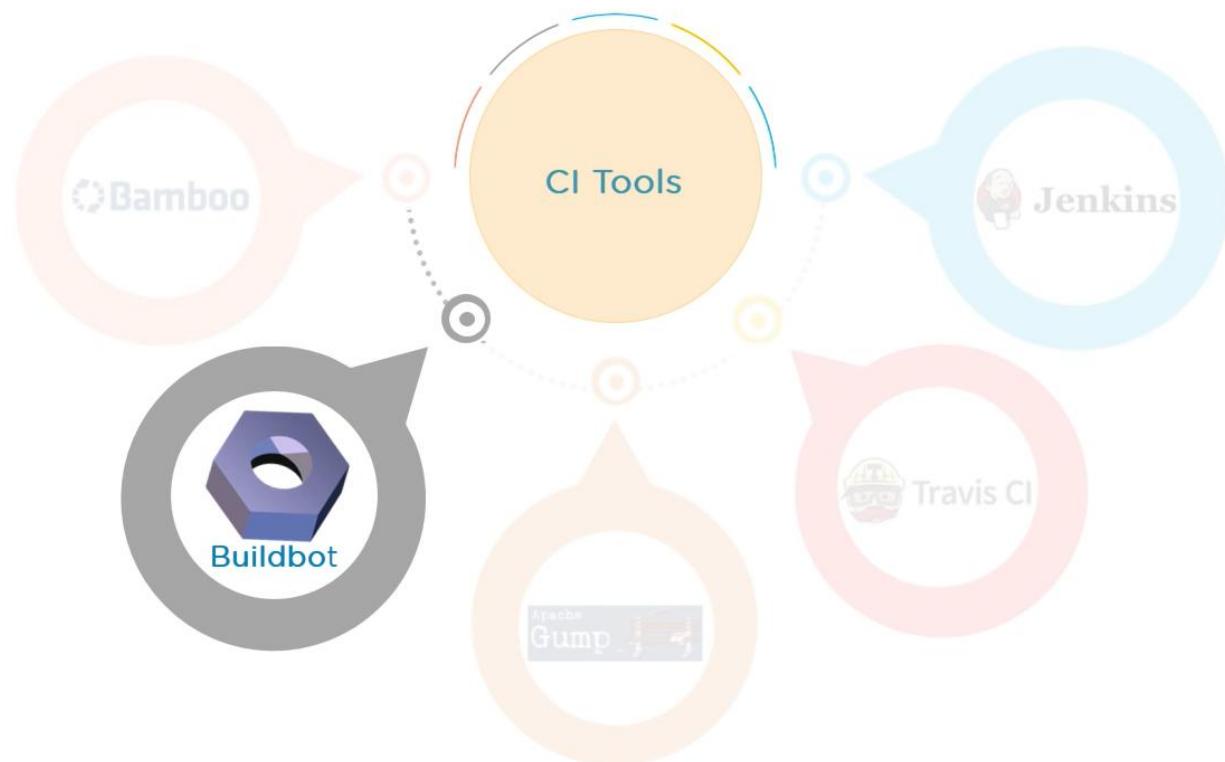
# Continuous Integration Tools

Bamboo is a CI tool that can run multiple builds in parallel for faster compilation. It has built in functionality to connect with repositories and has build tasks for Ant, Maven, etc.



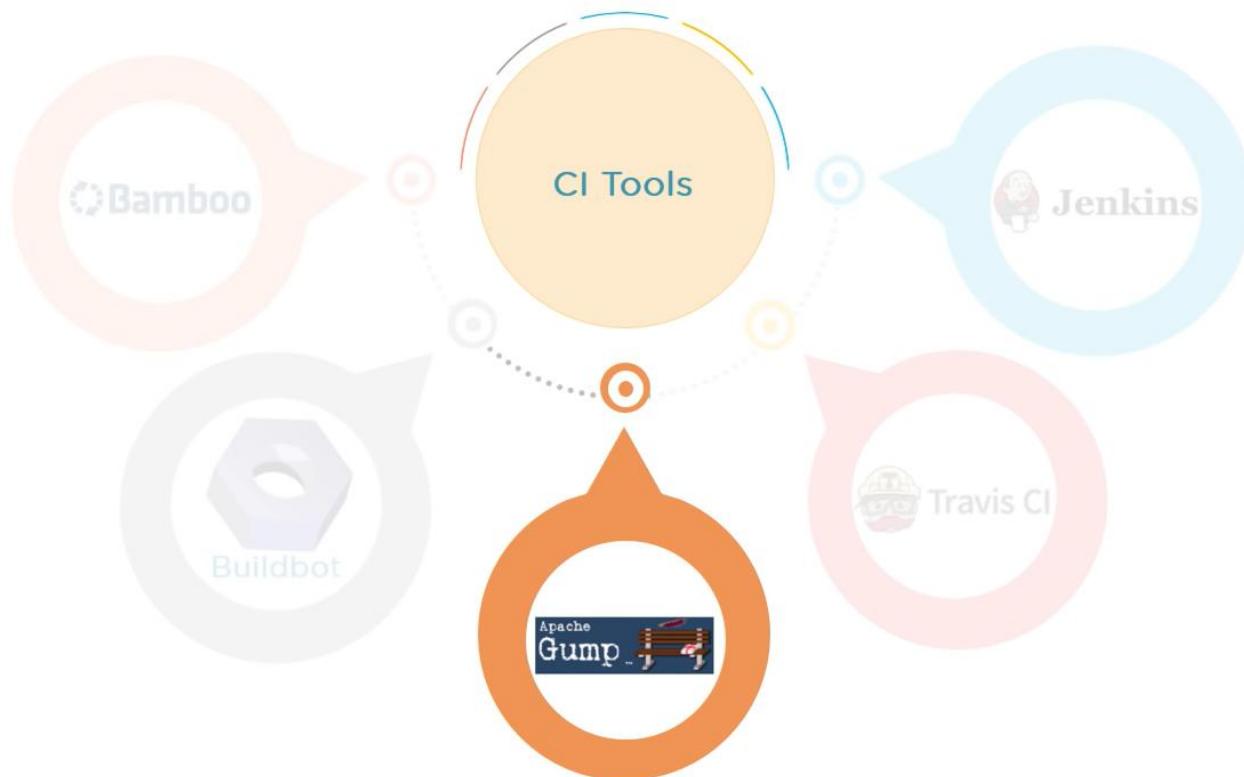
# Continuous Integration Tools

Buildbot is an open-source framework for automating software build, test and release processes. It is written in Python and supports distributed, parallel execution of jobs across multiple platforms.



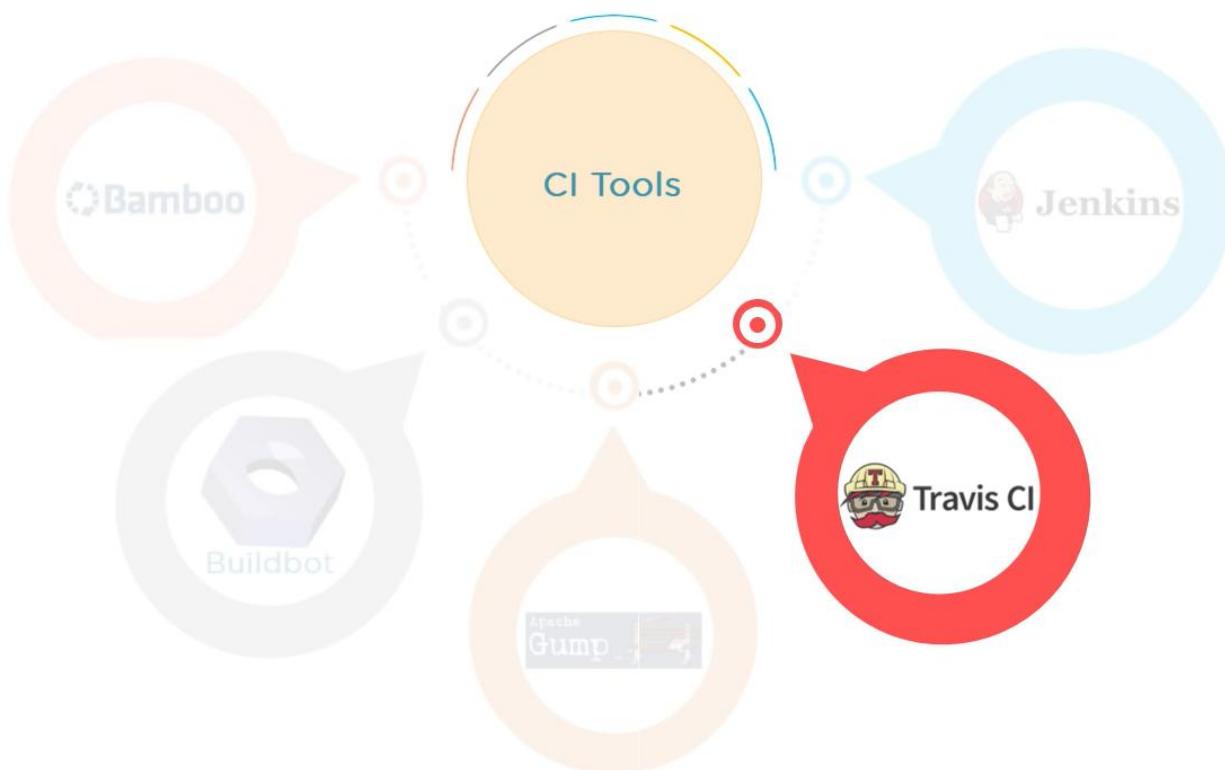
# Continuous Integration Tools

Apache Gump is designed with the aim to build and test all the open source Java projects, every night. It makes sure that all the projects are compatible at both API level and functionality level.



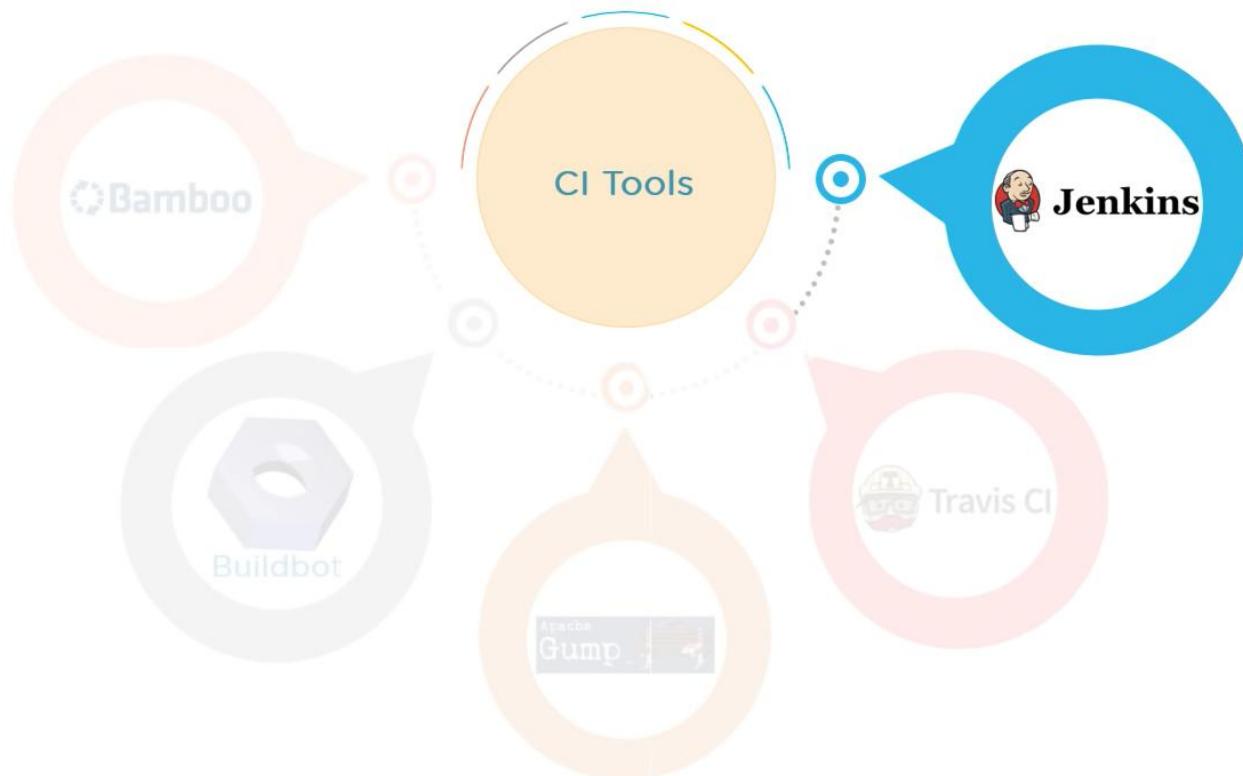
# Continuous Integration Tools

Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub. It's built for projects and team of all sizes and supports over 20 different languages.

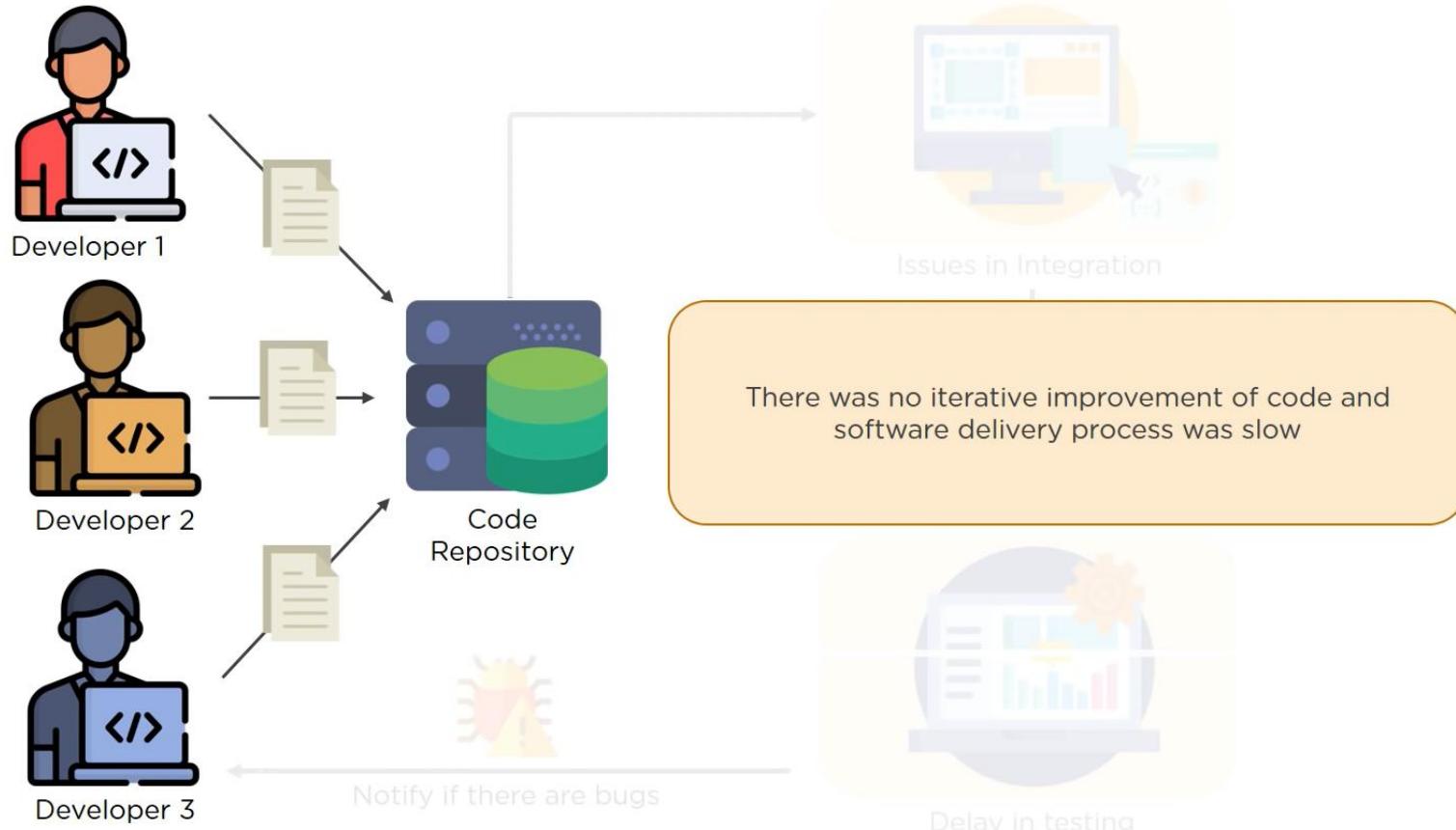


# Continuous Integration Tools

Jenkins is an open source automation server written in Java. It is used to automate software development process via continuous integration and facilitates continuous delivery.

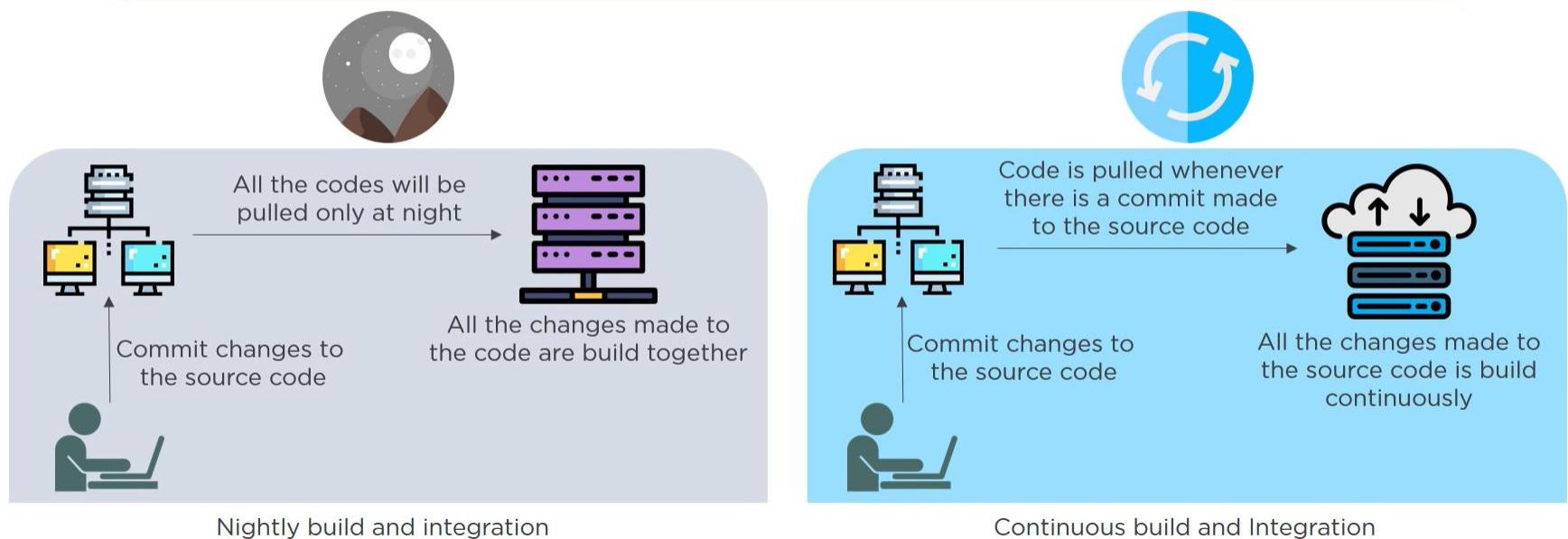


# Problems Before jenkins



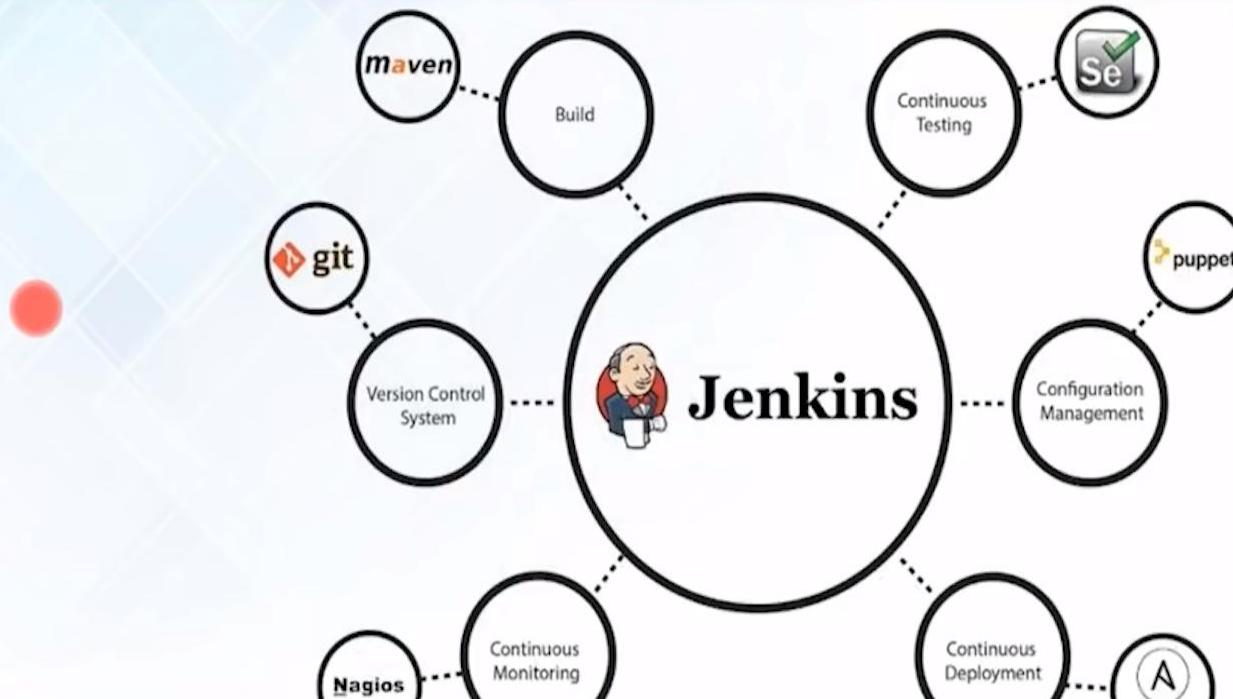
# What is Jenkins

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



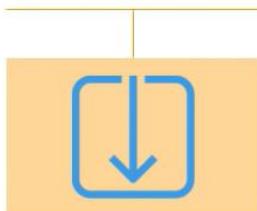
# What is Jenkins

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

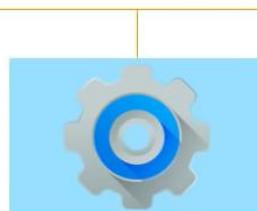


# Jenkins Features

---



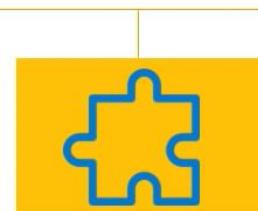
Easy  
Installation



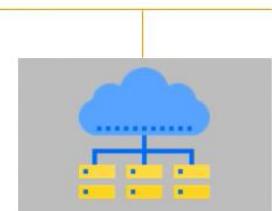
Easy  
Configuration



Plugins



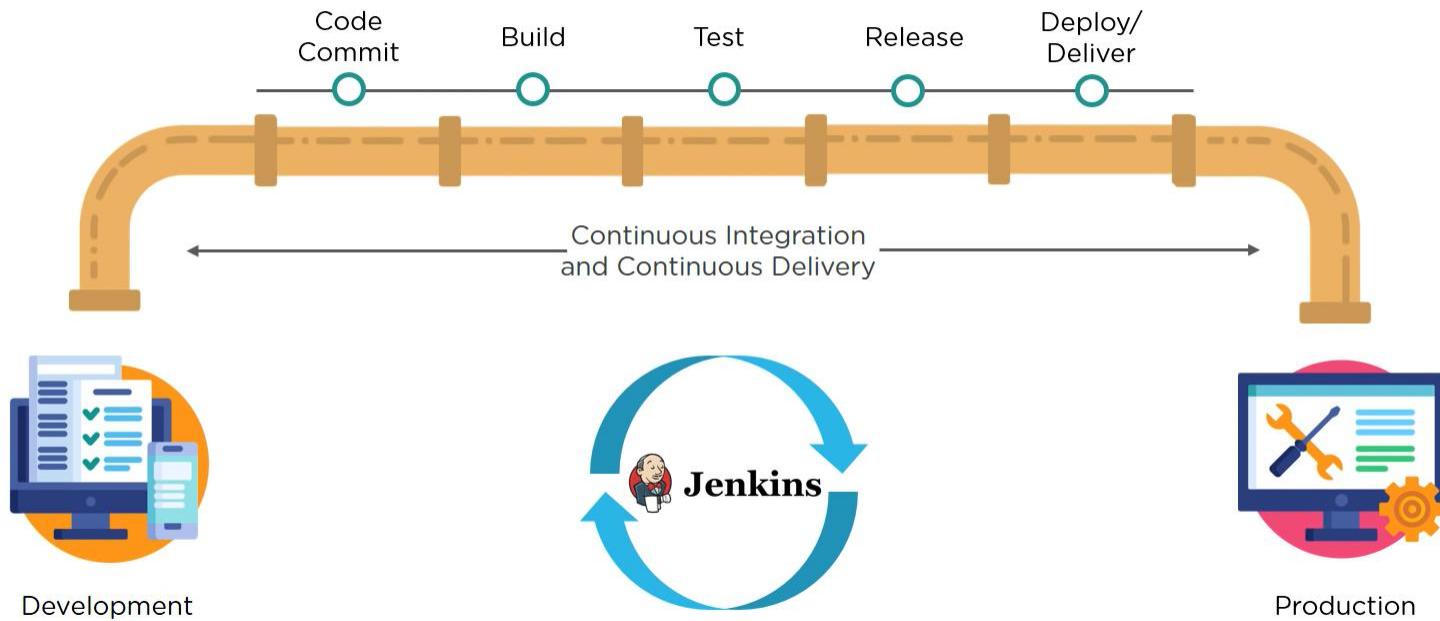
Extensible



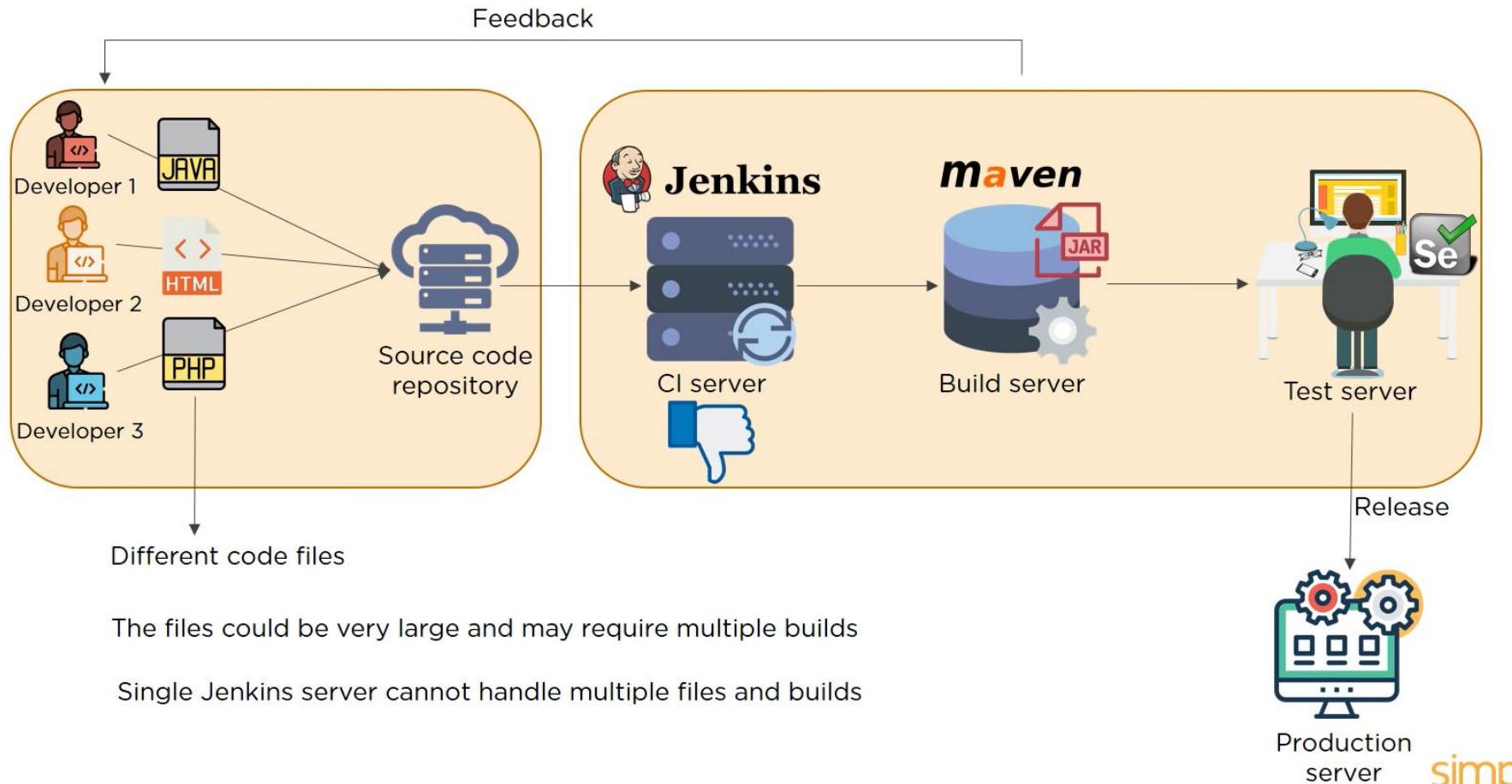
Distributed

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

# Jenkins Pipeline



# Jenkins Architecture



# Shortcomings of Single Jenkins Server



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

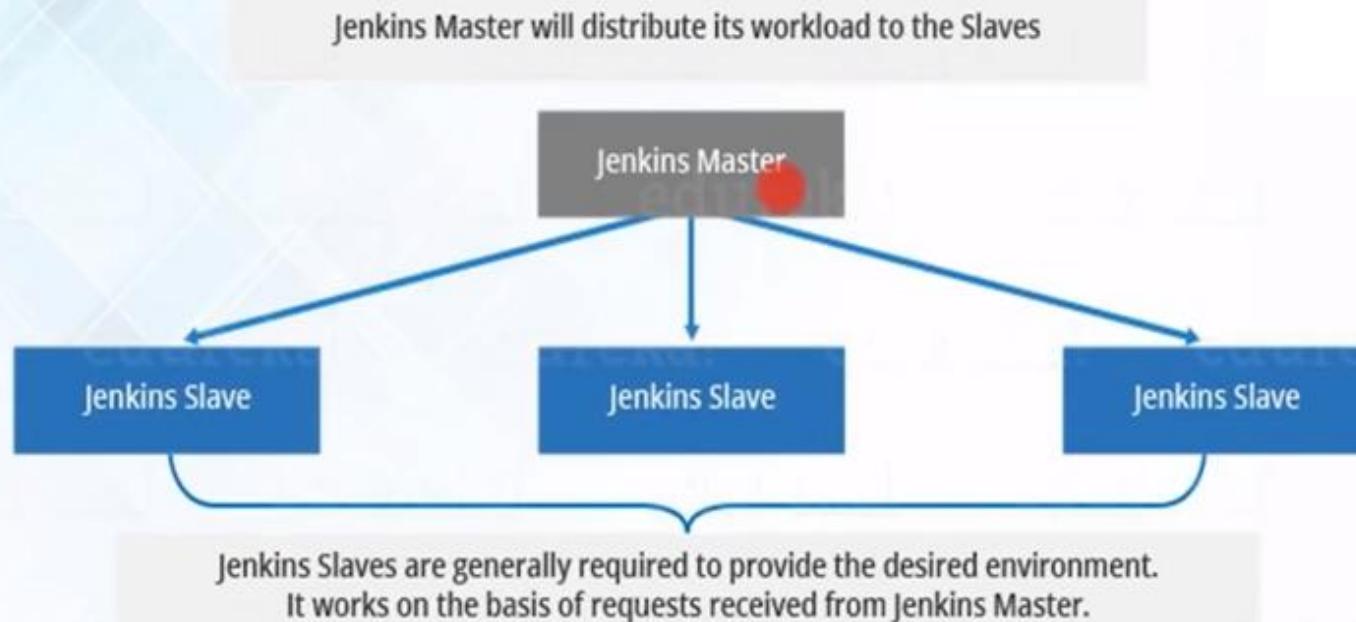


Another build job may require Linux box.

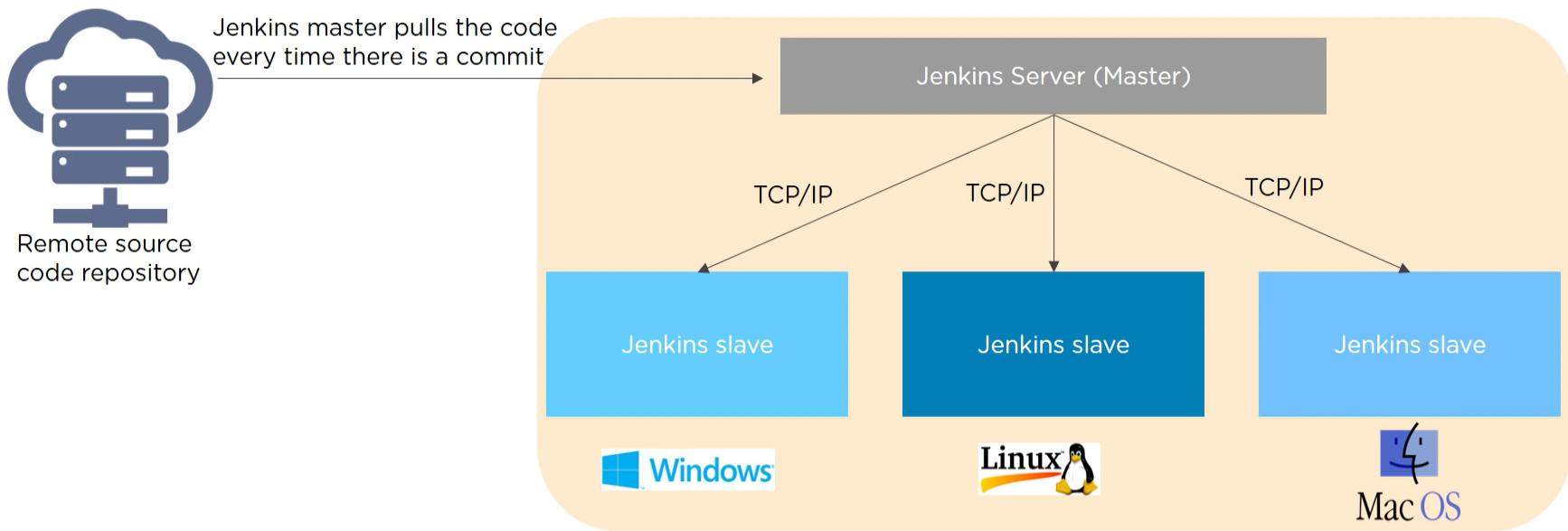


How to manage spikes in build activity

# Jenkins Distributed Architecture



# 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 Case study

---



The illustration features the Bosch logo at the top right. Below it, a man in a suit sits at a desk with a laptop, looking thoughtful. To his left is a red car icon. In the bottom left corner, there's a robotic arm icon with gears. The background is a blurred image of a factory or industrial setting with various machinery and boxes.

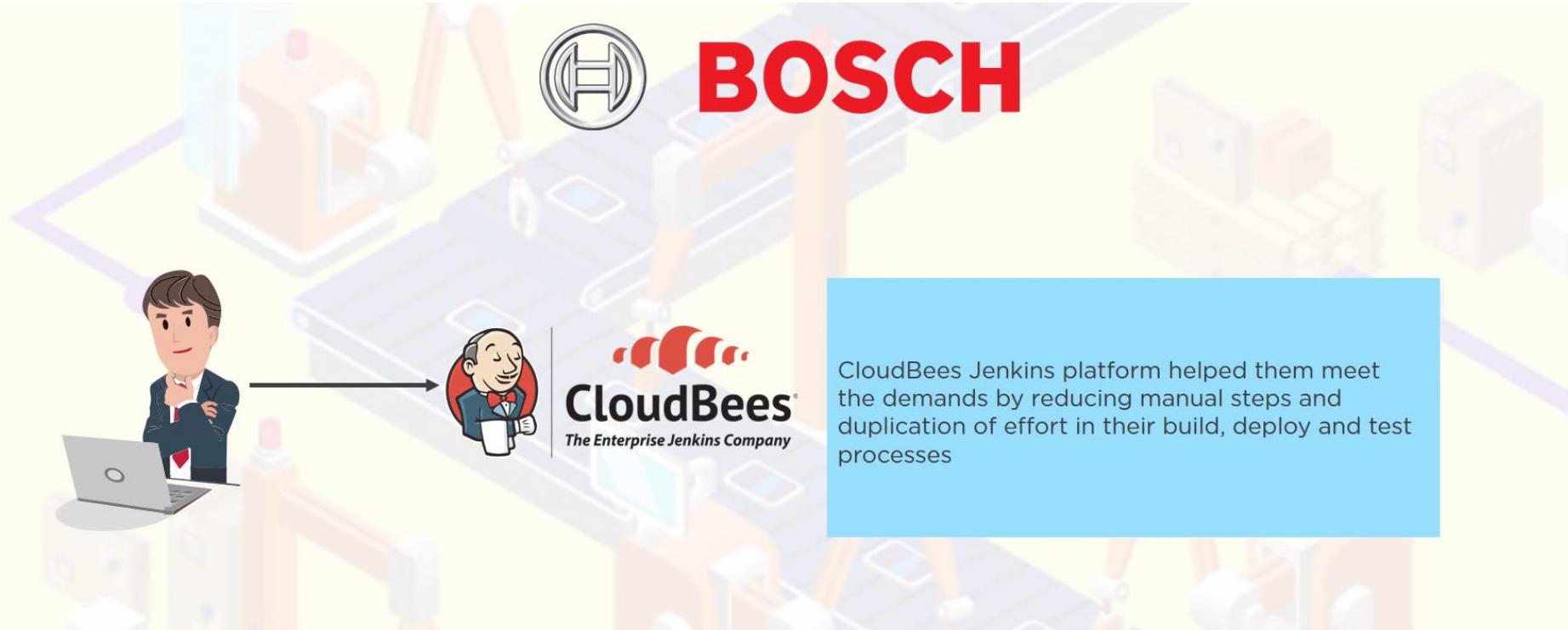
**BOSCH**

BOSCH found a growing need to help its software engineers produce and deliver higher quality software faster

**CHALLANGE**

Manage and streamline the development of increasingly complex automotive software by adopting CI and CD practices to shorten the entire development and delivery process

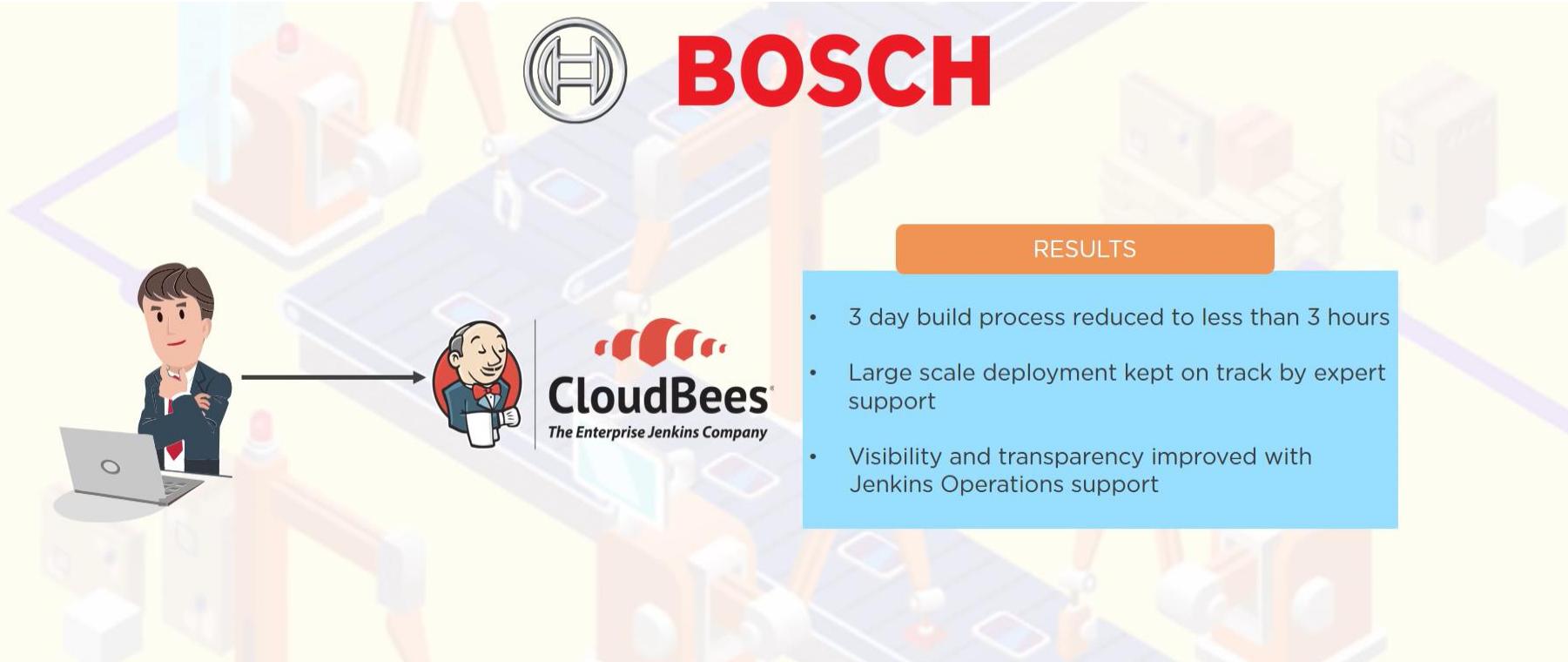
# Jenkins Case study



The illustration features the Bosch logo (a stylized 'H' inside a circle) and the word 'BOSCH' in red. In the foreground, a man in a suit sits at a desk with a laptop, looking thoughtful. An arrow points from him to a cartoon character of a man in a tuxedo holding a trophy. To the right of the character is the CloudBees logo, which includes a red bee icon and the text 'CloudBees' with the subtitle 'The Enterprise Jenkins Company'. The background shows a blurred industrial setting with various machinery and components.

CloudBees Jenkins platform helped them meet the demands by reducing manual steps and duplication of effort in their build, deploy and test processes

# Jenkins Case study



The background of the slide features a stylized illustration of a factory or assembly line. In the foreground, a man in a suit sits at a desk with a laptop, looking thoughtful with his hand on his chin. An arrow points from him towards a central logo. The logo consists of the Bosch 'H' emblem above the word 'BOSCH' in red, and below it is the CloudBees logo with the tagline 'The Enterprise Jenkins Company'. A blue callout box on the right side contains the word 'RESULTS' in white, followed by a bulleted list of achievements.

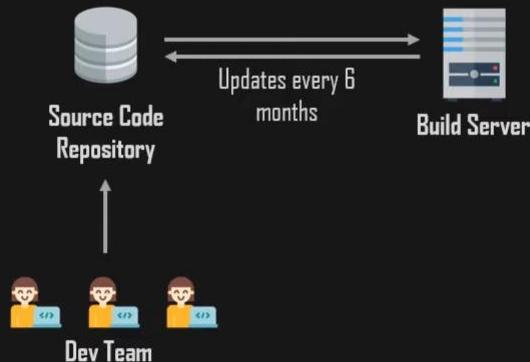
**RESULTS**

- 3 day build process reduced to less than 3 hours
- Large scale deployment kept on track by expert support
- Visibility and transparency improved with Jenkins Operations support

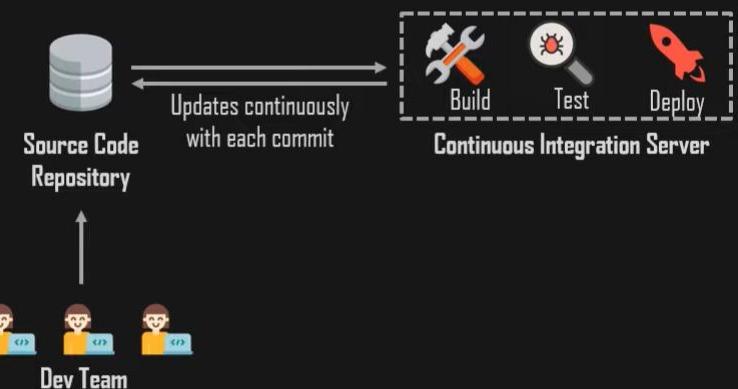
## Jenkins Case study

# CASE STUDY: ADOBE

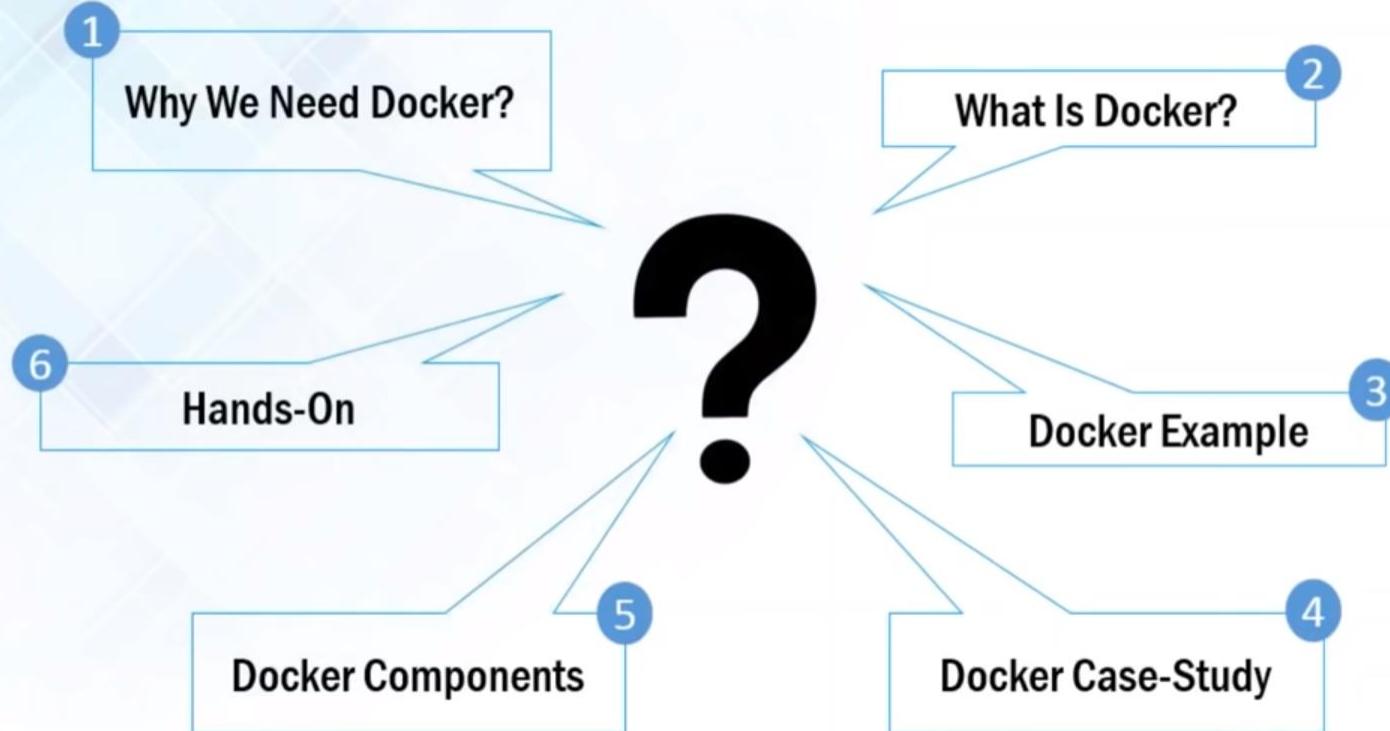
Adobe before CI: Semi Annual Releases



Adobe after CI: 60% Faster Releases

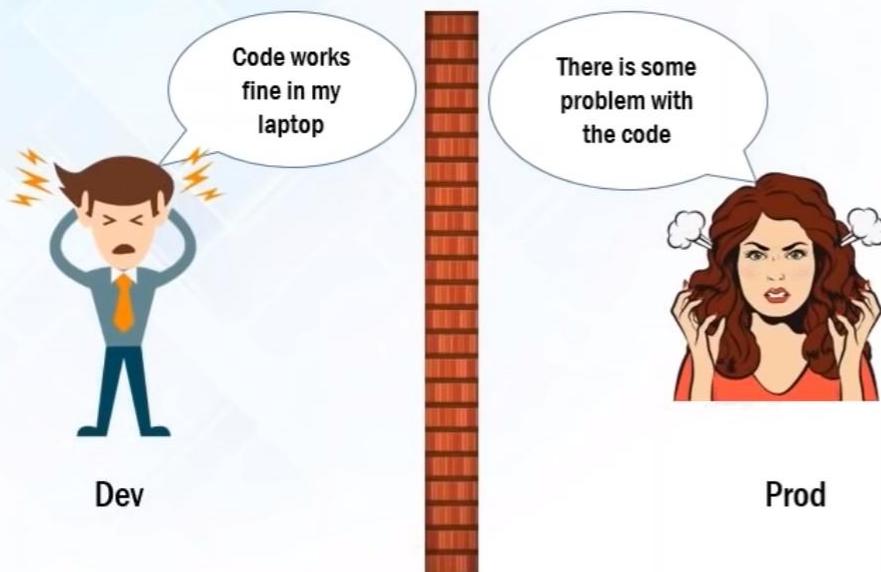


# Do you know the following?



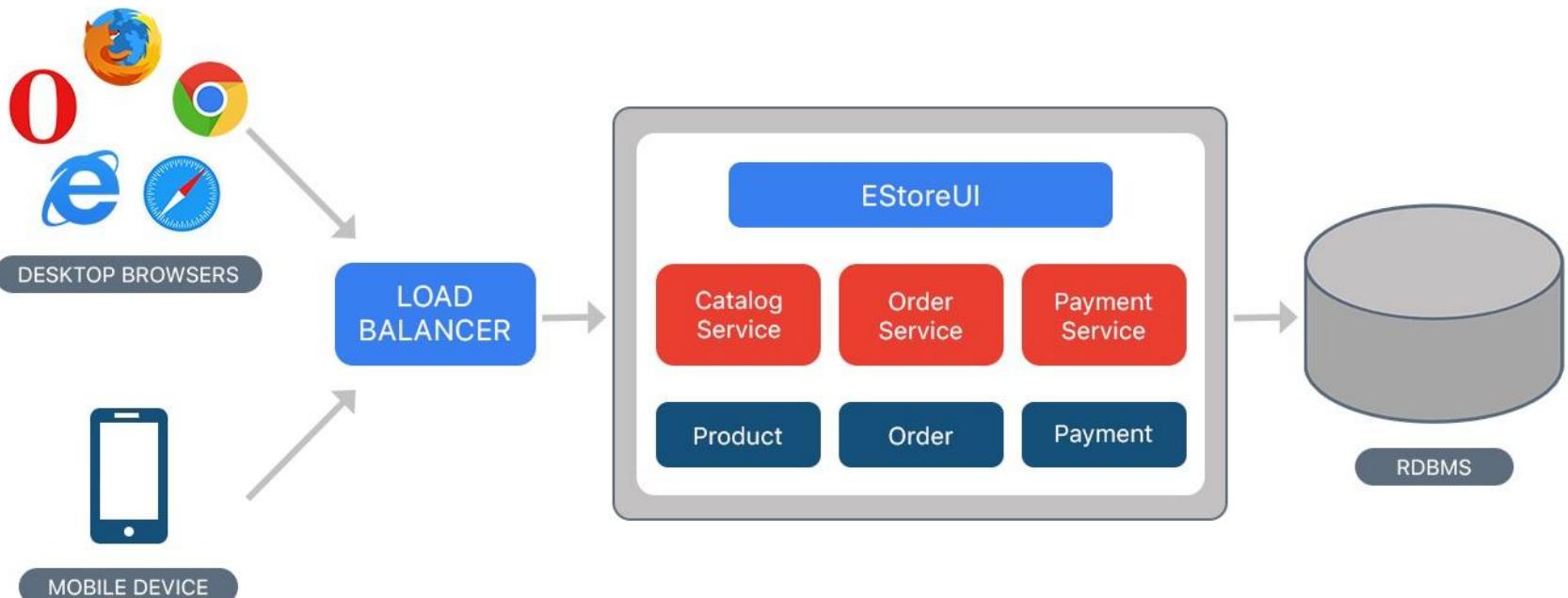
# Problems Before Docker

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.

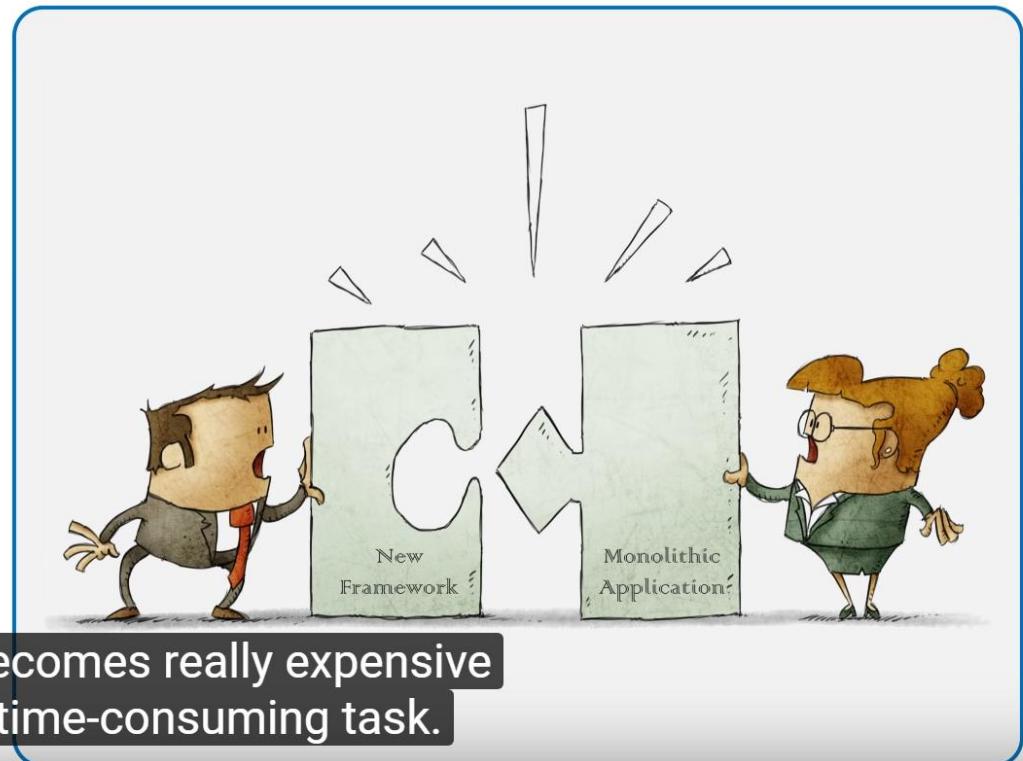


In Dev there can be a software that is upgraded and in Prod the old version of software might be present

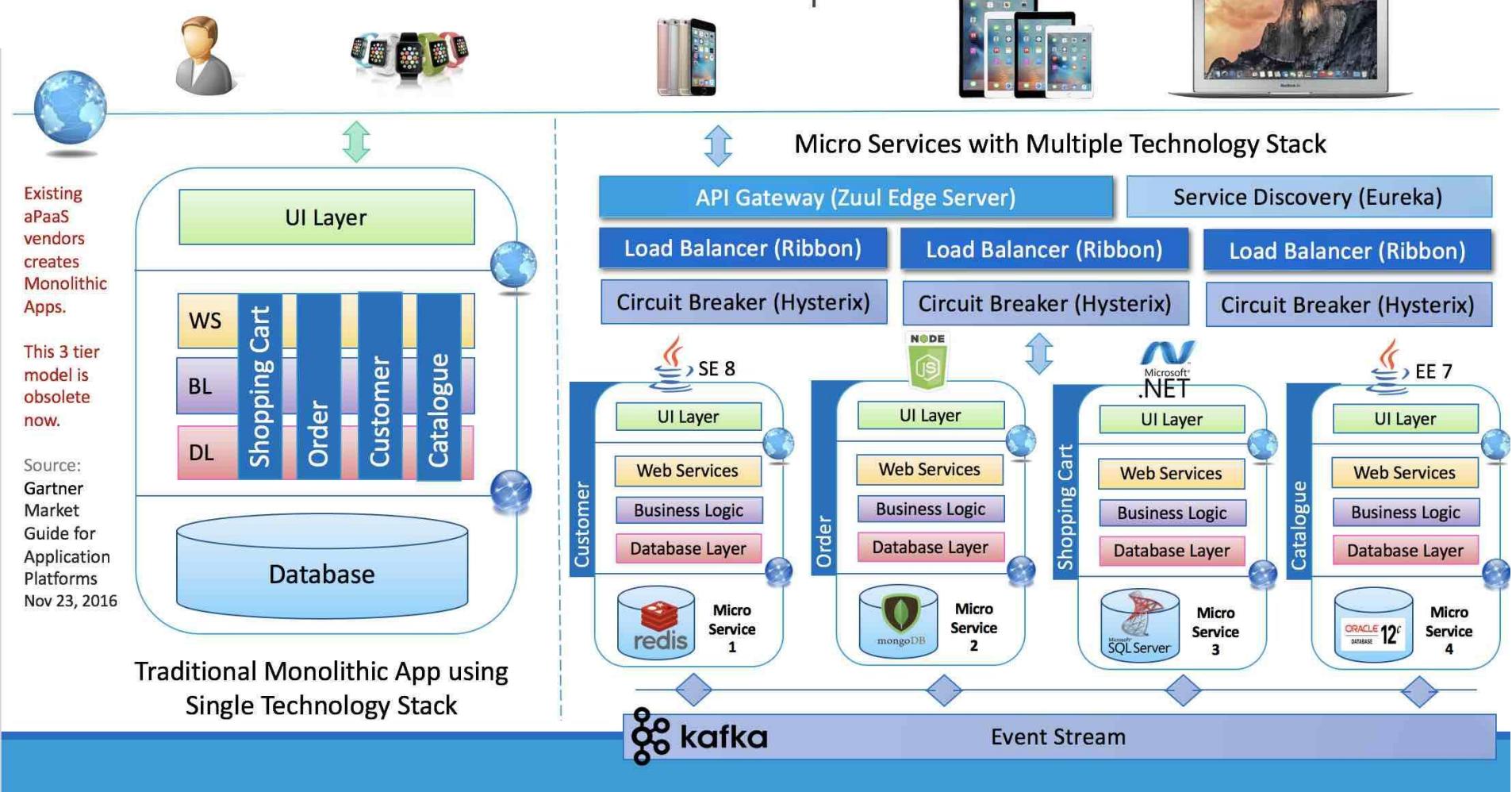
# Example of Monolithic Approach



# Monolithic Architecture

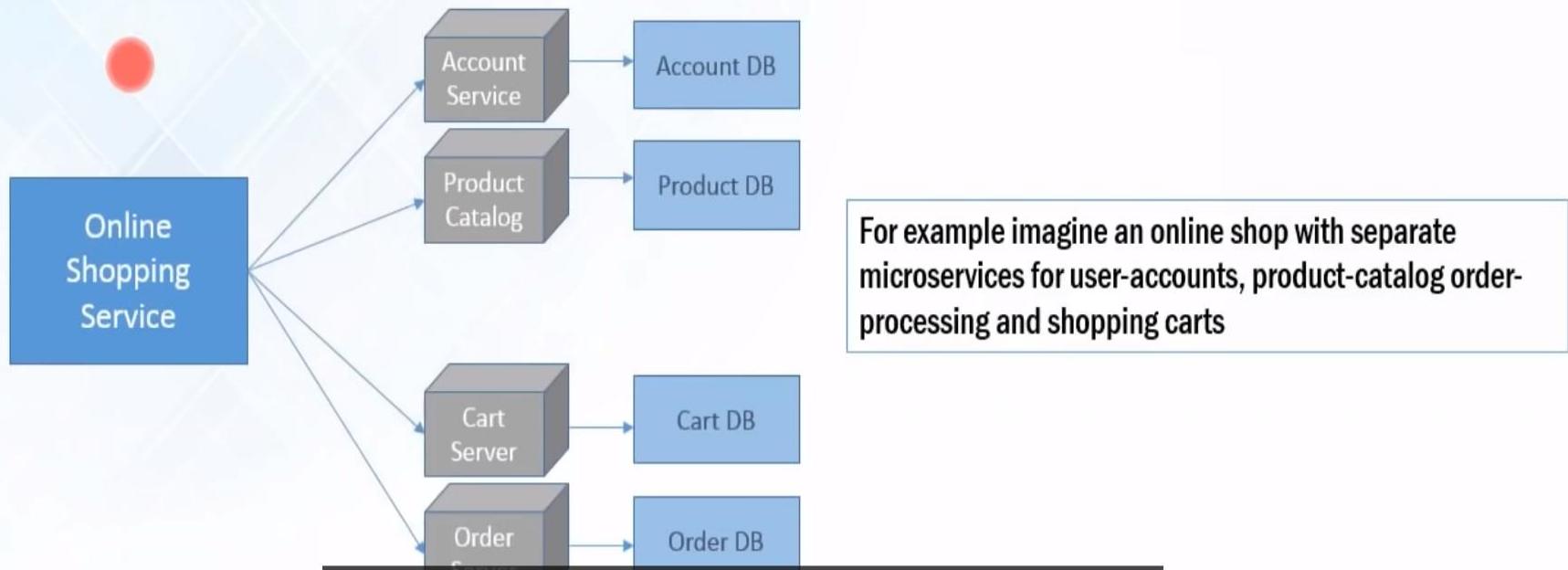


# Monolithic vs. Micro Services Example



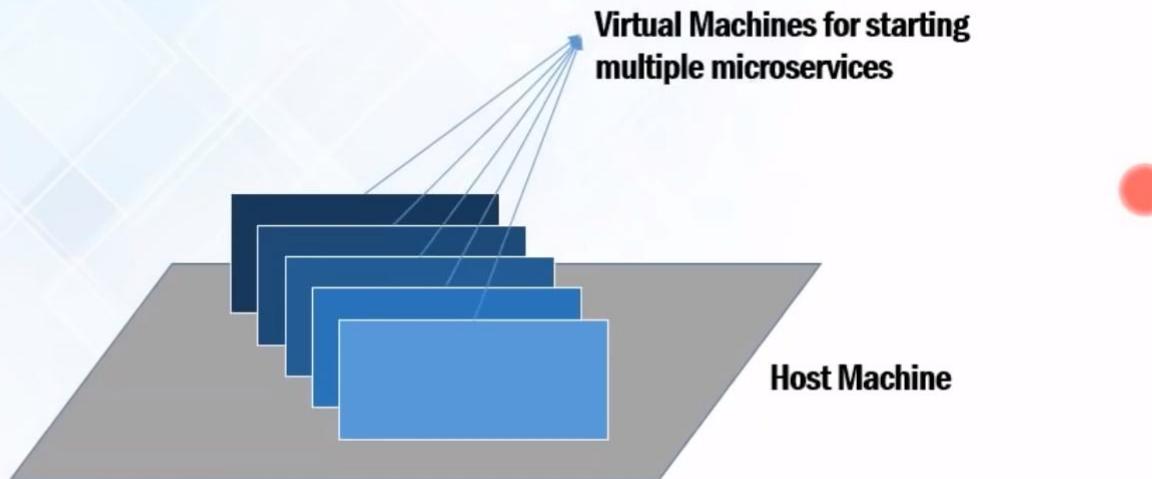
# Problems Before Docker

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

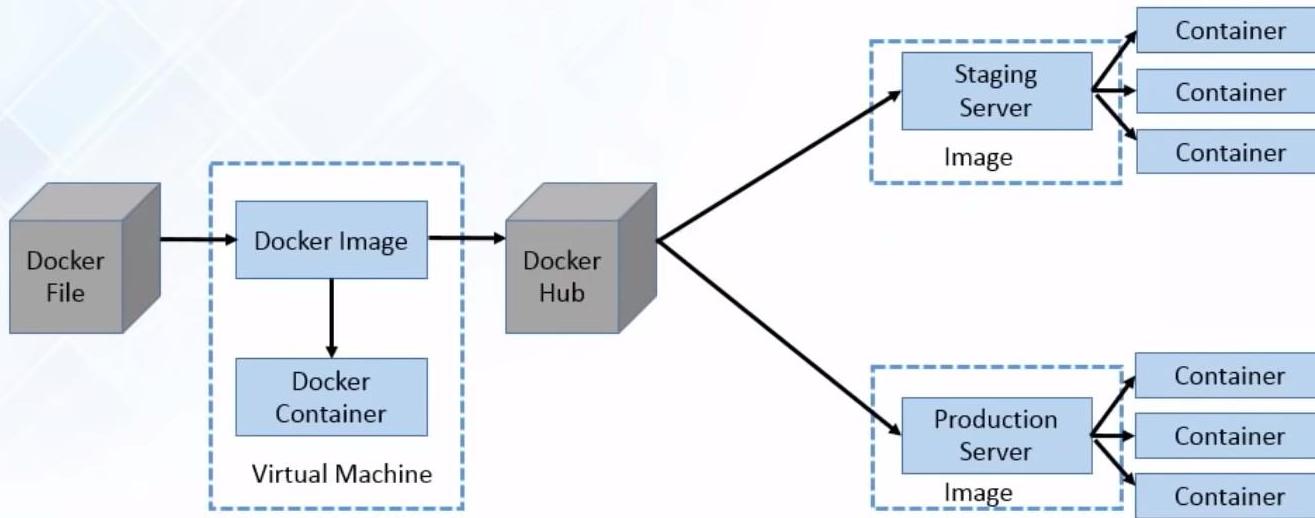
Developing an application requires starting several of microservices in one machine. So if you are starting five of those services you require five VMs on that machine.



# Docker in a Nutshell

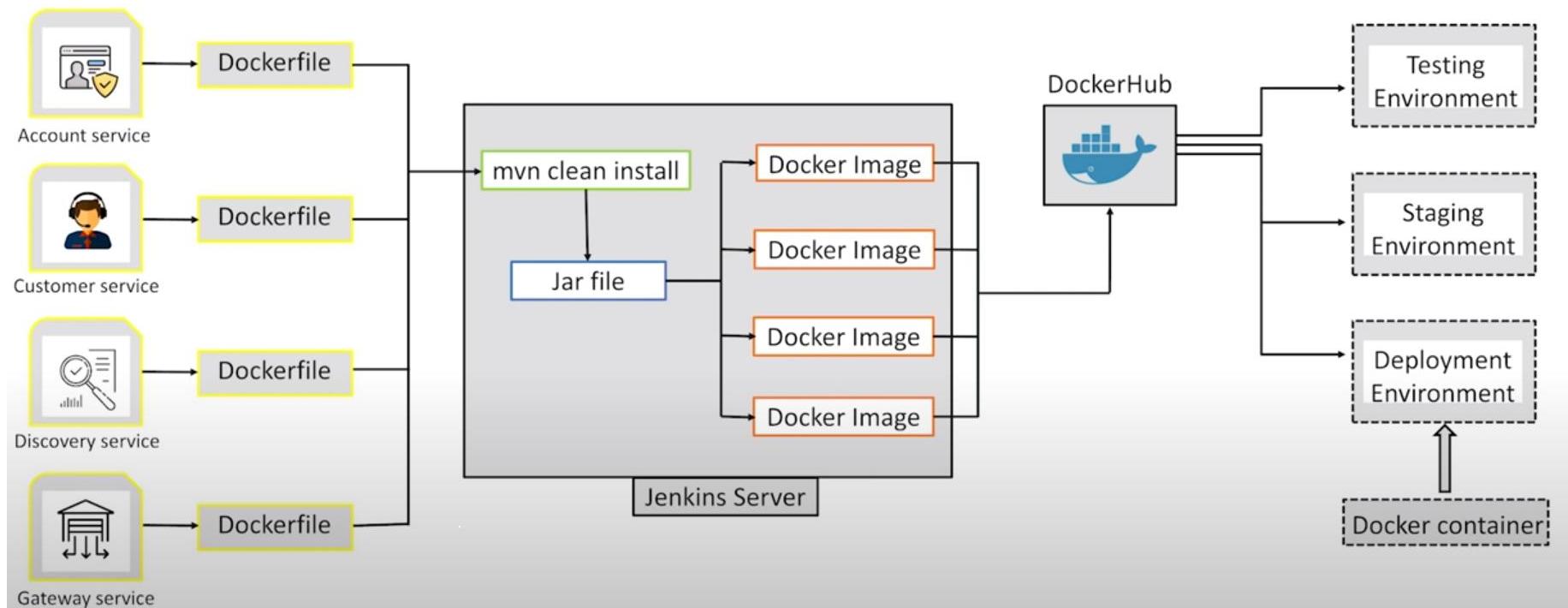


- Docker file builds a Docker image and that image contains all the project's code
- You can run that image to create as many Docker containers as you want
- Then this Image can be uploaded on Docker hub, from Docker hub any one can pull the image and build a container



# Continuous delivery through DevOps Integration

## Solution



## Docker jenkins

---

- <https://www.jenkins.io/doc/book/installing/docker/#setup-wizard>
- Refer G:\Local disk\ Docker\jenkins
- Follow blue ocean setup with dockerfile

# 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
```

# Large Repos are unhealthy

---

- Large git repos (2 GB+) often are a sign of git misuse (large binaries, etc.)
- Yes, there are cases where a large git repo is not a sign of misuse
  - Linux kernel repo is over 1 GB
  - Linux kernel repo is one of the oldest git repos
- ○ Git is able to handle large repos, some operations are slower on some systems
- Git misuse doesn't matter - we need to run our Jenkins jobs anyway
  - The 18 GB+ repository at my work isn't pretty, but it is business critical...

# Jenkins tricks for large repos

---

- Use command line git rather than JGit
  - Command line git explicitly manages memory and handles large repos
  - JGit large repo support has improved, will likely always lag behind command line git
- Reduce data transfer & disc space for history
  - ○ Reference repositories - only clone what's new
  - ○ Shallow clone - only clone recent history
  - ○ Don't fetch tags
  - ○ Narrow refs - only clone specific branches
- Reduce disc space for working directory
  - ○ Sparse checkout

# Jenkins free style project sparse checkout

Only specific project from repo

The screenshot shows the Jenkins Source Code Management configuration page for a free-style project. A blue arrow points from the text "Only specific project from repo" to the "Sparse Checkout paths" section.

**General**    **Source Code Management**    Build Triggers    Build Environment    Build    Post-build Actions

**Branch Specifier (blank for 'any')**  (x) (?)

**Add Branch**

**Repository browser**  (?) ▼

**Additional Behaviours**

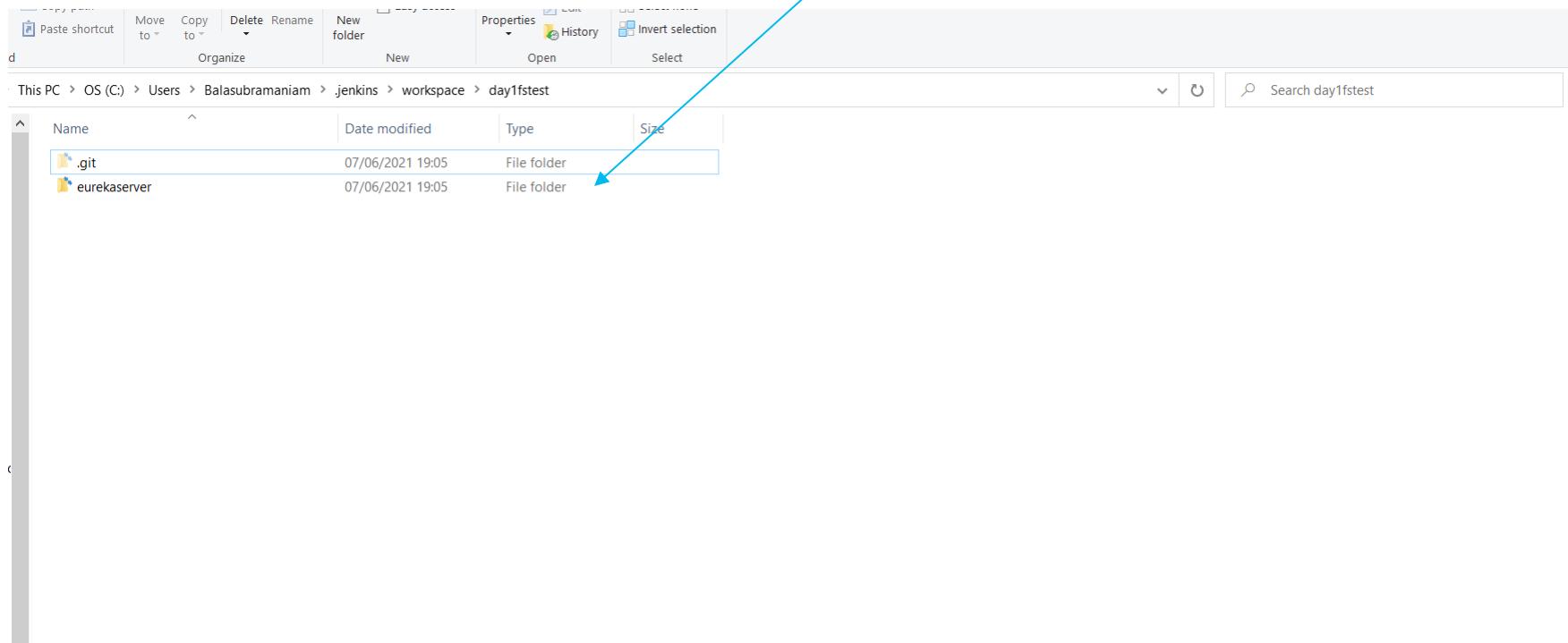
**Sparse Checkout paths** (x) (?) (x)

**Path**

**Add**

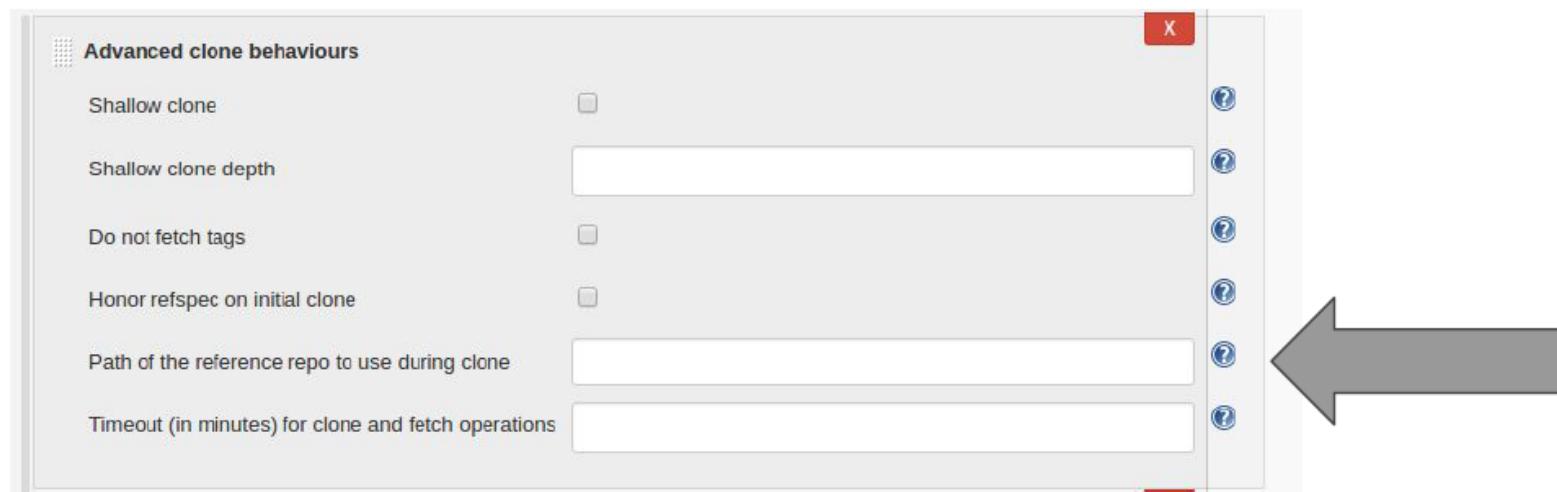
# Jenkins free style project sparse checkout

Only specific project from repo



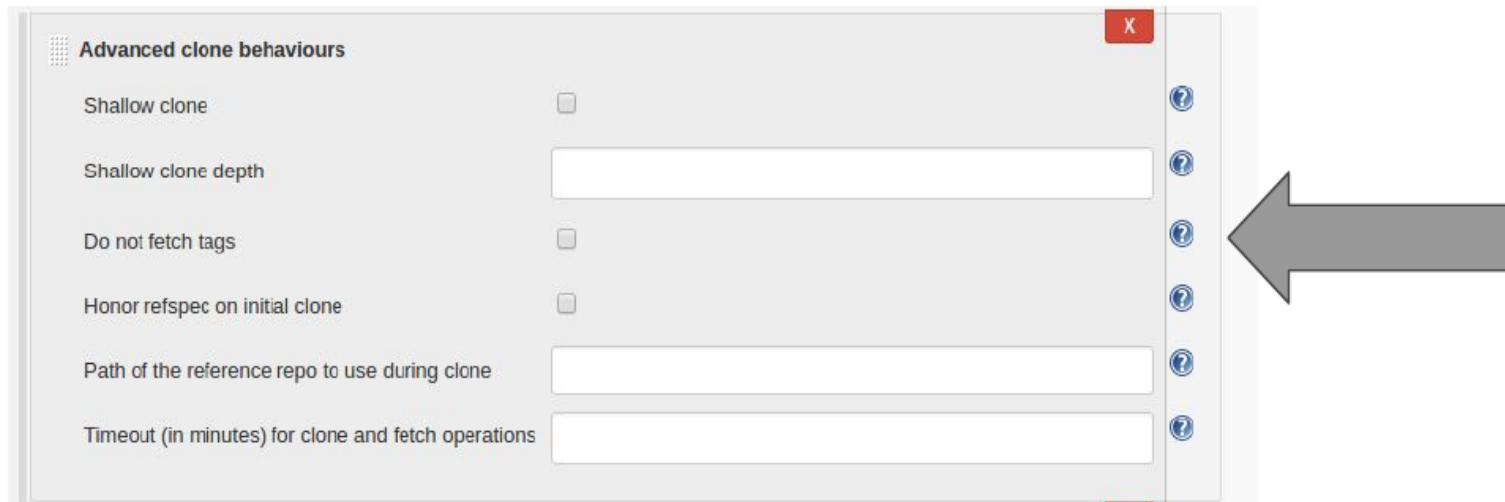
# Reference repository - pointers, not copies

- A bare git repository can be used as a “reference repository” for other repositories
- Git creates pointers to the reference repo
  - Saves space by not copying data to disc
  - Saves time by not transferring over the wire



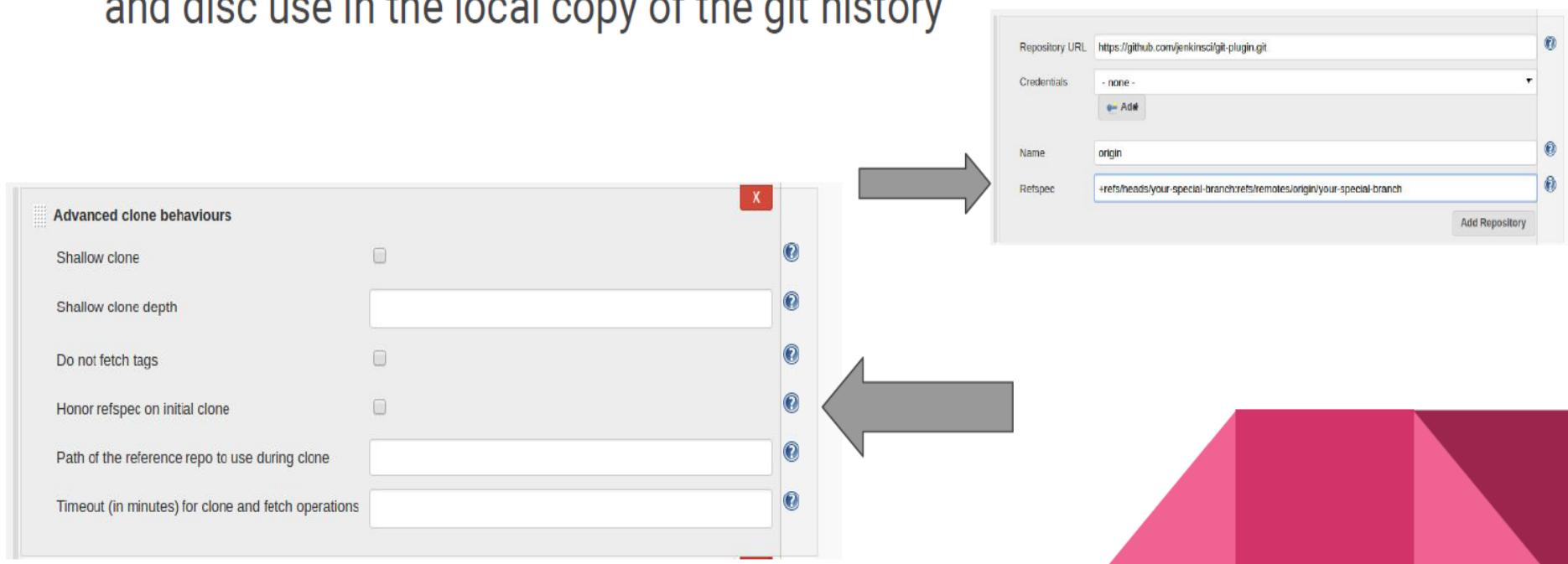
# Don't Fetch Tags

- Most Jenkins jobs don't need tags
- Skipping tags can reduce data transfer and local disc use



# Narrow refsspecs - clone only what you need

- Git refspec describes what to copy from remote
- Reducing the refspec can reduce the data transfer and disc use in the local copy of the git history



# Jenkins Freestyle Project for Docker Container Images with SonarQube

SonarQube Scanner

Name  
sonarqube

Install automatically ?

**Install from Maven Central**

Version  
SonarQube Scanner 4.6.2.2472 ▾ Delete Installer

**Install from Maven Central**

Version  
SonarQube Scanner 4.6.2.2472 ▾ Delete Installer

Add Installer ▾ Delete SonarQube Scanner

Add SonarQube Scanner

Save Apply

# Jenkins Freestyle Project for Docker Container Images with SonarQube

The screenshot shows the configuration page for a Jenkins Freestyle Project. The project is titled "Execute SonarQube Scanner".

**Task to run:** (empty field)

**JDK:** java8 (selected from dropdown)

**Path to project properties:** (empty field)

**Analysis properties:**

```
sonar.projectKey=uploaddocker
sonar.projectName=C:/Users/Balasubramaniam/jenkins/workspace/day1fsdockercproject/uploaddocker
sonar.projectVersion=1.0
sonar.login=admin
sonar.password=admin
sonar.language=java
sonar.sources=C:/Users/Balasubramaniam/jenkins/workspace/$JOB_NAME
sonar.java.binaries=C:/Users/Balasubramaniam/jenkins/workspace/$JOB_NAME/uploaddocker/target/classes
sonar.sourceEncoding=UTF-8
```

**Buttons at the bottom:**

- Save
- Apply

# Jenkins Freestyle Project for Docker Container Images with SonarQube

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

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

**Filters**

**Quality Gate**

- Passed: 4
- Warning: 0
- Failed: 0

**Reliability (Bug)**

- A: 3
- B and worse: 1
- C and worse: 1
- D and worse: 0
- E: 0

**Security (Vulnerability)**

- A: 4
- B and worse: 0
- C and worse: 0
- D and worse: 0
- E: 0

**Maintainability (Code Smell)**

- A: 4
- B and worse: 0
- C and worse: 0
- D and worse: 0
- E: 0

Perspective: Overall Status Sort by: Last analysis date Search by project name or key 4 projects

**C:/Users/Balasubramaniam/jenkins/workspace/day1fsdockerproject/uploaddocker** Passed Last analysis: June 7, 2021, 9:57 PM XS 573 Java

0 A Bugs 0 A Vulnerabilities 16 A Code Smells 0.0% Coverage 0.0% Duplications

**C:/Users/Balasubramaniam/jenkins/workspace/NetabankingMavenProject** Passed Last analysis: June 6, 2021, 7:57 PM XS 18 Java

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

**netbanking** Passed Last analysis: September 30, 2020, 9:45 AM XS 115 XML, Java

0 A Bugs 0 A Vulnerabilities 0 A Code Smells 0.0% Coverage 0.0% Duplications

**day14springmvcjpashopping** Passed Last analysis: October 3, 2019, 3:09 PM XS 867 Java, XML

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

# 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 ^

272

# SonarQube

A screenshot of a Windows Task Manager window titled "SonarQube". The window displays a list of processes for the application "SonarQube". The processes are listed in a tree view under the "SonarQube" node. The processes include:

- 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 -Xms512m -Xmx512m -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 Task Manager window also shows the system tray, taskbar icons, and a search bar at the bottom.

# 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 ENG

# 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

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

Filters

Quality Gate

Status	Count
Passed	1
Warning	0
Failed	0

day14springmvcjpasshopping Passed

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

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

XS 867 Java, XML

Reliability ( Bugs )

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

1 of 1 shown

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

09:06 30/09/2020 ENG

# 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

```
[INFO]
[INFO] 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
[INFO] 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
[INFO] 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
```

Type here to search



08:49  
ENG  
30/09/2020  
6

## Project-based Matrix Authorization Strategy

---

- Project-based Matrix authorization strategy is an extension to “Matrix-based security”, where you can setup ACL to individual projects.
- This means that you can allow a specific user to access only certain projects.
- The moment you click on the “Project-based Matrix Authorization Strategy” radio-button, it will display the following table right below it where you can add users and assign them individual privileges by clicking on the check-box.

# Project-based Matrix Authorization Strategy

## Authorization

- Anyone can do anything
- Legacy mode
- Logged-in users can do anything
- Matrix-based security
- Project-based Matrix Authorization Strategy

User/group	Overall								
	Administer	Configure	UpdateCenter	Read	RunScripts	Upload	Plugins	Create	Delete
Anonymous	<input type="checkbox"/>								

User/group to add:  Add

## Project-based Matrix Authorization Privileges

---

- There are 7 privilege groups.
- Within each privilege groups, there are individual privileges that you can assign to a specific user or group.

# Project-based Matrix Authorization Privileges

---

- Privilege Group 1: Overall
  - Administer – This permission grants the ability to make system-wide configuration changes, as well as perform highly sensitive operations that amounts to full local system access (within the scope granted by the underlying OS.)
  - ConfigureUpdateCenter – This permission allows a user to configure update sites and proxy settings
  - Read – The read permission is necessary for viewing almost all pages of Jenkins. This permission is useful when you don't want unauthenticated users to see Jenkins pages: revoke this permission from the anonymous user, then add “authenticated” pseudo-user and grant the read access.
  - RunScripts – Required for running scripts inside the Jenkins process, for example via the Groovy console or Groovy CLI command.
  - UploadPlugin – This permission allows a user to upload arbitrary plugins

# Project-based Matrix Authorization Privileges

---

- Privilege Group 2: Credentials
  - Create – The create permission is necessary to add credentials to a credentials provider.
  - Delete – The delete permission is necessary to remove credentials stored in a credentials provider.
  - Manage Domains – The manage domains permission is necessary to add/remove/configure the credential domains of a credentials provider (where the credentials provider supports multiple credential domains).
  - Update – The update permission is necessary to modify credentials in a credentials provider.
  - View – The view permission is necessary to view the credentials stored in a credentials provider.

# Project-based Matrix Authorization Privileges

---

- Privilege Group 3: Agent
  - Build – This permission allows users to run jobs as them on agents.
  - Configure – This permission allows users to configure agents.
  - Connect – This permission allows users to connect agents or mark agents as online.
  - Create – This permission allows users to create agents.
  - Delete – This permission allows users to delete existing agents.
  - Disconnect – This permission allows users to disconnect agents or mark agents as temporarily offline.

# Project-based Matrix Authorization Privileges

---

- Privilege Group 4: Job
  - Build – This permission grants the ability to start a new build.
  - Cancel – This permission grants the ability to cancel a scheduled, or abort a running, build.
  - Configure – Change the configuration of a job.
  - Create – Create a new job.
  - Delete – Delete a job.
  - Discover – This permission grants discover access to jobs. Lower than read permissions, it allows you to redirect anonymous users to the login page when they try to access a job url. Without it they would get a 404 error and wouldn't be able to discover project names.
  - Read – See a job. (You may deny this permission but allow Discover to force an anonymous user to log in to see the job.)
  - Workspace – This permission grants the ability to retrieve the contents of a workspace Jenkins checked out for performing builds. If you don't want a user to access files in the workspace (e.g. source code checked out from SCM or intermediate build results) through the workspace browser, you can revoke this permission.

# Project-based Matrix Authorization Privileges

---

- Privilege Group 5: Run
  - Delete – This permission allows users to manually delete specific builds from the build history.
  - Replay – Ability to perform a new Pipeline build with an edited script.
  - Update – This permission allows users to update description and other properties of a build, for example to leave notes about the cause of a build failure

# Project-based Matrix Authorization Privileges

---

- Privilege Group 6: View
  - Configure – This permission allows users to change the configuration of views.
  - Create – This permission allows users to create new views.
  - Delete – This permission allows users to delete existing views.
  - Read – This permission allows users to see views (implied by generic read access).
- Privilege Group 7: SCM
  - Tag – This permission allows users to create a new tag in the source code repository for a given build

# Role Based Strategy

Configure Global Security [Jenkins] × Role-based Authorization Strategy [localhost:7070/configureSecurity/] +

Insert title here Empire New Tab How to use Assert... Browser Automation Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

Dashboard > Configure Global Security

LDAP

None

**Authorization**

Anyone can do anything  
 Legacy mode  
 Logged-in users can do anything  
 Matrix-based security  
 Project-based Matrix Authorization Strategy  
 Role-Based Strategy

**Markup Formatter**

Markup Formatter

Plain text

Treats all input as plain text. HTML unsafe characters like < and & are escaped to their respective character entities.

**Agents**

TCP port for inbound agents

Fixed :   Random  Disable

Agent protocols...

**CSRF Protection**

Save Apply



84 11/01/2022 08:41 ENG IN 19

# Role Based Strategy

Manage Jenkins (Jenkins) x Role-based Authorization Strat... x + localhost:7070/manage Insert title here Empire New Tab How to use Assert... Browser Automation... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

Dashboard >

when the machine boots.

**Security**

 **Configure Global Security**  
Secure Jenkins; define who is allowed to access/use the system.

 **Manage and Assign Roles**  
Handle permissions by creating roles and assigning them to users/groups

 **Manage Credentials**  
Configure credentials

 **Configure Credential Providers**  
Configure the credential providers and types

 **Manage Users**  
Create/delete/modify users that can log in to this Jenkins

 **In-process Script Approval**  
Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions.

**Status Information**

 **System Information**  
Displays various environmental information to assist trouble-shooting.

 **System Log**  
System log captures output from java.util.logging output related to Jenkins.

 **Load Statistics**  
Check your resource utilization and see if you need more computers for your builds.

 **About Jenkins**  
See the version and license information.

Windows Search File Explorer Microsoft Edge Google Chrome Microsoft Edge Microsoft Word Microsoft Excel Microsoft Powerpoint 52 8 08:41 11/01/2022 ENG IN

# Role Based Strategy

Day1 project

localhost:8989/role-strategy/manage-roles

Manage and Assign Roles

Role	Overall				Credentials				Agent											
	Administer	Read	Create	Delete	ManageDomains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Provision	Build	Cancel	Configure	Create		
<input checked="" type="checkbox"/> admin	<input checked="" type="checkbox"/>																			
<input checked="" type="checkbox"/> developer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/> qa	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<input checked="" type="checkbox"/> test	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Role to add

Open Blue Ocean

Add

Item roles

Role	Pattern	Credentials				Job												Run		SCM		Loc	
		Create	Delete	ManageDomains	Update	View	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Workspace	Delete	Replay	Update	Tag	Res			
<input checked="" type="checkbox"/> test	"day1.*"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																	

Role to add

Pattern

# Role Based Strategy

Limited job access

The screenshot shows the Jenkins dashboard at [localhost:8989](http://localhost:8989). A blue arrow points from the text "Limited job access" to the "Name" column header of the main table, which is sorted in descending order.

**Jenkins Dashboard**

Dashboard

People, Build History, Project Relationship, Check File Fingerprint, My Views, Open Blue Ocean

Build Queue: No builds in the queue.

Build Executor Status: master (1 Idle), 2 Idle

All

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
●	●	day1fsandroidproject	20 hr - #4	20 hr - #3	1 min 7 sec	
●	●	day1fsdockerproject	8 hr 0 min - #21	22 hr - #19	53 sec	
●	●	day1fstest	1 day 0 hr - #20	1 day 3 hr - #7	25 sec	

Icon: S M L

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

# Jenkins Build log Regex

---

- Way 1 –
  - Put following lines as part of Default Content text box
  - \${BUILD\_LOG, maxLines=9999, escapeHtml=false}
  - this works for free style projects
- Way 2 – If you want send only specific line using regex
  - \${BUILD\_LOG\_REGEX, regex="\b(Buildfile|BUILD)\b", linesAfter=1}
  - \${BUILD\_LOG\_REGEX, regex="^.+?BUILD FAILED.+?\$", linesBefore=0, linesAfter=10, maxMatches=5, showTruncatedLines=false, escapeHtml=true}
  - \${BUILD\_LOG\_REGEX, regex="\b(FC0)\b", linesAfter=1}
  - \${BUILD\_LOG\_REGEX, regex="^.+?BUILD FAILED.+?\$", linesBefore=0, linesAfter=10, maxMatches=5, showTruncatedLines=false, escapeHtml=true}
  - \${BUILD\_LOG\_REGEX, regex="^.+?FC0.+?\$", maxMatches=95, showTruncatedLines=false, escapeHtml=false, matchedLineHtmlStyle=true}
- This will show exactly which is expected lines as part of the regEx.

# Jenkins Build log Regex

---

- List of other variables as part of email body
- \${PROJECT\_URL}/ws/
- \${BUILD\_URL}/artifact/
- \${BUILD\_LOG, maxLines=1000}
- More ref
- Content Token Reference
- In string arguments, escape “, \, and line terminators (\n or \r\n) with a \, e.g. arg1="\"quoted\""; arg2="c:\\path"; and arg3="one\\ntwo". The brackets may be omitted if there are no arguments.
- New Lines – Use <br> or “\”

# Jenkins Build log regex

day1fstest ➔

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Default Content ?

```
 ${BUILD_LOG, maxLines=1, escapeHtml=false}  
 ${BUILD_LOG_REGEX, regex="\b(FC0)\b", linesAfter=1}  
 ${BUILD_LOG_REGEX, regex="^.+?BUILD FAILED.*$?", linesBefore=0, linesAfter=10, maxMatches=5,  
 ${PROJECT_URL}/ws/  
 ${BUILD_URL}/artifact/  
 ${BUILD_LOG, maxLines=10}
```

Attachments ?

Can use wildcards like 'module/dist/\*\*/\*.zip'. See the [@includes of Ant fileset](#) for the exact format. The base directory is [the workspace](#).

Attach Build Log ?

Do Not Attach Build Log ▾

Content Token Reference ?

Advanced Settings...

# Jenkins Build log regex

A screenshot of an email client interface, likely Gmail, displaying an email from 'parameswaribala@gmail.com' to 'me, vebconsulting2021'. The subject of the email is 'day1fstest - Build # 23 - Successful!'. The email body contains the Jenkins build log, which includes several regular expression definitions for log parsing, URLs for the build workspace and artifacts, and standard INFO log entries. The log concludes with a WARNING message about a non-existent profile and information about the trigger for the email.

Search mail ② ⚙️ ⋮

day1fstest - Build # 23 - Successful! Inbox ×

parameswaribala@gmail.com  
to me, vebconsulting2021 ...  
[...truncated 5.09 KB...]

8:34 PM (3 minutes ago) ☆ ↶ ⋮

```
$[BUILD_LOG_REGEX, regex="\b(FC0)\b", linesAfter=1]
$[BUILD_LOG_REGEX, regex="^.?BUILD FAILED.*?$", linesBefore=0, linesAfter=10, maxMatches=5,
http://localhost:8989/job/day1fstest/ws/
http://localhost:8989/job/day1fstest/23//artifact/
[...truncated 4.43 KB...]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 12.822 s
[INFO] Finished at: 2021-06-08T20:34:39+05:30
[INFO] -----
[WARNING] The requested profile "test" could not be activated because it does not exist.
Email was triggered for: Always
Sending email for trigger: Always
```

↶ Reply ➡ Forward

# Android Jenkins Project

Administrator: Command Prompt - sdkmanager --update

Directory of C:\Users\Balasubramaniam\AppData\Local\Android\Sdk\tools\bin



```
05/08/2018  02:26 PM    <DIR>      .
05/08/2018  02:26 PM    <DIR>      ..
05/08/2018  02:26 PM            6,742 apk analyzer
05/08/2018  02:26 PM            2,227 archquery.bat
05/08/2018  02:26 PM            3,035 avdmanager.bat
05/08/2018  02:26 PM            2,215 jobb.bat
05/08/2018  02:26 PM            3,839 lint.bat
05/08/2018  02:26 PM            2,053 monkeyrunner.bat
05/08/2018  02:26 PM            3,042 sdkmanager.bat
05/08/2018  02:26 PM            2,189 uiautomatorviewer.bat
               8 File(s)        25,342 bytes
               2 Dir(s)   26,846,961,664 bytes free
```

C:\Users\Balasubramaniam\AppData\Local\Android\Sdk\tools\bin> sdkmanager --update  
[ ] 3% Fetch remote repository...

Android home

# Android Jenkins Project

## Android home

The screenshot shows the Jenkins configuration page for an 'Android' job. The URL in the browser is `http://localhost:8989/configure`. The page has a header with various tabs and links. On the left, there's a sidebar with 'Dashboard' and 'configuration' selected. The main content area is titled 'Android' and contains the following sections:

- SCM Polling**:
  - Max # of concurrent polling: 10
- Android**:
  - Android SDK root: C:\Users\Balasubramaniam\AppData\Local\Android\Sdk
  - Enter the path to the root of an Android SDK installation
  - Automatically install Android components when required (checked)
  - Keep emulators in the job workspace, in the .android subdirectory, to isolate them as much as possible (unchecked)
- Cloud**:

The cloud configuration has moved to a [separate configuration page](#).

At the bottom, there are 'Save' and 'Apply' buttons.

# Angular

---

- cd insuranceapp & npm install --save --legacy-peer-deps

# Before Jenkins Pipeline

---

Over the years, there have been multiple Jenkins pipeline releases including, **Jenkins Build flow**, **Jenkins Build Pipeline plugin**, **Jenkins Workflow**, etc.



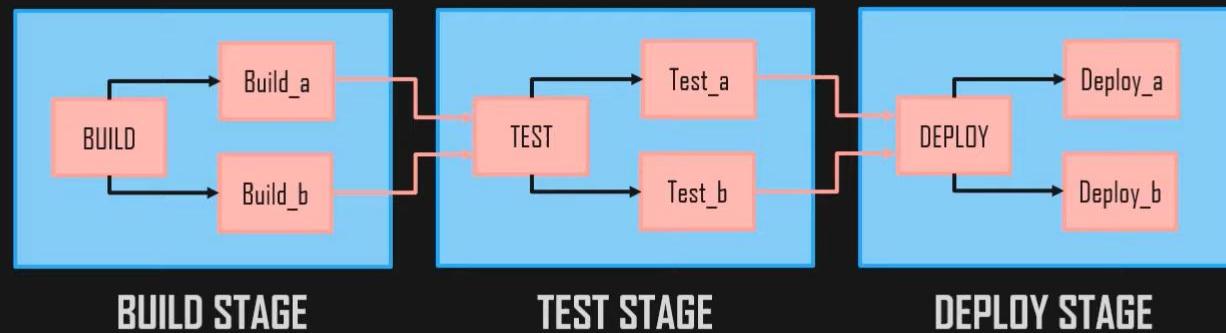
*What are the key features of these plugins?*

- Represent multiple Jenkins jobs as one pipeline
- What do these pipelines do?

These pipelines are a collection of Jenkins jobs which trigger each other in a specified sequence

# Jenkins Pipeline

---



# Jenkins Pipeline

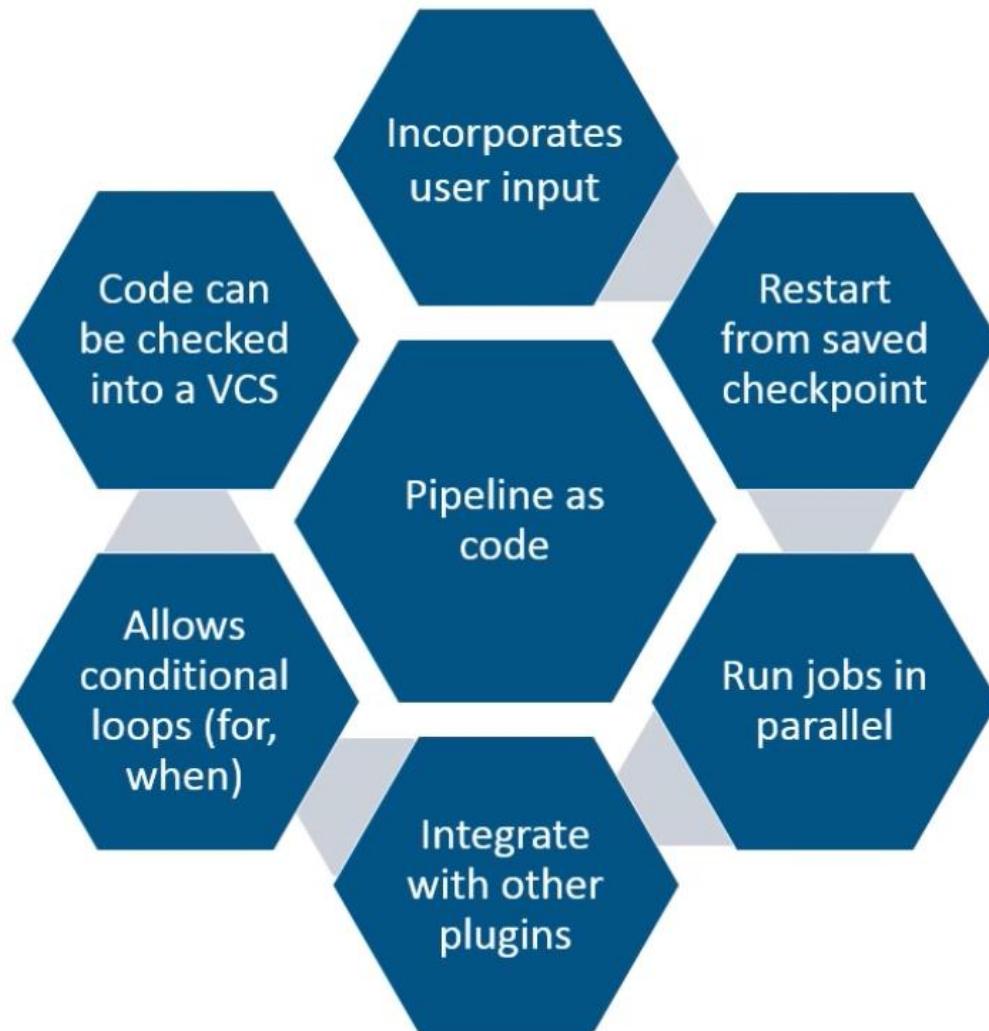
---

- Jenkins pipeline is a single platform that runs the entire *pipeline as code*
- All the standard jobs defined by Jenkins are manually written in one script and they can be stored in a VCS
- Instead of building several jobs for each phase, you can now code the entire workflow and put it in a Jenkinsfile



# Jenkins Pipeline key Features

---



# Jenkins File

---

A text file that stores the pipeline as code

It can be checked into a SCM on your local system

Enables the developers to access, edit and check the code at all times

It is written using the Groovy DSL

Written based on two syntaxes

# Jenkins Pipeline

```
// Example of Jenkins pipeline script

pipeline {
    stages {
        stage("Build") {
            steps {
                // Just print a Hello, Pipeline to the console
                echo "Hello, Pipeline!"
                // Compile a Java file. This requires JDKconfiguration from Jenkins
                javac HelloWorld.java
                // Execute the compiled Java binary called HelloWorld. This requires JDK configuration from Jenkins
                java HelloWorld
                // Executes the Apache Maven commands, clean then package. This requires Apache Maven configuration from
                Jenkins
                mvn clean package ./HelloPackage
                // List the files in current directory path by executing a default shell command
                sh "ls -ltr"
            }
        }
        // And next stages if you want to define further...
    } // End of stages
} // End of pipeline
```

then we're using me one as the build

# Jenkins Pipeline

---

- Jenkins Pipeline (or simply "Pipeline" with a capital "P") is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.
- A continuous delivery (CD) pipeline is an automated expression of your process for getting software from version control right through to your users and customers.
- Every change to your software (committed in source control) goes through a complex process on its way to being released.
- This process involves building the software in a reliable and repeatable manner, as well as progressing the built software (called a "build") through multiple stages of testing and deployment.
- Pipeline provides an extensible set of tools for modeling simple-to-complex delivery pipelines "as code" via the Pipeline domain-specific language (DSL) syntax.

# Jenkins Pipeline

---

- The definition of a Jenkins Pipeline is written into a text file (called a `Jenkinsfile`) which in turn can be committed to a project's source control repository.
- This is the foundation of "Pipeline-as-code"; treating the CD pipeline a part of the application to be versioned and reviewed like any other code.

# Jenkins Pipeline

---

- Creating a Jenkinsfile and committing it to source control provides a number of immediate benefits:
  - Automatically creates a Pipeline build process for all branches and pull requests.
  - Code review/iteration on the Pipeline (along with the remaining source code).
  - Audit trail for the Pipeline.
  - Single source of truth for the Pipeline, which can be viewed and edited by multiple members of the project.

# Declarative versus Scripted Pipeline syntax

---

- Declarative:
  - Declarative pipeline syntax offers an easy way to create pipelines.
  - It contains a predefined hierarchy to create Jenkins pipelines.
  - It gives you the ability to control all aspects of a pipeline execution in a simple, straight-forward manner.

# Declarative versus Scripted Pipeline syntax

---

- Scripted:
  - Scripted Jenkins pipeline runs on the Jenkins master with the help of a lightweight executor.
  - It uses very few resources to translate the pipeline into atomic commands.
  - Both declarative and scripted syntax are different from each other and are defined totally differently.

# Declarative versus Scripted Pipeline syntax

## SCRIPTED PIPELINE

- ✓ Strict & Traditional Syntax
- ✓ Code is written on a Jenkins UI Instance
- ✓ Code is defined within a "Node Block"

## DECLARATIVE PIPELINE

- ✓ Simpler Groovy Syntax
- ✓ Code is written into a file & checked into SCM
- ✓ Code is defined within a "Pipeline Block"

jenkins file from a virtual constant

## Declarative versus Scripted Pipeline syntax

---

- A Jenkinsfile can be written using two types of syntax - Declarative and Scripted.
- Declarative and Scripted Pipelines are constructed fundamentally differently.
- Declarative Pipeline is a more recent feature of Jenkins Pipeline which:
  - provides richer syntactical features over Scripted Pipeline syntax, and
  - is designed to make writing and reading Pipeline code easier.

## Declarative Syntax

---

- A valid Declarative pipeline must be defined with the “pipeline” sentence and include the next required sections:
  - agent
  - stages
  - stage
  - steps

## Declarative Syntax

---

- Also, these are the available directives:
  - environment (Defined at stage or pipeline level)
  - input (Defined at stage level)
  - options (Defined at stage or pipeline level)
  - parallel
  - parameters
  - post
  - script
  - tools
  - triggers
  - when

## Declarative Syntax

---

- Agents should be labeled so they can be easily identified from each other. For example, nodes can be labeled by their platform (Linux, Windows, etc), by their versions or by their location, among others.
- The “agent” section configures on which nodes the pipeline can be run. Specifying “agent any” means that Jenkins will run the job on any of the available nodes.

## Why Pipeline?

---

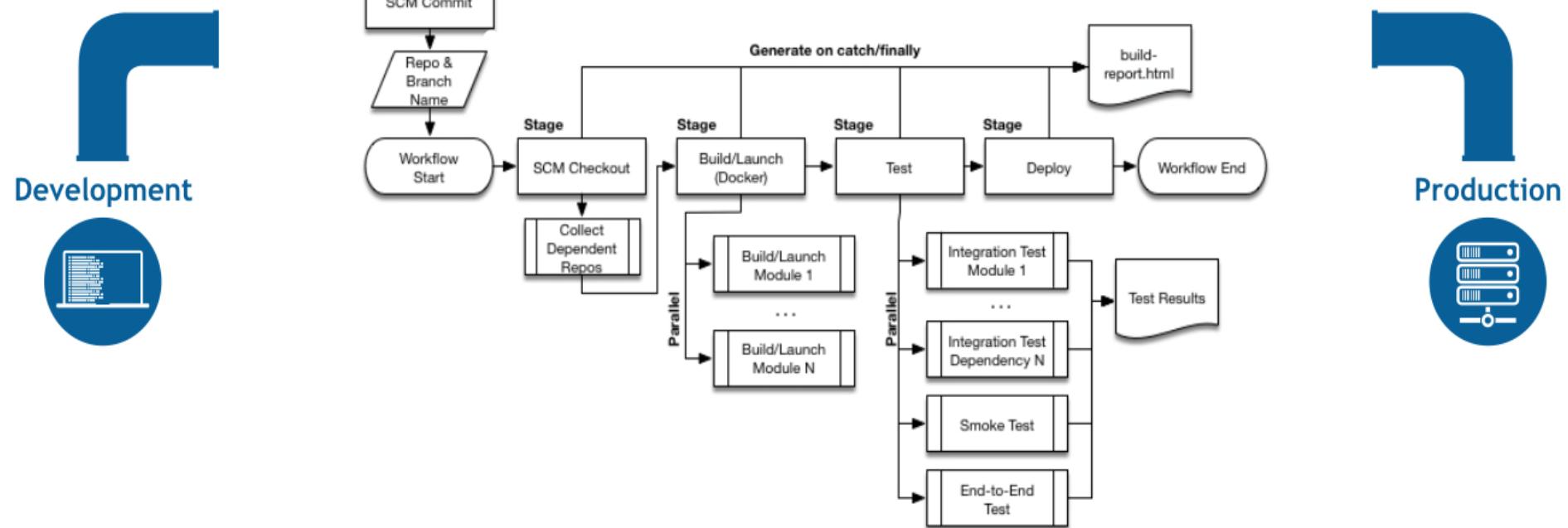
- Jenkins is, fundamentally, an automation engine which supports a number of automation patterns.
- Pipeline adds a powerful set of automation tools onto Jenkins, supporting use cases that span from simple continuous integration to comprehensive CD pipelines.
- By modeling a series of related tasks, users can take advantage of the many features of Pipeline:

# Why Pipeline?

---

- **Code:** Pipelines are implemented in code and typically checked into source control, giving teams the ability to edit, review, and iterate upon their delivery pipeline.
- **Durable:** Pipelines can survive both planned and unplanned restarts of the Jenkins controller.
- **Pausable:** Pipelines can optionally stop and wait for human input or approval before continuing the Pipeline run.
- **Versatile:** Pipelines support complex real-world CD requirements, including the ability to fork/join, loop, and perform work in parallel.
- **Extensible:** The Pipeline plugin supports custom extensions to its DSL footnote:dsl:[] and multiple options for integration with other plugins.

# Why Pipeline?



# Pipeline Concepts

---

**Agent:** instructs Jenkins to allocate an executor for the builds. It is defined for an entire pipeline or a specific stage.

It has the following parameters:

- *Any*: Runs pipeline/ stage on any available agent
- *None*: applied at the root of the pipeline, it indicates that there is no global agent for the entire pipeline & each stage must specify its own agent
- *Label*: Executes the pipeline/stage on the labelled agent.
- *Docker*: Uses docker container as an execution environment for the pipeline or a specific stage.

```
pipeline {  
    agent {  
        docker {  
            image 'ubuntu'  
        }  
    }  
}
```

# Pipeline Concepts

---

**Stages:** It contains all the work, each stage performs a specific task.

```
pipeline {  
    agent any  
    stages {  
        stage ('Build') {  
            ...  
        }  
        stage ('Test') {  
            ...  
        }  
        stage ('QA') {  
            ...  
        }  
        stage ('Deploy') {  
            ...  
        }  
        stage ('Monitor') {  
            ...  
        }  
    }  
}
```

**Steps:** steps are carried out in sequence to execute a stage

```
pipeline {  
    agent any  
    stages {  
        stage ('Build') {  
            steps {  
                echo 'Running build phase...'  
            }  
        }  
    }  
}
```

# Why Pipeline?

## Declarative Pipeline fundamentals

In Declarative Pipeline syntax, the `pipeline` block defines all the work done throughout your entire Pipeline.

*Jenkinsfile (Declarative Pipeline)*

```
pipeline {
    agent any ①
    stages {
        stage('Build') { ②
            steps {
                // ③
            }
        }
        stage('Test') { ④
            steps {
                // ⑤
            }
        }
        stage('Deploy') { ⑥
            steps {
                // ⑦
            }
        }
    }
}
```

# Why Pipeline?

---

- 1 Execute this Pipeline or any of its stages, on any available agent.
- 2 Defines the "Build" stage.
- 3 Perform some steps related to the "Build" stage.
- 4 Defines the "Test" stage.
- 5 Perform some steps related to the "Test" stage.
- 6 Defines the "Deploy" stage.
- 7 Perform some steps related to the "Deploy" stage.

# Why Pipeline?

---

- In Scripted Pipeline syntax, one or more node blocks do the core work throughout the entire Pipeline.
- Although this is not a mandatory requirement of Scripted Pipeline syntax, confining your Pipeline's work inside of a node block does two things:
- Schedules the steps contained within the block to run by adding an item to the Jenkins queue.
- As soon as an executor is free on a node, the steps will run.
- Creates a workspace (a directory specific to that particular Pipeline) where work can be done on files checked out from source control.
- Caution: Depending on your Jenkins configuration, some workspaces may not get automatically cleaned up after a period of inactivity.

# Why Pipeline?

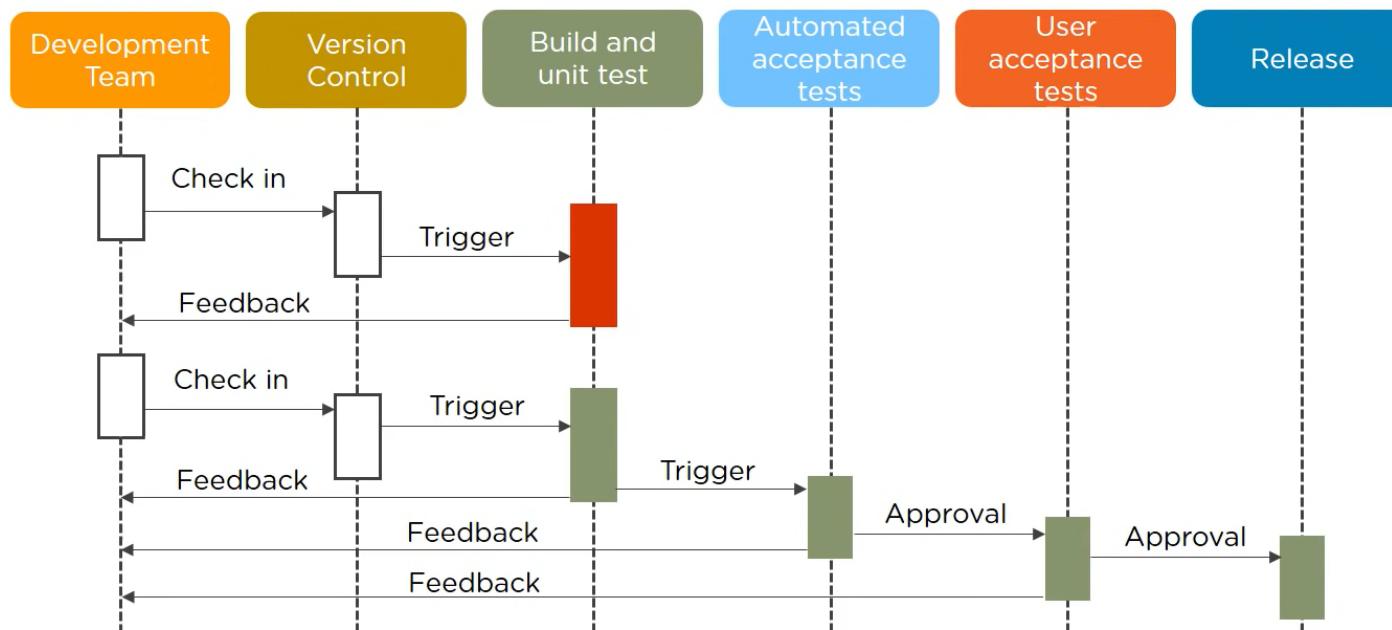
*Jenkinsfile (Scripted Pipeline)*

```
node { ①
    stage('Build') { ②
        // ③
    }
    stage('Test') { ④
        // ⑤
    }
    stage('Deploy') { ⑥
        // ⑦
    }
}
```

- ① Execute this Pipeline or any of its stages, on any available agent.
- ② Defines the "Build" stage. `stage` blocks are optional in Scripted Pipeline syntax. However, implementing `stage` blocks in a Scripted Pipeline provides clearer visualization of each `stage's subset of tasks/steps in the Jenkins UI.
- ③ Perform some steps related to the "Build" stage.
- ④ Defines the "Test" stage.
- ⑤ Perform some steps related to the "Test" stage.
- ⑥ Defines the "Deploy" stage.
- ⑦ Perform some steps related to the "Deploy" stage.

# Pipeline

Pipeline is a suite of Jenkins features, installed as plugins, which enable implementing and integrating continuous delivery pipelines into Jenkins.



- Every check-in triggers pipeline execution
- Feedback is sent to the team in every step
- Minimize execution time

## Open Blue Ocean

---

- Blue Ocean is created on the basis of the user's Jenkins experience.
- It is mainly designed for the pipeline process, but it can also perform freestyle jobs.
- Blue Ocean reduces the disorder and chaos caused by the Jenkins and also increases the clarity for each team member.
- Blue Ocean is very useful for a fully skilled user who knows Jenkins features very well.

# Main features of Blue Ocean

---

- **Supports freestyle jobs:** Although it is designed for the pipeline process, it also supports freestyle jobs. Even you can run a single job on it.
- **Personalization:** Blue Ocean performs in a personalized way. Each member of the team can easily visualize the execution and modification made in the builds.
- **Pinpoint precision:** Whenever any problem arises or any interference is required in the pipeline process, and Blue Ocean points out the exact location where the bug has arisen and where you can make attention in the pipeline.
- **Pipeline editor:** Blue Ocean makes it easy to create a pipeline by giving guidance to the user via some instructions and visual representation.
- **Open source:** The blue ocean is open source; you are able to make changes as per your requirements.
- **Free of cost:** The blue ocean is totally free of cost. It comes as a plugin in the Jenkins, and you can download it from the manage Jenkins section.

# Install Blue Ocean Plugins

localhost:8081/updateCenter/

Apps YouTube Maps Manage Jenkins

Jenkins > Update Center

Display URL for Blue Ocean	Success
Events API for Blue Ocean	Failure - <a href="#">Details</a>
Blue Ocean Pipeline Editor	Installing
i18n for Blue Ocean	Pending
Autofavorite for Blue Ocean	Pending
Blue Ocean	Pending
Loading plugin extensions	Pending
Pipeline SCM API for Blue Ocean	Pending
Pipeline implementation for Blue Ocean	Pending
GitHub Pipeline for Blue Ocean	Pending
Git Pipeline for Blue Ocean	Pending
Config API for Blue Ocean	Pending
Bitbucket Pipeline for Blue Ocean	Pending
Personalization for Blue Ocean	Pending
Blue Ocean	Pending
Restarting Jenkins	Pending

[Go back to the top page](#)  
(you can start using the installed plugins right away)

Type here to search

1:23 PM  
ENG IN  
11-Dec-20  
6

# Blue Ocean Plugins

localhost:8081

Apps YouTube Maps Manage Jenkins

## Jenkins

search ?

GovindTiwari log out

New Item People Build History Manage Jenkins My Views Open Blue Ocean Lockable Resources New View

All My\_View +

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
		addition	2 mo 4 days - #6	N/A	1.2 sec	

Icon: S M L Legend Atom feed for all Atom feed for failures Atom feed for just latest builds

No builds in the queue.

localhost:8081/blue/organizations/jenkins

Type here to search

1:33 PM 11-Dec-20 ENG IN 6

# Blue Ocean Pipeline

Jenkins download and deployment | Jenkins | eswaribala/rpsonmobilejenkinsja | eswaribala/rpsonmobilejenkinsja | Create a Pipeline in Blue Ocean | + | - | X

localhost:7070/blue/organizations/jenkins/create-pipeline

Insert title here Empire New Tab How to use Asserti... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 » Reading list

Jenkins Pipelines Administration Logout

Create Pipeline Classic Item Creation

Where do you store your code?

- Bitbucket Cloud
- Bitbucket Server
- GitHub
- GitHub Enterprise
- Git

1.25.2 · Core 2.319.1 · a7f3cba · 20th November 2021 08:03 AM

DevOps\_Course O....pdf Verified jenkins.war Verified Show all

ENG IN 22:16 10/01/2022 18

# Blue Ocean Pipeline

Jenkins download and deployment | Jenkins | eswaribala/rpsonmobilejenkinsja | Edit personal access token | Create a Pipeline in Blue Ocean | +

localhost:7070/blue/organizations/jenkins/create-pipeline

Insert title here Empire New Tab How to use Asserti... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 » Reading list

Create Pipeline Classic Item Creation

Where do you store your code?

- Bitbucket Cloud
- Bitbucket Server
- GitHub
- GitHub Enterprise
- Git

Connect to GitHub

Jenkins needs an access token to authorize itself with GitHub.  
[Create an access token here.](#)

Which organization does the repository belong to?

VebConsultinng

DevOps\_Course\_O.pdf Verified jenkins.war Verified Show all

22:18 10/01/2022 ENG IN 18

## Blue Ocean Pipeline

---

- In Which organization does the repository belong to?, click your GitHub account (where you forked the repository above).
- In Choose a repository, click your forked repository creating-a-pipeline-in-blue-ocean.
- Click Create Pipeline.
- Blue Ocean detects that there is no Jenkinsfile at the root level of the repository's master branch and proceed to help you create one. (**Therefore, you'll need to click another Create Pipeline at the end of the page to proceed.**)

# Blue Ocean Pipeline

The screenshot shows the Jenkins Blue Ocean Pipeline creation interface. The top navigation bar includes tabs for Jenkins download and deployment, Jenkins, eswaribala/rpsonmobilejenkinsjan2022projects, Edit personal access token, Create a Pipeline in Blue Ocean, and several other Jenkins instances. The main content area displays a pipeline flow:

- Step 1: Choose a repository** (Green circle with checkmark). It shows 381 repositories and a search bar with the query "rpsonmobilejenkinsjan2022projects". A blue button labeled "Create Pipeline" is visible.
- Step 2: Naming conflict** (Blue circle). It indicates that the name "onbileblueoceanpipeline" is available and provides a "Save" button.
- Step 3: Completed** (Grey circle).

At the bottom, the footer shows the version "1.25.2 · Core 2.319.1 · a7f3cba · 20th November 2021 08:03 AM". The taskbar at the bottom of the browser window shows files like "DevOps\_Course O....pdf" and "jenkins.war", along with various system icons and the date "10/01/2022".

# Blue Ocean Pipeline

Jenkins download and deployment | Jenkins | eswaribala/rpsonmobilejenkinsjan2022projects | Edit personal access token | Create a Pipeline in Blue Ocean | +

localhost:7070/blue/organizations/jenkins/pipeline-editor/rpsonmobilejenkinsjan2022projects/

Insert title here Empire New Tab How to use Assert... Browser Automation Freelancer-dev-810... Courses nodejs - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

## Jenkins

Pipelines Administration Logout

### rpsonmobilejenkinsjan2022projects

Cancel Save

Start

Pipeline Settings

Agent

any

Environment

Name Value

DevOps\_Course O....pdf Verified jenkins.war Verified Show all

ENG IN 22:21 18 10/01/2022

# Blue Ocean Pipeline

Screenshot of a Jenkins pipeline editor interface titled "onbileblueoceanpipeline". The pipeline consists of a single step named "build". The pipeline starts at a "Start" node, which connects to the "build" step. From the "build" step, there is a feedback loop back to the "Start" node. The pipeline is currently in edit mode, indicated by the "Cancel" and "Save" buttons in the top right corner.

The right panel shows a message box with the text "Rocking with Blue Ocean".

The browser tab bar shows multiple tabs, including "Jenkins download and deployment", "Jenkins", "eswaribala/rpsonmobilejenkinsja...", "Edit personal access token", "Create a Pipeline in Blue Ocean", and others.

The system tray at the bottom shows various icons and status information, including "ENG IN", "22:24", "10/01/2022", and a battery level of 18%.

# Blue Ocean Pipeline

The screenshot shows a Jenkins pipeline editor interface. At the top, there are several tabs: 'Jenkins download and deployment', 'Jenkins', 'eswaribala/rpsonmobilejenkinsja', 'Edit personal access token', 'Create a Pipeline in Blue Ocean', and a new tab indicator. Below the tabs, the URL is `localhost:7070/blue/organizations/jenkins/pipeline-editor/onbileblueoceanpipeline/`. The main navigation bar includes 'Insert title here', 'Empire', 'New Tab', 'How to use Asserti...', 'Browser Automatio...', 'Freelancer-dev-810...', 'Courses', 'node.js - How can I...', 'New Tab', 'Airtel 4G Hotspot', 'nt8F83', and 'Reading list'. The pipeline name 'onbileblueoceanpipeline' is displayed on the left. On the right, there are 'Cancel' and 'Save' buttons. A 'Logout' button is also visible. A central modal window titled 'Save Pipeline' contains the message: 'Saving the pipeline will commit a Jenkinsfile to the repository.' It has a 'Description' section with a text input field containing 'What changed?'. Below it are two radio buttons: 'Commit to master' (selected) and 'Commit to new branch'. A dropdown menu next to 'Commit to master' shows 'master'. At the bottom of the modal are 'Save & run' and 'Cancel' buttons. To the right of the modal, there is a message box with a 'build / Print Message' link and a 'Message\*' input field containing 'Rocking with Blue Ocean'. The bottom of the screen shows a taskbar with files like 'DevOps\_Course O....pdf' and 'jenkins.war', system icons, and a status bar indicating 'ENG IN' and the date '10/01/2022'.

# Blue Ocean Pipeline

Screenshot of a web browser showing the Jenkins Blue Ocean Pipeline interface.

The browser tabs are:

- Jenkins download and deployment
- Jenkins / onbileblueoceanpipeline
- eswaribala/rpsonmobilejenkinsja
- Edit personal access token
- Create a Pipeline in Blue Ocean

The address bar shows: localhost:7070/blue/organizations/jenkins/onbileblueoceanpipeline/activity

The Jenkins Blue Ocean Pipeline interface has the following structure:

- Header:** Jenkins logo, Pipeline name (onbileblueoceanpipeline), and navigation tabs: Pipelines, Administration, Activity (selected), Branches, Pull Requests, Logout.
- Activity Table:** A table showing pipeline activity details.

Status	Run	Commit	Branch	Message	Duration	Completed
○	1	-	master	Branch indexing	7s	-
- Footer:** Version information (1.25.2 · Core 2.319.1 · a7f3cba · 20th November 2021 08:03 AM) and system tray.

The system tray shows:

- DevOps\_Course O...pdf (Verified)
- jenkins.war (Verified)
- Show all
- ENG IN
- 22:25
- 10/01/2022
- 18

The taskbar icons include: File Explorer, Task View, Search, Start, Microsoft Edge, Google Chrome, File Explorer (48 items), Microsoft Word, Microsoft Excel, Microsoft Powerpoint, Microsoft Teams (8 notifications), and Microsoft Edge.

# Blue Ocean Pipeline (Run)

Jenkins download and deployment x jenkins / onbileblueoceanpipeline x eswaribala/rpsonmobilejenkinsja x | Edit personal access token x | Create a Pipeline in Blue Ocean x +

localhost:7070/blue/organizations/jenkins/onbileblueoceanpipeline/detail/master/1/pipeline

Insert title here Empire New Tab How to use Assert... Browser Automation Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

✓ onbileblueoceanpipeline 1

Pipeline Changes Tests Artifacts Logout

Branch: master 13s No changes  
Commit: 2567298 a few seconds ago Branch indexing

Start build End

build - <1s

Restart build

> Check out from version control 5s

> Rocking with Blue Ocean – Print Message <1s

DevOps\_Course O...pdf Verified jenkins.war Verified Show all

22:25 10/01/2022 ENG IN 18

# Blue Ocean Pipeline (Run)

Jenkins download and deployment x jenkins / onbileblueoceanpipeline x eswaribala/rpsonmobilejenkinsx | Edit personal access token x Create a Pipeline in Blue Ocean x +

localhost:7070/blue/organizations/jenkins/onbileblueoceanpipeline/detail/master/1/pipeline

Insert title here Empire New Tab How to use Asserti... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 » Reading list

✓ onbileblueoceanpipeline 1

Branch: master 13s No changes  
Commit: 2567298 a minute ago Branch indexing

Pipeline Changes Tests Artifacts

Start build End

build - <1s

Restart build 5s

> Check out from version control

Rocking with Blue Ocean – Print Message <1s

1 Rocking with Blue Ocean

DevOps\_Course O....pdf Verified jenkins.war Verified Show all

ENG IN 22:26 10/01/2022 18

# Kubernetes Cluster (Freestyle restrict node)

Dashboard [Jenkins] | eswaribala/rpsonmobilejenkinsja | eswaribala/rpsonmobilejenkinsja | testnode Config [Jenkins] | How to Configure Jenkins Master | +

localhost:6060/job/testnode/configure

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

Reading list

Dashboard > testnode >

**General** Source Code Management Build Triggers Build Environment Build Post-build Actions

[Plain text] Preview

Commit agent's Docker container ?  
 Run the build inside Docker containers  
 Define a Docker template  
 Discard old builds ?  
 This project is parameterized ?  
 Throttle builds ?  
 Disable this project ?  
 Execute concurrent builds if necessary ?  
 Restrict where this project can be run ?

**Label Expression** ?

winnode

Advanced...

**Source Code Management**

None

Save Apply

Build after other projects are built

DevOps\_Course O...pdf Verified jenkins.war Verified

Show all

# Kubernetes Cluster

Dashboard [Jenkins] | eswaribala/rpsonmobilejenkinsja | eswaribala/rpsonmobilejenkinsja | testnode #1 Console [Jenkins] | How to Configure Jenkins Master | + | localhost:6060/job/testnode/1/console | Insert title here | Empire | New Tab | How to use Assert... | Browser Automatio... | Freelancer-dev-810... | Courses | node.js - How can I... | New Tab | Airtel 4G Hotspot | nt8F83 | » | Reading list

## Jenkins

Dashboard > testnode > #1

Back to Project Status Changes Console Output View as plain text Edit Build Information Delete build '#1' Timings

### Console Output

Started by user unknown or anonymous  
Running as SYSTEM  
Building remotely on `winnode` in workspace `/data/worker3/workspace/testnode`  
[testnode] \$ cmd /c call C:\Users\BALASU~1\AppData\Local\Temp\jenkins6999150831734217739.bat

```
G:\data\worker3\workspace\testnode>docker -v
Docker version 20.10.11, build dea9396

G:\data\worker3\workspace\testnode>exit 0
Finished: SUCCESS
```

REST API Jenkins 2.328

DevOps\_Course\_O.pdf Verified jenkins.war Verified Show all

ENG IN 23:28 10/01/2022 19

# Kubernetes Cluster

```
Administrator: Command Prompt
Handling connection for 6060
Handling connection for 6060
Handling connection for 6060
Handling connection for 6060

G:\Local disk\jenkins\cluster>kubectl get pods
No resources found.

G:\Local disk\jenkins\cluster>kubectl get pods -n jenkins
NAME           READY   STATUS    RESTARTS   AGE
jenkins-master-5dc5d9fc57-hkc7h   1/1     Running   0          39m

G:\Local disk\jenkins\cluster>kubectl get svc -n jenkins
NAME        TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)           AGE
jenkins-master   LoadBalancer   10.106.222.154   localhost      80:31547/TCP,50000:31695/TCP   36m

G:\Local disk\jenkins\cluster>kubectl get deploy -n jenkins
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
jenkins-master   1/1     1           1          39m

G:\Local disk\jenkins\cluster>kubectl delete deploy jenkins-master -n jenkins
deployment.apps "jenkins-master" deleted

G:\Local disk\jenkins\cluster>kubectl delete svc jenkins-master -n jenkins
service "jenkins-master" deleted

G:\Local disk\jenkins\cluster>kubectl get pods -n jenkins
No resources found.

G:\Local disk\jenkins\cluster>kubectl get pods -n jenkins
```



# Cucumber jenkins

How To Integrate Cucumber With Jenkins | cucumbermavenex Config [Jenkins] | amexseleniumaug2021projects/cucumbermavenex | jacoco code coverage jenkins - Google Search | +

localhost:7070/job/cucumbermavenex/configure

Insert title here Empire New Tab How to use Assertions... Browser Automation Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

Dashboard > cucumbermavenex >

General Source Code Management Build Triggers Build Environment Pre Steps Build Post Steps Build Settings

Post-build Actions

Repositories

Repository URL: https://github.com/eswaribala/amexseleniumaug2021projects.git

Credentials: - none - Add

Advanced...

Add Repository

Branches to build

Save Apply

makeseleniumeasy....zip Verified Show all

52

# Cucumber jenkins

How To Integrate Cucumber With Jenkins | cucumbermavenex Config [Jenkins] | ameseleniumaug2021projects/cucumbermavenex | jacoco code coverage jenkins - Google Search | +

localhost:7070/job/cucumbermavenex/configure

Insert title here Empire New Tab How to use Assert... Browser Automation Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

Dashboard > cucumbermavenex >

General Source Code Management Build Triggers Build Environment Pre Steps Build Post Steps Build Settings

Post-build Actions

Add Branch

Repository browser

(Auto)

Additional Behaviours

Sparse Checkout paths

x ?

x ?

x ?

Path

/cucumbermaven

Add

Save Apply

makeseleniumeasy....zip Verified Show all

12/01/2022 01:10 ENG IN 18

# Cucumber jenkins

How To Integrate Cucumber With Jenkins | cucumbermavenex Config [Jenkins] | amexseleniumaug2021projects/cucumbermavenex | jacoco code coverage jenkins - Google Search | +

localhost:7070/job/cucumbermavenex/configure

Insert title here Empire New Tab How to use Assertions Browser Automation Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

Dashboard > cucumbermavenex >

General Source Code Management Build Triggers Build Environment Pre Steps Build Post Steps Build Settings

Post-build Actions

With Ant ?

**Pre Steps**

Add pre-build step ▾

**Build**

Root POM ?

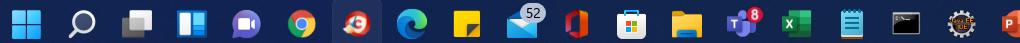
./cucumbermaven/pom.xml

Goals and options ?

-f ./cucumbermaven test

Save Apply Advanced...

makeseleniumeasy....zip Verified Show all



ENG IN 01:10 12/01/2022 18

# Cucumber jenkins

How To Integrate Cucumber With Jenkins | cucumbermavenex Config [Jenkins] | ameseleniumaug2021projects/cucumbermavenex | jacoco code coverage jenkins - Google Search | +

localhost:7070/job/cucumbermavenex/configure

Insert title here Empire New Tab How to use Assert... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 » Reading list

Dashboard > cucumbermavenex >

General Source Code Management Build Triggers Build Environment Pre Steps Build Post Steps Build Settings

Post-build Actions

Build Settings

E-mail Notification

Post-build Actions

Cucumber reports

Add post-build action ▾

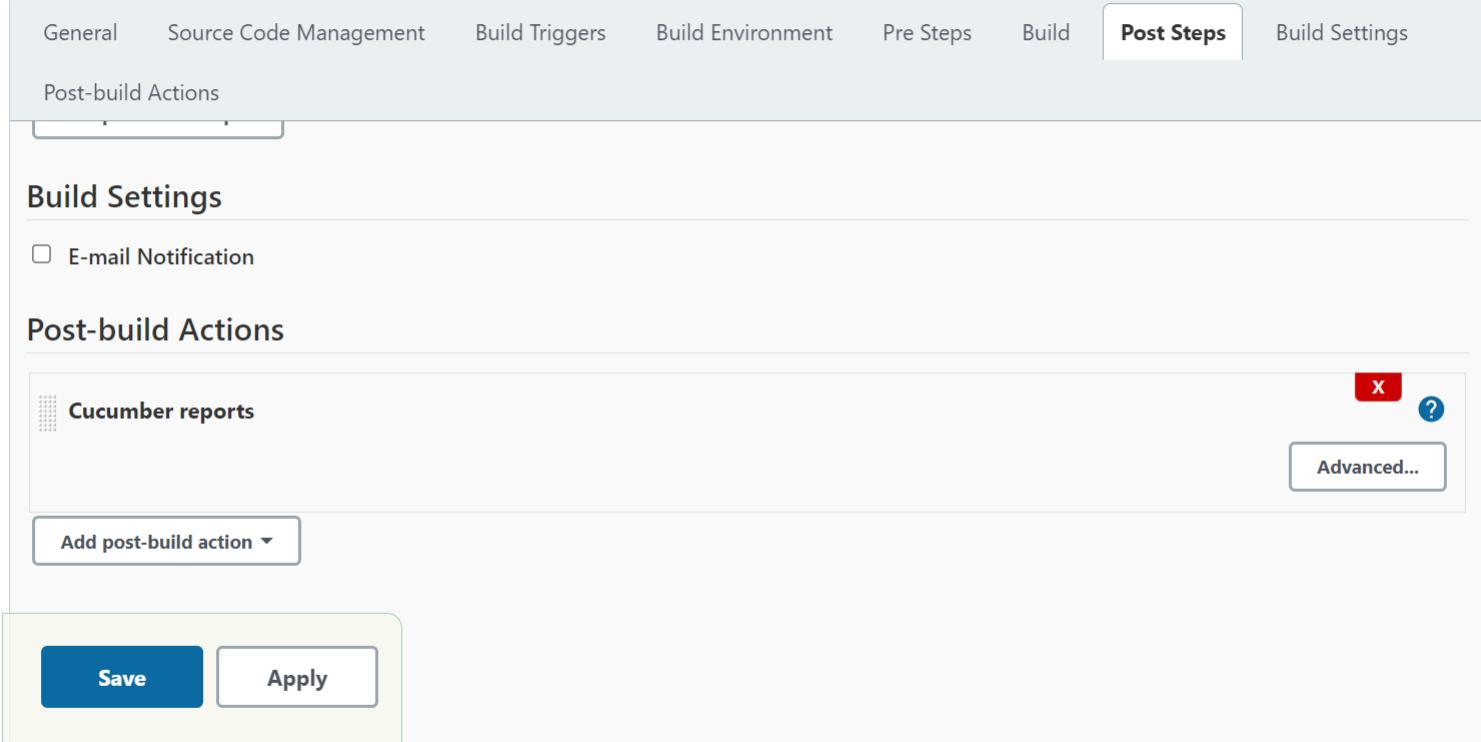
Advanced... X ?

Add post-build action ▾

Save Apply

makeseleniumeasy....zip Verified Show all

01:11 12/01/2022 18



# Cucumber jenkins

How To Integrate Cucumber With Jenkins | cucumbermavenex #9 Console | ameseleniumaug2021projects/c... | jacoco code coverage jenkins - G...

localhost:7070/job/cucumbermavenex/9/console

Insert title here Empire New Tab How to use Asserti... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel Airtel 4G Hotspot nt8F83

Reading list

Dashboard > cucumbermavenex > #9

Back to Project Status Changes Console Output View as plain text Edit Build Information Delete build '#9' Polling Log Timings Git Build Data Test Result

## Console Output

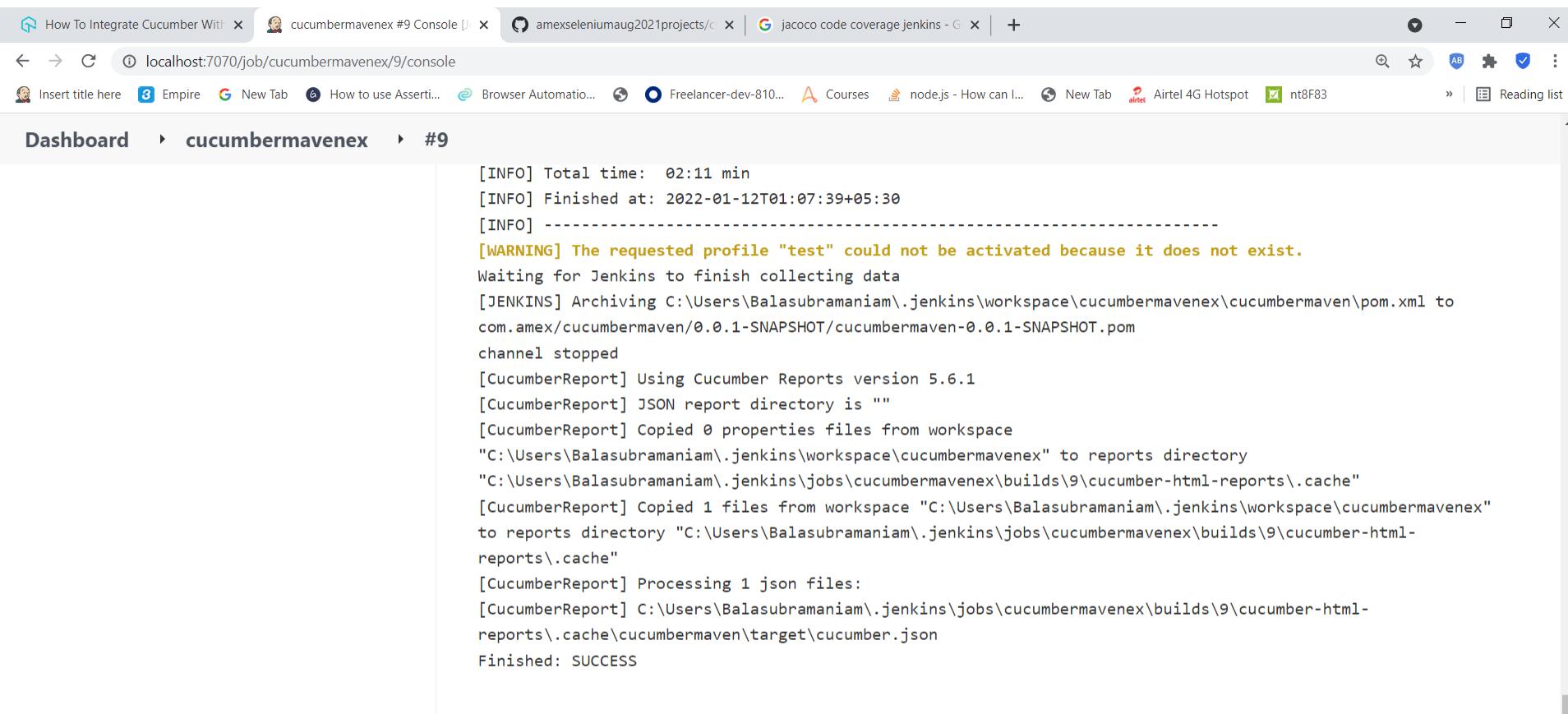
Started by an SCM change  
Running as SYSTEM  
Building in workspace C:\Users\Balasubramaniam\.jenkins\workspace\cucumbermavenex  
The recommended git tool is: NONE  
No credentials specified

```
> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir
C:\Users\Balasubramaniam\.jenkins\workspace\cucumbermavenex\.git # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\bin\git.exe config remote.origin.url
https://github.com/eswaribala/ameseleniumaug2021projects.git # timeout=10
Fetching upstream changes from https://github.com/eswaribala/ameseleniumaug2021projects.git
> C:\Program Files\Git\bin\git.exe --version # timeout=10
> git --version # 'git version 2.33.0.windows.1'
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress --
https://github.com/eswaribala/ameseleniumaug2021projects.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision c2148be505c8b74a7ad5d86f43950f705461c79c (refs/remotes/origin/master)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe read-tree -mu HEAD # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f c2148be505c8b74a7ad5d86f43950f705461c79c # timeout=10
Commit message: "updated"
```

makeseleniumeasy....zip Verified Show all

ENG IN 01:12 12/01/2022 18

# Cucumber jenkins



The screenshot shows a browser window with multiple tabs open. The active tab is 'localhost:7070/job/cucumbermavenex/9/console'. The page title is 'Dashboard > cucumbermavenex > #9'. The console output is as follows:

```
[INFO] Total time: 02:11 min
[INFO] Finished at: 2022-01-12T01:07:39+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\cucumbermavenex\cucumbermaven\pom.xml to com.amex/cucumbermaven/0.0.1-SNAPSHOT/cucumbermaven-0.0.1-SNAPSHOT.pom
channel stopped
[CucumberReport] Using Cucumber Reports version 5.6.1
[CucumberReport] JSON report directory is ""
[CucumberReport] Copied 0 properties files from workspace
"C:\Users\Balasubramaniam\.jenkins\workspace\cucumbermavenex" to reports directory
"C:\Users\Balasubramaniam\.jenkins\jobs\cucumbermavenex\builds\9\cucumber-html-reports\.cache"
[CucumberReport] Copied 1 files from workspace "C:\Users\Balasubramaniam\.jenkins\workspace\cucumbermavenex" to reports directory "C:\Users\Balasubramaniam\.jenkins\jobs\cucumbermavenex\builds\9\cucumber-html-reports\.cache"
[CucumberReport] Processing 1 json files:
[CucumberReport] C:\Users\Balasubramaniam\.jenkins\jobs\cucumbermavenex\builds\9\cucumber-html-reports\.cache\cucumbermaven\target\cucumber.json
Finished: SUCCESS
```

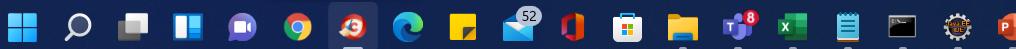
REST API

Jenkins 2.319.1

makeseleniumeasy....zip  
Verified

18 new notifications

Show all



ENG IN 01:13 12/01/2022 18

# Cucumber jenkins

How To Integrate Cucumber With Jenkins | Cucumber Reports (no 9) - Features | ameseleniumaug2021projects/cucumber-maven-examples | jacoco code coverage jenkins - Google Search | +

localhost:7070/job/cucumbermavenex/9/cucumber-html-reports/overview-features.html

Insert title here Empire New Tab How to use Assertions Browser Automation Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 » Reading list

## Cucumber Report

Jenkins Previous results Latest results **Features** Tags Steps Trends Failure

Project	Number	Date
cucumbermavenex	9	12 Jan 2022, 01:07

### Features Statistics

The following graphs show passing and failing statistics for features

#### Features

A donut chart with a single green segment representing 100% completion of features.

Feature	Steps						Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Reset Login Credentials	15	0	0	0	0	15	5	0	5	2:0.048	Passed
	15	0	0	0	0	15	5	0	5	2:0.048	1
	100.00%	0.00%	0.00%	0.00%	0.00%		100.00%	0.00%			100.00%

generate Cucumber HTML reports via: Jenkins Plugin | Standalone | Sandwich | Maven

makeseleniumeasy....zip Verified Show all

ENG IN 01:14 12/01/2022 18

# JoCaCo (jacoco:report checkstyle:checkstyle install)

How To Integrate Cucumber With ... x codecovagejacaco Config [Jen... x eswaribala/amexseleniumaug202 x | Setting up Jenkins for a Maven a... x +

localhost:7070/job/codecovagejacaco/configure

Insert title here Empire New Tab How to use Assertio... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83

Dashboard > codecovagejacaco >

General Source Code Management Build Triggers Build Environment Pre Steps Build Post Steps Build Settings

Post-build Actions

**Build**

Root POM  
./cucumbermaven/pom.xml

Goals and options  
jacoco:report checkstyle:checkstyle install

Advanced...

**Post Steps**

Post Steps  
 Run only if build succeeds  Run only if build succeeds or is unstable  Run regardless of build result  
Should the post-build steps run only for successful builds, etc.

Add post-build step ▾

**Build Settings**

E-mail Notification

**Post-build Actions**

Save Apply

makeseleniumeasy....zip Verified Show all

ENG IN 12/01/2022 18:02:22

# JoCaCo (jacoco:report checkstyle:checkstyle install)

General    Source Code Management    Build Triggers    Build Environment    Pre Steps    Build    Post Steps    Build Settings

**Post-build Actions**

**Record JaCoCo coverage report**

**Path to exec files (e.g.: `**/target/**.exec, **/jacoco.exec`)**: `**/*_.exec`

**Inclusions (e.g.: `**/*.class`)**: `*/*.class`

**Exclusions (e.g.: `**/*Test*.class`)**: `src/main/java`

**Path to class directories (e.g.: `**/target/classDir, **/classes`)**: `**/classes`

**Path to source directories (e.g.: `**/mySourceFiles`)**: `**/src/main/java`

**Inclusions (e.g.: `**/*.java, **/*.groovy, **/*.gs`)**: `**/*_.java, **/*_.groovy, **/*_.kt, **/*_.kts`

**Exclusions (e.g.: `generated/**/*.java`)**:

**Disable display of source files for coverage** ?

**Change build status according to the defined thresholds** ?

**Always run coverage collection, even if build is FAILED or ABORTED** ?

	Instruction	% Branch	% Complexity	% Line	% Method	% Class
	0	0	0	0	0	0
	0	0	0	0	0	0

**Save**    **Apply**    Build fails if any metric upgrades more than the delta thresholds ?

# JoCaCo (jacoco:report checkstyle:checkstyle install)

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** seleniumamexworkspace - MavenCucumber5JunitSampleProject/pom.xml - Eclipse IDE
- Menu Bar:** File Edit Navigate Search Project Run Design Window Help
- Toolbar:** Standard Eclipse toolbar with various icons for file operations.
- Project Explorer:** Shows the project structure:
  - MavenCucumber5JunitSampleProject [ame]
  - src/main/java: DemoPkg
  - src/main/resources: Resources, SomeData
  - src/test/java: runnerFiles (RunCucumberTest.java, RunJUnitTest.java, StepDefinitions.java, login.feature), stepDefinitions (DemoFeatureStepDef.java)
  - src/test/resources: featureFiles (DemoFeature.feature)
  - Maven Dependencies
  - JRE System Library [JavaSE-1.8]
  - src
  - target: cucumber-reports (formatter.js, index.html, jquery-3.4.1.min.js, report.js, style.css), generated-sources, generated-test-sources, maven-status, surefire-reports (cucumber.json, cucumberJsonReport), pom.xml
- Editor:** Displays the pom.xml file content:

```
50<plugins>
51<plugin>
52    <groupId>org.jacoco</groupId>
53    <artifactId>jacoco-maven-plugin</artifactId>
54    <version>0.7.9</version>
55    <executions>
56        <execution>
57            <goals>
58                <goal>prepare-agent</goal>
59            </goals>
60        </execution>
61        <execution>
62            <id>generate-code-coverage-report</id>
63            <phase>test</phase>
64            <goals>
65                <goal>report</goal>
66            </goals>
67        </execution>
68    </executions>
69</plugin>
```
- Outline View:** Shows the project structure and the selected plugin configuration.
- Task List:** Shows the available tasks for the project.
- Bottom Status Bar:** Shows system information including memory usage (244M of 299M) and system icons.

# JoCaCo (jacoco:report checkstyle:checkstyle install)

How To Integrate Cucumber Wit... x codecovergajacoco #3 Console x eswaribala/amexseleniumaug202 x Jacoco in Maven Projects - Perso... x +

localhost:7070/job/codecovergajacoco/3/console

Insert title here Empire New Tab How to use Assert... Browser Automation Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

Dashboard > codecovergajacoco > #3

```
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/doxia/doxia-logging-api/1.4/doxia-logging-api-1.4.jar (11 kB at 9.0 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/sslext/sslext/1.2-0/sslext-1.2-0.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/doxia/doxia-decoration-model/1.4/doxia-decoration-model-1.4.jar (61 kB at 45 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/struts/struts-core/1.3.8/struts-core-1.3.8.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/doxia/doxia-module-xhtml/1.4/doxia-module-xhtml-1.4.jar (15 kB at 11 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/struts/struts-taglib/1.3.8/struts-taglib-1.3.8.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/doxia/doxia-site-renderer/1.4/doxia-site-renderer-1.4.jar (53 kB at 38 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/struts/struts-tiles/1.3.8/struts-tiles-1.3.8.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/doxia/doxia-module-fml/1.4/doxia-module-fml-1.4.jar (38 kB at 26 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/commons-collections/commons-collections/3.2.1/commons-collections-3.2.1.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/sslext/1.2-0/sslext-1.2-0.jar (26 kB at 17 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/doxia/doxia-integration-tools/1.6/doxia-integration-tools-1.6.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/struts/struts-tiles/1.3.8/struts-tiles-1.3.8.jar (120 kB at 71 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/commons-io/commons-io/1.4/commons-io-1.4.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/struts/struts-taglib/1.3.8/struts-taglib-1.3.8.jar (252 kB at 148 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-9/plexus-container-default-1.0-alpha-9.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/commons-collections/commons-collections/3.2.1/commons-collections-3.2.1.jar (575 kB at 323 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-resources/1.1.0/plexus-resources-1.1.0.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/struts/struts-core/1.3.8/struts-core-1.3.8.jar (329 kB at 184 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/com/puppycrawl/tools/checkstyle/8.29/checkstyle-8.29.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/doxia/doxia-integration-tools/1.6/doxia-integration-tools-1.6.jar (45 kB at 25 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/info/picocli/picocli/4.1.4/picocli-4.1.4.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/commons-io/commons-io/1.4/commons-io-1.4.jar (109 kB at 55 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/antlr/antlr4-runtime/4.8-1/antlr4-runtime-4.8-1.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-9/plexus-container-default-1.0-alpha-9.jar (195 kB at 98 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/com/google/guava/guava/28.2-jre/guava-28.2-jre.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-resources/1.1.0/plexus-resources-1.1.0.jar (23 kB at 11 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/checkerframework/checker-qual/2.10.0/checker-qual-2.10.0.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/info/picocli/picocli/4.1.4/picocli-4.1.4.jar (352 kB at 165 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/net/sf/saxon/Saxon-HE/9.9.1-6/Saxon-HE-9.9.1-6.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/com/puppycrawl/tools/checkstyle/8.29/checkstyle-8.29.jar (1.3 MB at 579 kB/s)
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/checkerframework/checker-qual/2.10.0/checker-qual-2.10.0.jar (201 kB at 85 kB/s)
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/antlr/antlr4-runtime/4.8-1/antlr4-runtime-4.8-1.jar (338 kB at 138 kB/s)
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/com/google/guava/guava/28.2-jre/guava-28.2-jre.jar (2.8 MB at 1.0 MB/s)
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/net/sf/saxon/Saxon-HE/9.9.1-6/Saxon-HE-9.9.1-6.jar (5.5 MB at 1.7 MB/s)
```

makeseleniumeasy....zip Verified Show all

Windows Taskbar: File Explorer, Search, Task View, Microsoft Edge, Microsoft Word, Microsoft Outlook, Microsoft OneDrive, Microsoft Teams, Microsoft Excel, Microsoft Word, Microsoft Powerpoint, Microsoft Edge, Microsoft Edge, Microsoft Edge, ENG IN, WiFi, Battery, 01:35, 12/01/2022

# JoCaCo (jacoco:report checkstyle:checkstyle install)

```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 02:22 min
[INFO] Finished at: 2022-01-12T01:36:38+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\codecovergaejacoco\cucumbermaven\pom.xml to com.amex\cucumbermaven\0.0.1-SNAPSHOT\cucumbermaven-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving C:\Users\Balasubramaniam\.jenkins\workspace\codecovergaejacoco\cucumbermaven\target\cucumbermaven-0.0.1-SNAPSHOT.jar to com.amex\cucumbermaven\0.0.1-SNAPSHOT\cucumbermaven-0.0.1-SNAPSHOT.jar
channel stopped
[JaCoCo plugin] Collecting JaCoCo coverage data...
[JaCoCo plugin] **/*.exec;**/classes;**/src/main/java; locations are configured
[JaCoCo plugin] Number of found exec files for pattern **/*.exec: 1
[JaCoCo plugin] Saving matched execfiles: C:\Users\Balasubramaniam\.jenkins\workspace\codecovergaejacoco\cucumbermaven\target\jacoco.exec
[JaCoCo plugin] Saving matched class directories for class-pattern: **/classes:
[JaCoCo plugin] - C:\Users\Balasubramaniam\.jenkins\workspace\codecovergaejacoco\cucumbermaven\target\classes 1 files
[JaCoCo plugin] Saving matched source directories for source-pattern: **/src/main/java:
[JaCoCo plugin] Source Inclusions: **/*.java,**/*.groovy,**/*.kt,**/*.kts
[JaCoCo plugin] Source Exclusions:
[JaCoCo plugin] - C:\Users\Balasubramaniam\.jenkins\workspace\codecovergaejacoco\cucumbermaven\src\main\java 1 files
[JaCoCo plugin] Loading inclusions files...
[JaCoCo plugin] inclusions: [*/*.class]
[JaCoCo plugin] exclusions: [src/main/java]
[JaCoCo plugin] Thresholds: JacocoHealthReportThresholds [minClass=0, maxClass=0, minMethod=0, maxMethod=0, minLine=0, maxLine=0, minBranch=0, maxBranch=0, minInstruction=0, maxInstruction=0, minComplexity=0, maxComplexity=0]
[JaCoCo plugin] Publishing the results...
[JaCoCo plugin] Loading packages...
[JaCoCo plugin] Done.
[JaCoCo plugin] Overall coverage: class: 0.0, method: 0.0, line: 0.0, branch: 100.0, instruction: 0.0, complexity: 0.0
Finished: SUCCESS
```

REST API Jenkins 2.319.1

makeseleniumeasy....zip  
Verified

Show all



ENG IN 01:36 12/01/2022 18 352

# JoCaCo (jacoco:report checkstyle:checkstyle install)

How To Integrate Cucumber With Jenkins | eswaribala/amxseleniumaug2022 | Jacoco in Maven Projects - Personal

localhost:7070/job/codecovergajacoco/lastBuild/jacoco/ Insert title here Empire New Tab How to use Asserti... Browser Automatio... Freelancer-dev-810... Courses node.js - How can I... New Tab Airtel 4G Hotspot nt8F83 Reading list

## Jenkins

Dashboard > codecovergajacoco > #3 > Jacoco

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[Edit Build Information](#)

[Delete build '#3'](#)

[Polling Log](#)

[Timings](#)

[Git Build Data](#)

[Test Result](#)

[Redeploy Artifacts](#)

[Coverage Report](#)

[See Fingerprints](#)

[Open Blue Ocean](#)

[makeseleniumeasy...zip](#) Verified

### JaCoCo Coverage Report

[Download jacoco.exec binary coverage file](#)

The chart displays a single horizontal bar from 0 to 1. A green dot at position 1 indicates a line covered, while a red dot at position 0 indicates a line missed.

[line covered](#) [line missed](#)

### Overall Coverage Summary

name	instruction	branch	complexity	line	method	class
all classes	0%	100%	0%	0%	0%	0%
	M: 3 C: 0	M: 0 C: 0	M: 1 C: 0	M: 1 C: 0	M: 1 C: 0	M: 1 C: 0

### Coverage Breakdown by Package

name	instruction	branch	complexity	line	method	class
DemoPkg	M: 3 C: 0 0%	M: 0 C: 0 100%	M: 1 C: 0 0%			

## What Is Karate Framework

---

- Karate framework follows the Cucumber style of writing the program which follows the BDD approach.
- The syntax is easy to understand by non-programmers.
- And this framework is the only API testing tool that has combined API Automation and performance testing into a single standalone tool.
- It provides the users with the ability to execute the test cases in Parallel and perform the JSON & XML checks.

## What Is Karate Framework

---

- Karate is a BDD testing framework instead of a TDD.
- It is designed to be easy for non-programmers.
- This feature is a game-changer as it allows for more use and access by many people regardless of their technical background or capacity.
- It makes use of the Cucumber feature file and the Gherkins language to write the test which is very easy to understand.

## Karate Features

---

- Makes use of easy-to-understand Gherkins language.
- It requires no technical programming knowledge like Java.
- It is based on the popular Cucumber standards.
- Easy to create a framework.
- Parallel testing is the core functionality that is provided by the Karate itself, hence we need not depend on Maven, Gradle, etc.
- UI for debugging the Test.
- Calling a feature file from another file.

# Karate Features

---

- Provides supports for the Data Driver Testing that is built in-house, hence no need to depend on external frameworks.
- Built-in Native Rest Reports. Plus, it can be integrated with the Cucumber for better UI Reports and more clarity.
- Provides in-house support for switching configuration across different testing environments (QA, Stage, Prod, Pre-Prod).
- Seamless support for CI/CD integration that can be useful.
- Capable of handling various HTTP calls:
- Web Socket support
  - SOAP request
  - HTTP
  - Browser cookie handling
  - HTTPS
  - HTML-form data
  - XML request

**The following table enlists a few more prominent differences between Rest-Assured & Karate Framework:**

S.No	Basis	Karate Framework	REST-Assured
1	<b>Language</b>	It uses a combination of Cucumber and Gherkins	It makes use of Java Language
2	<b>Code Size</b>	Usually, the line of code is less, since it follows Cucumber-like structure	Line of code is more since it involves the usage of Java language
3	<b>Technical Knowledge required</b>	Non- Programmers can easily write the Gherkins code	Technical knowledge is required to write Java code
4	<b>Data-Driven Testing</b>	Need to make use of TestNG or equivalent to support the same	In-house tags can be used to support data testing
5	<b>Does it provide SOAP call support</b>	Yes, it does provide	It is only related to a REST request
6	<b>Parallel Testing</b>	Yes, parallel testing is easily supported with the parallel report generation too	Not to a large extent. Though people have tried doing this, the failure rate is more than the success rate

# Karate vs Rest Assured

7	<b>Reporting</b>	It provides in-house reporting, hence doesn't need to be dependent on external plugins. We can even integrate it with Cucumber reporting plugin for better UI.	Need to be dependent on External Plugins like Junit, TestNG
8	<b>CSV support for external Data</b>	Yes, from Karate 0.9.0	No, have to use Java Code or library
9	<b>Web UI Automation</b>	Yes, from Karate 0.9.5 Web-UI Automation is possible	No, it's not supported
10	<b>Sample GET</b>	Given param val1 = 'name1' And param val2 = 'name2' And path 'somedirection' When method get Then match response contains 'OKAY'	given(). param("val1", "name1"). param("val2", "name2"). when(). get("/some\location"). then(). body(containsString("OKAY"));

Hence, as demonstrated by the differences above, it is safe to say that Karate is one of the easiest things

# 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

