DevOps

Caltech | Center for Technology & Management Education

Post Graduate Program in DevOps

simplilearn

DevOps

Caltech | Center for Technology & Management Education

CI/CD Pipeline with Jenkins

simplilearn

Jenkins Integrations

Learning Objectives

By the end of this lesson, you will be able to:

- Setup JFrog Artifactory using installer
- Perform Continuous Delivery using JFrog Artifactory
- Integrate Jenkins and Slack collaboration tool
- Setup a Slack channel for build notifications
- Setup Apache Tomcat
- List the code scanning tools and metrics

JFrog Artifactory

Artifactory

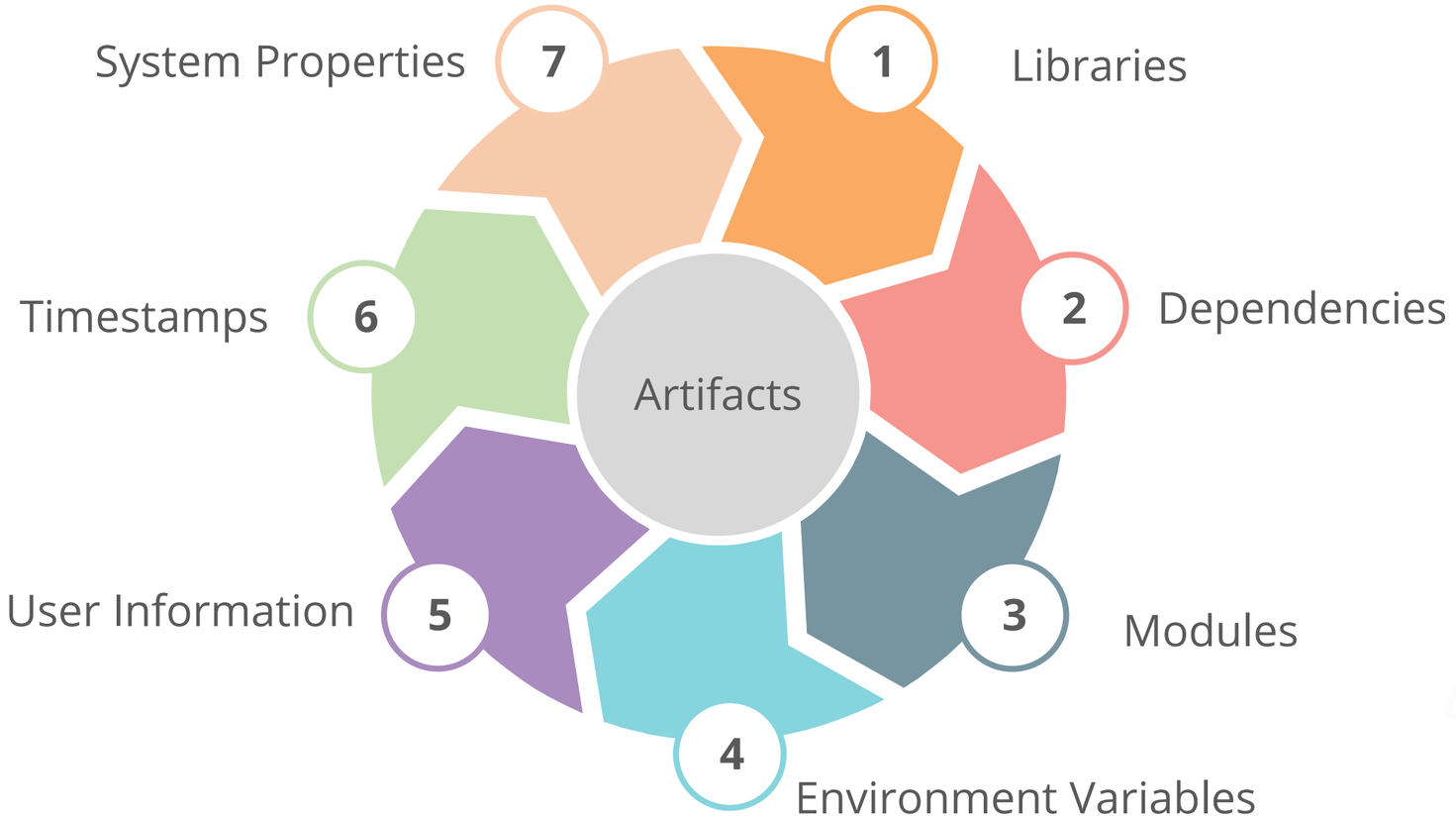
Artifactory is a distributed repository management tool that can store binary artifacts and share them using remote repositories.

- Helps developers manage software releases with proper versioning
- Supports high availability
- Tags libraries with metadata, making it easy to search for a specific artifact
- Stores Build artifacts post -CI process
- Supports authentication and authorization

Artifactory

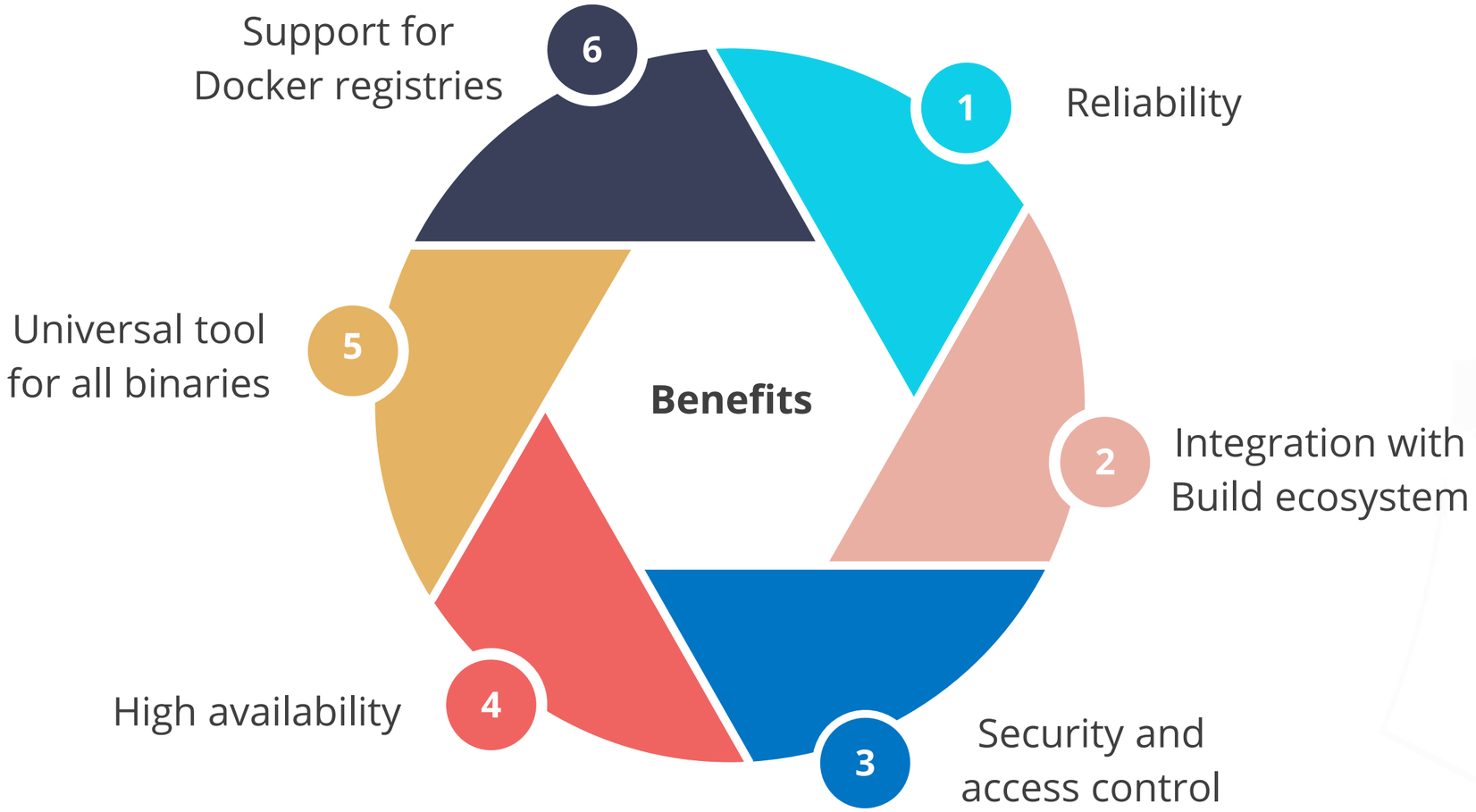
Artifactory separates binary data from their metadata. It stores the binaries in a file store while the metadata is stored in a database. It supports all popular package formats.

The Artifactory stores an exhaustive amount of information including:



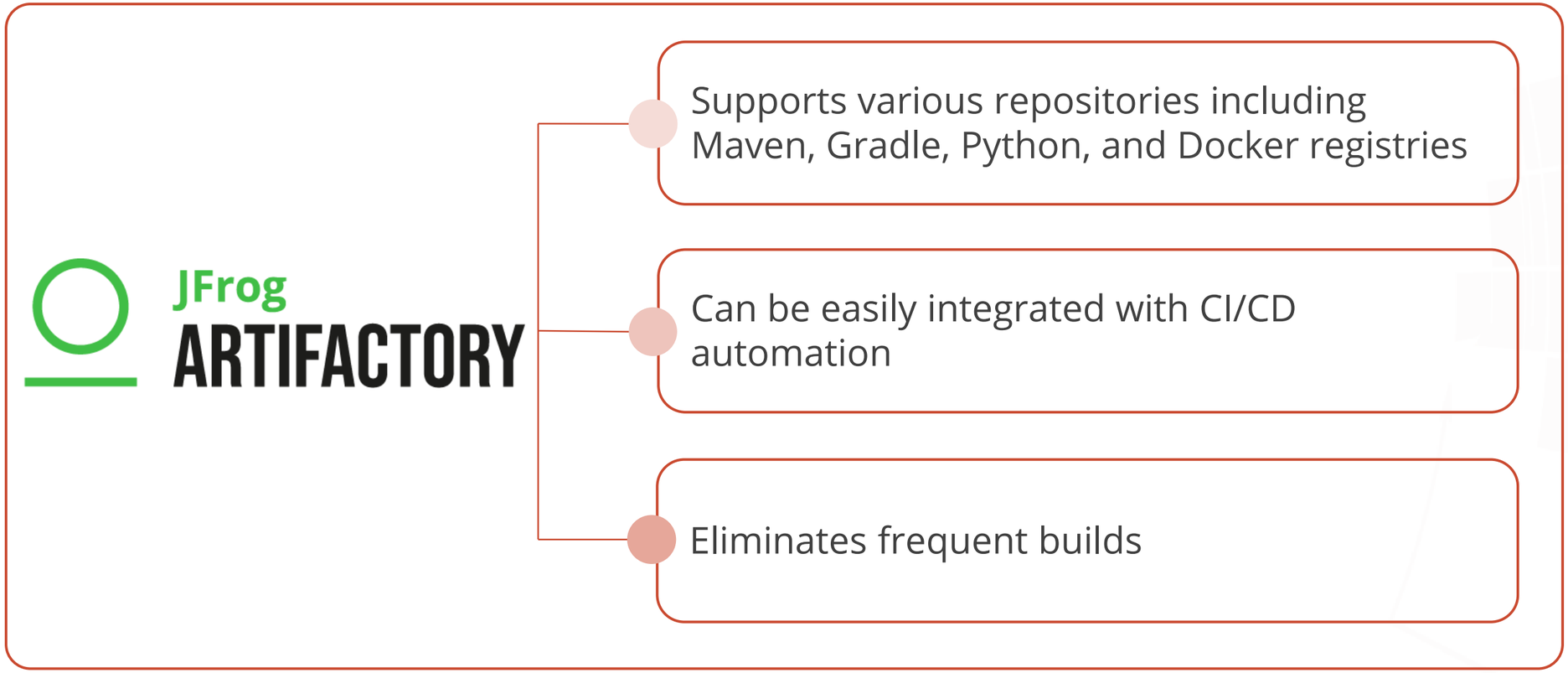
Artifactory

Artifactory provides the following benefits:



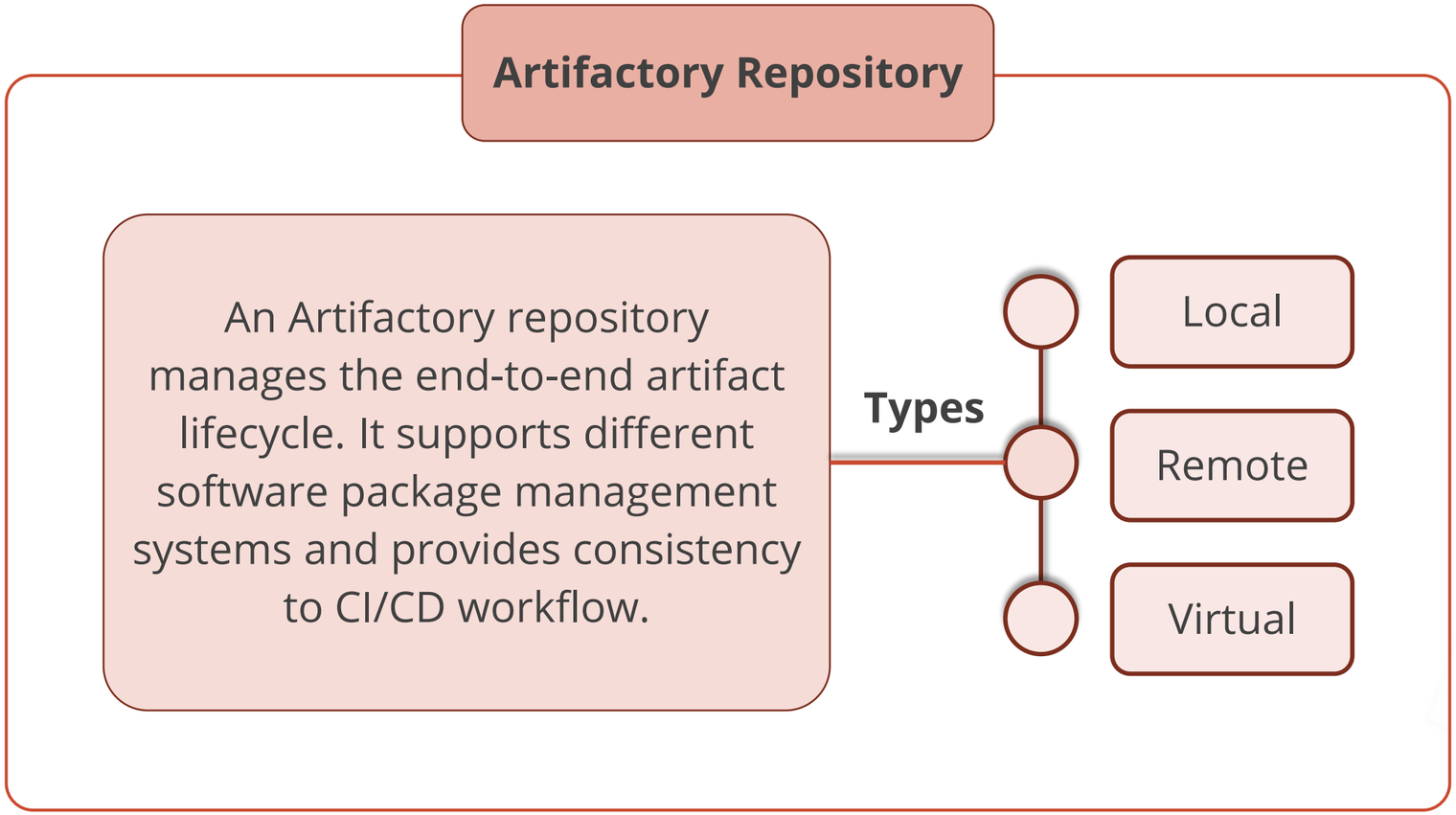
JFrog Artifactory

JFrog Artifactory is a universal artifact repository manager that supports all major packaging formats. It provides an end-to-end management and automation of binaries and artifacts.



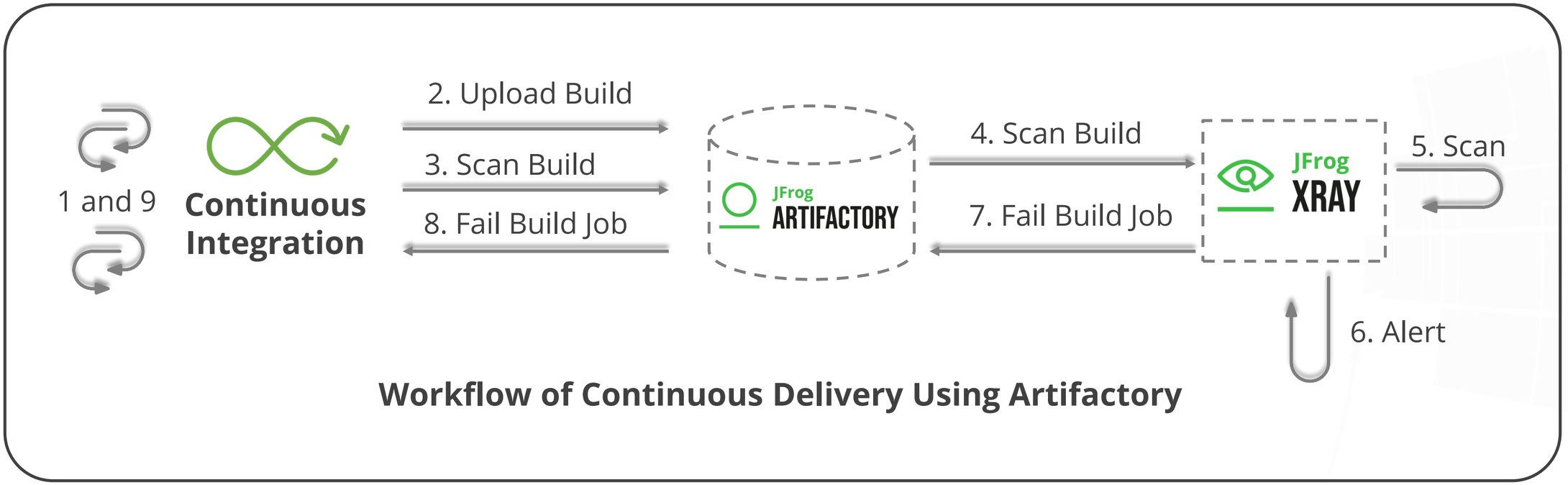
JFrog Repositories

JFrog Artifactory supports remote repositories that can be used to store different types of binaries and Build artifacts.



Continuous Delivery Using Artifactory

Continuous Delivery is the mechanism of delivering source code artifacts by automating compilation, packaging, and uploading artifacts to Artifactory.

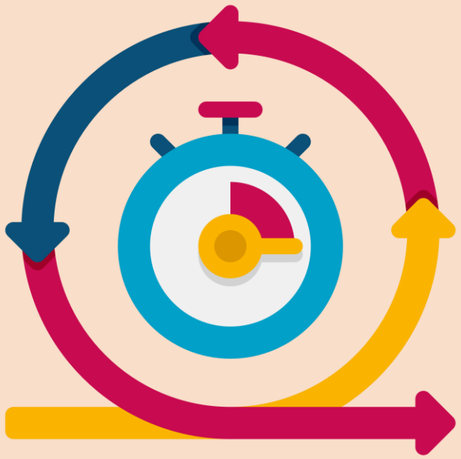


Workflow of Continuous Delivery Using Artifactory

Jenkins Artifactory Plugin

Jenkins Artifactory plugin helps developers push artifacts to Artifactory while implementing Continuous Integration.

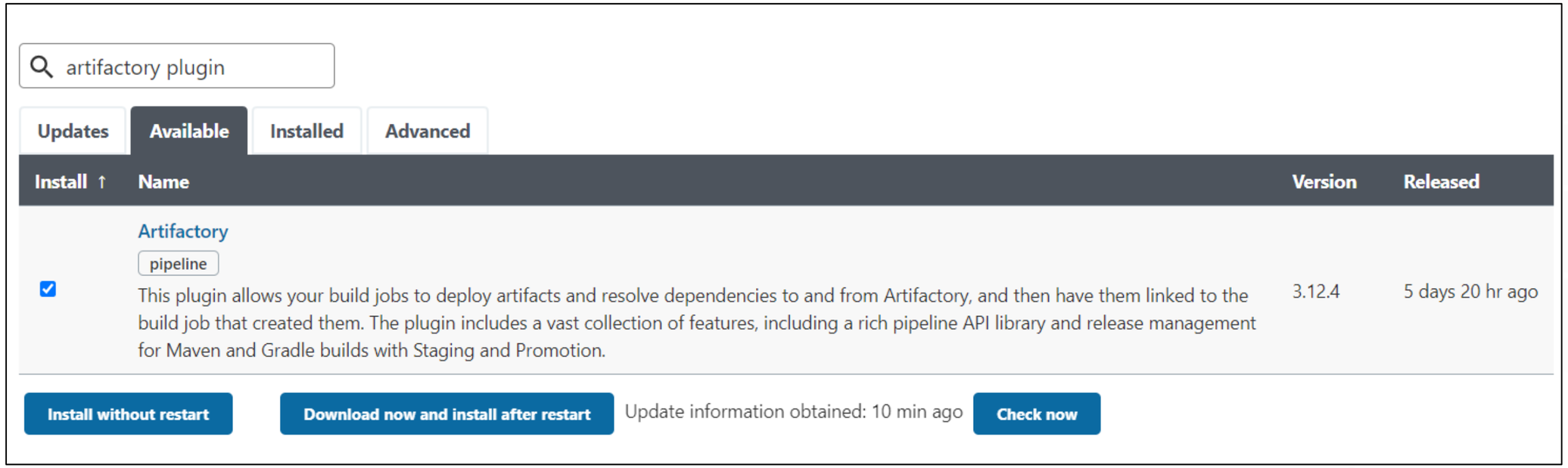
Jenkins Artifactory plugin can be used to upload any Build to the Artifactory repository



Artifacts can be used to perform deployment.

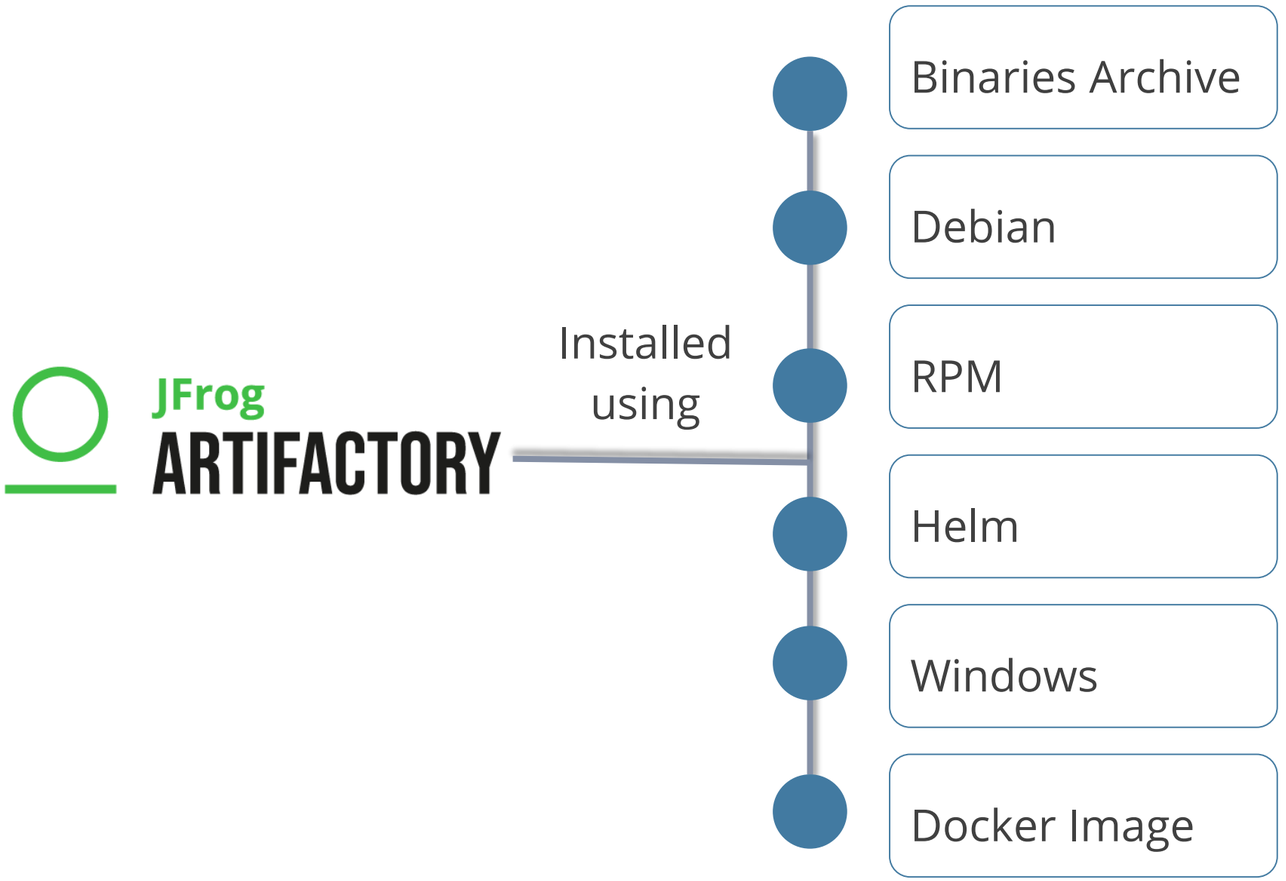
Jenkins Artifactory Plugin

To download the Jenkins Artifactory plugin, go to the Available tab in the Jenkins dashboard, select the plugin, and click Download now and install after restart, as shown below:



Set up Artifactory Using Installer

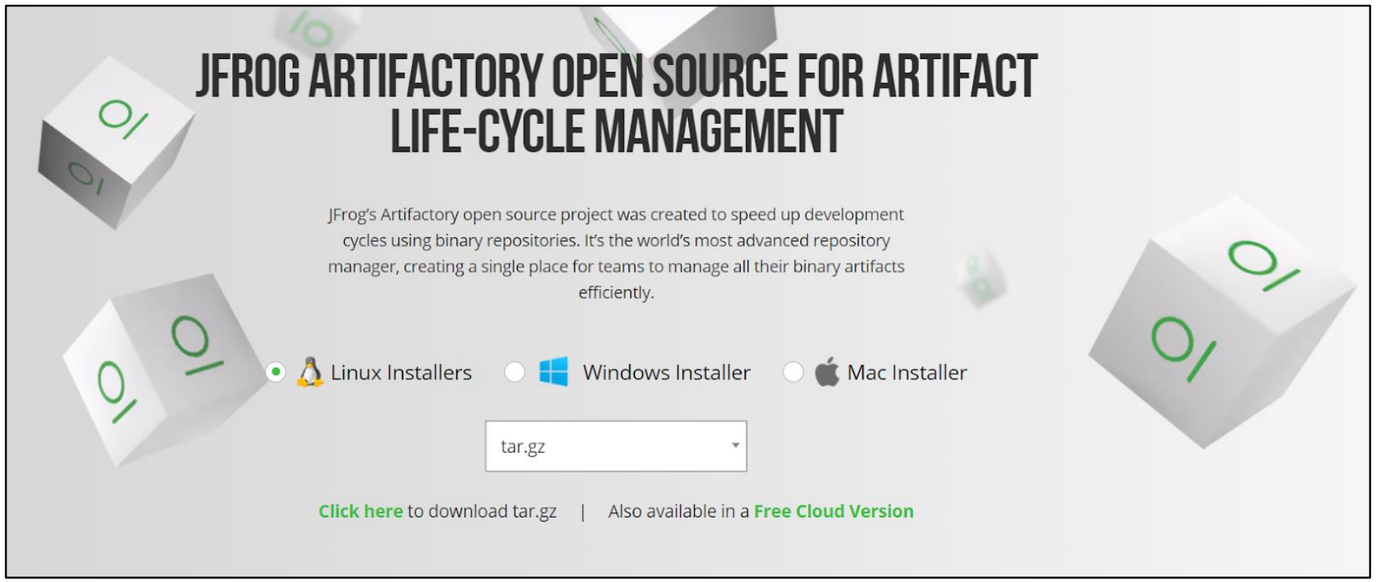
JFrog Artifactory can be installed as a single node installation or as a high availability cluster.



Set up Artifactory Using Installer

JFrog Artifactory installer details can be fetched by accessing the link mentioned below:

https://jfrog.com/open-source/#Artifactory



Install Artifactory Using Binary Archive

An Artifactory can be installed on a Linux server using Binary Archive.

Download the installer here:
https://releases.jfrog.io/artifactory/bintray-artifactory/org/artifactory/oss/jfrog-artifactory-oss/[RELEASE]/jfrog-artifactory-oss-[RELEASE]-linux.tar.gz

To complete the installation, execute the following commands:

Demo-1

```
cd /opt
     tar -xzf jfrog-artifactory-oss-\[RELEASE\]-linux.tar.gz
     ls -lart artifactory-oss-7.21.5/
     ./artifactory-oss-7.21.5/app/bin/installService.sh
     service Artifactory restart
     service Artifactory status
```

Install Artifactory Using Windows Installer

The .zip archive binaries file provided by the Artifactory can be extracted to deploy the Artifactory on a Windows machine.

Download the installer here:
https://releases.jfrog.io/artifactory/bintray-artifactory/org/artifactory/oss/jfrog-artifactory-oss/[RELEASE]/jfrog-artifactory-oss-[RELEASE]-windows.zip

1. Extract the .zip file

2. Execute the .bat file in %JFROG_HOME%\artifactory\app\bin directory

Install Artifactory Using Debian

Artifactory can be installed on a Debian or Ubuntu-based server using .deb package.

To install Artifactory on a Debian or Ubuntu-based operating system, execute the following commands:

Demo-1

```
wget -qO - https://releases.jfrog.io/artifactory/api/gpg/key/public | sudo apt-key
add -;
echo "deb https://releases.jfrog.io/artifactory/artifactory-debs {distribution} main"
| sudo tee -a /etc/apt/sources.list;
sudo apt-get update
sudo apt-get install jfrog-artifactory-oss
```

Install Artifactory Using Yum Installer

Artifactory supports installation on RPM-based operating system using .rpm package or yum command.

To install Artifactory on CentOS, Fedora, or RedHat machines, execute the following commands:

Demo-1

```
wget https://releases.jfrog.io/artifactory/artifactory-rpms/artifactory-rpms.repo -O
jfrog-artifactory-rpms.repo;
sudo mv jfrog-artifactory-rpms.repo /etc/yum.repos.d/;
sudo yum update
sudo yum install jfrog-artifactory-oss
```

Install Artifactory Using Yum Installer

Artifactory can also be deployed using a Docker image on a Docker host.

To deploy Artifactory on a Docker host, execute the following commands:

Demo-1

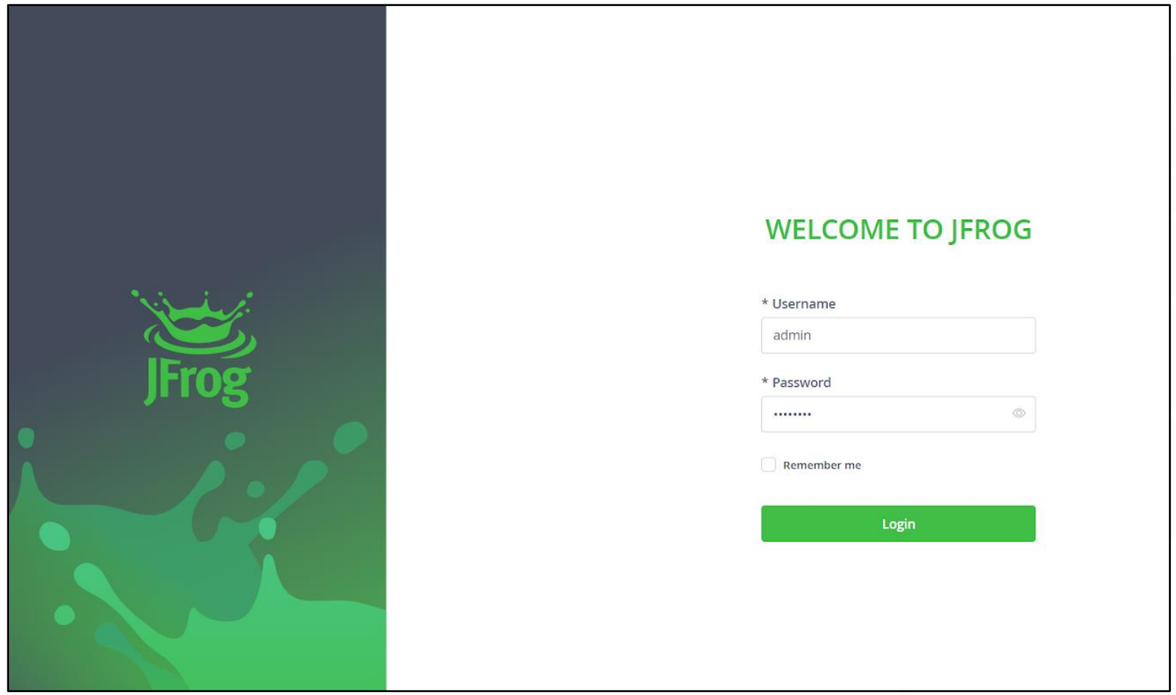
```
docker run --name Artifactory -d -p 8081:8081 -p 8082:8082 releases-
docker.jfrog.io/jfrog/artifactory-oss:latest
docker ps
docker logs -f Artifactory
docker <stop/start> Artifactory
```

Artifactory Interface

The Artifactory GUI can be accessed to create and manage Artifactory repositories.

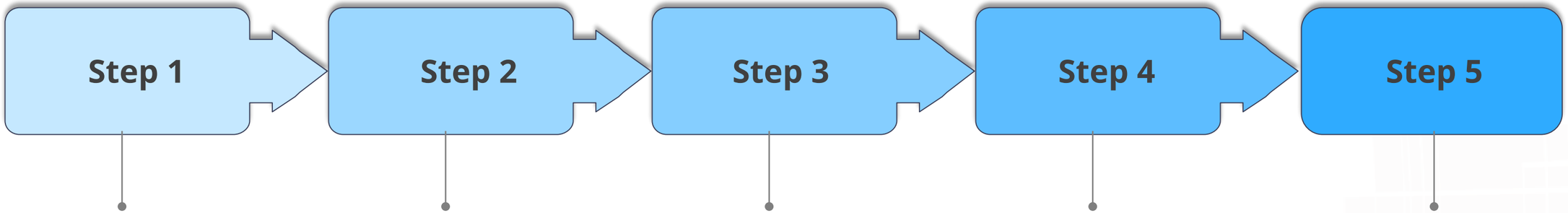
To access Artifactory from the browser:

http://[SERVER HOST]:8082/ui/
Username: **admin**
Password: **password**



Set up a New Repository Using Artifactory

To set up a new repository using Artifactory, follow the steps shown below:



Login to JFrog Artifactory; click **Create a Repository**

Select the **type of repository** (local, remote, or virtual) to be created

Select the **Package type** based on the binaries that will be deployed in the Artifactory

Provide the **details** of the repository, if created as a local repository

Once the repository is created, click on **Set me up** to configure upload or download

Artifacts: Uploading and Downloading

An artifact can be uploaded using the following command:

Demo-1

```
curl -u<USERNAME>:<PASSWORD> -T <PATH_TO_FILE> "http://[ ARTIFACTORY HOST
]:8081/artifactory/artifactory-repo-local/<TARGET_FILE_PATH>"
```

An artifact can be downloaded using the following command:

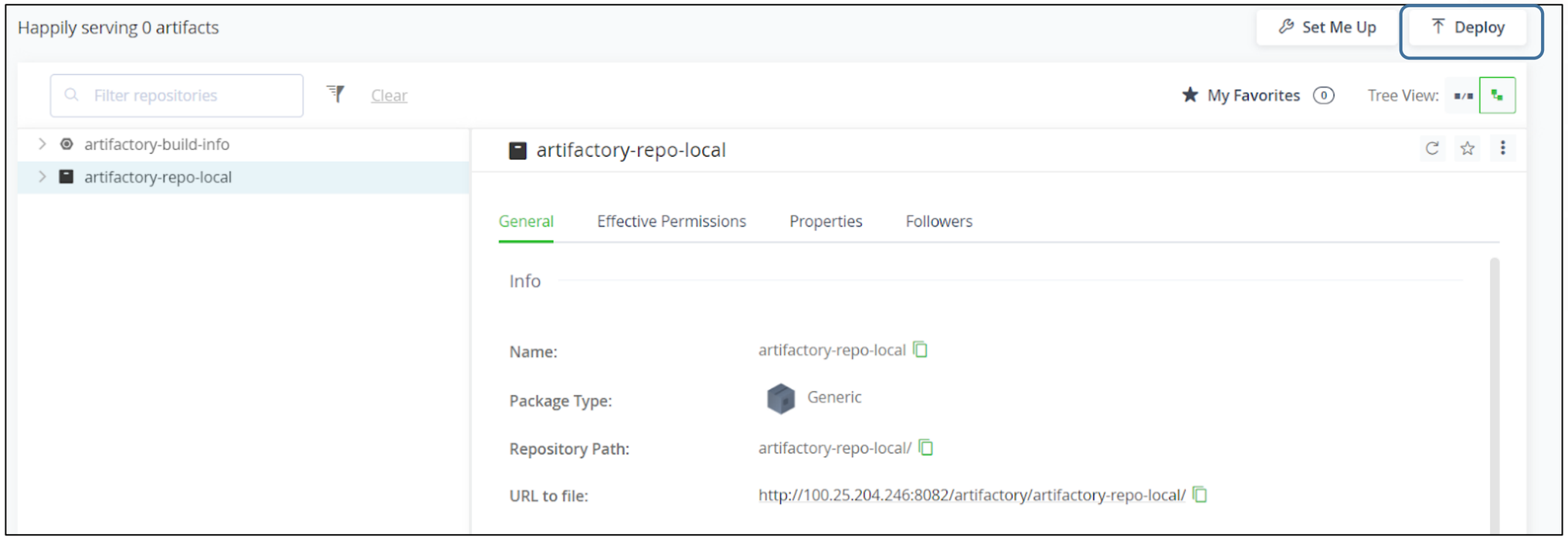
Demo-1

```
curl -u<USERNAME>:<PASSWORD> -O "http://[ ARTIFACTORY HOST
]:8081/artifactory/artifactory-repo-local/<TARGET_FILE_PATH>"
```

Upload Artifacts Using JFrog Artifactory

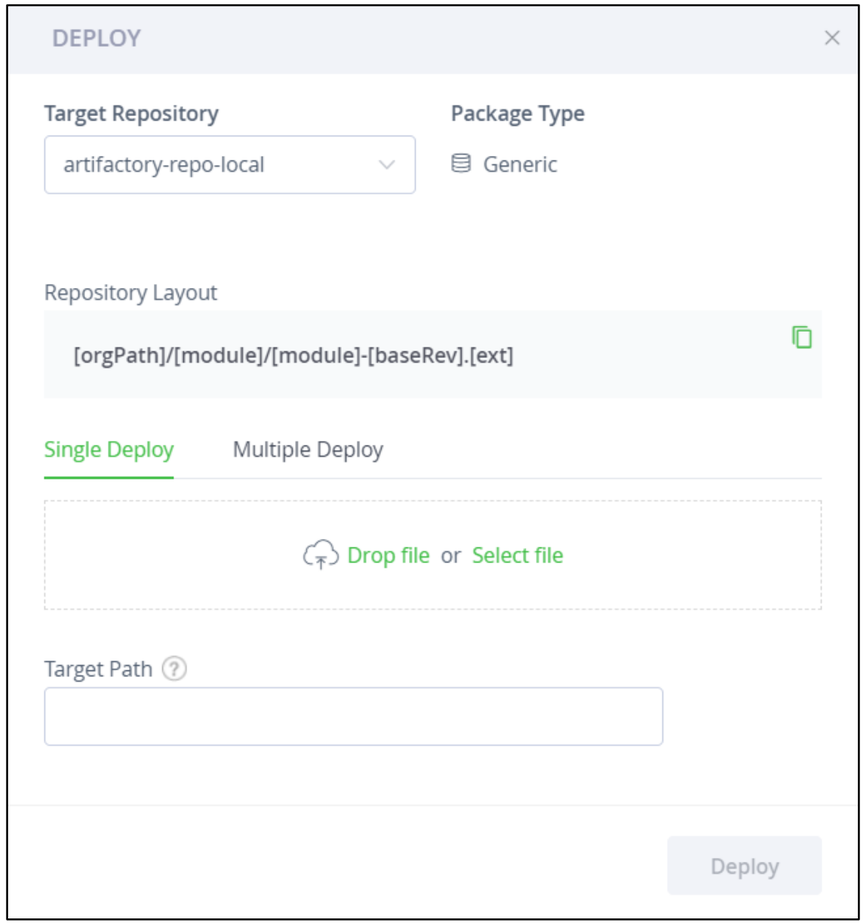
Custom Build artifacts can be uploaded to the Artifactory repository using Artifactory interface. Follow the steps below to achieve this:

Step 1: Click the Deploy button on the Artifactory interface



Upload Artifacts Using JFrog Artifactory

Step 2: Select the target repository and the package type, and click Deploy



Assisted Practice

Configure JFrog Artifactory for Publishing Artifacts

Duration: 20 min

Problem Statement:

Configure JFrog Artifactory to publish artifacts.

Assisted Practice: Guidelines

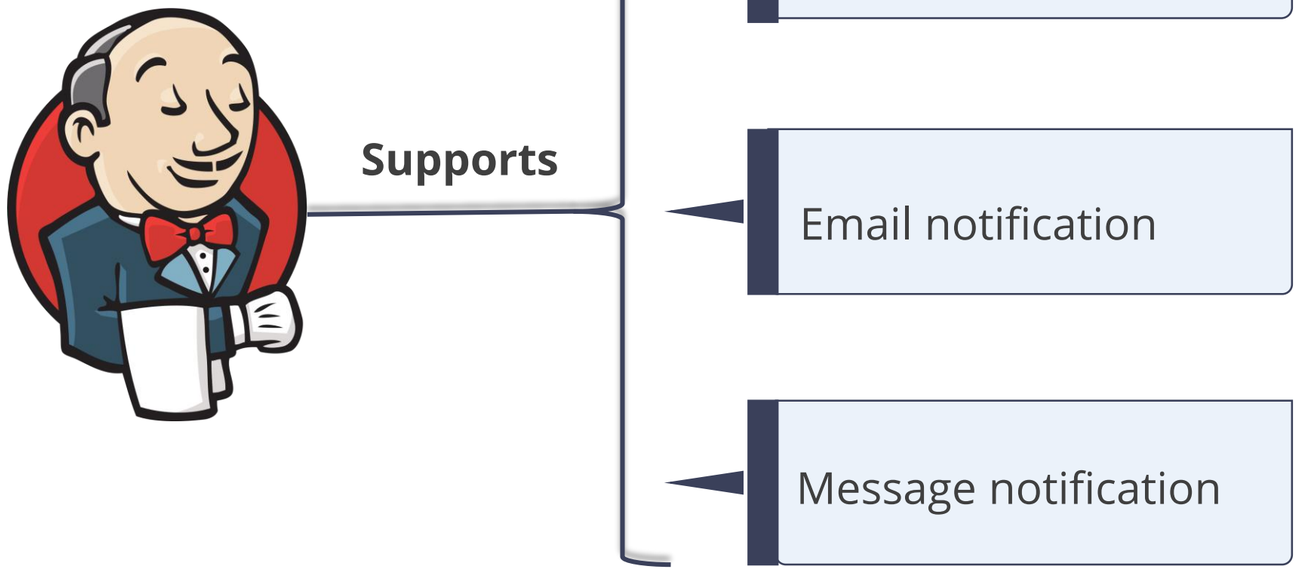
Steps to configure JFrog Artifactory:

1. Install JFrog Artifactory on Ubuntu VM.
2. Log in to Jenkins CI tool and install Artifactory plugin.

Slack Collaborative Tool

Jenkins Notifications

Jenkins enables report Build status and test results notifications to be sent to the development team. If the Build succeeds, developers will receive a success notification; otherwise, a failure notification will be sent.



Slack

Slack is a workplace communication platform that has hooks to provide communication from various tools to the development team.

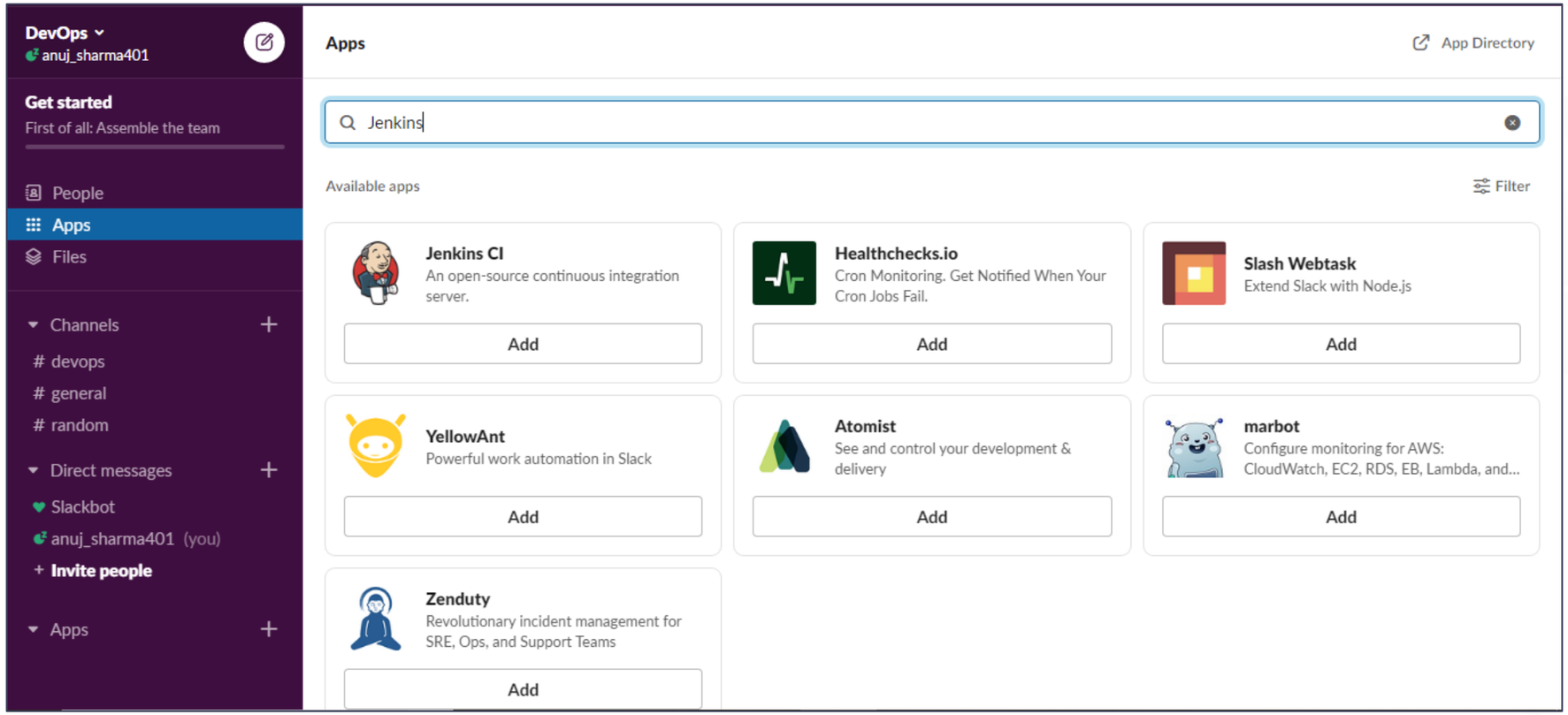
Slack can be used by Jenkins to send notifications from all the jobs to the team.

Slack channels can be used to send specific notifications to specific teams.

Slack provides Jenkins hook as an app that can be accessed on the Slack workspace.

Slack Channel Setup for Build Notifications

To set up Jenkins Build notifications, click the **Apps** tab on the left panel of the Slack workspace. Then, select **Jenkins CI** app.



Slack Channel Setup for Build Notifications

Configure the following details in Slack workspace:

1. Post-build details
2. Customized name
3. Customize icon

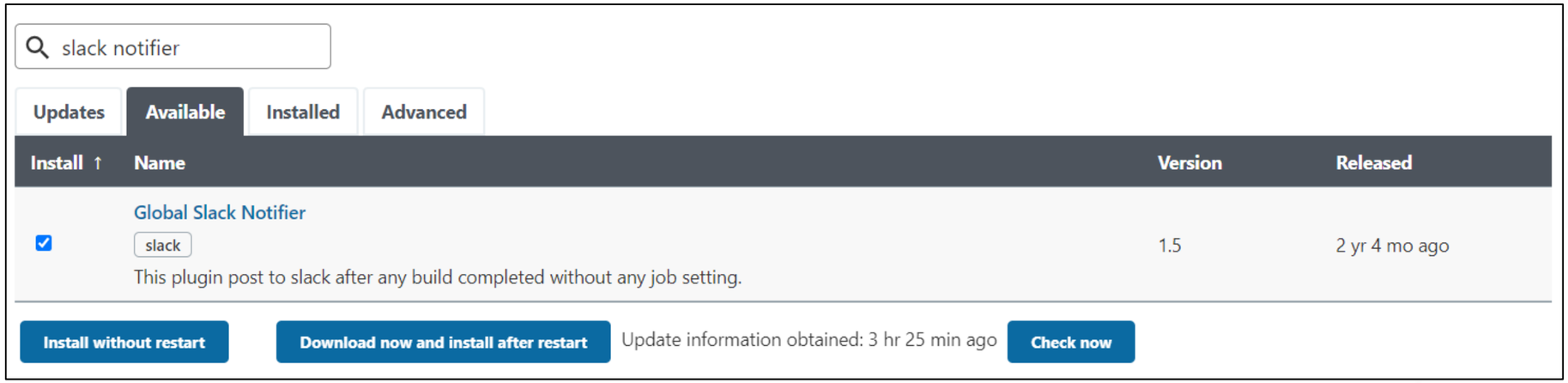
Note

Slack workspace must be configured before configuring Slack in the Jenkins CI server.

Integration of Jenkins and Slack

To integrate Slack with Jenkins, the Global Slack Notifier plugin must be installed. Follow the steps below to install the plugin and integrate Slack:

Step 1: Login to Jenkins; click **Manage Plugins**; install **Global Slack Notifier** plugin



Integration of Jenkins and Slack

Step 2: Click **Manage Plugins**; select **Configure System**; select **Global Slack Notifier settings** and add the workspace and Integration token credential ID

Note

Once the plugin is configured with Slack, an integration messaged is pushed to the configured channel in Slack workspace.

Integration of Jenkins and Slack

Step 3: Configure Slack alerts in Jenkins job **Post-build Actions** or **Jenkins Pipeline** script

Note

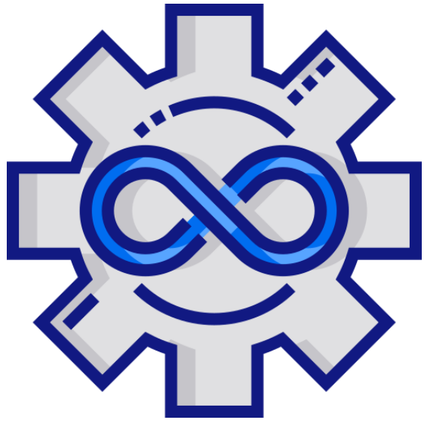
The events that require alerts can be configured. These alerts will be fast and efficient when compared to email notifications.

Continuous Deployment Using Jenkins

Continuous Deployment

Continuous Deployment is an important component of CI/CD Pipelines.

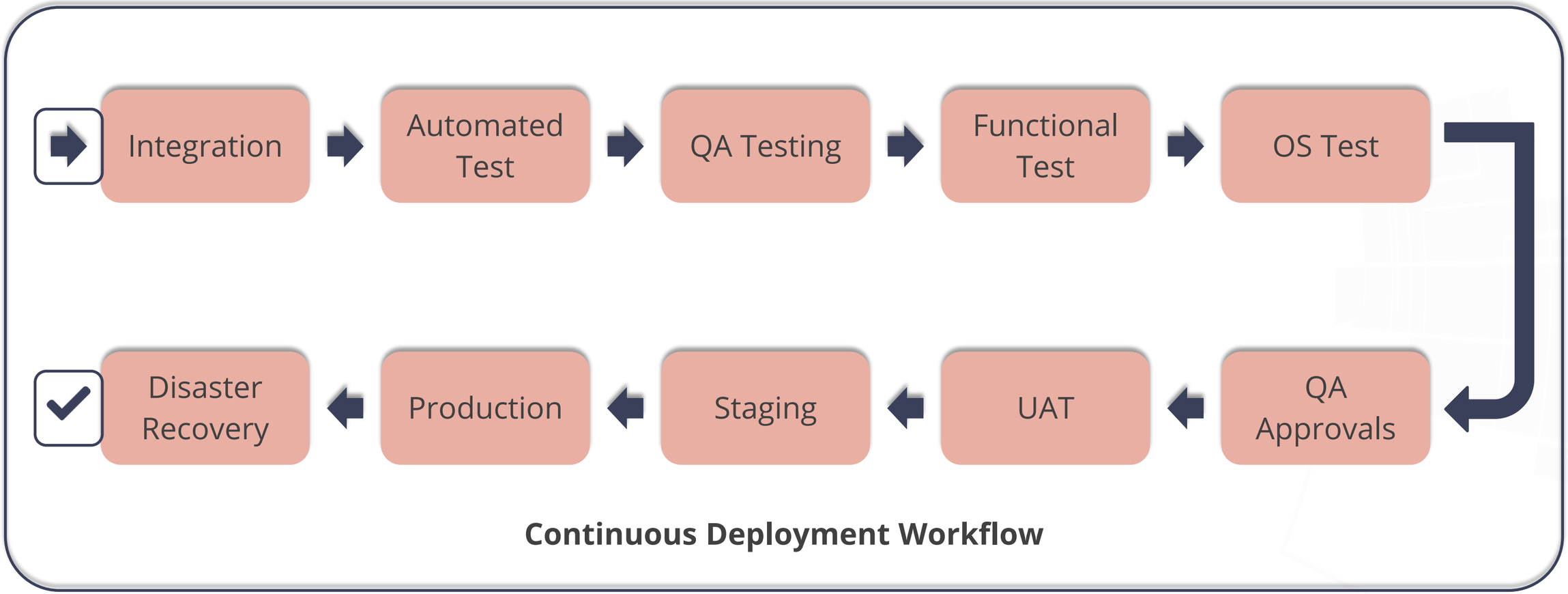
Software release process can be automated using Jenkins Pipeline. This is achieved by performing automated deployment to remote servers.



Jenkins Pipeline supports both automated and manual approvals for deciding whether or not the deployment needs to be performed.

Continuous Deployment Using Jenkins Pipeline

The Continuous Deployment workflow using Jenkins Pipeline is as shown below:



Continuous Deployment Workflow

Continuous Deployment Using Jenkins Pipeline

Continuous Deployment is composed of:

Continuous Integration

Automate the build process and unit test cases; improve the overall source code quality

Automated Test

Improves overall testing process; ensures more test cases are executed in less time

Load Test

Determines the performance of the application; simulates multiple users accessing the application concurrently

Continuous Deployment Using Jenkins Pipeline

Functional Test

Tests the features or the functionality of the application

OS Test

Includes operating system validation; application will be tested across multiple platforms

QA Approvals

Ensure that the software is ready for deployment

Continuous Deployment Using Jenkins Pipeline

User Acceptance Test (UAT)

Tests are performed by the actual users to determine if the application works as expected

Staging

Is an environment where deployment can be tested out before it is deployed on production

Production and Disaster Recovery

Is a live environment where end users access the application and provided failover in case of any issues

Install Deploy Plugin for Continuous Deployment

Jenkins provides several deployment plugins that can be used to implement Continuous Deployment.

Some of the available plugins are:

Deploy to Container

- Deployment plugin for Tomcat, JBoss, and Glassfish webservers
- Supports **Tomcat 4.x/5.x/6.x/7.x/8.x/9.x**, **JBoss 3.x/4.x/5.x/6.x/7.x**, and **Glassfish 2.x/3.x/4.x**

Deploy WebLogic

- Deployment plugin for WebLogic webserver

Deploy WebSphere

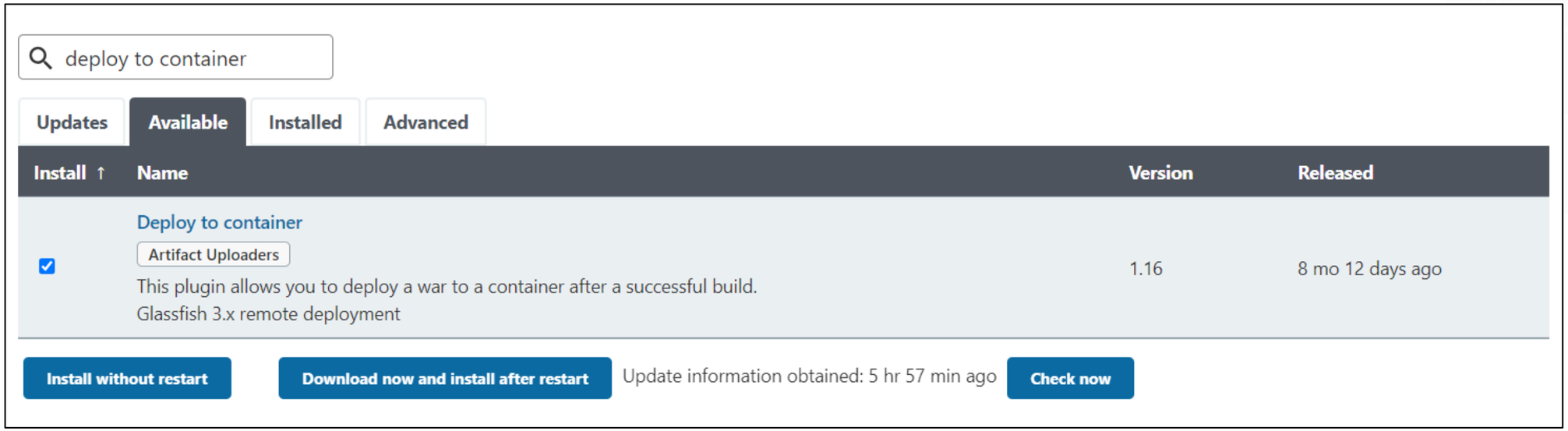
- Deployment plugin for WebSphere webserver

Deploy to Container Plugin

The Deploy to Container plugin deploys a .war or .ear file to a remote application server at the end of a Build. Note that the application server must be up and running.

To download and install Deploy to Container plugin:

Login to Jenkins; click Manage Plugins; look for Deploy to Container plugin in the Available tab; download and install the plugin



Deploy to Container Plugin

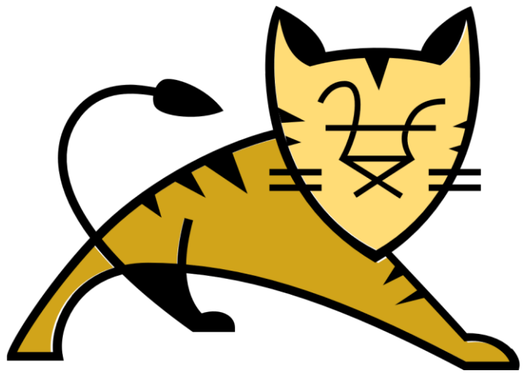
The Deploy to Container plugin also supports redeployment of a previous Build or rolling back to a previous Build. To rollback to a previous Build or redeploy a previous Build, follow the steps given below:

1. Install the Copy Artifact plugin
2. Create a new job that will be manually triggered
3. Configure the job with a Build parameter: Build selector for Copy Artifact
4. Select the Build to be rolled back or redeployed
5. Include a post-build action for deployment

Apache Tomcat

Apache Tomcat is a widely implemented, open-source implementation of Java Server Pages, Java Servlets, and Java web applications.

Working of Apache Tomcat



- Tomcat listens to client requests.
- It loads the respective servlet class using servlet mapping.
- It executes the servlet class.
- It unloads the servlet class.

Image Source: ™/®The Apache Software Foundation, Apache License 2.0

Apache Tomcat: Setting up

Mentioned below are the steps to install and configure Apache Tomcat :

Step 1

Update and upgrade the system

Demo-1

```
apt update && apt –y upgrade
```

Apache Tomcat: Setting up

Step 2

Install OpenJDK (Prerequisite)

Demo-1

```
apt-get install openjdk-11-jdk
```

Apache Tomcat: Setting up

Step 3

Install Apache Tomcat 9 and related packages

Demo-1

```
apt-get install tomcat9 tomcat9-docs tomcat9-examples tomcat9-admin
```

Apache Tomcat: Setting up

Step 4

Start Apache Tomcat service

Demo-1

```
service tomcat9 status
service tomcat9 restart
```

Apache Tomcat: Setting up

Step 5

Configure admin user in /var/lib/tomcat9/conf/tomcat-users.xml

Demo-1

```
<user username="username" password="password"
roles="manager-gui,admin-gui"/>
```

Apache Tomcat: Setting up

Step 6

Restart Apache Tomcat

Demo-1

```
service tomcat9 restart
```

Configure Maven Jenkins Job for Deploying Apache Tomcat

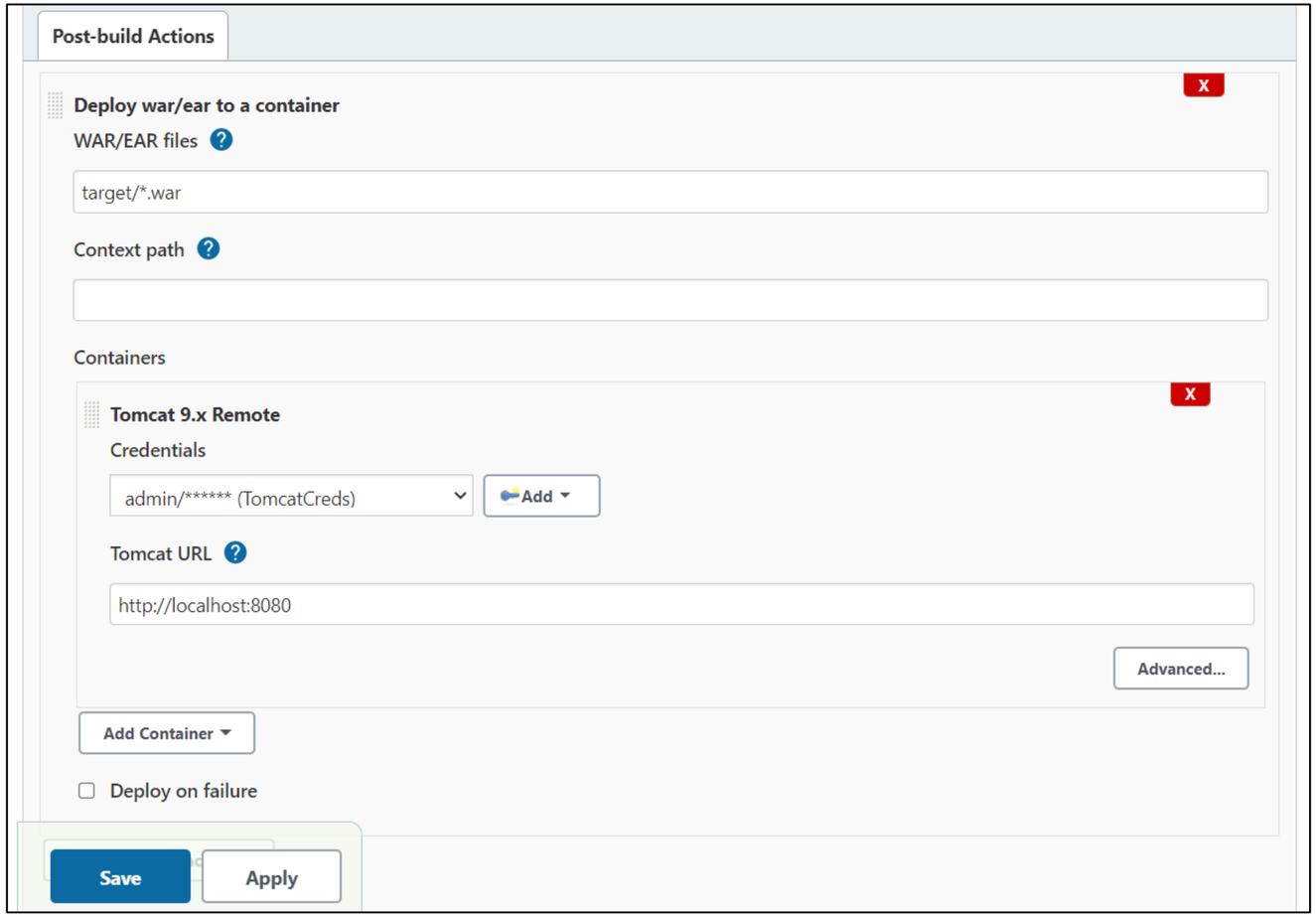
Maven build tool helps in automating the build process and generate build artifacts. To configure a Maven Jenkins job and deploy the build artifacts:

1. Maven Integration plugin that provides Maven Jenkins job must be installed

2. Maven build that generates all the required build artifacts must be run before the deployment is triggered

Configure Maven Jenkins Job for Deploying Apache Tomcat

The screenshot below displays the Post-build Actions tab for deploying Apache Tomcat using Maven tool.



Assisted Practice

Configure the Deploy Plugin for Performing Automated CD

Duration: 20 min

Problem Statement:

Configure the Deploy plugin to perform automated CD.

Assisted Practice: Guidelines

Steps to configure the Deploy plugin:

1. Install Tomcat Apache 9 on Ubuntu VM.
2. Log in to Jenkins CI tool and install Deploy to Container plugin.
3. Configure Deployment stage in Jenkins's pipeline to deploy the Build to Apache Tomcat.

Code Scanning Tools

Code Scanning Tools

Code scanning tools analyze code and share detailed reports with the developers. They are also known as **Static Application Security Testing** (SAST) tools.

Code Scanning Tools

- Can be easily integrated with IDE
- Can be integrated with Continuous Integration process
- Can check code vulnerabilities and coding standards

Benefits of Code Scanning Tools

The benefits offered by code scanning tools include:

- Provide automated code quality mechanism
- Help detect bugs and code issues
- Get integrated with the IDE easily
- Provide fast feedback to developers

Scanning Tools

Some of the popular code scanning tools are listed below:

1. SonarQube
2. Code Scan
3. Veracode
4. Codacy
5. Fortify
6. Coverity
7. PMD

Code Scanning Metrics

Code scanning metrics help decide the overall quality of the source code. Using these metrics, developers can decide whether to proceed with the release or make enhancements.

Listed below are a few code scanning metrics:

- Cyclomatic Complexity
- Maintainability Index
- Depth of Inheritance and Class Coupling
- Lines of Code

Assisted Practice

Integration with SonarQube

Duration: 20 min

Problem Statement:

Integrate Jenkins with SonarQube to scan source code vulnerabilities.

Assisted Practice: Guidelines

Steps to Integrate Jenkins with SonarQube:

1. Install SonarQube on Ubuntu VM.
2. Log in to Jenkins CI tool and install SonarQube plugin.
3. Configure Artifactory in Jenkins.
4. Enter SonarQube server details.
5. Create a Jenkins Freestyle job to integrate SonarQube scan.
6. Include an additional Build step – Execute SonarQube Scanner to scan the source code.

Key Takeaways

- Artifactory is a distributed repository management tool that can store binary artifacts and share them using remote repositories.
- JFrog Artifactory is a universal artifact repository manager that supports all major packaging formats.
- The Jenkins Artifactory plugin helps developers push artifacts to Artifactory while implementing Continuous Integration.
- The Slack collaboration tool can be used by Jenkins to send notifications from all the jobs to the team.
- Jenkins provides several deployment plugins that can be used to implement Continuous Deployment including Deploy to Container, Deploy WebLogic, and Deploy WebSphere.

Lesson-End Project Create a Jenkins Job to Download Artifacts from Artifactory

Problem Statement:

Perform the following:

- Create an automated deployment Jenkins job.
- Download artifacts from Artifactory.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

Thank You