Metapercept

JFrog user guide

Table of Contents

[Chapter 1-Introduction 0](#_Toc176299323)

[Purpose 0](#_Toc176299324)

[Intended Audience 0](#_Toc176299325)

[Prerequisites 0](#_Toc176299326)

[Techincal requirement: 0](#_Toc176299327)

[Knowledge Requirements: 1](#_Toc176299328)

[What is Artisfactory? 1](#_Toc176299329)

[What is JFrog Artifactory? 1](#_Toc176299330)

[JFrog Artifactory Responsibilities 1](#_Toc176299331)

[Key Feature 2](#_Toc176299332)

[Chapter 2-Setting up JFrog Account 3](#_Toc176299333)

[Create JFrog Trail account 3](#_Toc176299334)

[Set up JFrog environment 4](#_Toc176299335)

[User interface 5](#_Toc176299336)

[Work Flow 7](#_Toc176299337)

[Chapter 3-Build 10](#_Toc176299338)

[What is Build? 10](#_Toc176299339)

[Build is a cycle of converting source code into standalone executable programs that run on a computer. Build is a key step in the Continuous Integration/Continuous Deployment pipeline, enabling automation of the software delivery. 10](#_Toc176299340)

[Metadata 10](#_Toc176299341)

[Managing and monitoring build 11](#_Toc176299342)

[Build Dashboard 12](#_Toc176299343)

[Chapter4 -Artifacts 13](#_Toc176299344)

[What is Artifact? 13](#_Toc176299345)

[Artifact Repository 13](#_Toc176299346)

[Repository Types 14](#_Toc176299347)

[Artifacts management 0](#_Toc176299348)

[Browsing artifactory 1](#_Toc176299349)

[Deploy artifacts 5](#_Toc176299350)

[View artifacts information 6](#_Toc176299351)

[Conclusion 7](#_Toc176299352)

# Chapter 1-Introduction

The Introduction section provides an overview of the Jfrog Application user guide.This section includes Purpose , Intended Audience, and Prerequisites to help readers understand the application more effectively. JFrog plays an important role in managing and automating the Software Development Life Cycle (SDLC). It can be done with, Continuous Integration and Continuous Deployment (CI/CD).

## Purpose

The purpose of this user guide is to provide an overview of Jfrog. It also helps in installing the setup, configuring it, and understanding artifacts and builds

By following these guidelines, users can:

1.Install and setup the Jfrog web application.

2. Manage and store build artifacts using the JFrog Artifactory

## Intended Audience

This guide is intended for DevOps engineers, System administrators, and developers for the efficient and functional management and automation of the Software Development Life Cycle.

## Prerequisites

Follow these pre-requisites to ensure a smooth running of the web application.

### Technical requirements:

1. Java Runtime Environment (JRE) version 11 or higher
2. OpenJDK
3. 2GB RAM
4. Administrative access for the JFrog platform.
5. Internet access to download dependency.
6. Access to cloud environment preferably AWS.

### Knowledge Requirements:

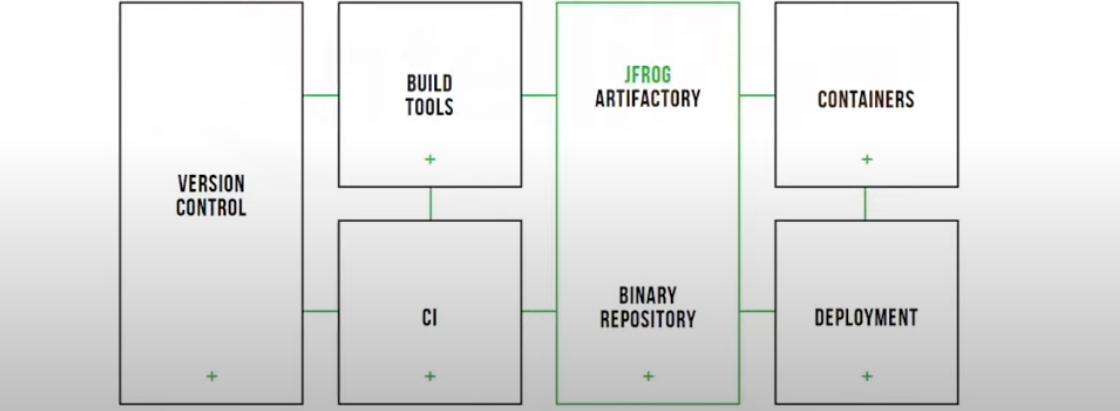
Users require a basic understanding of GIT, CI/CD pipeline and packages such as Docker.

# What is Artifactory?

Artifactory is a global binary repository manager where users can store all of the dependencies and artifacts.

## What is JFrog Artifactory?

JFrog is used in DevOps methodology to store the artifacts and sort them according to the requirements. JFrog Artifactory acts as a Google Drive for Artifacts. Users can store all the artifacts and choose whichever is required to perform the task.



From the above workflow, Jfrog can be integrated with the code pipeline. The version control will pass the main source code to build tools.Then. process it and create artifacts’ all this artifactory will be stored in Jfrog articatory.then

JFrog Artifactory Offers Divers Products For DevOps DevSecOps Operations.

From the above workflow, the code pipeline integrates JFrog by:

1. Passing the main source code through the version control to build tools.
2. Build Tools will create a build.
3. The Build processes the code and creates the Artifacts.
4. The JFrog artifactory stores the processed Artifacts.

Note: Jfrog will send the requested artifact required for a deployment.

## JFrog Artifactory Responsibilities

1. The JFrog Artifactory stores all the artifacts ready for deployment.
2. Jfrog helps manage and download dependencies.
3. Jfrog acts as a proxy for downloading and managing the required dependencies. śśJFrog provides the required dependencies from the remote repository.

Note: Users shall store it in a local server used by the Build Tool. to build the source code.

# Key Features

Here are the key features of JFrog:

* It is a Universal Artifact Repository
* It supports CI/CD Integration
* JFrog is equipped with Security and Compliance
* It helps with Artifact Management and Distribution

It is highly available and scalable.

* It provides DevOps Insights and Analytics
* JFrog supports Automation and API Integration

# Chapter 2-Setting up JFrog Account

This chapter helps users create and set up a JFrog account.

## Create JFrog Trail account

To create a JFrog Trail account.

**Step 1**: Open Your Web browser

**Step 2**:Enter the URL<https://jfrog.com/artifactory> to Navigate the JFrog Artifactory website.

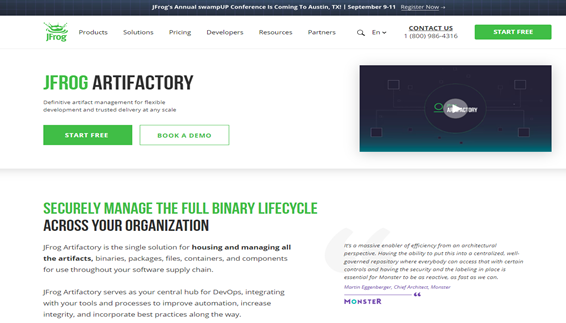
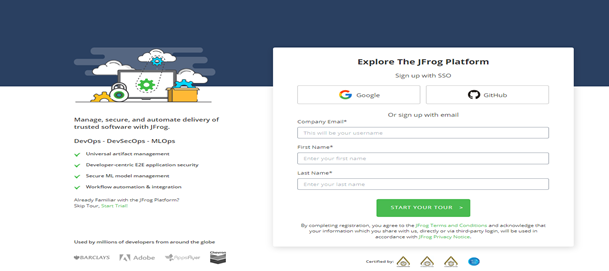


Figure 0‑1: JFrog Artifactory web application

**Step 2**: Click **Start Free** .

**Step output**: You will be re-directed to the Sign-up page.



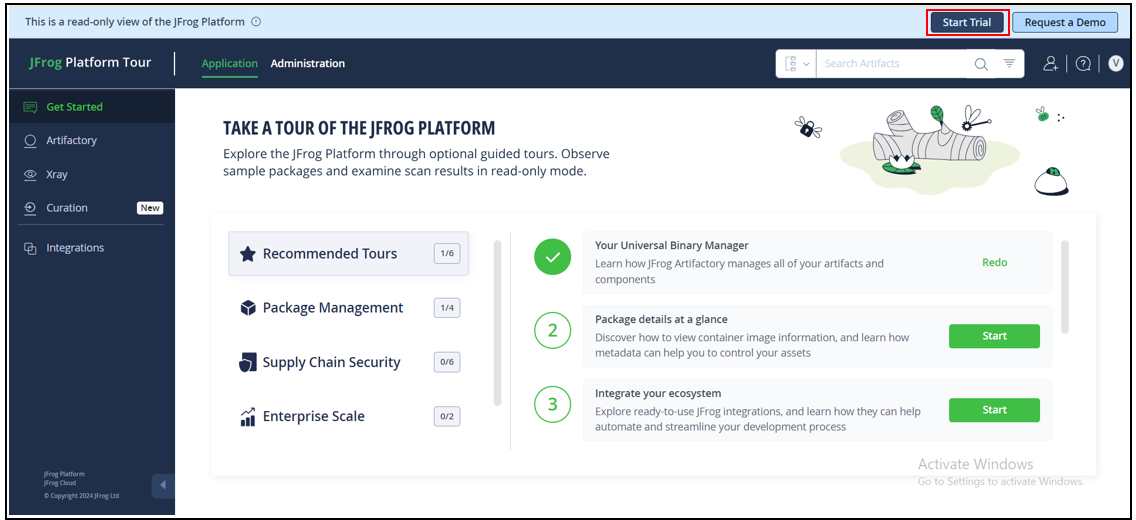
**Step 3**: Sign up on Google or GitHub or sign up with email.

**Step 4**: Click **Start your Tour**

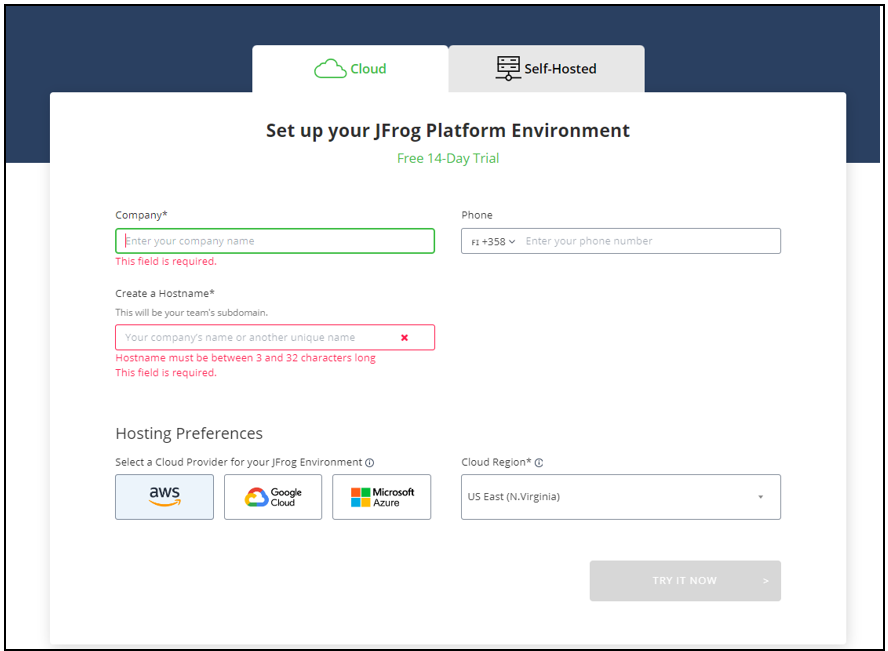
## Set up JFrog environment

This section helps you to set up the JFrog environment.

**Step 1**: Click Start Trail.



**Step 2**: Select a cloud or self-hosted environment.



**Step 3**: Enter Your details (Company, Phone, Hostname)

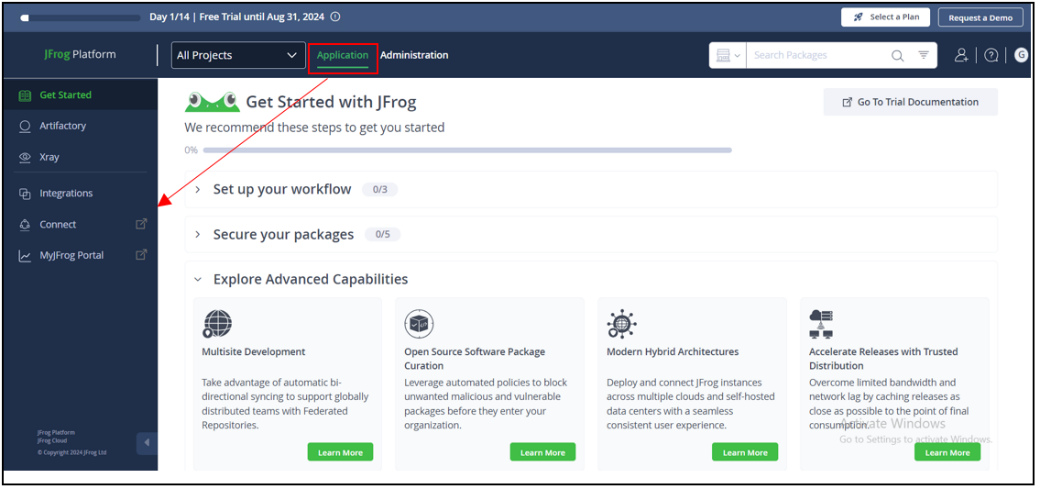
**Step 4**: Select Hosting Preferences.

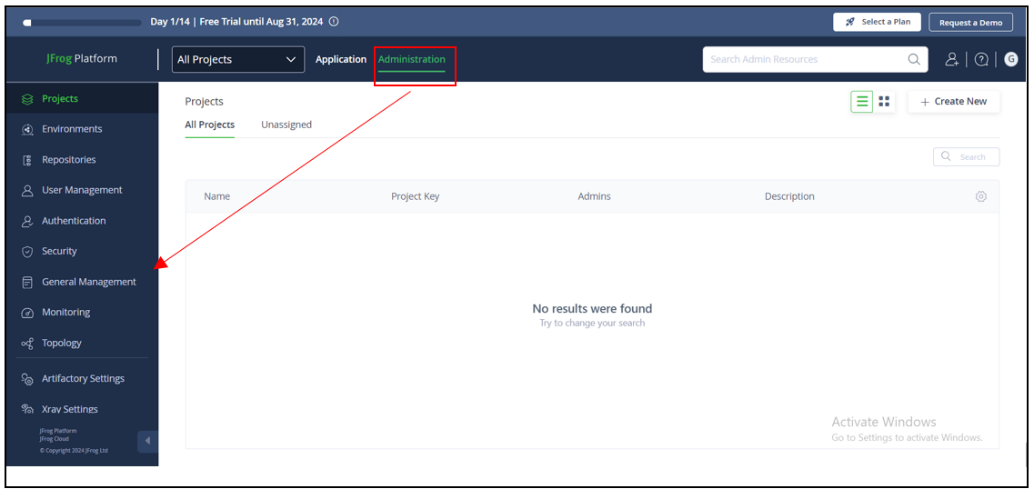
**Step 5**: Select the cloud region.

**Step 6**: Click TRY IT NOW.

## User interface

This section helps to understand the overview of the JFrog platform user interface.





|  |  |
| --- | --- |
| **Field** | **Description** |
| Application (Sidebar ) | This functionality allows users to   1. Create and manage a repository. 2. Manage Universal package for Builds. 3. Manage all artifacts in one place. 4. Integrate with CI/CD tools such as Jenkins Git lab and Git Hub. 5. Keep all of the artifacts in the JFrog Artifactory scanned by JFrog X-ray. |
| Administration (Sidebar) | This functionality allows users to   1. Create new projects and assign responsibilities 2. Create environments for repositories 3. Set restrictions to different environments. 4. Monitor tab tracks progress of the environment. |
| Search panel | This panel enables users to perform quick searches for builds, packages, and artifacts. |
| All projects | This panel allows users to browse different projects and create new projects. |

## Work Flow

A screenshot of a computer diagram

Description automatically generated

This diagram explains the workflow of the JFrog platform and its various components' interaction with each other to support software development and DevOps processes. Here’s an easy-to-understand explanation:

|  |  |
| --- | --- |
| Frontend | The user interface that users can interact with, through a web browser. |
| Router | The Router acts a traffic controller, directing requests to the appropriate JFrog services within the platform. |
| Artifactory | The Artifactory is the main component for managing your software packages and artifacts (like binaries, libraries, etc.). It handles storage, versioning, and retrieval of artifacts |
| Access Manages | The Access manages User Permissions and Security, ensuring that only authorized users can access certain resources |
| Events Handles | The Events Handles notify and automate triggers. |
| Metadata | The Metadata manages additional data associated with artifacts, such as tags or properties. |
| Pipelines | The Pipelines automates CI/CD processes, such as building, testing, and deploying applications |
| X-ray Scans | The X-ray scans artifacts for security vulnerabilities and license compliance issues. |
| Insight | The Insight provides analytics and insights into your CI/CD workflows to help improve efficiency and performance. |
| Distribution | The Distribution manages the distribution of software packages across different environments, ensuring consistent deployments. |
| REST APIs | The REST API / CLI Provides command-line and API access to automate tasks and integrate JFrog with other tools. |
| Users | The Users interact with JFrog through a web browser to access the platform’s features. |
| Build Node | The Build Node Represents the CI/CD servers or build agents that perform the actual software builds and deployments. |
| Load Balancer | The Load Balancer ensures that traffic is efficiently distributed across multiple services, providing high availability and reliability. |

# Chapter 3-Build

This chapter helps users understand Build and various tools that help in the managing and monitoring.

## What is Build?

## Build is a cycle of converting source code into standalone executable programs that run on a computer. Build is a key step in the Continuous Integration/Continuous Deployment pipeline, enabling automation of the software delivery.

## Metadata

It is the information generated during software Build, this includes details about artifacts deployed, dependencies, and information about the Build environment.

## Managing and monitoring Build

In the Build panel, users can manage and monitor Build created during continuous integration/ continuous delivery process.

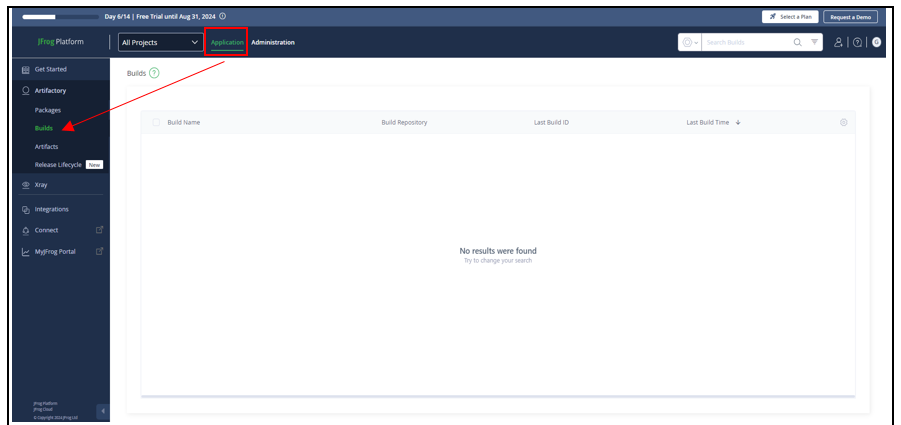
Here is a step by step process on how to navigate the Build tab.

**Step1**: Log in to the JFrog Artifactory.

**Step2**: Click on the Artifactory tab on the sidebar of the application functionality.

**Step outcome**: A drop-down list appears.

**Step3**: Click the Build form drop-down list.



## Build Dashboard

|  |  |
| --- | --- |
| Field | Description |
| Build column | * This column displays specific build information, such as the build name, build ID from the build repository, and the last build time. * The user can customize the column. |
| Search field | The search field helps users quickly find a specific build with a specific date range. |
| Get help and learn | This tab on the top left-hand side allows users to access the JFrog documentation for the build section. |

# Chapter4 -Artifacts

This chapter helps users understand artifacts and various tools that help in the managing and monitoring.

## What is Artifact?

The file that contains both the binary code and also the resources needed to compile the code is known as an artifact. For example, in Java .jar files are the artifacts used to compile and run code..

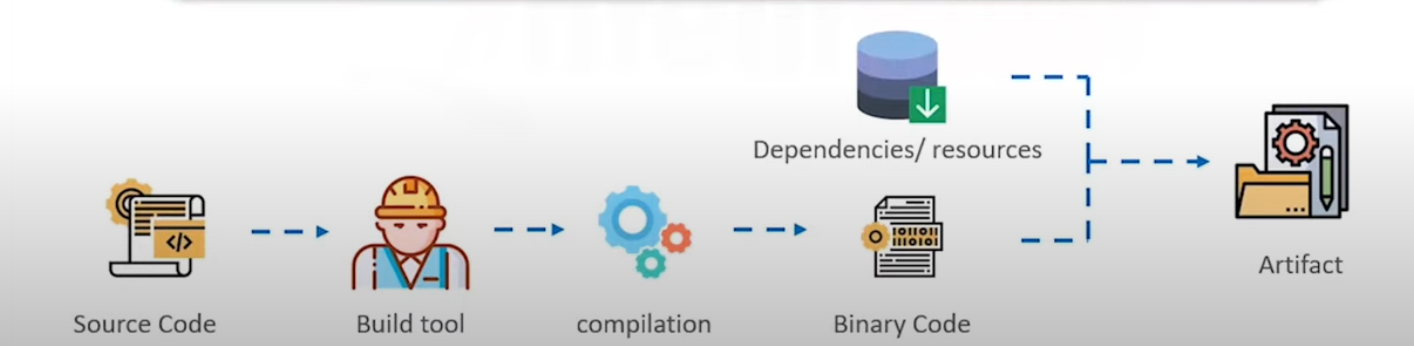


Figure 2Artifacts

## Artifact Repository

The repository that stores multiple versions of artifacts in it is known as an artifact repository. Each time a new artifact is created, it is stored in the server allocated to the repository. This helps in software development and allows for easy rollbacks and deployments

A computer network diagram with arrows and a folder

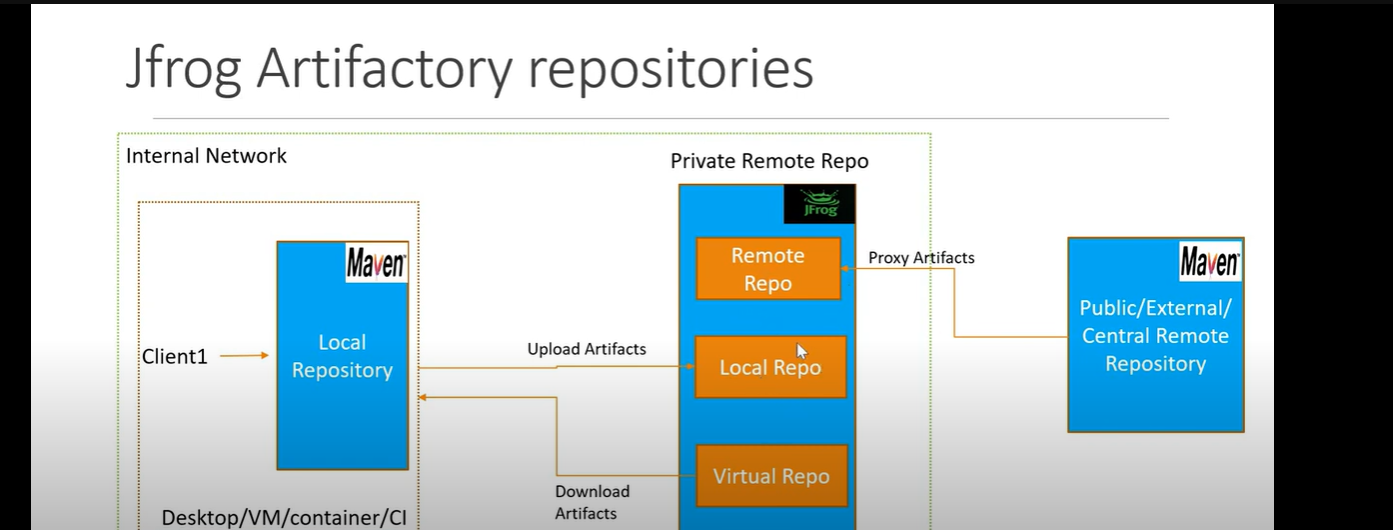
Description automatically generated

Figure 3 Artifact Repository

As shown in the above figure, it is very easy to make changes. To follow the Agile module for code deployment, deploy data from the Artifact versions 2 and 3 easily.

## Repository Types

There are various kinds of repositories with distinct functions.



**Local repository:**

Artifacts are stored internally. These can be shared with teams that are located in remote locations.

**Remote repository:**

Artifacts are stored in a remote location it serves as a cache and proxy artifacts from external repositories.

**Virtual repository:**

It aggregates local, remote and other repositories and allows access through a single URL.

## Artifacts management

In the artifacts panel, users can manage and monitor artifacts created during continuous integration/ continuous delivery process.

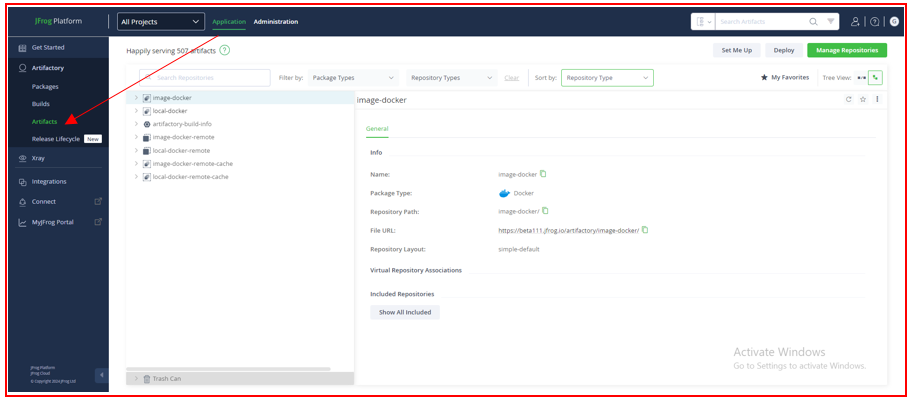
Here is step by step process on how to navigate Artifacts tab.

**Step1**: Log in to the JFrog Artifactory.

**Step2**: Click the Artifactory tab on the sidebar of the application functionality.

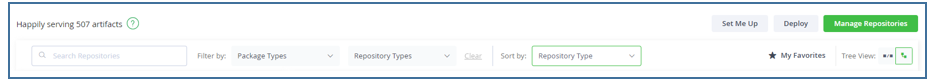
**Step outcome**: A drop-down list appears.

**Step3**: Click the Artifacts form drop-down lists.



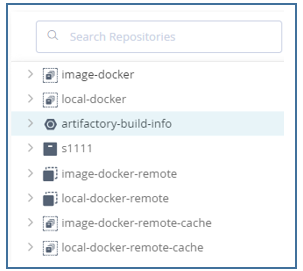
## Browsing artifactory

This tab helps users navigate the repository in JFrog Artifactory. This section takes you through how to browse artifacts and apply filters for quick searches.



**Simple browsing**

This search field allows the user to search a specific repository by name.



**Apply filter**

This section helps users apply filters to quickly navigate artifacts with specific criteria.

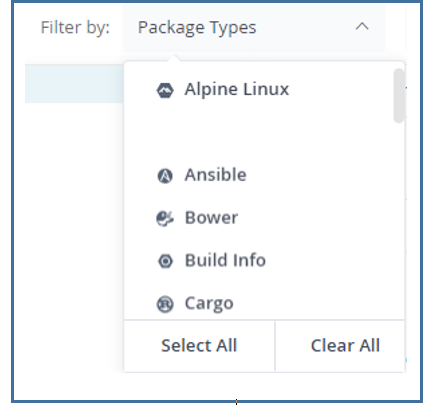
**Package type**

This tab allows users to filter artifacts based on package types. The user can select one or multiple package types.

**Step1**: Click the package type tab.

**Step outcome**: a drop-down list appears.

**Step2**: Select package type.



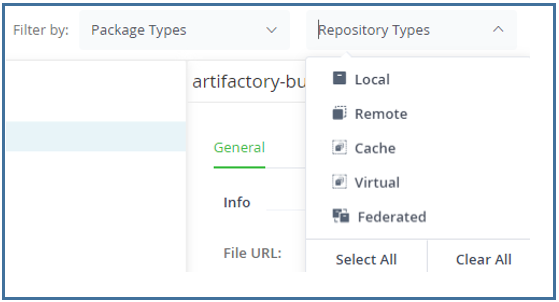
**Repository type**

This tab allows users to filter artifacts based on the repository type; user can select single or multiple repository types.

**Step1**: Click repository type tab.

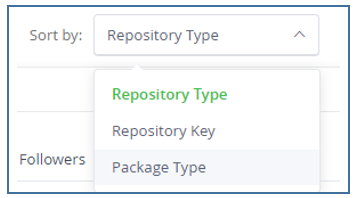
**Step outcome**: A drop-down list appears.

**Step2**: Select Repository Type.



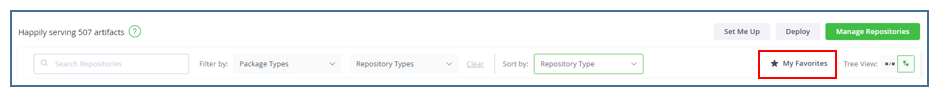
**Sort repository**

This tab allows users to sort the repository by their type.



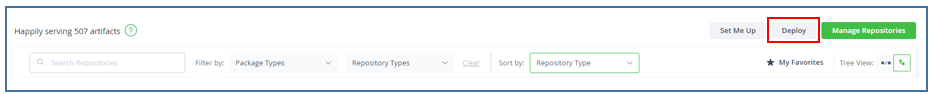
**Favorites repository**

This tab helps users to browse a repository that are marked as favorites.



## Deploy artifacts

This tab allows the users to deploy artifacts into a local repository. Artifacts can be deployed individually or in multiple.



**Step1**: Log in to the JFrog Artifactory.

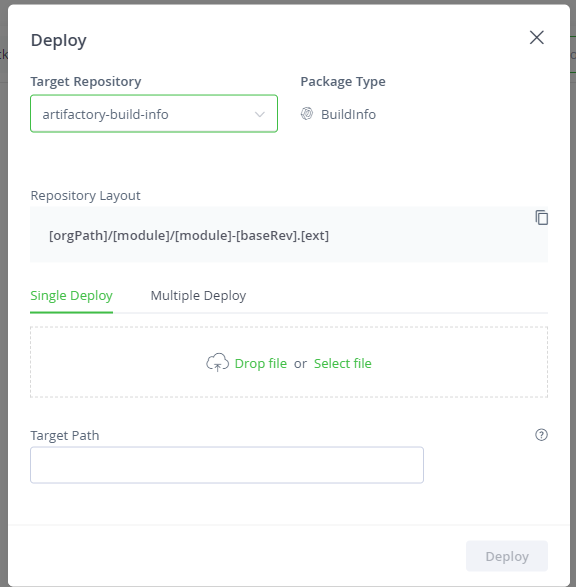
**Step2**: Click the Artifactory tab on the sidebar of the application functionality.

**Step outcome**: A drop-down list appears.

**Step3**: Click the Artifacts form drop-down list.

**Step4**: Click Deploy on the right hand of the browsing tab.

**Step5**: Deploy window appears.

  
**Step6**: Select the target repository for storing artifacts.

**Step7**: Drop a file else you can select file form device.

**Step8**: Specify the target path for repositories.

**Step9**: Click Deploy.

## View artifacts information

This tab allows users to monitor and manage information related to Artifacts.

|  |  |
| --- | --- |
| **Field** | **Description** |
| General tab | This tab shows comprehensive information about the artifacts, including the build name, when the builds were created, and the updated package type. |
| Effective permission tab | * This tab helps the user understand and manage the permissions applied to specific artifacts or sets of artifacts. * It displays a list of permissions for a user or group. |
| Properties tab | * This tab allows users to manage metadata associated with artifacts. This metadata is organized into key-value pairs known as "properties." * The user can add a new property to an artifacts from this property tab. Enter property name and property value. |
| Follower tab | * This tab allows the user to monitor artifacts folders or repository * User can add or remove follower from this tab.   C:\Users\satyam\Documents\My Projects\JFrog user guide 2\Content\Resources\Images\image30.png |

# Conclusion

We have looked at JFrog's advantages as a complete artifact management platform throughout this handbook. JFrog provides a reliable solution for automating deployment pipelines and maintaining binary repositories for companies of all sizes. It increases collaboration, expedites time-to-market, and improves development workflow advantages.

We invite you to begin experimenting with JFrog to see what exciting changes it can bring about in your software delivery process."