



# 12 REASONS

to use a Binary Repository Manager  
When Developing with Java



WHITE PAPER

# Introduction

The modern Java developer faces an accelerated and technologically diverse world. Open source and commercial components supplement in-house code to speed app development. Tools that automate build processes such as Maven, Gradle, and Ivy further encourage the usage of these components. A 2018 analysis found that 92 percent of all apps use open source components, which are typically the greater part of the delivered app.

Organizations that once issued a significant new release every year or twice a year now more commonly release a new version to customers monthly or weekly. Many enterprises now deploy updates to production multiple times a day.

Developers are adopting **DevOps** -- the tools and best practices that enable creating and deploying more releases, more frequently, and at high quality into production environments. DevOps practices aim to reduce friction at all points in the software development lifecycle, or SDLC, and automate workflows (through CI and CD servers and more) to greatly accelerate release velocity.

To succeed, today's Java developers need to face many challenges:

- A variety of package types and build tools
- Sharing internal and external artifacts with metadata
- External components retrieved through unreliable or speed-limited networks
- The need to produce and manage containerized applications
- Ensuring security through access control
- Binary version tracking to reproduce builds
- Searching for artifacts based on different criteria
- Integrating with the development ecosystem

We'll show the top 12 ways that Artifactory, the [binary repository manager](#) at the heart of the JFrog DevOps Platform, gives Java developer teams the tools to address these issues and more, boosting your organization's productivity by helping ensure a smooth and secure application development workflow from end to end.

# 1

## Maintain a single source of truth

As the power behind the JFrog Platform, Artifactory is a universal repository manager for all of your artifacts: dependencies, binaries, and configuration files.

Artifactory natively supports the package types Java and Kotlin developers depend on, such as Maven, Gradle, and Ivy, keeping all your local package repositories, JAR and WAR builds under one tool. Generic repositories can store your supporting artifacts, such as graphics or ZIP files. And you can containerize your apps for the cloud through private or public Docker registries in Artifactory.

Few organizations develop in Java alone. With native support for over 30 package types, including npm and PyPi, Artifactory is your single source for storing, securing, and tracing the entire artifact ecosystem of your development organization. Developers can store and retrieve their packages or images in Artifactory repositories through the same package management services they use every day.

When all of your binaries are governed through this one common tool, your entire enterprise can align onto the same SDLC workflows and best practices that will assure quality and accelerate release velocity. This is why Artifactory is the central component of a fully-automated software distribution pipeline that powers the JFrog DevOps Platform.

### LOCAL REPOSITORIES

Local repositories are physical, locally-managed repositories into which you can deploy artifacts. Typically these are used to store proprietary packages, internal and external releases as well as development builds. Using local repositories, all of your internal resources can be made available from a single access point across your organization from one common URL.



Alpine



Cargo



Chef



CocoaPods



Conan

CONDA

Conda



Cran



Debian



Docker



Gems



Generic



Git LFS



Go



Gradle



Helm



Ivy



Maven



Npm



NuGet



Opkg



Php Composer



Puppet



PyPI



Rpm



Sbt



Vagrant



Bower

# 2

## Reliable and consistent access to remote artifacts

For Java developers, external repositories are invaluable resources that you need every day. The open source software dependencies drawn from remote resources such as Maven Central, the npm public registry, or Docker Hub can easily be the more significant portion of code in today's applications.

Ensuring site reliability and speedy access is a vital key to maintaining release velocity, but can face several challenges:



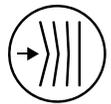
### Network latency

Inherent latency from physical distance - a remote site can be on the other side of the globe.



### Poor connectivity

Network outages, jitter, poor bandwidth, unstable connections.



### Heavy loads

Delays produced by heavy demand for a service.



### Site downtime

The remote site where your dependencies are stored may suffer a service disruption from failure, attack, or loss of service.

Artifactory is an intermediary between developers and external resources. As a developer, all of your requests are directed to Artifactory for quick and consistent access to remote artifacts that are cached locally in a remote repository. Your development efforts won't be hampered by network issues or a repository going down.

## REMOTE REPOSITORIES

A remote repository serves as a caching proxy for a repository managed at a remote site such as Maven Central. Artifacts are stored and updated in remote repositories according to various configuration parameters that control the caching and proxying behavior.

# 3

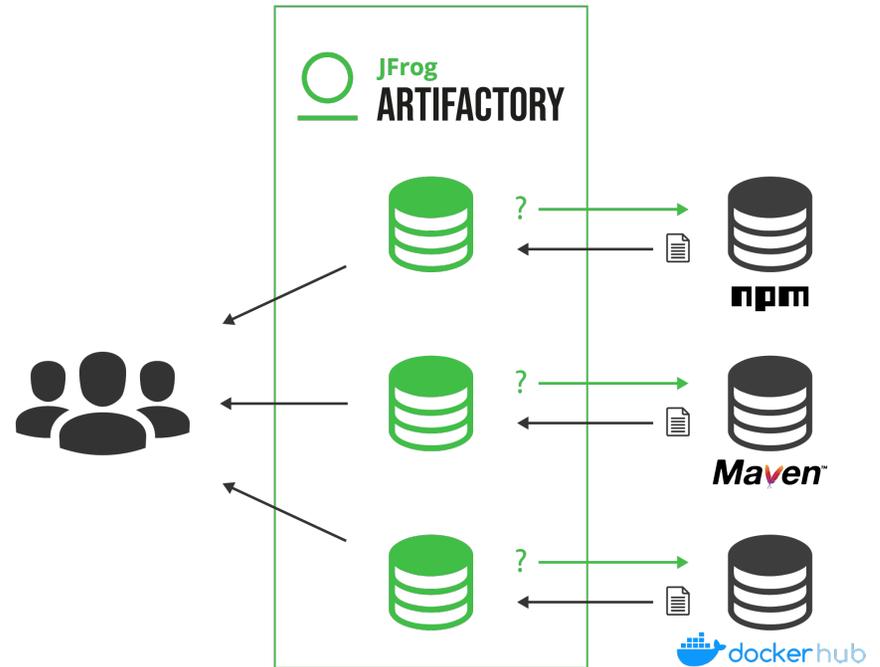
## Reduce network traffic and optimize builds

Since much of your code is likely to be assembled rather than built, you want to make sure that your access to artifacts downloaded from external resources is optimized. It makes no sense for two (or two hundred) developers using the same package from Maven Central or base image from DockerHub to download it separately.

Remote repositories reduce external networking. Once an artifact has been downloaded to an Artifactory remote repository, it is then locally available to all other developers in the organization (thus reducing network traffic). This is all transparent to the individual developer when accessing artifacts through Artifactory.

Moreover, pulling artifacts from a local cache helps guarantee that identified versions are always the same for every build, for everyone, guarding against any subsequent improper or malicious overwrite to the remote resource.

A typical project may depend on dozens, if not hundreds of artifacts from external resources. To build these projects, all remote artifacts must be available to the server environment. Downloading all those required artifacts may generate Gigabytes of data traffic on the network which takes a significant amount of time delaying the build process. By caching remote artifacts locally, the build process is much quicker, incurs much less network congestion, and assures consistency.



# 4

## Combine local and remote resources

Most applications will make use of both proprietary and open-source dependencies from several different sources. Rather than reference each place they can be found, it can be far more convenient to address dependencies through a single URL that treats multiple repositories as a unit.

With an Artifactory virtual repository, you can define a set of local and remote repositories that you can address as a single, logical repository. Each virtual repository behaves like any other single repository in Artifactory, including using fine-grained permission controls to grant different teams or users access to only the virtual resources they will need.

You can also optimize artifact resolution by defining the underlying repository order so that Artifactory will first look through local repositories, then remote repository caches, and only then Artifactory will go through the network and request the artifact directly from the remote resource. For the developer it's simple. Just request the package, and Artifactory will safely and optimally access it according to your organization's policies.

### VIRTUAL REPOSITORIES

A virtual repository encapsulates any number of local and remote repositories and represents them as a unified repository accessed from a single URL. It gives you a way to manage which repositories are accessed by developers since you have the freedom to mix, match and modify the actual repositories included within the virtual repository.

# 5

## Track a software bill of materials (SBOM)

Artifactory stores new metadata -- what we call "build info" -- with every build you make, linking to the package metadata of your open source and proprietary dependencies along with build artifacts and environment settings. With detailed build info, you can trace every build back to where it came from and out to every place it's been staged for service.

Your build info is the basis of a Software Bill of Materials (SBOM) -- a machine-readable inventory detailing all the items included in an application and their origin -- for every release put into production or delivered to a customer.

As a growing number of governments and regulated industries require an SBOM to help combat cyberattacks, the JFrog Platform is your turnkey solution for compliance.

Artifactory's build info helps ensure that you are never in the dark about where a build came from, how it was created, or where it was deployed.

### ARTIFACTORY QUERY LANGUAGE

Artifactory Query Language (AQL) empowers you to uncover any data related to the artifacts and builds stored within Artifactory repositories. Its syntax offers a simple way to formulate complex queries that specify any number of search criteria, filters, sorting options, and output parameters. AQL is exposed as a RESTful API which uses data streaming to provide output data resulting in extremely fast response times and low memory consumption.



#### ARTIFACTORY BUILD INFO

Build Name | Build Version | Date of Creation

#### SOFTWARE BILL OF MATERIALS (SBOM)



Dependencies



Custom Data



Artifacts



Vulnerabilities

#### BUILD HISTORY



Environment



Repositories



Issues Fixed



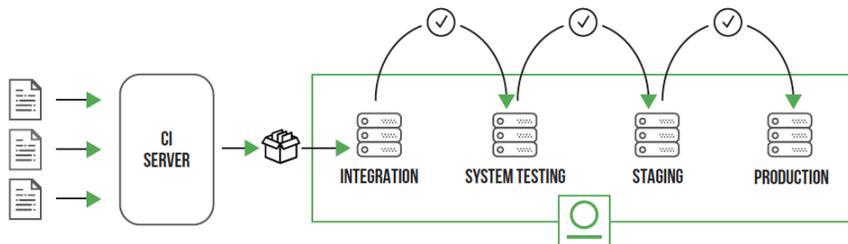
Build Step

# 6

## Build once, then promote

Under the agile methodology of continuous integration, every new software version must pass several quality gates in an SDLC. For example, a Maven “snapshot” build, once elevated to “release,” will be promoted successively to different teams for integration, testing, and staging before it is released for general use.

Artifactory’s binary repository management offers a consistent and reliable promotion method that carries a single, immutable binary through the entire SDLC. With a repository for each SDLC stage, a build with its metadata can be promoted in the JFrog Platform simply by shifting it to the next repo in sequence.



In this “build once and promote” approach, the same build is evaluated at every stage, assuring absolute consistency through the DevOps pipeline.

At each stage, teams can add additional metadata about stability, security, and more to the candidate’s build info. By passing critical metadata learned from one stage that can be used by the next, the JFrog Platform accumulates a comprehensive record about the lineage of every build put into production.

### BUILDS WITH MAVEN

Building and promoting Maven apps is easy with the JFrog command-line-interface (CLI):

```
jfrog rt config # configures JFrog CLI with the integration listed in the yaml
jfrog rt mvn-config # if configFileLocation are not set in yaml
jfrog rt mvn $mvnCommand # the main maven build command
add_run_variables # saves some information in run state for future steps to reference
jfrog rt build-collect-env # collects build environment, preparing for build publish
jfrog rt build-publish # only if autoPublishBuildInfo is true
write_output # updates the buildinfo resource, if present
jfrog rt build-scan # if forceXrayScan is true
add_run_files # adds BuildInfo to run state
```

### CHECKSUM-BASED STORAGE

Artifactory optimizes storage by ensuring that any binary and its metadata are only stored once on the file system, under the name of its unique calculated checksum. Repositories hold only references to files, so the physical file is never duplicated, and its checksum can be used to verify the binary’s integrity.

# 7

## Accelerate cloud native development

Increasingly, software development is cloud native development: applications written to effectively utilize cloud technology infrastructure, and enable the inherent best characteristics of running in the cloud.

This means producing container-based microservices, relying on established cloud native standards like OCI and cloud native tools like Docker and Kubernetes.

Docker repositories in the JFrog Platform fully support all Docker Registry APIs, so they can function natively with the Docker CLI. With local repositories, you can maintain as many private Docker registries in Artifactory as you need, to distribute and share both Docker and OCI container images within your organization.

Artifactory reveals the layers within every Docker or OCI image that compose it, and links the metadata for a fully traceable path back to the origins of all its parts.

When combined with the JFrog Platform's fine-grained access control, you can maintain secure, private Docker repositories that exceed the security offered by Docker Trusted Registry. Using Artifactory's local repositories instead of private repositories on Docker Hub avoids all internet connectivity concerns, providing reliable and consistent access to images.

Artifactory also supports Helm chart repositories, so you can also manage your Kubernetes orchestration manifests alongside your container images. In this way, the JFrog Platform can serve as your comprehensive Kubernetes registry, a central, traceable home for everything deployed to your clusters.

### ▶ NO LIMITS WITH DOCKER HUB

JFrog's partnership with Docker exempts JFrog Cloud users of Artifactory from Docker Hub's image pull rate limits. Ordinarily, anonymous free users are limited to 100 pulls per six hours, and authenticated free users to 200 pulls per six hours. But Docker Hub waives these limits when the image pull request is from a JFrog Cloud account.

By setting up an Artifactory remote repository to proxy Docker Hub, cloud users gain unlimited, high-performant access to Docker Hub and to Docker Official Images, simplifying cloud native application development.

# 8

## Full integration with your build ecosystem

Integrating a complex DevOps tool stack is critical for development. It starts with the tools that automates your builds: your CI/CD servers must integrate smoothly with the systems where they'll be stored. The JFrog Platform can connect to your choice for CI/CD automation through:



### CLI

A command line interface (CLI) tool enables developers to store and retrieve binaries and metadata to and from repositories through a command window, shell script, or CI pipeline.



### REST APIs

Developers can manage binary repositories using REST commands through curl or custom DevOps tools.



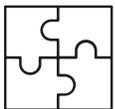
### Webhooks

Trigger an action in another service in response to an event in Artifactory, either to notify users of the event or initiate an automated flow.



### User plugins

Plugins for [Maven](#) and [Gradle](#) enable you to publish artifacts and build info to your Artifactory repositories with a fully traceable build context, and implement custom behaviors in your binary management workflow through an extended Artifactory REST API. These plugins are automatically installed on all JFrog cloud accounts.



### Build Integrations

Ready-made plugins and extensions for major CI tools such as Jenkins, Circle CI, TeamCity, Bitbucket, Pipelines, and Azure DevOps accelerate integration with your CI pipelines.

## PARTNER INTEGRATIONS

CI/CD is only the start of your integration needs. Tools for **automated testing, collaboration, ITSM, observability and analytics** are all part of your DevOps infrastructure fabric. You'll want to push the information these tools produce into your repositories, and connect that information to other tools, too. With a large family of JFrog partner integrations with top industry providers, the JFrog Platform can help quickly build a strong, tight weave of your DevOps tool stack through this single source of truth.

# 9

## Protect your software supply chain

Protecting your software supply chain -- the packages, artifacts and builds of applications -- from mistakes and attacks is critical to keeping the trust and confidence of those who rely on what you deliver.

The growth of supply chain attacks has led the World Economic Forum to rank them among the top cybersecurity challenges. That's why security is a top priority for the JFrog Platform:



### Authentication

All services of the JFrog Platform require authentication through secure credentials such as a password or access token.



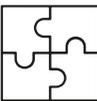
### Checksum Verification

Every artifact's computed checksum is integral to how it is stored in Artifactory repositories, and is used to verify its integrity.



### Permissions Management

With the JFrog Platform's fine-grained access control, administrators can ensure that developers and groups can access only the repositories through the CRUD operations they are authorized to.

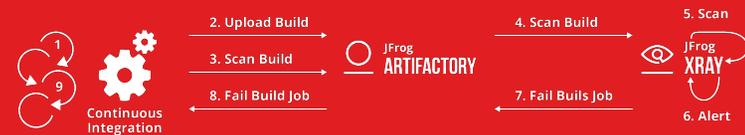


### SSO

Support for LDAP, SAML, OAuth and SCIM protocols empower admins to integrate with Single Sign-On services such as Active Directory, Crowd and others for secure, organization-wide credentials management.

## VULNERABILITY AND LICENSE SCANNING

The companion security solution JFrog Xray performs deep recursive scanning of binaries in your JFrog Platform repositories to identify all open source components that have known vulnerabilities. As part of the JFrog Platform, Xray is tightly integrated with Artifactory, maintaining additional security metadata and providing impact analysis so you can quickly remediate all binaries where a vulnerable dependency has been used.



Xray can also monitor the license types of your open source components, to alert you of all that are out of compliance with your organization's policies.

As your binary repository manager becomes your mission-critical tool, you'll need to ensure it can stretch seamlessly, even as your department grows.

These binaries-centric best practices for DevOps - a single source of truth, metadata, build promotion, security, etc. - have been proven to scale smoothly. Every day, they enable JFrog Platform users to release quality software at speed whether it's developed by 5 people in one room, or 500 around the globe.

Continuous and responsive access is paramount, and Artifactory can be configured in self-managed systems for High Availability, using redundant, load-balanced instances to keep delays down.

You can also subscribe to a JFrog cloud managed service hosted in the major cloud provider of your choice (AWS, GCP, or Azure), and leverage the "any time, from anywhere" availability and elasticity of the cloud. Or you can rely on the JFrog Platform's "same here, same there" promise of feature equivalency in every environment for a hybrid approach that divides your workloads between cloud and on-prem as you need.

## HIGH AVAILABILITY SYSTEMS

A High Availability (HA) configuration of replicated nodes in your self-managed cluster helps spread the load to accommodate large load bursts and ensures there is no single point of failure. This maximizes your uptime, even during most maintenance tasks, up to "five nines" level of availability.

## MULTI-CLOUD AND HYBRID

A multi-cloud strategy of maintaining multiple SaaS accounts hosted on different cloud providers can help avoid vendor lock-in, and enables allocating workloads to the most cost-effective provider.

A hybrid strategy combines self-managed servers behind your own firewall with a managed service in the cloud. Employ the scalability of the cloud for your dynamic workloads, while keeping sensitive workloads within your secure, on-premises data center.

The days when applications were created by a small team of developers in one room are long past. Enterprise software development is now a highly collaborative endeavour of packages shared by intersecting teams across multiple sites spread across the globe.

Artifactory enables multi-site geo-replication through a variety of push/pull replication topology options, empowering geographically distributed teams to work on the same artifacts (binaries and their metadata), with minimal latency through local repositories.

You can replicate your repositories to another instance of Artifactory located elsewhere, whether in an on-premises datacenter at another site, or in the cloud. Replicated repositories are automatically synchronized with their source periodically so that your artifacts can be made available to different teams wherever they may be located around the world.

JFrog's unique set of multi-site capabilities ensure locality in any global network topology. Developer teams located across the world can work on the same artifacts (binaries and their metadata), with minimal latency so that every build, at every site, can complete fast, without fail.

## FEDERATED REPOSITORIES

JFrog Artifactory Federated Repositories are an innovative, bidirectional mirroring technology that provides DevOps teams an option that's easy to set up and maintain for multi-site teams and projects. Through federation, a local repository in one Artifactory deployment is logically joined to synchronize with a local repository in an Artifactory deployment elsewhere. Artifactory repositories joined through federation provide each of these sites a unified, locally accessible repository of shared global data.

# A universal, end-to-end solution for all binaries

Artifactory is the core component that powers the JFrog DevOps Platform, a comprehensive end-to-end platform solution for one-stop DevOps.

For Java developers, along with the other technologies they use, the key to successful DevOps -- minimizing or eliminating the friction in your pipelines -- lies in effective management of your binaries through your entire SDLC. A binary repository manager is your key agent of digital transformation.

The reasons we've given here are the essential practices of this binaries-centered approach to DevOps. They're also the core values that the JFrog Platform is built around, to better assure your success.

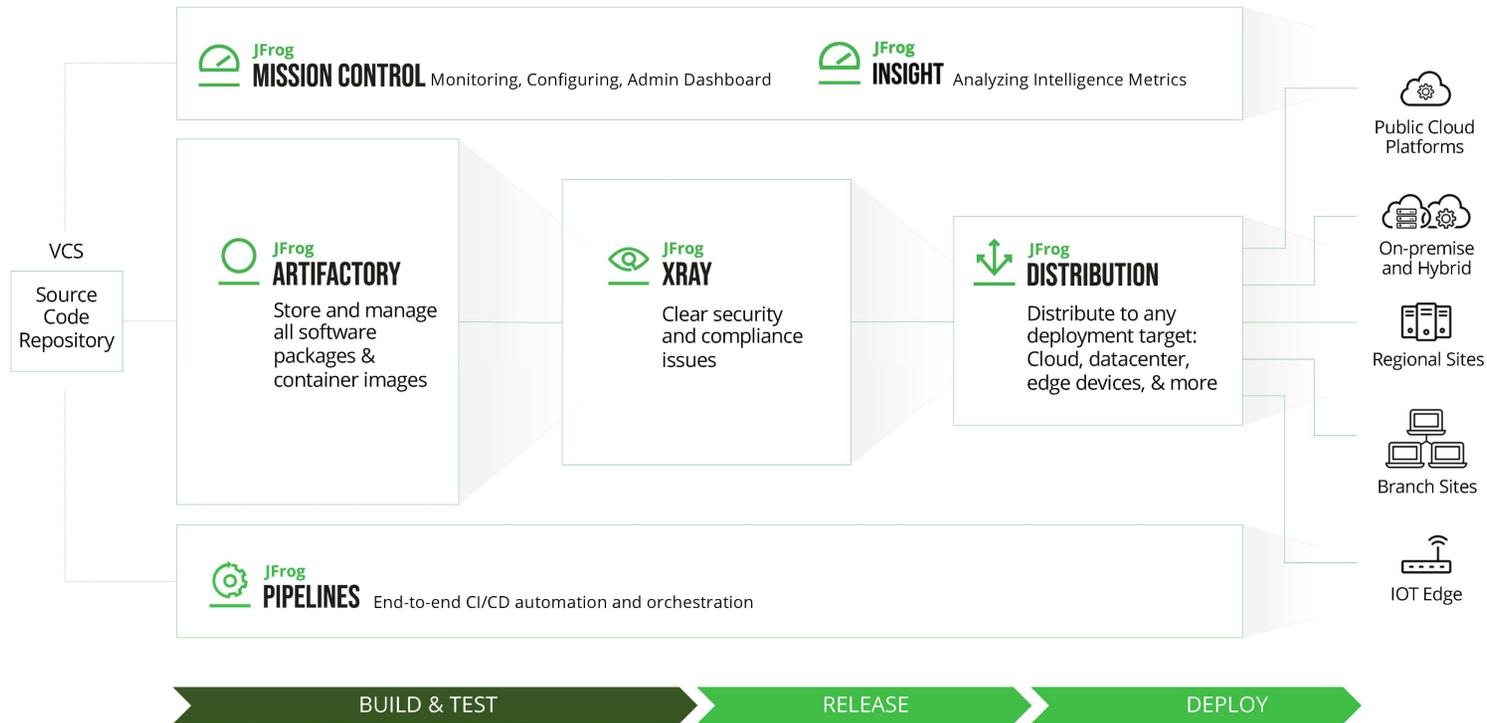
Artifactory's design and features empower all developers with the best practices of DevOps that reduce friction in key lifecycle stages, and accelerate software delivery.

The effectiveness of this binaries-centered approach to agile digital transformation is proven by thousands of JFrog customers every day, every hour.

Through your secure, single source of truth for all your binaries, the JFrog Platform can complete your DevOps modernization with hardened security, monitoring for vulnerabilities and compliance, and distributing securely to the farthest edges of your software consumption points.

The JFrog Platform is available at multiple subscription levels for on-prem or as a managed service in the cloud. Each level includes a set of JFrog components appropriate to your organization's needs. Whether you're a small development team for internal applications or a global enterprise delivering secured software to the edge, JFrog has the right set of solutions for you.

# THE JFROG PLATFORM



## ARTIFACTORY

Your secure source of truth for all packages, images, and artifacts as they move across the DevOps pipeline. The mission-critical core powering the JFrog Platform's binaries driven formula for DevOps success.



## XRAY

Harden trust in your single source of truth for DevSecOps. Keep known vulnerabilities out of software delivered to production, and swiftly remediate across your entire application inventory for all hazards you later discover.



## PIPELINES

Power your JFrog circle of trust with natively integrated CI/CD automation that enables the best practices for DevOps success. Extend traceability of your binaries leftward to the pipelines that create them.



## DISTRIBUTION

Extend your JFrog-powered circle of trust to the global consumption points of your software. Deliver trusted, production-quality releases to the many clusters, desktops and devices where they'll run.

# Summary

A binary repository manager such as Artifactory can boost productivity of your organization's Java and Kotlin development teams by managing and optimizing access to binary artifacts, integrating with build systems and facilitating open source governance. Since Artifactory is a universal repository manager, it can aid your Java and Kotlin development using Maven, Gradle, and Ivy, as well as Python, Docker, and virtually any third-party repository type used by your organization, and provide central control over all binaries, both when uploading builds and when downloading artifacts. As the core component powering the JFrog DevOps Platform, Artifactory helps manage, secure, and deliver your binaries from end to end.

START FOR FREE: [jfrog.com/start-free/](https://jfrog.com/start-free/)



[www.jfrog.com](https://www.jfrog.com)



[www.twitter.com/jfrog](https://www.twitter.com/jfrog)



[www.facebook.com/artifrog/](https://www.facebook.com/artifrog/)



[www.linkedin.com/company/jfrog-ltd](https://www.linkedin.com/company/jfrog-ltd)

