



8 REASONS FOR DEVOPS TO USE A BINARY REPOSITORY MANAGER

Copyright © 2021 JFrog Ltd.
Publish: October 2021



INTRODUCTION

All companies are now software companies; software has taken over the world. Applications are now the core value driver for all companies, whether in technology, finance, manufacturing, health care, or entertainment.

Developing software has transformed from an activity of small, tightly-knit groups to one performed by many interdisciplinary teams, oftentimes sprawled across the globe. An iterative and incremental approach to software development is now the norm at most enterprises.

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**.

As releases become more frequent, expectations from end-users keep pace -- customers demand frequent software updates to fix bugs, remediate vulnerabilities, and provide new features.

TOTAL SPEND
\$6.8 TRILLION
on digital transformation
globally by 2023

70 PERCENT
of digital transformation
efforts fail to meet goals

This is why enterprises are increasingly focussed on DevOps -- the tools and best practices that enable creating and deploying more releases, more frequently, and at high quality into production environments. To succeed, 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.

We'll explain why a binaries-centric approach to DevOps is the best way to be successful. Then we'll show the top 8 ways that Artifactory, the binary repository manager at the heart of the JFrog DevOps Platform, gives SREs and DevOps teams the tools to execute it through efficient management of an ever-growing matrix of binaries, environments, and geographically distributed sites.

Along with the supporting solutions in the JFrog Platform, Artifactory can help ensure a smooth and secure application development workflow from end to end.

DEVOPS SUCESS LIVES IN YOUR BINARIES

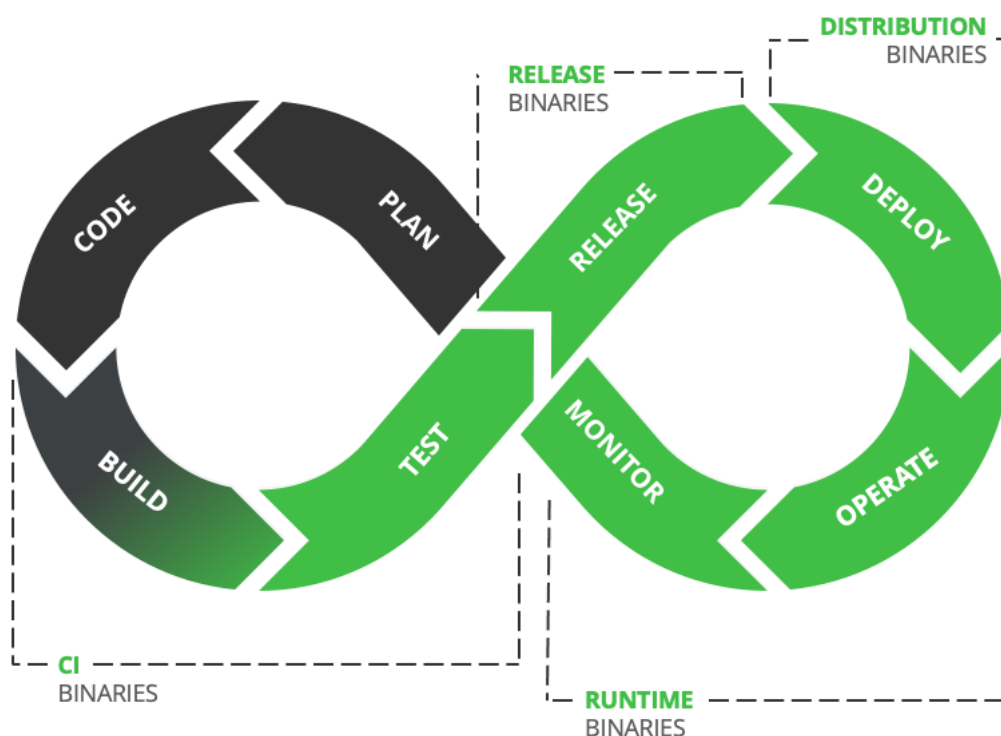
You're probably asking, "How can DevOps be about binaries? Isn't software development all about code?" Of course it is, and developers need to be able to focus on writing innovative, bug-free source code that creates value for customers. To most developers, source code is what matters.

But as important as coding is, source code isn't what your customers use -- they run an application, which is a composite of many binaries from many sources. The quality of the software you deliver to your customers depends on the quality of the binaries that go into it.

Software development is about code, and quality code comes from smart people who know how to write it.

Software delivery is about binaries -- ensuring quality builds and getting them swiftly into your customers' hands. **Quality binaries come from smart systems that know how to manage and distribute them.**

We can see this in the standard diagram of the DevOps loop for a best practice SDLC:



What's clear? Source code plays a foundational, yet proportionally small part of the total SDLC. Most processes in the DevOps loop don't deal directly with code at all, but instead center on the binaries that are built, tested, deployed, and run.

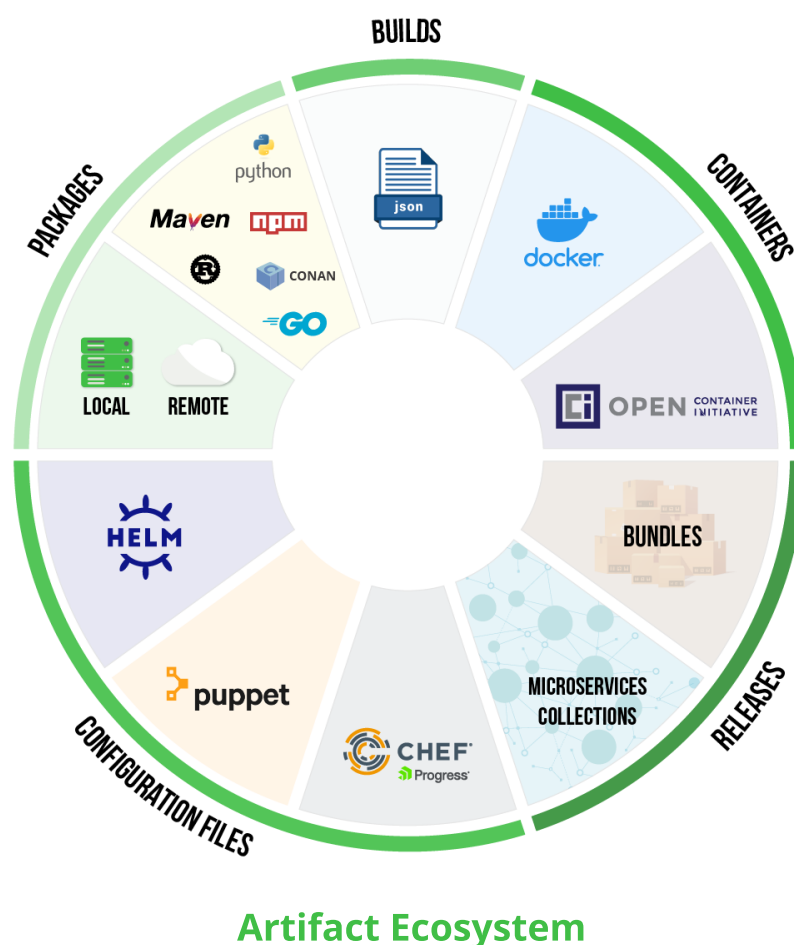
ARTIFACTS: WHERE DEVELOPMENT MEETS DELIVERY

Artifacts are the building blocks of software, from your packages and configuration files to the binaries that are the deployable runtime components of your application. Every delivered application is composed of a set of artifacts that connect to each other.

Your binaries are what travel through your software delivery pipeline(s) from end to end as they pass checkpoints of testing, validation, and security before final deployment to production.

While every application begins with source code, each source file both produces and relies on many artifacts (binaries and dependencies) through many build and test steps before resulting in a deliverable piece of software.

DevOps aims to **speed the delivery of quality software releases**. It does this by removing friction in the software delivery flow, which is most often found where software lifecycle processes meet. The consistency and certainty this brings enables automation for fast delivery.



Your ecosystem of artifacts -- deployable binaries and their associated files -- occupy the space where development meets delivery. The JFrog Platform's ability to **control, ease, and automate the flow of your binaries is the single most powerful key to successful DevOps**.

1 A SINGLE SOURCE OF TRUTH FOR DEVOPS

Adhering to a single set of best practices would be easy if all development used a single programming language.

But as organizations add languages and technologies to their tool chain, it becomes progressively harder to practice and enforce the same workflows.

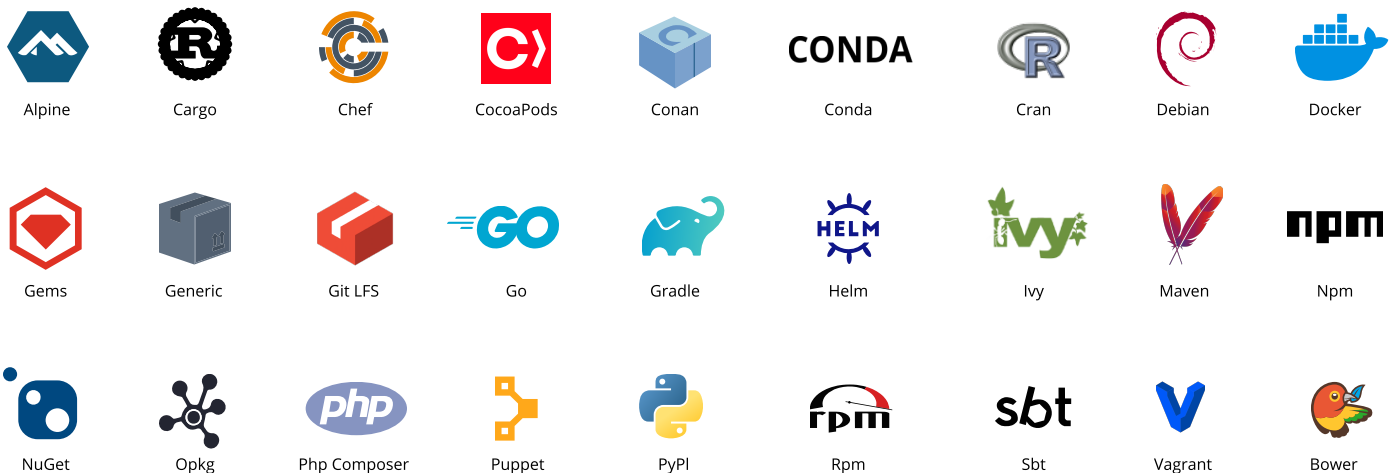
As the power behind the JFrog Platform, Artifactory is a universal repository manager for all of your artifacts: dependencies, binaries, and configuration files. With native support for over 30 package types (including generic repositories), it's your single source for storing, securing, and tracing 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.



Polyglot Development

Most enterprises use
12 or more
distinct package types

Over
50 percent
of developers expect to
adopt a new language
in the next 12 months



Artifactory Repository Types

Artifactory's universal binaries management serves all the developers in your organization. Whether they develop programs in Java, JavaScript, Python, Go, C++, or C#, Swift, Rust, or more, Artifactory is the central home for everyone's packages and builds.

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.

2 TRACK A SOFTWARE BILL OF MATERIALS (SBOM)

Making builds is easy. Knowing about everything you've built is hard. Especially when you make many new builds every day or every hour.



Artifact Counts

Over
5 million
artifacts per organization

On average
130%
more artifacts each year

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.



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

ARTIFACTORY QUERY LANGUAGE

Metadata is crucial for CI/CD processes; it's what enables automation to make smart decisions.

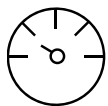
Artifactory Query Language (AQL) empowers you to uncover any data related to the artifacts and builds stored within Artifactory repositories. Its syntax offers a simple way to formulate complex queries that specify any number of search criteria, filters, sorting options, and output parameters. AQL is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

Using AQL, DevOps engineers can quickly identify relevant packages and builds quickly within growing repositories. They can employ AQL in their CI/CD servers to automate procedures that would otherwise require human intervention, or in analytics to improve workflow performance.

3

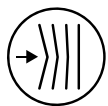
CACHE REMOTE REPOSITORIES BY PROXY

The open source software (OSS) dependencies drawn from remote resource repositories such as npm, Maven, Conan, and others 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.



Heavy loads

Delays produced by heavy demand for a service.



Site downtime

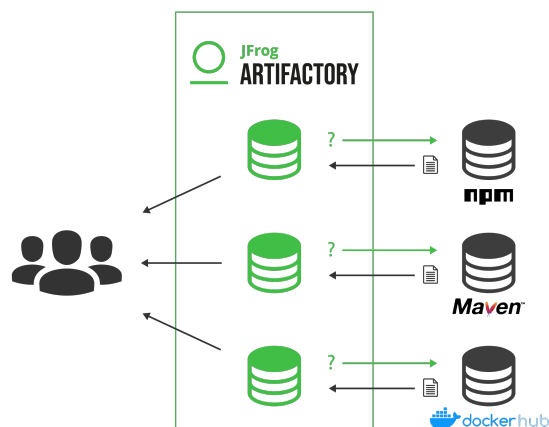
The remote site where your dependencies are stored may suffer a service disruption from failure, attack, or loss of service.



Poor connectivity

Network outages, jitter, poor bandwidth, unstable connections.

A remote repository in Artifactory is a local proxy that caches your remote resource dependencies on the same site where you keep your local repositories. Developers never access a remote resource directly, but build using an on-demand copy of the dependency in Artifactory.



Remote Repositories

Over
1.3 million
packages in main npm
registry

92 percent
of apps use open source
code

This helps eliminate network latencies inherent in physical distance or an unstable service connection, and keeps builds running as fast as possible. The proxy also protects against disruption due to connectivity breaks or if the remote site itself is unavailable.

Equally important, the cache in Artifactory helps maintain your dependencies as immutable versions -- after the first on-demand pull from the remote resource, the package in the cache never changes. This guards against any force-push overwrite into a remote resource (possibly with malicious intent) and guarantees deterministic builds.

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 it.

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 will Artifactory 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.

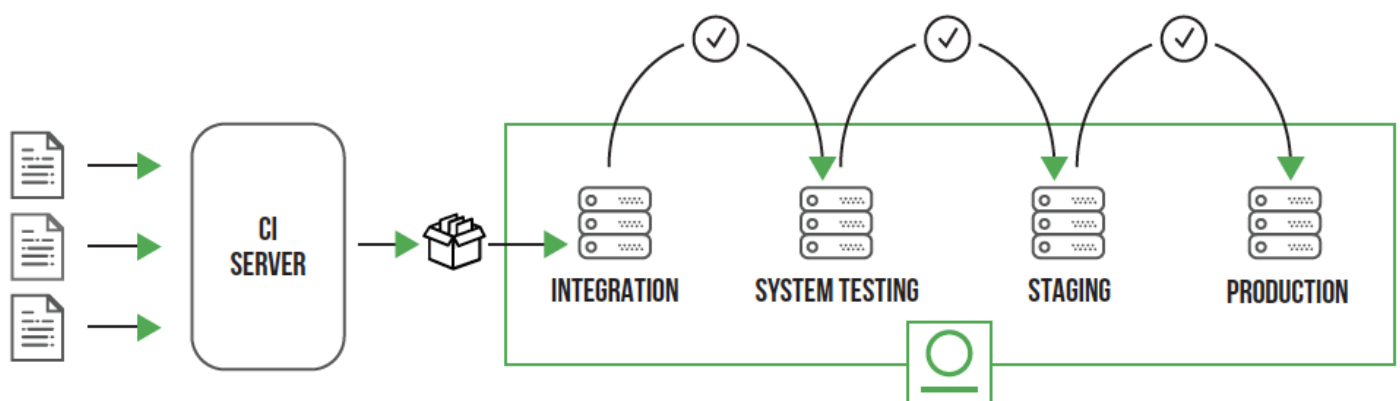
4 BUILD ONCE AND PROMOTE EVERYWHERE

Under the agile methodology of continuous integration, every new software version must pass several quality gates in an SDLC. A candidate version is promoted successively to different teams for integration, testing, and staging before it is released for general use.

But what passes through these gates makes the difference between a speedy or a plodding path to release. When version promotion is done through source code, each receiving team must perform its own deterministic build of the code in their own runtime environment, potentially leading to a different set of binaries. This lacks certainty that the software evaluated at each stage is exactly the same as the last.

Artifactory's binary repository management offers a more consistent and reliable 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.

Once free of having to perform their own builds or manage build environments, teams can apply the hours they recover to conducting more exhaustive tests and delivering feedback more quickly.

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**.



Continuous Integration

Over
61 percent
of developers spend only
2-4 hours each day coding

CHECKSUM-BASED STORAGE

The JFrog Platform optimizes storage by ensuring that any binary is only stored once on the file system. When a binary is stored in a local or remote repository, Artifactory calculates a unique checksum (both MD5 and SHA1 are supported) of the file and renames the file to its checksum. Repositories hold only references to files and their metadata, so when a binary is copied or promoted to another repository only the references are changed. The physical file is never duplicated, and its checksum can be used to verify the binary's integrity.



5

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 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 Docker images. In this way, the JFrog Platform can serve as your comprehensive Kubernetes registry, a central, traceable home for everything deployed to your clusters.



Cloud Modernization

80%
of edge applications are
deployed to Kubernetes

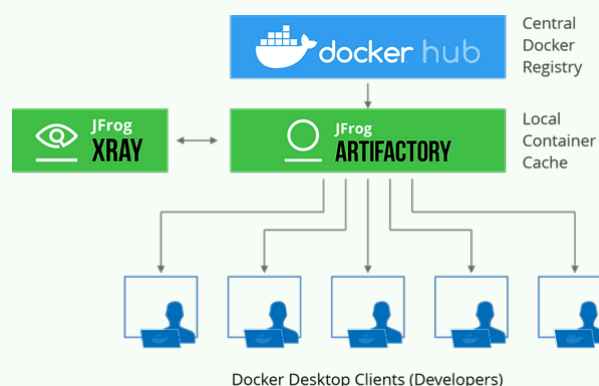
Over
\$24
billion/year
spent on cloud application
modernization globally
by 2025

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.

By leveraging JFrog Xray, developers can also gain continuous, comprehensive vulnerabilities scanning of the images they pull from Docker Hub.



6

PROTECT YOUR SOFTWARE SUPPLY CHAIN

Software development is like baking a cake. The code is the recipe, the binaries are what people eat. When you run a bakery, your customers never see the recipe but expect the ingredients to be pure and fresh.

When you run an enterprise, protecting your software supply chain -- your application's ingredients -- from mistakes and attacks is critical to protecting what's most precious: your business. Supply chain attacks like Tyupkin (2014), NotPetya (2017), Operation ShadowHammer (2019), and SolarWinds (2020) have spurred a heightened awareness and inspired the World Economic Forum to rank them among the top cybersecurity challenges. As the single source of truth for all of your binaries, a binary repository manager makes an attractive target for a supply chain cyberattack.



CyberSecurity

Median of
130
security breaches each
year at a typical firm

Up to
\$109 billion
lost to malicious cyber
activity per year in U.S.

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

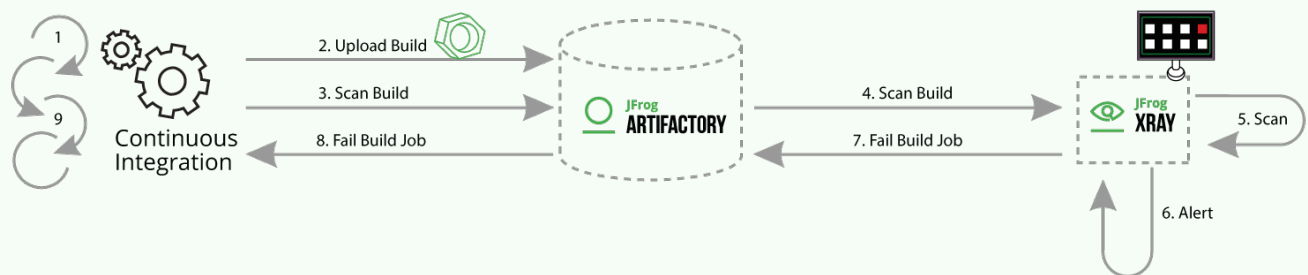
Support for LDAP, SAML, OAuth and SCIM protocols empower admins to integrate with Single Sign-On services such as Active Directory, Crowd and others for secure, organization-wide credentials management.



VULNERABILITY SCANNING

In a recent survey, only 18% of organizations reported being extremely confident in their open source components, which can typically be 60-80% of an application's code. Nearly twice as many were either not very or not at all confident.

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.



With Xray, you can prevent high-risk builds from being deployed into production, and enable recall of builds with risks that are newly discovered.

In addition, 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.



7

CONNECT TO YOUR UNIVERSE OF TOOLS

For developers, DevOps is a set of pipelines flowing in one direction toward a production-quality release.

But for the site reliability engineer, DevOps is an organizational fabric, weaving tools and procedures together to create a strong, resilient infrastructure.

Integrating a complex DevOps tool stack well is critical for reliable operations, and keeping this ecosystem running can consume far too much engineering time.

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:



Tool Stack Integration

Less than
50%
of SRE hours should be
spent on operations



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.



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.

Of course, that's 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.

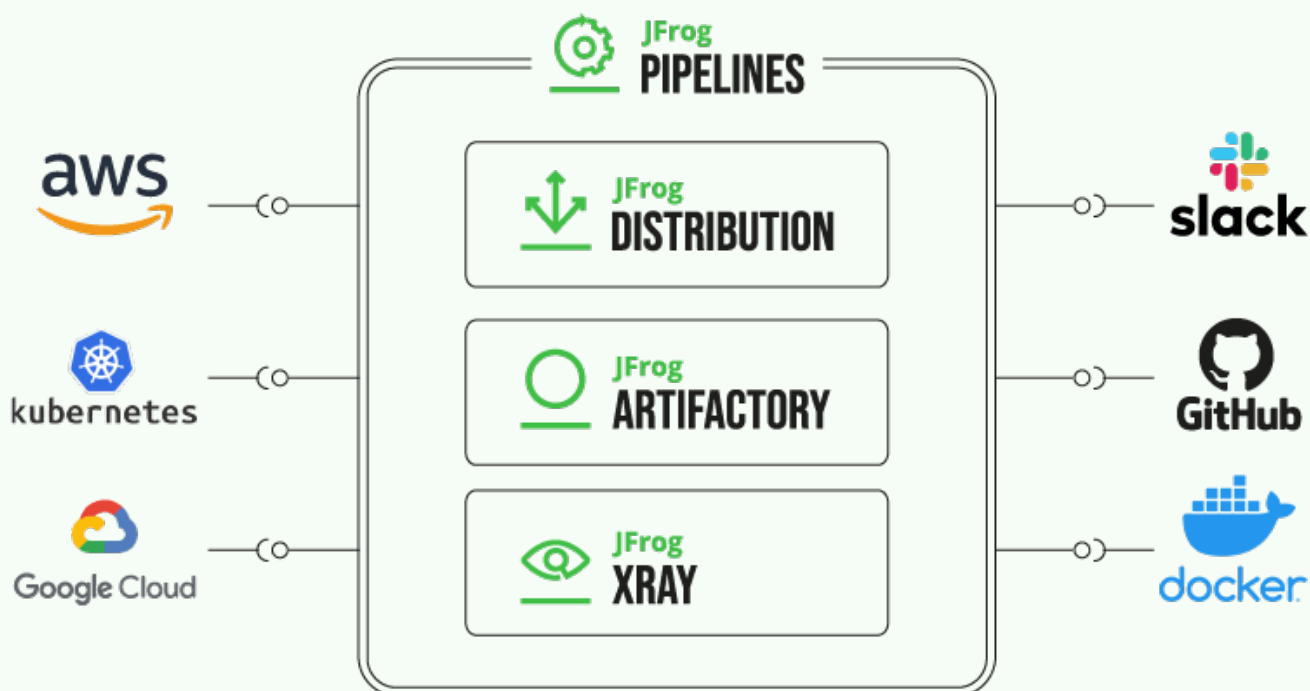
With a large family of JFrog partner integrations with top industry providers, the JFrog Platform can help quickly build a strong, tight weave of your DevOps tool stack through this single source of truth.



NATURALLY INTEGRATED CI/CD

JFrog Pipelines CI/CD is the fastest way to build your CI/CD ecosystem. As part of the JFrog DevOps Platform, Pipelines is naturally joined with Artifactory. Using Pipelines' declarative pre-built steps, you can focus on what goes into your repositories, not how to get it there.

With the remaining components of the JFrog Platform, Xray, and Distribution, the unified tools at the heart of your DevOps universe are pre-integrated at installation.



Pipelines also provides out-of-the-box integrations for the most popular tools in your ecosystem, including GitHub, GitLab, BitBucket, Slack, Jira, AWS, GCP, Docker, Kubernetes, and many more. Integrating most of these services with Pipelines is as simple as entering a URL endpoint and user credentials.

8 SCALE TO INFINITY

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.

Even small teams should expect to grow, and the best practices of DevOps have to stretch seamlessly along with them.

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.

As your mission-critical tool, you'll need to ensure continuous and responsive access even as your department grows.



Growth And Scale

Average of
5 servers
with Artifactory per
enterprise

Over
11 million
binaries per enterprise
account

Over
27 million
software developers
globally by 2023



On-Premises High Availability

Self-host on servers in your own secure datacenter on-premises, or self-manage as a BYOL installation in clusters in your own cloud account.

A High Availability configuration of replicated nodes in your 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 system updates, up to "five nines" level of availability.



Cloud (SaaS)

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. Employ a multicloud strategy by maintaining multiple SaaS accounts hosted on different cloud providers. This helps avoid vendor lock-in, and enables allocating workloads to the most cost-effective provider.



Hybrid

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

The JFrog Platform's "same here, same there" promise of feature equivalency in every environment means you can divide your workloads between cloud and on-prem however you need.



Multi-Site Geo-Replication

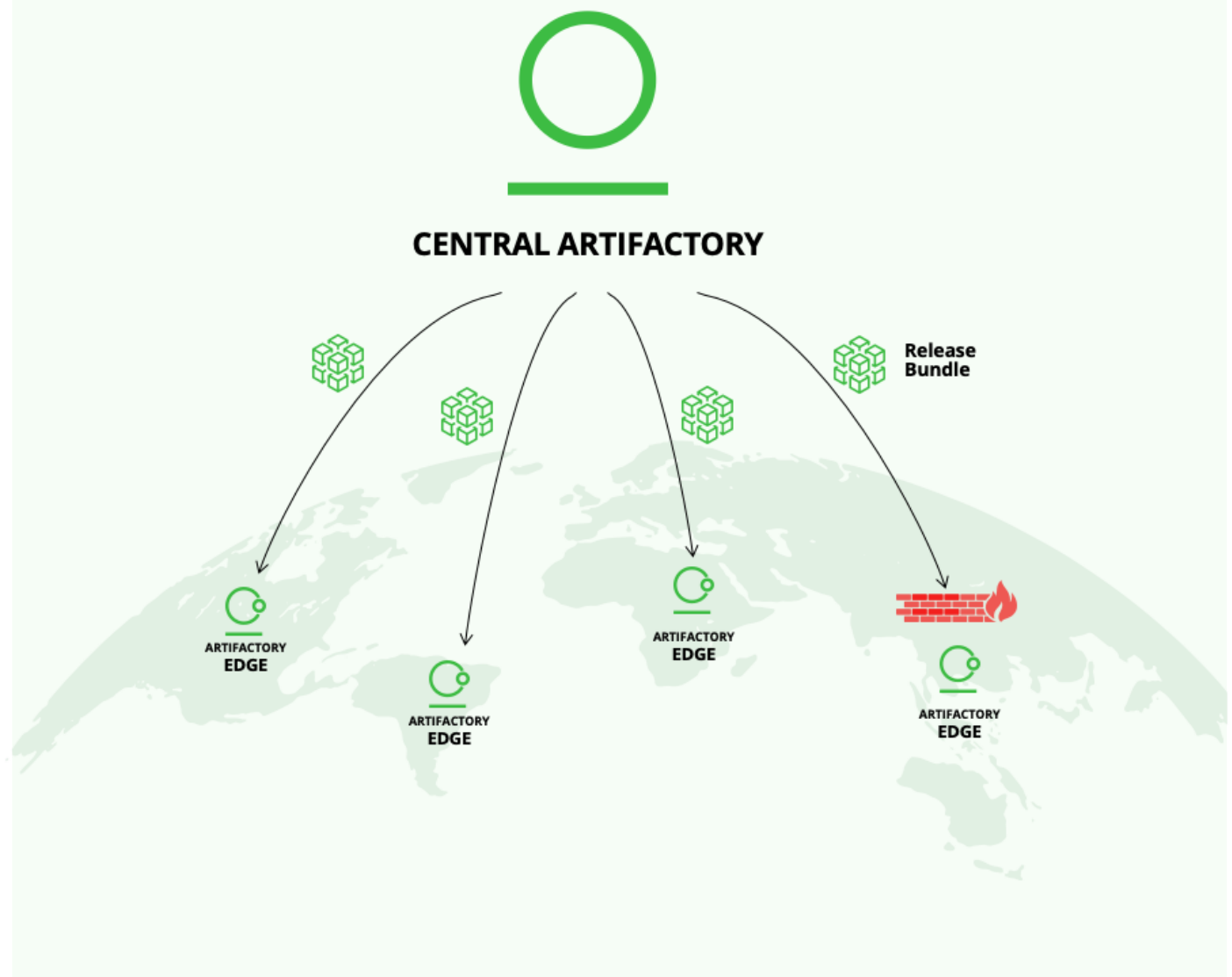
The JFrog Platform enables multi-site geo-replication through several push/pull replication topology options, or bidirectional federation of repositories.

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

DISTRIBUTE TO THE FARTHEST EDGES

Once you have a fully validated release, where does it need to go? Enterprises must distribute software and updates to a growing number of global endpoints, for delivery to the clusters, devices, and desktops where it will benefit users.

As part of the JFrog DevOps Platform, JFrog Distribution empowers DevOps teams to efficiently package release bundles (binaries, artifacts, and metadata) from Artifactory and automate trusted software delivery to dozens or thousands of remote sites across the globe. Using JFrog Edge Nodes -- cost-effective read-only Artifactory instances -- you can transmit signed, immutable releases through a private data network that are verified at each destination, while maintaining fine-grained access controls.

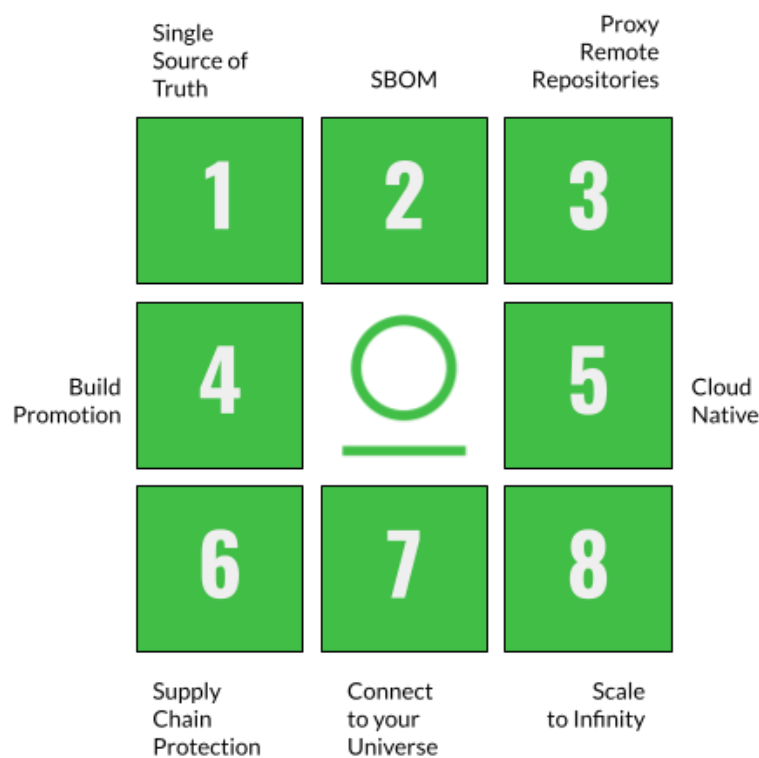


ARTIFACTORY PUTS YOUR BINARIES TO WORK

The evidence is clear: 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 8 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 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.



Growth And Scale

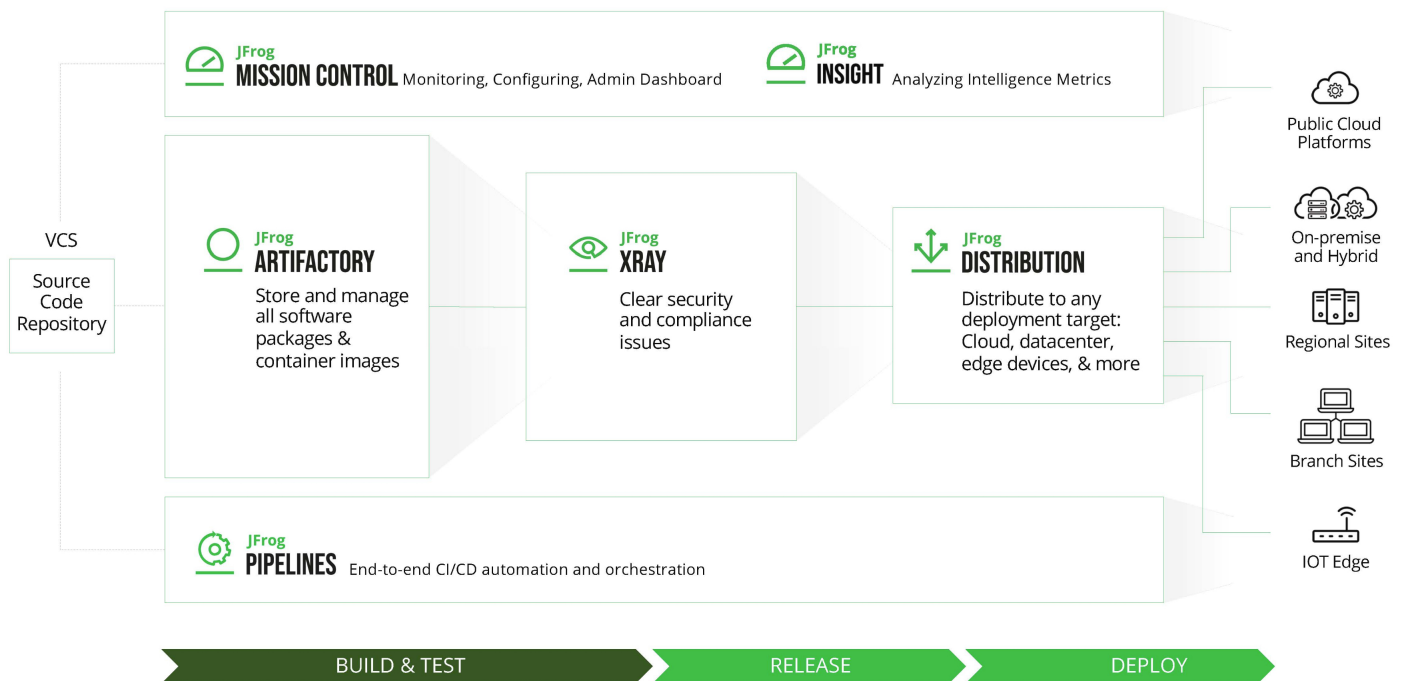
Over
8300
organizations
use JFrog

Over
70%
of the Fortune 500
use JFrog

Over
42 billion
artifacts in
JFrog Artifactory
repositories

JFROG PLATFORM

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



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.

Statistics and Sources

- Total spend \$6.8 trillion on digital transformation globally by 2023
IDC: Spend on Digital Transformation
- 70 percent of digital transformation efforts fail to meet goals
McKinsey: The 'how' of transformation
- Most enterprises use 12 or more distinct package types JFrog internal data
Over 50 percent of developers expect to adopt a new language in the
➤ next 12 months
JetBrains: The State of Developer Ecosystem 2020
- Over 5 million artifacts per organization JFrog internal data
- On average 130% more artifacts each year JFrog internal data
- Over 1.3 million packages in main npm registry npm blog, 2020
- 92 percent of apps use open source code Tidelift 2018 Open Source Survey
- Over 61 percent of developers spend only 2-4 hours each day coding
ActiveState 2019 developer survey
- 80% of edge applications are deployed to Kubernetes
- Over \$24 billion/year spent on cloud application modernization globally by 2025
Research and Markets, Application Modernization Services Market...Global
Forecast to 2025
- Median of 130 security breaches each year at a typical firm Ponemon Institute,
2017 Cost of Cybercrime
- Up to \$109 billion lost to malicious cyber activity per year in U.S.
*U.S. Council of Economic Advisors, The Cost of Malicious Cyber Activity to the U.S.
Economy, 2018*
- Less than 50% of SRE hours should be spent on operations
Google, Site Reliability Engineering
- Average of 5 servers with Artifactory per enterprise JFrog internal data
- Over 11 million binaries per enterprise account JFrog internal data
- Over 27 million software developers globally by 2023 Evans Data Corp., Global
Developer Population and Demographic Study 2019