



WHITE PAPER

THE JFROG PLATFORM

AN END-TO-END PLATFORM FOR GLOBAL DEVOPS



TABLE OF CONTENTS

Executive Summary	3
The JFrog Platform	3
Software: The Common Ground	4
A Platform for Liquid Software	5
JFrog Artifactory	6
JFrog Pipelines	7
JFrog Xray	8
JFrog Mission Control	9
JFrog Insight	9
JFrog Distribution	10
JFrog Artifactory Edge	11
JFrog Access	11
DevOps Success with the JFrog Platform	12
A Complete DevOps Solution	12
Plan to Succeed	13
Cloud Native Support	13
Built for Scale	13
Secure Your Success	14
Deliver Successfully	15
Common Use Cases for the JFrog Platform	16
Multiple Stores or Locations	16
Distribution of Different Release Versions from Many Different Technologies	16
Managing Remote Production Sites	16
Global Company with Geographically Dispersed Developers	16
Summary: Let the JFrog Platform Take Control of DevOps	17

EXECUTIVE SUMMARY

Today, we live in a very connected world, where our devices, homes and cars all communicate with each other, and every company with a product or service has the need to develop software. It is one of the primary mediums by which they strive to provide better products, services and solutions, and has become paramount to a company's success. To continuously improve their software, companies must have sound DevOps or DevSecOps practices in place. Over time, software methodologies have changed: agile has replaced waterfall, cloud providers have replaced server farms, and the next paradigm shift in DevOps is the JFrog Platform ("the Platform").

THE JFROG PLATFORM

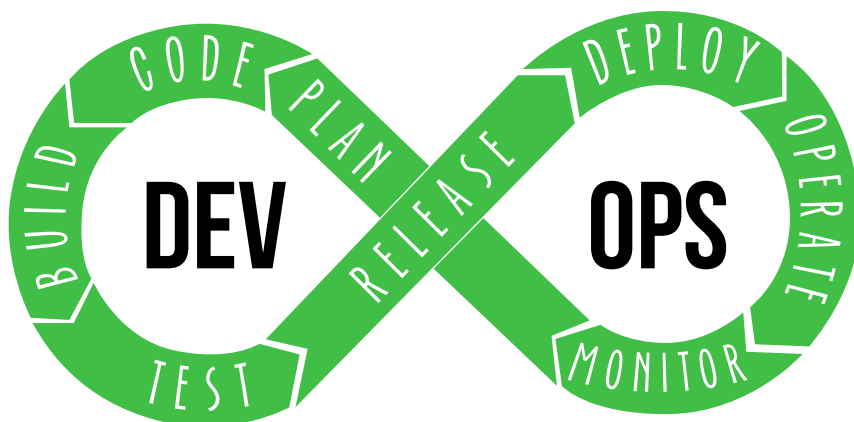
A solid DevOps or DevSecOps foundation is essential to a company's success and this is provided by the different components of the Platform. It enables software teams to collaborate across the globe through all phases of the development cycle. The Platform is designed to meet the growing needs of companies to develop and distribute software and provides DevOps teams with the tools needed to create, manage and deploy software with ease. Once a DevOps process is under control, an organization can focus on the innovation that drives its business without having to worry about infrastructure.

JFrog Artifactory stores and manages binaries throughout the development cycle, and through its replication capabilities it keeps international teams on equal ground with the exact same components and dependencies. JFrog Pipelines provides end-to-end orchestration/Automation, and optimization of all critical processes of your CI/CD pipelines. JFrog Xray deep-scans all components, builds and dependencies for vulnerabilities and license violations, providing policy alerts based on a wide variety of parameters. JFrog Distribution collects builds and their associated components into release bundles and distributes them to other Artifactory instances, or to multiple JFrog Artifactory Edge nodes. Manage and monitor the complete flow through JFrog Mission Control, using JFrog Insight to measure, monitor and continuously improve the pipeline while JFrog Access federates different services to enable single-sign-on.

The Platform with its tightly integrated components, can help build a system from scratch, or retrofit an existing infrastructure to meet your current and future DevOps or DevSecOps needs.

SOFTWARE: THE COMMON GROUND

In today's world where everything is connected, all companies are now software companies. Whether they manufacture refrigerators or medical devices or offer one of the many online services we all use to connect with our family, friends and business associates, the one component they all have in common is software. Providing a constant flow of better software anywhere on the planet has become paramount to a company's success. A company's software must continuously improve in terms of software quality, consistency, security and global reach. To meet that challenge, companies must have sound DevOps practices in place. While some have built up their software development organizations with DevOps in mind, others are struggling to retool their infrastructure to meet the ever-growing demand for better software.



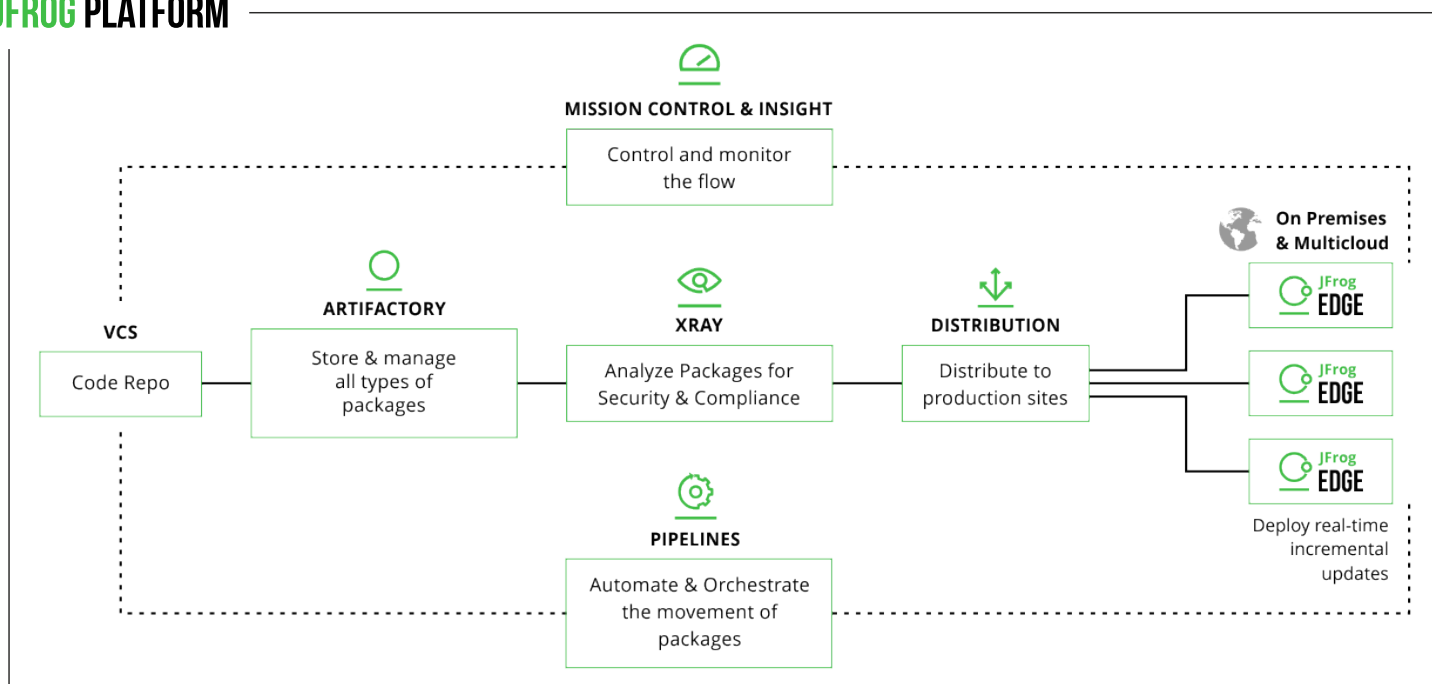
Over time, the methodologies of software development have changed. The “Waterfall” paradigm has been mostly replaced by “Agile,” and data centers filled with complex servers that are hard to maintain have been replaced with services hosted with cloud providers that can scale on demand using tools and technologies such as Kubernetes. This white paper will discuss the future of DevOps, The JFrog Platform.

A PLATFORM FOR LIQUID SOFTWARE

The software industry is undergoing a transformation in which release cycles keep getting shorter, and software updates are getting smaller and more frequent. JFrog envisions that companies will eventually transition from periodically transferring packages to continuously transferring micro-deltas of software. This is the liquid software revolution of continuous updates, and it is this vision that drives JFrog's design and implementation of the Platform.

The Platform is designed to meet the growing needs of companies to develop and distribute software and provides DevOps teams with the tools needed to create, manage and deploy software with ease. These tools cover everything from CI/CD tools, binary management, artifact maturity, security and vulnerability protection, release management, analytics and distribution.

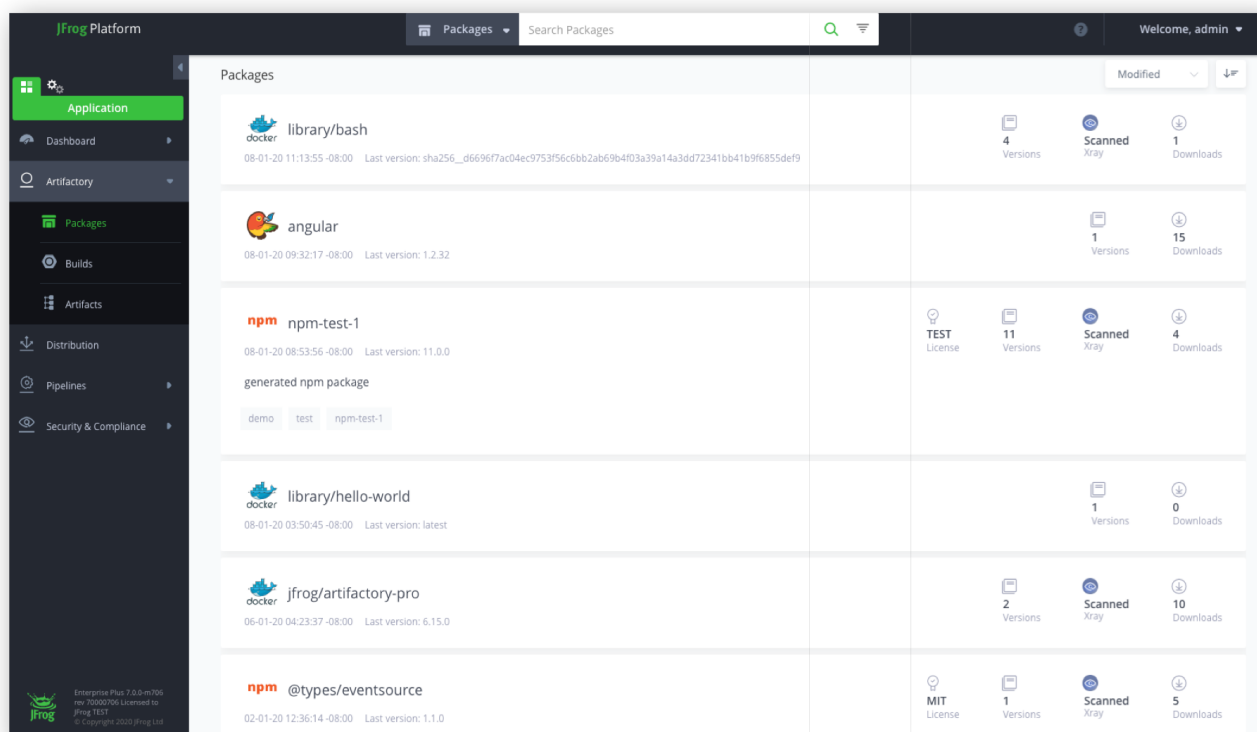
JFROG PLATFORM



In the next sections we will briefly describe each component of the Platform, and then show how they work together to construct a coherent, efficient and successful DevOps toolchain to get your software from developers' fingertips out to endpoint computers and devices.

JFROG ARTIFACTORY - BINARY REPOSITORY MANAGER

JFrog Artifactory is the heart of the Platform. As a scalable, universal, binary repository manager that automatically manages your artifacts and dependencies through the application development and delivery process, it plays a central role in your DevOps ecosystem in general and in the Platform in particular. Through a highly available clustered solution and a variety of multi-site replication capabilities, it provides consistent and reliable access to your artifacts globally. With a powerful REST API and integration into any CI/CD ecosystem, Artifactory empowers you to release software faster with a fully automated CI/CD pipeline.



Package Browsing and Artifactory as a Docker Registry

Artifactory, as part of the JFrog Platform features an optimized replication algorithm. Replication is used in a variety of use cases to synchronize repositories between remote Artifactory instances. Depending on repository size this may require the transportation of huge quantities of data. Since the speed of transportation is limited by bandwidth and network latency, this is a process that may take hours and even days to complete. The new replication algorithm optimizes replication when distributing software with JFrog Distribution, dramatically reducing the load on the network and the time taken to synchronize release bundles from a source Artifactory instance to target instance or a set of Edge nodes.

Using Artifactory to Manage Binaries Across Multi-Site Topologies: [Read the White Paper and Learn More](#)

JFROG PIPELINES - CI/CD MADE BETTER

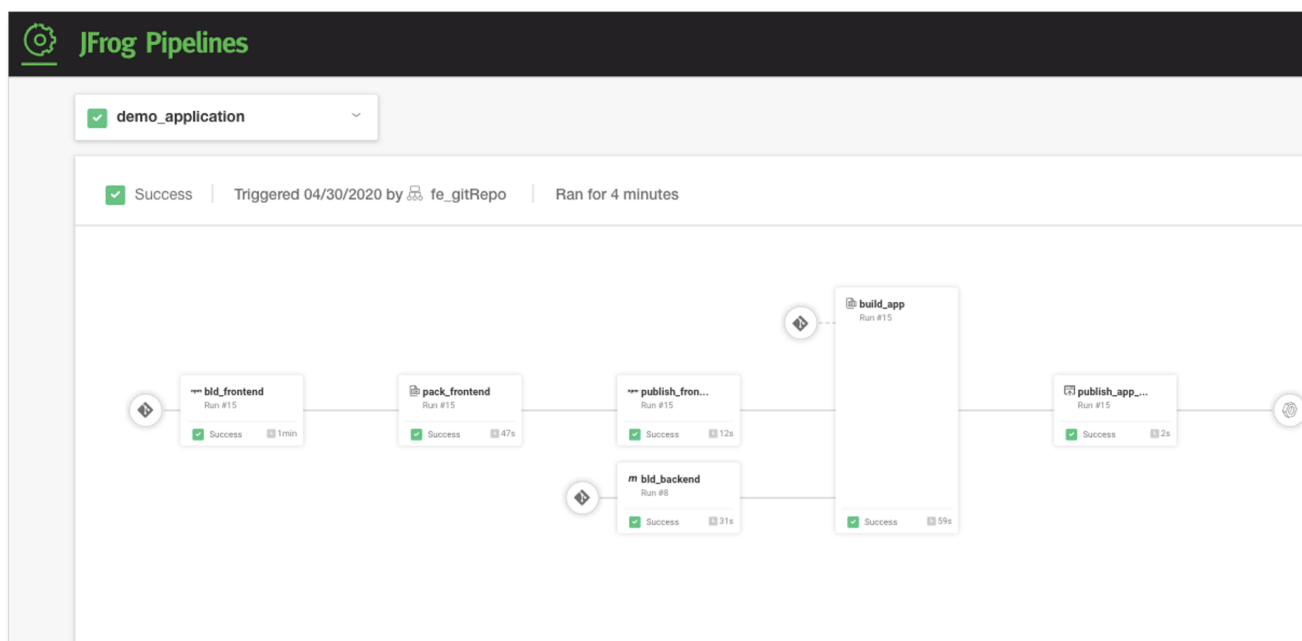
JFrog Pipelines provides end-to-end orchestration, automation and optimization of all critical processes of your CI/CD pipelines. It is a one stop DevOps platform that leverages a “No Code” approach to accelerate your DevOps journey.

```
pipelines:
  - name: demo_gosvc
    steps:
      - name: bld_svc
        type: DockerBuild
        configuration:
          dockerFileLocation: .
          dockerFileName: Dockerfile
          dockerImageName: 35.224.208.129:8082/docker-local/gosvc
          dockerImageTag: ${run_number}
          inputResources:
            - name: gosvc_gitRepo
          integrations:
            - name: demoArt
      - name: push_svc
        type: DockerPush
        configuration:
          targetRepository: docker-local
          forceXrayScan: false
          autoPublishBuildInfo: true
          integrations:
            - name: demoArt
          inputSteps:
            - name: bld_svc
          outputResources:
            - name: gosvc_build_info
```

JFrog Pipelines - Example JFrog Pipelines YAML Configuration

It is a Binary-driven Continuous Delivery (CD) tool that integrates with multiple Version Control Systems, has comprehensive REST APIs to enable easy extensibility and is built for enterprise scale. You can use it for either Continuous Integration (CI) (it also has integration with other CI servers), CD or both! Pipelines can also be used in combination with your current tools as it includes many integrations.

One of the major obstacles in writing CI/CD pipelines, is Imperative coding. You want to be able to do things quickly and efficiently, even without being a top-notch developer. This can be easily done using Pipelines, which provides you cross-team, multi-repository workflows. Using built-in steps, different out-of-the-box integrations (Docker, Kubernetes, Jenkins, Github etc) and a real focus on security (Fine-grained workflow permissions, Central secrets management and scopes) you have your full end-to-end solution - from Git to Kubernetes with one complete platform.



JFrog Pipelines - Pipeline View

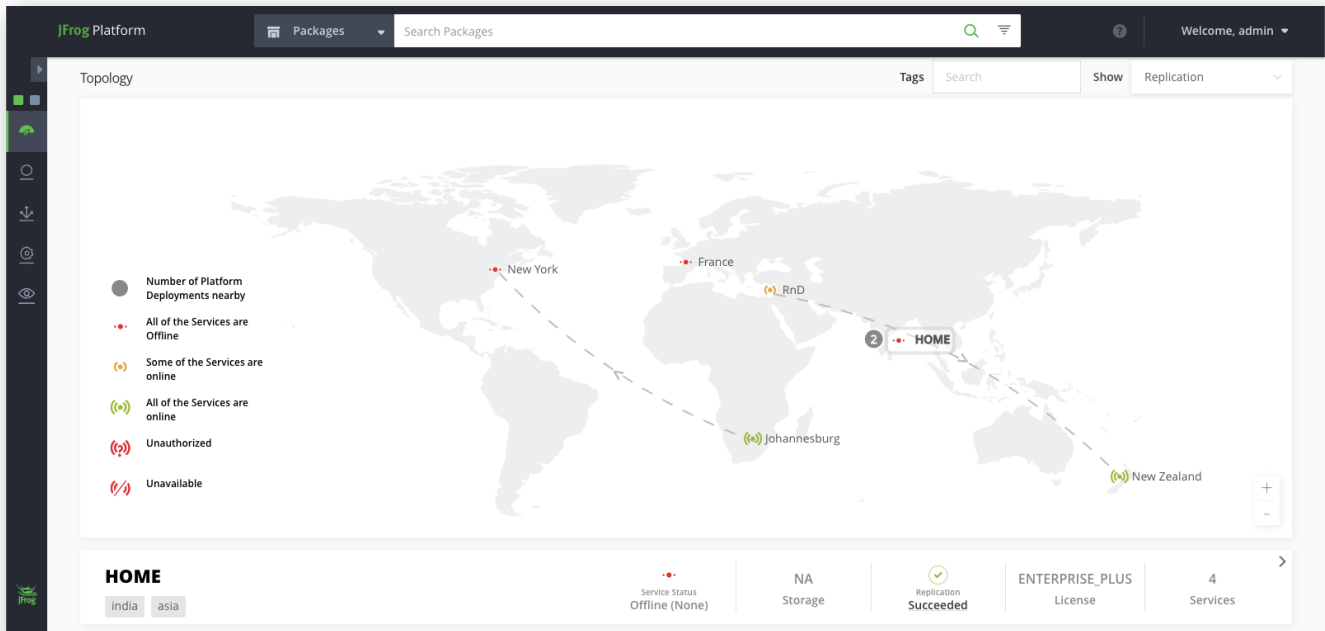
JFROG XRAY - SECURITY AND COMPLIANCE

JFrog Xray increases trust in your software releases by providing automated and continuous governance and auditing of software artifacts and dependencies throughout the software development lifecycle - from development, through testing, and on to production. As scanning for security vulnerabilities and license violations has become a critical step in any DevSecOps organization, Xray reinforces security standards through deep recursive scanning of artifacts to provide impact analysis and protection from vulnerabilities. As a sentinel guarding software security, Xray is a mission-critical part of your DevSecOps toolchain. As part of the JFrog Platform, Xray can be installed as a High Availability (HA) cluster with multiple active/active nodes.

If any node becomes unavailable, the cluster automatically distributes the workload among the remaining nodes thereby providing unparalleled stability and reliability. Xray contains a very large amount of data about known vulnerabilities in open source packages. Similar to Artifactory, Xray can scan many different binary and package types such as Docker, Java, NPM, Go and many others, ensuring what goes into your builds and your releases doesn't contain any high-risk vulnerabilities or license violations. Xray's main source of data for these vulnerabilities comes from VulnDB, which is the most timely and comprehensive vulnerability intelligence and includes new vulnerabilities discovered even before they are officially published. Xray also features a large repository of independent data, and is constantly in connection with public and private partners to keep the vulnerabilities database up-to-date from as many sources as possible.

Xray offers a much easier and integrated way to protect your code at every stage of the development cycle, and with the Platform, Xray can be installed in a HA configuration to meet the highest demands for scale, stability and performance.

JFROG MISSION CONTROL - ORCHESTRATING IMPLEMENTATIONS



Mission Control Site Map

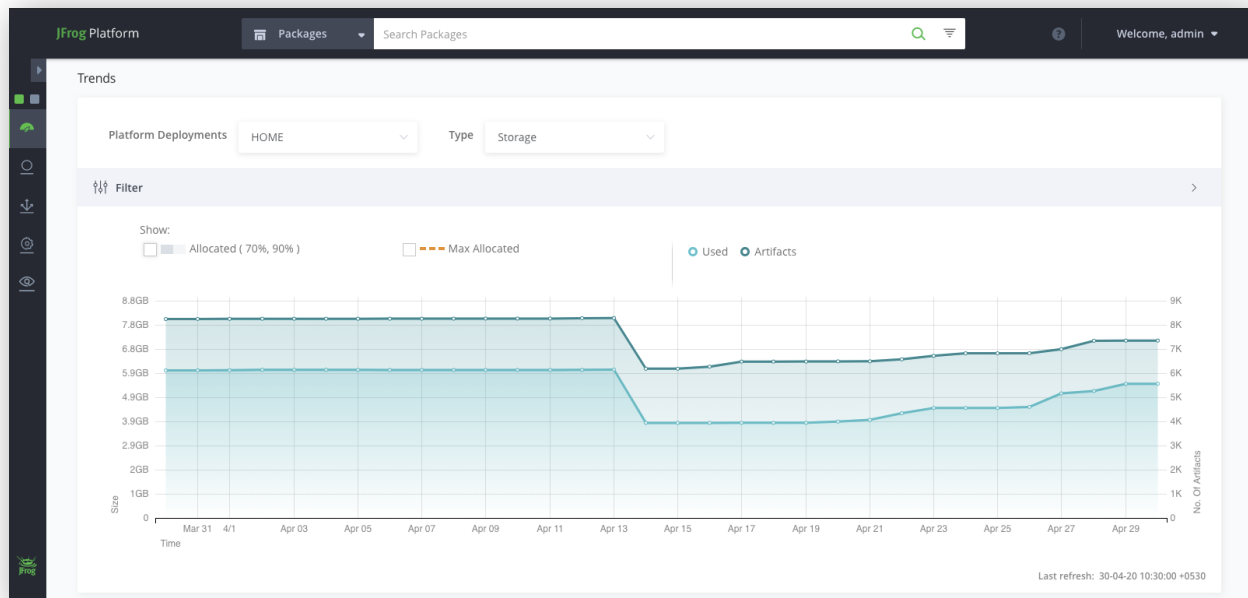
JFrog Mission Control has been established as a single access point for managing all Artifactory and Xray services under your administrative control, whether installed on-site or at remote geographical locations. As part of the Platform, Mission Control's capabilities have been enhanced to allow full control over additional services: Jenkins CI (with additional CI servers to be added in the future), JFrog Distribution and JFrog Artifactory Edge nodes. In addition to that, mission control can manage "access federation" setup, meaning that in multi-site topology, you will be able to sync not only your artifacts and their metadata (exists today using artifactory replication), but also all of the security entities such as users, groups, permissions and tokens so users will be able to connect to different artifactory instances located in different geographical locations seamlessly.

With companies spreading out their resources globally, the complete DevOps process may run through multiple geographically distant data centers over different cloud providers. Mission Control makes it easy to implement any plan for replication (either its data, metadata and/or security entities), distribution and deployment of your binaries through the platform.

JFROG INSIGHT - ANALYZE AND ADJUST FOR CONTINUOUS SUCCESS

JFrog Insight is a new DevOps analytics engine added to JFrog Mission Control as a part of the Platform. Insight gathers data about your DevOps tool chain and displays it in configurable dashboards so you can evaluate how your software development system is performing from source code check-ins to your CI/CD pipelines. For example, you can monitor parameters like total commits, build duration and number of dependencies across different builds of the same package. The different metrics you can monitor help you find bottlenecks in your pipeline so you can make better choices about how to adjust and fine-tune your system for optimal performance. And since Insight is integrated into Mission Control, you can manage, maintain and optimize the performance of your organization from one place.

Insight is the key to an optimal CI/CD pipeline. It can be a daily dashboard where DevOps administrators monitor the pipeline, view peaks of activity, detect bottlenecks and tweak systems to optimal performance. In the world of “Release Fast or Die,” gaining deep insight into the behavior of a pipeline to refine can be key to increasing ROI.

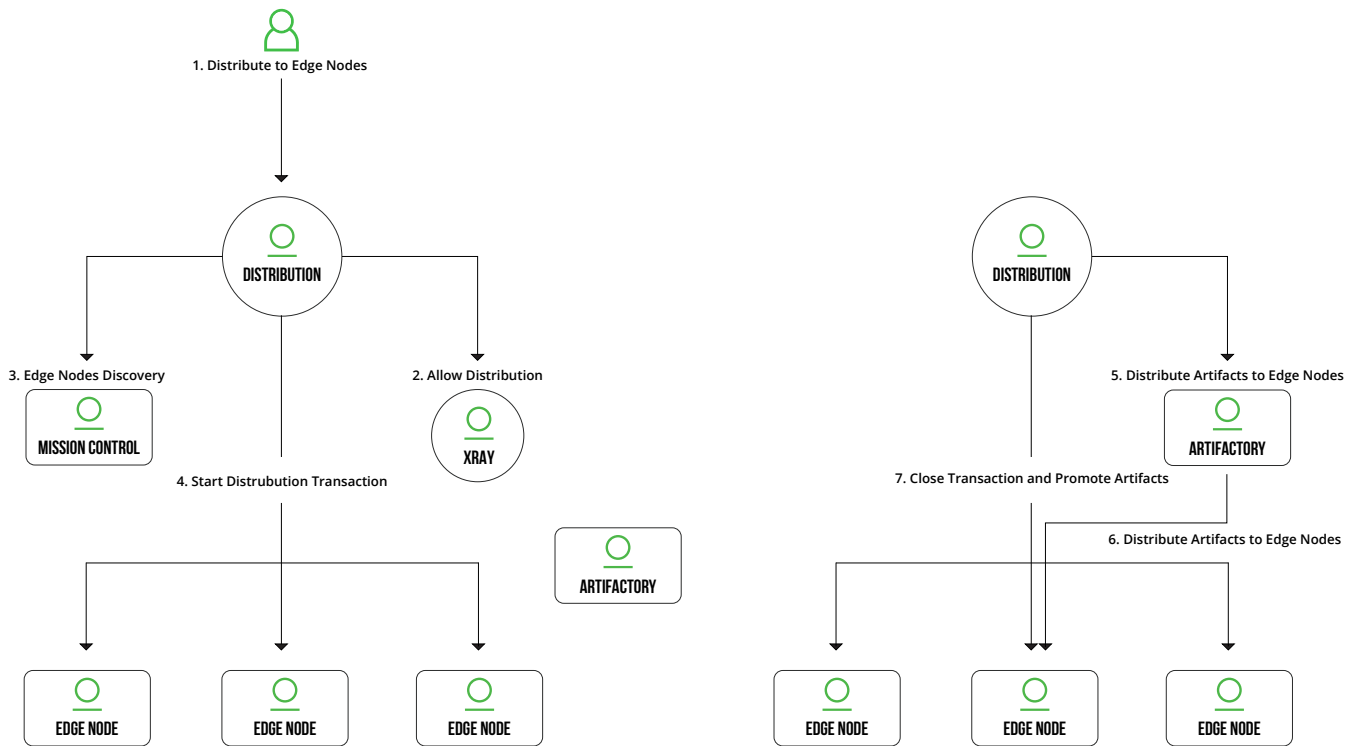


Insight Dashboard

JFROG DISTRIBUTION - ORCHESTRATE DEPLOYMENTS

JFrog Distribution is a tool that lets you orchestrate software distribution between two Artifactory instances or from Artifactory out to multiple Artifactory Edge nodes. Release bundles are central to the activities of Distribution. A release bundle is a new metadata format that groups together the different files and packages that make up a release into a single traceable bundle and provides the corresponding bill of materials for that release. For example, you can group the different build artifacts, such as Docker images, associated tools, documentation and other assets, that make up a release that should be pushed out to an Edge node and finally on to your point of sale devices. Release bundles are secure and immutable to ensure they cannot be manipulated by unauthorized individuals.

Distribution uses proprietary technology to reliably and optimally distribute release bundles to multiple remote locations and update them as new release versions are produced. As part of the release flow, release bundles are verified by the source destination to ensure that they are signed correctly and safe to use. By optimizing replication between source and target Artifactory instances (or Edge nodes), Distribution makes efficient use of the network dramatically reducing network load and release bundle synchronization time. For auditing and traceability, Distribution also tracks all changes made in release bundles across different versions.



JFROG ARTIFACTORY EDGE - MANAGE RELEASE BUNDLES

A JFrog Artifactory Edge node is a specialized version of Artifactory with the single purpose of delivering the contents of a release bundle directly to a consumer, and will therefore, be installed as geographically close to the compute edge as possible - whether the final destination of the release is a device, another computer or a point of sale terminal.

An Edge Node is installed just like any other Artifactory instance, supports the same databases, and can be run in an HA configuration, but is built especially to handle immutable release bundles. You cannot, therefore, just upload software to an Edge node in the same way you upload to a full Artifactory instance. You can only transfer software (release bundles) through Distribution, which ensures that all the required certification is in place to validate the source and integrity of the release bundles that are uploaded. Edge nodes can also pull by demand, binaries from a main Artifactory instance by adding it as a smart remote repository, so if you didn't create a release bundle for distribution yet, but a specific binary is needed it can be fetched directly from the source on the fly.

JFROG ACCESS - DEVOPS ADMINISTERED

JFrog Access provides a common authentication and authorization infrastructure for all of JFrog products to manage users, groups, permissions and tokens, as well as support for LDAP, SAML and OAuth. While not a separate product, Access is installed together with Artifactory and runs as a separate service under the same Tomcat. With the Platform, different Access services can be federated into a single "circle of trust". This lets you synchronize users, groups and permissions between sites rather than having to define them separately. It enables single-sign-on for all JFrog services connected to any Access service within the circle of trust, resolving the need to login to each product separately. Together with advanced capabilities for replication across multi-site topologies, this provides a complete solution for global artifact management.

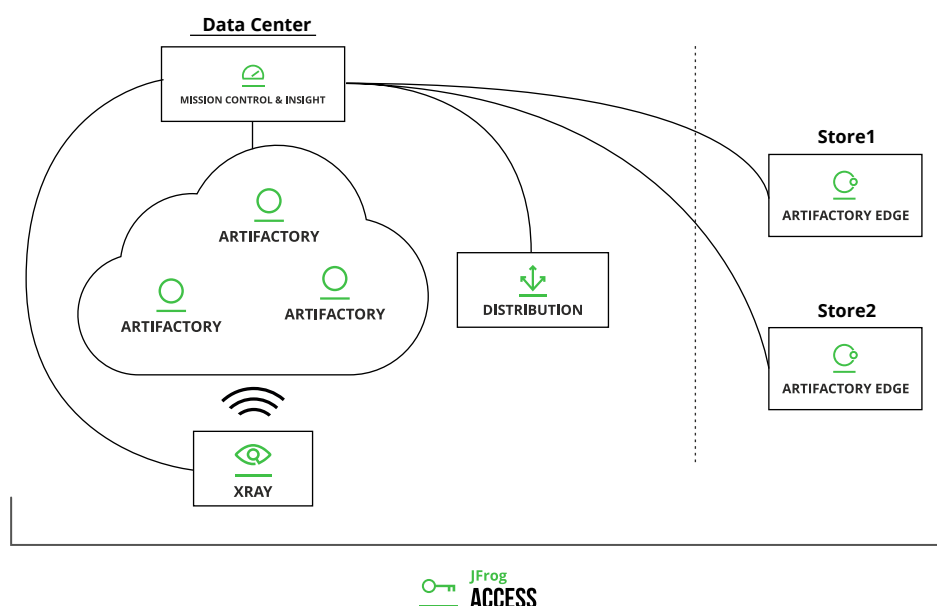
DEVOPS SUCCESS WITH THE JFROG PLATFORM

A COMPLETE DEVOPS SOLUTION

Much of DevOps (or DevSecOps when security is planned as an integral part of the flow) focuses on automation, software process, infrastructure, testing, security and deployment. While there are many tools to manage each of these tasks, they are mostly point solutions, and the question remains, “what happens between these stages of the DevOps flow?” This is where the Platform comes in to fill the gaps letting you lay down a solid DevOps foundation, so you can innovate more and worry less.

Using the Platform to manage binaries in the DevOps flow, enables software teams (along with their respective CI servers) to collaborate across the four corners of the globe, if necessary, with the same coherent and consistent set of binaries and dependencies through all phases of the development cycle. As soon as a developer downloads an external library, or a Docker base image, using Artifactory to cache the component along with all of its dependencies, and replicating them to all relevant teams puts everyone (and all CI servers) on equal ground. With all environments using the same exact components, no developer can ever use the lame old excuse of “But it works on my machine” again. By having Xray (in a HA configuration for stability and reliability) connected to Artifactory, all those external libraries, as well as internal builds and all the ensuing dependencies are deep-scanned for vulnerabilities and license violations.

Through Xray’s flexible “Watches,” you can configure policy violations to be triggered, based on a wide range of parameters, as early on as the developer’s IDE. You can stop a CI build process, or even get notified about newly discovered vulnerabilities in components that have already been deployed to production systems. Once your builds have been scanned, you can store them in Artifactory along with exhaustive build information for full traceability. Then, Distribution can collect your builds and associated components into release bundles and have them replicated to Artifactory Edge nodes for final delivery to the compute edge. Throughout this process, you can use the Insight feature of Mission Control to monitor and continuously optimize the whole pipeline into peak performance. With all the components federated into the same circle of trust, you will be able to access them all with single-sign-on provided by Access.



There are different ways that the Platform can help your organization. There is no one “perfect” DevOps solution that fits all. It is incumbent on DevOps practitioners to closely examine their systems and customize them with the tooling that builds the best foundation for their business needs so that everyone can concentrate on process and innovation. The following sections show how the flexibility of the Platform and its individual components can help build a system from scratch, or retrofit an existing infrastructure.

PLAN TO SUCCEED

Planning is an essential part of building a highly performant CI/CD pipeline. Responsible organizations will invest the time and resources to plan parameters such as timelines, features, GitHub repositories, testing, security reviews and deployment to name a few. Each team in the process builds on libraries built by other teams and using the Platform ensures consistency throughout the pipeline from developer through to the Edge nodes where you host your service.

Replication is a key factor to keeping teams synchronized. With companies spreading out their resources globally, the complete DevOps process may run through multiple geographically distant data centers over different cloud providers. The enterprise-grade replication capabilities in Artifactory can accommodate any plan for sharing binary resources across the globe. And using its built-in Groovy-based scripting language Mission Control makes it easy to implement any plan for replication, distribution and deployment of your binaries through the Platform.

CLOUD NATIVE SUPPORT

Today more and more companies don't have a single platform, and are moving toward a hybrid solution that combines on-prem and multiple cloud providers. The Platform can be deployed on-prem, on your own private cloud, or as a deployed SaaS solution on the cloud vendor of your choice and even spread across multiple regions. It can also be deployed in a hybrid environment combining multiple options.

The Platform has extensive support for cloud storage, which includes the option of managing massive scalable setups with minimum latency. The Platform can be deployed on any infrastructure, whether it is just a local VM or a Kubernetes Cluster using Docker installation of all the Platform components or as pre-configured Helm Charts. As demand increases, this capability makes scaling up, both horizontally and vertically much easier. As Artifactory serves as your private Docker and Helm registry, it is easy to pull Docker images and Helm Charts from the closest Artifactory or Edge node, in order to install your application on your Kubernetes Cluster in your production sites securely and with minimum latency.

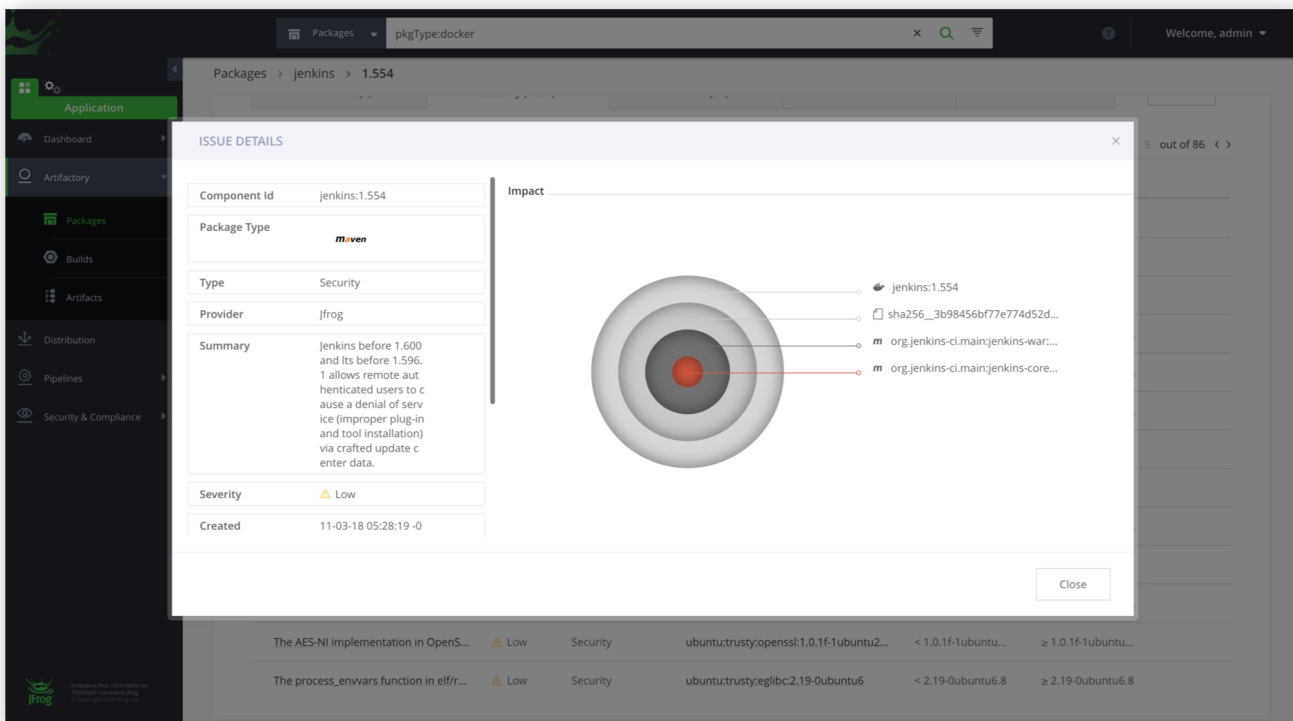
BUILT FOR SCALE

Artifactory can be optimized for the download of large binaries, such as Docker images, directly from the corresponding cloud storage. In order to scale out geographically spread downloads, Artifactory lets you leverage the CloudFront Content Distribution Network (CDN) and greatly accelerate the download speed for your artifacts.

SECURE YOUR SUCCESS

Cybercrime is booming, and a malware attack can cost an organization hundreds of millions of dollars to remediate. Responsible organizations must take exhaustive measures to protect their software. Some software organizations use source code scanners or peer review, but these methods have nothing to say about dependencies that developers use. Some companies will block external dependencies by using an “air gap,” disconnecting their network from the internet, but this makes the whole CI/CD cycle cumbersome, tedious and costly. Xray offers a much easier way to protect your code at every stage of the development cycle.

Through the JFrog IDE plugin, Xray flags dependencies with vulnerabilities right at the developer’s fingertips, before any affected code even gets into the CI/CD cycle. Once code gets pushed to Artifactory, it is scanned and indexed, along with any ensuing dependencies, and if vulnerabilities are detected at that stage, the affected components can be blocked for download. Similarly, once in the CI/CD pipeline, builds can be forced to fail as soon as vulnerable artifacts are detected. Xray has every stage of the pipeline covered. But blocking downloads or flagging an infected component is not enough. Finding infected dependencies and then determining which components in an organization are using them, directly or indirectly, can be costly and time-consuming. However, Xray has this covered too. Through recursive scanning, Xray can detect vulnerabilities hidden deep in a component’s dependency chain. It then performs an “impact analysis” to create a graph showing every component in the organization that is affected by that vulnerability. Once impact analysis is complete and the component graph stored, determining the impact of new vulnerabilities can be determined very quickly by traversing the component graph.



JFrog Xray - Docker Image Scan Revealing a Security Vulnerability

DELIVER SUCCESSFULLY

Your DevOps tool chain cannot end until your software is in the right place, ready for consumption by data centers, devices, points of sale etc. Distribution together with Artifactory Edge nodes ensure your software releases are where they need to be, anywhere in the world. This can be your own Datacenter, public cloud, Kubernetes cluster, any physical customer site or any hybrid mixture of these.

As a simple example, consider another common trend in DevOps, Microservices. You can run your CI system to build your Docker images and store them in Artifactory as your secure, private Docker registry. You promote your Docker images through the different gates as its quality gets verified until it reaches your “Production” Docker registry. Distribution can then replicate your certified Docker image, along with its corresponding Helm chart (stored in an Artifactory Helm repository) and any other artifacts that comprise a “release” to an Edge node that is located near the Kubernetes cluster that your release is destined for. Once your Docker image, its Helm chart and any other associated artifacts are on the Edge node, your release is ready for deployment to Kubernetes.

COMMON USE CASES FOR THE JFROG PLATFORM

MULTIPLE STORES OR LOCATIONS

One common use-case is a scenario where there are many stores and/or remote locations that need to consume the new updates for in-store devices. The stores can sometimes have connectivity problems - poor bandwidth and latency issues due to their remote locations. Some stores might not always be connected, and in this case these stores will be the “edge” locations and the Platform will distribute releases and software updates from the main Artifactory instance to these edge locations. The devices in the stores will be able to pull these updates from the internal network and be updated at all times.

DISTRIBUTION OF DIFFERENT RELEASE VERSIONS FROM MANY DIFFERENT TECHNOLOGIES

There can be a challenge to achieve distribution of different release versions from many different technologies to external customers, partners and vendors. Each one requires a different version of a release and there needs to be clarity on what was distributed to whom. There is often a large deployment process requiring dedicated permissions and entitlement models. The Platform can be set up to have satellites of Artifactory to handle this, in order to reduce the load on the production environment.

MANAGING REMOTE PRODUCTION SITES

This use case is the challenge of managing remote production sites, specifically when using Cloud Native applications such as Kubernetes. In this case, applications and services are hosted in distributed system environments that are capable of scaling to tens of thousands of self-healing, multi-tenant nodes. Those must be packaged in containers and managed dynamically. In this type of environment, each node needs to pull images (from Artifactory) in order to run the cluster. If there is a need to keep on pulling those images from the main Data Center there can be some issues:

- This is not scalable for runtime
- Latency and performance problems
- It creates a heavy load on the main site

(If you keep pulling large images from tens or hundreds of clusters over and over again)

Instead, the Platform can distribute those images to multiple edge locations and the Kubernetes clusters will just pull from them.

GLOBAL COMPANY WITH GEOGRAPHICALLY DISPERSED DEVELOPERS

A global company with many developers spread around the world, wants to reduce latency when accessing artifacts and builds. A common topology is a full mesh one where multiple Artifactory clusters are available in different geographical locations, where a geo-dns is responsible to route the request to the closest cluster. All the clusters are identical with their data and their configuration, so users will not know which Artifactory it hits. In this case, Access federation is a must to synchronize all the security entities between all clusters. Every user and every permission target that will be created in one cluster, will automatically be created in all the others. Combining with full-mesh replication, we make sure every developer or machine will be able to pull anything in real-time with minimum latency.

SUMMARY: LET THE JFROG PLATFORM TAKE CONTROL OF DEVOPS

A solid DevOps foundation is essential to a company's success and this is provided by the different components of the JFrog Platform. Artifactory, managing all of an organization's binaries, facilitates short release cycles. Xray ensures the security and compliance of software components in a release. JFrog Pipelines manages CI/CD automation and orchestration, with a centralized command and control capability for all your DevOps Pipelines. Distribution and Artifactory Edge nodes allow the efficient transfer of releases directly to the compute edge for deployment, and Mission Control together with Insight, manage, monitor, configure and adjust the complete tool chain for optimal performance.

Once DevOps is under control, an organization can focus on the innovation that drives its business without having to worry about infrastructure. The JFrog Platform lets you take control of your DevOps needs.

REFERENCES FOR FURTHER READING

[Getting started with the JFrog Platform \[Wiki\]](#)

[The DevOps Future is here – Discover the New JFrog Platform \[Webinar\]](#)

[GigaOm Radar for Enterprise CI/CD \[Report\]](#)

[How Box reduced microservice build times by 90x using Artifactory, Docker and Kubernetes \[Video\]](#)

[Devops Tools Accelerating Software Releases \[Webpage\]](#)