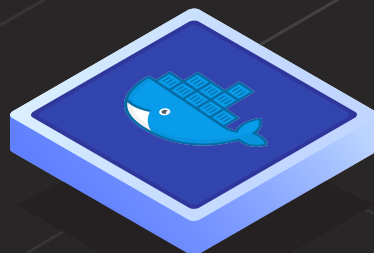




# SECURE CLUSTERED HA DOCKER REGISTRIES

## OVERCOMING THE CHALLENGES OF USING DOCKER IN DEVELOPMENT AND PRODUCTION

WHITE PAPER



# TABLE OF CONTENTS

---

<b>Executive Summary</b> .....	<b>3</b>
<b>Introduction</b> .....	<b>6</b>
1. Reliable, Consistent and Efficient Access to Remote Docker Registries .....	<b>6</b>
2. Reduce Network Traffic and Optimize Builds .....	<b>7</b>
3. Full Integration With Your Build Ecosystem .....	<b>8</b>
4. Security and Access Control .....	<b>9</b>
5. Vulnerability Detection and Remediation .....	<b>9</b>
6. Distribute and share images across your organization .....	<b>10</b>
7. Smart Search and Artifactory Query Language (AQL) .....	<b>11</b>
8. User Plugins .....	<b>11</b>
9. System stability and reliability with Artifactory High Availability .....	<b>12</b>
10. Maintenance and Monitoring .....	<b>13</b>
11. JFrog Enterprise+: A Universal, End-to-End Solution For All Binaries .....	<b>13</b>
<b>Summary</b> .....	<b>14</b>

# EXECUTIVE SUMMARY

Docker has become the defacto standard for containerization, however, the technology does not adequately address several needs of software development organizations. For example, how do you distribute and share images across an organization, and if you solve that one, how then do you control access to those images? Or, how do you overcome issues of network connectivity and latency that hinder the progress of developers and automated build processes alike?

The answer to all of these questions is JFrog Artifactory, the universal repository manager. Artifactory is a secure, robust Docker registry; a single access point to manage and organize your Docker images. With full support for the Docker registry API, Artifactory works transparently with the Docker client.

The following table shows how JFrog Artifactory removes the overhead connected to managing your Docker images.

## Reliable, Consistent and Efficient Access to Remote Docker Registries

**Remote public registries such as Docker Hub are critical for development and must be available at all times. If these become inaccessible due to outage or network issues, development and builds grind to a halt.**

Artifactory mitigates your dependence on Docker Hub and other external resources by caching remote Docker images in a **remote repository**; a local cache that serves as a proxy to the remote resource. This overcomes any problems with accessibility that stem from network issues or the resource going down.

## Reduce Network Traffic and Optimize Builds

**Many developers and build machines/CI servers constantly downloading components can generate a lot of network traffic and slow down builds.**

Once Artifactory has downloaded an image, it is locally available to all developers and build tools/CI servers resulting in reduced network traffic and quicker build processes.

## Full Integration With Your Build Ecosystem

**However your build ecosystem is constructed, your build systems, running several builds a day, must have easy access to your images.**

CI systems resolve dependencies through Artifactory, and also deploy builds to the corresponding Docker registry in Artifactory. Since Artifactory stores exhaustive build information, running builds through Artifactory enables fully traceable builds and allows you to compare builds with built-in "Diff" tools. Artifactory also simplifies release management through a series of simple settings like staging, build promotion and VCS tagging to fully automate the release management process.

## Security and Access Control

**Every organization needs to implement security policies so that users can only access internal and external resources that they are authorized to use.**

Artifactory provides security and access control at several levels. Using "includes" and "excludes" patterns, teams and permissions, and integration with common access protocols such as LDAP, SAML and Crowd, Artifactory provides fine-grained access control, from restricting complete repositories down to restricting a single artifact, and from a group of any size down to a single developer.

## Vulnerability Detection and Remediation

The use of open source components exposes you to security vulnerabilities and license compliance issues both in development and production systems.

Artifactory's unique and tight integration with **JFrog Xray** enables you to prevent issues and vulnerabilities from getting into your software by intervening at all stages of the pipeline. Xray integrates with popular IDEs and CI servers to detect vulnerabilities during development and CI builds. Even after your Docker images are deployed to production, Xray continues to scan them and provide policy violation alerts for new vulnerabilities that have been reported. Through deep impact analysis, Xray builds a component graph and identifies all Docker images in your organization that are affected by a vulnerability or policy violation.

## Distribute and Share Images Across Your Organization

To maximize reuse of your codebase, and to ensure your developers work under the same environment, you want an easy way to share images within your team and across your organization.

Using **local repositories**, Artifactory gives you a central location to store your internal Docker images so that all teams can access any artifact from a single URL. To support distant teams, Artifactory offers a variety of replication capabilities that let you synchronize any number of globally distributed sites with the flexibility to accommodate nearly any **multi-site topology**.

## Smart Search and Artifactory Query Language (AQL)

Between the many images that different developers download, and the internal images you develop within your own organization, finding something specific can become quite complex.

Artifactory offers a variety of options for search, from simple name search to common built-in search functions like "latest version search" as well as a **search by checksum** that uses Artifactory's unique **checksum-based storage**.

**Artifactory Query Language** (AQL) takes search to new levels offering a simple way to formulate complex queries based on any number of parameters.

## User Plugins

While Artifactory provides an extensive set of features to manage Docker images, it's impossible to accommodate all the requirements that different organizations may have.

User plugins extend the Artifactory REST API providing a simple way to implement complex behavior. This gives you enormous freedom to support virtually any custom requirement in your workflow.

## High Availability

As a mission critical component in your organization, any downtime in your repository manager can have severe consequences to your organization's productivity.

Artifactory can be deployed in a **high availability** configuration with two or more servers that can take your uptime to levels of five-nines availability.

## Maintenance and Monitoring

**The number of Docker images you generate can grow very quickly. Without proper management, your systems can quickly get clogged with old and irrelevant images.**

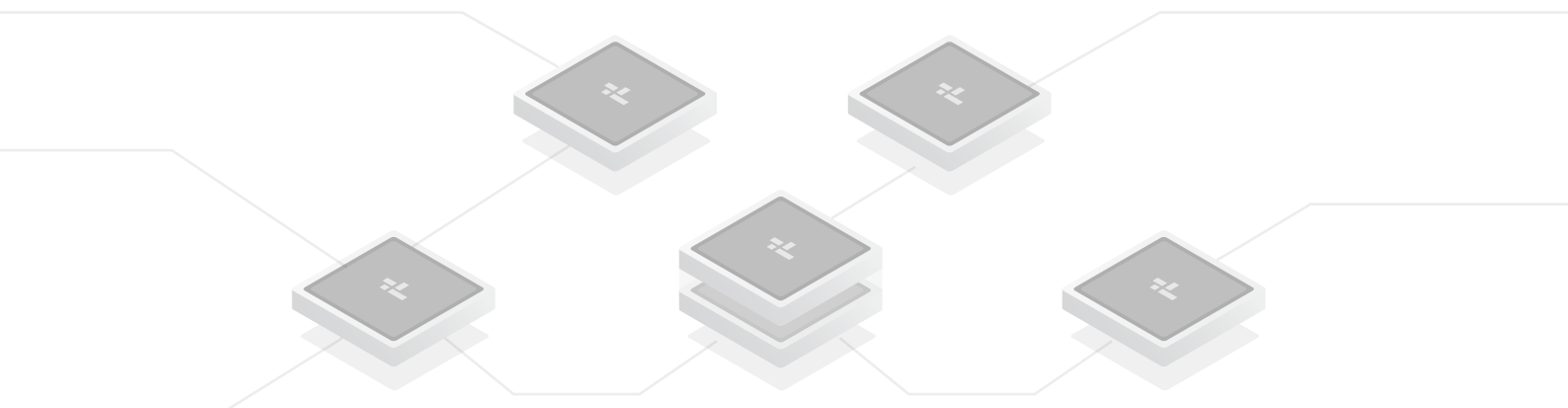
Artifactory keeps your system free of clutter with automatic, scheduled cleanup processes, monitoring and controlling disk space usage, and the ability to define “watches” on your most critical Docker images.

## JFrog Enterprise+: A Universal End-to-End Solution for All Binaries

**In order to be competitive in today's markets, companies must continuously improve their software in terms of quality, consistency, security and global reach. To meet that challenge, companies must have sound DevOps practices in place.**

**JFrog Enterprise+** is an end-to-end DevOps platform to create, manage, scan, distribute and deploy software with ease. It includes Artifactory to provide advanced management of Docker images and other binary artifacts created with any of today's major development technologies, and JFrog Mission Control, JFrog Xray, JFrog Distribution and Artifactory Edge nodes to cover artifact maturity, security and vulnerability protection, release management, analytics and distribution.

JFrog Artifactory overcomes the inhibitors to taking Docker to production through capabilities such as promoting images as immutable components through the development pipeline to production. Features like high availability and cloud-based storage provide the stability, scalability and security that enterprises require. And as a universal artifact repository, Artifactory provides the same native level support for all major development technologies giving it a central role in any development ecosystem.



# INTRODUCTION

Container technology is not new. It has been around since the year 2000 with **freeBSD jails** that had access to the operating system kernel and a few other system resources. Then **Docker** came on the scene and emerged as the “King of Containers” with more and more enterprises adopting Docker technology to run applications in data centers, on IT infrastructure and developer laptops alike. But just as component-based development spawned by the open source revolution created challenges in managing components within the enterprise, the ever-increasing number of Docker images used by an organization presents similar challenges:

- How do you distribute and share images within your organization?
- How do you manage who can access an image?
- How do you make it easy to find images?
- How do you support a variety of policies for managing images?
- How do you ensure your images are always available for use?

The answer to all of these questions is Artifactory, a Binary Repository Manager that functions as a single access point through which you can manage all of your Docker images. With full support for the Docker Registry API, Artifactory is transparent to the Docker client and can therefore boost your organization’s productivity by removing the overhead connected to managing your applications and images that are developed to run in Docker containers.

## 1. RELIABLE, CONSISTENT AND EFFICIENT ACCESS TO REMOTE DOCKER REGISTRIES

When developing your Docker images, many of dependencies will be other images hosted on remote Docker registries. These may be public resources such as Docker Hub, or even internal resources located at remote sites such as a private Docker registry on JFrog Bintray. These remote resources are critical for your development efforts and must be available at all times, however events beyond your control may render them unavailable. Public resources such as Docker Hub may go down and connectivity issues may prevent access to other remote resources beyond your internal network. Without access to Docker hub and other remote Docker registries, development and builds grind to a halt.

JFrog Artifactory is an intermediary between developers and remote Docker registries and fully implements the Docker Registry API specification. This allows Artifactory to proxy any public or private Docker registry such as Docker Hub, JFrog Bintray, or other private Docker registries, and treat them like any other **remote repository**. When an image is first downloaded, Artifactory stores it in a local cache. Upon receiving subsequent requests for the image, Artifactory performs a smart checksum search for it, and if it has already been downloaded, then the locally cached copy is provided. Therefore, each image is only downloaded once and is then locally available to all other developers in the organization. This reduces network traffic and effectively screens you from any issues with the network, Docker Hub or any other remote Docker registry providing you with consistent and reliable access to remote Docker images.

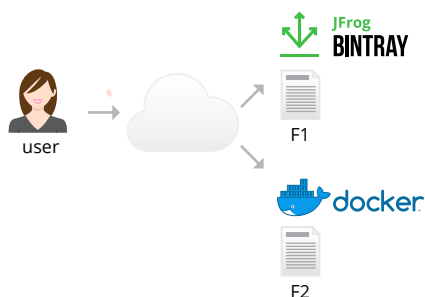
### Remote Repositories

A remote repository serves as a caching proxy for a Docker registry managed at a remote site such as Docker Hub or JFrog Bintray. Images are stored and updated in remote Docker Registries according to various configuration parameters that control the caching and proxying behavior.

[LEARN MORE >](#)

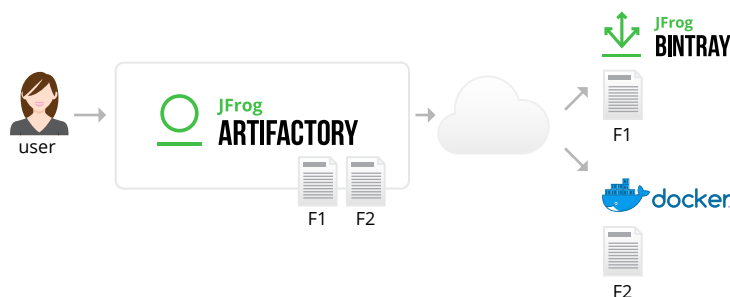
### Before:

Developer accesses Docker Hub, JFrog Bintray and other remote Docker registries through the internet.



### After:

First time an image is accessed, Artifactory downloads it. From then on, it is available locally to all developers.



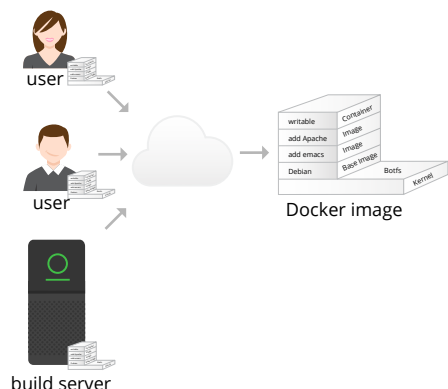
## 2. REDUCE NETWORK TRAFFIC AND OPTIMIZE BUILDS

Since much of your code is likely to be assembled rather than built, you want to make sure that your usage of Docker images downloaded from external resources is optimized. It makes no sense for two (or two hundred) developers using the same image to download it separately. In addition to reliability, another benefit of remote Docker registries is reduced networking. Once an image has been downloaded, it is then locally available to all other developers in the organization (thus reducing network traffic). Naturally, this is all transparent to the individual developer. Once images are accessed through Artifactory, the developer can get on with what she does best and leave the rest to Artifactory.

If we look at network traffic from the point of view of a build server, the benefits are clear. A typical project may depend on tens if not hundreds of images from external resources. For the server to build these projects, all remote Docker images must be available to the server environment. Downloading all those required images may generate Gigabytes of data traffic on the network which takes a significant amount of time delaying the build process. By caching remote Docker images locally, the build process incurs much less networking and is, therefore, much quicker.

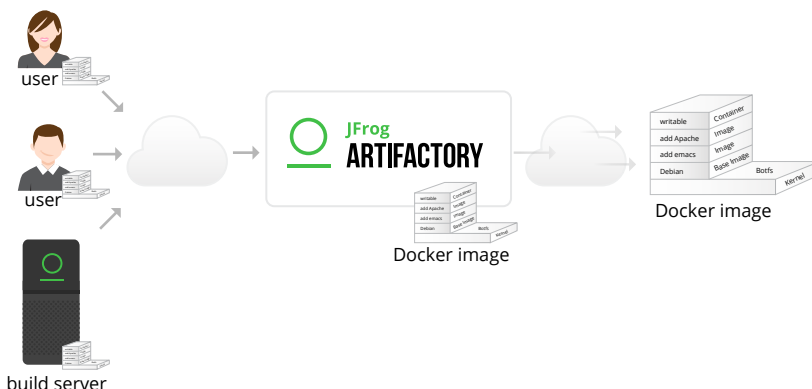
### Before:

Different users and build servers download the same file.



### After:

First time a file is accessed, Artifactory downloads it. From then on, it is available locally to all other users and build servers.



### 3. FULL INTEGRATION WITH YOUR BUILD ECOSYSTEM

While it's important to make it easy and efficient for your developers to access Docker images, it's even more important for your build systems which may be running builds many times a day.

Through a set of plugins, Artifactory provides tight integration with popular CI systems available today such as Jenkins, Bamboo, and TeamCity. These systems use Artifactory to supply artifacts and resolve dependencies when creating the build and also as a target to deploy build output to the corresponding local Docker registry. One of the main benefits of running builds through Artifactory is fully reproducible builds. Artifactory stores exhaustive build information including specific artifact versions, modules, dependencies, system properties, environment variables, user information, timestamps and more. With this information, it is easy to faithfully reproduce a build at any time. Moreover, with built-in "Diff" tools you can compare builds, and know exactly what changes were introduced from one version to another. These capabilities can be invaluable when trying to track down bugs that were reported in specific versions released.

Artifactory also simplifies release management. A series of simple settings configure things like staging, build promotion, VCS tagging and more essentially automating the release management process. Through Artifactory's support for multiple secure, private Docker registries, you can set up a promotion pipeline allowing you to deploy immutable Docker images to your production systems and run them with confidence.

But what happens if you are using cloud-based CI systems where you can't apply plugins? In that case, you can use **JFrog CLI** to resolve dependencies and upload build output to Artifactory providing the same level of build automation. Essentially, since Artifactory is platform agnostic, it can be integrated with generic tools across all the build ecosystems within your organization. Finally, once your builds are automated, Artifactory will keep your system free of clutter by cleaning up old builds according to your organization's maintenance policies.

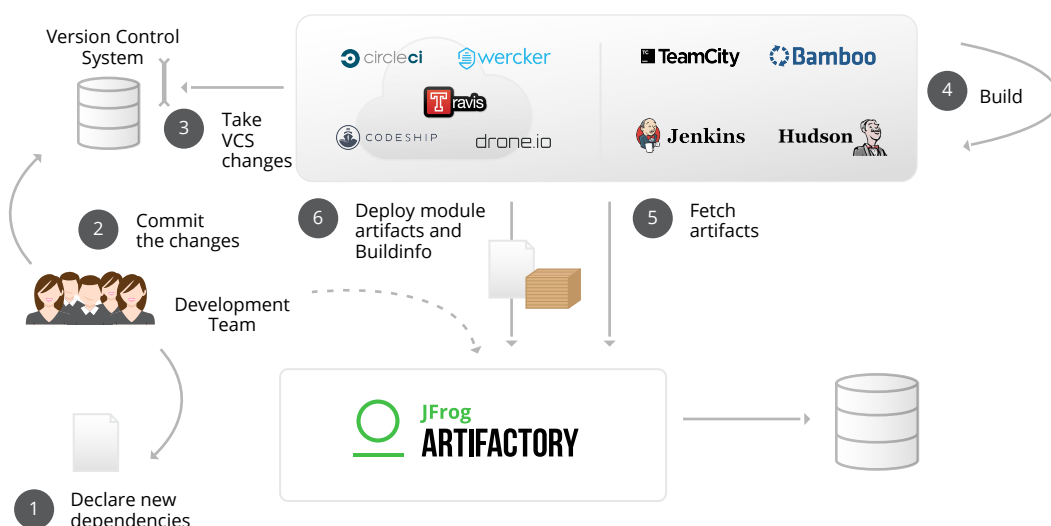
**Confidently take Docker to production by setting up promotion pipelines within Artifactory.**

[LEARN MORE >](#)

#### JFrog CLI

JFrog CLI is a compact and smart client that provides an interface to access JFrog products simplifying your automation scripts and making them more readable and easier to maintain. JFrog CLI accesses JFrog Artifactory through its REST API making your scripts more efficient and reliable in several ways: concurrent uploads and downloads let builds run faster, checksum optimization avoids redundant file transfers and wildcards and regular expressions give you an easy way to specify files for upload or download.

[LEARN MORE >](#)





## 4. SECURITY AND ACCESS CONTROL

Every organization needs to implement security policies so that developers can only store images in authorized locations, and access images that they are authorized to use.

Artifactory offers a complete security solution to provide any number of secure, private Docker registries. As a first line of defense, Artifactory lets you use naming patterns to define “Excludes” and “Includes” for access so you can control which packages can even be cached in any particular remote Docker registry. Then, you can assign different sets of permissions to users and groups to control access to each Docker registry. In this way, for example, you can allow developers to deploy a release candidate to a local QA Docker registry, but only allow authorized QA staff, who have ensured that the candidate has passed the required quality gates, to move it to the “releases” registry from which production images are pulled. Finally, you can even use Artifactory’s integration with LDAP, Active Directory, SAML, Crowd and others to control access to your servers.

Artifactory provides a level of security and access control that is unmatched in the market and effectively replaces all other solutions to let you manage any number of secure and private Docker registries within your organization.

## 5. VULNERABILITY DETECTION AND REMEDIATION

The use of open source components exposes you to security vulnerabilities and license compliance issues. Even dependencies deemed “safe” during development can later be discovered to contain security flaws, and by that time, your Docker image may already be in production systems exposing your products and services to security breaches.

Artifactory’s unique and tight integration with **JFrog Xray** enables you to prevent issues and vulnerabilities from getting into your software by intervening at all stages of the pipeline. By integration with popular IDEs, Xray can notify developers of suspicious dependencies as soon as they are included in their projects. Once code is committed and the CI process takes over, Xray scans builds and fails them if vulnerabilities are detected in any of the build artifacts and dependencies. And even after Docker images have been deployed to production systems, Xray continues to scan them and can alert administrators if new vulnerabilities have been found. Through deep impact analysis, Xray not only notifies you of vulnerabilities in your Docker images, but identifies the exact dependency containing the vulnerability and indicates all other images in your organization that contain that infected dependency. When available, Xray will even notify you if there is a later version of the dependency in which the vulnerability has been fixed.

### JFrog Xray

JFrog Xray provides continuous security through universal artifact analysis. It works with JFrog Artifactory to analyze software components, and reveal a variety of vulnerabilities at any stage of the software application lifecycle. By scanning binary components and their metadata, recursively going through dependencies at any level, JFrog Xray provides unprecedented visibility to reveal vulnerable components lurking anywhere in your organization.

[LEARN MORE >](#)

## 6. DISTRIBUTE AND SHARE IMAGES ACROSS YOUR ORGANIZATION

To maximize reuse of your codebase, and to ensure your developers work under the same environment, you want an easy way to share images within your team and across your organization.

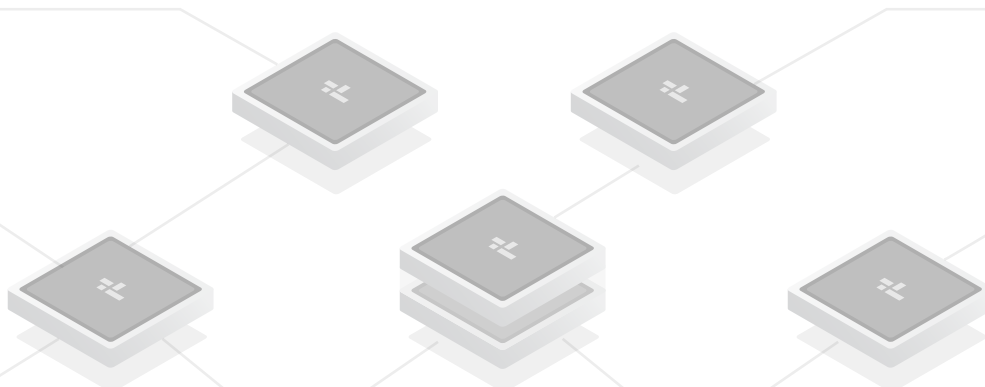
Using **local repositories**, Artifactory gives you a central location to deploy and store your images - effectively, a private Docker registry that can replace Docker Hub and Docker Hub Enterprise. When all teams know that any image can be accessed from a single URL, managing access to images between the different teams becomes very easy. But what if you want to share your images with colleagues who are in geographically remote sites of your organization?

To allow sharing between different sites, you can replicate your repositories to another instance of Artifactory which is outside of your local network. Replicated repositories are automatically synchronized with their source periodically, so that your images can be made available to different teams wherever they may be located around the world. With the capability for multi-push replication and event-based pull replication, Artifactory lets you synchronize any number of globally distributed sites and has the flexibility to accommodate a wide variety of **multi-site topologies**.

### Local Repositories

Local repositories are physical, locally-managed repositories that are typically used to deploy internal and external releases as well as development builds. By storing all your images in local repositories, they can be made available from a single access point across your organization from one common URL.

[LEARN MORE >](#)



## 7. SMART SEARCH AND ARTIFACTORY QUERY LANGUAGE (AQL)

Working with third-party images can get quite complex. Between the many images that different developers download, and the internal images you develop within your own organization, finding something specific can become quite a challenge.

Artifactory provides you with flexible search capabilities both through the UI, and using the extensive REST API. You can find images based on any combination of inherent attributes such as name, version, timestamp, **checksum** and more. Artifactory also provides some common built-in searches. For example, you can ask Artifactory for the “latest” version of any image without having to specify a particular build number. Artifactory knows how to compare all the different versions of an image in any of its repositories and provide the latest one available. You can also assign any set of custom properties to your images, which can later be used for search. For example, you can tag all the specific versions of images used in a product release with a “released” property to easily reproduce the released version later on. But the full power of search comes with the complete flexibility of AQL. Using **AQL**, you can define search queries to any level of complexity needed to extract just the right images you are looking for.

## 8. USER PLUGINS

While Artifactory provides an extensive set of features to manage images, it's impossible to accommodate all the requirements that different organizations may have. Enter user plugins.

User plugins present a long list of entry points which effectively extend the Artifactory REST API providing a simple way to implement complex behavior. This gives you enormous freedom to support virtually any custom requirement in your workflow including scheduling tasks, managing security and authentication, deployment, maintenance and cleanup and more. To keep things simple, user plugins are written as Groovy scripts and have a simple DSL to wrap them as closures within the extension points. The plugins can be changed and redeployed on-the-fly, and can even be debugged - all from within your favorite IDE.

### Checksum-based search

Searching for an image by its checksum is a powerful feature supported by Artifactory through its unique checksum-based storage. Even if an image has been renamed, moved or deployed outside of your organization, you can trace it back to the original version. Simply run the image through a checksum tool (SHA-256, SHA-1 and MD5 are supported) and run a “Checksum” search in Artifactory to retrieve the original version.

[LEARN MORE >](#)

### Artifactory Query Language (AQL)

AQL is flexible query language that offers a simple way to formulate complex queries to search through your repositories using any number of search criteria, filters, sorting options and output fields. It takes full advantage of the database underlying Artifactory's unique architecture and gives you unlimited degrees of freedom to formulate exactly the right query to find those very specific packages you are searching for. This is something that no other Binary Repository can offer.

[LEARN MORE >](#)

## 9. SYSTEM STABILITY AND RELIABILITY WITH ARTIFACTORY HIGH AVAILABILITY

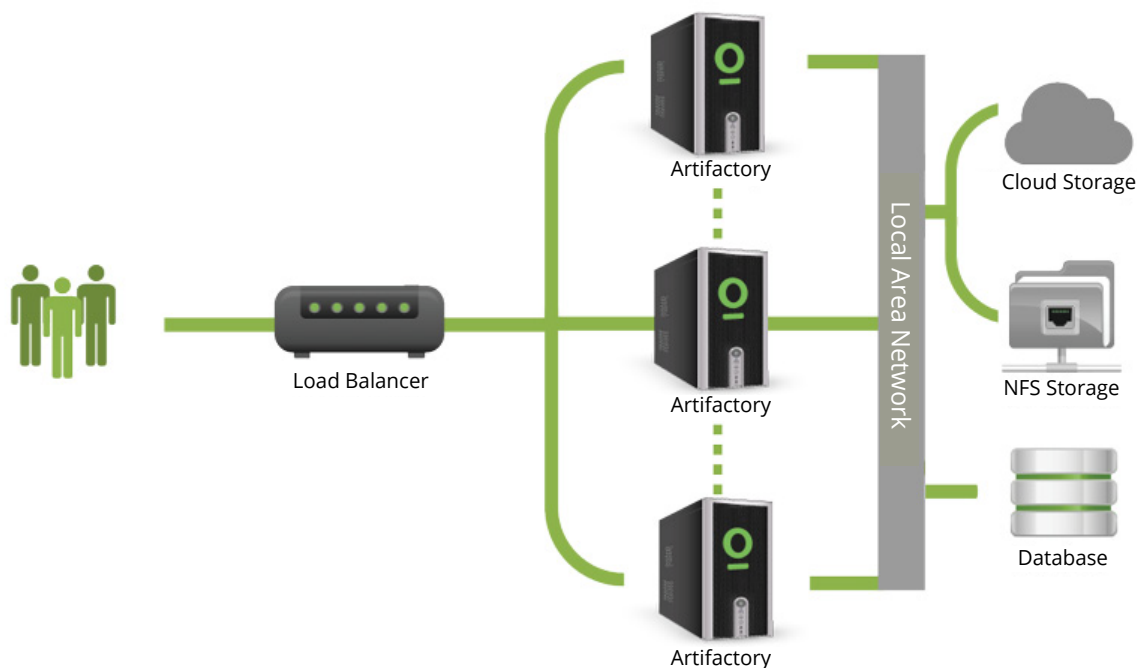
Playing such a central role in the management of images, your Binary Repository Manager can become a mission-critical component of your organization. Any downtime can have severe consequences to your productivity, and you need to ensure developers can access your Docker registries at all times.

Artifactory supports a **High Availability** network configuration with a cluster of 2 or more Artifactory servers on the same Local Area Network. A redundant network architecture means that there is no single-point-of-failure, and your system can continue to operate as long as at least one of the Artifactory nodes is operational. This maximizes your uptime and can take it to levels of up to "five nines" availability. Moreover, your system can accommodate larger load bursts with no compromise to performance. With horizontal server scalability, you can easily increase your capacity to meet any load requirements as your organization grows. Finally, by using an architecture with multiple servers, Artifactory HA lets you perform most maintenance tasks with no system downtime.

### High Availability Systems

Systems that are considered mission-critical to an organization can be deployed in a High Availability configuration to increase stability and reliability. This is done by replicating nodes in the system and deploying them as a redundant cluster to remove the complete reliability on any single node. In a High Availability configuration there is no single-point-of-failure. If any specific node goes down the system continues to operate seamlessly and transparently to its users through the remaining, redundant nodes with no down time or degradation of system performance as a whole.

[LEARN MORE >](#)



## 10. MAINTENANCE AND MONITORING

With the growth of automated software delivery systems, the number of images you generate can grow very quickly. Without proper management, your systems can quickly get clogged with old and irrelevant images.

Artifactory keeps your system organized and free of clutter with automatic, timed cleanup processes. With a few simple settings, you can schedule tasks to clean up old builds and unused images. You can set restrictions on disk space and monitor its usage, or define “watches” to receive an alert whenever there is a change to your most critical images. And with an extensive REST API, Artifactory can support virtually any rule-based cleanup protocol you would want to implement in your organization.

## 11. JFROG ENTERPRISE+: A UNIVERSAL, END-TO-END SOLUTION FOR ALL BINARIES

The JFrog Enterprise+ platform was designed to meet the growing needs of companies to develop and distribute software. JFrog Enterprise+ provides DevOps with the tools needed to create, manage and deploy software with ease and includes the following components:

**JFROG ARTIFACTORY** is the mothership of JFrog Enterprise+. It was designed from the ground up to fit in with any development ecosystem. Uniquely built on checksum-based storage, Artifactory supports any repository layout and can, therefore, provide native-level support for any packaging format. Essentially, regardless of the packaging format you are using, Artifactory can store and manage your binaries, and is transparent to the corresponding packaging client. The client works with Artifactory in exactly the same way it would work with its native repository. For example, if you are working with Docker images, Artifactory proxies Docker Hub and other remote Docker resources, lets you store your own images in local Docker registries, and works transparently with the Docker client. Similarly for npm, Vagrant, NuGet, Ruby, Debian, YUM, Python and more.

**JFROG MISSION CONTROL** is a single access points for managing all JFrog services and CI servers in your global DevOps tool chain. JFrog Insight is a DevOps analytics engine added to JFrog Mission Control as a part of the Enterprise+ Platform to measure, analyze and optimize your distribution flow.

**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 to production.

**JFROG DISTRIBUTION** is an on-premise tool that lets you orchestrate software distribution between two Artifactory instances or from Artifactory out to multiple Artifactory Edge nodes.

**JFROG ARTIFACTORY EDGE** 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.

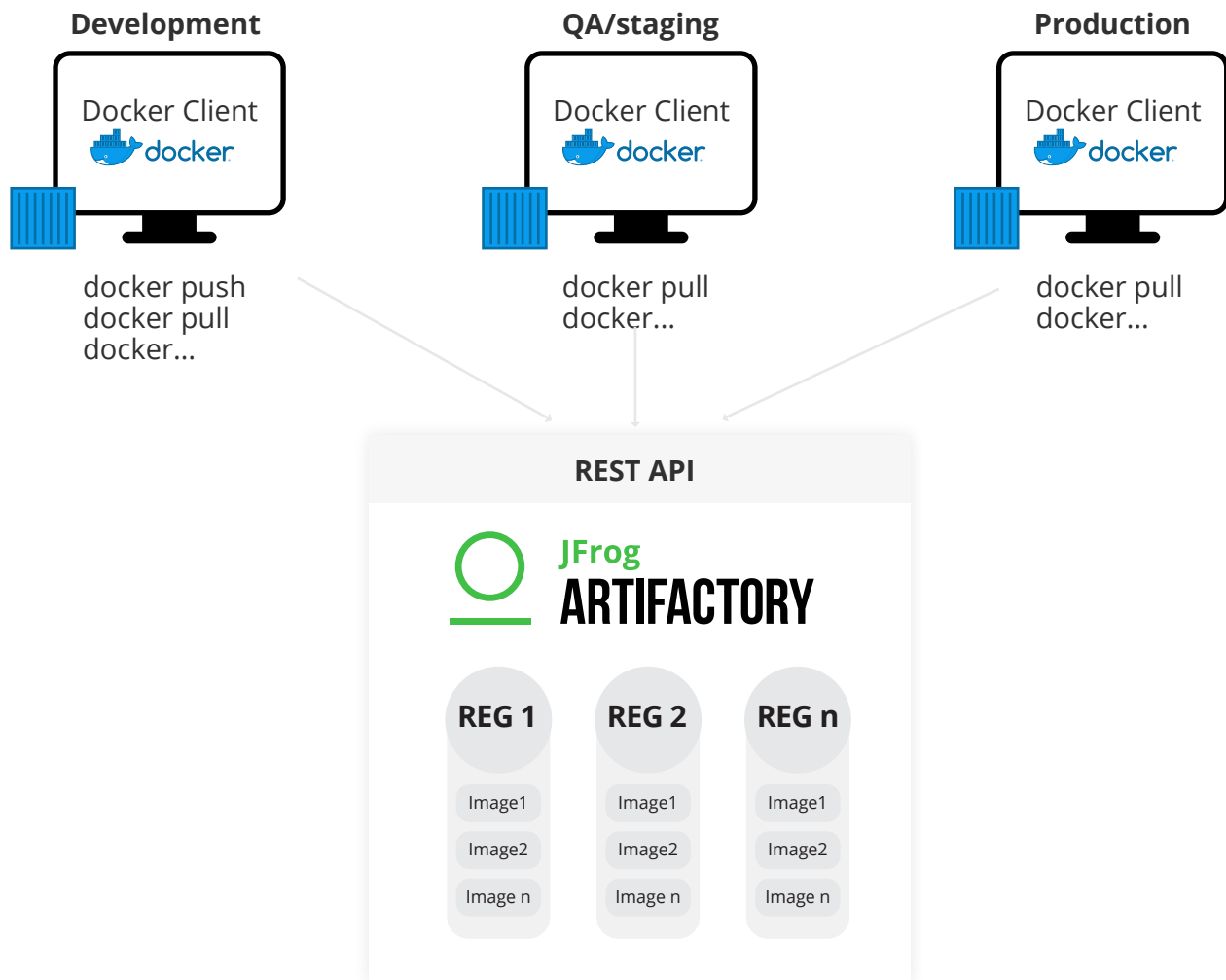
**JFROG ACCESS** provides a common authentication and authorization infrastructure for all JFrog products to manage security entities. With the launch of JFrog Enterprise+, JFrog Access can be used to federate different JFrog services into a single “circle of trust” enabling single-sign-on.

**JFROG ENTERPRISE+** provides the solid DevOps foundation that is essential to a company’s success and allows software organizations to focus on the innovation that drives their business without having to worry about infrastructure.

[LEARN MORE ABOUT THE JFROG ENTERPRISE+ PLATFORM](#)

[READ THE WHITE PAPER>](#)

## SUMMARY



Container technology has come a long way since freeBSD jails with Docker becoming commonplace in many development organizations. However, it is still struggling to gain widespread acceptance in production systems. JFrog Artifactory overcomes the inhibitors to taking Docker to production through capabilities such as promoting images as immutable components through the development pipeline to production. Features like high availability, cloud-based storage and image scanning and analysis provide the stability, scalability and security that enterprises require. And as a universal artifact repository, Artifactory provides the same native level support for all major development technologies like Maven, NPM, NuGet and Debian giving it a central role in any development ecosystem.

For more information on JFrog Artifactory, please contact us at [info@jfrog.com](mailto:info@jfrog.com).