



# SECURE PRIVATE VAGRANT BOXES AND MORE WITH A BINARY REPOSITORY MANAGER



White Paper

# Introduction

The importance of a uniform development environment among team members can't be overstated. Bugs stemming from differences in environment variables, versions of infrastructure and other causes that are extraneous to the code itself, are often the most difficult to track down and resolve. Any developer who has ever heard another developer say, "...but it works on my machine" understands this. But setting up a machine to be just like others can be a complex process. Configuring operating systems, databases, development workspaces and a host of cloud services can sometimes drag on for a hair-pulling week or more.

These are exactly the issues that Vagrant solves.

Since Vagrant joined the virtualization scene, these are problems of the past. Now, all a developer has to do is type vagrant up with the right box, and start coding within minutes. But the question remains, what is the right box? A development organization may need many different boxes.

Teams may have different requirements for the environment on which their project is being developed, and frequently, a version of the team's source code may reside in a Vagrant box together with the infrastructure. Consequently, an organization needs to maintain and manage hundreds of boxes at a time. Vagrant does offer private boxes on Atlas, Vagrant's repository in the cloud, but enterprises typically require a more secure environment for boxes in which the company's source code may end up.

Artifactory's support for Vagrant means you can easily setup local repositories for Vagrant boxes. This white paper presents the benefits of pointing your Vagrant client at Artifactory to replace Atlas as a source and target for Vagrant boxes in an enterprise software development organization.



# 01

## Distribute and share boxes 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 Vagrant Boxes within your team and across your organization.

Using **local repositories**, Artifactory gives you a central location to deploy and store software artifacts - effectively, a secure, private repository for your Vagrant boxes. When all teams know that any box can be accessed from a single URL, managing access to boxes between the different teams becomes very easy. But this may not be enough for global enterprises in which development teams may be spread out around the world.

To enable sharing boxes between geographically remote sites of an organization, 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 your boxes can be made available to different teams wherever they may be located.

### 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 >](#)



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

Artifactory uses a set of permissions you can configure to provide security and access control for Vagrant boxes. By defining appropriate users and groups, you can control where developers can access or deploy boxes, whether they can annotate metadata, or delete boxes and more. Artifactory also provides control over access to your servers through full integration with the most common access protocols such as LDAP, SAML, Crowd and others. By providing comprehensive security and

access control capabilities, Artifactory helps you manage your development process ensuring that developers can only access repositories which they are authorized to. For example, you can configure Artifactory to authorize your DevOps team to upload or remove specific boxes into a development team's Vagrant repository, while developers are only authorized to download boxes from the repository.

This fine-grained security and access control that Artifactory provides for local repositories is far greater than that currently offered by private boxes on Atlas. Artifactory effectively replaces Atlas to let you manage any number of secure and private boxes within your organization.



As Docker technology continues to evolve, its usage continues to grow. If you are not yet using Docker in your organization, it is likely you will do so soon. So now, in addition to managing Vagrant boxes, you also need to manage Docker images. But there's no need to onboard and maintain another tool. Artifactory is a fully-fledged Docker repository supporting all Docker Registry APIs. This allows the Docker client to work with Artifactory directly, presenting several benefits for enterprise Docker users.

Using local repositories, you can **distribute and share images** within your organization to make managing images between different teams easy. You can even replicate your Artifactory Docker repositories to remote instances of Artifactory to share images with colleagues in geographically distant sites.

Artifactory offers **fine-grained access control** to your organization's images with secure "docker push" and "docker pull" effectively providing **secure, private Docker repositories** that exceed the security offered by Docker Trusted Registry.

Using Artifactory, instead of private repositories on Docker Hub, removes any issues related to internet connectivity resulting in **reliable and consistent access to images**. And with Artifactory running in a **High Availability configuration** you get system stability and availability of your Docker images that is unmatched in the industry.

Artifactory's **smart search** makes it easy to find any Docker image stored in your system. Full support for the Docker Registry API supports basic search with the Docker client, but Artifactory offers much more. Built in searches answer common needs with single-click operations, custom properties provide the flexibility to meet a variety of specific needs, and Artifactory Query Language offers a simple way to formulate complex queries letting you find images based on any set of criteria.

Whether you're already on board with Docker or just evaluating how to introduce it to your organization, once you're using Artifactory to manage your Vagrant boxes, you're already covered for Docker images.

# 04

## Full integration with your build ecosystem

While it's important to make it easy and efficient for your developers to access binary artifacts, 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 a build, and also as a target to deploy build output to the corresponding local repository.

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 therefore 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.

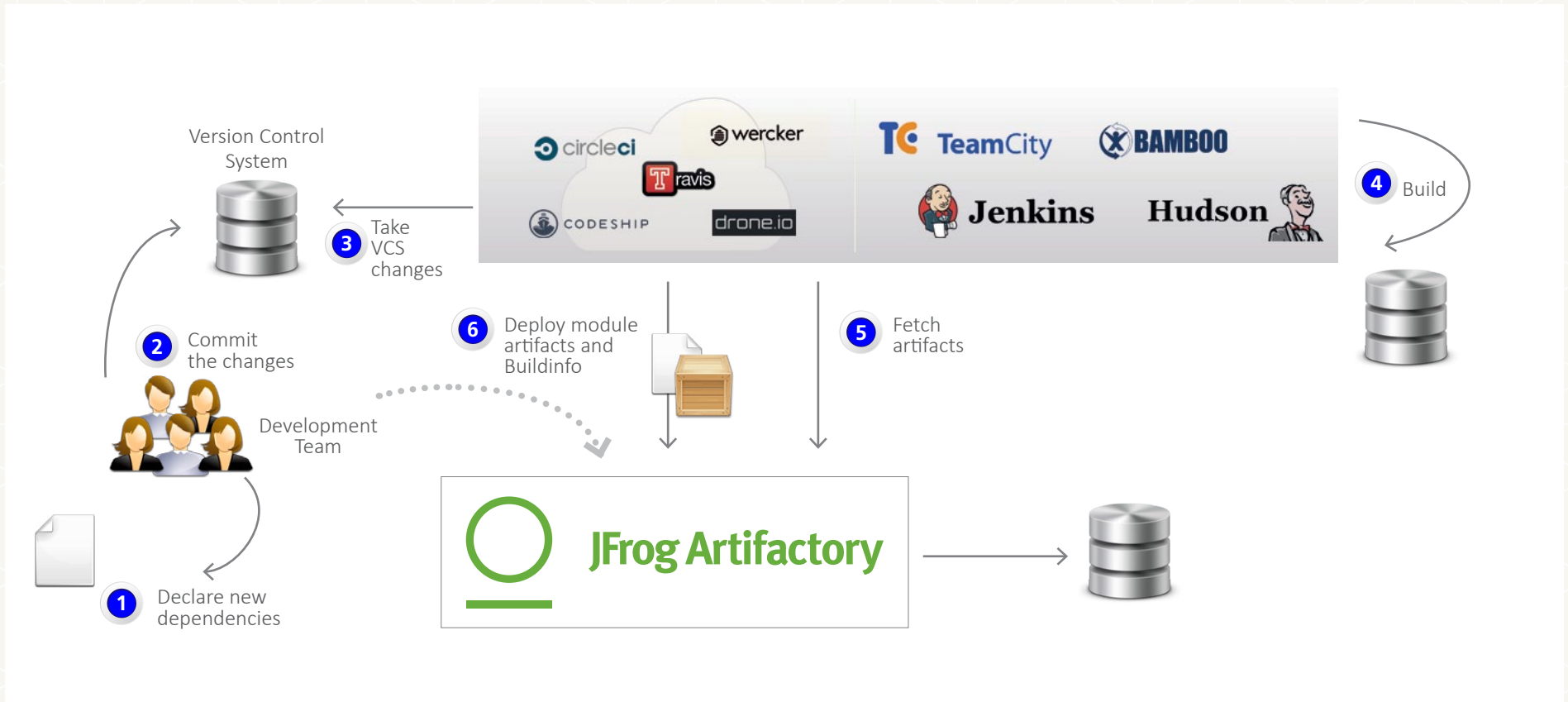
Artifactory also simplifies release management. A series of simple settings configure things like staging, build promotion, VCS tagging and more, essentially automating the release management process.

But what happens if you are using cloud-based CI systems where you can't apply plug-ins? In that case, Artifactory provides plug-ins directly for the build tools themselves (such as Maven and Gradle), which ultimately provides the same level of build information. 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.



# 04

## Full integration with your build ecosystem





# Smart search for boxes using custom properties and AQL

Given the multitude of boxes in your system, finding something specific can sometimes get quite complex.

Artifactory provides you with flexible search capabilities both through the UI, and using the extensive REST API. You can find boxes based on any combination of inherent attributes such as name, version, timestamp, **checksum** and more. For convenience, common searches come built-in with Artifactory. For example, you can ask for the “latest” version of any box without having to specify a particular build number. Artifactory knows how to compare all the different versions of a box in any of its repositories and provide the latest one available.

To provide more flexibility, Artifactory lets you define custom properties that you can assign to your boxes, and then use in your searches. For example, you could define a property to differentiate boxes used in your development, QA and production environments.

## Checksum-based search

Searching for a box by its checksum is a powerful feature supported by Artifactory thanks to 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 (both MD5 and SHA1 are supported) and run a “Checksum” search in Artifactory to retrieve the original version.

[Learn more >](#)





# Smart search for boxes using custom properties and AQL

But the full power of smart search comes with **Artifactory Query Language (AQL)**. AQL gives you unprecedented flexibility in how you search for artifacts. It offers a simple way to formulate complex queries that specify any number of search criteria, filters, sorting options and output fields. This allows you to dig into the far reaches of your repositories and extract a gold-mine of information that could otherwise never be found.

## Artifactory Query Language (AQL)

AQL is a flexible query language that gives you unprecedented flexibility to search for artifacts with any number of search criteria, filters, sorting options and output fields. As a RESTful API which streams output data, AQL is extremely fast and efficient with unbeatable response time and low memory consumption. AQL gives you unlimited degrees of freedom to formulate exactly the right query to find very specific artifacts you are searching for. Based on the database underlying Artifactory's unique architecture, AQL is something no other Binary Repository Manager can offer.

[Learn more >](#)

While Artifactory provides an extensive set of features to manage boxes, 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. For example, you could write a user plugin to scan every Vagrant box for viruses before it is uploaded to your repositories. 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.



Playing such a central role in the management of boxes, 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 Vagrant boxes at all times.

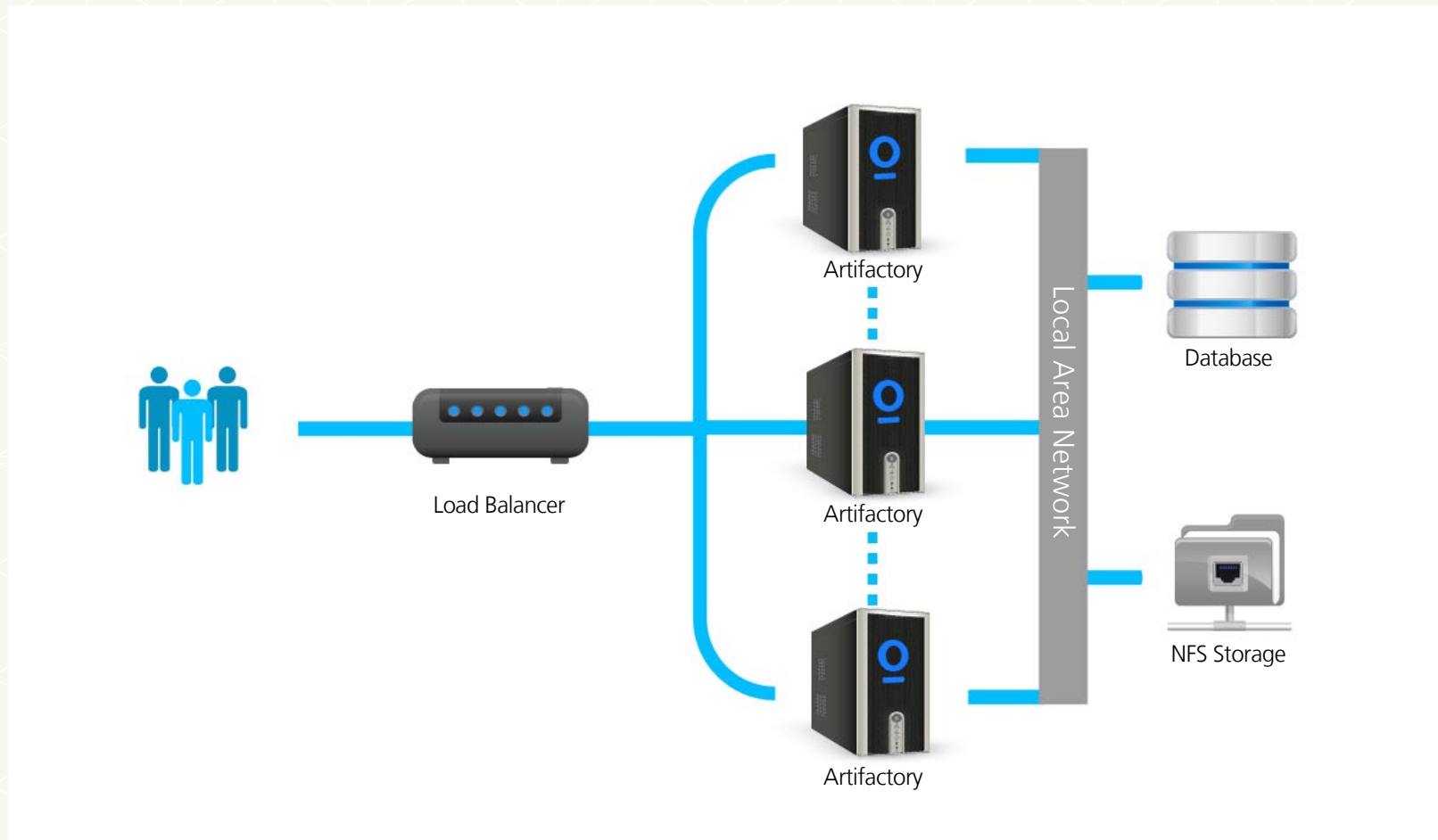
Artifactory supports a High Availability network configuration with a cluster of 2 or more Artifactory servers on the same Local Area Network. A redundant network architecture means that there is no single-point-of-failure, and your system can continue to operate as long as at least one of the Artifactory 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 >](#)





Ever been held up because someone suddenly realized that there are a bunch of licensing requirements to which you must comply?

Artifactory can help prevent such a scenario. Upon deployment of any binary to your repositories Artifactory performs a license check on your artifact and on all ensuing dependencies, and provides immediate feedback on all

license requirements. This lets you prepare ahead of time to ensure you comply early on in the development cycle avoiding unnecessary delays at “crunch time” when you want to release. And by integrating with BlackDuck Code Center, you can take advantage of a full range of license compliance and open source governance features while managing all of your binary uploads through Artifactory.



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

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 boxes. You can set restrictions on and monitor disk space usage, or define “watches” to receive an alert whenever there is a change to your most critical boxes. And with an extensive REST API, Artifactory can support virtually any rule-based cleanup protocol you would want to implement in your organization’s scripts.



No single packaging format or technology is sufficient to support development in a modern organization. There is a multitude of formats, a variety of build tools, different continuous integration systems and other technologies that go into building a flexible and maintainable software development ecosystem. Managing binaries for all the different packaging formats and integrating with all the moving parts of the ecosystem can become a maintenance nightmare.

Artifactory 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, Artifactory proxies Docker Hub (or any other public Docker registry), lets you store and manage your own images in local Docker repositories, and works transparently with the Docker client. If you are working with Vagrant, Artifactory lets you store your own packages in local Vagrant repositories, and works transparently with the Vagrant client. Similarly for NuGet, npm, Ruby, Debian, YUM, Bower, Python and more.

But development is only one end of the software delivery pipeline. Before a package makes it into a product, it needs to go through processes of build and integration. There are

many build and integration tools on the market, but there is only one product that works with them all. 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 a build, and also as a target to deploy build output. And to support cloud-based CI systems on which you are not able to apply plugins, Artifactory provides plugins for the build tools you use (such as Maven and Gradle) which ultimately provides the same level of build automation. That takes care of development and deployment, but what about distributing your software once it's ready for consumption. That's where Bintray comes in.

Bintray is JFrog's download center in the cloud offering rapid downloads, fine-grained access control, detailed stats and logs and an extensive REST API. Promoting releases for distribution from Artifactory is a matter of a single-click or API call. Like Artifactory, Bintray is package-agnostic and works seamlessly with all the different package clients, so it can be fully integrated into any continuous integration/continuous delivery ecosystem.

Artifactory is a universal repository. It is the single tool that sits in the center of your development ecosystem and "talks" to all the different technologies, increasing productivity, reducing maintenance efforts and promoting automated integration between the different parts. Together, Artifactory and Bintray are the central components of a fully-automated software distribution pipeline.



# Summary

This paper has shown how a Binary Repository Manager such as Artifactory can boost productivity of your organization's development and DevOps teams by managing and optimizing access to Vagrant boxes. Through local repositories with security and access control, Artifactory can replace Vagrant's Atlas and provide a facility where organizations can manage secure, private boxes. And since Artifactory is agnostic to the binary types that it manages, it can work with Linux packages, RubyGems, NuGet Gallery and virtually any binary package format to support the different platforms used by your organization, and provide central control over all binaries, both when uploading builds and when downloading artifacts.

For more information on how Artifactory can boost your organization's performance, please contact us at [info@jfrog.com](mailto:info@jfrog.com).

