PHP Composer
9 Benefits of Using a Binary Repository Manager

White Paper

March 2017 | www.jfrog.com

Executive Summary

PHP development has become one of the most popular platforms for client and server side web development. Each framework used for PHP development has its own set of advantages, but they all use PHP Composer to manage dependencies, alongside Packagist as the central repository.

PHP Composer may be able to find the right packages for you, but comes up short in case of network issues and cannot ensure that all developers in your organization are using the same version of a package. It's issues like these that Artifactory solves for you.

This white paper describes the benefits of using PHP Composer together with Artifactory, including:

Reliable Access
Overcome network issues restricting you from being able to download or update packages.

Optimized Build Process
Manage resource sharing within your organization to eliminate unnecessary network traffic.

Full Support for Docker
Support all Docker Registry APIs providing security features needed by enterprise Docker users.

Secure Solution
Enable controlled access through secure private PHP Composer repositories.

Smart Search and Artifactory Query Language (AQL)
Find the packages you need using advanced search tools and top-level search capabilities.

Distribution and Sharing
Enable efficient distribution of proprietary packages to give developers access to the same package version, resolve dependencies, and seamlessly share proprietary code regardless of physical location.

Artifactory High Availability
Give access to PHP Composer packages in a high availability configuration providing up to five-nines availability for PHP development.

Maintenance and Monitoring
Keep an organized managed system with automatic, timed cleanup processes, eliminating old and irrelevant artifacts.

Universal, End-to-End Solution
Developers use multiple protocols, build tools, packaging formats, and CI systems as part of software development, integration, and distribution processes. Administering the process and its **complexities is a major challenge for any organization**. Artifactory is a universal repository that supports any package manager and provides a line of plugins to handle the build and integration process.

Summary

Artifactory's support for PHP Composer streamlines PHP development, letting you concentrate on writing code without having to worry about security or availability of the Packagist public repository. Additional features such as smart search, AQL, High Availability, maintenance, monitoring and more help you work more efficiently and speed up development cycles, ultimately getting your product out to market as quickly as possible.

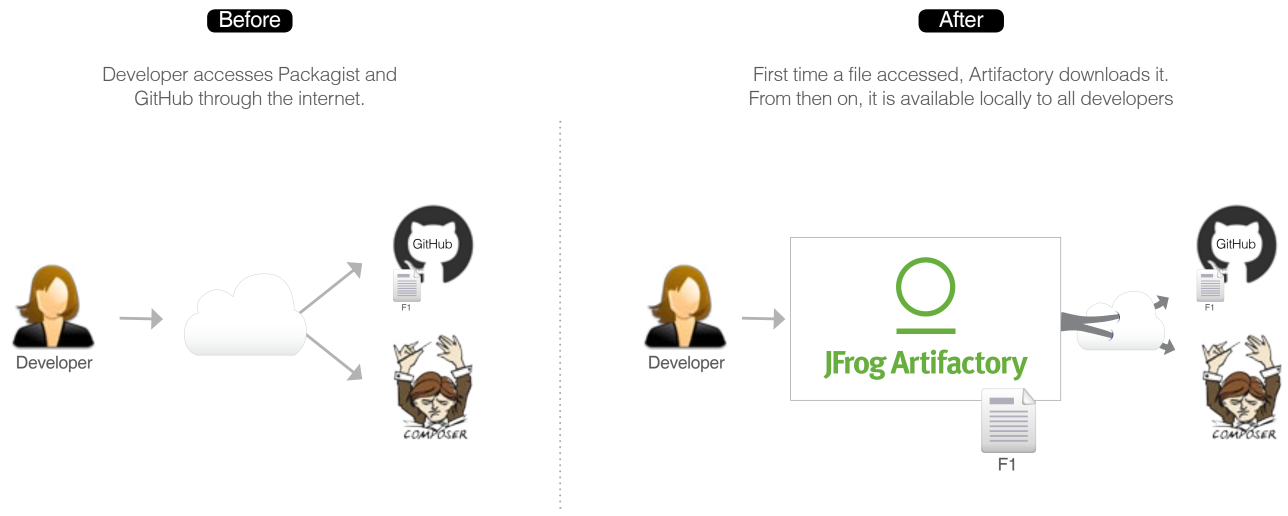
Introduction

PHP development has become one of the most popular platforms for client and server side web development. The huge popularity of PHP has sprouted many frameworks, such as Zend, Laravel, and Symfony, that make PHP development easier. While each framework may have its own set of advantages, they all use PHP Composer to manage dependencies, alongside Packagist as the central repository.
PHP Composer may be able to find the right packages for you, but what happens if you have a connectivity problem at 2 a.m. when you're trying to beat a deadline? How do you make sure that the ten people in your organization using the same package are all using the same version? It's issues like these that Artifactory solves for you.

Artifactory acts as a proxy for Packagist - developers only need to create a remote repository to emulate the relevant Git repository and then point their PHP commands at Artifactory.
Artifactory implements the Composer API and transparently replaces the Packagist Registry for you. When you request a component, Artifactory will delegate that request to Packagist, get the right pointer to the VCS repo where the package is located, and then download and zip the requested files into a single package along with its corresponding metadata file. For you, the developer, all this happens transparently in the background so you can continue to be productive while Artifactory does all the work of getting your packages and keeping them updated.

#1 Reliable Access

PHP Composer can potentially fail in cases such as either Packagist or GitHub go down and network issues, restricting you from being able to download or update packages. Artifactory overcomes these issues by caching your packages on-demand in a remote repository so you are independent of GitHub, Packagist and the network. If any of these go down, your packages are still available and you may continue to work oblivious of any external issues.


Before
Developer accesses Packagist and GitHub through the internet.
Developer
GitHub
F1
COMPOSER
After
First time a file accessed, Artifactory downloads it. From then on, it is available locally to all developers
Developer
JFrog Artifactory
F1
GitHub
F1
COMPOSER

Remote Repositories

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

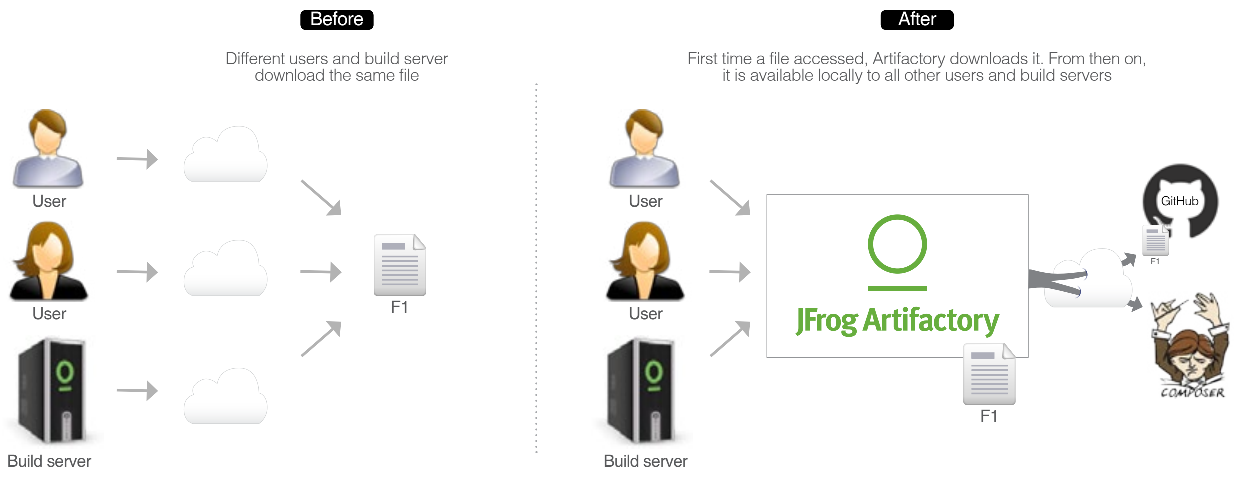
Learn more >

#2 Reduce network traffic and optimize builds

Developers use many of the same external package resources when assembling their code. In an organization, it is redundant for different developers to download the same packages. When using Artifactory, each package is only downloaded once and then cached for further reuse. From then on, it is provisioned directly from Artifactory making it unnecessary to download it again through the network - hence reducing network traffic. In the context of a build server, the need to download the same remote resources for builds over and over again makes the build process long and network intensive.

Integrating Artifactory optimizes this process by making downloaded packages available locally to be shared by all developers in the organization (thus reducing network traffic). Naturally, this is all transparent to the individual developer. Once packages are accessed through Artifactory, the developer can get on with what she does best and leave the rest to Artifactory.

Looking 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 packages from external resources. For the server to build these projects, all remote artifacts must be available to the server environment. Downloading all required artifacts may generate Gigabytes of data traffic on the network which takes a significant amount of time delaying the build process. By caching remote artifacts locally, the build process is much quicker and incurs much less networking.


Before
Different users and build server download the same file
User
User
Build server
F1
After
First time a file accessed, Artifactory downloads it. From then on, it is available locally to all other users and build servers
User
User
Build server
JFrog Artifactory
F1
GitHub
F1
COMPOSER

#3 Full Support for Docker

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 Composer packages, you also need to manage Docker images. But there's no need to onboard and maintain another tool. Artifactory is a fully-fledged Docker registry 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 create local Docker registries to **distribute and share images** within your organization and make managing images between different teams easy. You can even replicate your Artifactory Docker registries 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 registries** 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 Bower packages, you're already covered for Docker images.

#4 Security

Packagist, the main public repository for PHP Composer, does not inherently provide security or privacy. If you want to control access to your components you need to use private repositories, such as a private repo on GitHub. This results in many pointers registered in Composer, that will come up blank for most users since they point to someone else's private repository. Also, this approach makes your private GitHub repository pointers publicly available.

Artifactory offers a more complete security solution for PHP Composer. 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 repository. Then you can assign different sets of permissions to users and groups to control access to each repository. You can even use Artifactory's integration with LDAP, Active Directory, Crowd and others to control access to your servers. Effectively Artifactory becomes your own private and secure internal Packagist repository.

#5 Smart Search and Artifactory Query Language (AQL)

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

Artifactory provides you with flexible search capabilities to help you find the packages stored in your system. Artifactory supports the Composer API so your most basic search for packages remains the same as if you are searching through Packagist. It is also possible to find packages based on any combination of inherent attributes such as name, version, timestamp, checksum and more. In addition, Artifactory offers a set of common built-in searches. For example, you can easily find the "latest" version of any package without having to specify a version number. Artifactory also lets you assign any set of custom properties to your components, which can later be used in a search. For example, you can tag all the specific versions of components used in a product release with a "released" property to easily reproduce the released version later on.

Finally, Artifactory Query Language enables very specific searches. Using AQL, you can define search queries to any level of complexity needed to extract exactly the right packages you are looking for, with minimum effort and time.

With these capabilities, Artifactory lets you search for packages using virtually any set of rules relevant to your workflow.

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 >

Checksum-based search

Searching for a package by its checksum is a powerful feature supported by Artifactory thanks to a unique method of storing files by their checksum. Even if a binary has been renamed, moved or even deployed outside of your organization, you can trace it back to the original version and obtain its complete build information. Simply run the package through a checksum tool (both MD5 and SHA1 are supported) and run a "Checksum" search in Artifactory to retrieve the original version. Learn more >

#6 Distribute and share packages across your organization

As already mentioned, resources should efficiently be shared across your organization. This means that packages originating from remote resources only get downloaded once, and proprietary packages are securely available within your organization.

Using local repositories, Artifactory gives you a central location to store your internal packages. When all teams know that any package can be accessed from a single URL, access to local packages and managing dependencies between the different teams becomes very easy. But what if you want to share your packages with colleagues who are in geographically remote sites of your organization?

Artifactory supports replication of 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 packages can be made available to different teams wherever they may be located around the world.

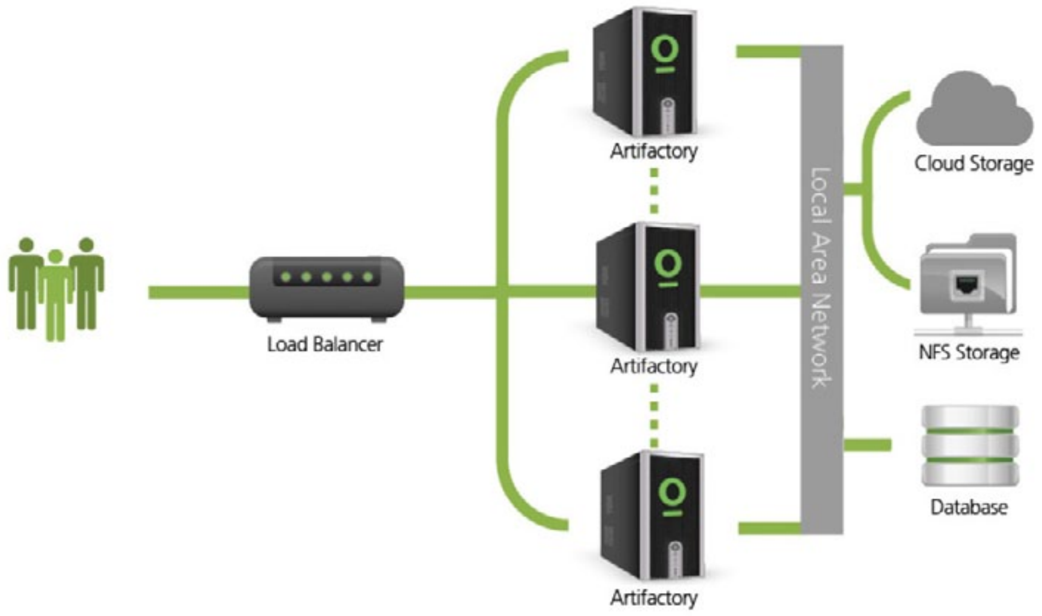
Local Repositories

Local repositories are physical, locally-managed repositories into which you can deploy artifacts. Typically these are used to deploy internal and external releases as well as development builds, but they can also be used to store packages that are not widely available on public repositories such as 3rd party commercial components. Using local repositories, all of your internal resources can be made available from a single access point across your organization from one common URL. Learn more >

#7 Artifactory High Availability

Playing such a central role in the management of binaries, your Binary Repository can become a mission-critical component of your organization meaning that any downtime can have severe consequences.

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. And 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 downtime or degradation of performance of the system as a whole.

Learn more >

#8 Maintenance and Monitoring

With current use of build servers and CI systems, the number of artifacts you generate can grow very quickly. Without proper management, your systems can quickly get clogged with old and irrelevant artifacts. 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 packages. 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 binaries. 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.

#9 A Complete Solution for Binaries

No single tool or technology is enough to support development in a modern organization. There is a multitude of packaging 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 PHP Composer, Artifactory proxies Packagist (or other metadata repositories) for the metadata, and GitHub (or other Git repos) for the binaries, and lets you store your own packages in local PHP repositories, working transparently with the Composer client. Similarly for Npm, Vagrant, NuGet, Ruby, Debian, YUM, Python, Chef, Puppet 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.

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.

Summary

By using Artifactory as a repository for PHP Composer packages, developers eliminate the hassles associated with network access, traffic, and build optimization. Furthermore, Artifactory supplies a wide range of other advantages such as support for Docker; enhanced distribution, maintenance, search, and security capabilities; and a High Availability option. Lastly, when used with Bintray, Artifactory provides an end-to-end binary solution.

Artifactory is a PHP Composer repository that optimizes development with PHP. It eliminates dependence on external resources such as Packagist, GitHub and even the external network, and optimizes the development and build processes. A range of security capabilities provide fine-grained access control while advanced facilities for search, maintenance and monitoring let developers focus on their primary task of developing code. As a universal repository, with full support for Docker and all other major package formats, Artifactory serves all the binary management needs of any development organization.

For more information on how Artifactory can boost your organization's performance, please contact us at info@jfrog.com