



Developing Fast with **⟨COCOAPODS⟩**

10 Reasons to Use an Artifact
Repository Manager

Contents

Executive Summary.....	3
Introduction.....	6
01 Reliable and consistent access to Podspecs and Pods.....	7
02 Reduce Network Traffic and Optimize Builds.....	8
03 Full Support for Docker.....	9
04 Full Integration With Your Build Ecosystem.....	10
05 Security and Access Control.....	11
06 Distribute and Share Pods Across Your Organization.....	12
07 Smart Search and Artifactory Query Language (AQL).....	13
08 System stability and reliability with Artifactory High Availability.....	14
09 Maintenance and Monitoring.....	15
10 A Universal, End-to-End Solution For All Binaries.....	16
Summary.....	17

Executive Summary

Until CocoaPods came on to the scene, using 3rd party components with Xcode projects involved a lot of manual and error-prone work keeping components and their dependencies up-to-date and coherent with each other. CocoaPods makes things easier by resolving and downloading dependencies based on a simple specification of the pods you need for your project. However, CocoaPods still faces challenges such as access to pods and specs if the network or repository are down, managing security, optimizing builds, supporting additional package formats and more. These challenges are met by JFrog Artifactory, the Universal Artifact Repository Manager that functions as a single access point organizing all of your binary resources including proprietary libraries, remote artifacts and other 3rd party resources, including pods and podspecs in particular. Fully supporting the CocoaPods client, Artifactory transparently replaces your CocoaPods repositories to boost your organization's productivity when developing with CocoaPods and open source libraries.

Reliable and consistent access to packages

The CocoaPods master repo and GitHub are invaluable resources for development with CocoaPods, but as external resources, they may be inaccessible if there is an outage, or if there are network issues.

Artifactory mitigates your dependence on external resources by caching remote artifacts in a "[remote repository](#)"; a local cache that serves as a proxy to the remote resource thus removing your dependence on the resource itself or on the network.

Reduced network traffic and optimized 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 a pod, it is locally available to all developers and build tools/CI servers resulting in greatly reduced network traffic and quicker build processes.

Full support for Docker

As the popularity of Docker continues to grow, you may find yourself on this bandwagon sooner than you think. So now you need to manage pods and podspecs as well as Docker images.

In addition to supporting CocoaPods, Artifactory is a fully-fledged Docker registry supporting all Docker Registry APIs. There is no need to onboard additional tools to support both pods and Docker images.

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 pods and podspecs.

Through a set of plugins, Artifactory is tightly integrated with the most build tools and CI systems in common use today, whether on-prem or in the cloud. These integrations enable Artifactory to produce exhaustive build information enabling fully reproducible builds.



Security and access control

Every organization needs to implement security policies so that people 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.

Distribute and share artifacts across your organization

To make the most of your development efforts, you need to share your internally developed pods with other developers and teams in your organizations.

Using [local repositories](#), Artifactory gives you a central location to store your internal binaries so that all teams can access any artifact from a single URL. To support distant teams, Artifactory offers both push and pull replication.

Smart search and Artifactory Query Language

Given the multitude of packages that can accumulate in any software development organization, finding something specific can sometimes get quite complex.

Artifactory offers a variety of options for search, from simple name search to common built-in search functions like “latest version search”. Artifactory also uniquely offers search by [checksum](#). [Artifactory Query Language \(AQL\)](#) takes search to new levels offering a simple way to formulate complex queries based on any number of parameters.

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 pods you generate can grow very quickly. Without proper management, your systems can quickly get clogged with old and irrelevant pods.

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

Universal end-to-end solution for all binaries

No single packaging format or technology is sufficient to support development in a modern organization. Managing binaries for all the different packaging formats and integrating with all the moving parts of the ecosystem can become a maintenance nightmare.

As a universal repository manager, Artifactory supports all major packaging formats including CocoaPods and integrates with all major build tools and CI servers. Artifactory also integrates with JFrog Bintray, the universal distribution platform. Together they form a universal, fully automated software distribution pipeline.



Summary

JFrog Artifactory is a universal artifact repository that supports software development for Apple platforms with CocoaPods as well as all other packaging formats in common use today. Integrating with all major build tools and CI servers, Artifactory serves as the central hub in any development ecosystem providing unmatched stability with a high availability configuration and offering reliable and consistent access to software artifacts. By managing and optimizing access to pods and podspecs, Artifactory boosts the productivity of your organization's development and DevOps teams. Artifactory is also tightly integrated with JFrog Bintray, the universal software distribution platform. Together, Artifactory and Bintray offer the most comprehensive end-to-end solution for software development and distribution using CocoaPods or any other development technology.

Introduction

The benefit of using third-party open source components in terms of cost and code quality are well known. However, without the right dependency manager, using 3rd party components involves a lot of manual and error-prone work to keep all those components up-to-date and coherent with each other. CocoaPods stepped into this domain for Xcode projects to make working with third-party libraries easy. Copying source files into Xcode projects, and then manually updating the project's dependencies was replaced by a simple specification of dependencies and pods for CocoaPods to manage. CocoaPods manages resolving and downloading dependencies between the different libraries you need, keeping versions up to date, searching for new libraries and generally making development with third-party components more manageable. However, there are issues that CocoaPods cannot overcome:

- What happens if the CocoaPods master repo or GitHub are not accessible, whether they are experiencing an outage or if there is simply a problem with the network?
- How do you optimize builds that need hundreds (or thousands) of dependencies?
- How do you prevent two (or two hundred) developers from downloading the same pod?
- How do you manage access control to the pods you create internally?
- How do you find a specific pod as their numbers multiply in your organization?
- How do you share internal pods with teams on your site and with those at geographically distant sites?
- CocoaPods may be able to manage your pods and podspecs, but what about managing your Docker images and other packages from additional development technologies that you may be using?

The answer to all of these questions is JFrog Artifactory, the Universal Repository Manager that functions as a single access point organizing all of your binary resources including proprietary libraries, remote artifacts and other 3rd party resources, including pods and podspecs in particular. Fully supporting the CocoaPods client, Artifactory transparently replaces your CocoaPods repositories to meet these challenges and boost your organization's productivity when developing with CocoaPods and open source libraries.

1>

Reliable and consistent access to Podspecs and Pods

To manage dependencies for your project, the CocoaPods client needs access to remote Specs repositories such as the public [CocoaPods Specs](#) repository on GitHub. Then, CocoaPods needs to access the dependency pods themselves which are also hosted on remote resources, usually on GitHub also. This means that to build a project you rely on external network access and on GitHub being up and available, both of which present a potential point-of-failure. Remote resources, like GitHub, can experience outages, and network access isn't always guaranteed, and in both of these scenarios, your work may come to a grinding halt.

JFrog Artifactory is an intermediary between developers and external resources. As a developer, all of your requests are directed to Artifactory, which gives you quick and consistent access to remote artifacts by caching them locally in a **remote repository**. Since remote artifacts are readily available from the cache on your local network, you are independent of external networking issues and are not affected if GitHub or any other remote resource goes down. Even in the extreme case that a remote resource ceases to exist altogether, any artifacts already downloaded to the local cache are still available to you. As a developer, you can continue your development efforts, and your builds won't be hampered by network issues or a repository going down.

Remote Repositories

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

[Learn more >](#)

Accessing Pods and Podspecs



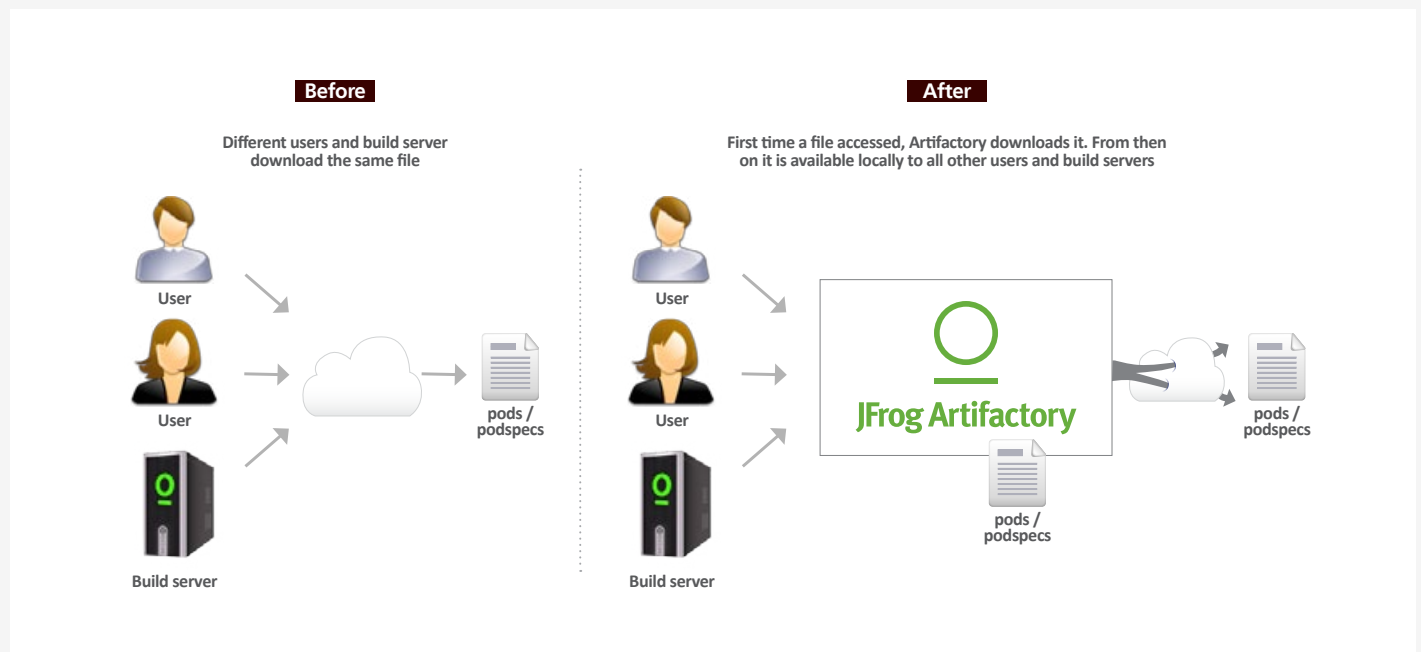
2>

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 pods downloaded from external resources is optimized. It makes no sense for two (or two hundred) developers using the same pod to download it separately.

In addition to reliability, another benefit of remote repositories is reduced networking. Once a package 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 pods 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 pods from external resources. For the server to build these projects, all remote artifacts must be available to the server environment. Downloading all those 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.



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 CocoaPods 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 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 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 when running in a **High Availability configuration**, Artifactory provides 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 CocoaPods packages, you're already covered for Docker images.

4>

Full Integration With Your Build Ecosystem

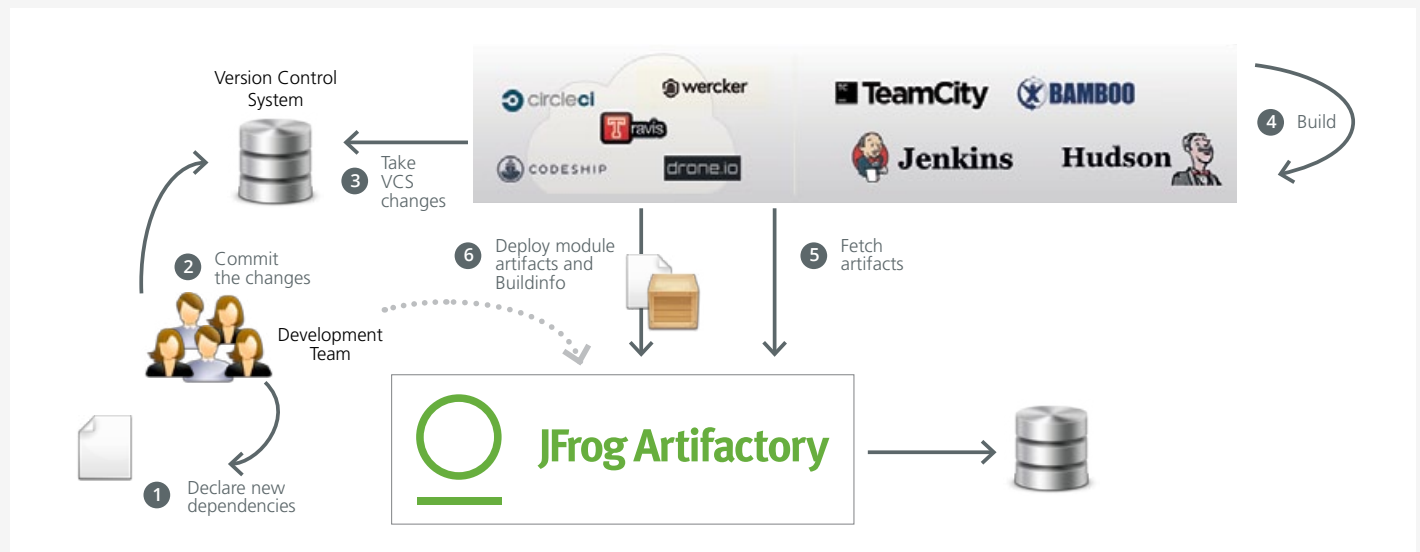
While it's important to make it easy and efficient for your developers to access pods, it's even more important for your build systems which may be running builds many times a day.

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 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 plugins? In that case, Artifactory provides plugins directly for the build tools themselves, which ultimately provides 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.



5>

Security and Access Control

If you want to control access to your pods you need to use private repositories on GitHub. But the result of working like this is that there are many pointers in the Podspecs Registry that come up blank for most users since they point to someone else's private repository. And then, do you really want to make pointers to your private GitHub repositories publicly available?

Artifactory offers a complete security solution for CocoaPods. 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, SAML, Crowd and others to control access to your servers. Effectively Artifactory becomes your own private and secure internal CocoaPods repository.





Distribute and Share Pods Across Your Organization

Remote repositories proxy the CocoaPods specs repository and other pod repositories on GitHub making. This makes sure that pods are only downloaded once, and are then shared across your organization. The same principle holds true for your own proprietary pods which you also want to be able to share securely with others in your organization.

Using **local repositories**, Artifactory gives you a central location to store your internal pods. When all teams know that any pod can be accessed from a single URL, access to local pods and managing dependencies between the different teams becomes very easy. But what if you want to share your pods 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 pods 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 >](#)



Smart Search and Artifactory Query Language (AQL)

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

Artifactory provides you with flexible search capabilities to help you find the pods stored in your system. First, Artifactory supports the CocoaPods spec, so your most basic search for packages is done in exactly the same way you are used to using the CocoaPods client. Then, you can find packages based on any combination of inherent attributes such as name, version, timestamp, **checksum** and a variety of other properties that Artifactory stores about pods. Artifactory also lets you assign any set of custom properties to your components, which can later be used for 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. 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 pods you are looking for.

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 >](#)

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 >](#)

8> System stability and reliability with Artifactory High Availability

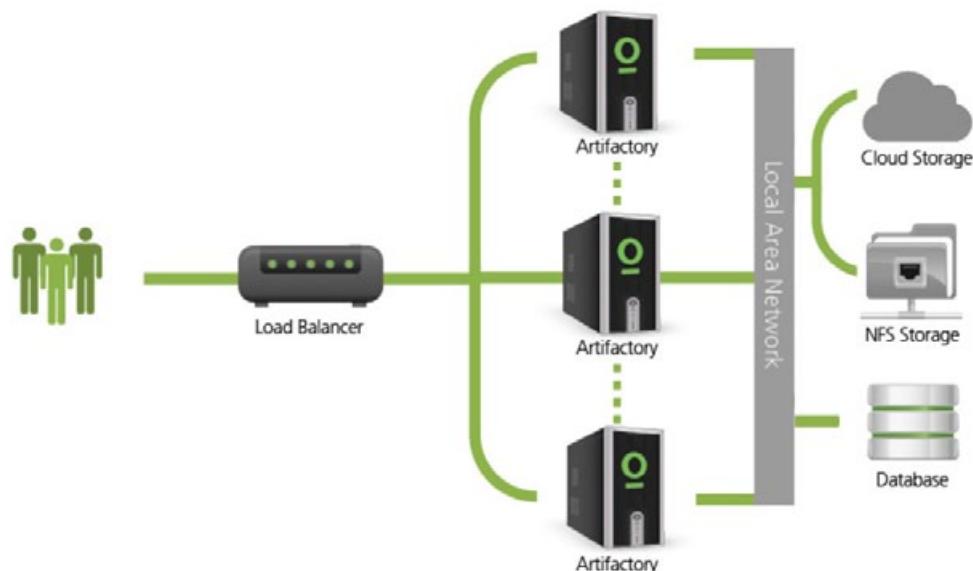
Playing such a central role in the management of pods and other packages, your repository manager can become a mission-critical component of your organization meaning that any downtime can have severe consequences.

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 >](#)



9 >

Maintenance and Monitoring

With current use of build servers and CI systems, the number of pods 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 pods. 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 pods. 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.



10> A Universal, End-to-End Solution For All Binaries

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 CocoaPods, Artifactory proxies the public remote Podspecs and pod repositories on GitHub, lets you store your own pods in local CocoaPods repositories, and works transparently with the CocoaPods client. If you are working with Docker, Artifactory proxies the Docker Hub (or any other remote public Docker resource), lets you store and manage your own images in local Docker repositories, and works transparently with the Docker client. Similarly for npm, Vagrant, NuGet, Ruby, Debian, YUM, 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 CDN 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

Artifactory's support for CocoaPods lets you concentrate on writing code for your iOS, watchOS, tvOS or OS X devices, without having to worry about security or availability of the CocoaPods Master Repository or GitHub. 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.



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