

Direct Cloud Storage Download

Overview

Artifactory's support for cloud storage provides all the benefits of massive scalability and allows your binary storage to grow, effectively, to any scale required and accommodate binaries of any size. From version 6.11, Artifactory can be further optimized for the download of large binaries, such as Docker images, from cloud storage, by delegating its function as a registry so it responds to download requests with a link through which the requesting client can download the binary directly from the corresponding cloud storage.

When a repository is configured to redirect downloads, a client requesting Artifactory for an artifact hosted in that repository receives an HTTP 302 response together with a Location header that contains a signed URL to the cloud storage location for direct download. The client can then use that signed URL to download the binary directly from cloud storage without it having to go through Artifactory first.

By downloading binaries directly from cloud storage:

- Artifactory is more efficient and reduces download latency since the binary object is obtained in a single phase, directly from the cloud, rather than having to go through Artifactory's cache first
- The requirements for local storage is reduced since Artifactory's local cache does not need to accommodate large binary objects from the cloud
- Artifactory registries operate more quickly since more compute power can be dedicated to calculating metadata rather than threads waiting for download requests to be completed

Requirements

Version and Platform

This feature requires an Enterprise+ or Edge license, and is available on JFrog Artifactory On-Premise when your Artifactory binary storage is configured with AWS S3.

This feature is supported from Artifactory version 6.11.

Supported Package Formats

JFrog Artifactory officially supports direct cloud storage downloads for Generic, Helm, Docker, Maven, RPM, Npm, Debian (supported from Client version 9), PyPI, Bower, Chef, CRAN, Composer, Conan, Gradle, Vagrant, Git LFS and RubyGems with any client that supports HTTP 302 redirect responses.



In order to work with curl, wget and Gradle clients please make sure to have a supported client version. Refer to this [knowledge base article](#) for more information.



Requires an Enterprise+ license

Page Contents

- Overview
 - Requirements
 - Version and Platform
 - Supported Package Formats
- Configuration
 - Configuring the Artifactory Filestore
 - Using the UI
 - Using the REST API
 - Using the YAML Configuration File
 - Fine-Tuning Direct Download
 - Getting Through Firewalls
 - Using CloudFront (Optional)

Configuration

To configure Artifactory for direct cloud storage download, follow the steps below:

1. [Configuring the Artifactory Filestore](#) to allow direct downloads
2. [Specify the repositories](#) that should redirect requests for direct download
3. [Fine-tune the configuration](#) with direct download parameters
4. [Configure your firewall](#) to allow direct downloads
5. Add the necessary configuration to [use CloudFront](#) (optional)

Configuring the Artifactory Filestore

As described under [Configuring the Filestore](#), you can control how Artifactory stores binaries using the `$ARTIFACTORY_HOME/etc/binarystore.xml` file.

To enable direct cloud storage download, follow the steps below:

1. In your `binarystore.xml` file, for s3 set `useSignature` to **true**, for s3-storage-v3 set `enableSignedUrlRedirect` to **true**.
2. Set the following parameters in the [S3 Binary Provider](#) or [AmazonS3 Official SDK Client](#) of your `binarystore.xml` file:

<code>enableSignedUrlRedirect</code>	Enables the feature for s3-storage-v3
<code>useSignature</code>	Enables the feature for s3

signedUrlExpirySeconds	Default: 30 (optional) Specifies the number of seconds that a signed URL provided to a requesting client for direct download from cloud storage is valid.
------------------------	--

The following snippet shows an example of how these parameters may look in the [S3 Binary Provider](#) of your *binarystore.xml* file:

```
<useSignature>true</useSignature>
<signedUrlExpirySeconds>30</signedUrlExpirySeconds>
<signatureExpirySeconds>300</signatureExpirySeconds>
```

Specifying Repositories for Direct Download

Artifactory can be configured to redirect requests for direct download at the level of repositories (there is no global configuration to automatically redirect all requests). This allows you to support both clients that support download redirect responses (HTTP 302) as well as those that do not. Download redirection is supported for all repository types - local, remote, virtual, and generic repositories. For local and remote repositories configured for redirection, all requests are redirected. For virtual repositories, it depends which aggregated local or remote repository ultimately provides the requested artifact. If that repository is configured for redirection, then the requesting client will receive the corresponding redirection response, otherwise, Artifactory will fetch the requested artifact and provide it to the client as usual without redirection.

You can configure Artifactory to redirect requests for direct download in the following ways:

- [Using the UI](#)
- [Using REST API](#)
- [Using the YAML configuration file](#)

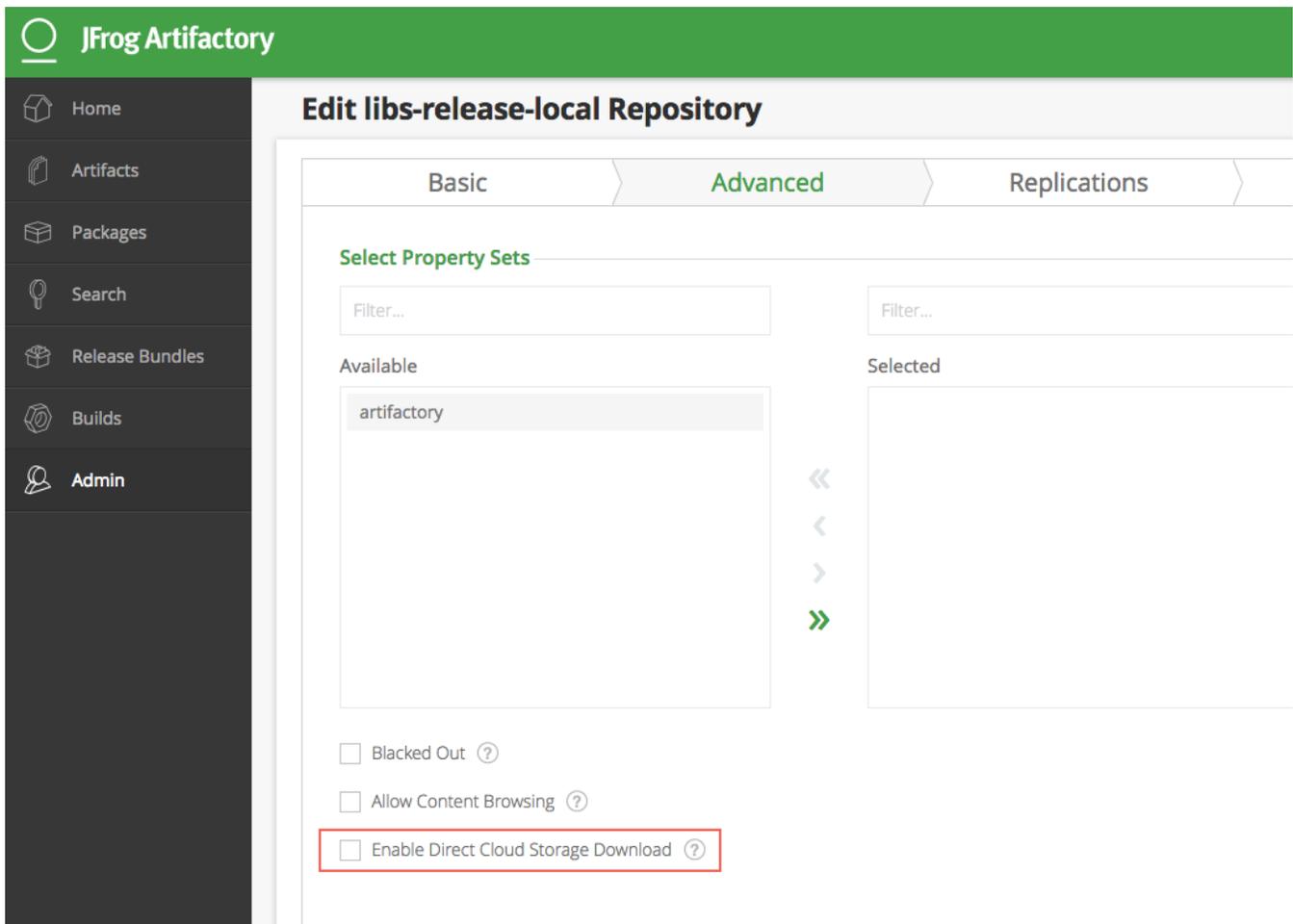


Remote repositories must be configured to store artifacts locally

Artifactory can only redirect requests for direct cloud storage download if the requested artifact is available locally. For local repositories, this is the natural state, however, for remote repositories, you have the option to configure the repository NOT to cache artifacts locally by deselecting the [Store Artifacts Locally](#) checkbox in the Advanced tab of the repository configuration (or by setting `storeArtifactsLocally=false` when [creating](#) or [updating](#) the repository using REST API). Therefore, for remote repositories, you need to ensure that the repository is configured to store artifacts locally (i.e., [Store Artifacts Locally](#) checkbox is checked, or `storeArtifactsLocally=true`). Failing to do so will generate an error.

Using the UI

To configure a local or remote repository to redirect requests for direct download, set the Enable Direct Cloud Storage Download checkbox in the [Advanced tab](#) of the repository configuration.



Using the REST API

To configure a local or remote repository to redirect requests for direct download using the REST API, set the `downloadRedirect` attribute to be `true` (default, `false`). For example:

```
curl -u admin:password -X POST -H "Content-Type: application/json" -d '{"rclass":"local", "downloadRedirect": true}' http://localhost:8080/artifactory/api/repositories/s3-generic
```

Alternatively, you can set the `downloadRedirect` parameter in the [repository configuration JSON](#) to be `true`.

Using the YAML Configuration File

If you are [configuring repositories using a YAML file](#), you can use the `downloadRedirect` attribute.

For example, to configure a repository called "generic-remote" for direct download:

```
curl -u admin:password -X PATCH -H "Content-Type: application/yaml" "http://localhost:8080/artifactory/api/system/configuration" -T remote.yaml
```

where "remote.yaml" is:

```
remoteRepositories:
  generic-remote:
    downloadRedirect:
      enabled: true
```

Fine-Tuning Direct Download

One of the benefits of the direct cloud storage downloads feature is that it reduces the load on Artifactory registries when multiple download requests for large artifacts must be served simultaneously. But this benefit is only really realized when the artifacts being downloaded are indeed large. To let you fine-tune Artifactory's behavior with this feature, in Artifactory's General Configuration, you can set the minimal artifact size for which a download request may be redirected for direct cloud storage download using the [Minimum Direct Cloud Storage Download Size](#) parameter (the default is 1 MB).

Getting Through Firewalls

When direct cloud storage downloads are enabled in Artifactory, it means that artifacts requested will actually be served from the S3 or CloudFront domains rather than from Artifactory's domain. Since these domains are constant in the signed URLs Artifactory provides, you should add them to your organization's firewall to enable smooth and unhindered downloads. The domains you should add are:

For S3: `https://<bucketName>.s3.amazonaws.com/...`

where <bucketName> is the value of the corresponding attribute specified in the S3 Binary Provider of your `binarystore.xml` file.

For example:

```
https://data1234.s3.amazonaws.com/myStoreOfBinaries/60/60b27f004e454aca92b0958209cck4829ec52430?response-content-disposition=attachment%3Bfilename%3D%22connector-java-5.1.4.txt%22&AWSAccessKeyId=AKIAPLS9HS&Expires=1539374539&Signature=1PDrAHLf
```

For CloudFront: `https://<cloudFrontDomainName>.cloudfront.net/...`

where <cloudFrontDomainName> is the value of the corresponding attribute specified in the S3 Binary Provider of your `binarystore.xml` file.

For example:

```
https://d2h935dfm2pkds.cloudfront.net/yuval/60/60b27f004e454aca92b0958209cck4829ec52430?response-content-disposition=attachment%3Bfilename%3Djava.jar&Policy=eyJTdGF0ZWludmZM2ZGZtNnB&Signature=NVMUyUTG~930ub3~mr6~&Key-Pair-Id=APPFISI9VDAF
```

Using CloudFront (Optional)

To further optimize direct cloud storage downloads, Artifactory lets you specify CloudFront settings which will determine in which locations your artifacts will physically reside which means downloads from your S3 bucket will be via a rapid CDN.

To use CloudFront with direct cloud storage downloads, follow the steps below:

1. Create a CloudFront distribution as described in [Getting Started with CloudFront](#) in the AWS documentation.
2. Generate a key pair as described in [Creating CloudFront Key Pairs for Your Trusted Signers](#) in the AWS documentation.
3. In the AWS console, under CloudFront | Behaviors, set a pattern on the distribution with `Forward Query Strings = yes`. This allows files to be downloaded using the actual filename as opposed to their SHA1 value.
4. Match up your distribution with your S3 bucket



i.e. make sure that the distribution points to the same S3 bucket configured in your [S3 Binary Provider](#) in your `$ARTIFACTORY_HOME/etc/binarystore.xml` file.

5. In your `binarystore.xml` file, for `s3` set `useSignature` to `true`, or for `s3-storage-v3` set `enableSignedUrlRedirect` to `true`.
6. Set the following CloudFront parameters in the [S3 Binary Provider](#) of your `binarystore.xml` file:

<code>cloudFrontDomainName</code>	The unique domain provided by AWS
<code>cloudFrontKeyId</code>	The unique identifier of the key pair you created in step 2 above
<code>cloudFrontPrivateKey</code>	The private key you created in step 2 above. This should have the following format: <pre>-----BEGIN RSA PRIVATE KEY----- -----END RSA PRIVATE KEY-----</pre>

The following snippet shows an example of how these parameters may look in the [S3 Binary Provider](#) of your `binarystore.xml` file:

```
<useSignature>true</useSignature>
<cloudFrontDomainName>d2n9783dfm6kdog.cloudfront.net</cloudFrontDomainName>
<cloudFrontKeyPairId>APKPDIVCZ4OS2GY6VMGS</cloudFrontKeyPairId>
<cloudFrontPrivateKey>-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAmVbCi33YzIyQMGJkNjN7NbRAEzZIxHv43nmq0b+tDM2CKX+f
.....
-----END RSA PRIVATE KEY-----
</cloudFrontPrivateKey>
```