

Alpine Linux Repositories

Overview

Artifactory provides full support for managing Alpine Linux packages through local, remote and virtual repositories. To learn more about Alpine, see [Alpine Linux Documentation](#).

Artifactory's support for Alpine Linux provides:

- The ability to provision Alpine Linux packages from Artifactory to the [Alpine Linux command line tool](#) from all repository types.
- Calculation of Metadata for Alpine Linux packages hosted in Artifactory's [Local Repositories](#).
- Access to remote Alpine Linux registries (such as <https://pkgs.alpinelinux.org>) through [Remote Repositories](#) which provide proxy and caching functionality.
- The ability to access multiple Alpine Linux registries from a single URL by aggregating them under [Virtual Repositories](#).
- Management of multiple RSA key pairs and enables you to set different RSA keys per Alpine repository, which allows you to sign Alpine Linux indexes in your Local and Virtual Repositories.



Alpine Linux version support

Artifactory supports Alpine Linux version 3.9.6 and above and *apk-tools* client version 2.10.3 and above.

Alpine Linux Repository Structure

An Alpine Linux repository is a directory with a collection of *apk* files, and an Alpine Linux repository consists of the following three main coordinates:

- branch
- repository
- architecture

Artifactory uses the same convention of the directory layout as the Alpine Linux repository uses. For example:

```
3.9 (branch)
main (repository)
  aarch64 (architecture)
    APKINDEX.tar.gz (index file)
    a2ps-4.14-r7.apk
    ...apk
  x86
    APKINDEX.tar.gz (index file)
    a2ps-4.14-r7.apk
    ...apk
  ...
...
```



Deployment Structure

All deployment of Alpine Linux packages into Artifactory must be under the `<BRANCH>/<REPOSITORY>/<ARCHITECTURE>/` structure.

If packages are not deployed under this structure, **they will not be included in any index file.**

Page Contents

- [Overview](#)
- [Setting up an Alpine Linux Repository](#)
 - [Prerequisite](#)
 - [Setting up a Local Repository](#)
 - [Setting up a Remote Repository](#)
 - [Setting up a Virtual Repository](#)
 - [Configuring Alpine Linux Package Manager to work with Artifactory](#)
- [Alpine Artifact Metadata](#)
- [REST API Support](#)

Setting up an Alpine Linux Repository

You can set up the following repository types:

- [Local Repository](#)
- [Remote Repository](#)
- [Virtual Repository](#)

Follow the steps according to each repository type below. An Alpine Linux package (apk) is deployed to a local Alpine Linux repository, and resolved using all repository types.

You can download packages from a local, remote, or virtual Alpine Linux repository.

Prerequisite

Alpine Linux requires RSA keys by default. To add RSA keys, please refer to [RSA Key Pairs](#).

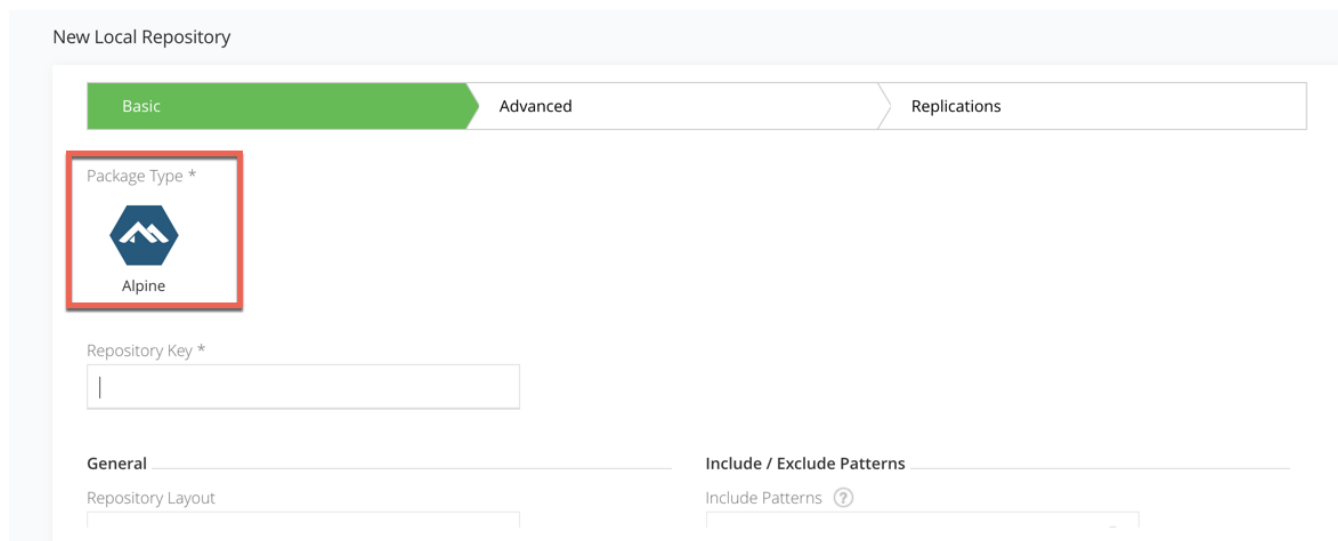
If you do not configure RSA keys, users have to use the `allow-untrusted` flag as described in [Resolving a Package](#).

Artifactory supports the signing of Alpine Linux index files, and not packages. To learn more about creating keys for Alpine Linux packages, click [here](#).

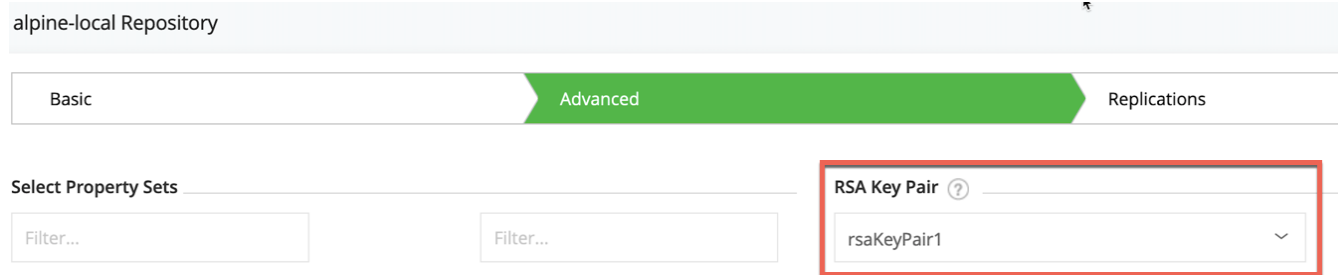
Setting up a Local Repository

Local repositories enable you to deploy Alpine Linux (.apk) packages. Artifactory calculates the metadata for all packages and indexes them to allow users to download these packages through the Alpine Linux client.

To create an Alpine Linux local repository, navigate to the **Administration** module, go to **Repositories | Repositories | Local | New Local Repository** and select **Alpine** as the **Package Type**.



In the **Advanced** tab, you can select an RSA key from the list to sign the Alpine Linux index file.



Setting up a Remote Repository

Remote Repositories enable you to proxy and cache Alpine Linux packages.

To specify that a Remote Repository supports Alpine Linux packages, you need to set its **Package Type** to **Alpine** when it is created.

New Remote Repository

Basic **Advanced** Replications

Package Type *

Alpine

Repository Key *

URL * ⓘ

General

Repository Layout: simple-default

Remote Layout Mapping: Select Remote Layout

Retrieving RSA keys in a Remote Repository

When setting up a Remote Repository, you will have to retrieve the RSA keys manually and set it in the apk client.

You can avoid retrieving the keys manually by aggregating a Remote Repository in a [Virtual Repository](#) that will enable you to use the Virtual Repository's key-pair to re-sign the aggregated index file.

In order to do so, configure your Virtual repository with a key-pair and then use [Set Me Up](#) to retrieve the Virtual public key to your local machine.

Setting up a Virtual Repository

A Virtual Repository defined in Artifactory aggregates packages from both local and remote repositories.

This allows you to access both locally hosted Alpine Linux packages and remote proxied Alpine Linux repositories from a single URL defined for the Virtual Repository.

To define a virtual Alpine Linux repository, do the following:


1. Create a [Virtual Repository](#), and set the **Package Type** to be **Alpine**.
2. Select the underlying local and remote Alpine Linux repositories to include in the **Basic** settings tab.
3. You can select an RSA key-pair from the list to sign the Alpine Linux packages.

RSA Key-pair

The RSA key-pair defined for the virtual repository will be used to sign the virtual index file. In case local repositories are defined with RSA key-pairs, these keys will be ignored.

New Virtual Repository +

Basic

Package Type *  Alpine

Repository Key *

General

Repository Layout: simple-default

Public Description:

Internal Description:

Repositories

Filter...

Filter...

Available Repositories

- alpine-local
- alpine-local-2
- alpine-remote
- alpine-remote-to-local
- alpine

Selected Repositories

New Virtual Repository +

Basic

New Pattern +

**/* x

Exclude Patterns ?

New Pattern +

Alpine Settings

Default Deployment Repository

RSA Key Pair ?

Configuring Alpine Linux Package Manager to work with Artifactory

In order to use Artifactory with your Alpine Linux client, first you need to set Artifactory as an Alpine Linux repository, and then add the relevant RSA public key to verify the index signature. Then you can proceed to resolve and deploy the relevant Alpine Linux package.

Step 1: Add Artifactory to your `/etc/apk/repositories` File

1. Navigate to **Application Module | Artifactory | Artifacts**.
2. Select the desired repository.
3. Select **Set Me Up**.
4. Copy the **General's** section command and run it.

General

For your Alpine Linux command line client to work with Artifactory, you first need to set Artifactory as an Alpine Linux repository in your `/etc/apk/repositories` file by using the following command:

```
1 sudo sh -c "echo 'http://<USERNAME>:<PASSWORD>@localhost:8081/artifactory/alpine-local/<BRANCH>
/<REPOSITORY>' >> /etc/apk/repositories"
```



Step 2: Verify the Index Signature

In order to verify index signature you need to add the repository RSA public key into your `/etc/apk/keys` folder.

Copy the **Set Me Up** public RSA key [unique retrieval command](#) and run it.

To verify index signature you need to add the repository public RSA key into your `/etc/apk/keys` folder by using the following command:

```
1 wget -O /etc/apk/keys/alpine-local.rsa.pub http://<USERNAME>:<PASSWORD>@localhost:8081/artifactory
/api/security/keypair/public/repositories/alpine-local
```

(note: the above image is an [example only](#) and you should not use it as-is. Use the **Set Me Up** dialog to get a unique command per your public key file name, as used to sign the index file)

If a local or virtual repository do not contain an RSA key-pair, you can either use the `--allow-untrusted` flag or request form your Admin to set a key-pair for the repository.

For signed indexes of remote repositories, please refer to [Setting up a Remote Repository](#).

Step 3: Resolve and Deploy Alpine Linux Packages

Resolving a Package

Resolve

After setting up your `/etc/apk/repositories`, you may use the following command to resolve packages from Artifactory. Your apk client will use the added Artifactory repositories to install the package.

```
1 apk add <PACKAGE>
```

In case your administrator didn't configure an RSA key per repository, you need to use the following flag:

```
1 apk add <PACKAGE> --allow-untrusted
```

Deploying a Package

To deploy an Alpine Linux package into an Artifactory repository you can use the following cURL with the relevant path parameters:

cURL

```
curl -H 'X-JFrog-Art-API:<API_KEY>' -XPUT "https://localhost:8080/artifactory/alpine-local/<BRANCH>
/<REPOSITORY>/<ARCHITECTURE>/<ALPINE_PACKAGE_NAME>" -T <PATH_TO_FILE>
```

Parameter	Example
branch	v3.9
repository	main
architecture	x86
alpine_package_name_	grep-3.1-r2.apk

Deploying a Package Using the UI

To deploy an Alpine Linux package to Artifactory, do the following:

1. Navigate to **Artifactory | Artifacts | Deploy**.

2. Select your Alpine Linux repository as the **Target Repository**.
3. In the **Target Path**, specify the relative path in the target repository:

```
<BRANCH>/<REPOSITORY>/<ARCHITECTURE>/<ALPINE_PACKAGE_NAME>
```

Deployment Structure

For your files to be indexed properly, it is very important to ensure that all deployment of Alpine Linux packages into Artifactory occurs under the `<BRANCH>/<REPOSITORY>/<ARCHITECTURE>/` structure. Packages deployed anywhere else will not be indexed.

Deploy

To deploy an Alpine package into an Artifactory repository you can use cURL with the relevant path parameters. The required parameters are package branch, repository and architecture in the following way:

```
1 curl -H 'X-JFrog-Art-API:<API_KEY>' -XPUT "http://localhost:8081/artifactory/alpine-local/<BRANCH>/<REPOSITORY>/<ARCHITECTURE>/<PACKAGE>" -T <PATH_TO_FILE>
```



Alpine Artifact Metadata

Artifactory writes several entries from the Alpine Linux package's metadata as properties on all of the artifacts.

These properties can be used to search for Alpine Linux packages more efficiently using the **Artifacts Search**, by **Package** type **Alpine**.

Alpine Linux package properties are the following:

- `alpine.name`
- `alpine.version`
- `alpine.branch`
- `alpine.repository`
- `alpine.architecture`

REST API Support

The Artifactory REST API enables the recalculation of the repository index, as described in [Calculate Alpine Repository Metadata](#).