Upgrading an Enterprise HA Cluster

Overview

This page describes the process to upgrade your Artifactory Enterprise HA cluster.

No down time

Since your cluster contains more than one node, you may complete the upgrade process without incurring any down time to the Artifactory service your organization is using.

Before You Begin

1. **Backup your system:** As a precaution, before you begin the upgrade process, we strongly recommend performing a complete **Complete System Backup**.
2. **Backup your database**
3. **Read through the process:** The backup procedure may vary slightly depending on your current version and your installation type (ZIP, RPM, Debian or Docker). To familiarize yourself with the specific backup process that you should use, we recommend reading through all the steps of the process before you begin

> ⚠ **Upgrading from 5.4.5 and below to 5.5 and above**
>
> In version 5.5, Artifactory's database underwent a schema change to accommodate SHA256 checksums. As a result, when upgrading from version 5.4.5 and below to version 5.5 and above, you need to follow one of the options described in the **detailed instructions** below.

Page Contents

The Upgrade Process

Upgrading Artifactory HA depends on which version you are starting from. Read the sections below carefully to make sure you complete the process correctly.

Upgrading From Versions Below 5.4.6

Artifactory 5.5 implements a database schema change to support SHA-256 checksums. **If your current version is 5.4.6,** you may proceed with the normal upgrade procedure described in Upgrading Steps below.

If your current version is below 5.4.6, to accommodate this change, you may select one of the following two upgrade options:

1. **Two-phase, zero downtime**
   In this option, you first need to upgrade your HA cluster to version 5.4.6. Once this upgrade is completed, you can then proceed to upgrade your HA cluster to version 5.5 and above. In both phases, you follow the normal upgrade procedure described in in Upgrading Steps below.
2. **One phase with downtime**
   In this upgrade option, your HA cluster will be down until you complete the upgrade for all nodes. This option requires you to execute the following preprocessing:

   - While the primary and all secondary nodes are up and running, add the following flag to `artifactory.system.properties` on the **primary node**:

     ```
     artifactory.upgrade.allowAnyUpgrade.forVersion=<to_version>
     For example:
     artifactory.upgrade.allowAnyUpgrade.forVersion=5.5.0
     ```

   - Wait until this property is synchronized to the database. You can verify that it has been synchronized by checking the *`artifactory.log`* file in each of the secondary nodes. Your nodes should display a message that looks like this:

```
[Node ID: some_node_id] detected remote modify for config 'artifactory.system.properties'
```

- Now you can proceed with the normal upgrade procedure described in in Upgrading Steps below.

> ⚠ **System will be down during this particular upgrade**
>
> Normally, upgrading an enterprise HA cluster does not incur downtime. As you upgrade each node, the other nodes continue to provide service.
>
> In this particular scenario of upgrading from version 5.4.5 and below to version 5.5 and above in a single phase, once you upgrade your primary node, your cluster will be in an incoherent state and will not be able to provide service until all nodes in the cluster have been upgraded. You can expect to see many errors in the log file until the upgrade is complete.
>
> As a result, we recommend completely taking down the whole cluster (i.e. taking down all cluster nodes), and then following the upgrade procedure, bringing up each node as it is upgraded.

If you try upgrading directly to version 5.5 and above **without** following one of these options, the upgrade will fail and the following message will be logged in the *artifactory.log* file:

```
To upgrade your HA installation to this version, you first need to upgrade to version 5.4.6 which
implements changes required to accommodate a database schema change.
```

Continue to recovering from an Attempt to Upgrade Directly to Version 5.5 and Above.

Upgrading Steps

Upgrading to the latest version is conducted in three phases:

1. Upgrading the primary node
2. Upgrading the secondary nodes
3. Verifying the HA installation and configuration

> ✓ **Want to stop using NFS?**
>
> If you want to stop using a shared NFS once the upgrade procedure is complete **(this is optional)**, please refer to **Migrating Data from NFS** to migrate to alternative storage.

> ⚠ **Using the UI without sticky sessions during an upgrade**
>
> If you do not configure your HA cluster to enable sticky sessions, the UI will not work until you have completed upgrading the whole cluster. To learn about configuring sticky sessions for an HA cluster, please refer to Session Management.

Upgrading the Primary Node

1. Remove the primary node from the load balancer, so all requests are directed to the secondary nodes. You may look at *$ARTIFACTORY_NODE_HOME/logs/request.log* and ARTIFACTORY_URL/api/tasks (search for "running") to ensure that Artifactory is completely inactive.
2. Perform a graceful shutdown of the primary node. While the primary node is down, the load balancer should redirect all queries to the secondary nodes.
3. Continue with the upgrade according to the instructions for your installation type.

> ⚠ **Upgrading from 3.x?**
>
> If your current version is 3.5 or higher, you first need to upgrade to the latest version 4.x using the procedure in this link and then upgrade to 5.x.
>
> Upgrading Artifactory HA from a version below 3.5 to version 5.x directly is not supported. To upgrade to version 5.x, you first need to upgrade your system to v3.5+, and then upgrade again to the final version 4.x, and then finally to 5.x.

ZIP Installation

a. Unzip the Artifactory distribution archive.
b. If you have not yet done so, perform a graceful shutdown of your currently running installation.
c. If the *$ARTIFACTORY_HOME/tomcat/conf/server.xml* has been modified keep it in a temporary location.
d. Backup files to a temporary location according to the conditions described below:
    i. In all cases, backup *$ARTIFACTORY_HOME/bin/artifactory.default*
    ii. If Artifactory is configured to work with a database that is not Derby, backup the *$ARTIFACTORY_HOME/tomcat/lib/<JDBC>* driver.
e. Remove the following files and folders from your *$ARTIFACTORY_HOME* folder:
    - *webapps/artifactory.war*
    - *webapps/access.war (this will only be present if your current version is 5.4.0 and above due to addition of the Access Service)*

- *tomcat*
- *bin*
- *misc*

f. Replace the removed files and folders with the corresponding ones from the new unzipped version.

g. Any files that were stored in temporary locations should now be returned to their original location under the new installation.

> **Running Artifactory as a ROOT application in Tomcat**
>
> If you have configured Artifactory to run as the ROOT application in Tomcat, before you proceed, you need to follow the steps described in **this Knowledge Base article**.

> **Version-specific Special Instructions**
>
> To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.
>
> **Upgrading to version 5.4.0 and above**
>
> From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase="webapps" startStopThreads="2"/>):
>
> ```
> ...
>         <Engine name="Catalina" defaultHost="localhost">
>             <Host name="localhost" appBase="webapps" startStopThreads="2"/>
>         </Engine>
> ...
> ```
>
> **Upgrading to version 5.6.1 and above**
>
> In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:
>
> ```
> <Connector port="8081" sendReasonPhrase="true"/>
> ```
>
> **Upgrading to version 5.7.0 and above**
>
> From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file
>
> ```
> <Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
> ```
>
> **Upgrading to version 6.3 and above**
>
> From version 6.3, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate special characters following this upgrade, the following new attributes, **relaxedPathChars='[]' relaxedQueryChars='[]'** were added for each Tomcat connector in the server.xml as shown in the example below:
>
> ```
>   <Service name="Catalina">
>         <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
> relaxedQueryChars='[]'/>
> ```

h. **If you installed Artifactory as a service, you now need to run the service**

   i. For a Linux service, browse to *`$ARTIFACTORY_HOME/bin`* and execute the following command **as root**: *`$ARTIFACTORY_HOME/bin/installService.sh [USER [GROUP]]`*

   ii. For Windows service, browse to %ARTIFACTORY_HOME%\bin. First uninstall the service by running `uninstallService.bat`, then reinstall the service by running `installService.bat`. You can now run the service through the Windows UI or as described in Running Artifactory.

Debian Installation

a. Log in as root (or use `sudo su -`).
b. If you have not yet done so, perform a graceful shutdown of your currently running installation
c. Execute the following command:

```
dpkg -i $jfrog-artifactory-<oss|pro>-6.y.z.deb
```

Managing Configuration Files

When upgrading a **Debian** installation the upgrade process overwrites the following set of configuration files:

- *system.properties*
- *config.xml*
- *default*
- *logback.xml*
- *mimetypes.xml*
- All files under *opt/jfrog/artifactory/misc*
- All files under *opt/jfrog/artifactory/webapps*

If any of these files were modified, a backed up file will be created automatically with a notification in the upgrade log. If you need to restore the configuration changes you can restore them from the backup created.

> ⚠ **Running Artifactory as ROOT**
>
> If you have configured Artifactory to run as the ROOT application in Tomcat, before you proceed, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase=" webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading to version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upg

```
  <Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
relaxedQueryChars='[]'/>
```

To ensure that all of the data remains in Artifactory following the upgrade to Pro (assuming you are using the default DB), please follow these steps:

a. Perform a graceful shutdown of your currently running installation.
b. Perform a backup of $ARTIFACTORY_HOME/data folder.

c. Delete the OSS installation.

```
apt-get remove jfrog-artifactory-oss)
```

d. Download and install the Pro package.

```
dpkg -i $jfrog-artifactory-pro-<version>.deb)
```

e. Import the ‘data’ backup folder to the new installation location.

```
$ARTIFACTORY_HOME/data
```

f. Start Artifactory.

```
$ARTIFACTORY_HOME/data
```

Docker Installation

In order to keep your data and configuration between versions, when upgrading the Artifactory Docker image, you need to use an external mounted volume as described under Managing Data Persistence.

To upgrade the Artifactory Docker image, follow these steps:

a. Stop the current container
b. Start a container with the new version using same data and configuration
c. Remove the old container

The example below shows this process for upgrading Artifactory from v5.0.0 to v5.1.0.

```
$ # Stop the currently running container
$ docker stop artifactory-5.0.0

$ # Start a new container from the new version image
$ docker run -d --name artifactory-5.1.0 --volumes-from=artifactory-5.0.0 -p 8081:8081 docker.bintray.io/jfrog/artifactory-pro:5.1.0

$ # Remove old container
$ docker rm artifactory-5.0.0
```

Once these commands have completed successfully, you would have the new version (5.1.0 in the above example) running with the data and configuration from the old version that was removed.

⚠ **Running Artifactory as ROOT**

If you have configured Artifactory to run as the ROOT application in Tomcat, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase=" webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading from version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upgraded. To accomodate special characters following this upgrade, the following new attributes, **relaxedPathChars='[]' relaxedQueryChars='[]'** were added for each Tomcat connector in the server.xml as shown in the example below:

```
 <Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
relaxedQueryChars='[]'/>
```

RPM Installation

Make sure you are upgrading from v3.6 or above

When running as an RPM installation, you can only upgrade to v5.x if your current version is 3.6 or above. If necessary, first upgrade your current version to 3.6, and then upgrade to v5.x .

If you try to upgrade a version below 3.6 using `rpm --force` you may end up deleting all of your data.

a. Download the **Artifactory Pro RPM Installer.** The latest version can be downloaded from the JFrog Artifactory Pro Download Page. Previous versions can be downloaded from JFrog Bintray.
b. Log in as root (or use `sudo su -`).
c. If you have not yet done so, perform a graceful shutdown of your currently running installation
d. Execute the following command:

```
rpm -U jfrog-artifactory-pro-6.y.z.rpm
```

Switching from Artifactory OSS to Pro

```
If you are just switching from Artifactory  with the same version number, you need to
append the command with --force --nodeps as follows:

rpm -U jfrog-artifactory-pro-6.y.z.rpm --force --nodeps
```

During an upgrade of an RPM installation different files may get backed up, where the backup file is appended with either a *.rpmorig* or a *.rpmnew* extension.

A *.rpmorig* extension means that the original file in your installation, the one that was there before performing the upgrade, was backed up before being replaced in the upgrade process.

A *.rpmnew* extension means that the original file in your installation, was **not** replaced in the upgrade, and instead, the new file with the same filename was backed up.

In either case, Artifactory will display a message such as:

```
warning: /etc/opt/jfrog/artifactory/default saved as /etc/opt/jfrog/artifactory/default.rpmorig
```

In these cases we recommend comparing the file installed once the upgrade has been completed with the backed-up file to see which best fits your needs, and using that one in the final setup.

If you make any changes, you may need to restart Artifactory for the change to be applied.

Upgrading Using YUM

An easy way to upgrade Artifactory from version 3.x or 4.x to the latest version is to use YUM with the Bintray Artifactory repository. The code snippets below show how to do this depending on whether your current version is below 3.6, or 3.6 and above.

Upgrading an Artifactory HA cluster?

If you are upgrading an Artifactory HA cluster, and you are running with a version that is older than version **5.4.6**, you should review the instructions on Upgrading an Enterprise HA Cluster prior to upgrading.

```
curl https://bintray.com/jfrog/artifactory-pro-rpms/rpm -o bintray-jfrog-artifactory-pro-rpms.
repo && sudo mv bintray-jfrog-artifactory-pro-rpms.repo /etc/yum.repos.d
yum install jfrog-artifactory-pro-6.y.z
```

To install the latest version of Artifactory, you can run:

```
yum install jfrog-artifactory-pro
```

```
curl https://bintray.com/jfrog/artifactory-pro-rpms/rpm -o bintray-jfrog-artifactory-pro-rpms.
repo && sudo mv bintray-jfrog-artifactory-pro-rpms.repo /etc/yum.repos.d
yum upgrade artifactory
yum install jfrog-artifactory-pro-6.y.z
```

To install the latest version of Artifactory, you can run:

```
yum install jfrog-artifactory-pro
```

Running Artifactory as ROOT

If you have configured Artifactory to run as the ROOT application in Tomcat, before you proceed, you need to follow the steps described in **this Knowledge Base article**.

> **Version-specific Special Instructions**
>
> To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.
>
> **Upgrading to version 5.4.0 and above**
>
> From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase=" webapps" startStopThreads="2"/>):
>
> ```
> ...
>         <Engine name="Catalina" defaultHost="localhost">
>             <Host name="localhost" appBase="webapps" startStopThreads="2"/>
>         </Engine>
> ...
> ```
>
> **Upgrading to version 5.6.1 and above**
>
> In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:
>
> ```
> <Connector port="8081" sendReasonPhrase="true"/>
> ```
>
> **Upgrading to version 5.7.0 and above**
>
> From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file
>
> ```
> <Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
> ```
>
> **Upgrading from version 6.3 and above**
>
> From version 6.3, the version of the Tomcat embedded in Artifactory was upgraded. To accomodate special characters following this upgrade, the following new attributes, **relaxedPathChars='[]' relaxedQueryChars='[]'** were added for each Tomcat connector in the server.xml as shown in the example below:
>
> ```
>  <Service name="Catalina">
>         <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
> relaxedQueryChars='[]'/>
> ```

4. Start up the primary node. When the primary node starts up, it will recognize that the HA cluster nodes are not all running the same version of Artifactory, and consequently, the system will be limited to allowing uploads and downloads.

   Any attempt to perform other actions such as changing the DB schema, modifying permissions, changing repository configuration and more, are strictly blocked. This limitation will continue until all the cluster nodes are once again running the same version.

   > **Version inconsistency generates exceptions**
   >
   > Running the HA cluster nodes with different versions generates exceptions. These can be seen in the log files and reflect the temporary inconsistent state during the upgrade process. This is normal and should be ignored until all the cluster nodes are, once again, running the same version.

5. Put the primary node back to the load balancer.

Upgrading the Secondary Node

For **each secondary node** in your HA cluster, perform the following steps:

1. Remove the node from the load balancer, so all requests are directed to the other nodes. You may look at *$ARTIFACTORY_NODE_HOME/logs/request.log* and ARTIFACTORY_URL/api/tasks (search for "running") to ensure that Artifactory is completely inactive.
2. Perform a graceful shutdown of the node. While the node is down, the load balancer should redirect all queries to the other nodes.
3. Continue with the upgrade according to the instructions for your installation type.

> ⚠ If your current version is 3.5 or higher, you first need to upgrade to the latest version 4.x using the following procedure and then upgrade to 5.x.
>
> Upgrading Artifactory HA from a version below 3.5 to version 5.x directly is not supported. To upgrade to version 5.x, you first need to upgrade your system to v3.5+, and then upgrade again to the final version 4.x, and then finally to 5.x.

ZIP Installation

a. Unzip the Artifactory distribution archive.
b. If you have not yet done so, perform a graceful shutdown of your currently running installation.
c. If the *$ARTIFACTORY_HOME/tomcat/conf/server.xml* has been modified keep it in a temporary location.
d. Backup files to a temporary location according to the conditions described below:
    i. In all cases, backup *`$ARTIFACTORY_HOME/bin/artifactory.default`*
    ii. If Artifactory is configured to work with a database that is not Derby, backup the *$ARTIFACTORY_HOME/tomcat/lib/<JDBC>* driver.
e. Remove the following files and folders from your *$ARTIFACTORY_HOME* folder:
    - *`webapps/artifactory.war`*
    - *`webapps/access.war (this will only be present if your current version is 5.4.0 and above due to addition of the Access Service)`*
    - *`tomcat`*
    - *`bin`*
    - *`misc`*
f. Replace the removed files and folders with the corresponding ones from the new unzipped version.
g. Any files that were stored in temporary locations should now be returned to their original location under the new installation.

> ⚠ **Running Artifactory as a ROOT application in Tomcat**
>
> If you have configured Artifactory to run as the ROOT application in Tomcat, before you proceed, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase="webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading to version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate special characters following this upgrade, the following new attributes, **relaxedPathChars='[]' relaxedQueryChars='[]'** were added for each Tomcat connector in the server.xml as shown in the example below:

```
  <Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]' 
relaxedQueryChars='[]'/>
```

h. **If you installed Artifactory as a service, you now need to run the service**
    i. For a Linux service, browse to *`$ARTIFACTORY_HOME/bin`* and execute the following command **as root**: *`$ARTIFACTORY_HOME/bin/installService.sh [USER [GROUP]]`*
    ii. For Windows service, browse to %ARTIFACTORY_HOME%\bin. First uninstall the service by running `uninstallService.bat`, then reinstall the service by running `installService.bat`. You can now run the service through the Windows UI or as described in Running Artifactory.

Debian Installation

a. Log in as root (or use `sudo su -`).
b. If you have not yet done so, perform a graceful shutdown of your currently running installation
c. Execute the following command:

```
dpkg -i $jfrog-artifactory-<oss|pro>-6.y.z.deb
```

Managing Configuration Files

When upgrading a **Debian** installation the upgrade process overwrites the following set of configuration files:

- *system.properties*
- *config.xml*
- *default*
- *logback.xml*
- *mimetypes.xml*
- All files under *opt/jfrog/artifactory/misc*
- All files under *opt/jfrog/artifactory/webapps*

If any of these files were modified, a backed up file will be created automatically with a notification in the upgrade log. If you need to restore the configuration changes you can restore them from the backup created.

Running Artifactory as ROOT

If you have configured Artifactory to run as the ROOT application in Tomcat, before you proceed, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase=" webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading to version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upg

```
  <Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
relaxedQueryChars='[]'/>
```

To ensure that all of the data remains in Artifactory following the upgrade to Pro (assuming you are using the default DB), please follow these steps:

a. Perform a graceful shutdown of your currently running installation.

b. Perform a backup of $ARTIFACTORY_HOME/data folder.

c. Delete the OSS installation.

```
apt-get remove jfrog-artifactory-oss)
```

d. Download and install the Pro package.

```
dpkg -i $jfrog-artifactory-pro-<version>.deb)
```

e. Import the ‘data’ backup folder to the new installation location.

```
$ARTIFACTORY_HOME/data
```

f. Start Artifactory.

```
$ARTIFACTORY_HOME/data
```

Docker Installation

In order to keep your data and configuration between versions, when upgrading the Artifactory Docker image, you need to use an external mounted volume as described under Managing Data Persistence.

To upgrade the Artifactory Docker image, follow these steps:

a. Stop the current container
b. Start a container with the new version using same data and configuration
c. Remove the old container

The example below shows this process for upgrading Artifactory from v5.0.0 to v5.1.0.

```
$ # Stop the currently running container
$ docker stop artifactory-5.0.0

$ # Start a new container from the new version image
$ docker run -d --name artifactory-5.1.0 --volumes-from=artifactory-5.0.0 -p 8081:8081 docker.
bintray.io/jfrog/artifactory-pro:5.1.0

$ # Remove old container
$ docker rm artifactory-5.0.0
```

Once these commands have completed successfully, you would have the new version (5.1.0 in the above example) running with the data and configuration from the old version that was removed.

Running Artifactory as ROOT

If you have configured Artifactory to run as the ROOT application in Tomcat, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase=" webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading from version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upgraded. To accomodate special characters following this upgrade, the following new attributes, **relaxedPathChars='[]' relaxedQueryChars='[]'** were added for each Tomcat connector in the server.xml as shown in the example below:

```
 <Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
relaxedQueryChars='[]'/>
```

RPM Installation

Make sure you are upgrading from v3.6 or above

When running as an RPM installation, you can only upgrade to v5.x if your current version is 3.6 or above. If necessary, first upgrade your current version to 3.6, and then upgrade to v5.x .

If you try to upgrade a version below 3.6 using `rpm --force` you may end up deleting all of your data.

a. Download the **Artifactory Pro RPM Installer.** The latest version can be downloaded from the JFrog Artifactory Pro Download Page. Previous versions can be downloaded from JFrog Bintray.
b. Log in as root (or use `sudo su -`).
c. If you have not yet done so, perform a graceful shutdown of your currently running installation
d. Execute the following command:

```
rpm -U jfrog-artifactory-pro-6.y.z.rpm
```

Switching from Artifactory OSS to Pro

```
If you are just switching from Artifactory  with the same version number, you need to
append the command with --force --nodeps as follows:

rpm -U jfrog-artifactory-pro-6.y.z.rpm --force --nodeps
```

During an upgrade of an RPM installation different files may get backed up, where the backup file is appended with either a *.rpmorig* or a *.rpmnew* extension.

A *.rpmorig* extension means that the original file in your installation, the one that was there before performing the upgrade, was backed up before being replaced in the upgrade process.

A *.rpmnew* extension means that the original file in your installation, was **not** replaced in the upgrade, and instead, the new file with the same filename was backed up.

In either case, Artifactory will display a message such as:

```
warning: /etc/opt/jfrog/artifactory/default saved as /etc/opt/jfrog/artifactory/default.rpmorig
```

In these cases we recommend comparing the file installed once the upgrade has been completed with the backed-up file to see which best fits your needs, and using that one in the final setup.

If you make any changes, you may need to restart Artifactory for the change to be applied.

Upgrading Using YUM

An easy way to upgrade Artifactory from version 3.x or 4.x to the latest version is to use YUM with the Bintray Artifactory repository. The code snippets below show how to do this depending on whether your current version is below 3.6, or 3.6 and above.

Upgrading an Artifactory HA cluster?

If you are upgrading an Artifactory HA cluster, and you are running with a version that is older than version **5.4.6**, you should review the instructions on Upgrading an Enterprise HA Cluster prior to upgrading.

```
curl https://bintray.com/jfrog/artifactory-pro-rpms/rpm -o bintray-jfrog-artifactory-pro-rpms.
repo && sudo mv bintray-jfrog-artifactory-pro-rpms.repo /etc/yum.repos.d
yum install jfrog-artifactory-pro-6.y.z
```

To install the latest version of Artifactory, you can run:

```
yum install jfrog-artifactory-pro
```

```
curl https://bintray.com/jfrog/artifactory-pro-rpms/rpm -o bintray-jfrog-artifactory-pro-rpms.
repo && sudo mv bintray-jfrog-artifactory-pro-rpms.repo /etc/yum.repos.d
yum upgrade artifactory
yum install jfrog-artifactory-pro-6.y.z
```

To install the latest version of Artifactory, you can run:

```
yum install jfrog-artifactory-pro
```

Running Artifactory as ROOT

If you have configured Artifactory to run as the ROOT application in Tomcat, before you proceed, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase=" webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading from version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upgraded. To accomodate special characters following this upgrade, the following new attributes, **relaxedPathChars='[]' relaxedQueryChars='[]'** were added for each Tomcat connector in the server.xml as shown in the example below:

```
 <Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
relaxedQueryChars='[]'/>
```

4. **Complete this step only when upgrading from 5.4.5 and below to 5.7 and above with 'One phase with downtime'**
   Copy the Master Key from the primary node located under the *$ARTIFACTORY_HOME/etc/security/master.key* directory, to the secondary node under the *$ARTIFACTORY_HOME/etc/security/master.key* directory.
5. Start up the secondary node.
6. Add the secondary node back to the load balancer.
7. Repeat this process for each secondary node.

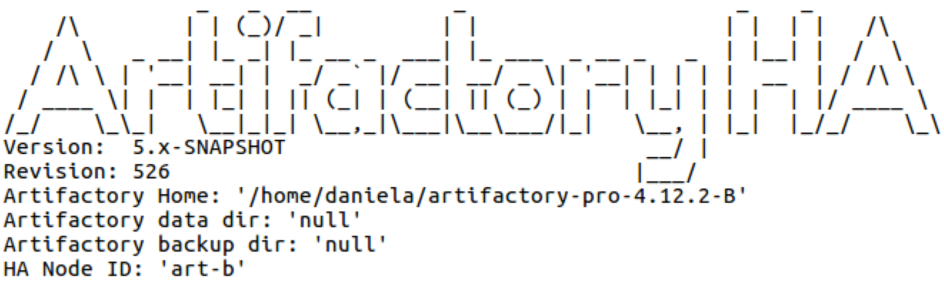
Check your installation

After starting up each secondary node, we recommend inspecting the *`ha-node.properties, db.properties`* `and` *`binarystore.xml files`* (under *$ARTIFACTORY_NODE_HOME/etc)* to ensure they are correctly configured

Verify the HA Installation and Configuration

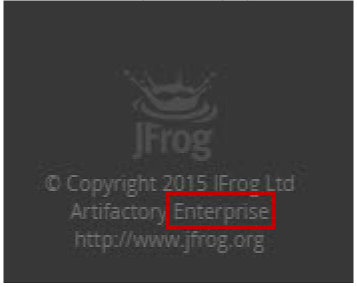
Once you have completed upgrading your HA cluster, you can verify that your cluster has been installed and configured correctly use the following series of tests:

1. Directly Access the Artifactory UI for the server you have just configured
2. In the **Admin** module go to **Advanced | System Logs** to view the log and verify that you see an entry for **HA Node ID.**

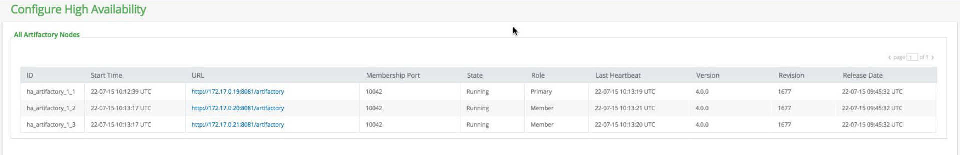
   ```
                 _   _  __            _                    _    _
        /\        | | (_)/ _|          | |                  | |  | |   /\
       /  \   _ __| |_ _| |_ __ _  ___| |_ ___  _ __ _   _ | |__| |  /  \
      / /\ \ | '__| __| |  _/ _` |/ __| __/ _ \| '__| | | ||  __  | / /\ \
     / ____ \| |  | |_| | || (_| | (__| || (_) | |  | |_| || |  | |/ ____ \
    /_/    \_\_|   \__|_|_| \__,_|\___|\__\___/|_|   \__, ||_|  |_/_/    \_\
   Version:  5.x-SNAPSHOT                             __/ |
   Revision: 526                                     |___/
   Artifactory Home: '/home/daniela/artifactory-pro-4.12.2-B'
   Artifactory data dir: 'null'
   Artifactory backup dir: 'null'
   HA Node ID: 'art-b'
   ```

3. The bottom of the module navigation bar should also indicate that you are running with an Enterprise license. In case of an error you will see an error message in the page header.

   © Copyright 2015 JFrog Ltd
   Artifactory Enterprise
   http://www.jfrog.org

4. Access Artifactory through your load balancer and log in as **Admin**.
5. In the **Admin** module go to **Configuration.** There should be a section called **High Availability.** When selected you should see a table with details on all the Artifactory nodes in your cluster as displayed below.

   Configure High Availability

   All Artifactory Nodes

   ‹ page 1 of 1 ›

ID	Start Time	URL	Membership Port	State	Role	Last Heartbeat	Version	Revision	Release Date
ha_artifactory_1_1	22-07-15 10:12:39 UTC	http://172.17.0.19:8081/artifactory	10042	Running	Primary	22-07-15 10:13:19 UTC	4.0.0	1677	22-07-15 09:45:32 UTC
ha_artifactory_1_2	22-07-15 10:13:17 UTC	http://172.17.0.20:8081/artifactory	10042	Running	Member	22-07-15 10:13:21 UTC	4.0.0	1677	22-07-15 09:45:32 UTC
ha_artifactory_1_3	22-07-15 10:13:17 UTC	http://172.17.0.21:8081/artifactory	10042	Running	Member	22-07-15 10:13:20 UTC	4.0.0	1677	22-07-15 09:45:32 UTC

6. In the **Admin** module under **Configuration | General,** verify that the **Custom URL Base** field is correctly configured to the URL of the Load Balancer.

✓ **Want to stop using NFS?**

If you want to stop using a shared NFS once the upgrade procedure is complete **(this is optional)**, please refer to **Migrating Data from NFS** to migrate to alternative storage.

Troubleshooting

Recovering from an Attempt to Upgrade Directly to Version 5.5 and Above

If you try to upgrade from version 5.4.5 and below to version 5.5 and above without following the procedure in one of the two options above, the upgrade will fail with the following message:

```
To upgrade your HA installation to this version, you first need to upgrade to version 5.4.6 which implements
changes required to accommodate a database schema change.
```

Depending on your installation type, the message may be displayed in the *artifactory.log* file, the *$ARTIFACTORY_HOME/tomcat/logs/localhost.log* file or in the command line output.

To recover from this error and, you first need to replace your current version as described below (according to your installation type).

Then you can proceed with upgrading to version 5.4.6 as required using the normal upgrade procedure, and then on to your final desired version, also, using the normal upgrade procedure.

Reinstalling Your Current Version

Reinstalling is similar to the process you went through when upgrading to your current version according to your installatian type.

For example, if you tried to upgrade from version 5.2 directly to version 5.5, you now need to reinstall version 5.2 and follow the instructions below, according to your installation type.

Zip Installation

Follow the **upgrade instructions for a Zip installation** using your current version.

Debian Installation

Follow the **upgrade instructions for a Debian installation** using your current version.

If your current version is below 5.4.0, make sure to remove the *$ARTIFACTORY_HOME/tomcat/webapps/access* folder. This folder is a remnant of the failed attempt to upgrade to version 5.5 and is not needed in versions previous to 5.4 where the Access Service is bundled together with the Artifactory WAR.

Docker Installation

Follow the **upgrade instructions for a Docker installation** using your current version.

RPM Installation

Follow the **upgrade instructions for an RPM installation** using your current version.

If your current version is below 5.4.0, make sure to remove the *$ARTIFACTORY_HOME/tomcat/webapps/access* folder. This folder is a remnant of the failed attempt to upgrade to version 5.5 and is not needed in versions previous to 5.4 where the Access Service is bundled together with the Artifactory WAR.