Setting Up Cold Artifact Storage

JFrog Subscription Levels

SELF-HOSTED

ENTERPRISE X
ENTERPRISE+

Overview

To set up a Cold Artifact Storage, you must first configure a designated Artifactory instance as the Cold Artifactory, connect to it a storage provider (either an inexpensive self-hosted storage or a cloud storage provider, such as Amazon S3 Glacier), and then connect the Cold Artifactory instance to an existing Artifactory instance, also called a Live Artifactory instance.

The steps required for this process include:

1. Set up a Live Artifactory instance
2. Create a Cold Artifactory instance
3. Add the Live and Cold Artifactory Instances to Mission Control
4. Bind the Live Artifactory and Cold Artifactory instances

Prerequisites

Before setting up Cold Artifact Storage, you will need to verify that the following prerequisites are in place:

- Two Artifactory instances

  ⚠ The two Artifactory instances - Live and Cold - do not have to be the same version.

- Enterprise or Enterprise Plus licenses
- Unique license for each Artifactory instance
- Administrator access to both instances
- The Artifactory instance that is set as the Live Artifactory instance must have the Mission Control microservice enabled (see Migrating Platform Deployments and License Buckets for details)

ⓘ Every cluster of Artifactory that needs to archive requires a single Cold Artifact Storage instance with a dedicated license.

Page Contents

Steps for Setting Up Cold Artifact Storage Using the Platform UI

Step 1: Set up a Live Artifactory Instance

The Live Artifactory instance can be a newly installed instance or an existing instance that must be configured as follows.

1. Open the Artifactory System YAML file of the instance.
2. Add the `shared.jfrogColdStorage.coldInstanceEnabled` property and set it as `false` to configure the instance as the Live instance.

Step 2: Create a Cold Artifactory Instance

On the Artifactory instance that you are designating as the Cold instance, perform the following steps:

1. **Update the Artifactory System YAML file**:
   Add the `shared.jfrogColdStorage.coldInstanceEnabled` property and set it as `true` to configure the instance as the Cold instance.

```
shared:
    jfrogColdStorage:
        coldInstanceEnabled: true
```

2. **Configure the Archive Binary Provider**:
   When setting up the Cold instance, you will need to connect it to a binary storage provider. You can use one of the following storage options:

   - **Self-hosted Storage**: If you wish to use the local file system, see Configuring the Filesystem Binary Provider.
   - **Amazon S3 Glacier Cloud Storage**: If you wish to use Amazon S3 Glacier, follow these steps:

     a. Open the `binarystore.xml` configuration file located in the `$JFROG_HOME/artifactory/var/etc/artifactory` folder.
     b. Specify the `s3-storage-v3-archive` chain.
     c. Within the chain, define the standard Amazon S3 Binary Provider template as outlined in the section Amazon S3 Official SDK Template and configure the parameters as per your requirements.

     Example

     s3-storage-v3-archive Template

```
<config version="1">
    <chain template="s3-storage-v3-archive"/> //Dedicated Cold Storage parameters
    <provider type="s3-storage-v3" id="s3-storage-v3"> //Based on s3-storage-v3 Template
        <bucketName>artifactory-ice-test-regular-bucket</bucketName>
        <path>artifactory-on-ice</path>
        <credential>creds</credential>
        <identity>ident</identity>
        <provider.id>aws-s3</provider.id>
        <region>eu-central-1</region>
    </provider>
</config>
```

3. **Restart the Cold Artifactory instance**:
   After making the above changes, restart the Cold Artifactory instance for the changes to take effect.

> **Disabling Cold Artifact Storage feature**
>
> The Cold Artifact Storage feature is enabled by default. If you wish to disable the feature, in the Artifactory System Properties file, set `artifactory.retention.enabled` as `false`.

> **Custom Base URL**
>
> It is recommended that you configure a Custom Base URL for this Artifactory instance. A custom URL base is especially useful when the system is running behind a proxy. For more information, see General System Settings.

Step 3: Add the Live and Cold Artifactory Instances to Mission Control

In this step you will need to add both instances to Mission Control. If Mission Control is enabled in the instance that you have designated the Live instance, then you will only need to do this step for the Cold instance.

1. First, create a binding token for Mission Control by following the steps in Pairing Tokens (available from Artifactory 7.29.7).
2. Next, add the Cold Artifactory instance to Mission Control by following the steps detailed in Registering a Platform Deployment.

> When you enter the JFrog Platform URL in step 6, this will generate the token you will need to pair the instance with Mission Control.

3. If the Live Artifactory instance you set up above is **not** the JPD that has the Mission Control enabled it it, add the Live Artifactory instance to Mission Control in the same way.

Remember to copy the token you generate to a location where you will be able to find it easily.

Step 4: Bind the Live and Cold Artifactory Instances

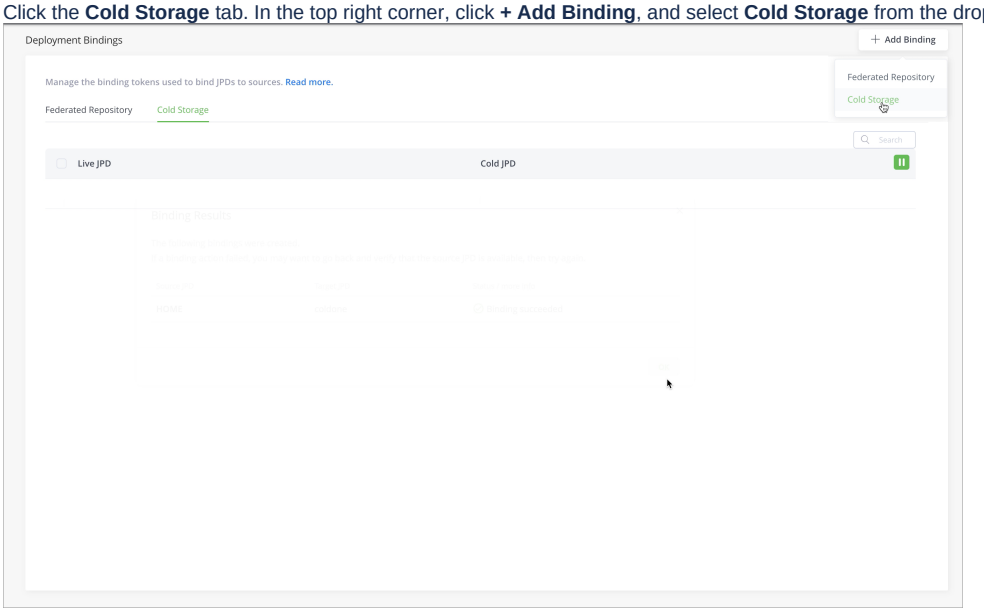
Available from Artifactory 7.38.4

After completing the previous two steps, your Platform Deployments will have two JPDs that are both connected to the Mission Control microservice: the Live instance and the Cold instance. The next step will be to bind the two instances to each other so that the Cold Artifactory instance will trust and receive requests from the Live Artifactory instance. You can only bind one Live instance to one Cold instances, and this is a **one-way trust**.

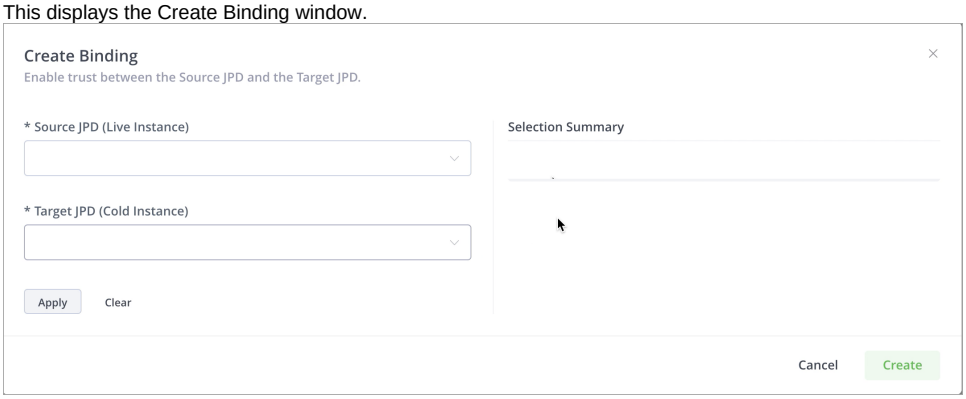
> **Important: Wait at least 5 minutes**
>
> After registering a new instance (Register JPD) and binding it using Cold Storage binding, it will take 5 minutes for the new instance to be identified in the JFrog Platform as a Cold instance (this is based on the polling configuration in Mission Control and the default is 5 minutes). As a result, it will take 5 minutes for the Cold instance to appear in the Target JPD so you should wait before proceeding with the instructions below.

Perform the following steps on the Artifactory instance that you have set as the Live instance.

1. From the **Administration** module, click **Platform Deployments | Bindings**.
2. Click the **Cold Storage** tab. In the top right corner, click **+ Add Binding**, and select **Cold Storage** from the dropdown list.

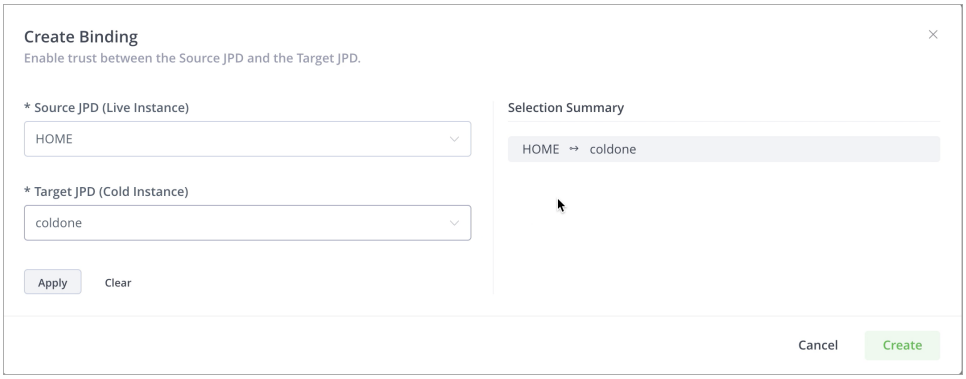


This displays the Create Binding window.

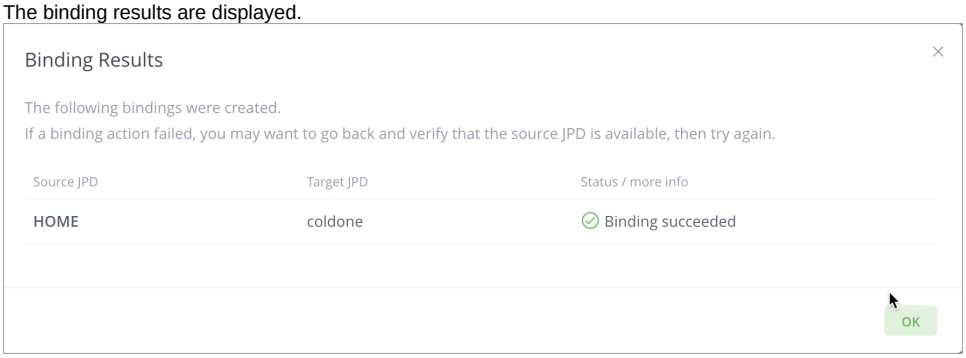


3. From the Source JPD dropdown, select the Live instance.
4. From the Target JPD dropdown, select the Cold instance.

5. Click **Apply**.
   The Summary displays the target and source you selected.



6. Click **Create** to create the new binding.
   The binding results are displayed.



Setting Up Cold Artifact Storage Using APIs

The preferred method of binding the two instances is using the flow described above. However, you can also bind them using the API steps described below.

1. First generate the Pairing Token to connect the Live and Cold instances.
2. On the Cold instance, execute the following POST request.

```
curl -u admin:password -X POST https://<cold_instance_ip_address>/artifactory/api/v1/service_trust
/pairing/artifactory-cold
```

   This generates the pairing token. For example:

```
{
  "pairing_token":"IiwiYWxnIjoiUlMyNTYiLCJraWQiOiJaWkhkQzBzenlqd2d..."
}
```

3. Next, on the Live instance, use the generated pairing token to execute the following PUT request. This sets up trust between the Live instance and the Cold instance:

```
curl -u admin:password -X PUT
-H "Content-Type: application/json"
-d '{"pairing_token" : "IiwiYWxnIjoiUlMyNTYiLCJraWQiOiJaWkhkQzBzenlqd2d..."}'
https://<live_instance_ip_address>/artifactory/api/v1/service_trust/pairing/artifactory-cold
```

   If the two instance are paired successfully, the request returns the following status:

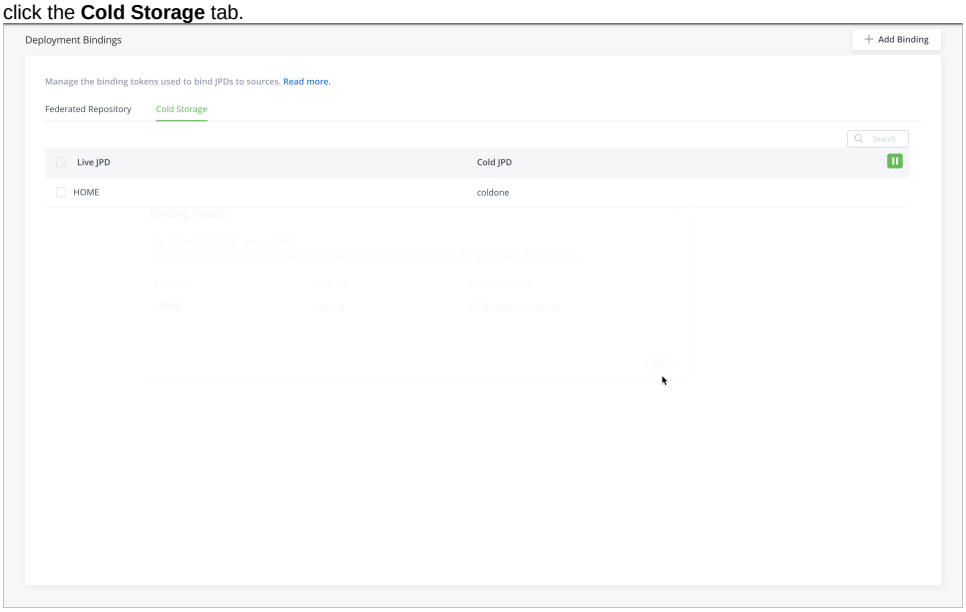
```
Trust with service artifactory-cold was established successfully
```

Managing Cold Storage Bindings

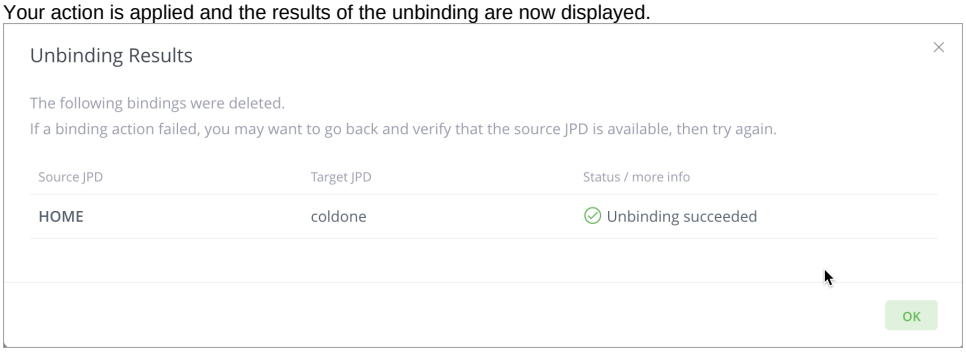
Revoking a Binding Using the Platform UI

> **Important Note on Revoking Bindings**
>
> You cannot remove the binding between the Live instance and the Cold instance in cases where the Cold instance is not available anymore.

1. To revoke the binding between a Live and Cold instance, from the Administration module, click **Platform Deployments | Bindings** and then click the **Cold Storage** tab.



2. Select the checkboxes for the bindings you wish to remove, and then click the **Delete** button to the right of the binding.
3. If you are deleting multiple bindings, select all checkboxes, and at the bottom of the window, select **Delete**.
   This displays a confirmation message that asks if you are sure you wish to remove the binding.
4. Click **OK** to confirm.
   Your action is applied and the results of the unbinding are now displayed.



Re-establishing Trust

Each connection between a Live instance and a Cold instance is bound to a namespace in the Cold instance, which is generated during the paring process. The namespace is unique and the trust is established for that specific namespace.

This namespace is required when you need to re-establish the trust between the Live and Cold instances, for example, after the token was revoked by mistake. In such scenarios:

1. Retrieve the namespace.
2. Regenerate the pairing token by providing the namespace as an argument in the POST request.

   Example: `curl -u admin:password -X POST https://<cold_instance_ip_address>/artifactory/api/v1/service_trust/pairing/artifactory-cold/gbbxmcuj`

   Where: `gbbxmcuj` is the namespace.

Optimizing Cold Artifact Storage Performance

Cold Artifact Storage includes several properties that can be used to configure and optimize its performance.

The list of parameters and their corresponding values that are shown below are for reference only. It is highly recommended that you do **not** change these values.

Live Artifactory Properties

The following properties can be configured in the Live Artifactory's `$JFROG_HOME/etc/artifactory/artifactory.system.properties` file.

Parameter	Description	Range /Default
Basic Properties		
artifactory.retention.warm.archive.concurrency.maxLevel	Number of threads that will be used per policy to archive items.	Default: 5
artifactory.retention.warm.restore.concurrency.maxLevel	Number of threads that will be used to restore items.	Default: 5
Advanced Properties		
artifactory.retention.cold.http.client.max.total.connections	Number of http connections that are allowed to deploy artifacts from the Live instance to the Cold instance.	Default: 50
artifactory.retention.cold.http.client.socketTimeout	Maximum socket timeout for an http connection.	Default: 15000
artifactory.retention.cold.http.client.max.connections.per.route	Number of http peers per route that are allowed to deploy artifacts from the Live instance to the Cold instance.	Default: 50
artifactory.retention.warm.restore.pull.items.intervalSecs	Setting for how often the Live instance will try to pull restored items from the Cold instance.	Default: 120 Recommended: 1800
artifactory.retention.warm.restore.batchSize	Setting for how many restored items will be pulled from the Cold instance to the Live instance.	Default: 1000
artifactory.retention.warm.restore.bandwidth.maxSizeMb	Approximate maximum allowed bandwidth for all restore operations.	Default: 0
artifactory.retention.warm.restore.error.threshold.percent	Error threshold to stop restore operation.	Default: 10
artifactory.retention.warm.archive.batchSize	Setting for how many items are fetched from the database during archive policy execution.	Default: 10000
artifactory.retention.warm.archive.bandwidth.maxSizeMb	Approximate max allowed bandwidth for a single archive policy execution.	Default: 0
artifactory.retention.warm.archive.error.threshold.percent	Stop execution if there are 10% errors during an archive process.	Default: 10

Cold Artifactory Properties

The following properties can be configured in the Cold Artifactory System YAML.

Parameter	Description	Range /Default
Advanced Properties		
artifactory.retention.cold.restore.render.tree.concurrency.level	Number of threads that will be used when processing the restore request.	Default: 5
artifactory.retention.cold.restore.status.mapper.intervalSecs	By default, on the Cold Artifactory instance, the cron job rate to check the restore status for each archived artifact is 120 seconds (2min). This can be reconfigured for cost optimization.	Default: 120

		Recommended: 10800
artifactory.retention.cold.restore.cleanup.intervalSecs	Interval for the cleanup job.	Default: 300

The following property can be configured in the Cold Artifactory's `$JFROG_HOME/etc/artifactory/artifactory.system.properties` file.

Parameter	Description	Range /Default
retention.warm.restore.artifact.limit	The maximum number of artifacts that can be restored during a single run is 30k. This value can be configured using the `retention.warm.restore.artifact.limit` property. During the restore process, if the number of artifacts crosses the max limit, the restore process terminates after completing the restoration of 30k artifacts and the cause for termination is logged.	Default: 30k