

Artifactory Administration

Regular Maintenance Operations

- Garbage Collection
- Storage Quota
- Cleanup Unused Cached Artifacts
- Cleanup Virtual Repositories
- Compress the Internal Database
- Prune Unreferenced Data(PUD)

Garbage Collection

- Garbage Collection disposes of the binaries from the filestore which are marked as "delete" candidates in the Artifactory. Along with the UI you also have an option to use this REST API for GC.
- When a new file is deployed, Artifactory checks if a binary with the same checksum already exists and if so, links the repository path to this binary. Upon deletion of a repository path, Artifactory does not delete the binary since it may be used by other paths. However, once all paths pointing to a binary are deleted, the file is actually no longer being used. To make sure your system does not become clogged with unused binaries, Artifactory periodically runs a "Garbage Collection" to identify unused ("deleted") binaries and dispose of them from the datastore. By default, this is set to run every 4 hours and is controlled by a cron expression.

Storage Quota

- Artifactory lets you set a limit on how much of your entire system disk space storage may be used to ensure that your server file system capacity is never used up. This helps to keep your system reliable and available. When using filesystem storage, the partition checked is the one containing the `$ARTIFACTORY_HOME/data/filestore` directory.

Cleanup Unused Cached Artifacts

- When configuring a remote repository, the Keep Unused Artifacts setting lets you specify how long a cached unused artifact from that repository should be kept before it is a candidate for cleanup. This setting does not immediately clean up the unused cached artifact, but merely marks it for clean up after the specified number of hours. The Cleanup Unused Cached Artifacts setting specifies when the cleanup operation should run, and only then unused, cached artifacts marked for cleanup are actually removed from the system.

Cleanup Virtual Repositories

- Virtual repositories use an internal cache to store aggregated metadata such as POM files. The Cleanup Virtual Repositories operation deletes cached POM files that are older than 168 hours (one week)

Compress the Internal Database

- This feature is only relevant when using the internal Derby database. A Derby database may typically contain unused allocated space when a large amount of data is deleted from a table or its indices are updated. By default, Derby does not return unused space to the operating system. For example, once a page has been allocated to a table or index, it is not automatically returned to the operating system until the table or index is destroyed.

Prune Unreferenced Data(PUD)

- Unreferenced binary files may occur due to running with wrong file system permissions on storage folders, or running out of storage space. When you invoke this action, Artifactory removes unreferenced binary files and empty folders present in the filestore or cache folders.
- Prune Unreferenced Data(PUD) deletes the randomly existing binaries in the filestore which are not referenced in the Artifactory (in case for some reason the filestore had unreferenced binaries in the filestore). When you invoke this action, Artifactory removes unreferenced binary files and empty folders present in the filestore or cache folders. PUD does not delete the binaries marked as delete candidates in Artifactory(those would be deleted by GC). PUD doesn't have a REST API associated with it; considering this would be not very common scenario you can use the UI click option for PUD as a one-off task(Admin->Advanced->Maintenance->Storage->Prune Unreferenced Data).

Security Configuration

- LDAP
- Crowd / JIRA
- SAML SSO
- OAuth SSO
- HTTP SSO
- Access Tokens
- Users
- Groups
- Permissions
- Certificates
- Signing Keys Management
- SSH Server Configuration

LDAP

- Artifactory supports authenticating users against an LDAP server out-of-the-box. When LDAP authentication is active, Artifactory first attempts to authenticate the user against the LDAP server. If LDAP authentication fails, Artifactory tries to authenticate via its internal database. For every LDAP authenticated user Artifactory creates a new user in the internal database (provided that the user does not already exist), and automatically assigns that user to the default groups.
- <https://www.jfrog.com/confluence/display/RTF/Managing+Security+with+LDAP>

Crowd / JIRA

- The Atlassian Crowd Integration allows you to delegate authentication requests to Atlassian Crowd, use authenticated Crowd users and have Artifactory participate in a transparent SSO environment managed by Crowd. In addition, Atlassian Crowd Integration allows the use of JIRA User Server as an authentication server, but without support of SSO.
- <https://www.jfrog.com/confluence/display/RTF/Atlassian+Crowd+and+JIRA+Integration>

SAML SSO

- SAML is an XML standard that allows you to exchange user authentication and authorization information between web domains. JFrog's Artifactory offers a SAML-based Single Sign-On service allowing federated Artifactory partners (identity providers) full control over the authorization process. Using SAML, Artifactory acts as service provider which receives users authentication information from external identity providers. In such case Artifactory is no longer responsible to authenticate the user although it still has to redirect the login request to the identity provider and verify the integrity of the identity provider's response.
- <https://www.jfrog.com/confluence/display/RTF/SAML+SSO+Integrati+on>

OAuth SSO

- OAuth integration allows you to delegate authentication requests to external providers and let users login to Artifactory using their accounts with those providers. Currently, Google, OpenID Connect, GitHub Enterprise and Cloud Foundry UAA are supported.
- <https://www.jfrog.com/confluence/display/RTF/OAuth+Integration>

HTTP SSO

- The Single Sign-on (SSO) Add-on allows you to reuse existing HTTP-based SSO infrastructures with Artifactory, such as the SSO modules offered by Apache HTTPd. Artifactory's authentication will work with commonly available SSO solutions, such as native NTLM, Kerberos, etc... SSO works by letting Artifactory know what trusted information it should look for in the HTTP request, assuming that this request has already been authenticated by the SSO infrastructure, which sits in front of Artifactory.
- <https://www.jfrog.com/confluence/display/RTF/Single+Sign-on>

Access Tokens

- Artifactory offers the option for authentication through access tokens. An access token may be assigned to a user, or to an entity that is not an Artifactory user such as a job in a CI server. Permissions are assigned to access tokens by including them in Groups. Access tokens offer advantages such as cross-site authentication, limited-time access, authenticated access for non-users and more.
- <https://www.jfrog.com/confluence/display/ACC/Access+Tokens>

Certificates

- Some remote repositories (e.g. Red Hat Networks) block access from clients that are not authenticated with an SSL/TLS certificate. Therefore, to use a remote repository to proxy such resources, Artifactory must be equipped with the corresponding SSL/TLS certificate.

Adding Certificates

- Certificates are managed in the Admin module under Security | Certificates.
- A certificate entered into this module should be a PEM file that includes both a private key and its corresponding certificate.

Using a Certificate with a Remote Repository

- When a remote repository proxy's a resource that requires authentication with a certificate, you need to obtain the certificate from the resource's owner and add it to the list of certificates as described above.
- Under the remote repository's Other Settings, select the certificate you want to use from the list provided in the SSL/TLS Certificate field. Other Settings, select the certificate you want to use from the list provided in the SSL/TLS Certificate field.

Signing Keys Management: Manage GPG Signing Keys

- Artifactory lets you manage a pair of GPG signing keys so you can sign packages for authentication in several formats such as Debian, Opkg and YUM. You can manage your GPG signing keys in the Admin module under Security | Signing Keys.
- <https://www.jfrog.com/confluence/display/RTF/GPG+Signing>

Signing Keys Management: Manage Artifactory Signing Keys

SSH Server Configuration

- From version 4.4, Artifactory supports SSH authentication for Git LFS and the JFrog CLI using RSA public and private keys. This allows these tools to exchange sensitive information with the Artifactory server that is authenticated via SSH.tory.
- Artifactory supports SSH authentication for Git LFS and the JFrog CLI using RSA public and private keys. SSH has the benefit of two-way authentication. In other words, before any sensitive data is exchanged between Artifactory and the client, the Artifactory server is authenticated to the client, and then the user operating Git LFS or JFrog CLI client is authenticated to Artifactory.
- <https://www.jfrog.com/confluence/display/RTF/SSH+Integration>

