Configuring a Reverse Proxy

Overview

In many cases, an organization may provide access to Artifactory through a reverse proxy such as NGINX or Apache. In some cases, for example with Docker, this set up is even mandatory. To simplify configuring a reverse proxy, from version 4.3.1, Artifactory provides a **Reverse Proxy Configuration Generator** screen in which you can fill in a set of fields to generate the required configuration snippet which you can then download and install directly in the corresponding directory of your reverse proxy server. You can also use the REST API to manage reverse proxy configuration.

If you are using Artifactory behind a reverse proxy, we recommend that you set your Custom URL Base to match your Artifactory Server Name.

Page Contents

- Overview
- Reverse Proxy Settings
- Docker Reverse Proxy Settings
    - Using a Reverse Proxy
        - Using Subdomain
        - Using Port Bindings
    - Using Direct Access
- REST API

Reverse Proxy Settings

To configure a reverse proxy, in the **Admin** module, select **Configuration** | **HTTP Settings** and execute the following steps in the **Reverse Proxy Settings** panel:

- Fill in the fields according to your configuration.
- Generate the configuration file. You may click the icons in the top right of the screen to view your configuration (which you may copy) or download it as a text file.
- Place the configuration file in the right place under your reverse proxy server installation and reload the configuration.

Using NGINX? Note these requirements.

To use NGINX as a reverse proxy to work with Docker, you need NGINX v1.3.9 or higher.

The NGINX configuration file should be placed under the *sites-enabled* directory.

For more details, please refer to Configuring NGINX.

Using Apache? Note these requirements.

Some features in the Apache configuration are only supported from Apache HTTP Server v2.4.

To use Apache as your reverse proxy server, make sure you have the following modules installed and activated:

- proxy_http
- proxy_ajp
- rewrite
- deflate
- headers
- proxy_balancer
- proxy_connect
- proxy_html
- ssl
- lbmethod_byrequests
- slotmem_shm
- proxy

Support to generate Apache reverse proxy configuration is available from Artifactory version 4.4.1.

For more details, please refer to Configuring Apache.

Best practice

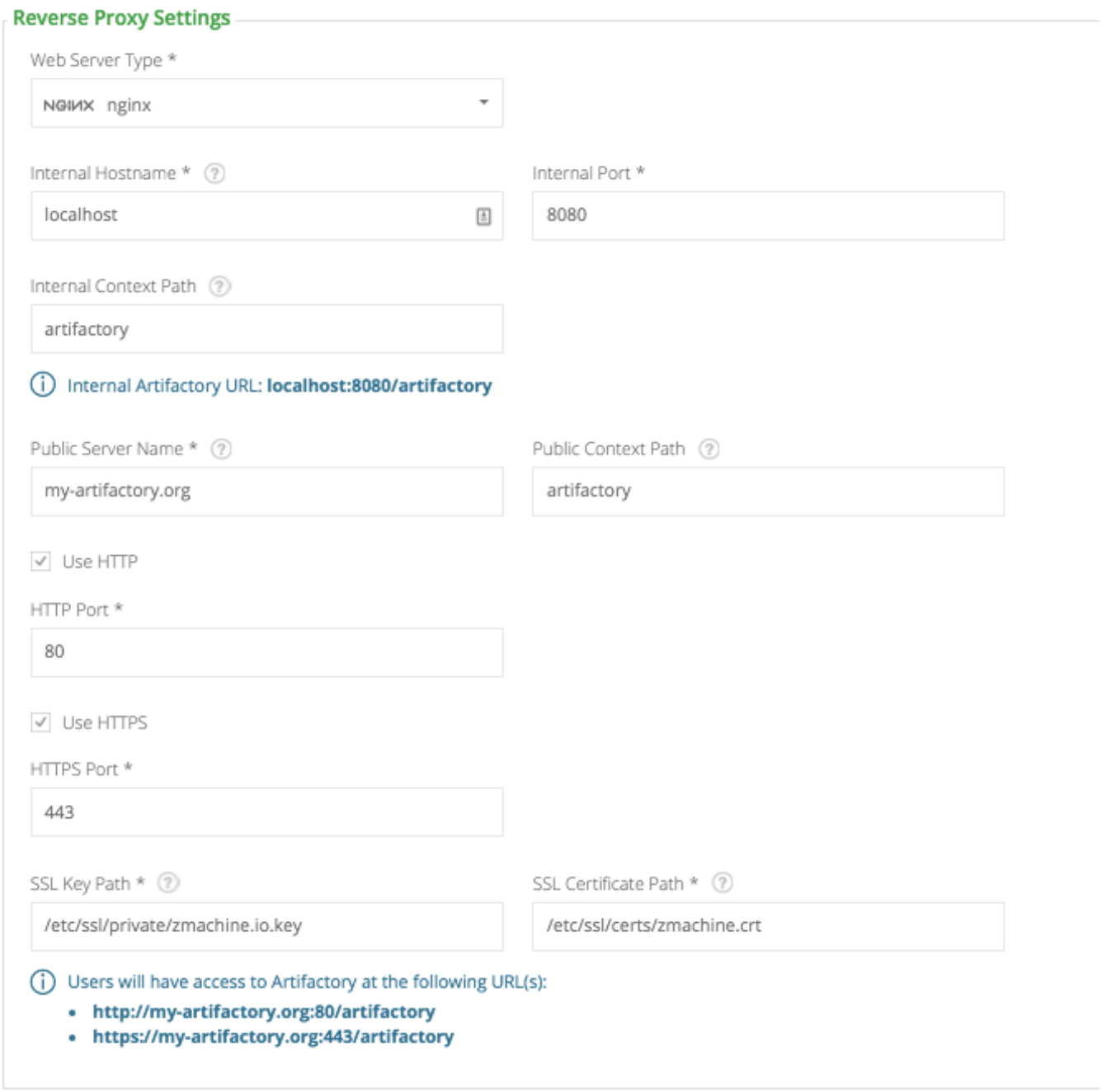
When using a reverse proxy, we recommend passing it the **X-Artifactory-Override-Base-Url** header as follows:

For NGINX:

```
proxy_set_header X-Artifactory-Override-Base-Url  $http_x_forwarded_proto://$<host>:<server port>
/<public context>
```

For Apache:

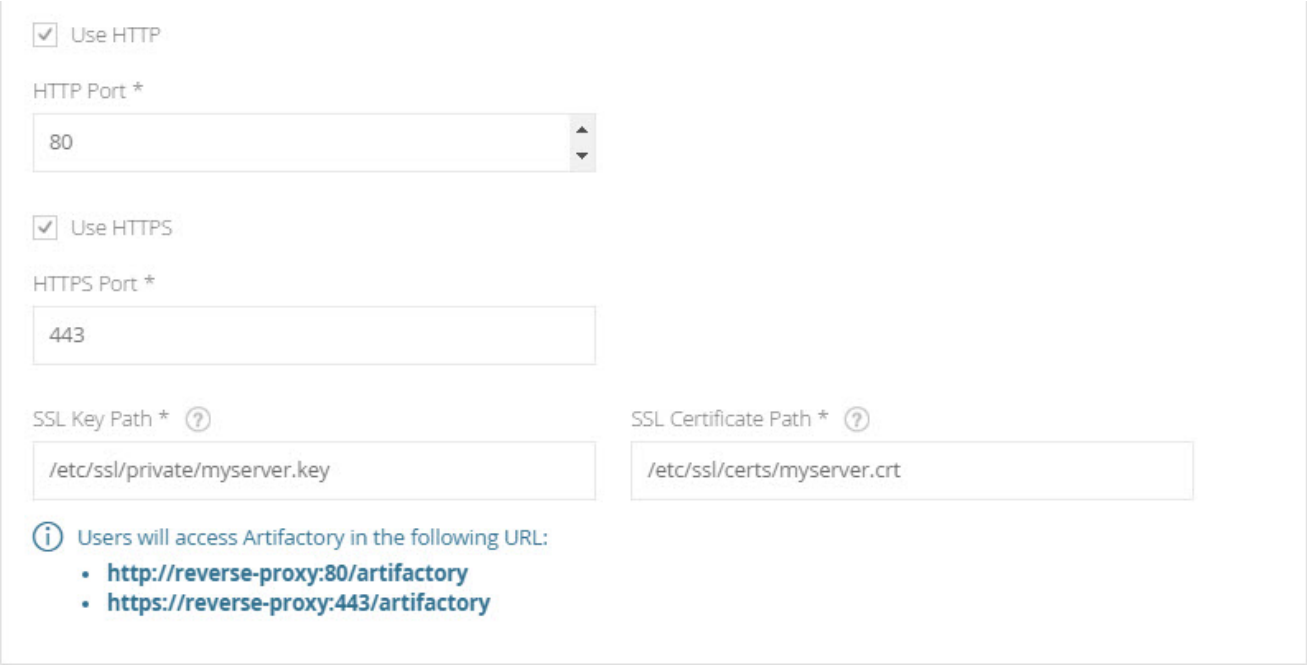
```
RewriteCond %{REQUEST_SCHEME} (.*)
RewriteRule (.*) - [E=my_scheme:%1]
[...]
RequestHeader set X-Artifactory-Override-Base-Url %{my_scheme}e://<server_name>/<app_context>
```



Web Server Type	The reverse proxy type. Currently NGINX and Apache are supported. Selecting Embedded Tomcat actually means that you are accessing Artifactory as a Docker registry using the Repository Path method, so the Tomcat embedded within Artifactory is routing requests from your Docker client to your Artifactory Docker registries.
Artifactory Server Name	The internal server name for Artifactory. If the Web Server is installed on the same machine as Artifactory you can use **localhost**, otherwise use the **IP address** or the **machine name**.
Artifactory Port	The port configured for Artifactory. The default value is 8081.
Artifactory Context Path	The path which will be used to access Artifactory. If Artifactory is accessible at the root of the server, leave this field empty.

Balance Members (Apache) Upstream Name (NGINX)	Only available in an Artifactory HA installation. Defines the group of servers in the HA cluster for load balancing. (default: artifactory). For more details, please refer to the NGINX documentation or Apache documentation accordingly. **Multiple Artifactory instances under the same domain** If using multiple Artifactory instances under the same domain, e.g. artdev.mycompany.org and artprod.mycompany.org you must assign a different names for balance members / upstream name to each cluster configuration since the session cookies will be available to both clusters and can cause an issue if trying to access both clusters in the same time.
Public Server Name	The server name which will be publicly used to access Artifactory within the organization.
Public Context Path	The path which will be publicly used to access Artifactory. If Artifactory is accessible on the root of the server leave this field empty.

You can configure access to Artifactory via HTTP, HTTPS or both (at least one is required). For each of these check boxes that you set, you need to fill in the corresponding fields as follows:



Use HTTP	When set, Artifactory will be accessible via HTTP at the corresponding port that is set.
HTTP Port	The port for access via HTTP. The default value is 80.
Use HTTPS	When set, Artifactory will be accessible via HTTPS at the corresponding port that is set.
HTTPS Port	The port for access via HTTPS. The default value is 443.
SSL Key Path	The full path to the key file for access via HTTPS.
SSL Certificate Path	The full path to the certificate file for access via HTTPS.

Docker Reverse Proxy Settings

When using Artifactory as an on-prem private Docker registry, the Docker client can access Artifactory through a reverse proxy or directly through Artifactory's embedded Tomcat.

JFrog Artifactory SaaS Docker Registries

Note that accessing an Artifactory Docker registry on a JFrog Artifactory SaaS installation does not use a reverse proxy since it is external to your organization.

Using a Reverse Proxy

The Docker client can access Artifactory through a reverse proxy using the Subdomain method (recommended) or through the Ports method.

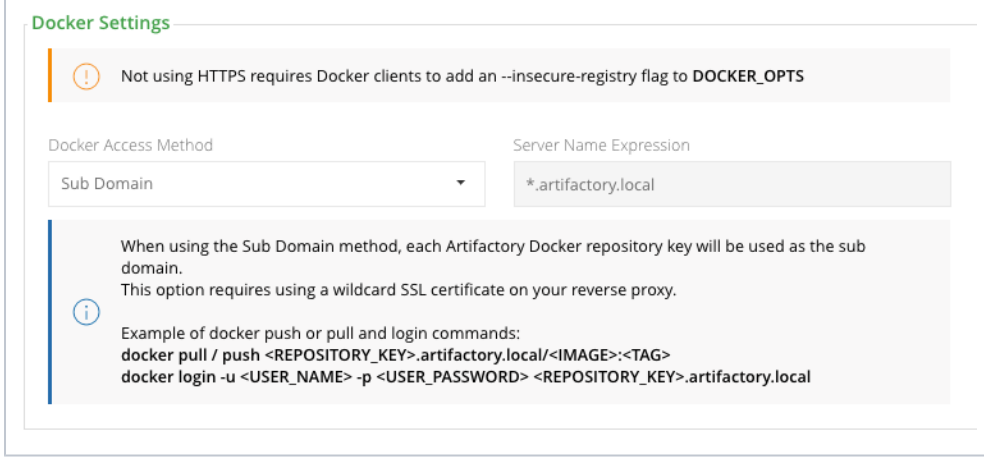
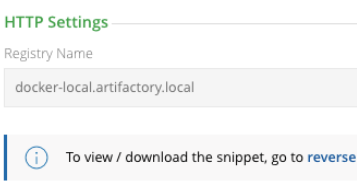
For each of these methods, your Docker repositories must be configured with the corresponding Reverse Proxy settings in the **Docker Repository Configuration Advanced** tab**.** The **Reverse Proxy Configuration** screen also sets up your Docker Repository configuration**.**

Using Subdomain

If you select **Subdomain** as the **Reverse Proxy Method**, when configuring a Docker Repository, the **Registry Name** in the **Docker Repository Configuration Advanced** tab will be set automatically to the required value, and will use the **Repository Key** as the **Subdomain**.

Wildcard certificate

Using the **Subdomain** method requires a **Wildcard** certificate such as. ***.myservername.org**. You also need to ensure that the certificate you use supports the number of levels used in your subdomain.

Docker Settings in HTTP Settings	Corresponding HTTP Settings in Doc
	

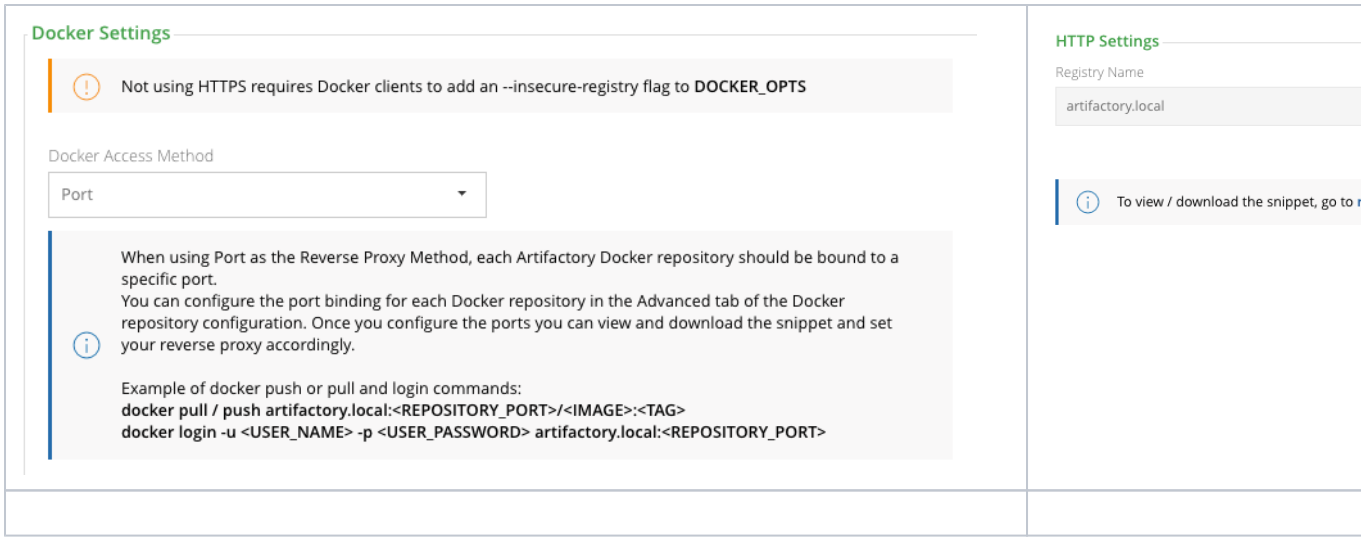
Using Port Bindings

If you select **Port** as the **Reverse Proxy Method**, when configuring a Docker Repository, you will need to set the **Registry Port** in the **Docker Repository Configuration Advanced** tab. Together with the Public Server Name, this is the port the Docker client will use to pull images from and push images to the repository. Note that in order for all of your Docker repositories to be included in your reverse proxy configuration, you first you need to set the port for each Docker repository defined in your system, and only then generate the reverse proxy configuration. Note also that each repository must be bound to a unique port

Best Practice

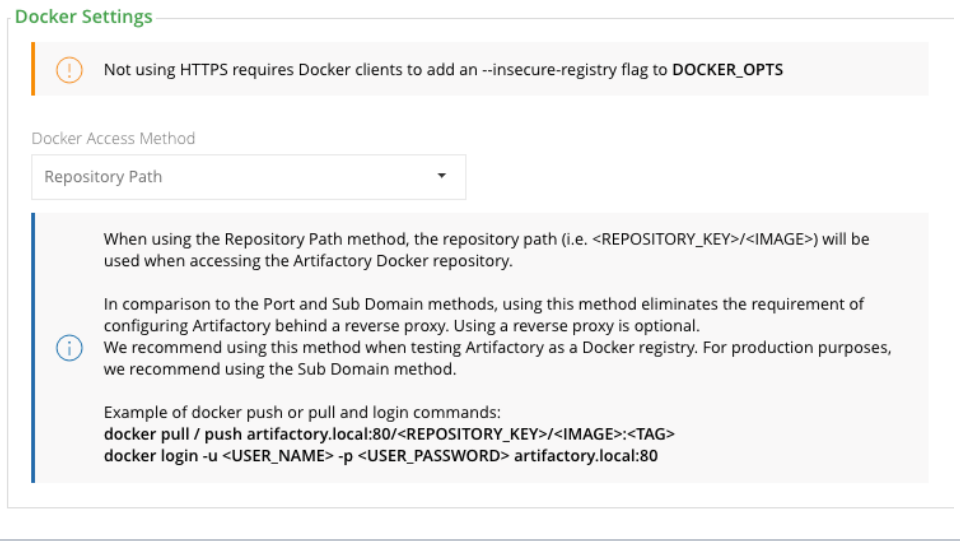
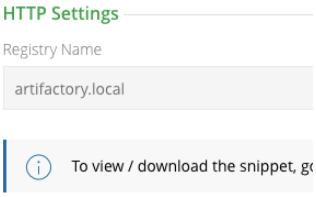
We recommend creating a Docker Virtual Repository which aggregates all of your other Docker repositories, and use that to pull and push images. This way you only need to set up the NGINX configuration for that virtual repository.

Docker Settings in HTTP Settings	Corresponding HTTP Settings in



Using Direct Access

To access your Docker repositories without using a reverse proxy, you should select Repository Path as the Docker Access Method in the Docker Setting Panel of the HTTP Settings screen.

Docker Settings in HTTP Settings	Corresponding HTTP Settings in
	

REST API

Artifactory also supports managing reverse proxy configuration through the REST API using the following endpoints:

Get Reverse Proxy Configuration	Retrieves the reverse proxy configuration JSON.
Update Reverse Proxy Configuration	Updates the reverse proxy configuration.
Get Reverse Proxy Snippet	Gets the reverse proxy configuration snippet in text format.