

Getting Started

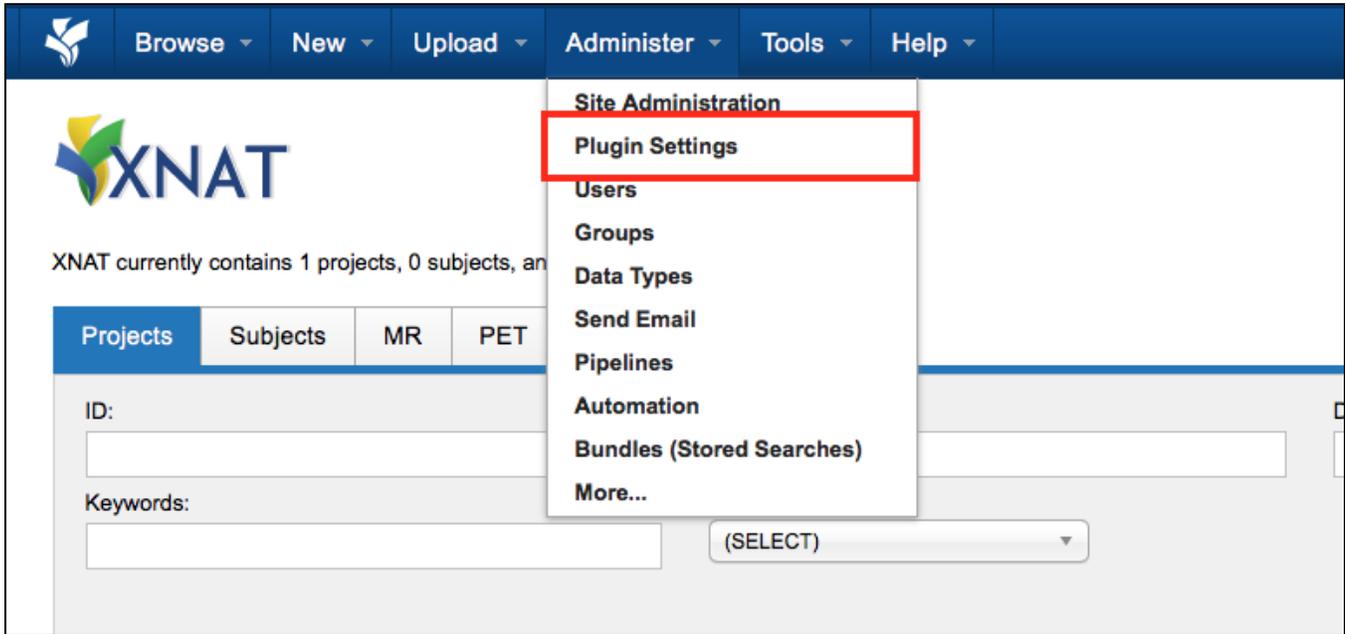
The XNAT Container Service plugin lets you execute external tools as lightweight containers. To get a container running, you need to

1. Find a tool, or Make one
2. Tell your XNAT about your tool
3. Launch it

(Technically there is also a step 0. Make sure your XNAT is set up properly. But in the most basic use case everything will be set up correctly out of the box.)

XNAT Settings

The settings for the container service are found under **Administer > Plugin Settings**.



Docker server URL

Plugin Settings

Container Service

- Container Server Setup
- Images & Commands
- Command Configurations
- Command Automation
- Command History

Docker Server Setup

Host Name	Host Path	Default	Actions
Local socket	unix:///var/run/docker.sock	<input type="radio"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Image Hosts

ID	Name	URL	Default	Actions
1	Docker Hub	https://index.docker.io/v1/	<input checked="" type="radio"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

XNAT needs to know how to communicate with your Docker server over HTTP. If your docker is installed with the default settings, it will listen for connections on a UNIX socket. If docker is listening on [the default socket](#), and XNAT can access that socket, then you won't need to change anything. The default container service settings correspond to the default docker settings.

If your docker is set up to listen on the default socket but your XNAT cannot communicate with docker, you may need to [add the xnat system user to the docker group](#).

If your docker server is on a separate machine from your XNAT server, and XNAT has no access to the docker socket, then you need to enable communication on both sides.

- [Enable docker to listen for TCP connections](#). These can be either secured (HTTPS) or unsecured (HTTP).
- Set up a new docker server in the container service administration page. (See screenshot above). Click the "New Container Host" button. Enter a user-friendly name for the server, the URL at which your docker server is listening, and (if you configured docker to listen over HTTPS) the path to the certificate file used to secure communications.



Docker HTTP

We recommend only allowing access to the docker server through the socket or HTTPS. Do not enable HTTP unless...

- Your docker server is behind a firewall. Enabling HTTP with no firewall would allow anyone on the internet to execute arbitrary code on your docker server.
- You are careful about the containers you allow XNAT users to execute. In particular, be very careful about allowing users to execute custom scripts. The containers run inside your firewall, which means a container could use the docker HTTP API to start new containers. This would allow users to completely control the docker server and view any XNAT data, bypassing the XNAT security model.

Enabling access to docker only through the socket or HTTPS does not create these problems.

Docker Hub

XNAT will, by default, attempt to pull images from the public Docker Hub with no authentication. If you need to pull private images, or you need to pull from a non-default Hub, you can configure those details by adding a new Hub to the list your XNAT knows.

Find or Create a Tool

What I'm calling a "tool" here is a Docker image that will execute whatever you need given the appropriate inputs, plus some metadata describing that image. We can currently only use metadata descriptors that are in our own proprietary format, which we call Commands (so named because we can use them to build the command-line string that will launch your containers). You need both the image and the Command for XNAT to launch a container.

Even though this section is titled "Find a Tool", there aren't currently any places you can do the "find" part. As of today, it is very possible that you'll have to create your own Docker images; and whether or not you created an image or found one already made, you will almost certainly have to write your own Commands. We have [a few Commands in a repo on GitHub](#), and we will add to that over time. But there is not yet a way to search for Commands for a particular image, or easily share them.

Docker Image

To execute your tool with the Container Service, it needs to be in a Docker image first. We plan to provide you with documentation to help you through this process, but it doesn't exist yet. Until then, you can consult the Docker documentation; [here is their guide on getting started](#).

Image Descriptor

For XNAT to understand how to launch containers from your image, you need to describe the contents of the image using a metadata format we call the Command. ([See this page or more detail on the Command format](#).)

The Command is an object that gives XNAT all the information it will need to launch your container given some inputs.

Other descriptor formats—like Boutiques format (and, by extension, anything from the Flywheel Exchange) and Common Workflow Language—are not yet supported. You can let us know if you have an image descriptor format that you want us to support in the Container Service. We'll consider adding support for it.

Get your Tool into XNAT

There are several ways to get your tool into XNAT. The easiest way is to pull an image from Docker Hub that is XNAT-enabled. You can try this by pulling the image `xnat/debug-command:latest`. The container service will first instruct docker to pull the image from docker hub, then will read the command definitions from the docker image's labels.

Plugin Settings

Container Service

Container Server Setup

Images & Commands

Command Configurations

Command Automation

Command History

Installed Container Images and Commands

xnat/debug-command:1.2

Add New Command

Delete Image

Command	XNAT Actions	Site-wide Config	Version	Actions
debug	Run the debug container with a project mounted <ul style="list-style-type: none">xnat:projectData	N/A	1.2	<p>View Command</p> <p>Delete</p>
	Run the debug container with a subject mounted <ul style="list-style-type: none">xnat:subjectData			
	Run the debug container with a session mounted <ul style="list-style-type: none">xnat:imageSessionData			
	Run the debug container with a scan mounted <ul style="list-style-type: none">xnat:imageScanData			
	Run the debug container with a assessor mounted <ul style="list-style-type: none">xnat:imageAssessorData			

xnat/dcm2nix:1.4

Add New Command

Delete Image

Command	XNAT Actions	Site-wide Config	Version	Actions
---------	--------------	------------------	---------	---------

alpine:latest

Add New Command

Delete Image

Command	XNAT Actions	Site-wide Config	Version	Actions
---------	--------------	------------------	---------	---------

Add New Image

Pull New Image ✕

Pull New Image

Image Name

Specify the image name to pull, including the account name (if necessary). For example, to pull the XNAT dcm2niix image, enter "xnat/dcm2niix"

Version Tag

"Specify the tag version of the image to pull. The default is :latest"

If you pull an image from docker hub that is not XNAT-enabled, you will need to add the command definition(s) yourself. In the AdministerPlugin SettingsImages & Commands interface, find the image for which you want to add a command and click the `Add new command` button. A window will appear into which you can paste or type the command JSON.

Plugin Settings

Container Service

Container Server Setup

Images & Commands

Command Configurations

Command Automation

Command History

Installed Container Images and Commands

xnat/debug-command:1.2

Add New Command

Delete Image

Command	XNAT Actions	Site-wide Config	Version	Actions
debug	Run the debug container with a project mounted <ul style="list-style-type: none">xnat:projectData	N/A	1.2	<p>View Command</p> <p>Delete</p>
	Run the debug container with a subject mounted <ul style="list-style-type: none">xnat:subjectData			
	Run the debug container with a session mounted <ul style="list-style-type: none">xnat:imageSessionData			
	Run the debug container with a scan mounted <ul style="list-style-type: none">xnat:imageScanData			
	Run the debug container with a assessor mounted <ul style="list-style-type: none">xnat:imageAssessorData			

xnat/dcm2nix:1.4

Add New Command

Delete Image

Command	XNAT Actions	Site-wide Config	Version	Actions
---------	--------------	------------------	---------	---------

alpine:latest

Add New Command

Delete Image

Command	XNAT Actions	Site-wide Config	Version	Actions
---------	--------------	------------------	---------	---------

Add New Image

Add New Command to xnat/dcm2niix:1.4
✕

1

Changes are not submitted automatically.
The containing form will need to be submitted to save.
Cancel
Save Command

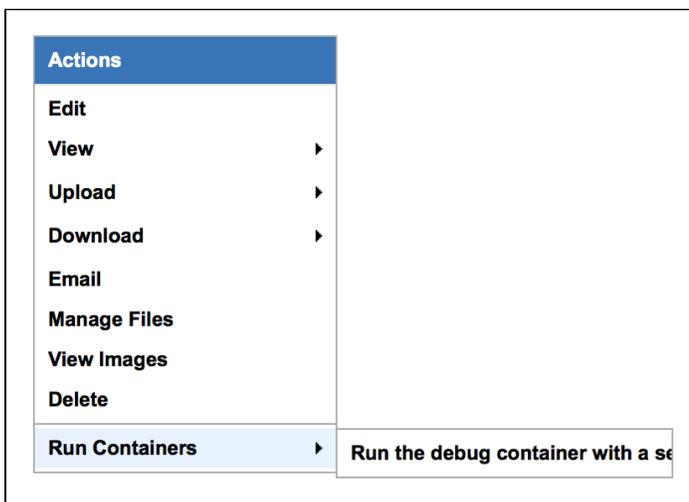
Alternatively, you can use a REST client to POST the command JSON to `$(your_xnat)/xapi/commands` to create a new command.

Launching a Container

You can launch a container through the REST API or through the XNAT user interface.

The REST API is direct: launch a container by POSTing any parameter values it needs to one of various REST endpoints (see [How to Launch Containers with the Container Service](#) for a list).

To launch through the user interface, just navigate around XNAT as normal. When a container can be launched from where you are (say, a subject page or a session page) a "Run containers" Action will appear in the Actions box. Hover over it to see the list of available containers.



Clicking on one brings up a window with that container's parameters.

Set Container Launch Values ✕

Please specify settings for this container.

session 5Yp0E

output-file
Name of the file to collect stdout

command
The command to run (Required)

Fill out the parameters as appropriate, and click the "Run Container" button to launch. You will see a confirmation window telling you if the container was successfully launched or not. If it was launched, you will see the container's docker ID; if not, you will see an error message telling you what happened.

Container Launch Success ✕

Container ID: fbd080e1af1d0407c5c433d0ee334c557c62b1fd9084617ec3c4b98cb00a436c

See [How to Launch Containers with the Container Service](#) for more information about how this works.