Introduction

What Is The XNAT Container Service?

🔄 PASSED

XNAT plugin for controlling containers (primarily Docker containers).

To use it, you will need an XNAT running 1.7.3+. Get the `containers-<version>-fat` jar (through one of the methods below) and put it into your `{xnat.home}/plugins` directory. Restart tomcat and you are ready to run containers. See the guide to getting started.

This document is cross-posted on the XNAT wiki and the README in the source repository.

Getting the jar

Download

Releases are posted to the repository's Releases tab on GitHub. Download the version you want (probably the latest release) and deploy it to XNAT.

Build the jar

If you clone the source repository, you can build an XNAT plugin jar by running

```
[container-service] $ ./gradlew fatJar
```

The jar will be created as `build/libs/containers-${VERSION}-fat.jar`

Deploy to XNAT

One you have a jar, copy it to the `${xnat.home}/plugins` directory, and restart tomcat.

Where is `${xnat.home}`? If you are using a VM generated by the XNAT Vagrant project, `xnat.home` is in the `~/${PROJECT}` directory (and the default value for `${PROJECT}` is `xnat`). If you aren't using the Vagrant project, or even if you are and you're still confused, then `${xnat.home}/logs` is where XNAT writes its logs; you'll want the `plugins` directory which should be right next to the `logs` directory.

Run the Tests

The various unit tests can be run with

```
$ ./gradlew unitTest
```

If you have a docker server that you can use for testing, there are some additional tests that can run. Make sure your docker environment is all set up (you can test this by making sure `$ docker version` works). Then run all the tests with

```
./gradlew test
```

We do not have any tests that can integrate with a running XNAT. All of the tests in this library use bespoke databases and mocked interfaces any time the code intends to communicate with XNAT. We welcome your contributions!