

Testing

Introduction

jclouds owes a lot of its success to the reliability it gives its users, and good test coverage is an essential ingredient in this reliability. Testing can be made more challenging because the services it connects to are live, remote and changeable -- but an immense amount of effort has gone in to ensuring testability with the result that with a bit of up-front learning, you shouldn't find it too difficult. (And if you do, or it could be easier, just mail the list!)

The two main categories of tests used in jclouds are:

Live Tests: running with credentials against a service provider, asserting that the resources (compute VM, blob storage, etc) actually get created, changed, etc.

Unit Tests: asserting that functions do what they should -- like normal unit tests -- but focused in many cases on ensuring that clients parse responses correctly and do the right thing: when writing provider unit tests, expect to use curl extensively (cheat sheet below), with the nice side benefit that you have evidence when, heaven forbid, a provider mysteriously changes schema!

These types of tests are described below, particularly unit tests where a lot of magic has been done to facilitate testing.

Running tests from the command-line

There are two common ways of running tests that connect directly against the service, through your favorite IDE or via the command-line using Maven. This section is about how to run using Maven.

Preliminaries

We currently use Maven 3.0.3, so ensure you have "mvn" in your path. To test this out, issue the mvn -version command. It should look similar to the output below:

```
Apache Maven 3.0.3 (r1075438; 2011-02-28 09:31:09-0800)
Maven home: /Users/<username>/apache-maven-3.0.3
Java version: 1.6.0_29, vendor: Apple Inc.
Java home: /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home
Default locale: en_US, platform encoding: MacRoman
OS name: "mac os x", version: "10.7.2", arch: "x86_64", family: "mac"
```

Note you should be in the directory of the service you'd like to test. For example, if you are testing the terremark module, you should already have cloned jclouds and changed into the vcloud/terremark directory. For example:

```
$ git clone git://github.com/apache/jclouds.git
Initialized empty Git repository in /private/tmp/jclouds/.git/
remote: Counting objects: 58795, done.
remote: Compressing objects: 100% (14698/14698), done.
remote: Total 58795 (delta 28015), reused 58636 (delta 27916)
Receiving objects: 100% (58795/58795), 17.41 MiB | 45 KiB/s, done.
Resolving deltas: 100% (28015/28015), done.
$ cd jclouds/vcloud/terremark/
```

Running Unit Tests

Unit tests run automatically as part of the build. To run these for the whole project, simply run: mvn clean install

This operation may take a while. To run for only the project you are actively working on, run the same command from the root directory of that project.

Running Live Tests

To run against the live service, you'll need to specify the Maven live Profile (-Plive). When this profile is enabled, any tests that have LiveTest suffix will be run during the Maven integration-testphase. In order for this to operate, you must specify the following either inside your ~/.m2/settings.xml or directly on the command line:

- test.provider.identity
- test.provider.credential (some clouds do not require this)

The following parameters can also be specified (optional):

- test.provider.endpoint

- test.provider.api-version
- test.provider.build-version (when an implementation targets a specific build running on the server)
- test.provider.image-id (compute)
- test.provider.image.login-user (compute, as username or [username:password](#))
- test.provider.image.authenticate-sudo (compute, password for username above is required for sudo)

Here's an example of running a live test with a specific username and password (from the providers/trmk-vcloudexpress directory):

```
mvn -Plive clean install -Dtest.trmk-vcloudexpress.identity=user@jclouds.apache.org -Dtest.trmk-vcloudexpress.endpoint=https://services.vcloudexpress.terremark.com/api -Dtest.trmk-vcloudexpress.credential=12312412
```