Travis CI

Our Travis CI is no longer the current method for building Grouper artifacts. See the Jenkins page for the current method.

Quick start

For a normal commit: Just push to the master branch, wait about 10 minutes and check Travis (https://travis-ci.com/Internet2/grouper) for a successful build

For a numbered release:

- Tag the desired commit with string GROUPER_RELEASE_a.b.c (a, b, c are numbers)
- Check Travis (https://travis-ci.com/Internet2/grouper) in about 20 minutes to check for successful build and upload to Sonatype
- Log into Sonatype and release the artifacts from holding (see this page for details)

Current Status

As of December 2021, our open source account on travis-ci.com had run out of free credits, and had stopped building releases. There was an option to petition for more free credits that would have lasted for a while. However, it was decided at that time to move the process to our own Jenkins build process which used servers owned by Internet2.

Detailed Description

Grouper is configured against Travis CI to execute the full build on every commit. For tags matching a specific pattern of "GROUPER_RELEASE_a.b.c (rc#)" it will also build release artifacts using that version from the tag, and publish those artifacts to the Maven Sonatype repository. There is an Internet2 /Grouper account in Travis connected to the Grouper Github repository, and you (?) should have access to view the Travis build status if you have developer access in Github (CR Feb/2020 not sure about this). .

Travis builds all grouper branches where there is a .travis.yml at the root of the repository, as long as the current branch is specified as an allowed branch in that file. This is a YAML configuration that tells Travis what version of Java should be available, which commands to use for the build lifecycle and which scripts to execute after a successful build. Before Grouper 2.5.0, the Travis setup included a post-build step to publish the Maven snapshot artifacts to the Sonatype snapshot repository. As of version 2.5.0, this no longer publishes snapshot artifacts, as they were of limited use. Instead, it simply does a `mvn package` phase of grouper-parent (repeated for the set of Java vendors and versions defined in the YAML file) to test for a successful build of all projects. If the Travis build status for the current branch changes from success->fail or fail->success, the committer should get an email reporting the status change.

For a pushed tag that matches GROUPER_RELEASE_n.n.n or GROUPER_RELEASE_n.n.n"rc"n, a second Travis job will be initiated. As with normal commits, the job will start with a maven "package" goal for all defined Java targets. If successful, the next step will be to execute the script travis/deploy-to-sonatype.sh, to rebuild the artifacts as release versions and publish them to our Sonatype staging repository. The script will parse out the dotted version out of the tag string, update the versions in the pom.xml, rebuild all the projects, generate source and javadoc artifacts, and then sign them with an included PGP key. Finally, the script will deploy all these artifacts and associated pgp signatures (*.asc files) to the Sonatype staging repository.

If successful, the artifacts will appear in a new folder in our Sonatype staging repository (https://oss.sonatype.org/), Currently (Feb 2020), there is a shared account to access this repository. A subfolder will appear with a name such as "eduinternet2middlewaregrouper-####". The status should be "closed", indicating that Travis was able to finalize its upload of the artifacts. In this state, the repository can be tested as a private repository, by adding it as a profile in maven settings.xml (see below), or it can be promoted to "release" which will publish it in the public repository. After being released, the artifacts will eventually be propagated to other Maven repository sites, such as https://search.maven.org/ and https://mvnrepository.com/.

Notes on encrypting values for Travis

Because Travis requires certain secrets for the Sonatype login and PGP encryption to be accessible via a Git repository, encrypted versions of the files can be stored and then decrypted as needed by the Travis jobs. If a value needs to be modified or added, There is a ruby-based script that can do the encryption. It can be installed in a custom Docker container as a one-off process, or installed directly on a Unix workstation. The following steps have been successful in getting a working Docker container for the travis client.

1) Install an improved RNG service on the host

sudo apt install rng-tools5
sudo rngd -r /dev/urandom

2) Set up a new API key in your Github account

- Go to https://github.com/settings/tokens

- (roles for old legacy .org site) ~~Generate a personal access token (name "travis command line client", access read:org, repo:status, repo_deployment, user:email, write:repo_hook)~~
- Generate a personal access token (name "travis command line client", access `read:org, repo, user:email, write:repo_hook` – see https://github.com/travis-ci/travis.rb/issues/708#issuecomment-697005010)
- Record the resulting API token, since it won't be repeated

If you try to execute `travis login` without setting up the api key, you will get an email detailing these same steps

3) Initialize the docker container

host

```
#host
docker pull ubuntu:bionic
docker run --name travis-client -it ubuntu:bionic bash
```

container

```
#container
apt-get update
apt-get install -y gcc make ruby-dev
gem install travis
```

4) Log into the travis client

```
travis login --com --github-token {token}
```

(result: Successfully logged in as <github-account>!)

5a) Encrypt a value

```
travis encrypt SONATYPE_USER=_secret_ -r Internet2/grouper --pro
travis encrypt SONATYPE_PWD=_secret_ -r Internet2/grouper --pro
# (not currently used) travis encrypt GH_TOKEN=_secret_ -r Internet2/grouper --pro
```

(Our Travis account is on travis-ci.com; if it were on travis-ci.org we would remove parameter --pro)

The result will go into .travis.yml, added to the section:

```
env:
  global:
    - secure: "....."
```

5b) OR encrypt a file

```
travis encrypt-file secretfile secretfile.enc -r Internet2/grouper
```

The resulting encrypted file can then be added to the Git repository (preferably the /travis/ subdirectory). The output will give helpful information on additional openssl job steps to be added to .travis.yml that will decrypt the file before it is needed.