
nebulizer Documentation

Release 0.4.3

Peter Briggs

Oct 05, 2018

1 Quick Start	3
1.1 Getting Nebulizer	3
1.2 Nebulizer Basics	3
1.3 Managing Users	4
1.4 Managing Data Libraries	4
1.5 Managing Tools	5
1.6 Checking status of a Galaxy server	5
2 Commands	7
2.1 User Management	7
2.2 Data Library Management	7
2.3 Tool Management	7
2.4 Bulk Tool Repository Management	8
2.5 Local API Key Management	8
2.6 Other commands	8
3 Hints and Tips	9
3.1 Managing Galaxy API keys	9
3.2 Handling SSL Certificate Verification Failures	10
3.3 Accessing Galaxy with Email & Password instead of API key	10
3.4 Installing Multiple Tool Repositories from a List	10
4 Deprecated Utilities	11
5 License	13
6 History	15
6.1 v0.4.3 (2018-10-05)	15
6.2 v0.4.2 (2017-08-24)	15
6.3 v0.4.1 (2016-12-19)	15
6.4 v0.4.0 (2016-11-18)	16
6.5 v0.3.0 (2016-10-26)	16
6.6 v0.2.0 (2016-10-17)	16
6.7 v0.1.1 (2016-05-16)	17
6.8 v0.1.0 (2015-11-06)	17
7 Indices and tables	19

Contents:

nebulizer

Command-line utilities to help with managing users, data libraries and tools in a [Galaxy](#) instance, using the Galaxy API via the [Bioblend](#) library. `build` `passing`

- Free software: Academic Free License version 3.0
- Documentation: <https://nebulizer.readthedocs.io>
- Code: <https://github.com/pjbriggs/nebulizer>

Warning: **CAVEAT** `nebulizer` is still a work in progress; please exercise caution when attempting irreversible operations especially against production Galaxy instances (for example when creating users or data libraries).

This quick start gives some examples of using `nebulizer` commands to perform remote administration tasks on a Galaxy instance from the command line.

1.1 Getting Nebulizer

It is recommended to install Nebulizer via `pip` in a `virtualenv`, for example:

```
% virtualenv .venv; . .venv/bin/activate
% pip install nebulizer
```

This will provide an executable called `nebulizer` with a number of subcommands for performing different tasks remotely on Galaxy instances.

1.2 Nebulizer Basics

To interact remotely with a Galaxy instance using Nebulizer requires at minimum the URL of the instance and then either an API key or a user login name.

For example to list data libraries available on Galaxy Main:

```
% nebulizer -k 9b376af2250818d14949b3c list_libraries https://usegalaxy.org
```

or:

```
% nebulizer -u peter.briggs@manchester.ac.uk list_libraries https://usegalaxy.org
```

(in this second case `nebulizer` will prompt for the Galaxy password to authenticate the user login.)

Specifying full API keys and Galaxy URLs each time a command is run is tedious, so Nebulizer can store URL-key pairs locally to make this easier.

For example to store the API key for Galaxy main:

```
% nebulizer add_key main https://usegalaxy.org 9b376af2250818d14949b3c
```

stores the API key and URL pair and associates it with the alias main.

Alternatively Nebulizer can fetch the API key itself if the user login is provided instead, for example:

```
% nebulizer -u peter.briggs@manchester.ac.uk add_key main https://usegalaxy.org
```

The stored alias can then be used as a substitute for the URL with the the stored API key being fetched behind the scenes. Then to list the data libraries again it is sufficient to do just:

```
% nebulizer list_libraries main
```

To find out which user is associated with an alias do:

```
% nebulizer whoami main
```

The following sections contain examples of how Nebulizer might be used to perform various administrative tasks.

1.3 Managing Users

List users matching specific name:

```
nebulizer list_users galaxy --name="*briggs*"
```

Add a new user:

```
nebulizer create_user galaxy -p pa55w0rd a.non@galaxy.org
```

1.4 Managing Data Libraries

List data libraries:

```
nebulizer list_libraries galaxy
```

Create a data library called NGS data and a subfolder Run 21:

```
nebulizer create_library galaxy \  
  --description="Sequencing data analysed in 2015" "NGS data"  
nebulizer create_library_folder localhost "NGS data/Run 21"
```

List contents of this folder:

```
nebulizer list_libraries galaxy "NGS data/Run 21"
```

Upload files to it from the local system:

```
nebulizer add_library_datasets galaxy "NGS data/Run 21" ~/Sample1_R*.fq
```

Add a file which is on the Galaxy server filesystem to a library as a link:


```
nebulizer add_library_datasets galaxy --server --link "NGS data/fastqs" \  
/galaxy/hosted_data/example.fq
```

1.5 Managing Tools

List all tools that are available in a Galaxy instance:

```
nebulizer list_tools galaxy
```

List all the cuff... tools that were installed from a toolshed:

```
nebulizer list_tools galaxy --name="cuff*" --installed
```

List all the tool repositories that are installed along with the tools that they provide:

```
nebulizer list_installed_tools localhost --list-tools
```

List all the tool repositories that have available updates or upgrades:

```
nebulizer list_installed_tools localhost --updateable
```

Install the most recent FastQC from the main toolshed:

```
nebulizer install_tool localhost \  
--tool-panel-section="NGS: QC and manipulation" \  
toolshed.g2.bx.psu.edu devteam fastqc
```

Update FastQC tool to latest installable revision:

```
nebulizer update_tool localhost toolshed.g2.bx.psu.edu devteam fastqc
```

Warning: By default checks on the availability of updates for tools performed by the `list_installed_tools` and `update_tool` commands are done using information cached by the Galaxy instance in question. As a result these commands may not always indicate when updates are available.

To force these commands to check the installed revisions against those in the toolshed, add the `--check-toolshed` option. Note however that this can impose a significant overhead which can make the commands much slower.

1.6 Checking status of a Galaxy server

‘Ping’ a Galaxy instance to check it’s alive and responding to requests:

```
nebulizer ping localhost
```


All functionality is available as subcommands of the `nebulizer` utility.

2.1 User Management

- `list_users`: List users in Galaxy instance.
- `create_user`: Create new Galaxy user.
- `create_batch_users`: Create multiple Galaxy users from a template.
- `create_users_from_file`: Create multiple Galaxy users from a file.

2.2 Data Library Management

- `list_libraries`: List data libraries and contents.
- `create_library`: Create new data library.
- `create_library_folder`: Create new folder in a data library.
- `add_library_datasets`: Add datasets to a data library.

2.3 Tool Management

- `list_tools`: List tools in Galaxy instance.
- `list_tool_panel`: List tool panel contents.
- `list_installed_tools`: List installed tool repositories.
- `install_tool`: Install tool from toolshed.

2.4 Bulk Tool Repository Management

- `list_repositories`: List installed tool repos for (re)install.
- `install_repositories`: Install tool repositories listed in a file.

2.5 Local API Key Management

- `add_key`: Store new Galaxy URL and API key.
- `list_keys`: List stored Galaxy API keys.
- `remove_key`: Remove stored Galaxy API key.
- `update_key`: Update stored Galaxy API key.

2.6 Other commands

- `ping`: 'Ping' a Galaxy instance.
- `whoami`: Print user details associated with API key.

3.1 Managing Galaxy API keys

Nebulizer stores the URL-key pairs in the file `.nebulizer` located in the user's home directory. This file consists of tab-delimited lines with the following columns:

```
alias|Galaxy_URL|API_key
```

This file can be edited by hand using a text editor such as `vi`; however Nebulizer provides a set of commands for querying and modifying the file contents.

To list the stored aliases with associated Galaxy URLs and API keys:

```
% nebulizer list_keys
```

To add a new alias called 'production' for a Galaxy instance:

```
nebulizer add_key production http://galaxy.org/ 5e7a1264905c8f0beb80002f7de13a40
```

Update the API key for 'production':

```
nebulizer update_key production --new-api-key=37b6430624255b8c61a137abd69ae3bb
```

Remove the entry for 'production':

```
nebulizer remove_key production
```

Multiple URL-key pairs can be stored; only the associated aliases need to be unique. For example:

```
% nebulizer -u admin@galaxy.org add_key palfinder https://palfinder.ls.manchester.ac.uk
...prompt for password...
% nebulizer list_libraries palfinder
```

3.2 Handling SSL Certificate Verification Failures

Nebulizer commands will fail for Galaxy instances which are served over `https` protocol without a valid SSL certificate, reporting an error like:

```
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:590), 0 attempts_
↪left: None
```

In this case adding the `--no-verify (-n)` option turns off the certificate verification and should enable a connection to be made.

3.3 Accessing Galaxy with Email & Password instead of API key

It is possible to use your normal Galaxy login credentials (i.e. your email and password) to access the API on a Galaxy instance without using the API key, using the `-u/--username` option, e.g.:

```
nebulizer -u joe.bloggs@example.com list_libraries "NGS data/Run 21"
```

You will be prompted to enter the password; however you can also use the `-P/--galaxy_password` option to specify it explicitly on the command line.

3.4 Installing Multiple Tool Repositories from a List

It is possible to install a list of tool repositories into a Galaxy instance by using the `install_repositories` command:

```
nebulizer install_repositories galaxy tools.tsv
```

The `tools.tsv` file must be a tab-delimited list of repositories, one repository per line in the format:

```
TOOLSHED|OWNER|REPOSITORY|REVISION|SECTION
```

For example:

```
toolshed.g2.bx.psu.edu      devteam bowtie_wrappers 9ca609a2a421    NGS: Mapping
```

A list of tool repositories already installed in a Galaxy instance can be generated in this format using the `list_repositories` command:

```
nebulizer list_repositories galaxy > tools.tsv
```

In principle the combination of these two commands can be used to ‘clone’ the installed tools from one Galaxy instance into another.

For example to replicate the tools installed on the ‘Palfinder’ instance:

```
nebulizer list_repositories https://palfinder.ls.manchester.ac.uk > palfinder.tsv
nebulizer install_repositories http://127.0.0.1 palfinder.tsv
```

Deprecated Utilities

The following additional utilities are included for backwards compatibility but are deprecated and likely to be removed in a future version:

- `manage_users`: list and create user accounts
- `manage_libraries`: list, create and populate data libraries
- `manage_tools`: list and install tools from toolsheds

They are not documented further here.

CHAPTER 5

License

Nebulizer is licensed under the [Academic Free License \(AFL\)](#).

6.1 v0.4.3 (2018-10-05)

- Ensure that `click` dependency is version 6.7 or earlier, to avoid subcommand names changing from e.g. `list_users` to `list-users` (PR #49)

6.2 v0.4.2 (2017-08-24)

- Commands now explicitly return appropriate exit code values indicating success (0) or failure (non-zero values).
- New option `--check-toolshed` added to `list_installed_tools` and `update_tool` commands, to check installed revisions directly against those available in the toolshed (PR #41)
- Update `install_tool`, `update_tool` and `install_repositories` to install tool dependencies through a resolver (e.g. `conda`) by default (issue #43)
- New options added to `install_tool`, `update_tool` and `install_repositories` commands, to explicit control how tool and repository dependencies should be handled (PR #44):
 - `--install-tool-dependencies [yes|no]`: install tool dependencies via the toolshed, if any are defined (default is yes)
 - `--install-tool-dependencies [yes|no]`: install tool dependencies via the toolshed, if any are defined (default is yes)
 - `--install-resolver-dependencies [yes|no]`: install dependencies through a resolver that supports installation (e.g. `conda`) (default is yes)

6.3 v0.4.1 (2016-12-19)

- Fix broken `update_tool` command (PR #40).

6.4 v0.4.0 (2016-11-18)

- New subcommand `ping`: ‘ping’ a Galaxy instance to see if it’s responsive (PR #33).
- New subcommand `whoami`: reports user associated with the API key (PR #37).
- `add_library_datasets`: refuses to perform upload if using the master API key (essentially API key must have an associated user).
- `install_repositories`: prints a list of all tool repositories that couldn’t be installed.
- New `--timeout` and `--nowait` options added for `install_tool`, `update_tool` and `install_repositories` subcommands.
- Fix to treat tool repositories with status `New` as still installing when trying to install tools (PR #31).
- Some improvements to logging (PR #38).

6.5 v0.3.0 (2016-10-26)

- New class `tools.ToolPanel` and updates to existing `tools.ToolPanelSection` class.
- `install_tool`: fix behaviour so that command does nothing if a version is not specified and at least one version of the tool is already installed.
- `list_repositories` and `install_repositories`: new commands to generate a list of installed tool repositories from a Galaxy instance and then reinstall tool repositories from a list with the same format (PR #19).
- `install_tool`: fix incorrect reporting of target tool panel section (PR #20)
- `add_key` and `update_key`: fix automatic retrieval of API key, which only worked previously if connecting user was an admin account (‘PR #23 <<https://github.com/pjbriggs/nebulizer/pull/23>’ _)
- `list_tool_panel`: shows tools in order they appear in Galaxy when using `--list-tools` option.
- Deprecated utilities (`manage_users`, `manage_tools` and `manage_libraries`) issue warnings when run.
- License updated to Academic Free License (AFL).
- Initial version of documentation also made available via [ReadTheDocs](#) (PR #21)

6.6 v0.2.0 (2016-10-17)

- Implemented new `nebulizer` utility which provides all previous functionality via subcommands, plus commands for managing API keys automatically (old `manage_users`, `manage_tools` and `manage_libraries` utilities are still available for backwards-compatibility but are deprecated).
- New general options:
 - `-q/--suppress-warnings`: prevent warning messages from `nebulizer` commands.
- Various fixes and improvements to underlying functionality:
 - `install_tools`: now checks if tool is already installed; handles tool revisions that include the revision number; polls Galaxy until tool is installed, or operation times out; exit status reflects the success or failure of the installation.

- `update_tool`: now works even if original tool isn't in a tool panel section
- `list_installed_tools`: now groups tools under correct repo revision when using `--list-tools` option.

6.7 v0.1.1 (2016-05-16)

- Add `-u/--username` and `-P/--galaxy_password` options to all commands to allow interaction with Galaxy instance via API using normal login credentials instead of API key.

6.8 v0.1.0 (2015-11-06)

- Initial release of `nebulizer` utilities for administering Galaxy instances via the command line.

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`