stackyter Documentation

N. Chotard

Oct 15, 2021

Contents

CHAPTER 1

Quick install and how-to

Local display of a Jupyter notebook running on a distant server

1. Install the latest version of `stackyter` on you local machine:

```
pip install stackyter
```

2. Install Jupyter on your distant host if not done yet
3. Create a file with instructions to make Jupyter (and anything else you need) available (e.g, `mysetup.sh`)
4. Run `stackyter.py` on your local machine:

```
stackyter.py --host thehost --user myusername --mysetup /path/on/the/host/mysetup.
↪sh
```

5. Copy/paste the given URL into your local browser to display Jupyter

CHAPTER 2

Purpose

This script allow you to run a jupyter notebook (or lab) on a distant server while displaying it localy in your local brower. It can be used by anyone and on any host using the `--host` and `--mysetup` options. The only prerequisite is that **Jupyter must be available on the distant host for this script to work.**

CHAPTER 3

Installation

Latest stable version can be installed with `pip`:

```
pip install stackyter
```

To upgrade to a newer version:

```
pip install --upgrade stackyter
```

To install in a local directory:

```
pip install --user stackyter            #  in your home directory
pip install --prefix mypath stackyter   #  in 'mypath'
```

CHAPTER 4

Usage

```
stackyter.py [options]
```

Then click on the green link given by `stackyter`, as followed:

```
Copy/paste this URL into your browser to run the notebook localy
   http://localhost:20001/?token=38924c48136091ade71a597218f2722dc49c669d1430db41
```

`Ctrl-C` will stop the Jupyter server and close the connection.

You can use the following set of options to adapt `stackyter` to your personal case.

CHAPTER 5

Optional arguments

An option used on the command line will always overwrite the content of the configuration file for the same option, if defined. See the next section for a description on how to use the configuration file. Available options are:

```
-h, --help            show this help message and exit
-c CONFIG, --config CONFIG
                      Name of the configuration to use, taken from your
                      default configuration file (~/.stackyter-config.yaml
                      or $STACKYTERCONFIG). Default if to use the
                      'default_config' defined in this file. The content of
                      the configuration file will be overwritten by any
                      given command line options. (default: None)
-f CONFIGFILE, --configfile CONFIGFILE
                      Configuration file containing a set of option values.
                      The content of this file will be overwritten by any
                      given command line options. (default: None)
-H HOST, --host HOST  Name of the target host. Allows you to connect to any
                      host on which Jupyter is available, or to avoid
                      conflit with the content of your $HOME/.ssh/config.
                      (default: None)
-u USERNAME, --username USERNAME
                      Your user name on the host. If not given, ssh will try
                      to figure it out from you ~/.ssh/config or will use
                      your local user name. (default: None)
-w WORKDIR, --workdir WORKDIR
                      Your working directory on the host (default: None)
-j JUPYTER, --jupyter JUPYTER
                      Either launch a jupiter notebook or a jupyter lab.
                      (default: notebook)
--mysetup MYSETUP     Path to a setup file (on the host) that will be used
                      to set up the working environment. A Python
                      installation with Jupyter must be available to make
                      this work. (default: None)
--runbefore RUNBEFORE
                      A list of extra commands to run BEFORE sourcing your
```

(continues on next page)

(continued from previous page)

```
                      setup file. Coma separated for more than one commands,
                      or a list in the config file. (default: None)
--runafter RUNAFTER   A list of extra commands to run AFTER sourcing your
                      setup file. Coma separated for more than one commands,
                      or a list in the config file. (default: None)
-C, --compression     Activate ssh compression option (-C). (default: False)
-S, --showconfig      Show all available configurations from your default
                      file and exit. (default: False)
```

CHAPTER 6

Configuration file

A configuration dictionnary can contain any options available through the command line. The options found in the configuration file will always be overwritten by the command line.

The configuration file can be given in different ways, and can contains from a single configuration dictionnary to several configuration dictionnaries:

- The **configuration file** can either be a default file located under `~/stackyter-config.yaml` or defined by the `STACKYTERCONFIG`, or given in command line using the `--configfile` option.
- The **configuration name**, which should be defined in your configuration file, must be given using the command line option `--config`. If not given, a `default_config`, which should be defined in your configration file, will be used by default.

Here are a few example on how to use it:

```
stackyter.py  # 'default_config' in default file if it exists, default option values␣
↪used otherwise
stackyter.py --config config1  # 'config1' in default file which must exist
stackyter.py --config config2 --configfile myfile.yaml  # 'config2' in 'myfile.yaml'
stackyter.py --configfile myfile.yaml  # 'default_config' in 'myfile.yaml'
```

In principal, your default configuration file should look like that:

```
{
 'default_config': 'host1',

 'host1': {
           'host': 'myhost.domain.fr',  # 'myhost' if you have configured your ~/.ssh/
↪config
           'jupyter': 'lab',            # if installed
           'username': 'myusername',
           'mysetup': '/path/to/my/setup/file.sh',
           'workdir': '/path/to/my/directory/'
            },
```

(continues on next page)

(continued from previous page)

```
 'host2': {
           'host': 'otherhost.fr',
           'username': 'otherusername',
           'mysetup': '/path/to/my/setup'
          },

 'host3': {
           'host': 'somewhere.edu',
           'username': 'ausername',
           # Jupyter is available by default on this host, 'mysetup' is not needed
          },
}
```

or simply as followed if only one configuration is defined:

```
{
 'host1': {
           'host': 'myhost.domain.fr',  # or 'myhost' if you have configured your ~/.
↪ssh/config file
           'jupyter': 'lab',  # if installed
           'username': 'myusername',
           'mysetup': '/path/to/my/setup/file.sh',
           'workdir': '/path/to/my/directory/'
            },
}
```

You can use the example configuration file as a template to create your own. You can also find several example configuration files in the configs directory for different user cases.

CHAPTER 7

Distant host configuration

The `--host` option allows you to connect to any distant host. The default option used to create the `ssh` tunnel are `-X -Y -tt -L`. If you want to configure your `ssh` connection, edit your `~/.ssh/config` file using, for instance, the following template:

```
Host myjupyter
Hostname thehostname
User myusername
GSSAPIClientIdentity myusername@HOST
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
GSSAPITrustDns yes
```

You only need to replace `thehostname`, `myusername`, and `myusername@HOST` by the appropriate values. You can then use the `stackyter` script as follows:

```
stackyter.py --host myjupyter
```

Or put the value for that option (along with others) in your `config.yaml` file.

CHAPTER 8

Working environment

There are several ways to setup your personnal working environment, using the `--mysetup`, `--runbefore`, and `runafter` options. Given a setup file located on your distant host, you can simply do:

```
stackyter.py --mysetup /path/to/my/setup.sh (--username myusername)
```

Your local setup file will be sourced at connection as followed:

```
source /path/to/my/setup.sh
```

The `runbefore` and `runafter` options allow you to respectively run command lines before or after your setup file is sourced. It can be useful if you need to pass argument to your setup file through environment variables, or add extra command after the sourcing.

Your setup must **at least** contains what is needed to make Jupyter available. If Jupyter is available by default on the distant host (it might be set up on connection), you only need to use the `--host` and `--username` option to run.

You can of course add any kind of personal setups with these three options, related or not to Jupyter.

CHAPTER 9

Help

- If you have any comments or suggestions, or if you find a bug, please use the dedicated github issue tracker.
- Why `stakyter`? For historical reason: `stackyter` = LSST `stack` + `Jupyter`. It was initially intended for LSST members to easily use the LSST software stack and interact with data sets.