tg-hammer

Release 0.6.7

Aug 29, 2018

Contents

1 Introduction **1**

1.1 Features 1

1.1.1 vcs 1

1.1.2 service_helpers 1

2 Indices and tables **5**

Python Module Index **7**

CHAPTER 1

Introduction

Hammer provides unified helpers fabric based deployments.

1.1 Features

1.1.1 vcs

Unified helper api for both git and Mercurial based projects. It can automatically detect which version control system to use based on the current project (by inspecting project_root env variable).

1.1.2 service_helpers

Management helpers for the following unix service daemon utilities:

- upstart
- systemd
- supervisor

Install

Install tg-hammer:

```
pip install tg-hammer
```

Then use it in your fabfile:

```
from hammer.vcs import Vcs
```

(continues on next page)

(continued from previous page)

```
# Provide configuration to the VCS logic
# Note: You can omit both of these keys when you
#       want them to be retrieved from fabrics `env` dictionary
vcs_config = {
    'use_sudo': False,              # Set to True if your target machine requires␣
↪elevated privileges when running vcs commands
    'code_dir': '/srv/project',     # Directory on the target machine that will be␣
↪operated on
}
vcs = Vcs.init(project_root='path to root dir of project', **vcs_config)

# Now you can use the vcs api
vcs.repo_url()  # > git@github.com:thorgate/tg-hammer.git
```

API

Vcs

class hammer.vcs.**BaseVcs**(*project_root*, *use_sudo=None*, *code_dir=None*, ***kwargs*)

Core VCS api

changed_files(*revision_set*, *filter_re=None*)

Returns list of files that changed in the given revset, optionally filtered by the given regex or list of regex values.

Each returned item is a combination of the action and the file name separated by space: > [‘A test.tx’, ‘M foo.bar’, ‘R hello_world’]

Parameters

- **revision_set** – Revision set to use when building changed file list
- **filter_re** – optional regex filter (or list of regex values)

Returns files changed

Return type list

clone(*revision=None*)

Clones the project to a target machine. Will abort if something goes wrong.

Parameters **revision** – Can be used to specify a branch or commit that should be activated after cloning.

deployment_list(*revision=''*)

List revisions to apply/un-apply when updating to given revision (if not specified defaults to tip of currently active branch).

The returned dict can have the following keys:

- forwards: If there are revisions to apply, this value will contain a list with information about each commit
- backwards: If there are revisions to un-apply, this value will contain a list with information about each commit
- revset: If there are some revisions to apply/un-apply this will contain a string that can be passed on to changed_files

- message: If no revisions are to be applied or something else is wrong, this will contain this information

Parameters **revision** – Specific revision to diff against

Return type dict

get_branch()
Get current active branch in target machine.

Returns current active branch

Return type string

get_revset_log(*revs*)
Returns lines returned by hg|git log as a list.

Parameters **revs** – Revision set passed onto *hg|git log*

Returns list of commits (lines are in the following format: *commit_hash branch author description*)

Return type list

pull()
Update the cloned repository on the target machine without changing the working copy. Internally this is done via *git fetch* or *hg pull*.

repo_url()
Retrieve Url of the remote repository (origin|default). If remote url can't be determined None is returned.

Returns remote url

Return type str|None

update(*revision=''*)
Update the target to specified revision or tip of currently active branch if revision is omitted.

Parameters **revision** – Specific revision to update to

version()
Get the commit id, branch, message and author active on the target machine

Returns (commit_id, branch, message, author)

Return type tuple

service_helpers

hammer.service_helpers.**DAEMON_TYPES**
Hammer supports the following daemon types out of the box

- upstart
- systemd
- supervisor

The daemon type can specified via: `env.service_daemon`

hammer.service_helpers.**install_services_cp**(*services*, *daemon_type=None*, *daemon_target_dir=None*)
Install provided services by copying the remote file to the detected `daemon_type` specific directory

Parameters

- **services** – List of services to install where each item is a tuple with the signature: `(target_name, remote_file_path[, transform])`
- **daemon_type** – Can be used to override `env.service_daemon` value
- **daemon_target_dir** – Can be used to override `env.service_daemon_target_dir` value

The remote_file_path supports the following keywords:

- `${DAEMON_TYPE}`: Replaced with the detected daemon type (see *DAEMON_TYPES*)
- `${DAEMON_FILE_EXTENSION}`: Replaced with the *file_extension* value for the detected daemon type (see *DAEMON_TYPES*)

***transform* is an optional function w/ signature *(target_name, remote_file_data) -> (target_name, remote_file_data)* which** can be used for dynamic service configuration

Warning: For supervisor the default include dir is */etc/supervisord/conf.d/*, this directory must be included in the global supervisor configuration.

`hammer.service_helpers.`**`install_services`**(*services*, *daemon_type=None*, *daemon_target_dir=None*)

Install provided services by uploading configuration files to the detected `daemon_type` specific directory

Parameters

- **services** – List of services to install where each item is a tuple with the signature: `(target_name, file_data)`
- **daemon_type** – Can be used to override `env.service_daemon` value
- **daemon_target_dir** – Can be used to override `env.service_daemon_target_dir` value

Warning: For supervisor the default include dir is */etc/supervisord/conf.d/*, this directory must be included in the global supervisor configuration.

`hammer.service_helpers.`**`manage_service`**(*names*, *action*, *raise_errors=True*, *daemon_type=None*)

Perform *action* on services

Parameters

- **names** – Can be a list of services or a name of a single service to control
- **action** – Action that should be executed for the given services
- **raise_errors** – A way to control if errors generated by command should be captured by fabric or not. By default

it is set to raise errors :param daemon_type: Can be used to override env.service_daemon value

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

h

hammer.service_helpers, 3
hammer.vcs, 2

Index

B

BaseVcs (class in hammer.vcs), 2

C

changed_files() (hammer.vcs.BaseVcs method), 2
clone() (hammer.vcs.BaseVcs method), 2

D

DAEMON_TYPES (in module hammer.service_helpers), 3
deployment_list() (hammer.vcs.BaseVcs method), 2

G

get_branch() (hammer.vcs.BaseVcs method), 3
get_revset_log() (hammer.vcs.BaseVcs method), 3

H

hammer.service_helpers (module), 3
hammer.vcs (module), 2

I

install_services() (in module hammer.service_helpers), 4
install_services_cp() (in module hammer.service_helpers), 3

M

manage_service() (in module hammer.service_helpers), 4

P

pull() (hammer.vcs.BaseVcs method), 3

R

repo_url() (hammer.vcs.BaseVcs method), 3

U

update() (hammer.vcs.BaseVcs method), 3

V

version() (hammer.vcs.BaseVcs method), 3