dploi-fabric Documentation

Release 0.1.10

Benjamin Wohlwend, Kristian Oellegaard, Stefan Foulis

September 07, 2016

Contents

1 General Information **3**

2 Buildout Utilities **5**

3 Django Utilities **7**

4 Git Utilities **9**

5 Github integration **11**

6 Nginx utilities **13**
6.1 Configuration files . . . 13
6.2 Process management . . . 13
6.3 Settings . . . 13

7 Project layout and setup **15**

8 South Utilities **17**

9 Supervisor utilities **19**
9.1 Configuration files . . . 19
9.2 Process management . . . 19
9.3 Settings . . . 19

10 Remote Utilities **21**
10.1 Remote processes . . . 21
10.2 Media files sync . . . 21

11 Virtualenv Utilities **23**

12 NewRelic utilities **25**
12.1 Settings . . . 25
12.2 Deployment Tracking . . . 25

13 Internal utilities **27**

14 Indices and tables **29**

Usage:

CHAPTER 1

General Information

This project is designed to work together with dploi-puppet. Even though we prefer to make it as pluggable as possible, certain things (like directory structure) might be asserted to follow the guidelines from dploi-puppet.

CHAPTER 2

Buildout Utilities

The following functions are available through the CLI, e.g. fab <dev/stage/live/...> <command>

dploi_fabric.buildout.**run**()
: runs buildout

CHAPTER 3

Django Utilities

The following functions are available through the CLI, e.g. fab <dev/stage/live/...> <command>

dploi_fabric.django_utils.**manage**()
Proxy for manage.py

dploi_fabric.django_utils.**collectstatic**()

dploi_fabric.django_utils.**append_settings**()

CHAPTER 4

Git Utilities

The following functions are available through the CLI, e.g. fab <dev/stage/live/...> <command>

dploi_fabric.git.**update**()

dploi_fabric.git.**diff**()

dploi_fabric.git.**status**()

dploi_fabric.git.**reset**()
: discard all non-committed changes

dploi_fabric.git.**incoming**()
: Displays incoming commits

CHAPTER 5

Github integration

The following functions are available through the CLI, e.g. fab <dev/stage/live/...> <command>

dploi_fabric.github.**upload_ssh_deploy_key**()

CHAPTER 6

Nginx utilities

The following functions are available through the CLI, e.g. fab <dev/stage/live/...> <command>

6.1 Configuration files

dploi_fabric.nginx.**update_config_file**()

6.2 Process management

dploi_fabric.nginx.**reload_nginx**()

6.3 Settings

Can be set in config.ini in the `[nginx]` section or in `deployment.py` in this site `nginx` section:

- `client_max_body_size` (default: `10m`)
- `template` (default: bundled template): path to template for nginx.conf template (relative to project root)

CHAPTER 7

Project layout and setup

The following functions are available through the CLI, e.g. fab <dev/stage/live/...> <command>

dploi_fabric.project.**init**()

dploi_fabric.project.**upload_ssl**()
Upload the SSL key and certificate to the directories and with the filenames specified in your settings.

CHAPTER 8

South Utilities

The following functions are available through the CLI, e.g. fab <dev/stage/live/...> <command>

class dploi_fabric.south.**SouthMigrateTask**(*alias=None*, *aliases=None*, *default=False*, *name=None*, **args*, ***kwargs*)

CHAPTER 9

Supervisor utilities

The following functions are available through the CLI, e.g. fab <dev/stage/live/...> <command>

9.1 Configuration files

dploi_fabric.supervisor.**add**()

dploi_fabric.supervisor.**update**()

dploi_fabric.supervisor.**update_config_file**()

9.2 Process management

dploi_fabric.supervisor.**start**()

dploi_fabric.supervisor.**stop**()

dploi_fabric.supervisor.**restart**()

dploi_fabric.supervisor.**status**()
print status of the supervisor process

Note: "status" does not yet support the group syntax

9.3 Settings

Can be set in config.ini in the `[supervisor]` section or in `deployment.py` in this site `supervisor` section:

- `template` (default: bundled `dploi_fabric/templates/supervisor/supervisor.conf`)
- `group_template` (default: bundled `dploi_fabric/templates/supervisor/supervisor-group.conf`)
- `gunicorn_command_template` (default: bundled `dploi_fabric/templates/supervisor/gunicorn_comman`
- `celeryd_command_template` (default: bundled `dploi_fabric/templates/supervisor/celeryd_command`)
- `celerycam_command_template` (default: bundled `dploi_fabric/templates/supervisor/celerycam_comm`

CHAPTER 10

Remote Utilities

The following functions are available through the CLI, e.g. fab <dev/stage/live/...> <command>

10.1 Remote processes

dploi_fabric.utils.**ps**()
show processes of this user

10.2 Media files sync

dploi_fabric.utils.**download_media**()
Downloads media from a remote folder, default ../uploads/ -> ./tmp/media/

- Example: download_media:from_dir="py_src/project/media/"

dploi_fabric.utils.**upload_media**()
Uploads media from a local folder, default ./tmp/media -> ../uploads/

- Example: upload_media:to_dir="py_src/project/media/"

CHAPTER 11

Virtualenv Utilities

The following functions are available through the CLI, e.g. fab <dev/stage/live/...> <command>

dploi_fabric.virtualenv.**update**()
: updates a virtualenv (pip install requirements.txt)

dploi_fabric.virtualenv.**create**()
: creates a virtualenv and calls update

CHAPTER 12

NewRelic utilities

12.1 Settings

Can be set in config.ini in the `[newrelic]` section or in `deployment.py` in this site `newrelic` section:

- `enabled` (default: `False`)
- `config_file` (default: 'newrelic.ini'): path to `newrelic.ini` (relative to project root)
- `environment_name` (default: `''`): new relic environment name (for deployment specific settings in newrelic.ini)
- `license_key` (default: `''`): license key to override the one in `newrelic.ini`.
- `deployment_tracking_apikey` (default: `''`): new relic deployment tracking API key (this has to be set in `deployment.py`)

12.2 Deployment Tracking

You can track deployments and send them to New Relic's API. There are two parts to it:

- `@newrelic.register_deployment` by using this decorator the deployment info will be sent to New Relic's API after the deployment has run through
- `newrelic.log_output()` by using this context manager, you can specify which additional logging data should be sent

12.2.1 Example usage

```
from dploi_fabric import newrelic

@task
@newrelic.register_deployment
def deploy():
    with newrelic.log_output():
        pg.dump.run()
        git.update()
    virtualenv.update()
    south.migrate.run()
    django_utils.collectstatic()
```

```
    django_utils.manage('compress --force')
    run('~/bin/gunicorn restart')
```

For developers:

CHAPTER 13

Internal utilities

class dploi_fabric.utils.**Configuration**

This class is the only correct source of information for this project. To reduce the amount of times config.ini is downloaded, it should always be used from utils.config, which is an instance of Configuration

build_environment_export(*environment*)

takes a dict with environment variables and products a shell compatible export statement: 'export PYTHONPATH="stuff/here:more/here" USER="mysite-dev";'

defaults = {'sendfile': {}, 'supervisor': {'group_template': '/home/docs/checkouts/readthedocs.org/user_builds/dploi-fa

Default values for the configuration

deployment(*site*, *env_dict*)

Here we add the information from deployments.py and merge it into our site dictionaries. Can also be used to output warnings to the user, if he is using an old deployments.py format.

django_manage(*command*, *site='main'*)

Wrapper around the commands to inject the correct pythonpath.

Example: django_manage("migrate"), could result in

export PYTONPATH=/home/app-dev/app/; /home/app-dev/app/bin/python /home/app-dev/app/manage.py migrate

load_sites(*config_file_content=None*, *env_dict=None*)

Called from self.sites and returns a dictionary with the different sites and their individual settings.

processes(*site*, *env_dict*)

Returns a dictionary of dictionaries each having the following keys:

- **command** command to be run by supervisor
- **port** port number,
- **socket** path to unix socket
- **type** gunicorn/memcached/celeryd

CHAPTER 14

Indices and tables

- genindex
- modindex
- search

A

add() (in module dploi_fabric.supervisor), 19
append_settings() (in module dploi_fabric.django_utils), 7

B

build_environment_export() (dploi_fabric.utils.Configuration method), 27

C

collectstatic() (in module dploi_fabric.django_utils), 7
Configuration (class in dploi_fabric.utils), 27
create() (in module dploi_fabric.virtualenv), 23

D

defaults (dploi_fabric.utils.Configuration attribute), 27
deployment() (dploi_fabric.utils.Configuration method), 27
diff() (in module dploi_fabric.git), 9
django_manage() (dploi_fabric.utils.Configuration method), 27
download_media() (in module dploi_fabric.utils), 21

I

incoming() (in module dploi_fabric.git), 9
init() (in module dploi_fabric.project), 15

L

load_sites() (dploi_fabric.utils.Configuration method), 27

M

manage() (in module dploi_fabric.django_utils), 7

P

processes() (dploi_fabric.utils.Configuration method), 27
ps() (in module dploi_fabric.utils), 21

R

reload_nginx() (in module dploi_fabric.nginx), 13
reset() (in module dploi_fabric.git), 9
restart() (in module dploi_fabric.supervisor), 19
run() (in module dploi_fabric.buildout), 5

S

SouthMigrateTask (class in dploi_fabric.south), 17
start() (in module dploi_fabric.supervisor), 19
status() (in module dploi_fabric.git), 9
status() (in module dploi_fabric.supervisor), 19
stop() (in module dploi_fabric.supervisor), 19

U

update() (in module dploi_fabric.git), 9
update() (in module dploi_fabric.supervisor), 19
update() (in module dploi_fabric.virtualenv), 23
update_config_file() (in module dploi_fabric.nginx), 13
update_config_file() (in module dploi_fabric.supervisor), 19
upload_media() (in module dploi_fabric.utils), 21
upload_ssh_deploy_key() (in module dploi_fabric.github), 11
upload_ssl() (in module dploi_fabric.project), 15