
niteoweb.fabfile Documentation

Release 1.0

NiteoWeb Ltd.

November 15, 2011

CONTENTS

This package contains a selection of Fabric command we at NiteoWeb use all the time. By sharing them online we hope to save someone some time researching how certain Plone-oriented tasks are performed with Fabric. Commands contain some hardcoded internal stuff, so they are not really usable out-of-the-box, more so as a point of reference. If there is interest, we'll rewrite them to be more reusable.

- [Source code @ GitHub](#)
- [Releases @ PyPI](#)
- [Sphinx docs @ ReadTheDocs](#)

TABLE OF CONTENTS

1.1 Headquarters (HQ) server

Nothing here yet ...

1.2 Projects server

A Projects server is a server that runs your Plone projects. This is a prerequisite to have before you can run any commands from the *Project* group of commands.

Below is a `fabfile.py.in` buildout template that uses commands from *Server* group to set up a Projects server (based on Ubuntu 10.04).

```
import os
from fabric.api import env
from niteoweb.fabfile.server import *

env.path = os.getcwd()
env.hosts = ['${ips:server}']
env.hostname = '${config:hostname}'
env.shortname = '${config:shortname}'
env.temp_root_pass = '${pass:temp_root}'
env.server_ip = '${ips:server}'
env.hq_ip = '${ips:hq}'
env.bacula_ip = '${ips:bacula}'
env.officel_ip = '${ips:officel}'
env.office2_ip = '${ips:office2}'

env.email = 'maintenance@company.com'
env.admins = ['bob', 'jane', ]

env.rules = [
    # allow SSH access from our offices
    'ufw allow from %(officel_ip)s to any port ssh' % env,
    'ufw allow from %(office2_ip)s to any port ssh' % env,

    # allow access to Bacula File Deamon from our backup server
    'ufw allow from %(bacula_ip)s to any port bacula-fd' % env,

    # allow access to Munin from HQ server
    'ufw allow from %(hq_ip)s to any port munin' % env,

    # allow HTTP from everywhere
    ufw allow http
    ufw allow https
]
```

```
def deploy():
    """The highest-level meta-command for deploying Projects
    server. Use this command only on a vanilla Ubuntu 10.04 server."""

    with settings(user='root', password=env.temp_root_pass):
        create_admin_accounts(default_password='secret123')

    create_projects_group()

    # security
    harden_sshd()
    install_ufw()
    disable_root_login()

    # bootstrap server
    set_hostname()
    set_system_time()
    install_unattended_upgrades()
    raid_monitoring()
    install_rkhunter()

    # install software stack
    install_system_libs()
    install_nginx()
    install_sendmail()

    # install python
    install_python_26()
    install_python_24()
    configure_egg_cache()

    # monitoring, backup, etc.
    install_munin_node()
    install_bacula_client()
    configure_hetzner_backup()
```

This `fabfile.py` template has a dependency on a specific version of *niteoweb.fabfile* and also expects to find certain buildout values and config files in certain directories. Here's a sample `buildout.cfg` that you can use to prepare an environment for using this `fabfile.py.in`. Save the `fabfile.py.in` in `etc/` directory in your buildout.

```
[buildout]
unzip = true
newest = false
extensions = buildout.dumppickedversions
versions = versions
prefer-final = true

parts =
    fabric
    fabfile
    bacula-fd-conf
    bacula-master-conf
    duplicity-sh

# Configuration constants
[config]
# domain on which this server runs
hostname = zulu.company.com

# server's name
shortname = zulu
```



```
# Ports of services running on this server
# (besides Nginx running on port 80 and 443)
[ports]
ssh = 22
munin = 4949
bacula = 9102

# Various IPs needed for deployment
[ips]
server = ?.?.?.?
hq = ?.?.?.?
bacula = ?.?.?.?
office1 = ?.?.?.?
office2 = ?.?.?.?

# Passwords
[pass]
bacula = strong_password_here
duplicity = strong_password_here
hetzner_ftp_user = whatever_hetzner_gives_you
hetzner_ftp_pass = whatever_hetzner_gives_you
temp_root = root_password_that_hetzner_gives_you_for_a_new_server
# temp_root password is changed and disabled later on in deployment

# Pin versions of niteoweb.fabfile and other dependencies
[versions]

# Prepare Fabric
[fabfile]
recipe = collective.recipe.template
input = ${buildout:directory}/etc/fabfile.py.in
output = ${buildout:directory}/fabfile.py

[fabric]
recipe = zc.recipe.egg
eggs =
    Fabric
    niteoweb.fabfile

# Generate config files from templates ./etc
[bacula-fd-conf]
recipe = collective.recipe.template
input = ${buildout:directory}/etc/bacula-fd.conf.in
output = ${buildout:directory}/etc/bacula-fd.conf

[bacula-master-conf]
recipe = collective.recipe.template
input = ${buildout:directory}/etc/bacula-master.conf.in
output = ${buildout:directory}/etc/bacula-master.conf

[duplicity-sh]
recipe = collective.recipe.template
input = ${buildout:directory}/etc/duplicity.sh.in
output = ${buildout:directory}/etc/duplicity.sh
```

Samples of config files that you need to put inside etc/ directory in your buildout:

- bacula-fd.conf.in.
- bacula-master.conf.in.
- duplicity.sh.in.
- duplicityfilelist.conf.

- `nginx.conf`.

1.3 Bacula server

Nothing here yet ...

1.4 Available commands and how to use them

Commands are separated into two groups:

- **Project:** commands for setting up and managing Plone/Python projects
- **Server:** commands for setting up and managing servers

1.4.1 Project commands

`niteoweb.fabfile.project.configure_nginx` (*shortname=None*)

Upload Nginx configuration for this site to `/etc/nginx/sites-available` and enable it so it gets included in the main `nginx.conf`.

`niteoweb.fabfile.project.download_code` (*shortname=None*, *prod_user=None*,
svn_params=None, *svn_url=None*,
svn_repo=None, *svn_dir=None*)

Pull project code from code repository.

`niteoweb.fabfile.project.download_data` ()

Download Zope's Data.fs from the server.

`niteoweb.fabfile.project.enable_nginx_config` (*shortname=None*)

Make a link from `sites-available/` to `sites-enabled/` and reload Nginx.

`niteoweb.fabfile.project.prepare_buildout` (*prod_user=None*, *python_version=None*,
production_cfg=None)

Prepare `zc.buildout` environment so we can use `bin/buildout -c production.cfg` to build a production environment.

`niteoweb.fabfile.project.run_buildout` (*prod_user=None*, *production_cfg=None*)

Run `bin/buildout -c production.cfg` in production user's home folder on the production server.

`niteoweb.fabfile.project.start_supervisord` (*prod_user=None*)

Start *supervisord* process monitor which in turn starts Zope and optionally others (Varnish, HAProxy, etc.).

`niteoweb.fabfile.project.supervisorctl` (**cmd*)

Runs an arbitrary *supervisorctl* command.

`niteoweb.fabfile.project.upload_blobs` (*prod_user=None*, *path=None*)

Upload BLOB part of Zope's data to the server.

`niteoweb.fabfile.project.upload_data` (*prod_user=None*)

Upload Zope's data to the server.

`niteoweb.fabfile.project.upload_nginx_config` (*shortname=None*, *nginx_conf=None*)

Upload Nginx configuration to `/etc/nginx/sites-available`.

`niteoweb.fabfile.project.upload_sphinx` (*hq_ip=None*, *sphinx_dir=None*, *path=None*)

Uploads HTML files generated by Sphinx.

`niteoweb.fabfile.project.upload_zodb` (*prod_user=None*, *path=None*)

Upload ZODB part of Zope's data to the server.

1.4.2 Server commands

TODO

- evangelize to others and teach them how to use these commands
- make commands less NiteoWeb specific and more generalized

CREDITS

- Initial release by Nejc Zupan, NiteoWeb Ltd.

CHANGELOG

4.1 2.1 (unreleased)

- Lots of minor bugfixes. [zupo]
- Added *gitk* to list of libraries to install. [zupo]
- Added *buildout.cfg* to test how sphinx docs are generated. [zupo]
- Enabled choosing filename for ‘production’ buildout configuration. [zupo]
- The *configure_egg_cache()* command is now more resilient to multiple runs. [zupo]
- Added instructions and examples on how to use *niteoweb.fabfile* for setting up a new server for running Plone projects. [zupo]
- Added commands for installing and configuring a server that will run Plone projects. [zupo]

4.2 2.0.2 (2011-11-13)

- You can now specify python version that is used for bootstrapping buildout. [zupo]

4.3 2.0.1 (2011-11-13)

- *HISTORY.txt* missing from release. [zupo]

4.4 2.0 (2011-11-13)

- Use *niteoweb.fabfile.err* instead of *_verify_opts*. [zupo]
- Breaks backwards compatibility with commands in *project.py* [zupo]

4.5 0.1.2 (2011-10-21)

- Added many new commands for setting up servers. [zupo]

4.6 0.1.1 (2011-08-28)

- Packaging fixes. [zupo]

4.7 0.1 (2011-08-28)

- Initial release. [zupo]

LICENSE

Copyright (c) 2011, NiteoWeb Ltd. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of NiteoWeb Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL NITEOWEB LTD BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

n

niteoweb.fabfile.project,??