

---

# Deployment Documentation

*Release 2.0.14*

**TEONITE**

December 06, 2016



<b>1</b>	<b>Usage</b>	<b>3</b>
1.1	Default directories for configuration file . . . . .	3
1.2	List of available commands . . . . .	3
1.3	Requirements . . . . .	5
<b>2</b>	<b>Deployment flow</b>	<b>7</b>
2.1	Folder structure . . . . .	7
2.2	Deployment process . . . . .	7
<b>3</b>	<b>Configuration description</b>	<b>9</b>
3.1	Sample configuration file . . . . .	9
3.2	Configuration description . . . . .	11
<b>4</b>	<b>Indices and tables</b>	<b>15</b>



Contents:



---

## Usage

---

### 1.1 Default directories for configuration file

```
./src/conf/deploy.json
./src/conf/deployment.json
./src/settings/deploy.json
./src/settings/deployment.json
./conf/deploy.json
./conf/deployment.json
./settings/deploy.json
./settings/deployment.json
./src/conf/deployment/production.json
./src/settings/deployment/production.json
./conf/deployment/production.json
./settings/deployment/production.json
```

### 1.2 List of available commands

- **db\_migrate:**
  - Migrate database to new version
  - *Arguments:* <migration\_folder>
- **deploy:**
  - Deploys new version
  - *Arguments:* <local\_subfolder>
- **mysql\_db\_clone:**
  - Clone db: <db\_name> -> <db\_name>\_<current\_date>\_<current\_time>
  - *Arguments:* <db\_name>
- **mysql\_db\_dump:**
  - Dump database to selected file
  - *Arguments:* <database>
- **mysql\_db\_migrate:**
  - Runs .sql files from selected folder

- *Arguments:* <migration\_folder>
- **mysql\_db\_restore:**
  - Restore database from file
  - *Arguments:* <database>
- **mysql\_dump\_remove:**
  - Remove dump file from remote host
  - *Arguments:* None
- **restart\_supervisor:**
  - Restart process in Supervisor
  - *Arguments:* <process\_name> <process\_name> ...
- **src\_clean:**
  - Delete file/folder provided as argument
  - *Arguments:* <to\_delete>
- **src\_clone:**
  - Clone repo to subfolder in local folder
  - *Arguments:* <folder> <subfolder>
- **src\_post\_deploy:**
  - Runs post\_deploy command from config file
  - *Arguments:* None
- **src\_pre\_deploy:**
  - Runs pre\_deploy command from config file
  - *Arguments:* None
- **src\_prepare:**
  - Archive repo from subfolder in local folder to file
  - *Arguments:* <folder> <subfolder>
- **src\_remote\_config:**
  - copy config from previous to current
  - *Arguments:* None
- **src\_remote\_deploy:**
  - deploys new version
  - *Arguments:* <subfolder>
- **src\_remote\_extract:**
  - extract uploaded file to selected subfolder of deploy\_dir, default current date
  - *Arguments:* <subfolder>
- **src\_remote\_rollback:**
  - backs to previous version



- *Arguments:* None
- **src\_remote\_test:**
  - test remote host
  - *Arguments:* <user> <host>
- **src\_remote\_venv:**
  - check if venv exists, if not it will be created, and populated with requirements.txt
  - *Arguments:* <directory>
- **src\_upload:**
  - upload packed file from local folder to remote host
  - *Arguments:* none

## 1.3 Requirements

Deployment requires the following modules:

- Python 2.7+ (3+ untested)
- Fabric 1.5.1+ (<http://fabfile.org/>)
- GitPython 0.3.2.RC1+ (<http://pythonhosted.org/GitPython/0.3.2/index.html>)



---

## Deployment flow

---

### 2.1 Folder structure

```
remote_dir
|-deploy_dir
|   |-20130411_212959
|   |-20130411_213213
|   |-20130412_181014
|   |-current -> 20130412_181014
|   |-previous -> 20130411_213213
|-migration_dir
|-upload_dir
|-dump_file
```

### 2.2 Deployment process

1. **Test remote server** Checks if connection to remote server is possible.
2. Execute pre-deploy commands
3. **Get sources** If local git repository is found, it will be pulled, else will be cloned.
4. Archive sources
5. Upload sources to remote server
6. Extract source to target subfolder
7. Switch current/previous links
8. Copy config files
9. Run post-deploy commands



---

## Configuration description

---

### 3.1 Sample configuration file

The config is in JSON format:

```
{
  "source": {
    "git": {
      "repo": "git@git.teonite.net:teonite/status.git",
      "branch": "master",
      "local": "repo",
      "dirs" : ["src"]
    },
    "local": "test",
    "file": "arch.tar.gz"
  },
  "remote": {
    "host": "192.168.56.101",
    "user": "kkrzysztofik",
    "port": 22,
    "dir": "~",
    "clean": true
  },
  "deploy": {
    "dir": "Status",
    "pre": [
      "cat ~/.bashrc",
      "env"
    ],
    "post": [
      "pip install -r ~/Status/current/src/requirements.txt",
      "python ~/Status/current/src/tools/manage.py collectstatic --noinput"
    ],
    "changelog": "CHANGELOG.rst"
  },
  "config": [
    "src/settings/gunicorn.conf.py",
    "src/settings/supervisor.conf",
    "src/settings/localsettings.py",
    "src/settings/logger.py"
  ],
}
```

```
"mysql" : {
  "server" : {
    "host" : "localhost",
    "user" : "root",
    "password" : "pass",
    "port" : 3306,
    "database" : "status"
  },
  "shell" : {
    "dumpfile" : "temp.sql",
    "user" : "kkrzysztofik",
    "host" : "192.168.56.101",
    "port" : 22,
    "migration_dir" : "test"
  }
},

"supervisor" : {
  "host" : "192.168.56.101",
  "port" : 9001,
  "user" : "admin",
  "password" : "pass",
  "apps" : [
    "status",
    "status_celery"
  ]
},

"venv": {
  "requirements": ["requirements/production.txt"],
  "update": false,
  "dir": "env"
},

"mail": {
  "project_name": "Status",
  "from_mail": "TEONITE <notification@teonite.com>",
  "server": {
    "user": "notification@teonite.com",
    "password": "notification_sample_password",
    "host": "smtp.gmail.com",
    "port": 587
  },
  "people": [
    "kkrzysztofik@teonite.com"
  ],
  "template_path": "deployment/template.py"
},

"logger" : {
  "version": 1,
  "formatters": {
    "simple": {
      "format": "%(message)s"
    },
    "verbose": {
      "format": "[% (asctime)s] \"%(message)s\"",
      "datefmt": "%d/%b/%Y %H:%M:%S"
    }
  }
}
```

```

    }
  },
  "handlers": {
    "console": {
      "level": "DEBUG",
      "class": "logging.StreamHandler",
      "formatter": "simple",
      "stream" : "ext://sys.stdout"
    }
  },
  "loggers": {
    "root": {
      "handlers": ["console"],
      "level": "INFO"
    },
    "deployment": {
      "handlers": ["console"],
      "level": "INFO",
      "qualname": "deployment",
      "propagate": false
    }
  }
}

```

## 3.2 Configuration description

### 3.2.1 source

- *local* - Directory where all local files are placed (GIT repository, archive file)  
*Default:* current working directory
- *file* - Filename of file used to deploy on remote host, currently supported extensions are .tar.gz, .tgz, .tar  
*Default:* src.tar.gz

### git

Section connected with GIT repository

- *repo* Repository used to clone source.  
*Default:* fetched from current working directory
- *branch* Branch used to clone source  
*Default:* master
- *local* Local repository directory, placed inside main local directory  
*Default:* current working directory
- *dirs* Directories and files which deployment archive is made of  
*Default:* whole repository

### 3.2.2 remote

During application deploy process, SSH with key-based authentication is used. You need to set host address and user. Ports default value is 22.

- **dir** Directory where archive is uploaded during deployment  
*Default:* current working directory
- **clean** Flag used to specify if archive after deployment has to be removed

### 3.2.3 deploy

- **dir** Directory where application lives, for rest of commands in this section is base directory  
*Default:* app
- **pre** List of commands launched before deployment
- **post** List of commands launched after deployment
- **changelog** File, which will be overwritten during deployment with changelog based on all commits messages

### 3.2.4 config

In this section you can configure list of files that should be copied after deployment. Using new format, paths are relative to `deploy_dir\previous` and are copied to `deploy_dir\current`:

```
new format:
  "config": [
    "path",
    "path2"
  ]

old format: (deprecated):
  "config": {
    "display name": {
      "src": "absolute path to source file",
      "dst": "absolute path to destination"
    }
  }
```

### 3.2.5 venv

In this section, parameters of virtual environment are set.

- **dir** Directory where virtualenv should be located. If not defined, no check is made.
- **requirements** List of requirements files, that are installed after create/check of virtual env  
*Default:* ["requirements/production.txt"]
- **update** Update packages during check of virtual environment and requirements  
*Default:* false



### 3.2.6 mysql

#### shell

Shell from which all MySQL commands are executed, standard requirements are:

- *user*
- *host*
- *port*

Extra:

- ***dumpfile*** File used to make dumps of database and as a temporary file
- ***migration\_dir*** Into this dir will be uploaded `.sql` files used in migration process

#### server

MySQL server configuration used in all commands. Server must be accessible from shell, mentioned before

Requirements:

- *host*
- *user*
- *password*
- *port*
- *database*

### 3.2.7 supervisor

Supervisor is a client/server system that allows its users to monitor and control a number of processes on UNIX-like operating systems.

Requirements:

- *host*
- *port*
- *user*
- *password*
- ***apps*** Supervisor processes which will be restarted

### 3.2.8 mail

Section used by `notify` command to send mails with notification about deployment.

Required:

- *project\_name*
- *from\_mail* - mail displayed in From: field
- *people* - list of emails to notify

- *user*
- *password\**

Optional:

- *host* - default: localhost
- *port* - default: 25
- *template\_path* - custom mail template. Must be .py file and contain two variables SUBJECT\_TEMPLATE and MESSAGE\_TEMPLATE. Available variables: {project\_name}

### 3.2.9 logger

The logger is configured by using `logging.config.dictConfig()` function, format is described [here](#)

*Default:* same as listed above

---

## Indices and tables

---

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