

BangPyppers April Meetup

Fabric and App Deployment

Deepak Garg

Citrix R&D



Openstack India Meetup by Deepak Garg is licensed under a
Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License
Based on a work at github.com.

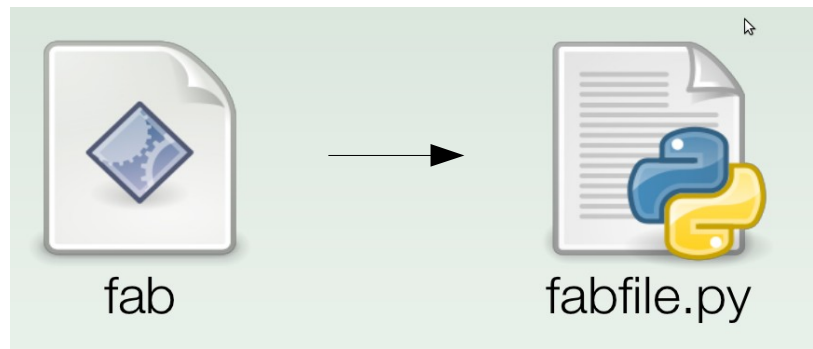
Contents

- Fabric ?
- Example Code
- App Deployment
- Fab cmds
- Fab decorators
- Fabric for Django App Deployment
- Why Fabric ?
- Issues and Alternatives
- Q & A

Fabric ?

Automated deployments for lazy people

- A command **fab**
- and a py file **fabfile.py**



Get it: <http://fabfile.org>
\$ sudo pip install fabric

Example Code

```
def clean(site=None):  
    """Remove .pyc files from a site."""  
    from os import path  
    if not site:  
        site = prompt('Please specify which site(s) to clean (a  
comma delimited list): ', validate=r'^[\w-.,]+$')  
    site_list = site.split(',')  
    for item in site_list:  
        env.hosts = [ item ]  
        path = path.join('/var/www', item)  
        sudo("find %s -name '*.pyc' -depth -exec rm {} \;" % path)
```

```
$ fab clean:site=xenstack.org
```

```
[172.16.12.6] sudo: find /var/www/xenstack.com -name  
 '*.pyc' -depth -exec rm {} \;
```

```
Password for testuser@192.168.56.3:
```

```
Done.
```

```
Disconnecting from 172.16.12.6... done.
```

```
$
```

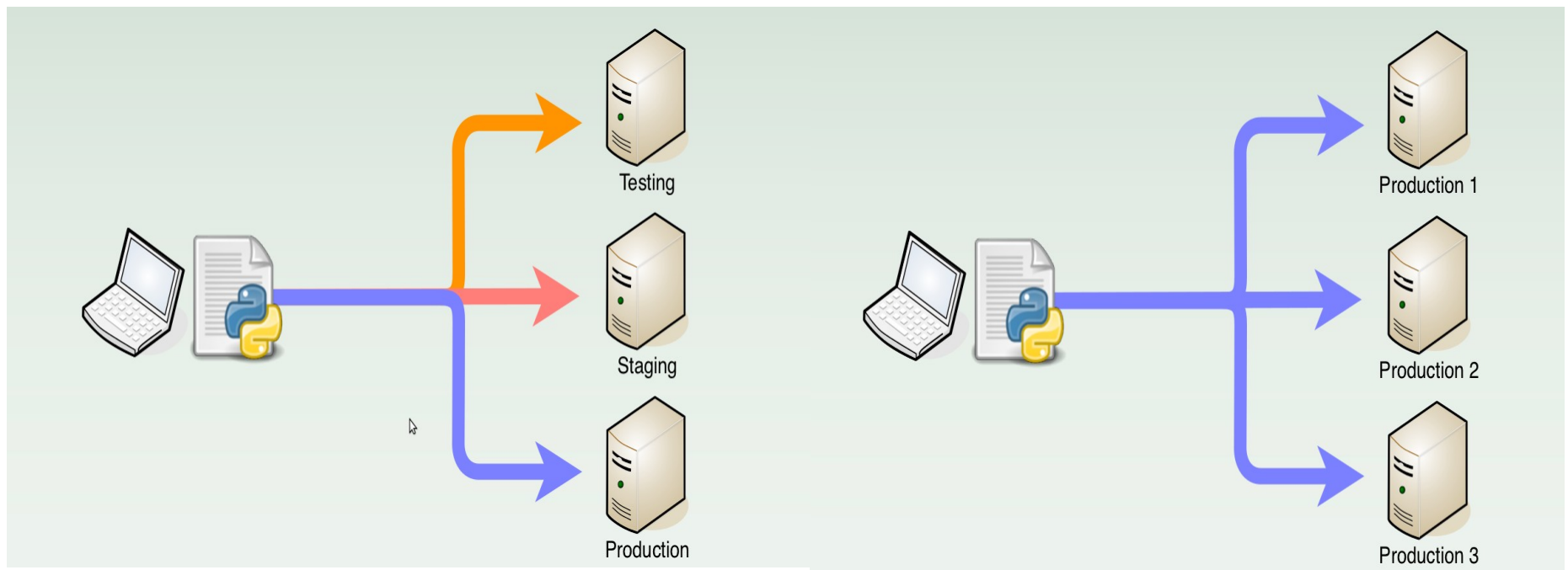
App Deployment



- **Order Server:** Provision VMs
- **Setup Server:** Create users, groups, Customize config files, Install base packages
- **Deploy App:** install app specific packages, deploy application, start services

App Deployment

- Deploy on multiple servers – scale out
- Deploy on multiple environments
- Make this process AUTOMATED, Simple and fast



Sample snippet 1

```
def provision_vm():  
    # Use boto to provision server in AWS  
    .....
```

```
def install_baseline():  
    local("git add -p && git commit")  
    .....
```

```
def install_app_specific_packages():  
    .....
```

```
def install_db(db_name=None):  
    .....
```

```
with_settings(warn_only=True)  
def safe_merge():  
    .....
```

```
@roles('production_web')  
def production_server():  
    provision_vm()  
    install_baseline()  
    install_app_specific_packages()  
    install_db(db_name='mysql')  
    configure_production()
```

```
@roles('testvm')  
def testenv_server():  
    install_baseline()  
    install_app_specific_packages()  
    install_db(db_name='mysql')  
    configure_test()
```

\$ fab testenv_server #Local Env

\$ fab production_server # Prodn Env

Sample snippet 2

```
def basic_tests():  
    local("./manage.py test my_app")  
    .....
```

```
def commit():  
    local("git add -p && git commit")  
    .....
```

```
def local_deploy():  
    basic_tests()  
    Commit()
```

```
$ fab local_deploy #Local Env
```

```
def grab_code():  
    # git fetch  
    .....
```

```
$ fab stage_deploy # Staging Env
```

```
$ fab production_deploy # Prodn Env
```

```
with_settings(warn_only=True)  
def safe_merge():  
    # git merge  
    .....
```

```
@hosts('user1@host1', 'host2', 'user3@host3')
```

```
def stage_deploy():  
    grab_code()  
    basic_tests()  
    stage_tests()  
    safe_merge()
```


Why Fabric ?

- **Testing:** Deploy some vms
execute some api/cmds
collect the results, logs back, analyse etc...
- **Deploy and Scale your App:** All the servers should have exactly the same configurations and same code
- **Systems Administration**
 - Clean .pyc files
 - Check out new revision from a SCM repository
 - Install packages, Apply patches
 - Available fabric recipes !!
- **Crux: Interact with Systems !
Make admin and deployment tasks Pythonic !!**

Fab Cmds

- prompt: Display a prompt to the user and store the input in the given variable. Validation is optional.
- put: Upload files to the remote host(s).
- get: Download a file from a remote host.
- run: Run a shell command on the remote host(s).
- sudo: Run a root privileged shell command on the remote host(s).
- local: Run a command locally.
- upload_template: Render and upload a template text file to a remote host.
- rsync_project: Synchronize a remote directory with the current project directory via rsync
- append: Append string (or list of strings) text to filename.

Fab Decorators

- `@hosts`: Defines on which host or hosts to execute the wrapped function.
- `@parallel`: Forces the wrapped function to run in parallel
- `@with_settings`: Forces fab settings on that task
e.g: `@with_settings(warn_only=True)` -> turn aborts into warnings
- `@runs_once`: Prevents wrapped function from running more than once.
- `@serial`, `@roles` etc...

```
@hosts('user1@host1', 'host2', 'user2@host3')
@runs_once
def my_func():
    pass
```

```
@roles('web')
def my_other_func():
    pass
```

Lets look at a sample fabfile

Issues and Alternatives

- Roll back not easy
- State Management
- Alternatives: Puppet, Chef
 - Not easy to write
 - Need to maintain Master and Slave nodes

THANK YOU !

Presentation at:

<https://github.com/gargdeepak/Presentations>

Happy to help !

deepakgarg.iitg[at]gmail.com

@donji