

# DESPLEGANDO DJANGO

roberto.majadas  openshine

PyconES 2016  
Almería

PENSANDO UN POCO  
ANTES DE DESPLEGAR

# ¿QUÉ ES LO QUE VAS A DÉSPLEGAR?

Nunca es sólo django

- Python/Django
  - Base de datos SQL
    - Postgresql
    - Mysql
  - Servidor web
    - Apache
    - Nginx
  - Redis
  - Storage en Cloud
  - Celery
  - Big Data
  - EdgeTech <current year>
-

# ¿A QUIEN SE LO VAS A DESPLEGAR?

Cliente diferente,  
necesidad diferente



# ¿DONDE LO VAS A DESPLEGAR?

Muchas decisiones dependen  
del “donde”

- ¿CPD propio?
- ¿En algún Cloud?
- ¿En algún PASS?
- ¿Múltiples entornos?
- ¿Acceso remoto?
- ¿Integración continua?

# CONCLUSIONES PREVIAS

¿CUAL ES LA MEJOR  
MANERA DE  
DESPLEGAR?

NI P\*\*\*

IDEA

¡ASÍ QUE  
EMPECEMOS POR EL  
PRINCIPIO!

# DJANGO

El minimun minimorum

- Python
    - Versión 3, mejor que 2
    - Virtualenv
  - Base de datos
    - Postgresql
    - Mysql
  - Servidor web
    - Apache
    - Nginx
  - WSGI
    - apache-wsgi
    - uwsgi
    - gunicorn
-

# DJANGO

Cosas a tener en cuenta

- Los settings
    - Varios entornos
    - Variables de entorno
      - `django-environ`
  - Virtualenv
    - `requirements.txt`
  - Piensa dónde pones los datos (`static/media`)
  - Entorno aislado mejor
-

# BASES DE DATOS

Cosas a tener en cuenta

- Datos de conexión vía DATABASE\_URL
- ¿Quién gestiona la base de datos?

# WEBSERVER

Cosas a tener en cuenta

- ¿Quién controla el webservice?
- ¿Que WSGI uso?

MODOS DE  
DESPLIEGUE DE  
MENOS A MÁS  
(MÁS O MENOS)

CAPÍTULO I:  
LAS GUERRAS SCRIPT

# GREAT BALLS OF FIRE!

¿Desplegar yo?



# YO, YO Y TAMBIÉN YO

presentado por yo mismo

- Yo hago un tarball del código
- Yo lo subo al servidor
- Yo despliego a mano
- Yo reinicio lo que se tenga que reiniciar

---

# EL SCRIPT

En su momento pareció una  
buena idea

- Script en bash
  - Normalmente se ejecuta en el servidor
  - Se empieza a complicar cuando:
    - Tienes varios entornos
    - Tienes que ejecutar en remoto
    - Tienes que hacer cosas más complicadas que el lenguaje script no puede con ello.
-

# SCRIPT + SSH

Python Fabric

```
from fabric.api import run
```

```
def host_type():
```

```
    run('uname -s')
```

```
$ fab -H linuxbox host_type
```

```
[localhost] run: uname -s
```

---

CAPÍTULO II:  
OLD SCHOOL BAD ASS

# RPM/DEB

Usando los paquetes de las  
distros



# FPM

Effing package management

- Comando en ruby
  - Permite crear un paquete en una línea de comandos
  - Soporta
    - DEB
    - RPM
  - Scripts pre/post install
-

# OMNIBUS

Gotta Catch 'Em All

- Puedes crear un paquete con TODO lo necesario
- Wocu y Gitlab usa omnibus

CAPÍTULO III:  
DOING BETTER

# CAPISTRANO

Si, está hecho en ruby

```
role :demo, %w{example.com example.org example.net}
task :uptime do
  on roles(:demo), in: :parallel do |host|
    uptime = capture(:uptime)
    puts "#{host.hostname} reports: #{uptime}"
  end
end
```

# CAPISTRANO

Layout de capistrano

```
|— current -> /var/www/my_app_name/releases/20150120114500/
|— releases
|   |— 20150080072500
|   |— 20150090083000
|   |— 20150100093500
|   |— 20150110104000
|   |— 20150120114500
|— repo
|   |— <VCS related data>
|— revisions.log
|— shared
|   |— <linked_files and linked_dirs>
```

# ANSISTRANO

Ansible + Capistrano

```
-- /var/www/my-app.com
|-- current -> /var/www/my-app.com/releases/20100512131539
|-- releases
|   |-- 20100512131539
|   |-- 20100509150741
|   |-- 20100509145325
|-- shared
```

- vars:

```
ansistrano_deploy_from: "{{ playbook_dir }}" # Where my local project is (relative or absolute path)
ansistrano_deploy_to: "/var/www/my-app" # Base path to deploy to.
ansistrano_version_dir: "releases" # Releases folder name
ansistrano_current_dir: "current" # Softlink name. You should rarely changed it.
ansistrano_current_via: "symlink" # Deployment strategy who code should be deployed to current path. Opti
ansistrano_shared_paths: [] # Shared paths to symlink to release dir
ansistrano_shared_files: [] # Shared files to symlink to release dir
ansistrano_keep_releases: 0 # Releases to keep after a new deployment. See "Pruning old releases".
ansistrano_deploy_via: "rsync" # Method used to deliver the code to the server. Options are copy, rsync,
ansistrano_allow_anonymous_stats: yes
```

CAPÍTULO IV:  
LOS CABALLEROS DE LA  
GESTIÓN DE LA  
CONFIGURACIÓN

# ANSIBLE

Pequeño pero matón

```
- name: Clone/pull project repo
git: repo={{ project_repo }}
      dest={{ install_root }}/{{ project_name }}
      accept_hostkey=yes
notify:
- restart gunicorn
```

```
- name: install python packages
pip: requirements={{ install_root }}/{{ project_name
}}/requirements.txt
notify:
- restart gunicorn
```

---

# SALT/CHEF/PUPPET

Un gran Mjolnir, para una gran batalla

- Sistemas de gestión de configuraciones
- Mayormente centralizados
- Capaces de orquestar y desplegar muchas cosas a la vez.

CAPÍTULO V:  
LA REVOLUCIÓN DE  
LOS CONTENEDORES

# DOCKER

Containers, containers  
everywhere

- Usar y tirar
- Microservicios
- Composición
- Escalado dinámico

# DOCKER

Containers, containers  
everywhere

```
version: '2'
services:
  db:
    image: postgres
  web:
    build: .
    command: python manage.py runserver 0.0.0.0:8000
    volumes:
      - ./code
    ports:
      - "8000:8000"
    depends_on:
      - db
```

# CAPÍTULO VI: CLOUD ALTAS

# AWS Y DIGIEVOLUCIONES

Será por nubes!

- Tienen sus cosas buenas y sus cosas malas
- Se puede ajustar a muchos fines de despliegue diferentes
- Fácil de scriptear/automatizar

# PASS

Platform as a Service

- ¿Que es un PASS?
- ¿En la nube pública o privada?
  - Heroku vs OpenShift
- Muy bonito! dónde están las pegas...

CAPÍTULO VII:  
MATRIX Y  
EL ARQUITECTO DE  
CUENCA

# CASO NORMAL

django+bbdd+nginx

- Es un rollo, lo sé, pero haz mínimo dos entornos.
- fabric
- (capis|ansis)trano
- ansible
- Integración continua

---

# CASO MEDIO

Más de un server haciendo  
cosas

- Usar gestión de configuración puede ser muy útil
- ¿Alta disponibilidad?

---

# CASO "OH MY GOSH"

Quando django es lo de menos

- Django es API en el backend de algo más grande (BIGDATA)
- Producto de empresa: Gitlab
- Servicios de escalado horizontal

¿VEMOS ALGO EN  
PARTICULAR?

¿PREGUNTAS?

GRACIAS!!

