

Hoy estoy muy

ANSIBLE

...

isholgueras@letshackity.com

about.me

- @isholgueras
- drupal.org/u/isholgueras
- nacho@letshackity.com
- <http://www.isholgueras.com>



- CTO @ Hackity
- No, no soy DevOps, y tampoco llevo gafas

Índice

1. Resumen de la charla
2. ¿Cómo está el patio?
3. Ansible. Primeros pasos
4. Ansible. Organización de playbooks
5. Ansible + Vagrant. Producción en local
6. ¡Yo quiero! Pero, ¿por dónde empiezo?
7. ¿Dónde queda Docker? Futuro

1. Resumen de la charla

¡ Por el poder de los GIF !



Sysadmins



Un ojo en todo

- Terminator
- Logs
- Munin
- ¿web3 caído?



Mi objetivo



Porque Ansible es esto



```
ansible-galaxy install sta-claus.christmas-tree
```

2. ¿Cómo está el patio?

Los que juegan



ANSIBLE



¿Por qué Ansible?

```
class munin::master (
  $ensure      = present,
  $autoupgrade = true
) inherits munin::master::params {

  case $ensure {
    /(present)/: {
      $dir_ensure = 'directory'
      if $autoupgrade == true {
        $package_ensure = 'latest'
      } else {
        $package_ensure = 'present'
      }
    }
    /(absent)/: {
      $package_ensure = 'absent'
      $dir_ensure = 'absent'
    }
    default: {
      fail('ensure parameter must be present or absent')
    }
  }
}
```

```
include munin::master::install
include munin::master::config
```



```
---
# Useful for debugging.
- name: Cleanup previous attempt to bootstrap
  file: path=~/.my.cnf state=absent

# Start the service so we can perform bootstrap operations.
- name: Start mysql for bootstrap
  service: name={{ mysql_service }} state=started

# Note: Change password for root@localhost last, to not loose access to mysql.
- name: Set root password
  mysql_user: name=root host={{ item }} password={{ mysql_root_pass }}
  with_items:
    - "{{ ansible_hostname }}"
    - 127.0.0.1
    - ::1
    - localhost
  when: mysql_root_pass is defined

- name: Copy .my.cnf file with root password credentials
  template: src=.my.cnf.j2 dest=~/.my.cnf mode=0600
  when: mysql_root_pass is defined
```



<https://github.com/sbitio/ansible-mysql>

<https://github.com/sbitio/puppet-munin>

3. Ansible. Primeros pasos

¡Hola Ansible!



- Creador: Michael DeHaan (@laserllama)
- Pretende ser:
 - **Claro:** YAML.
 - **Rápido:** curva de aprendizaje y configuración.
 - **Completo:** Sin límites
 - **Eficiente:** Sin software extra (sólo ssh).
 - **Seguro:** SSH
- Documentación muy completa.
- Soporte oficial Ansible Inc.
- Consultoría y formación oficial.
- Comunidad (ansible-galaxy, FreeNode #ansible,...)



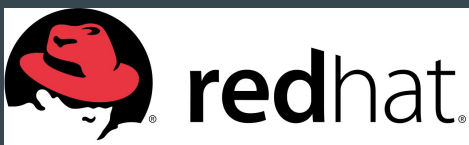
Instalación



```
pip install ansible
```



```
http://brew.sh/  
sudo brew install python  
pip install ansible
```



```
yum -i install ansible
```



```
Vagrant up (linux)  
cygwin*****
```

Configuración básica

SSH

```
mkdir ~/.ssh  
chmod 700 ~/.ssh  
ssh-keygen -t rsa  
ssh-copy-id -i ~/.ssh/id_rsa.pub user@myserver:/home/user/
```

Hosts de Ansible

```
vim /etc/ansible/hosts  
[myserver]  
127.0.0.1
```

Hola Mundo

```
ansible myserver -a "free -m"
```


Ejemplos de uso rápido

Uptime completo

```
ansible N.pro.all -a "uptime"
```

Reinicio de nginx

```
ansible webN.pro.es -a "/etc/init.d/nginx restart"
```

Ping a todo italia

```
ansible N.pro.it -a "ping"
```

“DF” a las bases de datos

```
ansible bdN.pro.all -a "df"
```

`/etc/ansible/hosts`

```
# Spain
[webN.pro.es]
web1.pro.es
web2.pro.es
web3.pro.es

[bdN.pro.es]
bd1.pro.es

[varn1.pro.es]
10.10.1.31

# Italy
# ...

# Portugal
# ...

# Spain group
[N.pro.es:children]
webN.pro.es
bdN.pro.es
varnN.pro.es

# All web-servers group
[webN.pro.all:children]
webN.pro.es
webN.pro.it
webN.pro.pt

# Variables applied to
[webN.pro.all:vars]
ansible_ssh_user=flynn
```

Ejemplos de uso rápido

¿Reinicio a Full?

```
ansible N.pro.all -a "reboot"
```



Parámetros útiles

Limitar servidores

```
ansible webN.pro.es -a "free -m" --limit web1.pro.es
```

Usuario concreto

```
ansible webN.pro.es -a "free -m" --remote-user=<user>
```

Sudo

```
ansible webN.pro.es -a "free -m" --sudo --sudo-user=<user> --ask-sudo-pass
```

Otros

```
ansible webN.pro.es -a "free -m" --list-host  
ansible webN.pro.es -a "free -m" --verbose  
ansible webN.pro.es -a "free -m" --forks=<num>  
ansible webN.pro.es -m <module> -a "<arguments>" --check
```

Parámetros útiles

Módulos de ansible

```
$ ansible webN.pro.es -m ping
```

```
$ ansible webN.pro.es -m service -a "name=nginx state=restarted enabled=yes"  
(restarted|reloaded|present|absent)
```

```
$ ansible webN.pro.es -m apt -a "name=nginx state=present"
```

Estirando más aún

```
$ ansible webN.pro.es -a "php -r 'phpinfo();'"
```

```
$ ansible webN.pro.es -a "tail -15 /var/log/apache2/error.log" --sudo --sudo-  
user=root \  
--ask-sudo-pass
```

```
$ ansible dbN.pro.es -s -m mysql_user -a "name=flynn host=% password=Br1dg3s \  
priv= *.*:ALL state=present"
```

4. Ansible. Organización de playbooks

Playbooks

install-apache.sh

```
#!/bin/bash
sudo apt-get install apache2
sudo a2enmod rewrites
sudo cp ports.conf
/etc/apache2/ports.conf
sudo update-rc.d apache2 defaults
sudo apache2 restart
```

apache.yml

```
---
- hosts: webN.pro.all
  sudo: yes

  tasks:
    - name: Install Apache2
      apt: name=apache2 state=present

    - name: Enable rewrites
      apache2_module: name=rewrites state=present

    - name: Copy configuration
      copy:
        src: ports.conf
        dest: /etc/apache2/ports.conf
        owner: root
        group: root
        mode: 0644

    - name: Ensure apache2 is started on boot
      service: name=apache2 state=started enabled=yes
```

Anatomía de un rol

apache2-role

```
~/ansible $ ansible-galaxy init apache2-role
~/ansible $ cd apache2-role
~/ansible/apache2-role $ tree
.
├── defaults          # default lower priority variables for this role
│   └── main.yml
├── files             # files for copy and script resources
├── handlers         # handlers
│   └── main.yml
├── meta             # dependencies
│   └── main.yml
├── README.md
├── tasks            # tasks
│   └── main.yml
├── templates        # files for template resource *.j2
├── vars             # variables associated with this role
│   └── main.yml
```

Anatomía de un rol: Ficheros

```
.
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
└── vars
    └── main.yml
```

defaults/main.yml

```
---
# default variables
listen_port: 8080
root_directory: /apps
web_user: vagrant
```

files/script.sh

```
#!/bin/bash
echo "hello world!"
```


Anatomía de un rol: Ficheros

```
.
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── vars
│   └── main.yml
```

handler/main.yml

```
---
- name: check apache2 configuration
  command: apachectl configtest
  register: result

- name: restart apache2
  shell: /etc/init.d/apache2 restart
  when: result|success
```

meta/main.yml

```
---
galaxy_info:
  author: <author>
  description: <description>
  # more metadata
dependencies: []
```

Anatomía de un rol: Ficheros

```
.
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── vars
│   └── main.yml
```

vars/main.yml

```
---
apache2:
  port: 80
  virtualhost:
    - { name: web1.dev, admin: flynn@web1.dev }
    - { name: web2.dev, admin: flynn@web2.dev }
  packages:
    - apache2
    - libapache2-mod-auth-cas
```

templates/virtualhosts.j2

```
{% for virtualhost in apache2.virtualhost %}
<VirtualHost *:{% apache2.port %}>
  DocumentRoot /var/www/{% virtualhost.name %}
  ServerName {% virtualhost.name %}
  ServerAdmin {% virtualhost.admin %}
</VirtualHost>
{% endfor; %}
```

Anatomía de un rol: Ficheros

```
.
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── vars
│   └── main.yml
```

tasks/main.yml

```
---
- name: Install apache2
  apt: pkg={{ item }} state=present
  with_items: {{ apache2.packages }}

- name: Ensure apache2 is running.
  service: name=apache2 state=started enabled=yes

- name: Add custom virtualhost
  template:
    src=virtualhosts.j2
    dest=/etc/apache2/sites-available/default
    owner=root
    mode=0644
  notify:
    - check apache2 configuration
    - restart apache2
  tags: templates
```

Anatomía completa

```
— dbservers.yml
— filter_plugins
— group_vars
  |— group1
  |— group2
— host_vars
  |— hostname1
  |— hostname2
— library
— production
  |— italy
  |— spain
— roles
  |— apache2-role
  |   |— defaults
  |   |   |— main.yml
  |   |— files
  |   |— handlers
  |   |   |— main.yml
  |   |— meta
  |   |   |— main.yml
  |   |— README.md
  |   |— tasks
  |   |   |— main.yml
  |   |— templates
  |   |— vars
  |   |   |— main.yml
  |— mysql-role
— site.yml
— staging
  |— italy
  |— spain
— webservers.yml
```

- Recomendada por Ansible
- Común en proyectos

```
---
# file: webservers.yml
- hosts: webN.pro.all
  roles:
    - apache2-role
---
# file: dbservers.yml
- hosts: bdN.pro.all
  roles:
    - mysql-role
```

Ejemplos de ejecución

con -i introducimos un fichero de hosts manual

```
ansible-playbook -i production/spain webserver.yml
```

Se ejecutan sólo las tareas taggeadas con "templates"

```
ansible-playbook -i production/spain webserver.yml --tags templates
```

Limitamos la ejecución a web0.pro.es y web1.pro.es

```
ansible-playbook -i production/spain webserver.yml --limit webN.pro.es[0:1]
```

Nos aseguramos qué hosts son los que se van a cambiar

```
ansible-playbook -i production/spain webserver.yml --list-hosts
```

Como a mí me gusta

```
▼ ▢ vagrant
  ▶ ▢ .vagrant
  ▼ ▢ ansible
    ▼ ▢ roles
      ▶ ▢ composer
      ▶ ▢ comunicacion
      ▶ ▢ drush
      ▶ ▢ elasticsearch
      ▶ ▢ local-dev
      ▶ ▢ memcached
      ▶ ▢ news
      ▶ ▢ nfs
      ▶ ▢ nginx
      ▶ ▢ percona-mysql
      ▶ ▢ php
      ▶ ▢ practicas-dudosas
      ▶ ▢ statics
      ▶ ▢ varnish
      ▶ ▢ warmer
```

```
▼ ▢ vars
  ▶ ▢ local.all
  ▶ ▢ pre.all
  ▼ ▢ pro.es
    ▢ vars.yml
  ▶ ▢ pro.it
  ▶ ▢ pro.pt
  ▢ .git
  ▢ README.md
  ▢ allN.pre.all.yml
  ▢ allN.pro.es.yml
  ▢ allN.pro.it.yml
  ▢ allN.pro.pt.yml
  ▢ bdN.pre.all.yml
  ▢ bdN.pro.es.yml
  ▢ bdN.pro.it.yml
  ▢ bdN.pro.pt.yml
  ▢ hosts
  ▢ local.all.yml
  ▢ memcN.pre.all.yml
  ▢ memcN.pro.es.yml
  ▢ memcN.pro.it.yml
  ▢ memcN.pro.pt.yml
```

```
---
# file: webN.pro.es.yml
- hosts: webN.pro.es
  vars_files:
    - vars/pro.es/vars.yml
  roles:
    - php
    - statics
    - nginx
    - news
    - comunicacion
    - practicas-dudosas
```

Branch

🐞 master

🐞 php5.6

Módulos de Ansible

- Los plugins oficiales de conexión con servicios

Module Index

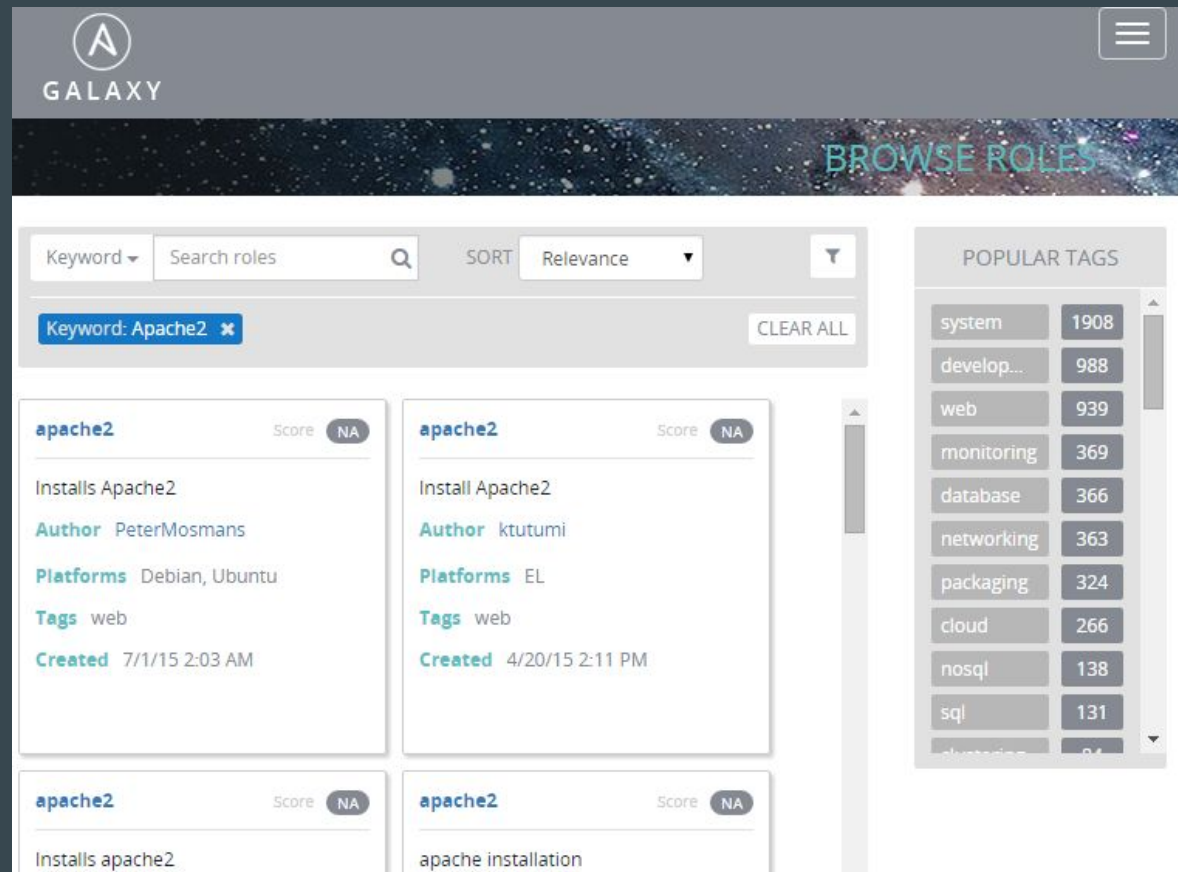
- All Modules
- Cloud Modules
- Clustering Modules
- Commands Modules
- Database Modules
- Files Modules
- Inventory Modules
- Messaging Modules
- Monitoring Modules
- Network Modules
- Notification Modules
- Packaging Modules
- Source Control Modules
- System Modules
- Test Modules
- Utilities Modules
- Web Infrastructure Modules
- Windows Modules

iptables lineinfile github_hooks haproxy
git ec2_* win_*
patch apt s3_bucket yum datadog_event
redis mount mysql_user docker postgresql_ext jabber
cron jira shell mail rabbitmq_*
puppet docker_container slack
vmware_* irc bower npm

Galaxia de ansible

- Roles creados por usuarios, como los contrib de Drupal.
- En Github (código e issues).
- Algunos revisados.
- Instalación sencilla
- Maneja dependencias

```
# ansible-galaxy install  
username.role  
ansible-galaxy install \  
geerlingguy.drupal
```



The screenshot shows the Ansible Galaxy website interface. At the top, there is a navigation bar with the Ansible logo and the word 'GALAXY'. A search bar is present with the text 'Keyword: Apache2' and a 'CLEAR ALL' button. The search results are displayed in a grid of cards. Each card shows the role name 'apache2', a score of 'NA', a description, the author's name, supported platforms, tags, and the creation date. A 'POPULAR TAGS' sidebar is visible on the right, listing various tags and their counts.

Keyword	Search roles	SORT	Relevance
Keyword: Apache2			

Role Name	Score	Description	Author	Platforms	Tags	Created
apache2	NA	Installs Apache2	PeterMosmans	Debian, Ubuntu	web	7/1/15 2:03 AM
apache2	NA	Install Apache2	ktutumi	EL	web	4/20/15 2:11 PM
apache2	NA	Installs apache2				
apache2	NA	apache installation				

POPULAR TAGS	Count
system	1908
develop...	988
web	939
monitoring	369
database	366
networking	363
packaging	324
cloud	266
nosql	138
sql	131
cluster...	84

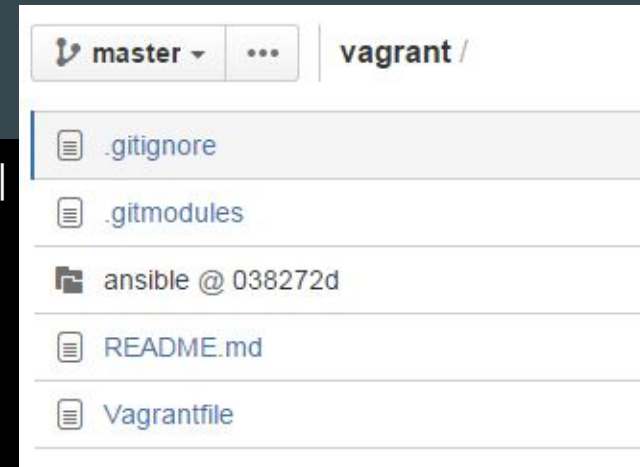
5. Ansible + Vagrant. Producción en local

Mi Vagrant

```
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "puphpet/debian75-x64"
  config.vm.hostname = "local1-all"

  #IP, memoria, puertos,...

  if which('ansible-playbook')
    # Provisioning configuration for Ansible.
    config.vm.provision "ansible" do |ansible|
      ansible.playbook = "ansible/local.all.yml"
      ansible.inventory_path = "ansible/roles/local-dev/files/hosts-local"
      ansible.limit = 'local.all'
      ansible.sudo = true
    end
  else
    config.vm.provision :shell, path: "ansible/windows.sh", args: ["dev"]
  end
end
```



Desventajas

- Necesitas un rol de desarrollo. Es feo.
- Un **windows.sh** en el repo de ansible. Ansible no provisiona en windows.
- Demasiada caché en local. (if local...)

Ventajas

- Configuración Varnish local = Varnish prod = Varnish pre (= Varnish testing?)
- Configuración *** local = *** prod = *** pre (= *** testing?)
- Reproducible al 95% la lógica de producción. (Balanceo, cores, RAM, ...)
- Entorno nuevo en 1 hora.
- Actualizaciones (A Php5.6, MySQL 5.6, drush8, ...)

6. ¡Yo quiero! Pero, ¿por dónde empiezo?

Empezando. Servidores instalados

1. Elegimos un servicio (nginx/apache2)
2. Organizamos el directorio de ansible.
3. Creamos rol local-dev y rol del servicio
4. local-dev instala el servicio (misma versión que producción)
5. Rol del servicio añade un fichero funcionando producción (ports.conf)
6. Creamos dos playbooks:
 - a. webservers.dev.yml (incluye local-dev, apache2 y variables locales)
 - b. webservers.pro.yml (incluye apache2 y variables locales)

6. ¡Yo quiero! Pero, ¿por dónde empiezo?

Empezando. Servidores instalados

7. Ejecutamos los playbooks.



6. ¡Yo quiero! Pero, ¿por dónde empiezo?

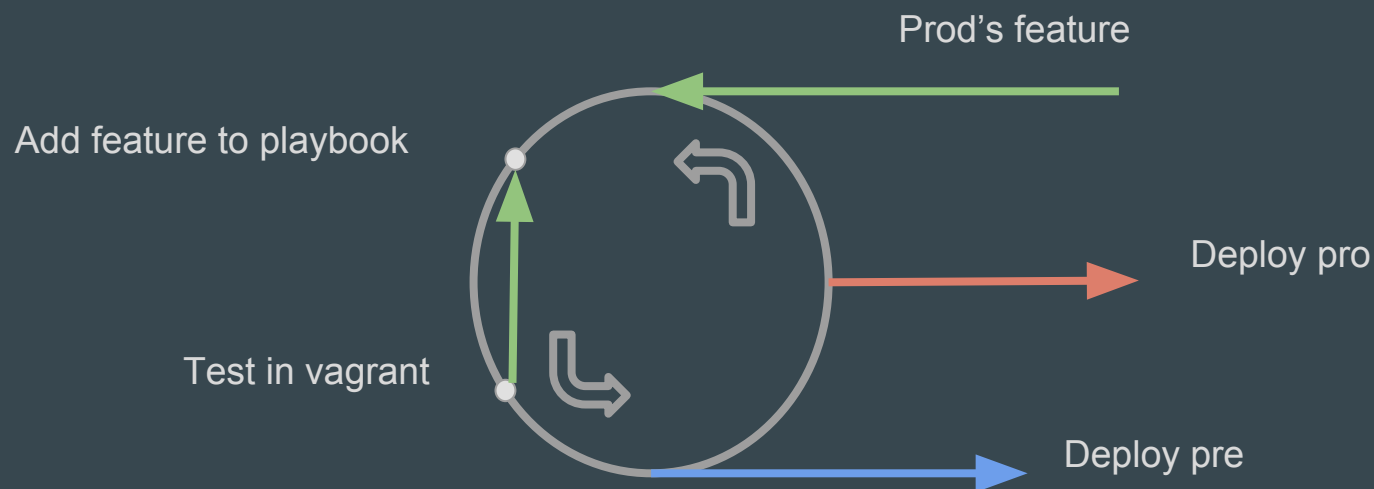
Empezando. Servidores instalados

8. Lo gozamos



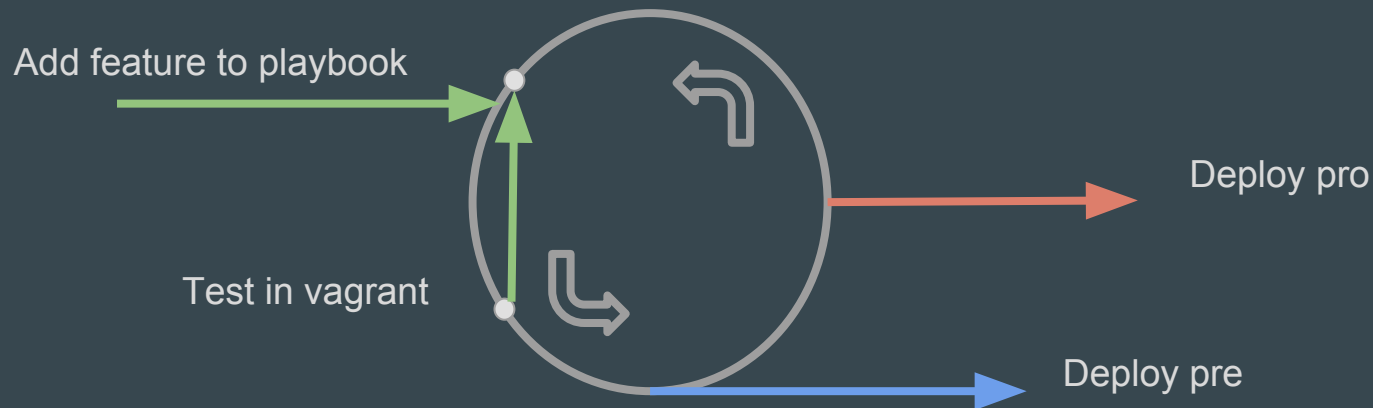
Empezando. Servidores instalados

9. Vamos iterando y añadiendo complejidad al rol
10. Hasta que dev tenga la misma configuración que prod



Empezando. Servidores nuevos

1. Añadir roles uno por uno
 - a. Mirar si en ansible-galaxy, github, ... hay algo que podamos usar.
2. Desplegar en producción



High five!

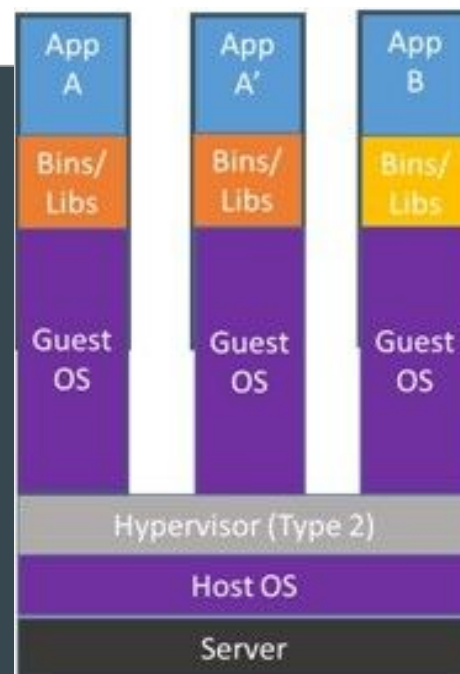
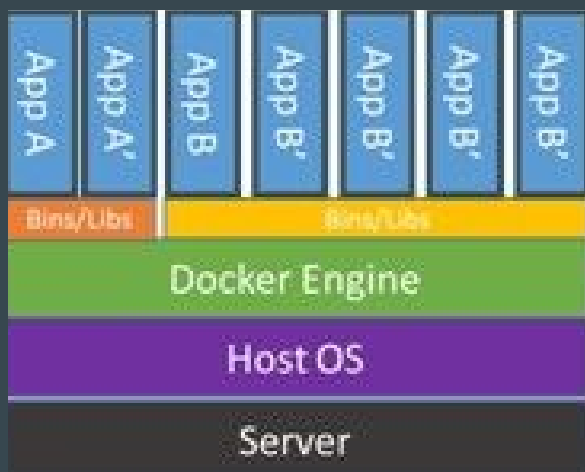
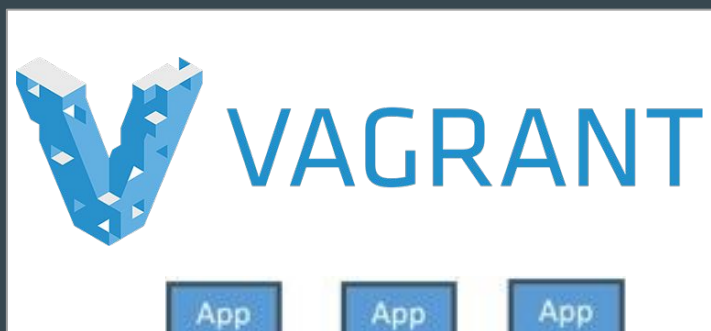
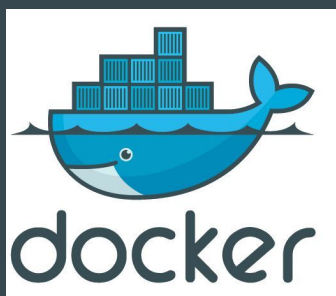


7. Amigos de Ansible. ¿Docker?

¿Algún DevOps en la sala?



Docker vs Virtual Machine



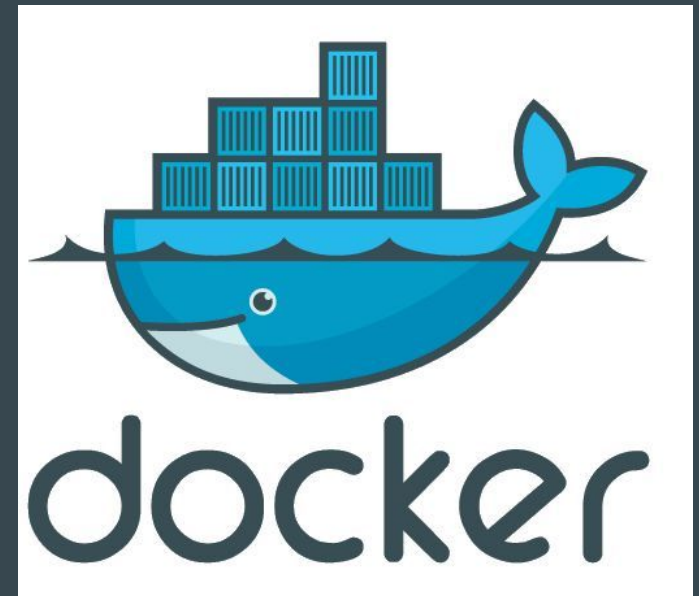
Docker + Ansible

- Docker puede ejecutar Ansible

```
FROM ansible/ubuntu14.04-ansible:stable

# Add playbooks to the Docker image
ADD ansible /srv/example/
WORKDIR /srv/example

# Run Ansible to configure the Docker image
RUN ansible-playbook site.yml -c local
```



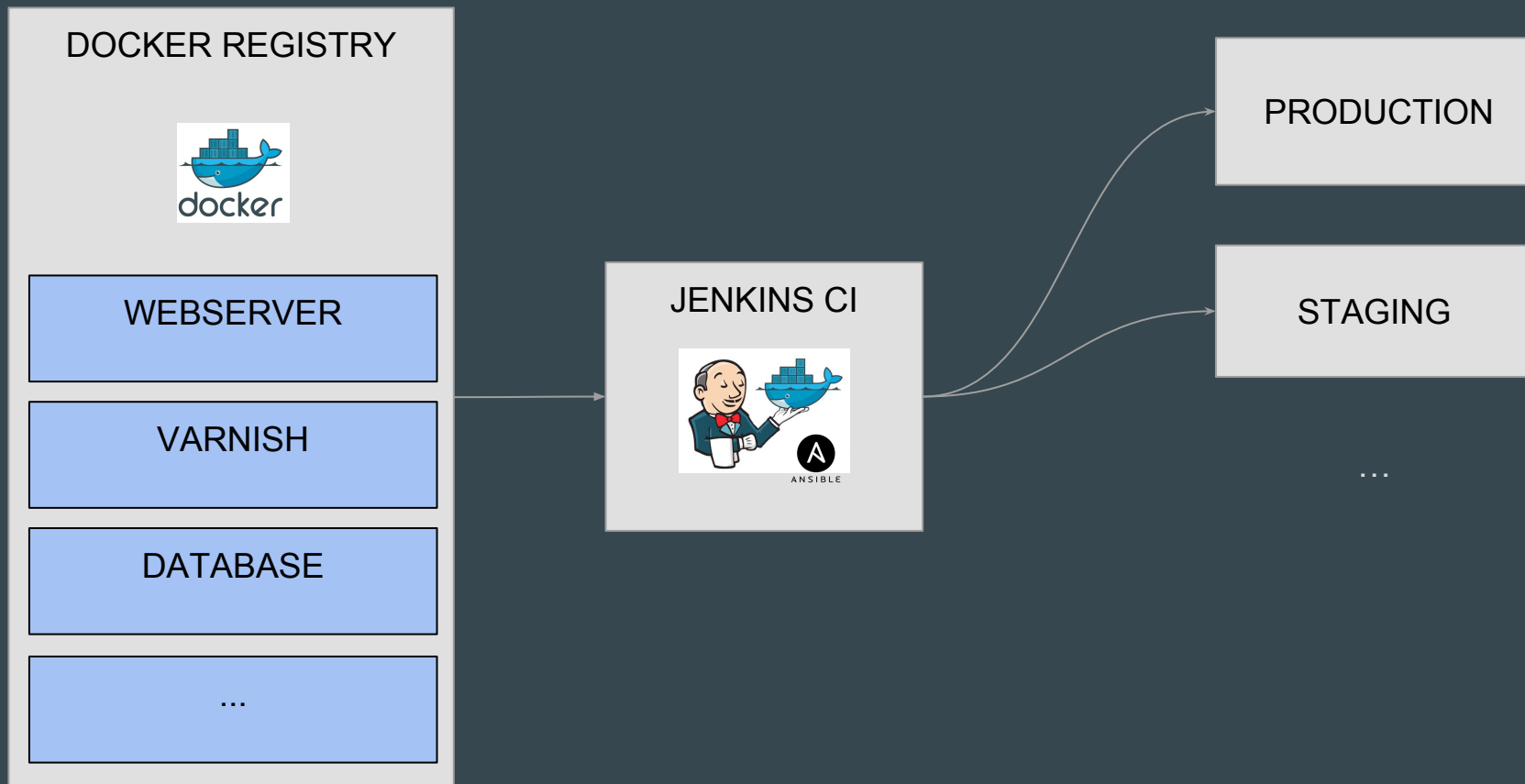
Docker + Ansible

- Y Ansible puede desplegar Docker

```
- name: php-fpm
  docker:
    image: php:5.6fpm
    volumes_from: NAS
    state: running
```



Hasta el infinito ¡y más allá!



Futuro

1. Despliegue de entornos de usar y tirar (<http://simplytest.me>)
2. Reusabilidad de componentes y recetas
3. Fiabilidad de componentes y recetas
4. Despliegue de código con Ansistrano (<https://github.com/ansistrano>)
 - a. Deploy
 - b. Rollback

¿Preguntas?

¡Gracias!

**¡No olvides darme feedback
y pillar pegatinas!**

@isholgueras

nacho@letshackity.com