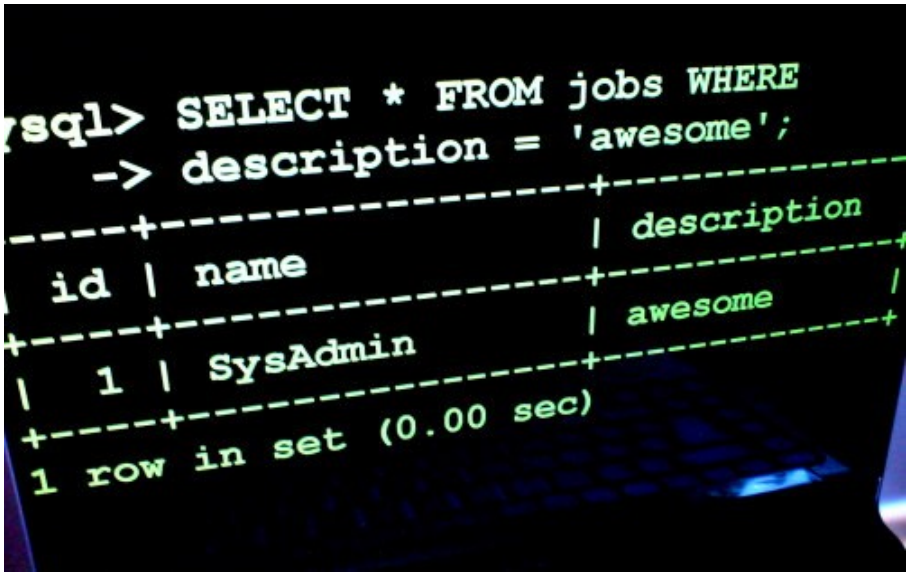
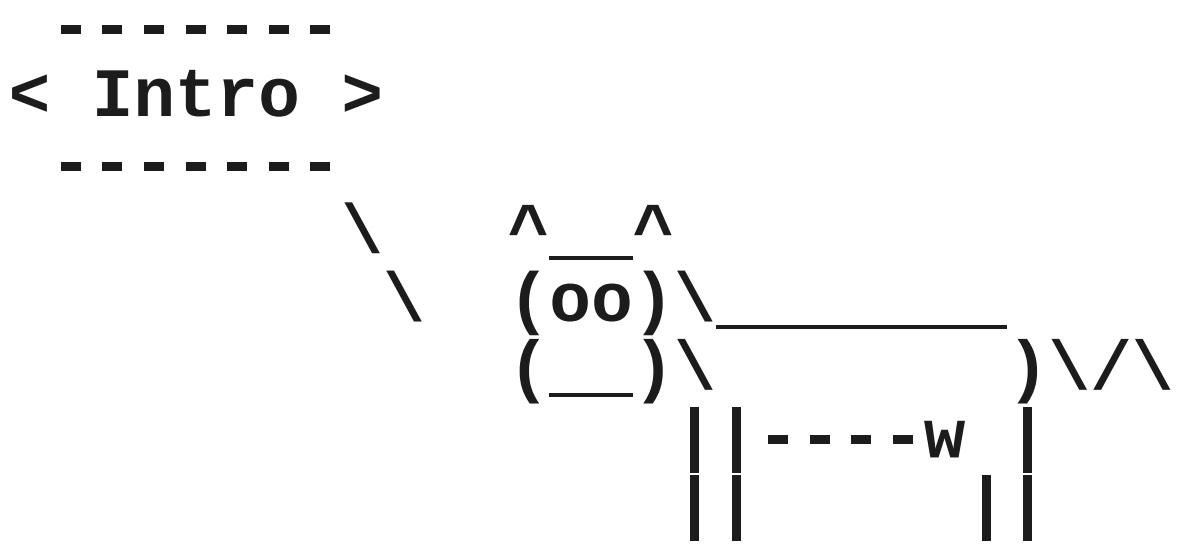
Ansible: Let the Sysadmins take a nap since 2012.

$ whoami

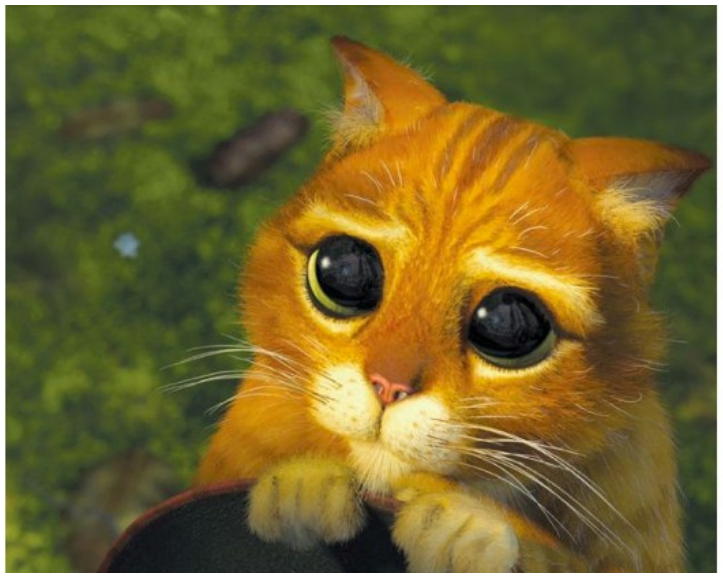
```
{
"name"      : "Adrià García-Alzórriz",
"job"       : "GNU/Linux Sysadmin",
"interests": ["FLOSS enthusiast", "cats"],
"remarks"   : "Not a guru",
"contact"   : "adria@fsfe.org",
"GPG"       : "0x09494C14",
"notes"     : "I'm just here for the food."
}
```

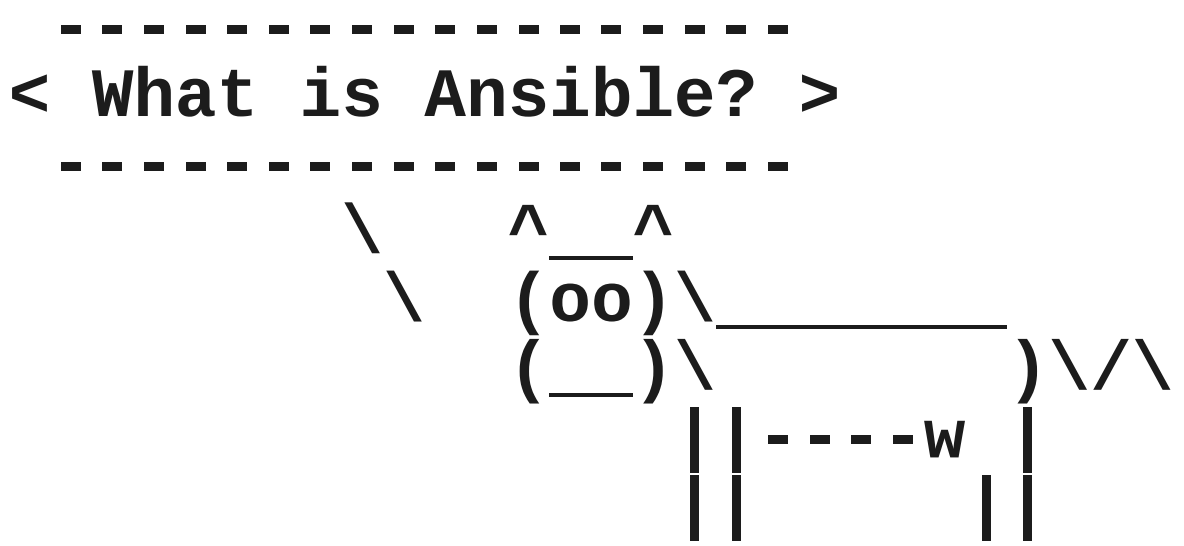



Intro

Intro

- **In the beginning, dinosaurs populated the Earth.**
- **“640K ought to be enough for anybody”.**
- **Pets and Cattle: new tools appear.**




What is Ansible?

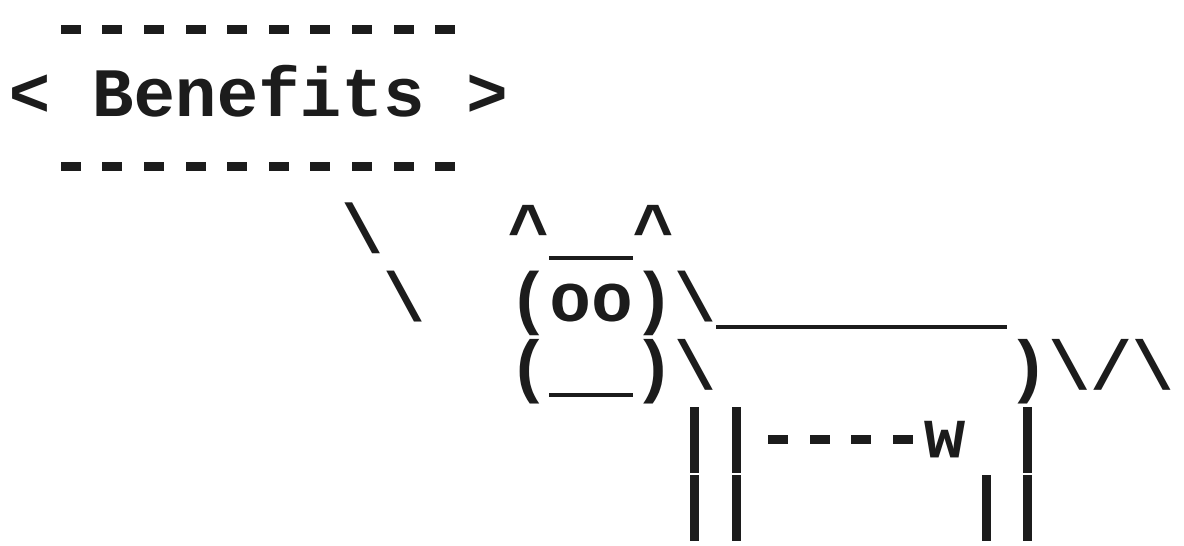
What is Ansible?

- **CM/ Orchestration software like Chef, Puppet, Saltstack…**
- **Free Software (GPL) created by Ansible inc.**
- **Python ~~2.6~~ 2.7.**
- **Agentless.**
- **No central node.**
- **SSH.**
- **Both push & pull modes.**

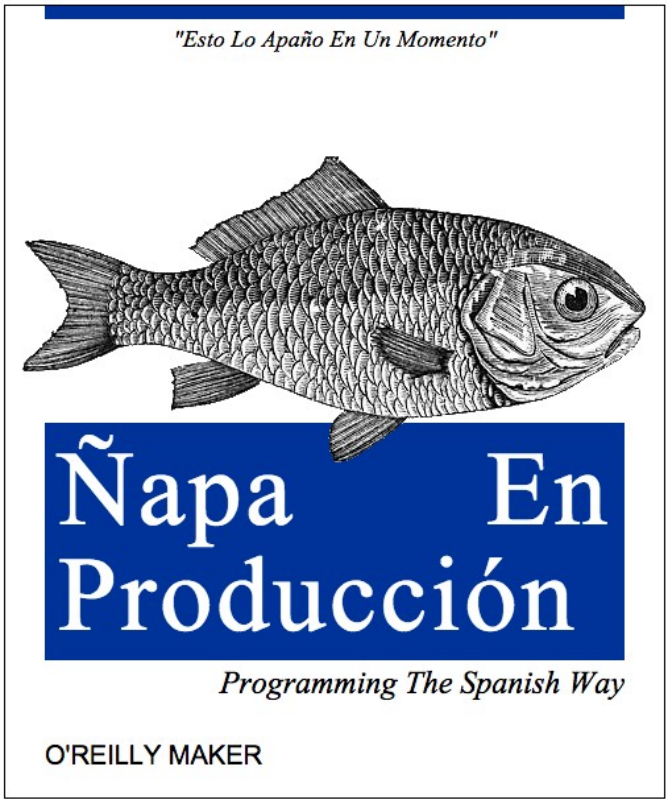
(…)

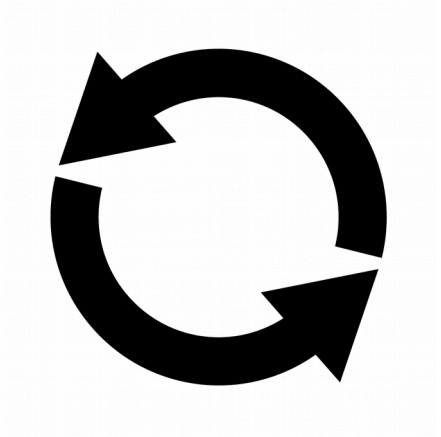
What is Ansible?

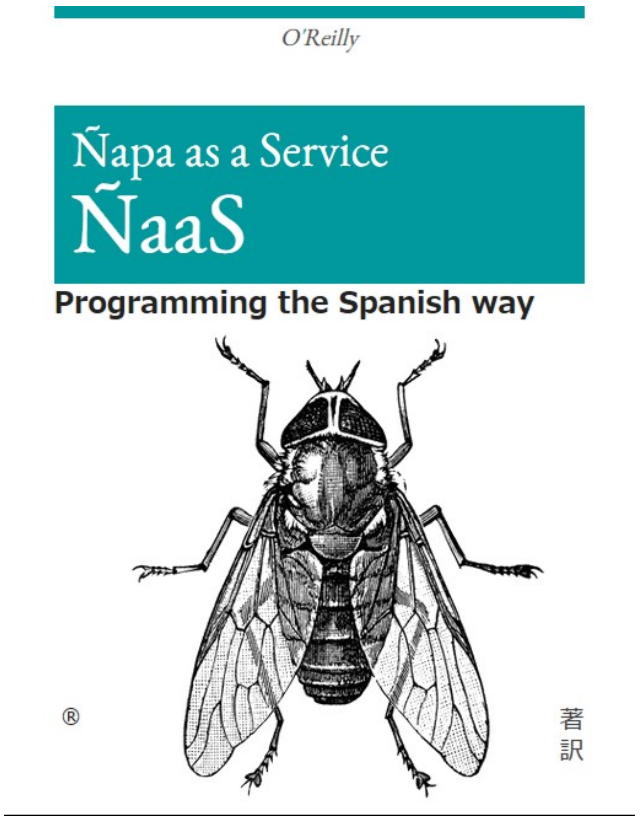
- **Idempotent.**
- **Parallel execution.**
- **Extensible: +340 modules such as AWS, RabbitMQ, file management, SELinux.**
- **YAML and Jinja2.**
- **Installation:** `apt-get` **or** `git clone`


Benefits

Benefits


"Esto Lo Apaño En Un Momento"
Ñapa En Producción
Programming The Spanish Way
O'REILLY MAKER



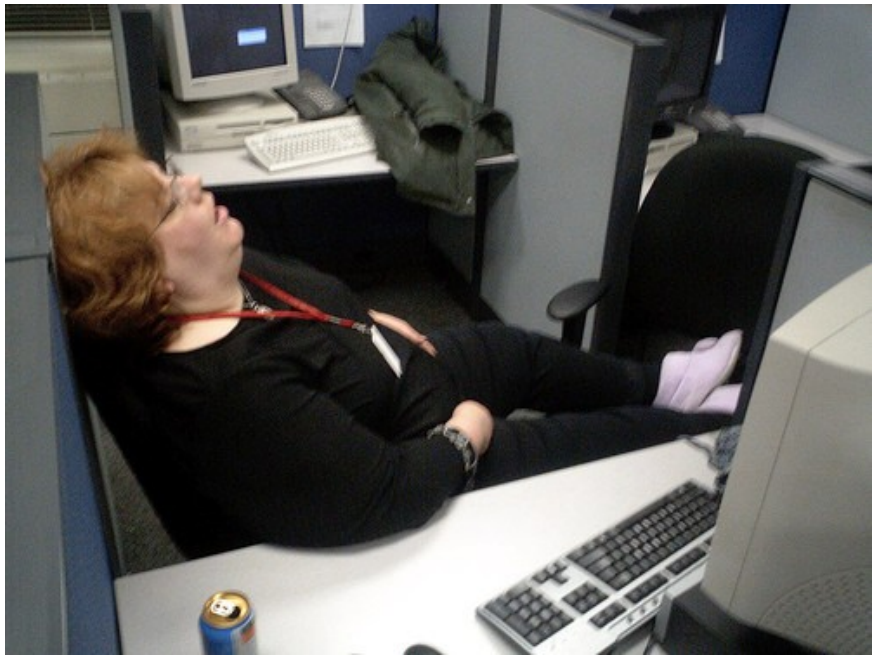

O'Reilly
Ñapa as a Service
ÑaaS
Programming the Spanish way
®
著
訳

Benefits

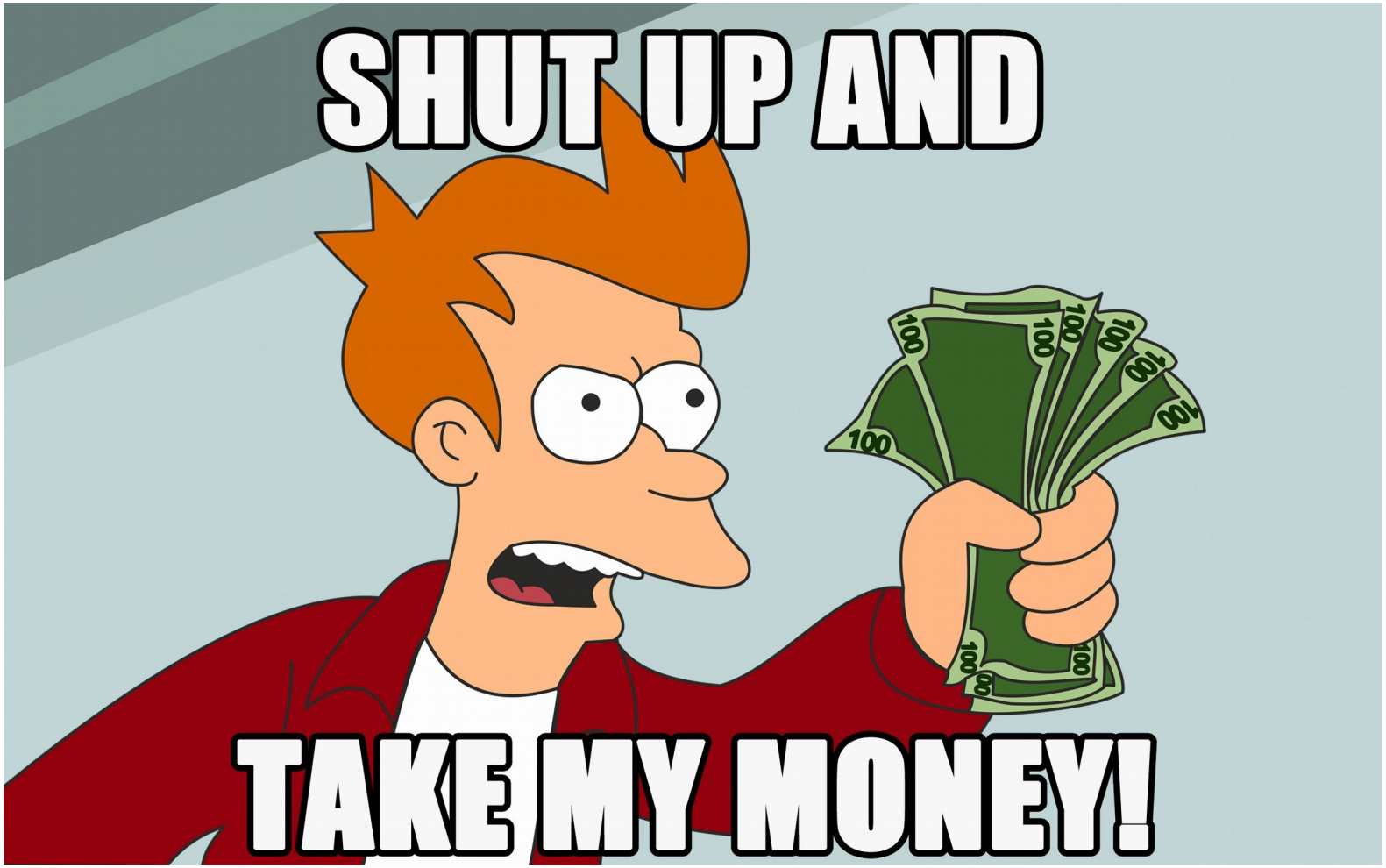
- It eliminates a bunch of the “works on my machine” issues.
- Fast to learn and fast to set up.
- Efficient: no extra software on your servers.
- Secure: strong use of SSH.
- No more weird ports or agents.
- Prevent repetititive tasks.
- Declarative code.

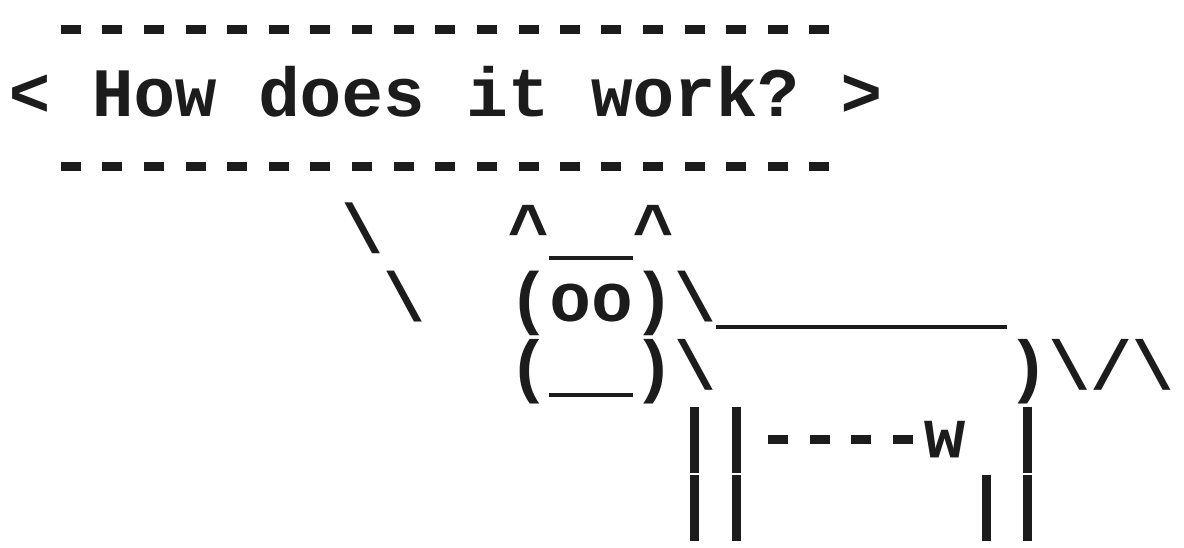
Benefits

- **Are you lazy? While magic happens, you are getting paid for.**



Benefits



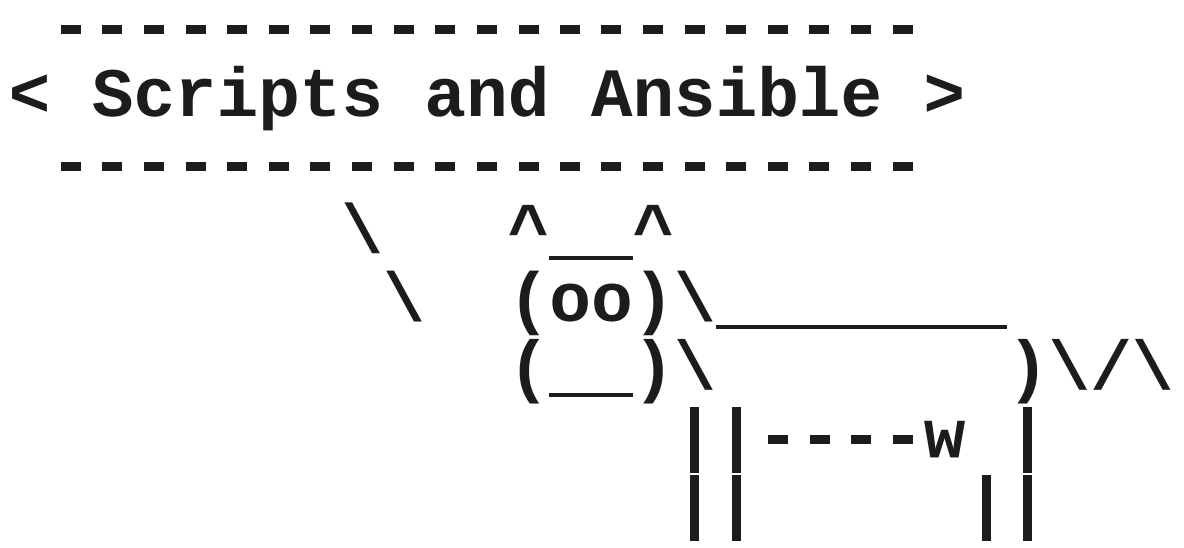

< How does it work? >
^__^
(oo)_____
(__)\ )\/\
||----w |
|| ||

How does it work?

- **$ ansible localhost -m ping**

```
localhost | success >> {
    "changed": false,
    "ping": "pong"
}
```

- **$ ansible-playbook -i production web.yml**
- **Ansible Tower (Web GUI, scheduling, mgmt, REST API).**


Scripts and Ansible

Scripts and Ansible

```
# Install the PGP key
gpg --keyserver keyserver.ubuntu.com --recv-keys 561F9B9CAC40B2F7
gpg --armor --export 561F9B9CAC40B2F7 | apt-key add -

# Install https support for apt
apt-get install apt-transport-https -y

# Add the passenger apt repository
echo "deb https://oss-binaries.phusionpassenger.com/apt/passenger
raring main" > /etc/apt/sources.list.d/passenger.list
chown root: /etc/apt/sources.list.d/passenger.list
chmod 600 /etc/apt/sources.list.d/passenger.list
```

(...)

Scripts and Ansible

```
(...)
# Update the apt cache so we can use the new repo
apt-get update

# Install nginx
apt-get install nginx-full passenger -y

# Set up passenger in the nginx configuration
sed -i "s/# passenger_root/passenger_root/" /etc/nginx/nginx.conf
sed -i "s/# passenger_ruby/passenger_ruby/" /etc/nginx/nginx.conf

# Start nginx
service nginx restart
```

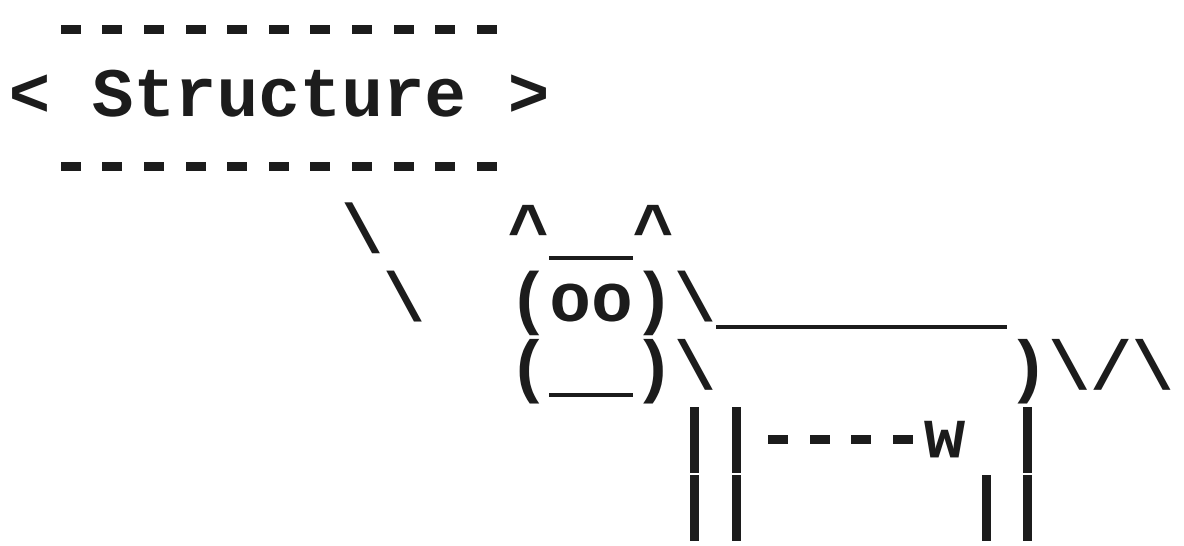
Scripts and Ansible

Scripts and Ansible

```
---
- hosts: all
  tasks:
  - name: Ensure the PGP key is installed
    apt_key: state=present id=AC40B2F7
      url="http://keyserver.ubuntu.com/pks/lookup?
op=get&fingerprint=on&search=0x561F9B9CAC40B2F7"

  - name: Ensure https support for apt is installed
    apt: state=present pkg=apt-transport-https

  - name: Ensure the passenger apt repository is added
    apt_repository: state=present
      repo='deb https://oss-binaries.phusionpassenger.com/apt/passenger raring
main'
                                                                    (...)
```

Scripts and Ansible

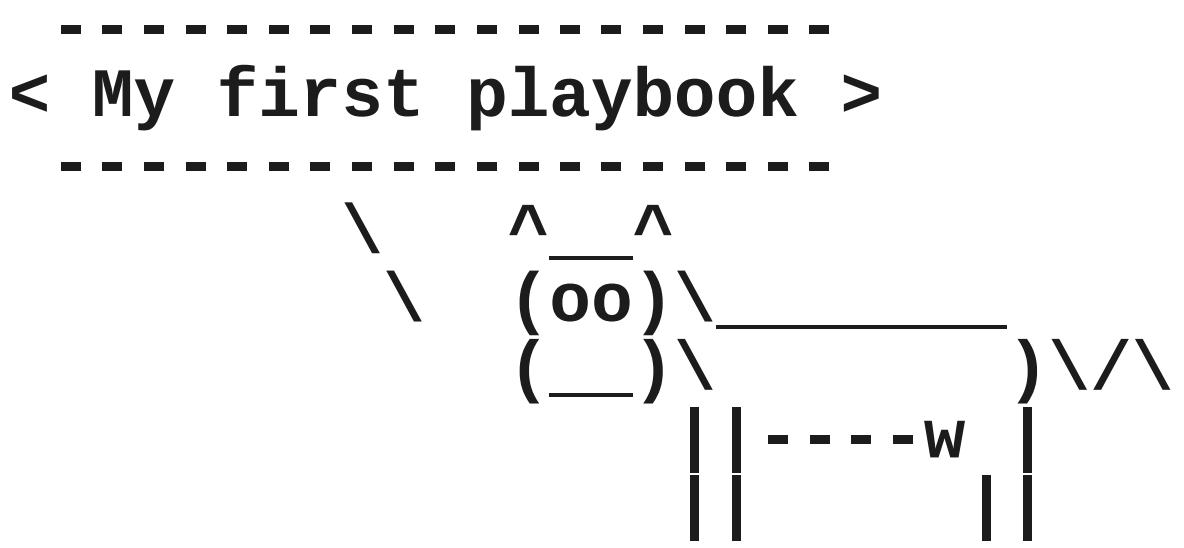
```
(...)
  - name: Ensure nginx is installed
    apt: state=present pkg=nginx-full

  - name: Ensure passenger is installed
    apt: state=present pkg=passenger update_cache=yes

  - name: Ensure the nginx configuration file is set
    copy: src=/app/config/nginx.conf dest=/etc/nginx/nginx.conf

  - name: Ensure nginx is running
    service: name=nginx state=started
```


Structure

Structure

```
$ tree ~/ansible/
ansible/
├── files
├── handlers
│   ├── apache2.yml
│   └── main.yml
├── inventory
│   ├── db
│   ├── production
│   └── webservers
├── meta

(...)
```

```
(…)

├── roles
│   ├── common.yml
│   ├── production.yml
│   └── stage.yml
├── tasks
│   └── main.yml
├── templates
└── vars
    └── group_vars
        └── all
```


My first playbook

My first playbook

```
---
# ~/ansible/roles/common/tasks/etckeeper.yml
- name: Install Etckeeper.
  apt:
    name: etckeeper
    state: present
  tags: etckeeper

# Creates an empty repository
- name: Init Etckeeper
  command: 'etckeeper init'
  tags: etckeeper
```

My first playbook

```
$ ansible-playbook -i myInventory common.yml -u user -K
SUDO password:

PLAY [Install all our base software] ******************************************

GATHERING FACTS ***************************************************************
ok: [10.10.99.163]

TASK: [common | Install Etckeeper.] *******************************************
ok: [10.10.99.163]

TASK: [common | Init Etckeeper] ***********************************************
changed: [10.10.99.163]

PLAY RECAP ********************************************************************
10.10.99.163               : ok=3    changed=1    unreachable=0    failed=0
```

My first playbook: Variables

```
[...]
command: ntpdate {{ ntpServer }}
[...]

$ head -n2 ntpServer group_vars/all/time
---
ntpServer: hora.uv.es
```

My first playbook: Facts

```
$ ansible -m setup localhost
localhost | success >> {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "10.10.99.193",
            "172.17.42.1"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::2289:84ff:fe63:52b"
        ],
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "02/19/2013",
        "ansible_bios_version": "V2.14",
        "ansible_cmdline": {
            "BOOT_IMAGE": "/vmlinuz-3.16.0-4-amd64",
            "quiet": true,
            "ro": true,
            "root": "/dev/mapper/whatever-vg-root"
        },
```

(...)

My first playbook: Templates

```
$ grep -v '#' ~/ansible/roles/common/templates/ssmtp.conf.j2
root={{ smtpRecipient }}
mailhub={{ smtpServer }}
rewriteDomain={{ domain }}
hostname={{ hostname.stdout }}
AuthUser={{ smtpUser }}
AuthPass={{ smtpPass }}
FromLineOverride=no
```

My first playbook: Conditionals

```
# ~/ansible/roles/common/tasks/main.yml
# [...]
- name: etckeeper check init commit.
  stat: path=/etc/.git/COMMIT_EDITMSG
  register: etckeeperYetCommited

- name: etckeeper first commit
  command: 'etckeeper commit “Initial commit.”'
  when: not etckeeperYetCommited.stat.exists
```

My first playbook: Loops

```
---
- name: Install Etckeeper
  apt:
    name: "{{ item }}"
    state: present
  with_items:
    - git
    - tig
    - etckeeper
```

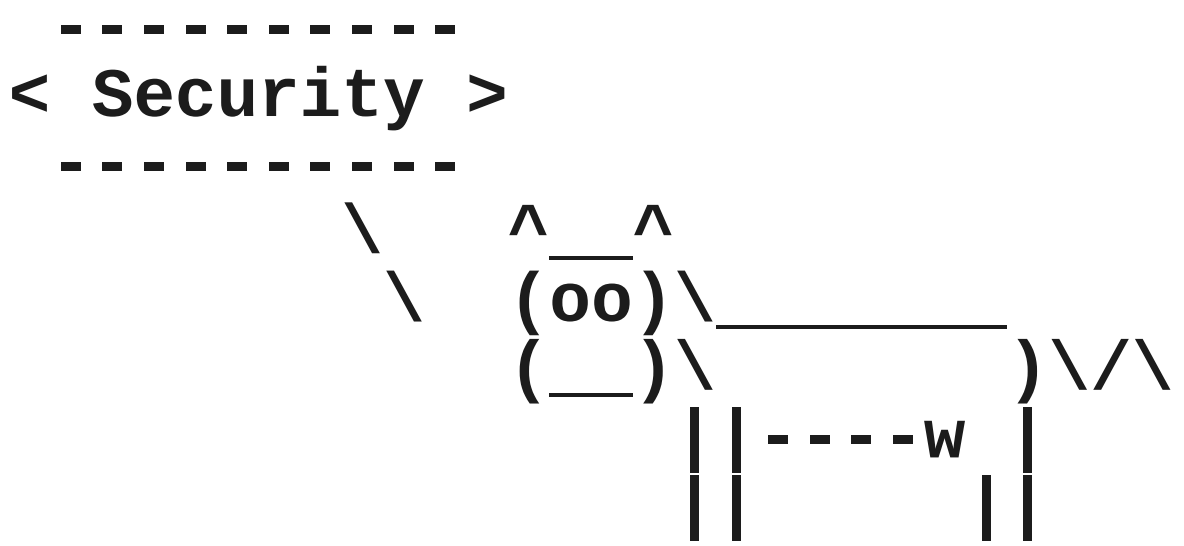
(...)

My first playbook: Loops

```
# [...]
- name: Etckeeper works together with Git
  lineinfile:
    dest: /etc/etckeeper/etckeeper.conf
    backrefs: yes
    regexp: "{{ item.regexp }}"
    line: "{{ item.line }}"
  with_items:
    - { regexp: '^VCS=”bzr”', line: '#VCS=”bzr”' }
    - { regexp: '^#?VCS=”git”', line: 'VCS=”bzr”' }
```

My first playbook: Handlers

```
# ~/ansible/roles/common/tasks/foo.yml
- name: commit etckeeper
  debug: msg=”Commit foo file changes to etckeeper”
  notify:etckeeper update

# ~/ansible/roles/common/handlers/main.yml
- shell: 'cd /etc && git-ls files -m'
  register: etckeeperNeedsToCommit
  notify: etckeeper commit

- name: etckeeper commit
  command: 'etckeeper commit “ {{ etckeeperNeedsToCommit.stdout}} ”'
  when: etckeeperNeedsToCommit.stdout
```


Security

Security

- `.gitignore`
- **Ansible Vault**

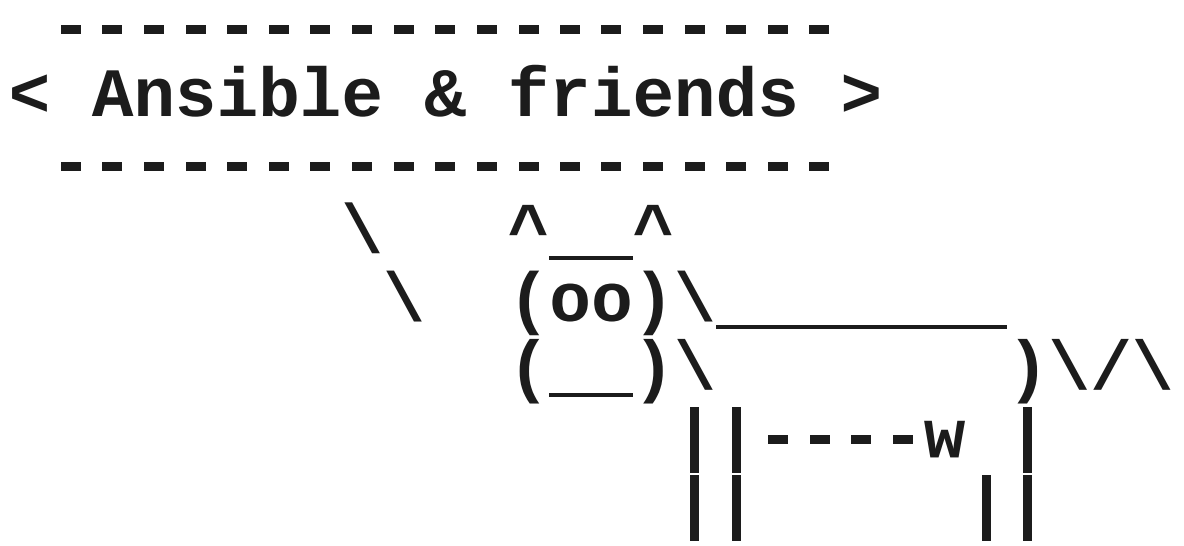
```
$ ansible-vault <encrypt|decrypt|edit> file

$ head group_vars/snmp.vault
$ANSIBLE_VAULT;1.1;AES256
35646539653361386635313264 33...
...386333

[--ask-vault-pass]
```

(...)

Security

- **OpenSSL to the rescue!**

```
$ openssl aes-256-cbc -salt -a -e -in «src» -out
«dst» -k Str0ngPazw0r && shred -u «src»


$ echo “decryptKey: Str0ngPazw0r” > secrets.vault


$ ansible-vault encrypt secrets.vault


task: openssl aes-256-cbc -salt -a -d -in «src»
-out «dst» -k {{ decryptKey }}
```


Ansible & friends

Ansible & friends

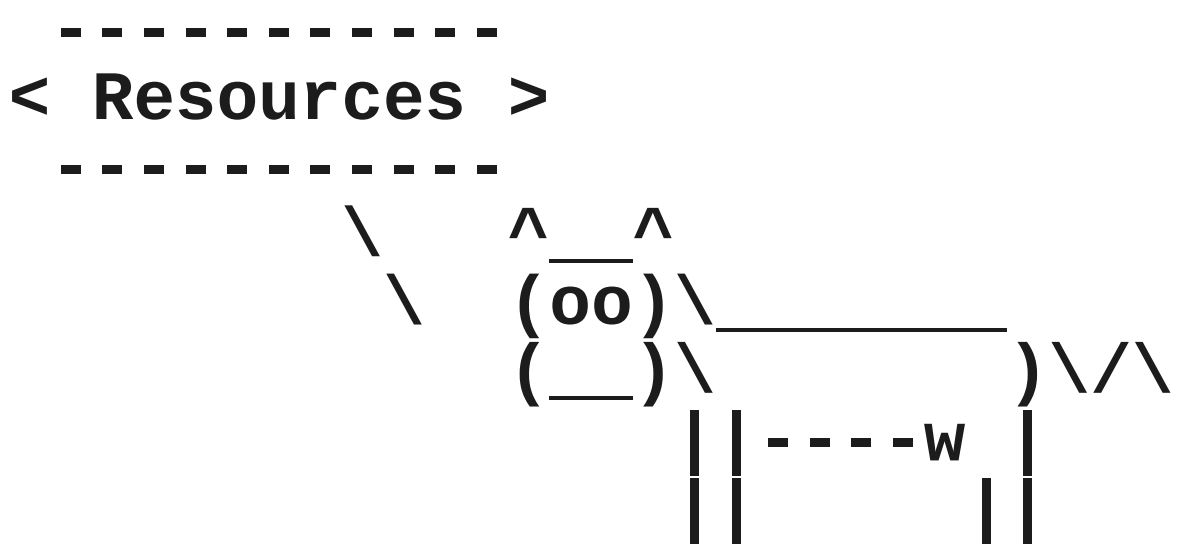
- **Vagrant**

```
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "debian/jessie64"

  config.vm.network "forwarded_port", guest: 80, host: 8080

  config.vm.provision :ansible do |ansible|
    ansible.playbook = "myPlaybook.yml"
  end
end
```

Ansible & friends

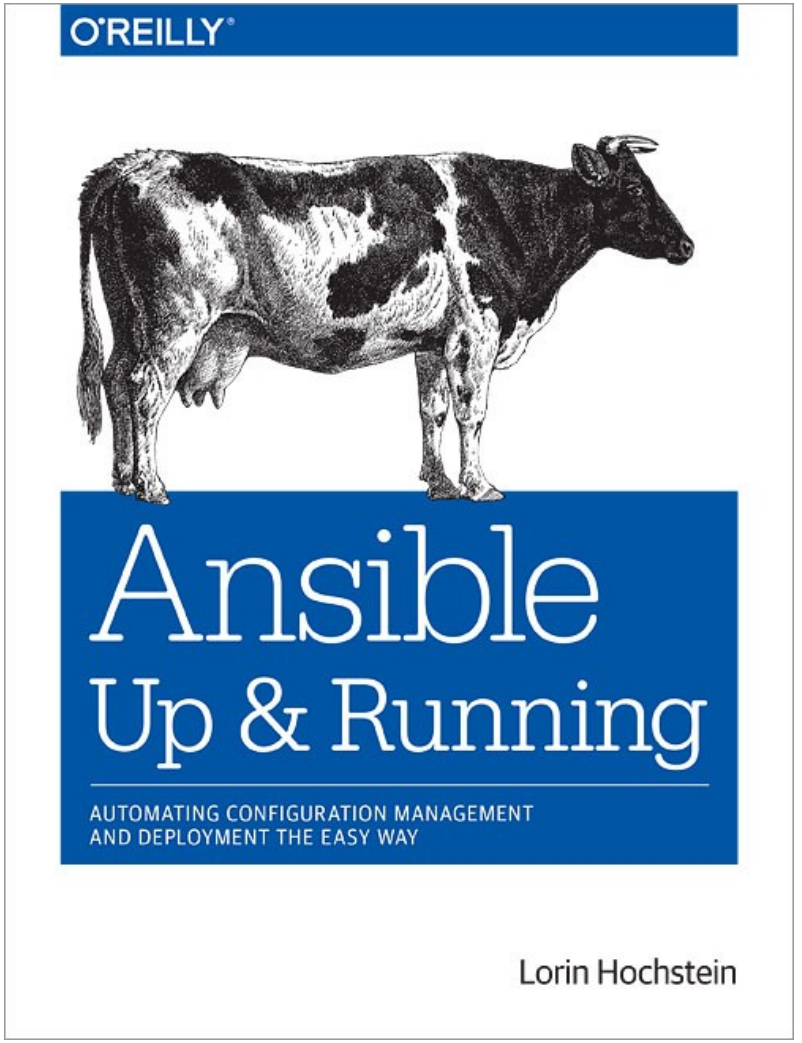
- **Docker**

```
FROM ansible/centos7-ansible:stable
# or, for example, 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

# Other Dockerfile directives are still valid
EXPOSE 22 3000 80
ENTRYPOINT ["/usr/local/bin/apachectl", "-DFOREGROUND"]
```


Resources

Resources

- **Ansible: Up & Running**
- **ansible-doc**
- **Google Groups**
- **Galaxy**
- **Github**
- **IRC**



Slides I sto... er borrowed (aka Credits)

- https://blog.engineyard.com/2014/pets-vs-cattle
- https://servercheck.in/blog/devops-humans-ansible-presentation-drupalcon
- http://www.slideshare.net/headmin/ansible-meetup-hamburg
- http://es.slideshare.net/johnthethird/ansible-presentation-24942953
- http://es.slideshare.net/pas256/how-ansible-makes-automation-easy
- http://es.slideshare.net/bcoca/ansible-config-mgmt
- http://es.slideshare.net/schedrov/ansible-44106255
- https://twitter.com/devsidestory

Questions?

Thank you!