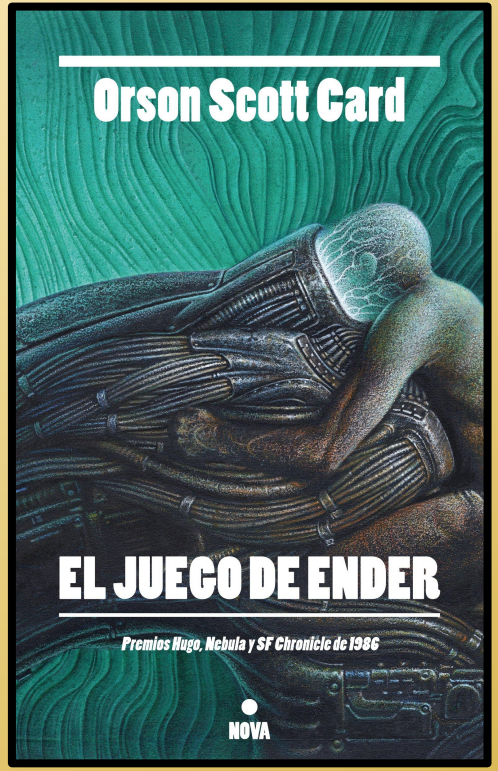
ANSIBLE

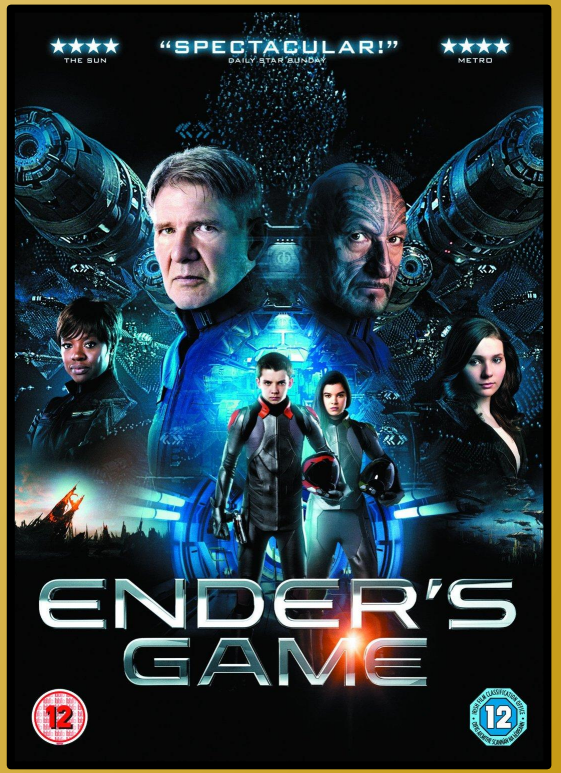
Una herramienta para controlarlas a todas

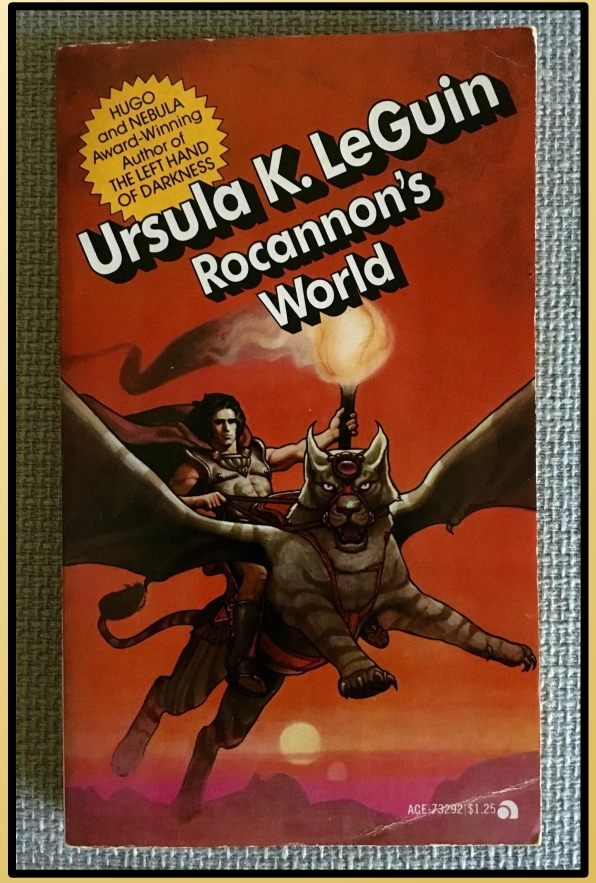
¿Que es y para que sirve Ansible?

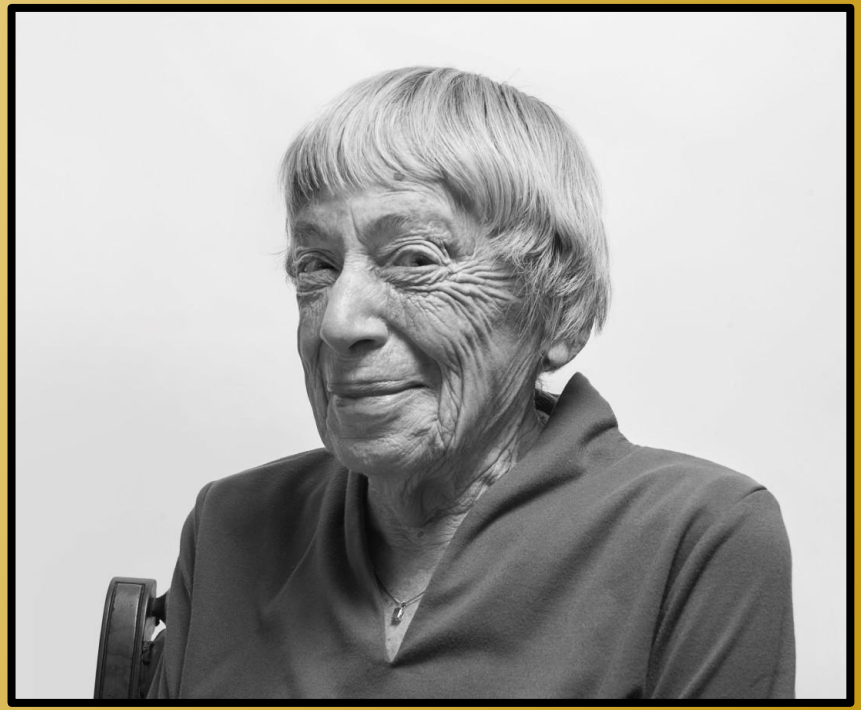
Gestión de configuración y provisión automática de aplicaciones, sistemas e infraestructuras.

Origen


Orson Scott Card
EL JUEGO DE ENDER
Premios Hugo, Nebula y SF Chronicle de 1986
NOVA


★★★★ THE SUN
"SPECTACULAR!" DAILY STAR SUNDAY
★★★★ METRO
ENDER'S GAME
12
12

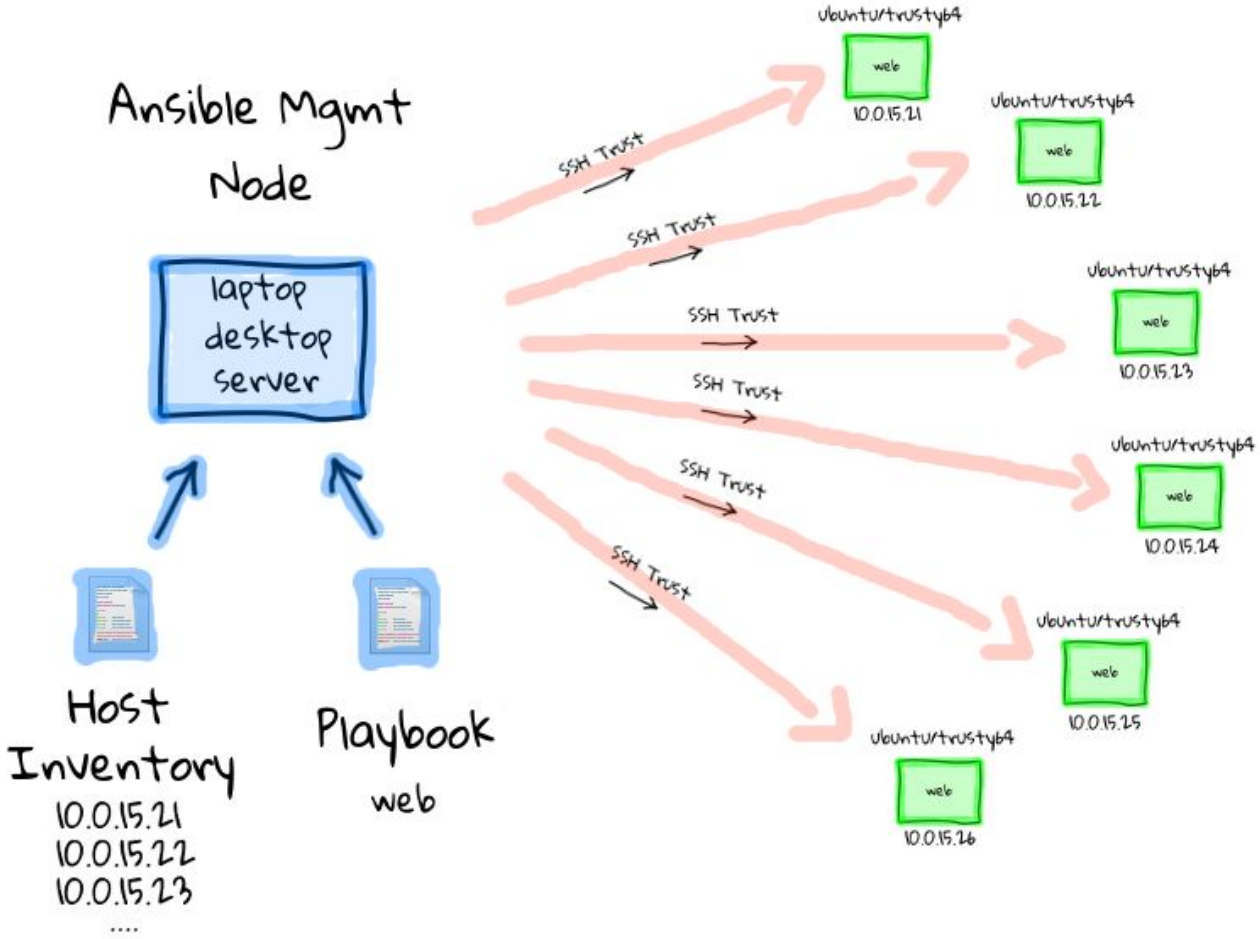




answerable -> ansible

Características

- Escrito en python
- Multisistema
- Propio DSL (declarative syntax language)
- Idempotente
- Basado en YAML
- Human Readable
- Agentless (ssh)
- Potente, sencilla y eficiente
- Gran comunidad y desarrollo activo:
  - + 3000 contribuidores
  - + 30000 commits
  - ansible inc (red hat)


Ansible Mgmt Node
laptop
desktop
server
Host Inventory
10.0.15.21
10.0.15.22
10.0.15.23
....
Playbook
web
SSH Trust
SSH Trust
SSH Trust
SSH Trust
SSH Trust
SSH Trust
ubuntu/trusty64
web
10.0.15.21
ubuntu/trusty64
web
10.0.15.22
ubuntu/trusty64
web
10.0.15.23
ubuntu/trusty64
web
10.0.15.24
ubuntu/trusty64
web
10.0.15.25
ubuntu/trusty64
web
10.0.15.26

Instalación

```
$ pip install ansible

https://bootstrap.pypa.io/get-pip.py
```

Preparación

- terminal
- maquinas!
- ansible docs / ansible github
- atom:
  - ansible-language
  - ansible-linter

Configuración global

ANSIBLE.CFG (INI format)

- ANSIBLE_CONFIG (Variable de entorno)
- ansible.cfg (directorio actual)
- .ansible.cfg (home de usuario)
- /etc/ansible/ansible.cfg

Inventario

Definición de hosts y variables

- Multiples formatos: ini, yml, json
- configuracion de conexion
- grupos de hosts (anidables)

```
mail.example.com
backup ansible_host=192.0.2.50

[madrid]
mad[1:10].example.com

[barna]
bar[1:10].example.com

[spain:children]
barna
madrid
mail.example.com

[spain:vars]
ansible_user=root
my_personal_variable=’hello world’
```

- ansible_host
- ansible_user
- ansible_ssh_pass
- ansible_ssh_private_key_file
- ansible_become
- ansible_become_pass
- ansible_become_method

Ansible CLI

```
$ ansible [host] [options]

$ ansible miserver -a “uptime”
```

Principales:

-m	module
-a	arguments
-i	inventory
-k/K	password / sudo password
-b	sudo (become)

debug:

--syntax check	syntax check
-v	(verbose)

otros:

-l	limit
-c	connection
-f	forks
-e	variables

Yaml Syntax

- Formato de serializacion
- json supermineralizado
- + potente, + legible
- - eficiente

Yaml Syntax

```
Int:          4
float:        4.1
string:       “texto”
bool:     true | True | Yes | yes
```

Yaml Syntax

```
clave: valor
una_lista:
 -  un elemento
 -  otro elemento
un_diccionario:     # un nivel de
indentación
    clave: valor
```

json

```
{
  "person": {
    "nombre": "Rick",
    "apellido": "Sanchez",
    "año": 1982,
    "deportes": ["padel", "golf"]
    “galaxia”: “C-137”
  }
}
```

yaml

```
person:
  nombre: Rick
  apellido: Sanchez
  año: 1982
  deportes:
    - tennis
    - golf
  galaxia: C-137
```

Yaml acepta json

```
Una_lista:
 -  [“una”, “lista”, “anidada”]
 -  { diccionario: anidado, }
 -  otro_diccionario:
      lista_vacia: []
      edad: 1
```

Bloque texto

```
literal: |
  todo lo que se encuentre
  dentro del nivel de indentación
  es texto multilínea.
  aquí se respetarán todos los saltos de
  línea.

plegado: >
  aunque haya varias líneas, los
  saltos de línea son reemplazados
  por espacios
```