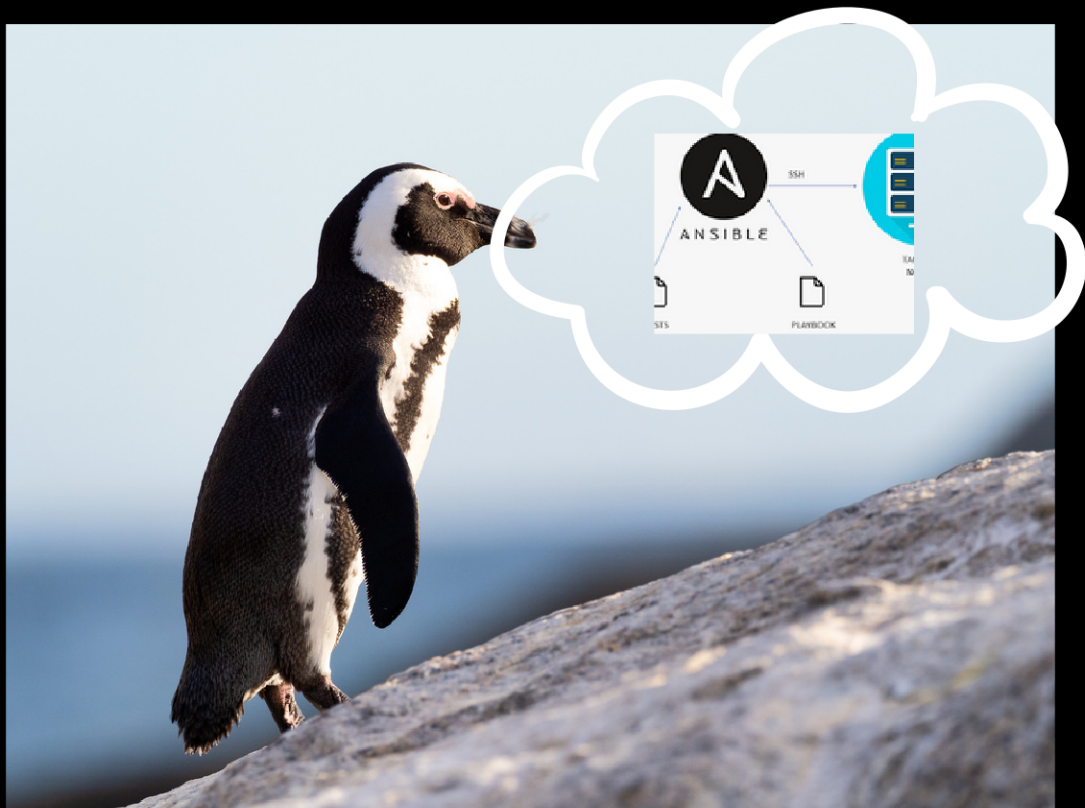


BASH

INTRODUCCION

SERVICES



LINUX

SHELL

ANSIBLE

2 situaciones hipoteticas y mecanismos de afrontamiento

CASE I

Pongamonos en este hipotetico caso tienes varios servidores(1 o 2) ,y tu sabes un poco de S0 linux te dan la tarea de saber en que estado estan esos servidores(1 o 2),puedes realizar algo como:

- 1.entrar a cada uno de ellos y ejecutar los comandos necesarios.
- 2.crear un script hacer que se ejecuten en los demas .
- 3.verificar , aprender algo de ansible
- 4.o 6 Million Ways como dice Harvey Specter.

Prerequisitos

tener remotos servidores de preferencia Ubuntu Servers ,una forma facil es mediante digital Ocean,te muestro como crearlos adecuados a nuestro caso.

Create Droplets

Choose an image

Distributions

Container distributions

Marketplace

Custom images

Ubuntu

20.04 LTS LTS

Fedora

Select version

Fedora

Select version

Debian

Select version

CentOS

Select version

Rocky Linux

Select version

Choose a plan

Help me choose

SHARED CPU

DEDICATED CPU

Basic

General Purpose

CPU-Optimized

Memory Optimized

Storage Optimized

Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and desktop environments.

CPU options

Regular Intel with SSD

Premium Intel with NVMe SSD

Premium AMD with NVMe SSD

\$5/mo

\$0.00/hour

1 GB / 1 CPU

25 GB SSD disk

100 GB transfer

\$10/mo

\$0.00/hour

2 GB / 1 CPU

50 GB SSD disk

2 TB transfer

\$15/mo

\$0.02/hour

2 GB / 2 CPUs

60 GB SSD disk

3 TB transfer

\$20/mo

\$0.02/hour

4 GB / 2 CPUs

80 GB SSD disk

6 TB transfer

\$40/mo

\$0.00/hour

8 GB / 4 CPUs

160 GB SSD disk

5 TB transfer

\$80/mo

\$0.00/hour

16 GB / 8 CPUs

320 GB SSD disk

6 TB transfer

Basic Droplet plans have replaced Standard Droplet plans. Users who have Standard Droplets using deprecated plans can still create Droplets with those plans using the API.

Each Droplet plan includes free outbound data transfer which is shared between all Droplets each billing cycle. Inbound bandwidth to Droplets is always free. Learn more or try our price calculator.

Add block storage

Add volumes

Choose a datacenter region

New York

San Francisco

Amsterdam

Singapore

London

Frankfurt

Toronto

Bangalore

Prerequisitos

en la forma de autenticacion es mediante el protocolo ssh y necesitaras tener tus llaves que debes crearlas en tu maquina y agregarla la llave publica(.pub) a los droplets a los que estamos creando, fijate en **NEW SSH KEY** y **apretas a crear droplets**.

WPL - RESOURCE

defaultKey1 ☐ DEFAULT

All resources created in this dashboard will be members of the same VPC network. They can communicate securely over their Private IP addresses. [What does this mean?](#)

Authentication

☒ **SSH key**
An SSH key pair authenticates without a password.

☐ Password
Create a new password to access Droplets (not secure)

SSH keys

☒ defaultKey1 ☐ defaultKey2 ☐ defaultKey3 ☐ defaultKey4

[New SSH Key](#)

Select additional options

☐ **Snapshot backups** [Learn more](#)
A system [full backup](#) is taken once a week, and each backup is retained for 4 weeks. \$1.00/mo per Droplet, 20% of the Droplet price

☐ Monitoring
Enables additional Droplet metrics collection, monitoring, and alerting. FREE

☐ IPv6
Enables public IPv6 networking. FREE

☐ Use data
Enter user data when you create a Droplet to perform tasks or run scripts on the next user during a Droplet's first boot. FREE

Finalize and create

How many Droplets?
Deploy multiple Droplets with the same [configuration](#).

Choose a hostname
Give your Droplets an identifying name you will remember them by. Your Droplet name can only contain alphanumeric characters, dashes, and periods.

2 Droplets

ub-01

ub-02

Add tags

Use tags to organize and relate resources. Tags may contain letters, numbers, colons, dashes, and underscores.



subsite enable script

Select Project

Assign Droplets to a project

test-project

[Create Droplet](#)

DROPLETS (3)	
 ub-01	178.128.1.1
 ub-02	178.128.1.1

crear un script hacer que se ejecuten en los demas

- 1.crea una lista de las IP's las publicas y guardalas en server.txt
- 2.crea un fichero checkstate.sh un script.
- 3.crea un fichero manage.sh un script

```
1.echo "178.111.222.123">> server.txt
   echo "178.111.222.123">> server.txt
```

2../chekstate.sh es donde pondremos el comando a ejecutar.

```
#!/bin/bash
# kernel_version
# memory_usage

server_name=$(hostname)

function main() {
    echo "|> WELCOME ${server_name} and you memory usage"
    free -h
    echo "======"
}
main
```

3. nano **./manage.sh** , algo importante no es seguro entrar como usuario **root** ,debes agregar otro usuario si quieres, en mi caso use root.

```
#!/bin/bash
main() {
    echo "-i opcion for ssh"
    read dir
    for server in $(cat server.txt) ;do ssh root@${server} -
i $dir 'bash -s' < ./checkstate.sh ; done
}
main
```

este loop va a todos los servidores encontrados en el fichero server.txt y ejecuta el comando ssh.

crear un script hacer que se ejecuten en los demas

OUTPUT

```
(kali@feather)~[/borrame]
$ bash manage.sh
-i option for ssh
/home/kali/ansible/.ssh/id_rsa
|> WELCOME ub-01 and you memory usage
      total      used      free
Mem:    976Mi    150Mi    126Mi
Swap:    0B       0B       0B
=====
|> WELCOME ub-02 and you memory usage
      total      used      free
Mem:    976Mi    151Mi    204Mi
Swap:    0B       0B       0B
=====
```

notice

es un simple ejemplo , que se podria volver mas complejo y necesitar mas conocimientos de bash scripting por eso no seria recomendable.

averiguar , aprender algo de ansible.....

- 1.que es.
- 2.partes importantes.

1 ¿que es?

Ansible es una solución de automatización de IT de línea de comandos que puede implementar cambios de configuración, software y realizar muchas otras tareas de forma automática. Para obtener más información sobre qué es Ansible vaya Ansible.

User-friendly no debemos tener el conocimiento de algun lenguaje de programacion ,solo realizamos scripts in YAML,pero tambien tiene algunas desventajas por ser stateless .

averiguar , aprender algo de ansible.....

1 partes importantes

- Control Node

Una computadora que tiene linux y ansible puede tomar el rol de nodo central.

no se puede utilizar una máquina con Windows como nodo de control.

- Managed Nodes

Los sistemas que manejas entran en esta categoria se conectan median ssh y deben tener python mejor establecerlo a Python3 para algunos modulos.

- Inventory

En este file llamando *inventory* establecemos los hosts tambien el usuario ,si es por ssh la llave ,etc. Solo establecemos los hosts como grupos encontraras mas en la documentacion.

- Task

una task esta incluida en un *playbook.yml* ,un task es una unidad de trabajo el alcance o quienes se ejecutaran se define por la etiqueta *hosts*.

```
host: webservers
task:
  - name: ....
```

- Playbook

Contiene las tareas ,los hosts y variables ,estos se ejecutan secuencialmente.

un playbook: *playbook.yml*

- Handlers

los hanlers se encargan de las siguientes acciones:

start , *restar* ,*stop* de un servicio. todo esto pasa al final de un task,existe un handler by default.

- Roles

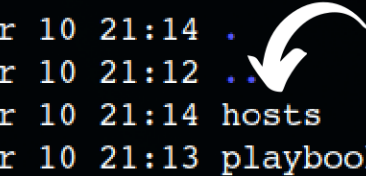
guardar variables y se basan en estructuras de directorios.

averiguar , aprender algo de ansible.....

1 partes importantes

estructura

```
(kali@feather)-[~/borrame/ansible-pl]  
$ ls -al  
total 16  
drwxr-xr-x 2 kali kali 4096 Mar 10 21:14 .  
drwxr-xr-x 3 kali kali 4096 Mar 10 21:12 ..  
-rw-r--r-- 1 kali kali 295 Mar 10 21:14 hosts  
-rw-r--r-- 1 kali kali 73 Mar 10 21:13 playbook.yml
```



solo necesitaremos crear 2 file uno host sin formato y el otro tipo YML .

averiguar , aprender algo de ansible.....

1.debemos establecer la informacion de los hosts, dijimos eso se realizaba en el inventory file.

/etc/ansible/hosts

esto seria la ruta por default ,pero puedes crear inventories en el folder de algun proyecto y especificar la direccion con la flag -i.

hosts

```
[servers]
server1 ansible_host=178.128.159.138
ansible_ssh_private_key_file=/home/kali/ansible/.ssh/id_rsa
ansible_user=root
server2 ansible_host=178.128.159.139
ansible_ssh_private_key_file=/home/kali/ansible/.ssh/id_rsa
ansible_user=root
[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

ansible_python_interpreter estableces el uso de python3, **firewalld** no esta disponible para python2 y nuevos modulos que no los encontraras usando python2.

ya lo configuramos nos aseguramos con .

```
ansible-inventory --list -y
ansible ansible all -m shell -a "free -h"
```

si lo ejecutas obtendras lo mismo es una forma mas sencilla.

CASE II

Pongamonos en este hipotetico caso ,tienes a esos hosts inseguros y debes hacer que cumplan con un conjunto de requisitos mínimos de seguridad :

- 0.Settings.
- 1.Patch el S0.
- 2.Acceso remoto seguro.
- 3.Quitar los servicios que no son utilizados .
- 4.Customizar el login.

estructura del inventory

```
[servers]
server1 ansible_host=164.92.90.10
ansible_ssh_private_key_file=/home/kali/ansible/.ssh/id_rsa

[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

0. Settings

```
/ansible-secutiry
|----deployer
|----inventory
|----issue
|----motd
|----playbook_patch.yml
|----playbook_create_user.yml
|----playbook_disable_unused.yml
|----playbook_more.yml
|----playbook_upload_authorized_keys.yml
|----ssh-setup.j2
```

deployer archivo donde incluimos a un usuario y le asignamos algunos privilegios.

inventory archivo donde establecemos nuestros hosts

issue archivo con la funcionalidad de `/etc/issue` de cualquier linux SO, si no se ingresara este igual se mostrara.

motd archivo con el mensaje al loguearse.

ssh-setup.j2-(j2 significa jinja) ,tiene la base de un archivo `sshd_config` seria todas las configuraciones de ssh server.

*playbook** son cada una de las configuraciones que serializaran.

NOTAS

- no te asustes si ves muchas cows.
- en la ejecucion del `playbook_disable_unused.yml` habra unos servicios que no se podren desactivar, pero **no** tienen una respuesta de salida como failed.
- si realizas un playblook con todas las task deberas omitir el **become yes** .
- es mejor realizar todas esas tareas en un playbook central ,pero solo para entender mejor lo realizo paso por paso.

1. Patch el SO.

|----playbook_patch.yml

- deberias saber que es patching software ,deberias tener en cuenta que todo tiene su ciclo de vida y surgen nuevos riesgos pero el significado de patch es mas amplio.
- para este caso el modulo de ansible >package< es el que utilizamos para el SO actualizaciones, etc.
- los aplicamos a todos los hosts.

codigo

```
1 ---
2 - hosts: all
3   tasks:
4     - name: Perform full patching
5       package:
6         name: '*'
7         state: latest
```

ejecutalo con

ansible-playbook -i inventory playbook_patch.yml -u root

1. ACCESO REMOTO SEGURO.

|----playbook_create_user.yml

|----playbook_upload_authorized_keys.yml

- seras el usuario autorizado llamado deployer, debes tener un password seguro lo puedes tener con `mkpasswd --method=sha-512`, te adicionas a un grupo(sudo).

-system y state los dos parametro deben coordinar.

-debes agregarlo al archivos sudores que nos permite ejecutar los comando de sudo, para no complicarnos solo copiamos un fichero configurado "deployer" para esto utilizamos el modulo copy.

codigo

```
---
- hosts: all
  tasks:
    - name: Add local user
      user:
        name: deployer
        password: '$6$4Ie7zkjKNGPMp1UH$g3..9
        system: no
        group: sudo
        shell: /bin/bash
        home: /home/deployer
        create_home: yes
        state: present
```

```
- name: Add to sudoer for deployer user
  copy:
    dest: /etc/sudoers.d/deployer
    src: deployer
    owner: root
    group: root
    mode: 0440
    validate: /usr/sbin/visudo -csf %s
```

ejecutalo con

`ansible-playbook -i inventory playbook_create_user.yml -u root`

- el modulo author... es especifico para la copia de las llaves SSH asi que no utilices el modulo copy.

- el modulo template o copy cualquiera pero uso este porque tengo un archivo de extension j2(jinja).

- el modulo service que puedes manejar los servicios la mejor forma deberia ser utilizar la etiqueta handlers.

codigo

```
1 ---
2 - hosts: all
3   tasks:
4     - name: upload ssh key
5       authorized_key:
6         user: deployer
7         state: present
8         manage_dir: yes
9         key: "{{ lookup('file', '/home/ka
```

```
10 - name: configure ssh server
11   template:
12     src: ssh-setup.j2
13     dest: /etc/ssh/sshd_config
14     owner: root
15     mode: '0600'
16     validate: /usr/sbin/sshd -t -f %s
17     backup: yes
18
19 - name: restart ssh
20   service:
21     name: sshd
22     state: restarted
```

ejecutalo con

`ansible-playbook -i inventory playbook_upload_authorized_keys.yml -u root`

2. QUITAR LOS SERVICIOS QUE NO SON UTILIZADOS.

l----playbook_disable_unused.yml

- se quiere **desactivar** los paquetes en desuso y despues **desactivar** servicios que no vamos a utilizar.
- el modulo package es el que interactua con el manejador de paquetes y para nuestro ubuntu hosts seria apt ,ansible es stateless no elimina los paquetes por eso fijamos absent.
- el modulo sevice que ya utilizamos con ssh desactivara los sevicios que no se utilizan tambien con with_items estamos realizando un efecto loop que se cada item se sobrescribira en la etiqueta name.
- vars es una etiqueta global al igual que tasks donde fijamos variables pero deben estar casi siempre al principio.

codigo

```
1 ---
2 - hosts: all
3   vars:
4     unnecessary_services:
5       - postfix
6       - telnet
7     unnecessary_software:
8       - tcpdump
9       - nmap-ncat
10
```

```
11   tasks:
12     - name: remove packages
13       package:
14         name: "{{ unnecessary_software }}"
15         state: absent
16         become: yes
17
18     - name: Stop and disable services
19       service:
20         name: "{{ item }}"
21         state: stopped
22         enabled: no
23         become: yes
24         with_items: "{{ unnecessary_services }}"
25         ignore_errors: yes
```

ejecutalo con

ansible-playbook -i inventory playbook_disable_unused.yml -u deployer



3. CUSTOMIZAR EL LOGIN

|----playbook_more.yml

- solo utilizamos el modulo copy.
- el archivo motd es el banner de inicio solo cuando estas logueado el que un linux SO va reconocer bajo su especial ruta.
- el fichero issue puede contener ciertos códigos de escape para mostrar diferente información al iniciar el sistema.

codigo

```
1 ---
2 - hosts: all
3   tasks:
4     - name: Set a message of the day
5       copy:
6         dest: /etc/motd
7         src: motd
8         owner: root
9         group: root
10        mode: 0644
11        become: yes
```

```
12 - name: Set a login banner
13   copy:
14     dest: "{{ item }}"
15     src: issue
16     owner: root
17     group: root
18     mode: 0644
19     become: yes
20   with_items:
21     - /etc/issue
22     - /etc/issue.net
```

ejecutalo con

ansible-playbook -i inventory playbook_more.yml -u deployer



4. salida

ejecutalo con

```
ssh deployer@164.92.90.10 -i /home/kali/ansible/.ssh/id_rsa
```

```
Last login: Fri Mar 11 18:27:26 2022 from 190.129.181.10
USER DEPLOYER_
/ Q:      What's tiny and yellow and very, very, dangerous?
A:      A canary with the super-user password.
-----
      ^ ^
      (oo)_____
      (__)      )/
          ||----w |
          ||     ||
deployer@ubuntu-01:~$ exit
```

si ejecutas de esta forma no deberia estar permitido

```
ssh root@164.92.90.10 -i /home/kali/ansible/.ssh/id_rsa
```

```
└─$ ssh root@164.92.90.10 -i /home/kali/ansible/.ssh/id_rsa
only deployer is authorized \n \l
root@164.92.90.10: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

5. los firewalls ????

-Ansible tiene el modulo firewalld .

- pero eso depende del caso,por que podrias configurarlo para una aplicacion o una aplicacion que esta sobre docker y despues ver que lo de iptables por que habra un choque de reglas etc.

-si quieres configurarlas de forma basica puedes leer estos articulos.

[5 ways to harden a new system with Ansible](#)

[manage firewalld with ansible](#)

CONGRATULATIONS

Como siguiente paso deberias considerar aprender las fortalezas y debilidades de Ansible cual es su punto de enfoque ,esos conceptos importantes de “infrastructure-as-code”,este mini ebook esta inspirado en los ebooks que me encantaron de @bobbyiliev.