

# Script Servicio Correo

Ponemos el script para ejecutarlo en el directorio donde nos encontremos o queramos ejecutarlo.

```
GNU nano 7.2          inatalar_ansible.sh
#!/bin/bash
if command -v ansible &> /dev/null
then
    echo "Ansible ya esta instalado en tu ordenador"
    exit 0
fi
sudo apt update && sudo apt install -y ansible

if command -v ansible &> /dev/null
then
    echo "Ansible se ha instalado correctamente."
else
    echo "Error: La instalación de Ansible ha fallado."
    exit 1
fi

ansible --version

echo "Configuración completada."
```

Lo ejecutamos:

```
ismael@ismael-VMware-Virtual-Platform:~$ bash inatalar_ansible.sh
Ansible ya esta instalado en tu ordenador_
```

Y borramos todo para comprobar cada función del script.

```
ismael@ismael-VMware-Virtual-Platform:~$ bash instalar_ansible.sh
Obj:1 http://es.archive.ubuntu.com/ubuntu noble InRelease
Des:2 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Des:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [871
kB]
Des:6 http://es.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [197
kB]
Des:7 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [1
51 kB]
Des:8 http://es.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Package
s [660 kB]
Des:9 http://es.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-e
n [129 kB]
```

Vemos que ejecuta el comando a la perfección.

Ahora vamos con la configuración de un script de ansible para instalar correo, después hacemos un script para la creación y configuración de ansible.

```
GNU nano 7.2
#!/bin/bash

Directorio="playbook_correo"
archivo="$Directorio/instalacion.yml"
configuracion="$Directorio/configuracion"
principal="$configuracion/principal.cf.j2"
inventario="$Directorio/inventory.ini"

mkdir -p "$configuracion"
```

Y una vez aplicado creamos un conjunto de carpetas y pedimos un correo al usuario que esta ejecutando el script el cual queremos escribir, \*todo en local\*.

```
echo "Ingrese los correos electrónicos a los que desea enviar el mensaje (separados por espacios):"
read -r correos_input

IFS=' ' read -r -a usuarios_destino <<< "$correos_input"

# Contraseña simple (12345)
contrasena="12345"
```

Si el usuario esta creado lo omitirá. Una vez creado empezamos a meter comandos en los archivos:

```
cat > "$archivo" <<EOL
---
- name: Instalar y configurar Postfix para correos locales
  hosts: local
  connection: local
  become: true
  vars:
    myhostname: "localhost"
    mydomain: "local"
    usuarios_destino:
$(printf "    - %s\n" "${usuarios_destino[@]}")
    contrasena: "$contrasena"
```

Instalamos postfix con usuario en local, imprime el nombre que le acabas de proporcionar y la contraseña básica que hemos facilitado como "12345".

```
- name: Crear usuarios si no existen
  user:
    name: "{{ item.item.split('@')[0] }}"
    state: present
    shell: /bin/bash
    password: "{{ contrasena | password_hash('sha512') }}"
    create_home: yes
    update_password: on_create
  loop: "{{ usuario_existe.results }}"
  when: item.rc != 0
```

Crea el usuario en caso de que no exista.

Actualiza el archivo de configuración para ponerlo en local y que funcione el correo, y añadimos el inventario necesario para ejecutar un ansible.

```
cat > "$principal" <<EOL
myhostname = [{ myhostname }]
mydomain = [{ mydomain }]
myorigin = \${mydomain}
inet_interfaces = all
mydestination = \${myhostname}, localhost.\${mydomain}, \${mydomain}
relayhost =
mynetworks = 127.0.0.0/8
smtpd_relay_restrictions = permit_mynetworks reject_unauth_destination
smtp_use_tls = no
smtp_sasl_auth_enable = no
EOL

cat > "$inventario" <<EOL
[local]
localhost ansible_connection=local
EOL
```