

# Instalación de Ansible

Una vez creadas las máquinas de *nodo controlador* y de *plantilla para los servidores* con su correspondiente IP procedemos con la instalación manual de Ansible.

Si realizamos la instalación mediante *apt install ansible* descargaremos una versión antigua como es la *ansible 2.10*.

```
user@ControlNode:~$ su -
Conraseña:

user@ControlNode:~# apt-cache show ansible
Package: ansible
Version: 7.7.0+dfsg-3+deb12u1
Installed-Size: 258814
Maintainer: Debian Python Team <team+python@tracker.debian.org>
Architecture: all
Replaces: ansible-base (<= 2.10.5+dfsg-2)
Depends: ansible-core (>= 2.11.5-1~), python3:any, openssh-client | python3-paramiko (>= 2.6.0), python3-distutils, python3-dnspython, python3-httplib2, python3-jinja2, python3-netaddr, python3-yaml
Recommends: python3-argcomplete, python3-cryptography, python3-jmespath, python3-kerberos, python3-libcloud, python3-selinux, python3-winrm, python3-xmlltodict
Suggests: cowsay, sshpass
Breaks: ansible-base (<= 2.10.5+dfsg-2)
Description-en: Configuration management, deployment, and task execution system
Ansible is a radically simple model-driven configuration management,
multi-node deployment, and remote task execution system. Ansible works
over SSH and does not require any software or daemons to be installed
on remote nodes. Extension modules can be written in any language and
are transferred to managed machines automatically.
.
This package contains the ansible collections.
Description-md5: 4f8f35d982ca2a9d6d35c1bb2053b0e1
Homepage: https://www.ansible.com
Tag: admin::automation, admin::configuring, admin::file-distribution,
admin::package-management, implemented-in::python,
interface::commandline, role::program, use::configuring,
works-with::software:running
Section: admin
Priority: optional
Filename: pool/main/a/ansible/ansible_7.7.0+dfsg-3+deb12u1_all.deb
Size: 16948940
MD5sum: cbc1b38f2dd42d49997b36ea70f667a3
SHA256: 625d3724fe2e75bfcd8710a1549af473163cc5971350ac5d785c6ad1dec4af3
```

Para instalar Ansible es necesario contar previamente con *Python*, ya que este lenguaje es el motor sobre el cual se ejecuta la herramienta. Ansible está desarrollada principalmente en Python, lo que significa que depende de este intérprete tanto para su funcionamiento básico como para la ejecución de sus módulos.

```
user@ControlNode:~$ python3 --version
Python 3.11.2
```

El siguiente paso consiste en instalar *pip*, que es el gestor de paquetes de Python. Esta herramienta resulta esencial porque nos permite descargar, instalar, actualizar y gestionar librerías y módulos de Python. En el caso de Ansible, pip es especialmente útil porque facilita su instalación en cualquier sistema sin depender únicamente de los repositorios de la distribución, que en ocasiones pueden ofrecer versiones más antiguas.

```
root@ControlNode:~# apt update && apt install -y python3-pip
...
root@ControlNode:~# python3 -m pip -V
pip 23.0.1 from /usr/lib/python3/dist-packages/pip (python 3.11)
```

Procedemos a instalar ansible con la herramienta pip (Pip Instala Paquetes), la cual es una herramienta que empleada para instalar y administrar librerías y dependencias de software escritas en Python vemos como tenemos la siguiente advertencia.

```
root@ControlNode:~# pip install ansible
error: externally-managed-environment

  × This environment is externally managed
  ↳ To install Python packages system-wide, try apt install
     python3-xyz, where xyz is the package you are trying to
     install.

  If you wish to install a non-Debian-packaged Python package,
  create a virtual environment using python3 -m venv path/to/venv.
  Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make
  sure you have python3-full installed.

  If you wish to install a non-Debian packaged Python application,
  it may be easiest to use pipx install xyz, which will manage a
  virtual environment for you. Make sure you have pipx installed.

  See /usr/share/doc/python3.11/README.venv for more information.

  note: If you believe this is a mistake, please contact your Python installation or
        OS distribution provider. You can override this, at the risk of breaking your
        Python installation or OS, by passing --break-system-packages.
  hint: See PEP 668 for the detailed specification.
```

A continuación procedemos a instalar ansible en nuestro *ControlNode* de forma global pero con la recomendación que se nos indica por consola.

```
root@ControlNode:~# pip install ansible --break-system-packages
...
root@ControlNode:~# ansible --version
ansible [core 2.19.2]
  config file = None
  configured module search path = ['~/root/.ansible/plugins/modules',
  '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.11/dist-packages/ansible
  ansible collection location =
  /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.11.2 (main, Apr 28 2025, 14:11:48) [GCC 12.2.0]
  (/usr/bin/python3)
  jinja version = 3.1.6
  pyyaml version = 6.0.2 (with libyaml v0.2.5)
```

*Nota:* Si hicieramos la instalación de ansible mediante `apt install ansible` no tendremos la última versión.

## Documentación

### Documentación de ansible

Otro forma de ver información sobre Ansible es desde CLI. El comando `ansible-doc` es una de las herramientas más útiles para aprender y consultar documentación de Ansible directamente desde la línea de comandos. Gracias a él podemos ver información detallada sobre los módulos y colecciones instalados en nuestro sistema, sin necesidad de recurrir a la documentación web.

Una *colección* es un formato de distribución que agrupa varios tipos de contenido relacionados, entre ellos: módulos, roles, plugins, playbooks y documentación. Se introdujo como un cambio estructural desde Ansible 2.8 para facilitar la organización, mantenimiento y distribución del contenido más allá del núcleo principal.

Vemos las colecciones instaladas.

```
root@ControlNode:~# ansible-doc -l
amazon.aws.autoscaling_group
amazon.aws.autoscaling_group_info
amazon.aws.autoscaling_instance
amazon.aws.autoscaling_instance_info
amazon.aws.autoscaling_instance_refresh
amazon.aws.autoscaling_instance_refresh_info
...
```

Un *módulo* es la unidad básica de trabajo en Ansible. Es un pedazo de código, generalmente escrito en Python, que realiza una tarea específica en un nodo gestionado (equipo remoto), como instalar un paquete, copiar un archivo, gestionar usuarios o configurar servicios. Cada tarea en un playbook es una llamada a un módulo. Vemos información de un *módulo* en concreto.

```
root@ControlNode:~# ansible-doc -s ping
- name: Try to connect to host, verify a usable python and return `pong` on
success
ping:
  data:          # Data to return for the `ping` return value. If this
  # parameter is set to
  # `crash`, the module will
  # cause an exception.
```

Para ver información de un módulo en concreto podemos hacer *ansible-doc ansible.builtin.apt*.