modulos

multitud de módulos...

- sistemas
- dispositivos
- usuarios
- etc...

cada tarea está asociada a un módulo, con argumentos obligatorios y opciones

categorias de módulos que se veran (que no todas las disponibles)

- cloud
- clustering
- commands
- crypto
- database
- files
- identity
- inventary
- messaging
- monitoring
- network
- notification
- packaging
- remote management
- source control
- storage
- system
- utilities
- web infraestructures
- windows

`ansible-doc -l` muestra los módulos instalados en nuestro equipo

`ansible-doc <módulo>` nos da información del módulo

temario

- ficheros y openssl
- gestión de paquetes
- utilidades
- notificaciones
- bases de datos
- gestionar el sistema
- Windows
- Sistema de Control de Versiones
- Web y messaging
- 3:45 Cloud / Cluster
- 3:46 Monitorización

3:27 Ficheros y OpenSSL (I)

permite trabajar con ficheros, plantillas y directorios

- acl : establece y obtiene información de la listas de control de acceso
- archive : crea un fichero comprimido a partir de una lista de ficheros o estructura de directorios
- assemble : asambla un fichero de configuración desde fragmentos
- blockinfile : Inserta/Actualiza/Elimina un bloque de texto de un fichero
- copy : copiar ficheros a ubicaciones remotas (desde servidor Ansible → nodo remoto)
- fetch : copiar del nodo remoto al servidor
- file : establece atributos a ficheros
- find : devuelve una listsa de ficheros a partir de un patrón
- inifile : manejo de ficheros INI
- iso_extract : extrae ficheros de una imagen ISO
- lineinfile : asegura que una línea está en un fichero o reemplaza la misma con el uso de REGEX
- patch : aplica parches usando GNU/Patch
- replace : reemplaza las coincidencias de un texto por otro
- stat : obtiene información del fichero o del FS
- synchronize : rsync
- tempfile : crear ficheros/directorios temporales
- template : uso de plantillas
- unarchive : extraer ficheros (en remoto)
- xatrr : atributos extendidos

los módulos para OpenSSL:

- openssl_privatekeys : generar claves privadas
- openssl_publickey : generar claves públicas

copy

https://docs.ansible.com/ansible/latest/modules/copy_module.html#copy-module

- obligatorios
  - dest = /path/destino
- opcionales
  - backup = yes/no
  - content = «contenido»
  - force = yes/no
  - owner = usuario
  - group = grupo
  - mode = modo
  - src = /path/origen

```
- name: copiar configuración
  copy: src=apache2.conf dest=/etc/apache2/apache2.conf owner=www-data group=www-
data
- name: crear contenido en fichero
  copy: content="Mi contenido en un fichero" dest=/etc/mi.conf
```

template

https://docs.ansible.com/ansible/latest/modules/template_module.html#template-module

- obligatorios
  - dest = /path/destino
  - src = /path/origen
- opcionales
  - backup = yes/no
  - force = yes/no
  - owner = usuario
  - group = grupo
  - mode = modo

```
- name: copiar pantilla de configuración
  template: src=apache2.conf.j2 dest=/etc/apache2/apache2.conf backup=yes
```

file

https://docs.ansible.com/ansible/latest/modules/file_module.html#file-module

- obligatorios
  - path = /path/al/fichero
- opcionales
  - backup = yes/no
  - force = yes/no
  - owner = usuario
  - group = grupo
  - mode = modo
  - state
    - file
    - link
    - directory
    - hard
    - touch
    - absent

```
...
tasks:
  - name: propiedades del fichero
    file: path="/path/al/fichero" backup=yes mode="777"
  - name: borrar el fichero
    file: path="/path/al/fichero" backup=yes absent=yes
  - name: verificar que directorio existe
    file:
      path: "/path/to/directory"
      state: directory
      owner: root
      group: systemd-journal
      mode: 2755
    notify: reiniciar_journald
handlers:
  - name: reiniciar_journald
    service: name=systemd-journald state=restarted
```

3:28 Ficheros y OpenSSL (II)

stat

https://docs.ansible.com/ansible/latest/modules/stat_module.html#stat-module

- obligatorio
  - path = /path/al/fichero
- opcional
  - get_attributes = True / False
  - get_checksum = True / False
  - get_md5 = True / False
  - get_mime = True / False

```
- name: obtener datos de fichero
  stat: path="/path/to/file"
  register: datos_fichero
- name: mostrar información
  debug: var=datos_fichero
- name: en condicional
  debug: msg="es diretorio"
  when: datos.stat.isdir # o cualauier otro atributo
```

fetch

https://docs.ansible.com/ansible/latest/modules/fetch_module.html#fetch-module

- obligatorio
  - src : /path/to/file en nodo remoto
  - dest : server Ansible
- opcional
  - fail_on_missing = yes/no
  - flat = yes/NO : recrea la estructura de directorios de **src** en **dest**

```
- name: copiar configuración red
  fetch: src=/etc/network/interfaces dest=/tmp/backup
```

unarchive

https://docs.ansible.com/ansible/latest/modules/unarchive_module.html#unarchive-module

- obligatorio
  - src
  - dest
- opcional
  - owner
  - group
  - mode
  - remote_src = true / FALSE
  - list_files = yes / NO → lista los ficheros que se han extraído (se puede guardar en **registry**)

```
- name: copiar y extraer fichero en remoto
  unarchive: src=<file.tgz> dest=/opt/fichero
-name: extraer en remoto fichero ya existente allí
  unarchive: src=<file.tgz> dest=/opt/fichero remote_src=true
```

lineinfile (!)

https://docs.ansible.com/ansible/latest/modules/lineinfile_module.html#lineinfile-module

- obligatorio
  - line = «texto»
  - dest = /path/to/file
    - en versiones más modernas, **path**
- opcionales
  - owner
  - group
  - mode
  - backup = yes / NO
  - insertafter = REGEX
  - insertbefore = REGEX
  - regexp = REGEX
  - state = present / absent : ??

```
- name: deshabilita SELinux
  lineinfile: dest=/etc/selinux/config regexp='^SELINUX=' line='SELINUX=disabled'
- name: eliminar la línea que permite al grupo wheel del fichero de configuración
de sudoers
  lineinfile: dest=/etc/sudoers state=absent regexp="^wheel"
- name: añadir antes de una línea
  lineinfile: dest=/etc/apache2/ports.conf regexp='^Listen' insertafter="Listen 80"
line="Listen 8080"
```

blockinfile (!)

https://docs.ansible.com/ansible/latest/modules/blockinfile_module.html#blockinfile-module

- obligatorio
  - block = «texto»
  - dest = /path/to/file
- opcional
  - owner
  - group
  - mode
  - backup = yes / NO
  - insertafter = REGEX
  - insertbefore = REGEX
  - marker = REGEX
  - state = present / absent : ??

```
- name: asegurar que el texto está en el fichero
  dest: /etc/ssh/sshd_config
  block: |
    Match user monitor
```

```
    Password Authentication no
```

openssl_privatekey (!)

https://docs.ansible.com/ansible/latest/modules/openssl_privatekey_module.html#openssl-privatekey-module

- obligatorio
    - path = /path/to/file
- opcional
    - force = true / false
    - size = 4096
    - state = present / absent
    - type = RSA / DSA

```
- name: instalar módulo phyton requerido
  apt: name=phyton-openssl state=latest
- name: generar clave privada
  openssl_privatekey: path=/etc/ssl/private/private.pem
```

openssl_publickey (!)

https://docs.ansible.com/ansible/latest/modules/openssl_publickey_module.html#openssl-publickey-module

- obligatorio
    - path = /path/to/file
    - privatekey_path = /path/to/privatekey
- opcional
    - force = false / true
    - state = present / absent

```
- name: generar clave pública
  openssl_publickey:
    path: /etc/ssl/private/public.key
    privatekey_path: /etc/ssl/private/private.pem
```

3:29 Gestor Paquetes (I)

para lenguajes de programación:

- bower : desarrollo web
- bundler : dependencias Ruby Gem
- composer : librerias PHP
- cpam : gestor módulos Perl
- easy_install : gestión módulos / librerias Phyton
- gem : gestiona Ruby Gems
- maven_artifact : descarga **artifacts** desde un repositorio **maven**
- npm : gestiona paquetes node.js
- pear : paquetes pear / pcl
- pip : gestión módulos / librerias Phyton (más que easy_install)

para OS:

- apk : gestión paquetes android
- apt :
- apt_key:
- apt_repositiry
- dnf : fedora
- macports : paquetes macports OSX
- openbsd_pkg : paquetes openBSD
- opkg : paquetes OpenWRT (routers, firmware)
- package : módulo genérico que llama a los otros módulos (wrapper)
- pacman : paquetes arch linux
- pkg5 : paquetes Solaris 11
- pkgin : paquetes SmartOS, NetBsd y otros
- pkgng : paquetes FreeBSD >= 9.0
- portage : Gentoo
- redhat_subscription : administra repositorios y subscripciones Red Hat, usando el comando **subscription-manager**
- slackpkg : paquetes slackware >=12.2
- swdepot : paquetes HP_UX
- yum :
- yum_repository
- zypper : paquetes / repositorios OpenSuse / Suse
- zypper_respository

cpanm

PERL
https://docs.ansible.com/ansible/latest/modules/cpanm_module.html#cpanm-module

- from_path = ruta
- name = nombre
- localib = ruta
- mirror = mirror
- mirror_only = no / yes
- notest = no / yes
- version = version
- system_lib = directorio
- ~~state~~

```
- name: instalar gcc
  yum: name=gcc state=latest
- name: instalar paquete básico
  yum: name=perl-App-cpanminus state=latest
- name: instalar módulo DBI
  cpanm : name=DBI
- name: instalar versión específica
  cpanm: name=DBI version="1.360"
```

easy_install

https://docs.ansible.com/ansible/latest/modules/easy_install_module.html#easy-install-module

- obligatorio

    - name = nombre
- opcional
    - state = present | latest
    - virtualenv = no / yes
    - virtualenv_command = comando
    - virtualenv_site_packages = no / yes
    - exectutable = ruta ejecución easy_install

```
- name: instalar PiP
  easy_install: name=pip state=latest
# se usa PiP en lugar easy_install
```

pip

https://docs.ansible.com/ansible/latest/modules/pip_module.html#pip-module

- obligatorio
    - name = nombre
- opcional
    - state = present | latest | absent | forcereinstall
    - virtualenv = no / yes
    - virtualenv_command = comando
    - virtualenv_site_packages = no / yes
    - exectutable = ruta ejecución
    - requirements = fichero.txt (dependencias) → requirements.txt
    - version = version
    - chdir = ruta

```
- name: instalar módulo requests
  pip: name=requests state=latest
```

3:30 Gestor Paquetes (II)

apt

https://docs.ansible.com/ansible/latest/modules/apt_module.html#apt-module

- name = nombre[=versión]
- state = { latest | absent | PRESENT | build-dep }
    - latest : a la última
    - absent : eliminar
    - build-dep : dependencias
- upgrade = { no | yes | safe | full | dist }
- force = no / yes
- update_cache = no / yes
- purge = no / yes
- deb = ruta/fichero/.deb
- autoremove = no / yes
- default_release = release

```
- name: actualizar lista paquetes
```

```
    apt: update_cache=yes
- name: actualizar paquetes
   apt: upgrade=dist
- name: instalar nginx
  apt:
    name: nginx
    state: latest
```

apt_key

https://docs.ansible.com/ansible/latest/modules/apt_key_module.html#apt-key-module

- data = contenido de la key a añadir (desde Ansible Server)
- file = ubicación fichero en nodo remoto
- id = identificador
- keyring = /ruta/trusted
- keyserver = servidor
- state = { PRESENT | absent }
- url = dirección
- validate_certs = yes / no

```
- name: añadir clave usando servidor
  apt_key:
    keyserver: keyserver.ubuntu.com
    id: 36A1D7869245C8950F...
- name: añadir utilizando un fichero adjunto
  apt_key:
    url: "https://ftp-master.debian.org/keys/archive-key-6.0.asc"
    state: present
```

3:31 Gestor Paquetes (III)

apt_repository

https://docs.ansible.com/ansible/latest/modules/apt_repository_module.html#apt-repository-module

- obligatorio
    - repo = origen
- opcional
    - state = { PRESENT | absent }
    - filename = nombre fichero repositorio
    - update_cache = yes / no
    - validate_certs = yes / no
    - mode = modo_fichero

```
- name: anyadir repositorio google chrome
  apt_repository:
    repo: "deb http://dl.google.com/linux/chrome/deb/ stable main"
    state: present
    filename: "google-chrome"
- name: anaydir en Ububtu a través de PPA
  apt_repository:
```

```
    repo: "ppa:nginx/stable"
```

package

wrapper, usar si no requerimos alguna opción concreta de otro módulo de paquetes
https://docs.ansible.com/ansible/latest/modules/package_module.html#package-module

- requerido
  - name = origen
  - state = { present | absent | latest }
- opcional
  - use = { auto | yum | apt }

```
- name instalar ntpdate
  package:
    name: ntpupdate
    state: latest
```

</code>

redhat_subscription

https://docs.ansible.com/ansible/latest/modules/redhat_subscription_module.html#redhat-subscription-module

- state = { present | absent }
- activationkey
- username
- password
- autosubscribe = yes / no
- server_hostname = nombre servidor
- org_id = organización
- pool = nombre
- force_register = yes / no

```
- name: registrar sistema
  redhat_subscription:
    state: present
    username: usuario@dominio
    password: contraseña
    autosubscribe: yes
- name: registrar sistema 2
  redhat_subscription:
    state: present
    activationkey: mi-clave-RHEL
    org_id: 2468
    pool "^Red Hat Enterprise Server$"
```

yum

https://docs.ansible.com/ansible/latest/modules/yum_module.html#yum-module

- requerido
    - name = nombre / ruta
- opcional
    - state = { present | absent | latest }
    - conf_file = /ruta/al/fichero
    - disable_gpg_check = true / false
    - disablerepo = nombre (desactiva temporalmente)
    - enablerepo = nombre (activa temporalmente)
    - update_cache = yes / no

```
- name: instalar última versión apache
  yum:
    name: httpd
    state: latest
- name: actualizar todos los paquetes
  yum:
    name: "*"
    state: latest
- name: Instalar grupo
  yum:
    name: "@development tools"
    state: present
```

yum_repository

https://docs.ansible.com/ansible/latest/modules/yum_repository_module.html#yum-repository-module

- requerido
    - name = nombre / ruta
- opcional
    - state = { present | absent }
    - description = descripción
    - baseurl = dirección
    - file = nombre_fichero
    - mirrorlist = dirección
    - enabled = YES / no
    - gpgcheck = YES / no

```
- name: añadir EPEL
  yum_repository:
    name: epel
    state: present
    description: EPEL YUM Repo
    baseurl: http://download.fedoraprojects.org/pub/epel/$releaseserver/$basearch/
```

3:32 Comando Utilidades (I)

ejecución de comandos en el nodo remoto

- command
- expect : ejecuta un comando y responde a la introducción de dataos
- raw : envía comandos sin filtrar por SSH
- script : transfiere y ejecuta un script

- shell : permite uso de && || » (command no lo permite)

command

https://docs.ansible.com/ansible/latest/modules/command_module.html#command-module

- chdir : cambiar directorio ejecución
- creates : si existe el fichero, NO ejecuta
- executable : ruta binario
- removes : si no existe el fichero, NO ejecuta

```
- name: obtener uname
  command: uname -a
  register: salida_uname
- name: crear base de datos si no existe
  command: /sbin/createdb.sh
  args:
    chdir: /var/lib/mysql
    creates: /basededatos/existe
- name: ejecutar si existe
  command uname -a removes=/tmp/hadeexistir
```

expect

https://docs.ansible.com/ansible/latest/modules/expect_module.html#expect-module

- requerido
  - command = comando
  - response = respuestas
- opcional
  - chdir
  - creates
  - removes
  - echo
  - timeout

```
- name: instalar pexpect, necesario en el nodo remoto
  yum: name=pexpect state=latest
- name: cambiar contraseña usuario
  expect:
    command: passwd usuario
    responses: (?i)password: "SuperSecreta"
```

instalar versión específica (en módulo Phyton): `pip install pexpect==3.3`

3:33 Comando Utilidades (II)

raw

usar en casos muy concretos (dispositivos sin Phyton o versiones viejas), viaja sin encriptar

https://docs.ansible.com/ansible/latest/modules/raw_module.html#raw-module

- executable = /ruta/ejecutable

```
- name: actualizar paquetes e instalar uno
  raw: apt-get update && apt-get install vim
```

script

copia el script en el nodo y se ejecuta allí
https://docs.ansible.com/ansible/latest/modules/script_module.html#script-module

- creates = /fichero/comprobar ← si existe, no ejecuta
- removes = /fichero/comprobar ← si no existe, no ejecuta
- decrypt = true / false ← vault

```
- name: copia y ejecuta el script
  script: /mi/fichero/local.sh argumentos
```

shell

ejecuta comandos, pero permite (**command** no lo hace) el uso de tuberías, redirecciones, etc...
https://docs.ansible.com/ansible/latest/modules/shell_module.html#shell-module

- chdir = /directorio
- creates = /fichero/comprobar ← si existe, no ejecuta
- removes = /fichero/comprobar ← si no existe, no ejecuta
- executable

```
- name: obtener uname
  shell: uname -a | tee fichero.log
  register: salida_uname
- name obtener uname 2
  shell: uname -a | tee fichero.log
  args:
    executable: /bin/bash
    chdir: /tmp
```

assert

asegurarse que se cumplen ciertas condiciones
https://docs.ansible.com/ansible/latest/modules/assert_module.html#assert-module

- that : condiciones
- msg : mensaje a mostrar

```
- vars:
  - numero: 50
- assert:
    that:
      - numero <= 100
      - numero > 0
    msg: "Número ha de estar entre 0 y 100"
```

debug

muestra un texto personalizado o el valor de una variable
https://docs.ansible.com/ansible/latest/modules/debug_module.html#debug-module

- msg = «mensaje de texto»
- var= variable
- verbosity = [0-3]

```
- debug: msg="Hostname {{ ansible_hostname }}"
- debug: var=salida # muestra todo el array, se puede especificar cualquiera de ellos
```

pause

https://docs.ansible.com/ansible/latest/modules/pause_module.html#pause-module

- prompt = «texto a mostrar»
- minutes = minutos
- seconds = segundos

fail

generar mensaje error y salir
https://docs.ansible.com/ansible/latest/modules/fail_module.html#fail-module

- msg = «mensaje»

```
- fail: msg="Dato incorrecto"
  when: valor not in ['y','Y']
```

3:34 Comando Utilidades (III)

include

incluir otro playbook / tareas
https://docs.ansible.com/ansible/latest/modules/include_module.html#include-module

/roles/%%<rol

```
%%/tasks/main.yml>
include: name="configura.yml"
include: name="install.yml"
include: name="post-install.yml"
```

include_role

incluir rol
https://docs.ansible.com/ansible/latest/modules/include_role_module.html#include-role-module

- obligatorio
  - name
- opcional
  - private = true / false
  - tasks_from = main
  - vars_from = main
  - defaults_from = main
  - allow_duplicates = true / false

include_vars

incluir variables desde un fichero, en lugar de tener una sección **vars:**
https://docs.ansible.com/ansible/latest/modules/include_vars_module.html#include-vars-module

playbook

```
---
- hosts: localhost
- vars:
  - numero: 10
- tasks:
  - include_vars: variables.yml
  - include_rol: name=httpd # en directorio local o en /etc/ansible/roles o en /root/roles
```

variables.yml

```
otronumero: 20
```

self_fact

establece un fact
https://docs.ansible.com/ansible/latest/modules/set_fact_module.html#set-fact-module

permite modificar textos / variables

```
- tasks:
  - set_fact: nombre="{{ ansible_hostname }}"
  - debug: var=nombre
  - set_fact: nombre2="{{ ansible_hostname | upper }}"
```

wait_for

espera que se cumpla una condición para continuar (conexión SSH o si un fichero existe, por ejemplo)

Enllaç externhttps://docs.ansible.com/ansible/latest/modules/wait_for_module.html#wait-for-module]]

- state = { present | absent | started | stopped }
  - files: present/absent, puertos: started/stopped
- port = puerto
- timeout = segundos de espera a conectar si no hay respuesta
- host = servidor al que conectar
- connection_timeout =
- search_regex = cadena esperada (ya sea conexión o fichero)
- delay = segundos de espera antes de empezar a enviar peticiiones
- path = /ruta/del/fichero
- exclude_hosts

```
- hosts: localhost
  tasks:
    - name: esperando al puerto 8080
      wait_for: port=8080 delay=2
    - name: esperando a que exista el fichero
      wait_for: path=/fichero/esperado
    - name: esperando a que no exista el fichero
      wait_for: path=/fichero/espeado state=absent
```

```
- name: esperando a server
  wait_for:
    port: 22
    host: "{{ ansible_hostname ]]"
    search_regex: OpenSSH
    delay: 10
    delegate_to: localhost # ???
```

```
nc -l 8080
```

3:35 Notificaciones

sistemas de mensajería:

- cisco_spark
- flowdock
- hipchat
- irc
- jabber
- mattermost : tipo **slack** pero de código abierto
- mqt : mensajería IoT
- nexmo : SMS
- pushbullet : móbiles
- pushover : móbiles
- rocketchat
- sendgrid
- slack
- sns : Simple Notification Service Amazon
- telegram
- twilo

hipchat

https://docs.ansible.com/ansible/latest/modules/hipchat_module.html#hipchat-module

- requeridos
  - token
  - msg
  - room
- opcional
  - api
  - color
  - from
  - msg_format = { text | html }
  - notify = yes / no
  - validate_certs = YES / no

```
---
- hosts: localhost
  connection: localhost
  vars:
    - notificar: "hipchat"
  tasks:
    - hipchat:
        api: https://api.hipchat.com/v2/
        token: "..."
        room: destinatario
        msg: "Tarea finalizada"
      when: notificar == "hipchat"
    - mail:
        subject: "Tarea finalizada"
      delegate_to: localhost
      when: notificar == "mail"
    - pip: name=pushbullet.py
      when: notificar == "pushbullet"
```

mail

https://docs.ansible.com/ansible/latest/modules/mail_module.html#mail-module

- requerido
  - subject
- opcional
  - host
  - port
  - user
  - password
  - to
  - body
  - cc
  - bcc
  - secure = { always | never | try | starttls }

```
---
- hosts: localhost
  connection: localhost
```

```
  vars:
    - notificar: "mail"
  tasks:
    - mail:
        subject: "Tarea finalizada"
        host: servidor.correo
        port: 25
        to: alberto@correo
      delegate_to: localhost
      when: notificar == "mail"
```

pushbullet

https://docs.ansible.com/ansible/latest/modules/pushbullet_module.html#pushbullet-module

- requerido
    - api_key
    - title
- opcional
    - body
    - channel
    - device
    - push_type

```
---
- hosts: localhost
  connection: localhost
  vars:
    - notificar: "pushbullet"
  tasks:
    - name: instalar pushbullet.py
        pip: name=pushbullet.py state=latest
        when: notificar == "pushbullet"
    - name: enviar notificación
        pushbullet:
          api_key: <clave>
          device: <dispositivo>
          title: "Notificación Ansible"
        when: notificar == "pushbullet"
```

pushover

https://docs.ansible.com/ansible/latest/modules/pushover_module.html#pushover-module

- requeridos
    - app_token
    - user_key
    - msg
- opcional
    - pri = prioridad

rocketchat

https://docs.ansible.com/ansible/latest/modules/rocketchat_module.html#rocketchat-module

- requeridos:
  - token
  - domain
- opcional
  - mdg
  - channel
  - username
  - color = { normal | good | warning | danger }
  - protocol = { https | http }
  - validate_certs = true /false

slack

https://docs.ansible.com/ansible/latest/modules/slack_module.html#slack-module

- requerido
  - token
- opcional
  - msg
  - channel = #canal
  - username
  - color = { normal | good | warning | danger }
  - validate_certs = true / false

3:36 Bases Datos

- mysql:
  - mysql_db : añade o elimina BBDD
  - mysql_replication : administra replicación
  - mysql_user : administra usuarios
  - mysql_variables : administra variables globales
- postgresql
  - postgres_db : añade o elimina BBDD
  - postgres_ext : administra extensiones
  - postgres_lang : administra procedimientos almacenados
  - postgres_privs : administra privilegios
  - postgres_schema : administra esquemas
  - postgres_user : administra usuarios
- MongoDB
  - mongodb_parameter : gestionar parámetros
  - mongodb_user : administrar usuarios
- Influxdb
  - influxdb_database: administrar BDDD
  - retention_policy: administrar políticas de retención
- Vertica (HPE)
- Miscelanea
  - elasticsearch_plugin
  - kibana_plugin
  - redis

    - riak

mysql_db

- requiere
    - name = nombre BDD
- opcional
    - state = { present | absent | dump | import }
    - login_host
    - login_password
    - login_port
    - login_user
    - login_unix.socket
    - encoding
    - collation : (idioma) es_ES.UTF8
    - target

```
- name: Instalar libreria requerida
  pip: name=pytho_mysql state=latest

- name: crear si no existe la BDD
  mysql_db:
    name: <bdd>
    state: present
- name: copia de seguridad todas las BDD
  mysql_db:
    state: dump
    name: all # palabra clave
    target: /tmp/{{ ansible.hostname }}.sql
```

mysql_user

- requerido
    - name = nombre
- opcional
    - state = { present | absent }
    - password
    - encrypted = no / yes
    - login_host
    - login_password
    - login_port
    - login_user
    - login_unix.socket
    - priv = dbtabla:priv1,priv2 ← privilegios
    - append_privs = yes / no ← añadir o sustituir

```
- name: crear usuario y darle permisos
  mysql_user:
    name: <nombre>
    password: <password>
    state: present
    priv: "<bdd>.*:ALL"
```

postgres_db

- requerido
    - name = nombreBDD
- opcional
    - state = { present | absent }
    - login_host
    - login_password
    - port
    - login_user
    - login_unix.socket
    - encoding
    - lc_collate
    - template

```
- name: instalarlibrería requerida
  pip: name=pstcopg2 state=latest
- name: crear si no existe
  postgresql_db:
    name: <nombre_bdd>
    state: present
    encoding: utf-8
    become_user: postgres
```

postgres_user

- requerido
    - name = usuaroi
- opcionales
    - state = { present | absent }
    - login_host
    - login_password
    - port = 5432
    - login_user
    - login_unix.socket
    - password
    - encrypted = yes / no
    - priv = tabla:privilegio
        - role_attr_flags
            - (NO)SUPERUSER
            - (NO)CREATEROL
            - (NO)CREATEUSER
            - (NO)CREATEDB
            - (NO)INHERIT
            - (NO)LOGIN
            - (NO)REPLICATION
    - db

```
- name: crear si no existe
  postgresql_user:
    db: <base de datos>
    name: <usuario>
    state: present
    password: <password>
```

```
    priv: ALL
```

mongodb_user

- requerido
  - name = usuario
  - database = nombre
- opcional
  - state = { present | absent }
  - password = contraseña usuario
  - login_host
  - login_password
  - login_port = 27017
  - login_user
  - roles
    - READWRITE
    - read
    - dbAdmin
    - userAdmin
    - clusterAdmin
    - ...

```
- name: instalar librería requerida
  pip: name=pymongo state=latest
- name: crear usuario
  mongodb_user:
    database: admin
    name: <usuario>
    password: <password>
    state: present
```

3:37 Gestionar Sistema (I)

- alternatives : gestionar alternativas (versiones) para comandos
  - `/etc/alternatives/java`
- at : programar ejecución de comandos
- authorized_keys : gestión ficheros claves
- cron : gestión de cron
- crypttab : cifrado de dispositivos
- filesystem : sistema de ficheros
- firewald : equivalente en CentOS/RedHat de **iptables**
- gluster-volume : gestión de volumenes GlusterFS
- group : gestión de grupos
- hostname : gestión nombre servidor
- iptables : gestión reglas firewall
- known_hosts : gestsionar claves de otros servidores
- lvg : LVM
- lvol : LVM
- mount : montaje de FS
- open_iscsi : gestión dispositivos icsci
- openwrt_init : gestionar servicios OpenWRT

- pam_limits : gestión límites PAM
- pamd : gestión módulos PAM
- ping : comprobar conexión
- seboolean
- selcontext
- selinux
- selinux_permisive
- seport : soporte SELinux
- service : gestión servicios
- setup : información del sistema
    - gather_facts: false
- sysctl : configurar /etc/sysctl/conf
- systemd : gestión de servicios
- timezone : zonas horarias
- user: gestión usuarios

alternatives

gestión de **/etc/alternatives**
https://docs.ansible.com/ansible/latest/modules/alternatives_module.html#alternatives-module

- requerido
    - name = nombre
    - path = /ruta/al/fichero
- opcional
    - link = /ruta/al/fichero
    - priority = 50

```
- name: fijar la versión de Java a 8
  alternatives:
    name: java
    path: /usr/lib/jvm/java-8-openjdk-amd64/bin/java
```

`/usr/bin/java` → `/etc/alternatives/java` → `/usr/lib/jvm/java-8-openjdk-amd64/bin/java`

authorized_keys

https://docs.ansible.com/ansible/latest/modules/authorized_key_module.html#authorized-key-module

- requerido
    - user = <usuario>
    - key = <clave.ssh>
- opcional
    - state = { present | absent }
    - path = ~/.ssh/authorized_keys
    - manage_dir = yes / no ← crea carpeta / fichero y ajusta permisos
    - key_options = opciones
    - exclusive = no / yes ← si ya existe no se añade

```
- name: autorizar clave pública
  authorized_keys:
    user: <user>
    key: "..."
```

cron

https://docs.ansible.com/ansible/latest/modules/cron_module.html#cron-module

- name = nombre
- job = comando
- state = { present | absent }
- minute = [0 - 59]
- hour = [0 - 23]
- weekday = [0-6]
- month = [1-12]
- day = [1-31]
- special_time = { reboot | yearly | annualy | monthly | weekly | daily | hourly }
- cron_file = nombre ← crea el fichero en **/etc/cron.d/**
- backup = yes / no

3:38 Gestionar Sistema (II)

filesystem

https://docs.ansible.com/ansible/latest/modules/filesystem_module.html#filesystem-module

- requerido
  - dev = dispositivo
  - fstype = sistema de ficheros
- opcionales
  - force = no / yes
  - opts = opciones
  - resicefs = no /yes

```
- name: crear FS xfs
  filesystem:
    dev: /dev/sdX1
    fstype: xfs
```

firewalld

control del firewall en sistemas CentOS/RH
https://docs.ansible.com/ansible/latest/modules/firewalld_module.html#firewalld-module

- requerido
  - state = { enabled | disabled }
  - permanent = true / false ← hace falta **+reload** (handler)
- opcional
  - servicio
  - zone
  - port
  - source
  - rol_rule
  - immediate = false / true

```
- name: permitir acceso http/https
  firewalld:
    state: present
    service: "{{ item }}"
    permanent: true
  with_items:
    - "http"
    - "https"
```


```
- name: permitir acceso http/https
  firewalld:
    state: present
    service: "{{ item }}"
    permanent: true
  with_items:
    - "http"
    - "https"
```


group

https://docs.ansible.com/ansible/latest/modules/group_module.html#group-module

- requerido
  - name = nombre
- opcional
  - state = { present | absent }
  - gid = idgrupo
  - system = yes / no

```
- name: crear grupo para aplicación
  group:
    name: jboss
    state: present
    gid: 185
```

hostname

https://docs.ansible.com/ansible/latest/modules/hostname_module.html#hostname-module

- name = nombre

```
- name: cambiar el nombre del servidor
  hostname:
    name: servidor.dominio.com
```

ping ?

https://docs.ansible.com/ansible/latest/modules/ping_module.html#ping-module

```
- name: comprobar conexión
  ping:
```

timezone

https://docs.ansible.com/ansible/latest/modules/timezone_module.html#timezone-module

- hwclock = true / false
- name = Area/Ciudad

```
- name definir huso horario
  timezone:
    name: Europe/Barcelona
```

3:39 Gestionar Sistema (III)

iptables

https://docs.ansible.com/ansible/latest/modules/iptables_module.html#iptables-module

- state = { present | absent }
- chain = { INPUT | FORWARD | OUTPUT | PREROUTING | POSTROUTING | SECMARK | CONNSEMARK }
- source = dirección
- jump = { ACCEPT | DROP | ... }
- in_interface
- out_interface
- protocol = { tcp | udp | icmp }
- destination_port = puerto
- to_ports = puerto
- cstate = { INVALID | NEW | ESTABLISHED | RELATED | UNTRACKED | SNAT | DNAT }

```
- name: permitir acceso puerto 80
  iptables:
    chain: INPUT
    source: 0.0.0.0
    destination_port: 80
    jump: ACCEPT
    protocol: tcp
```

lvg

https://docs.ansible.com/ansible/latest/modules/lvg_module.html#lvg-module

- requerido
  - vg = grupo de volumen
- opcional
  - state = { present | absent }
  - pvs = /dev/vbX
  - pesize = 4 (Mb)
  - vg_options = opciones vgcreate
  - force = yes / no

```
- name: crear grupo de volúmenes
```

```
  lvg:
    vg: datavg
    pvs: /dev/vda1
    state: present
```

vgs vgdisplay pvdisplay

lvol

https://docs.ansible.com/ansible/latest/modules/lvol_module.html#lvol-module

- requerido
  - vg = nombre VG existente
  - lv = nombre volumen lógico
- opcional
  - state = { present | absent }
  - size = tamaño
  - pvs = /dev/vg1
  - opts = opciones
  - active = yes / no
  - force = yes / no

```
- name: creación volumen lógico
  lvol: vg=datavg lv=web size=2G state=present
```

lvscan

mount

https://docs.ansible.com/ansible/latest/modules/mount_module.html#mount-module

- requerido
  - name =
  - state = { present | absent | mounted |unmounted }
- opcional
  - pstype = tipo FS
  - opts = opciones
  - src = dispositivo
  - dump = 0
  - passno = 0

```
- name: montar partición WWW
  - filesystem: dev=/dev/datavg/web fstype=xfs # formatea
  - mount: src=/dev/datavg/web name=/var/www fstype=xfs # monta
```

3:40 Gestionar Sistema (IV)

service

https://docs.ansible.com/ansible/latest/modules/service_module.html#service-module

- requerido

        - name
    - opcional
        - state = { started | stopped | restarted | reloaded }
        - enabled = yes / no
        - arguments = argumentos
        - sleep = segundos

```
- name: iniciar y habilitar servicio
  service:
    name: apache2
    state: started
    enabled: true
```

setup

https://docs.ansible.com/ansible/latest/modules/setup_module.html#setup-module

- fact_path = /etc/ansible/fact.d ← información propia que almacenamos en ese path
- filter = *
- gather_subset = { all | hardware | network | virtual }
- gather_timeout = 10

```
- name: obtiene facts
  setup
   gather_subset: all
```

sysctl

cambios en el kernel
https://docs.ansible.com/ansible/latest/modules/sysctl_module.html#sysctl-module

- requerido
    - name
- opcional
    - value = valor
    - state { PRESENT | absent }
    - reload = yes / no ← `sysctl -p`
    - systcl_file = /etc/sysctl.conf
    - sysctl_set = yes / no ← establecer valores
    - ignoreerrors = no / yes

```
- name: permitir redirigir el tráfico
  sysctl:
    name: net.ipv4.ip_forward
    value: 1
    sysctl_set: yes
    state: present
    reload: yes
```

```
sysctl -a | grep ip_forward
```

systemd

https://docs.ansible.com/ansible/latest/modules/systemd_module.html#systemd-module

- name = nombre
- state = {started | stopped | restarted | reloaded }
- enabled = yes / no
- daemon_reload = no /yes
- masked = yes / no

```
- name: habilitar servicio y recargar systemd
  systemd:
    name: apache2
    enabled: yes
    state: started
    daemon_reload: yes
```

user

https://docs.ansible.com/ansible/latest/modules/user_module.html#user-module

- requerido
    - name = nombre
- opcional
    - state = { PRESENT | absent }
    - group = grupo
    - groups = grupo1,grupo2
    - append = yes / no ← añadir a los grupos existentes
    - createhome = yes / no
    - uid = id_usuario
    - home = directorio
    - shell = /bin/sh
    - password = clave
    - remove = yes / no ← eliminar direcotorio al hacer un **absent**
    - system = yes / no

```
- name: crear usuario
  user:
    name: pepito
    id: 1001
    home: /home/pepito
    shell: /bin/false
    state: present
```

3:41 Windows (I)

- win_acl : permisos ficheros/directorios, a usuario/grupos
- win_chocolatey : repositorio paquetes
- win_command : ejecución comandos
- win_copy : copiar fichero a nodos remotos
- win_environment : modificar variables entorno
- win_feature : activar/des características
- win_file : crear/eliminar ficheros/directorios

- win_get_url: descargar fichero de URL
- win_group: gestión grupos locales
- win_lineinfile: verifica la existencia de una línea en un fichero
- win_msi: instalar o desinstalar paquetes MSI
- win_package : idem, pero fichero local o URL
- win_ping : ping
- win_reboot : reiniciar
- win_regedit : gestión registro
- win_schedule_task : gestión tareas
- win_service : gestión de servicios
- win_share : gestión puntos de montaje
- win_shell : comandos de consola
- win_stat : información de un fichero
- win_template : copia y procesa plantillas
- win_timezone : establecer zona horaria
- win_unzip : descomprimir fichero
- win_updates : descargar e instalar actualizaciones
- win_uri : interactuar con servicios web
- win_user: administrar usuarios locales

win_command

https://docs.ansible.com/ansible/latest/modules/win_command_module.html#win-command-module

- chdir = directorio
- creates = c:\ruta\fichero
- removes = c:\ruta\fichero

```
-name: quien soy
 hosts: windows01
 tasks:
  - win_command: whoami
    register: usuario
  - debug: var=usuario
```

win_copy

https://docs.ansible.com/ansible/latest/modules/win_copy_module.html#win-copy-module

- requerido
    - src = /fichero/a/copiar
    - dest = c:\path\destino

```
...
- name: copiar fichero
  win_copy:
    src: config.txt
    dest: c:\config.txt
```

win_file

https://docs.ansible.com/ansible/latest/modules/win_file_module.html#win-file-module

- requerido
  - path = /ruta
- opcional
  - state = { file | directory | touch | absent }

```
- name: crear estructura de directorios
  win_file:
    path: c:\users\alberto\Documents\apache2\conf
    state: directory
```

win_lineinfile

https://docs.ansible.com/ansible/latest/modules/win_lineinfile_module.html#win-lineinfile-module

- requerido
  - path / dest (para >2.3)
- opcional
  - state = { present | absent }
  - line = línea
  - newline = { windows | unix } → \r\n o \n
  - insertafter = EOF / REGEX
  - insertbefore = BOF / REGEX
  - create = no / yes
  - backup = no / yes
  - regexp = REGEX

```
...
- name: editar puerto
  win_lineinfile:
    path: c:\httpd.conf
    state: present
    line: Listen 8080
    regexp: "^Listen"
```

3:42 Windows (II)

win_service

https://docs.ansible.com/ansible/latest/modules/win_service_module.html#win-service-module

- requerido
  - name = nombre
- opcional
  - state = { started | stopped | restarted }
  - start_mode = { auto | manual | disabled | delayed }

```
- name: reiniciar servicio
  win_service:
    name: spooler
    start_mode: manual
    state: stopped
```

win_shell

https://docs.ansible.com/ansible/latest/modules/win_shell_module.html#win-shell-module

- chdir
- creates : c:\path\fichero ← ejecutar si el fichero existe
- executable
- removes : c:\path\fichero ← ejecutar si el fichero no existe

```
name: ejecutar script
  win_shell: c:\script.ps1
    args:
      chdir: c:\
```

win_template

https://docs.ansible.com/ansible/latest/modules/win_template_module.html#win-template-module

- requerido
    - src
    - dest

j2 = jinga2

```
- name: copiar plantilla
  win_template:
   src: info.j2
   dest: c:\info.txt
```

win_user

https://docs.ansible.com/ansible/latest/modules/win_user_module.html#win-user-module

- requerido
    - name = usuario
- opcional
    - account_disabled = no / yes
    - account_locked = no / yes
    - description = descripción
    - fullname
    - groups
    - group_action = { replace | add | remove }
    - password
    - password_expired = yes / no
    - password_never_expires = yes / no
    - state = { present | absent | query }
    - update_password = { always | on_create }
    - user_cannot_change_password = no / yes

```
-name: crear usuario
  win_user:
    name: oforte
```

```
    password: oforte123
    state: present
    groups:
      - users
```

3:43 Control Versiones

- bzr
- git
- git_config
- github_hooks
- github_keys
- github_release
- gitlab_group
- gitlab_project
- gitlab_user
- hg
- subversion

git

https://docs.ansible.com/ansible/latest/modules/git_module.html#git-module

- requerido
    - repo = direccoión
    - dest = /directorio
- opcional
    - version
    - update = yes / no
    - remote = origin
    - recursive = yes / no
    - force = no / yes
    - accept_hostkey = no / yes

```
...
-name: obtener ejemplo
  git:
    repo: https://github.com/ansible/ansible-examples.git
    dest: /root/ansible_examples/
```

git_config

https://docs.ansible.com/ansible/latest/modules/git_config_module.html#git-config-module

- name = nombre_calve
- value = valor
- scope = { local | global | system }
- list_all = yes / no
- repo = ruta/repo

```
-name: usar vim como editor
  git_config:
    name: core.editor
```

```
    value: vim
    scope: global
```

hg

https://docs.ansible.com/ansible/latest/modules/hg_module.html#hg-module

- requerido
    - repo = dirección
    - dest = /path/
- opcional
    - force = yes / no
    - purge = no / yes
    - revision = version
    - update = yes / no

```
-name: obtener código
  hg:
    repo: https://www.selenic.com/repo/hello
    dest: /home/user/hg/
```

subversion

https://docs.ansible.com/ansible/latest/modules/subversion_module.html#subversion-module

- requerido
    - repo = dirección
    - dest = /directorio/
- opcional
    - force = no / yes
    - username = usuario
    - password = contraseña
    - revision = version
    - switch = yes / no ← cambio de rama

```
-name: obtener código de WP
 subversion:
   repo: https://develop.svn.wordpress.org/trunk/src/
   dest: /home/users/wp
```

3:44 Infraestructura Web / messaging

web:

- apache2_mod_proxy : establecer atributos
- apache2_module : gestión módulos (debian/ubuntu, requiere **a2enmod** y **a2dismod**)
- deploy_helper :
- djongo_manage :
- ejabber_user :
- htpasswd : gestion autentificación básica de usuarios

- jboss :
- jenkins_job :
- jenkings_plugin :
- jira : añade o modifica incidencias en Jira
- letsencrypt : crear certificados SSL usando Let's Encrypt
- supervisorctl : gestion programa(s) usando supervisord
- taiga_issue : gestión incidencias en la plataforma Taiga

messaging:

- rabbitmq_*

apache2_module

https://docs.ansible.com/ansible/latest/modules/apache2_module_module.html#apache2-module-module

- requerido
  - name = módulo
- opcional
  - state = { PRESENT / absent }
  - force = yes / no

```
-name: Habilitar móudlo wsgi
  apache2_module:
    name: wsgi
    state: present
```

htpasswd

https://docs.ansible.com/ansible/latest/modules/htpasswd_module.html#htpasswd-module

- requerido
  - name = módulo
  - path = /ruta/fichero/htpasswd
- opcional
  - state = { PRESENT / absent }
  - password = clave
  - crypt_scheme = { APR_MD5_CRYPT | des_crypt | ldap_sha1 | plaintext }
  - create = yes/no ← crear el fichero si no existe
  - mode = modo del fichero
  - owner: <usuario>

```
- name: instalar libreria necesaria passlib
  apt: name=python-passlib state=present
- name: añadir usuario
  htpasswd:
    name: oforte
    path: /var/www/html/.htpasswd
    password: test123
  delegate_to: localhost
```

```
AuthUserFile /var/www/html/.htpasswd
AuthGroupFile /dev/null
AuthName "Please enter password"
AuthType Basic
```

```
Require valid-user
```

supervisorctl

https://docs.ansible.com/ansible/latest/modules/supervisorctl_module.html#supervisorctl-module

- requerido
  - name = módulo
  - state = { present | started | stopped | restarted | absent }
- opcional
  - config = /ruta/al/fichero
  - password
  - server_url = http://localhost:9001
  - username
  - supervisorctl_path = /ruta/socket

```
-name: detener aplicación
  supervisorctl:
    name: long_script
    state: stopped
```

3:45 Cloud / Cluster

https://docs.ansible.com/ansible/latest/modules/list_of_clustering_modules.html
https://docs.ansible.com/ansible/latest/modules/list_of_cloud_modules.html

- amazon
  - cloudformation
  - cloudtrail
  - dynamodb
  - ec2
  - elasticache
  - route53
  - s3
  - sns/sqs/sts
- azure
  - network
    - interfaces
    - public ip
    - subnets
    - virtual networks
  - resource groups
  - security groups
  - storage accounts
  - virtual machines
- Docker
  - contenedor
  - imágenes
  - network
  - servicios
- Google

    - instancias
    - storage
    - DNS
    - load balancer
    - redes firewall
    - tags
    - backend service
- OpenStack
    - ...
- VMWare
    - ...