gestgionar el sistema

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
```