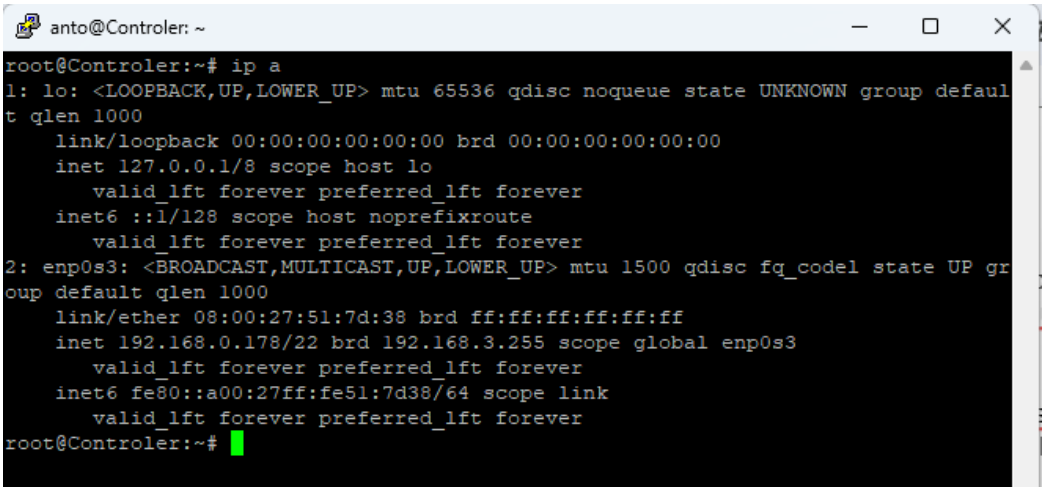
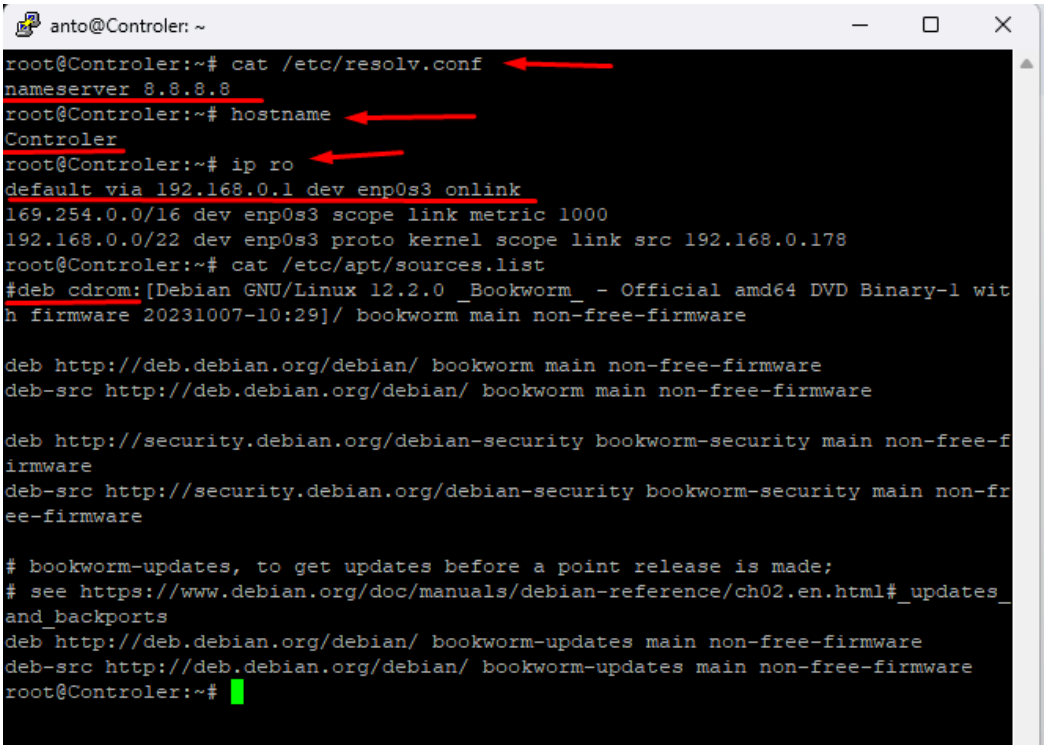
Setup (16 punts)

Ves a la diapositiva del setup laboratori feta a teoria i posa les configuracions a les 3 màquines que es demanen, tria si vols que els teus linux siguin GUI o CLI. Com a windows farem servir un windows10.

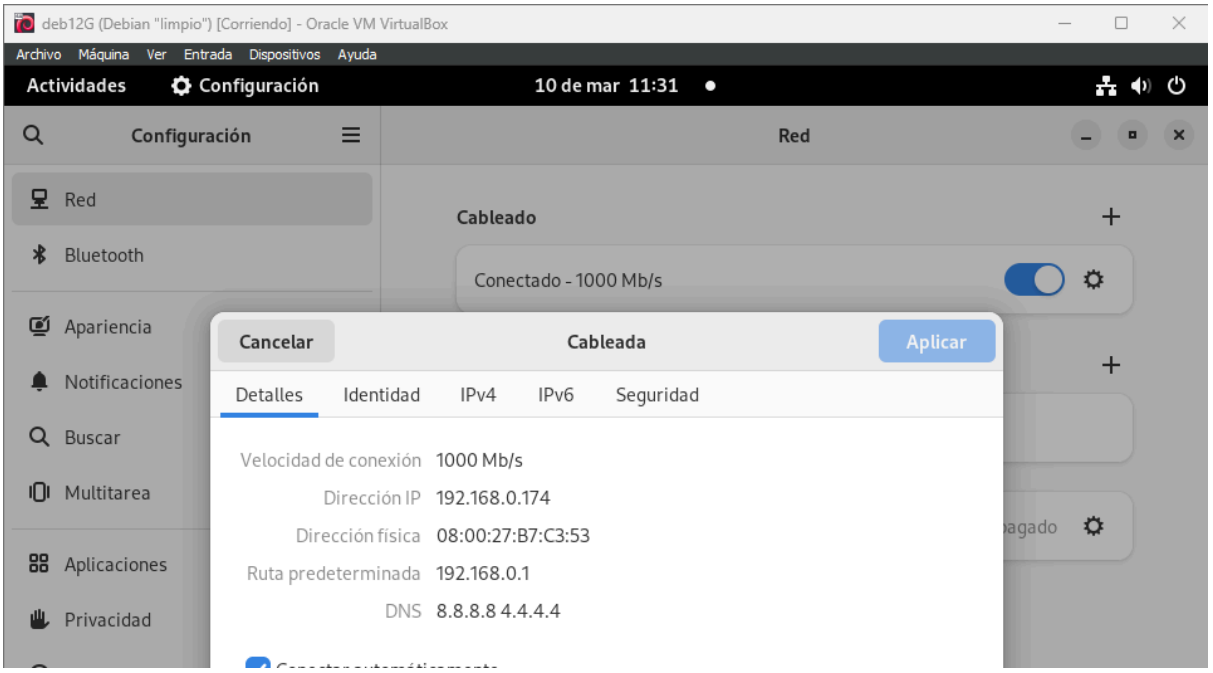
Adjunta la captura de les IPs (ip a), DNSs (cat /etc/resolv.conf), gateway (ip ro), nom de màquina, i cat /etc/apt/sources.list en el cas de Linux. Posa les captures corresponents de windows també. **(7 punts)**

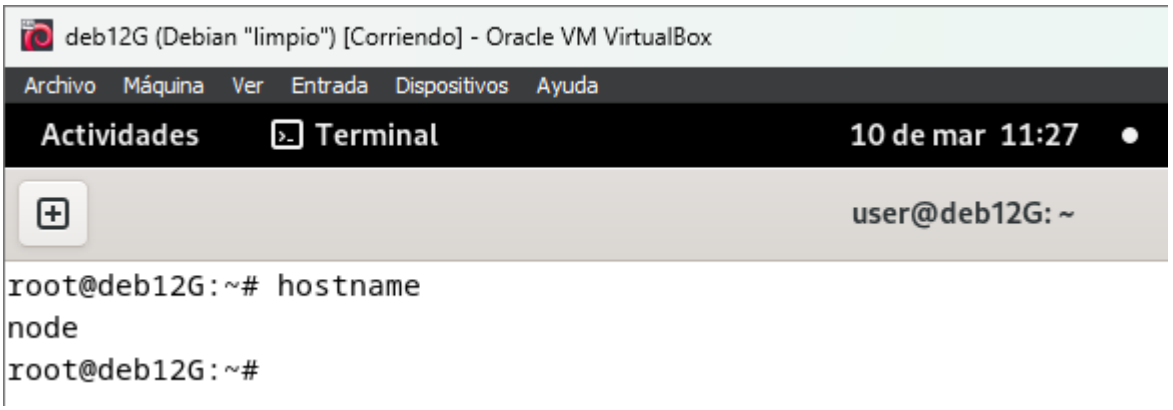
Debian Server (Controler)



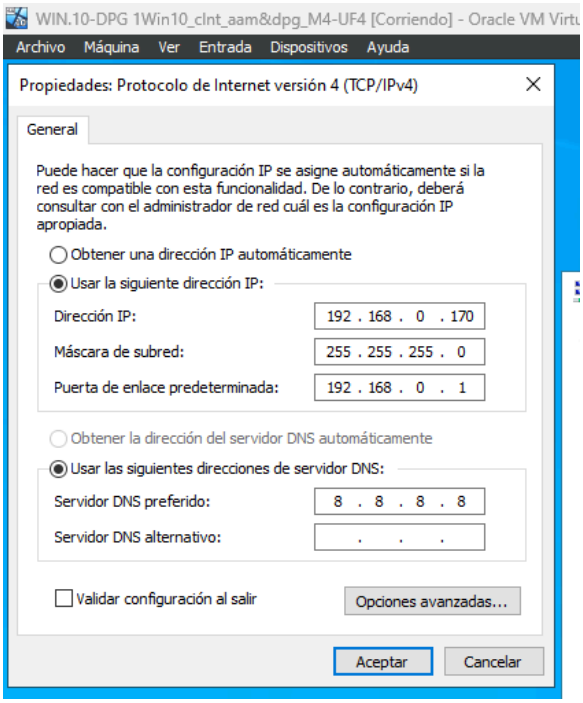


Debian12 (Node)

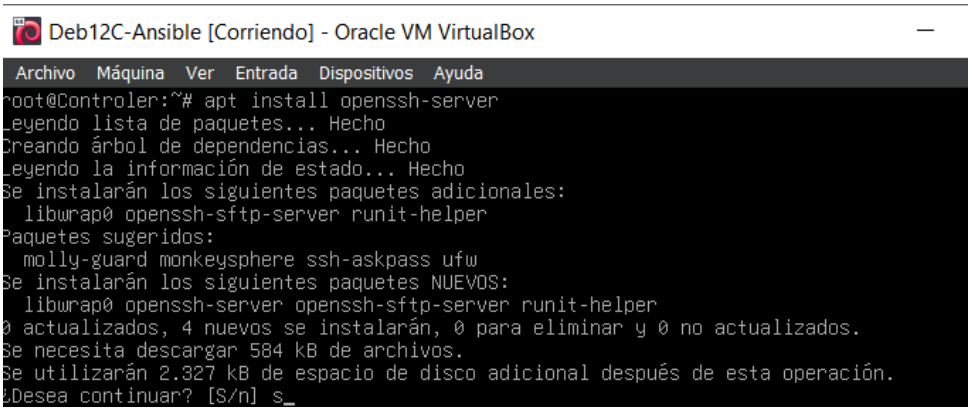


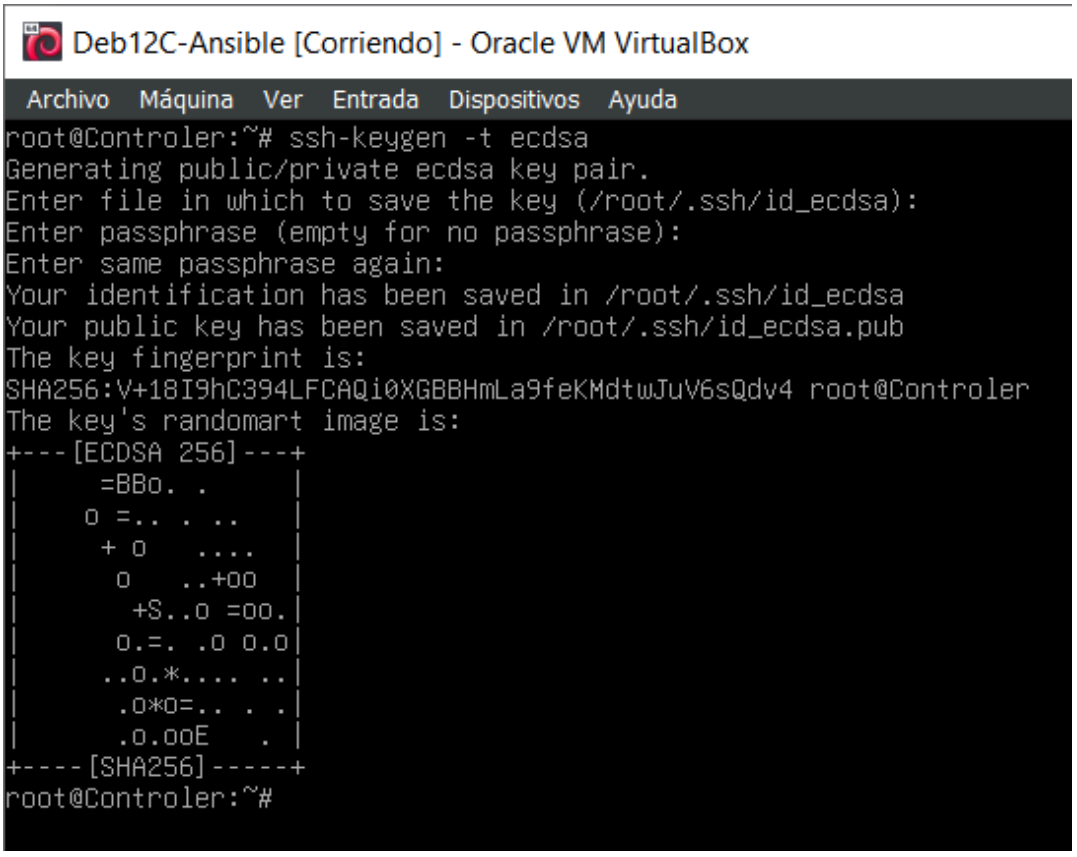


Windows 10


WIN.10-DPG 1Win10_clnt_aam&dpg_M4-UF4 [Corriendo] - Oracle VM Virtu
Archivo Máquina Ver Entrada Dispositivos Ayuda
Propiedades: Protocolo de Internet versión 4 (TCP/IPv4)
General
Puede hacer que la configuración IP se asigne automáticamente si la red es compatible con esta funcionalidad. De lo contrario, deberá consultar con el administrador de red cuál es la configuración IP apropiada.
Obtener una dirección IP automáticamente
Usar la siguiente dirección IP:
Dirección IP: 192 . 168 . 0 . 170
Máscara de subred: 255 . 255 . 255 . 0
Puerta de enlace predeterminada: 192 . 168 . 0 . 1
Obtener la dirección del servidor DNS automáticamente
Usar las siguientes direcciones de servidor DNS:
Servidor DNS preferido: 8 . 8 . 8 . 8
Servidor DNS alternativo: . . .
Validar configuración al salir
Opciones avanzadas...
Aceptar
Cancelar

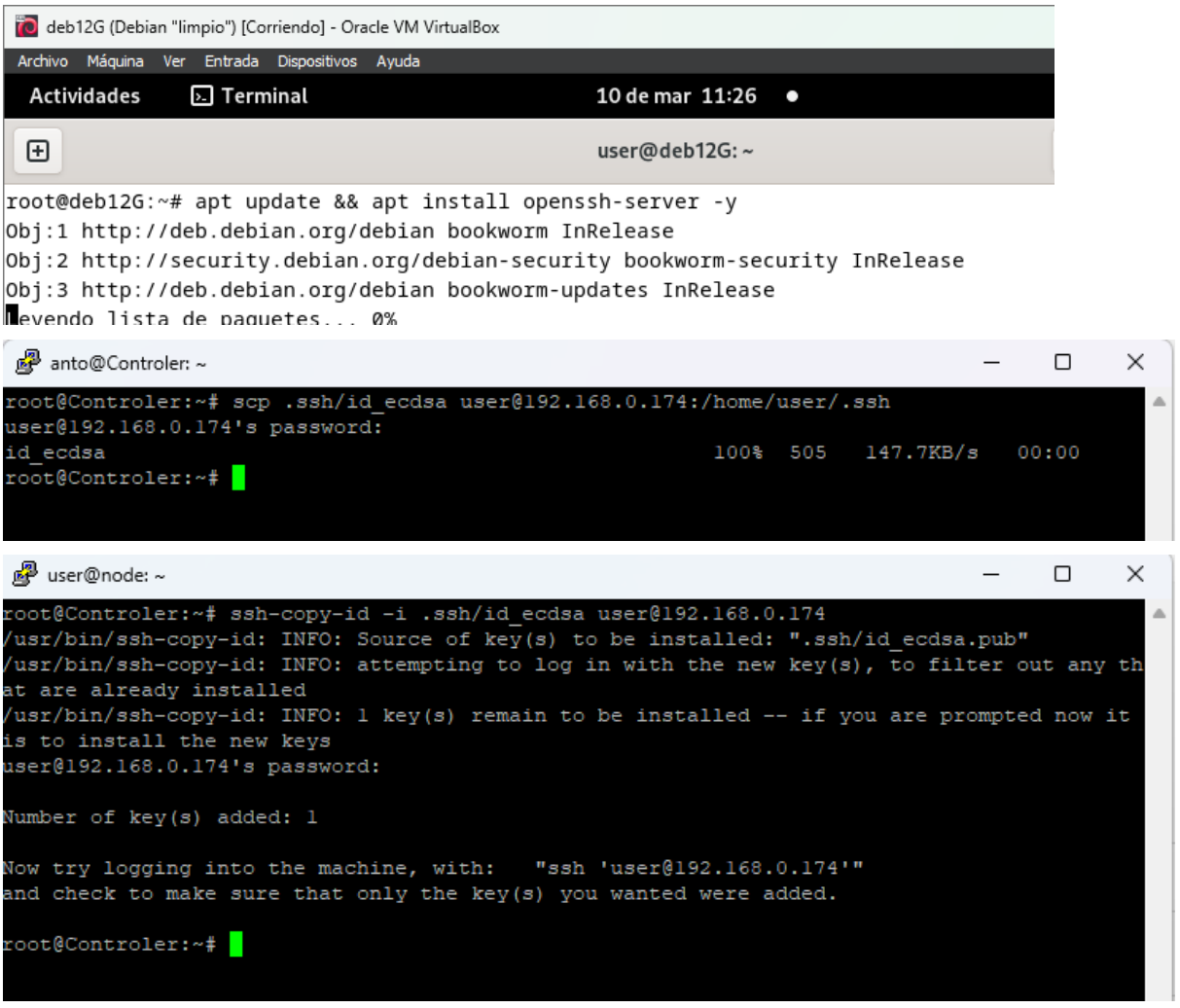
Genera les keys SSH al teu Controller Linux. Adjunta les captures de la creació. **(1punt)**

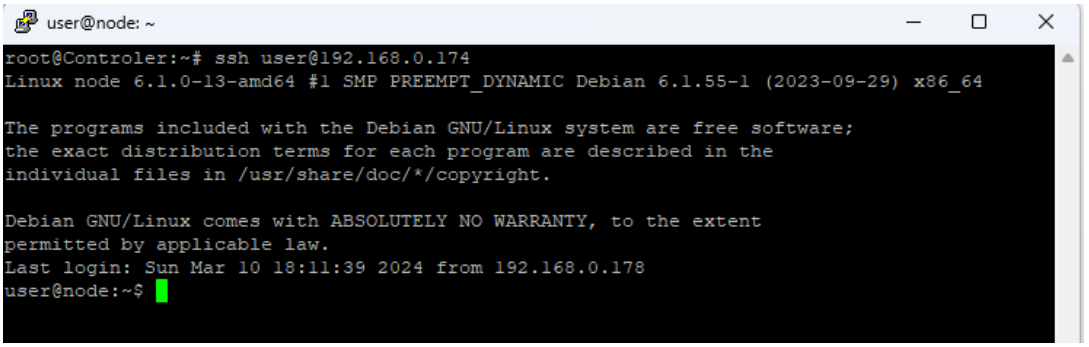


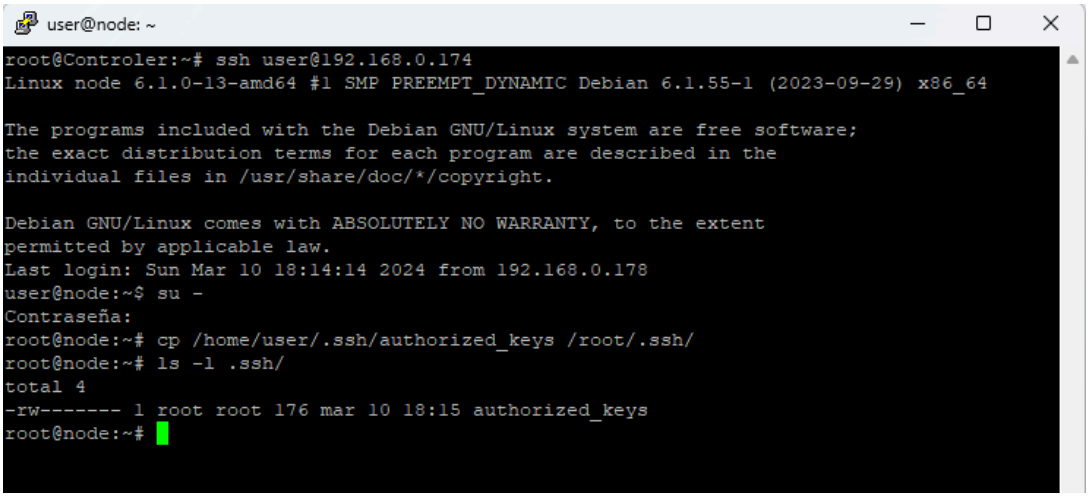


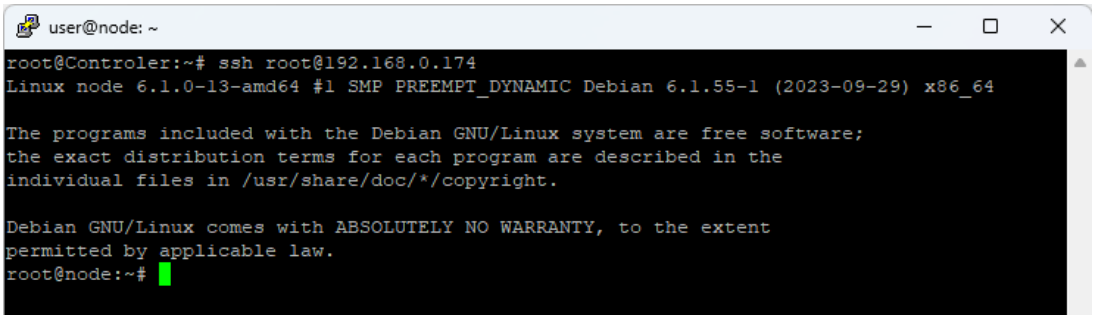
Busca als apunts com instal·lar openSSH a l'equip que farà de node Linux i com instal·lar openSSH al teu windows10 (a windows farem servir chocolatey, mireu la nota 3 i/o el video) per instal·lar OpenSSH, així no tindrem problemes amb l'openSSH que ve amb les característiques de windows, que l'hem d'instal·lar fora de la xarxa de l'escola perquè el firewall no ens el permet. Amb chocolatey no cal però si feu servir OpenSSH de les característiques opcionals de windows No oblideu configurar-lo perquè el servei no s'inicia automàticament).
Fes un manual d'instal·lació de OpenSSH server tant a windows com a Linux i com copies les keys a un node windows i un node linux. **(4 punts)**

Debian 12 (node):

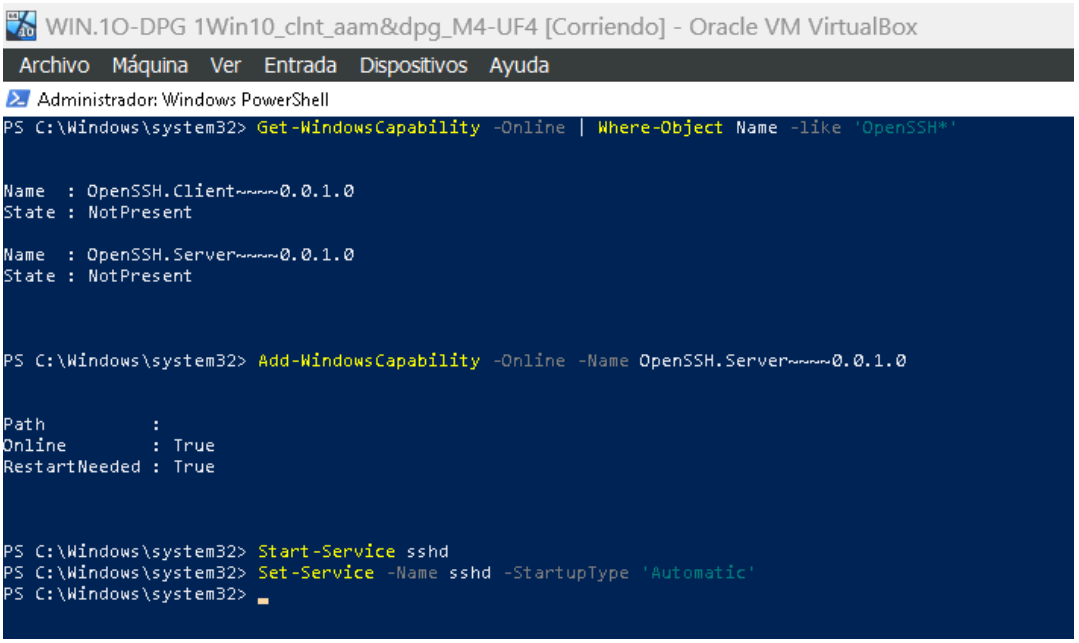



user@node: ~
root@Controler:~# ssh user@192.168.0.174
Linux node 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Mar 10 18:11:39 2024 from 192.168.0.178
user@node:~$


user@node: ~
root@Controler:~# ssh user@192.168.0.174
Linux node 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Mar 10 18:14:14 2024 from 192.168.0.178
user@node:~$ su -
Contraseña:
root@node:~# cp /home/user/.ssh/authorized_keys /root/.ssh/
root@node:~# ls -l .ssh/
total 4
-rw------- 1 root root 176 mar 10 18:15 authorized_keys
root@node:~#

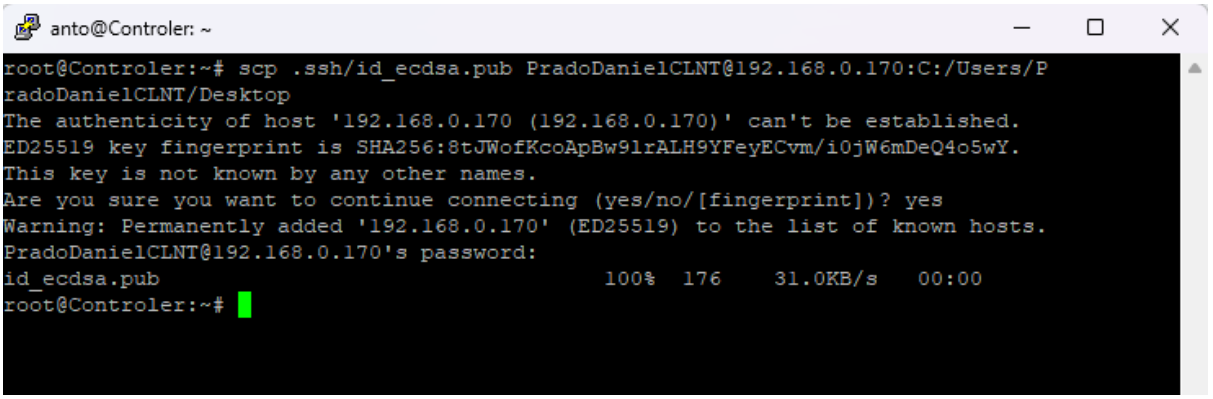

user@node: ~
root@Controler:~# ssh root@192.168.0.174
Linux node 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@node:~#

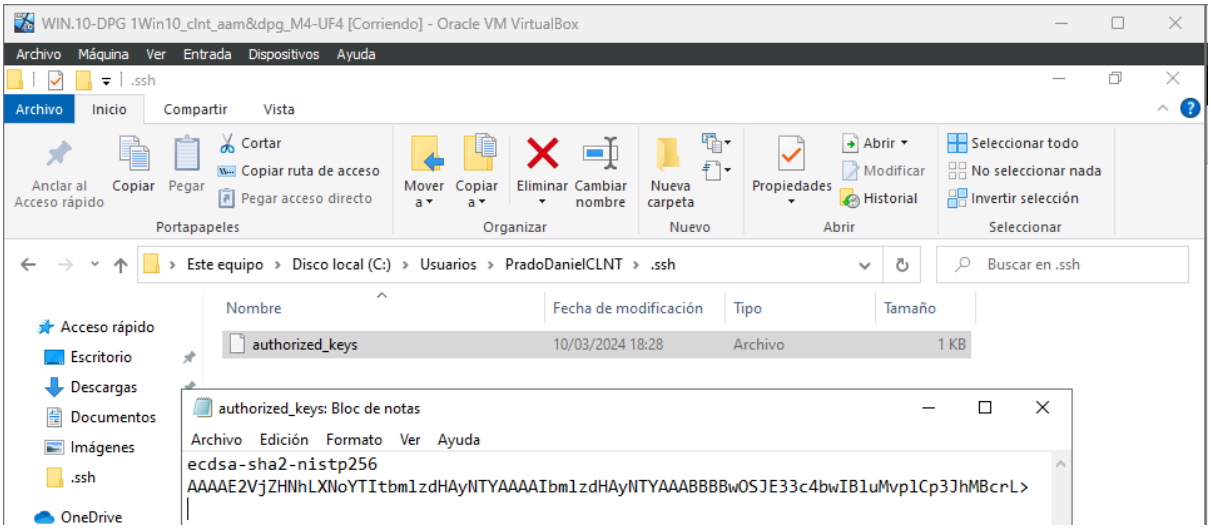
Windows 10:

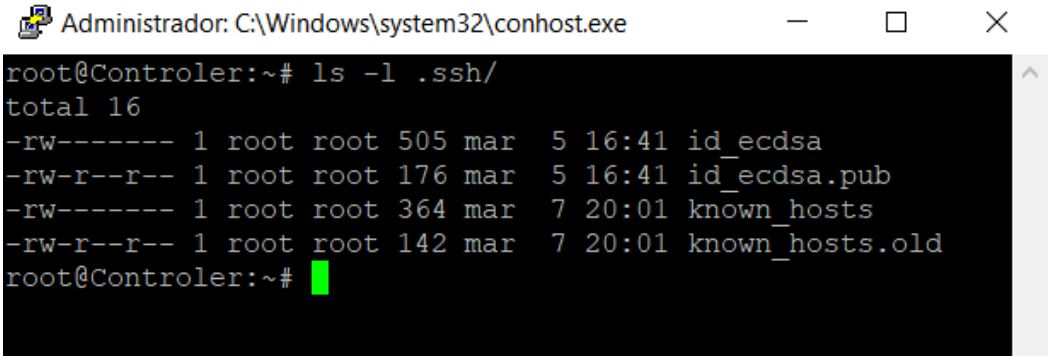


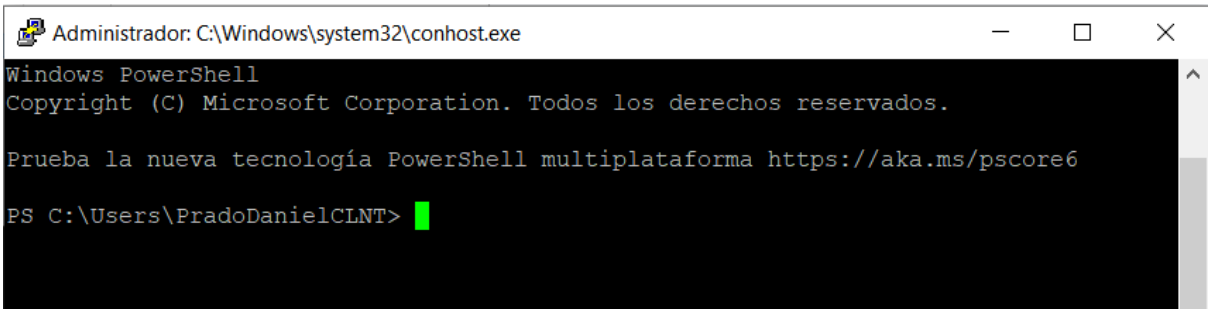
Copia la clau pública que has generat al teu Controller als 2 nodes. **(2 punts)**
→Las capturas generadas para Windows están más abajo, y las capturas generadas para debian están arriba.

NOTA 1: per windows el millor és copiar a l'escriptori de l'usuari, obrir la key amb el Notepad per exemple, copiar el contingut i enganxar-lo al fitxer .ssh/authorized_keys de l'usuari que feu servir. El directori .ssh i el fitxer authorized_keys s'han de crear i encara que comenci per . no és ocult.




WIN.10-DPG 1Win10_clnt_aam&dpg_M4-UF4 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
.ssh
Archivo Inicio Compartir Vista
Anclar al Acceso rápido
Copiar
Pegar
Cortar
Copiar ruta de acceso
Pegar acceso directo
Portapapeles
Mover a
Copiar a
Eliminar
Cambiar nombre
Organizar
Nueva carpeta
Nuevo
Propiedades
Abrir
Modificar
Historial
Abrir
Seleccionar todo
No seleccionar nada
Invertir selección
Seleccionar
Este equipo › Disco local (C:) › Usuarios › PradoDanielCLNT › .ssh
Buscar en .ssh
Acceso rápido
Escritorio
Descargas
Documentos
Imágenes
.ssh
OneDrive
Nombre
Fecha de modificación
Tipo
Tamaño
authorized_keys
10/03/2024 18:28
Archivo
1 KB
authorized_keys: Bloc de notas
Archivo Edición Formato Ver Ayuda
ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBBwOSJE33c4bwIBluMvplCp3JhMBcrL>



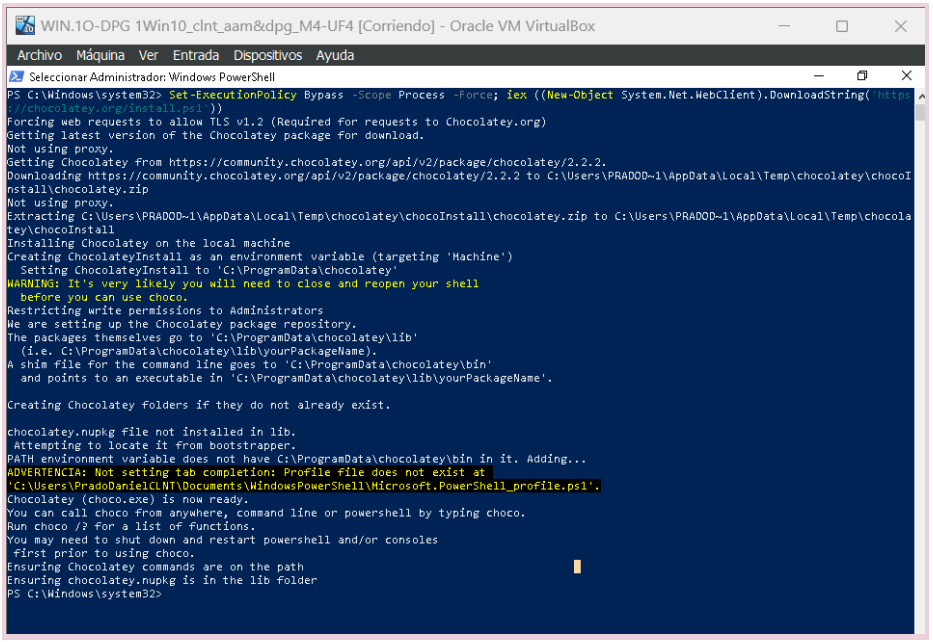


NOTA 2: que és chocolatey , chocolatey en Ansible
→Chocolatey es un gestor de paquetes para Windows, similar a apt para Ubuntu y pacman para Manjaro. Trabaja en armonía con Ansible, permitiéndote actualizar y administrar los ordenadores con sistemas operativos Windows de manera similar a como lo harías con GNU/Linux.

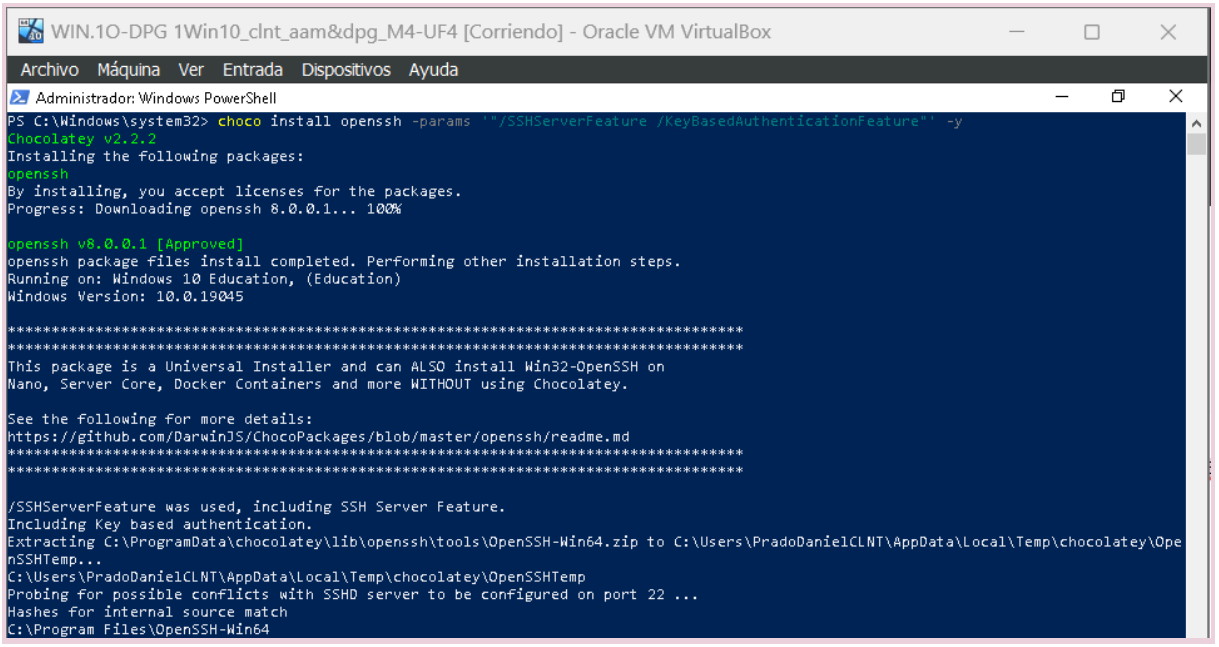
→En Ansible, puedes usar el módulo chocolatey.chocolatey.win_chocolatey ,qie se ua para administrar paquetes usando Chocolatey.
Este módulo es parte de la colección chocolatey.chocolatey y puedes instalarlo con el comando ansible-galaxy collection install chocolatey.chocolatey.

NOTA 3: comandes instal·lació chocolatey

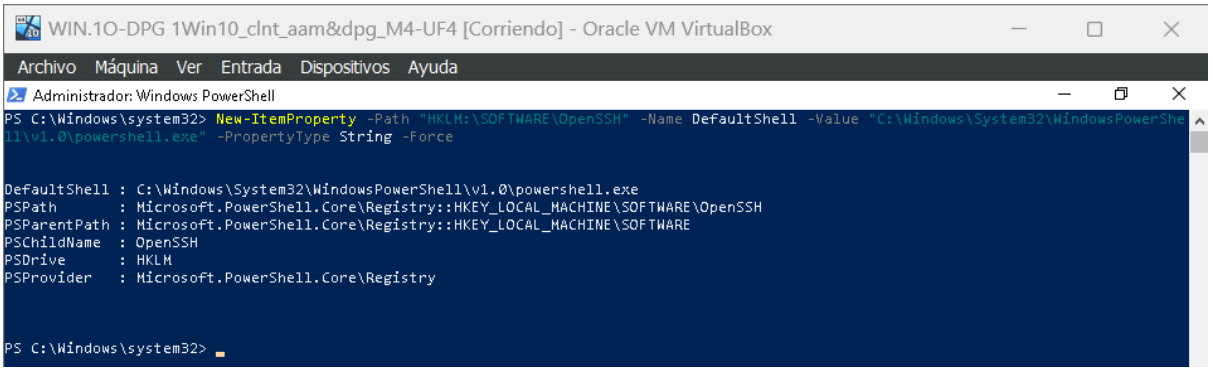
Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))



choco install openssh -params '"/SSHServerFeature /KeyBasedAuthenticationFeature"' -y

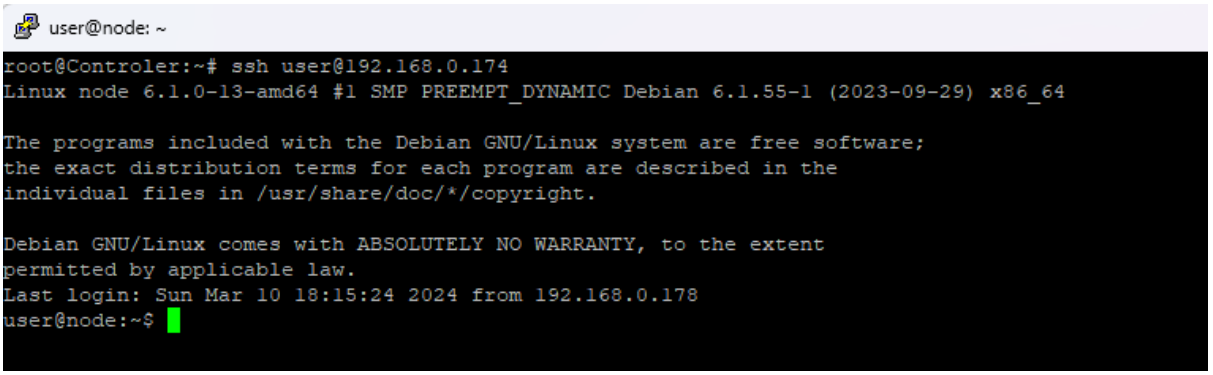


New-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell -Value "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -PropertyType String -Force



Prova que funciona la connexió ssh des del teu controller als nodes fent servir la Key. **(2 punts)**

Debian 12 (node):

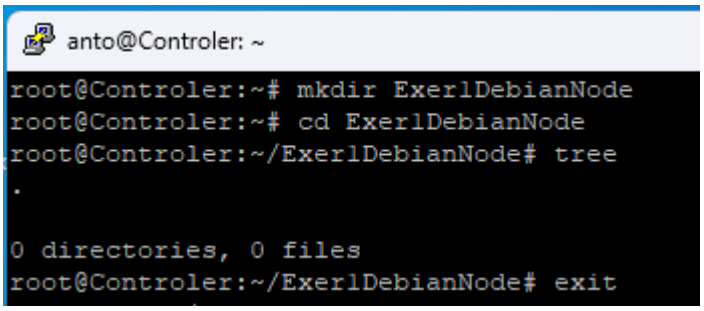


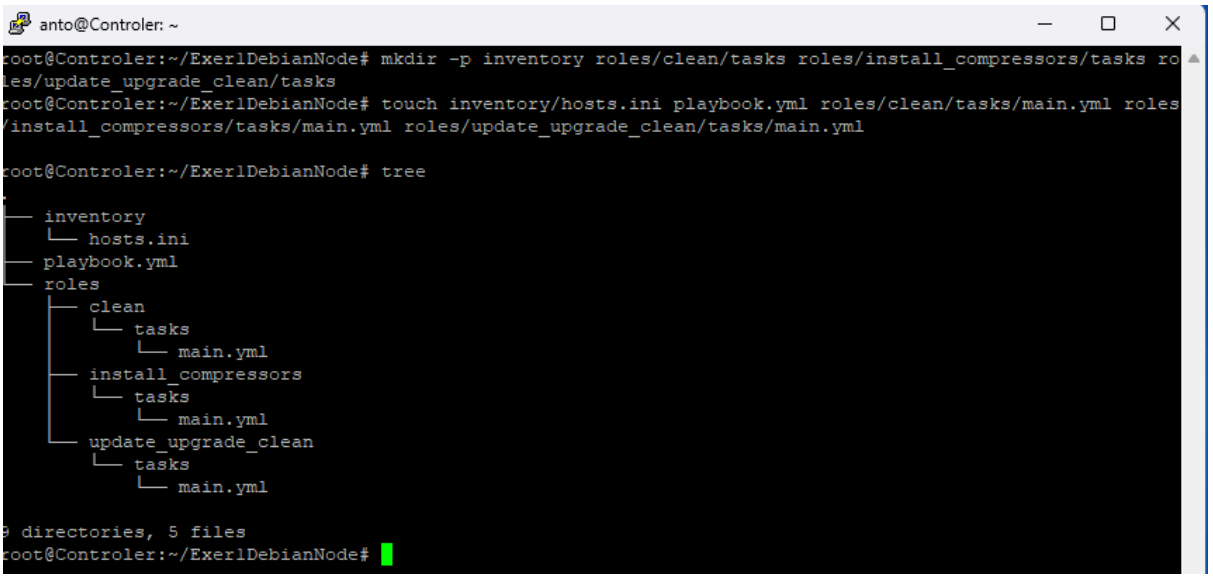
Windows 10:

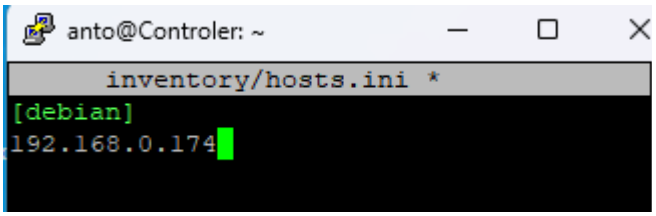
→Hemos estado hablando con algunos compañeros, y a muchos de ellos les estaba dando problemas la key de Windows 10…
Hemos estado leyendo varias páginas de la documentación oficial de Microsoft y Ansible y algún que otro video en YouTube, y no hemos podido hacer que funcione la Key…

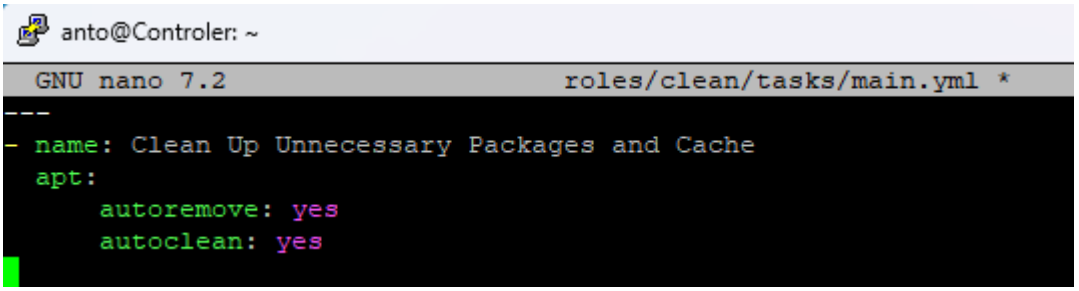
Exercici1 (10 punts)

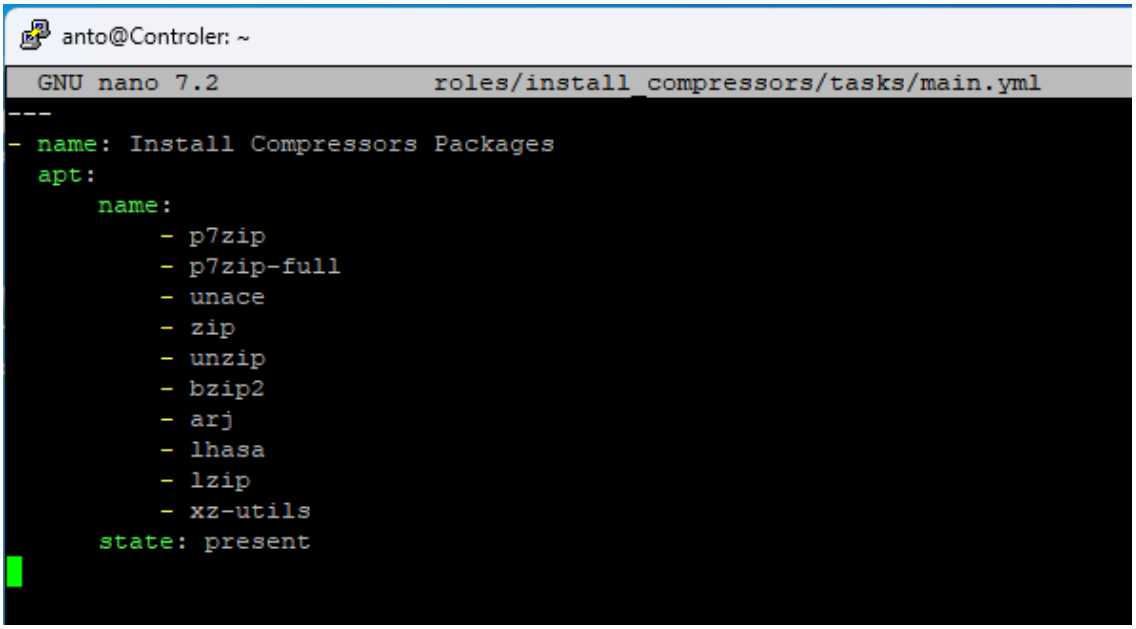
Realitza **i documenta** els passos necessaris (mkdir, nano, ansible-playbook …) per fer update, upgrade i instal·lar els següents paquets p7zip p7zip-full unace zip unzip bzip2 arj lhasa lzip xz-utils, i per últim netejar els paquets que no hagin fet servir (clean, cache …) a un node de debian fent servir roles. Crea un directori anomenat Exer1DebianNode al teu controller on ha d'anar tota l'estructura de directoris per fer servir els roles i els fitxers necessaris. L'invenetari s'ha de fer fent servir grups, encara que només sigui un node. Adjunta el contingut de tots els fitxer encara que sigui amb una captura.



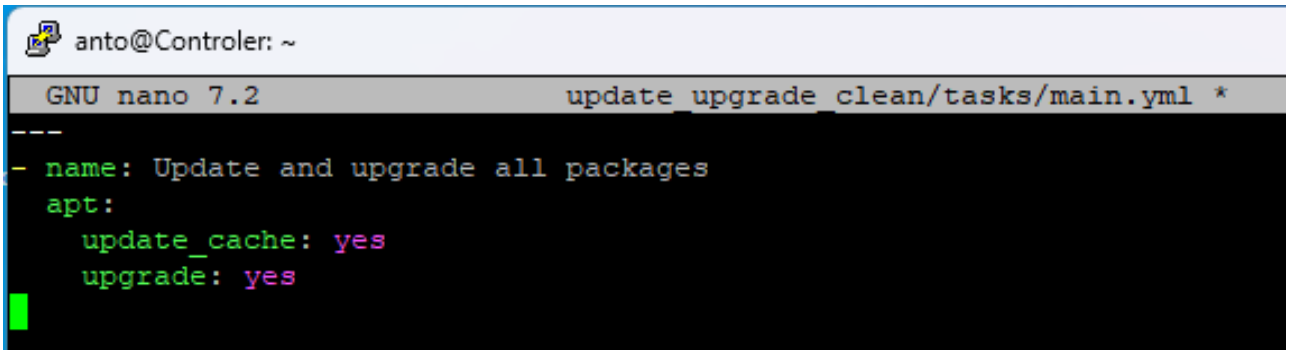




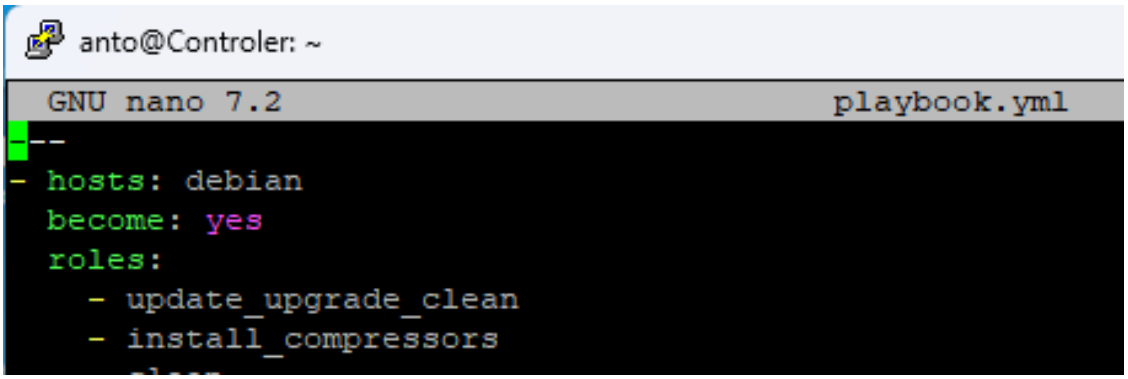



anto@Controler: ~
GNU nano 7.2 roles/install_compressors/tasks/main.yml

- name: Install Compressors Packages
apt:
name:
- p7zip
- p7zip-full
- unace
- zip
- unzip
- bzip2
- arj
- lhasa
- lzip
- xz-utils
state: present

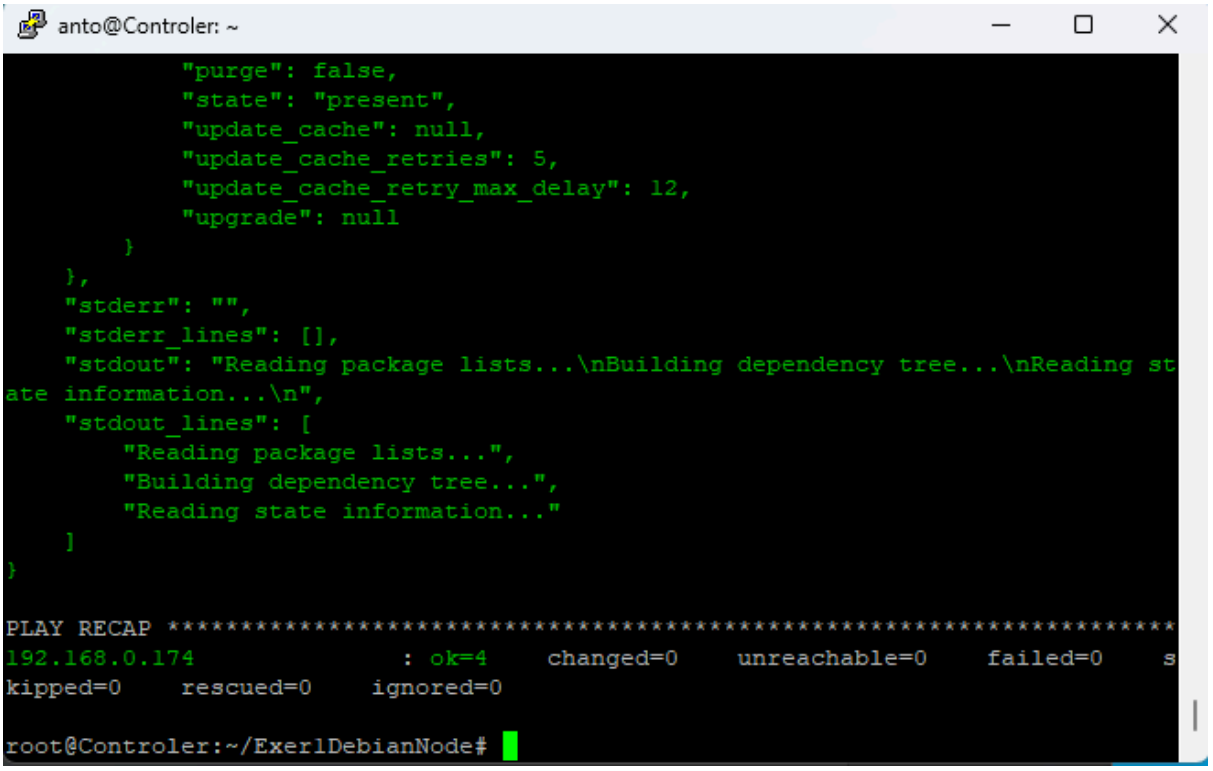

anto@Controler: ~
GNU nano 7.2 update_upgrade_clean/tasks/main.yml *

- name: Update and upgrade all packages
apt:
update_cache: yes
upgrade: yes


anto@Controler: ~
GNU nano 7.2 playbook.yml

- hosts: debian
become: yes
roles:
- update_upgrade_clean
- install_compressors

CARPETA DRIVE CON LOS FICHEROS/DIRECTORIOS CREADOS

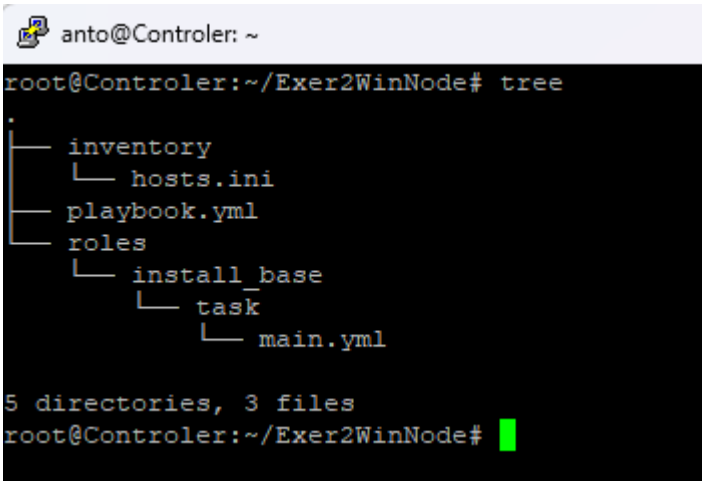


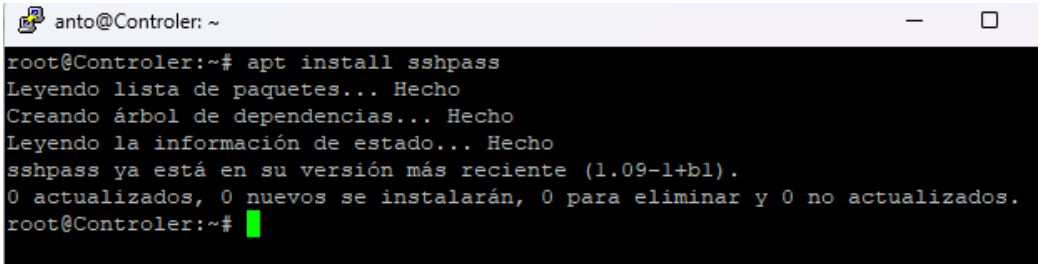
Exercici2 (10 punts)

Realitza i documenta els passos necessaris per instal·lar a un node windows mitjançant un playbook de ansible amb chocolatey amb ansible_ssh_private_key_file= si és possible en lloc de ansible_password (a casa em funciona amb private_key, a l'escola no,a l'escola l'he fet amb ansible_password) 7zip, Notepad++ i ksnip a un client windows. Crea un directori anomenat Exer2WinNode al teu controller on ha d'anar el fitxer d'inventari i el playbook

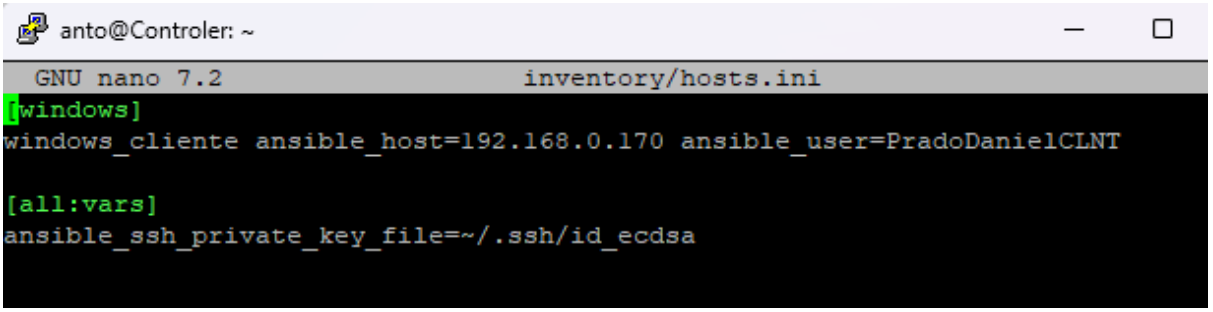
NOTA: Per fer servir ansible_password, hem d'instal·lar sshpass -> apt install sshpass

NOTA2: Tot el que es troba des d'aquí fins a l'exercici3 són ajudes que podeu trobar per realitzar aquest exercici

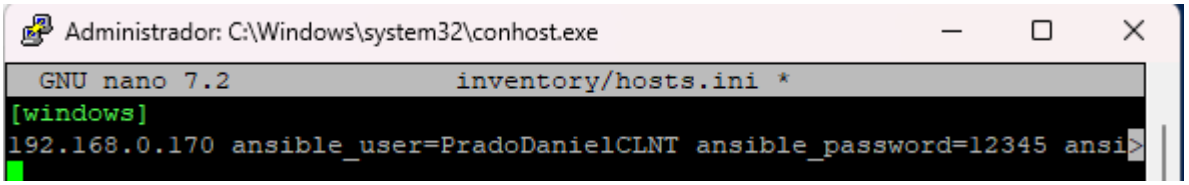




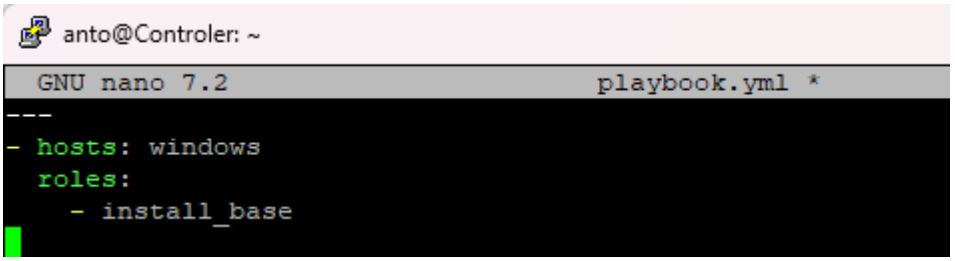
Configuracion con key (no me estaba funcionando)

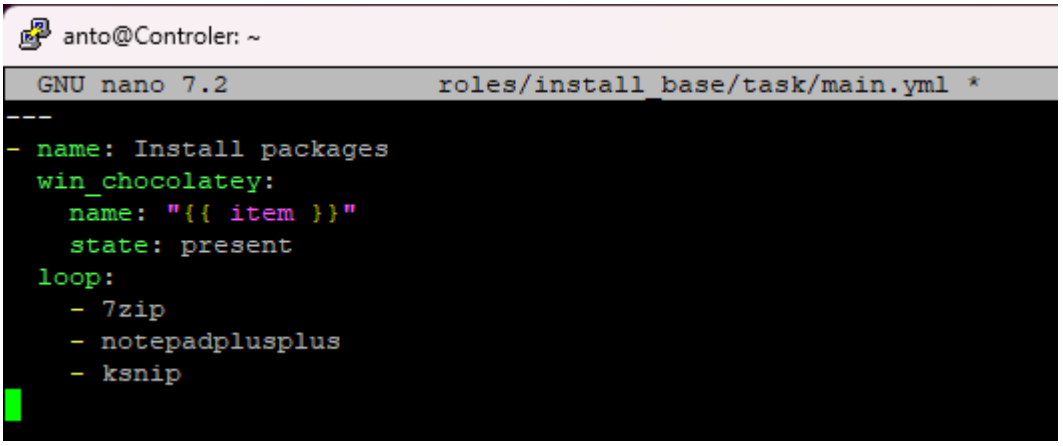


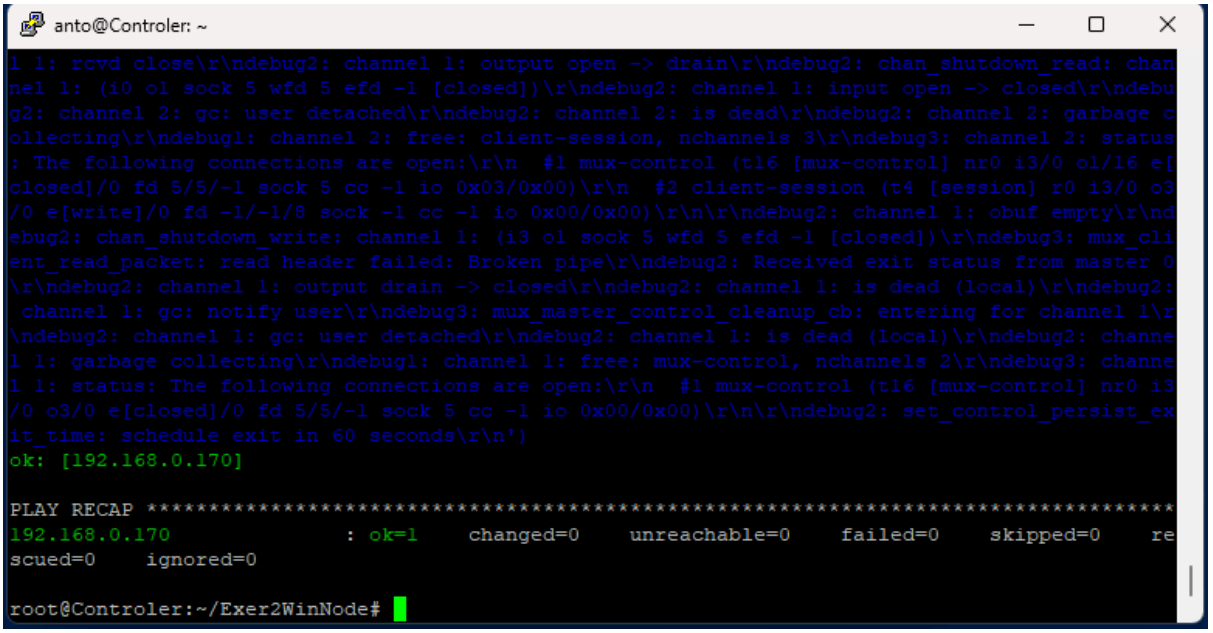
Configuracion con contraseña (me ha funcionado)



Comando completo (no se llega a ver en la captura)
[windows]
192.168.0.170 ansible_user=PradoDanielCLNT ansible_password=12345 ansible_connection=ssh ansible_shell_type=powershell ansible_winrm_server_cert_validation=ignore

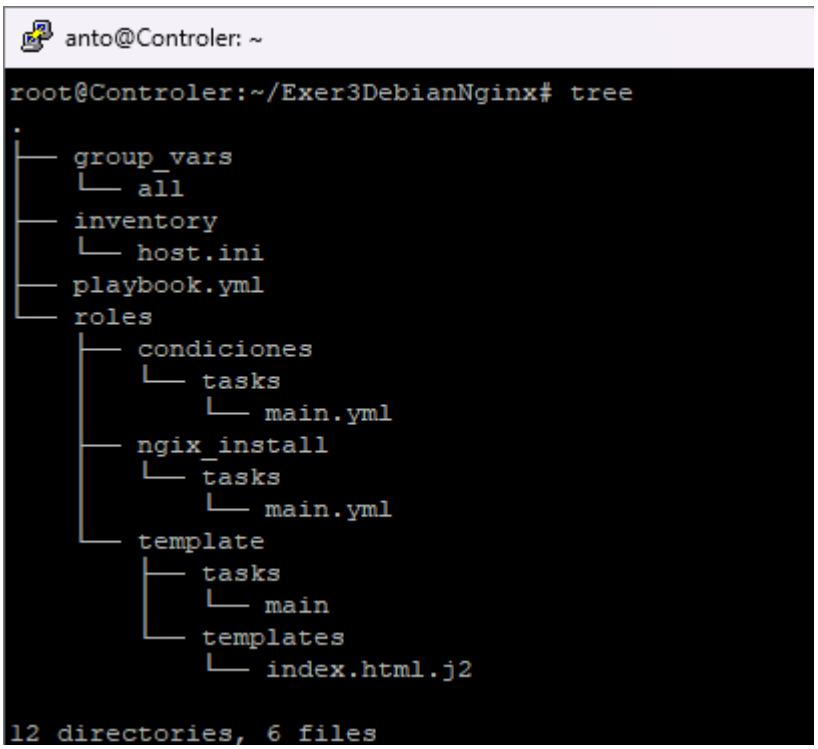


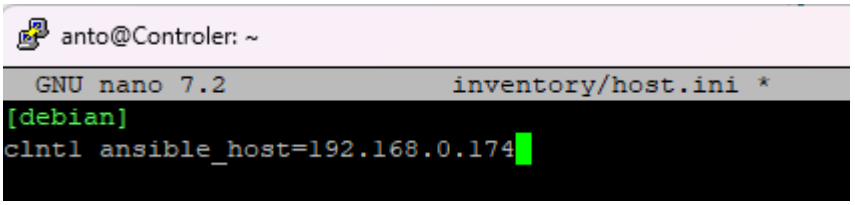


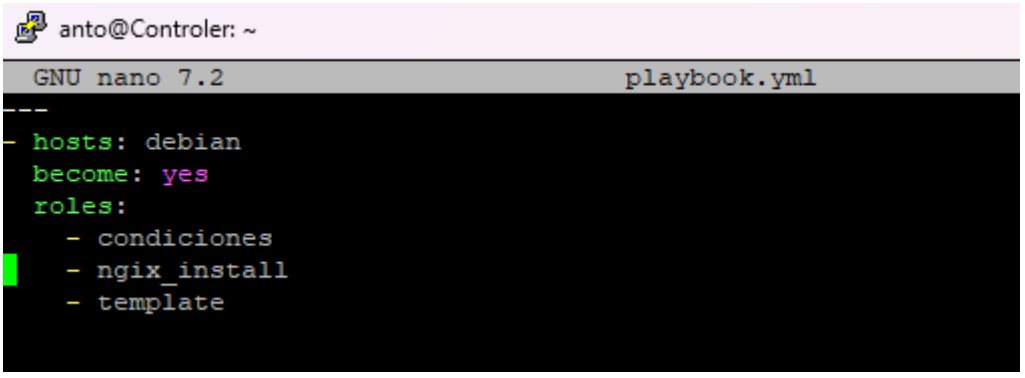

anto@Controler: ~
l 1: rcvd close\r\ndebug2: channel 1: output open -> drain\r\ndebug2: chan_shutdown_read: chan
nel 1: (i0 o1 sock 5 wfd 5 efd -1 [closed])\r\ndebug2: channel 1: input open -> closed\r\ndebu
g2: channel 2: gc: user detached\r\ndebug2: channel 2: is dead\r\ndebug2: channel 2: garbage c
ollecting\r\ndebug1: channel 2: free: client-session, nchannels 3\r\ndebug3: channel 2: status
: The following connections are open:\r\n #1 mux-control (t16 [mux-control] nr0 i3/0 o1/16 e[
closed]/0 fd 5/5/-1 sock 5 cc -1 io 0x03/0x00)\r\n #2 client-session (t4 [session] r0 i3/0 o3
/0 e[write]/0 fd -1/-1/8 sock -1 cc -1 io 0x00/0x00)\r\n\r\ndebug2: channel 1: obuf empty\r\nd
ebug2: chan_shutdown_write: channel 1: (i3 o1 sock 5 wfd 5 efd -1 [closed])\r\ndebug3: mux_cli
ent_read_packet: read header failed: Broken pipe\r\ndebug2: Received exit status from master 0
\r\ndebug2: channel 1: output drain -> closed\r\ndebug2: channel 1: is dead (local)\r\ndebug2:
 channel 1: gc: notify user\r\ndebug3: mux_master_control_cleanup_cb: entering for channel 1\r
\ndebug2: channel 1: gc: user detached\r\ndebug2: channel 1: is dead (local)\r\ndebug2: channe
l 1: garbage collecting\r\ndebug1: channel 1: free: mux-control, nchannels 2\r\ndebug3: channe
l 1: status: The following connections are open:\r\n #1 mux-control (t16 [mux-control] nr0 i3
/0 o3/0 e[closed]/0 fd 5/5/-1 sock 5 cc -1 io 0x00/0x00)\r\n\r\ndebug2: set_control_persist_ex
it_time: schedule exit in 60 seconds\r\n')
ok: [192.168.0.170]
PLAY RECAP ***
192.168.0.170   : ok=1   changed=0   unreachable=0   failed=0   skipped=0   re
scued=0   ignored=0
root@Controler:~/Exer2WinNode#

Exercici3 (43 punts)

Objectiu: Realitzar i documenta els passos necessaris per fer update, upgrade i instal·lar i configurar el paquet nginx a un node debian. Copiar i editar fitxers i Comprovar si existeixen.
Crea un directori anomenat Exer3DebianNginx on ha d'anar el fitxer d'inventari, que funcionarà amb el grup webservers i el seu playbook.yml corresponent per realitzar aquestes tasques al grup de hosts debianWebServers. Per cada tasca farem servir un role, nginx_install, nginx_config … L'estructura de l'exercici sencer m'ha quedat així (a falta dels role de fstab):





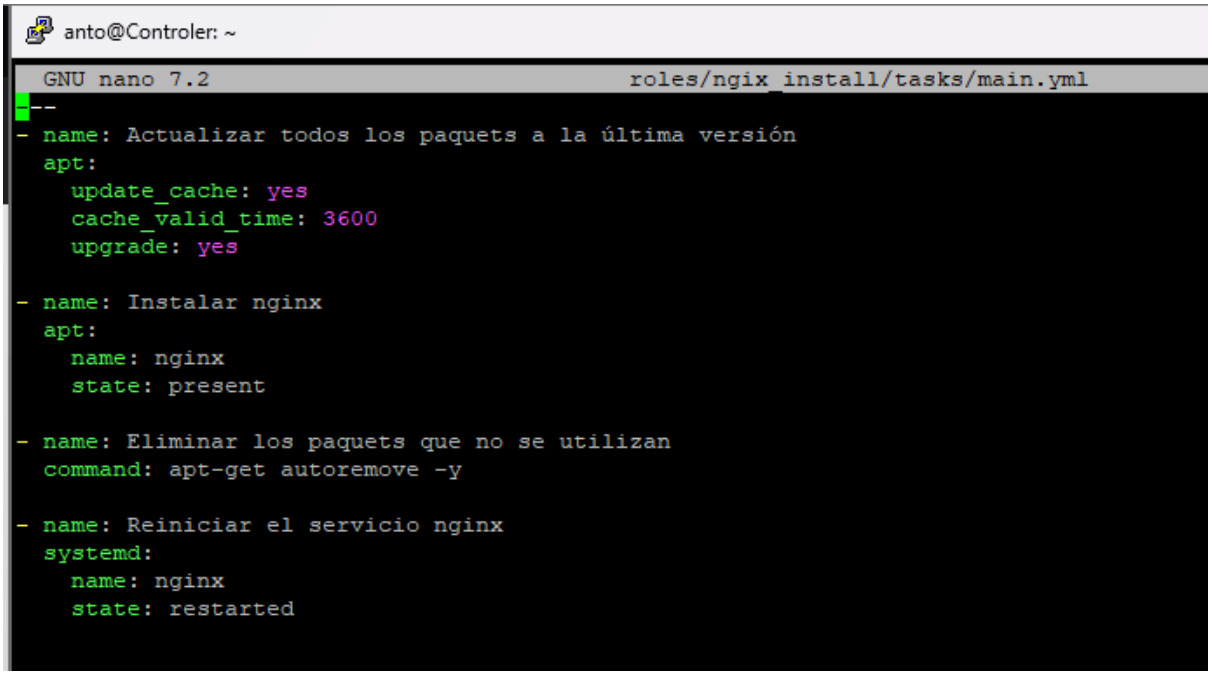


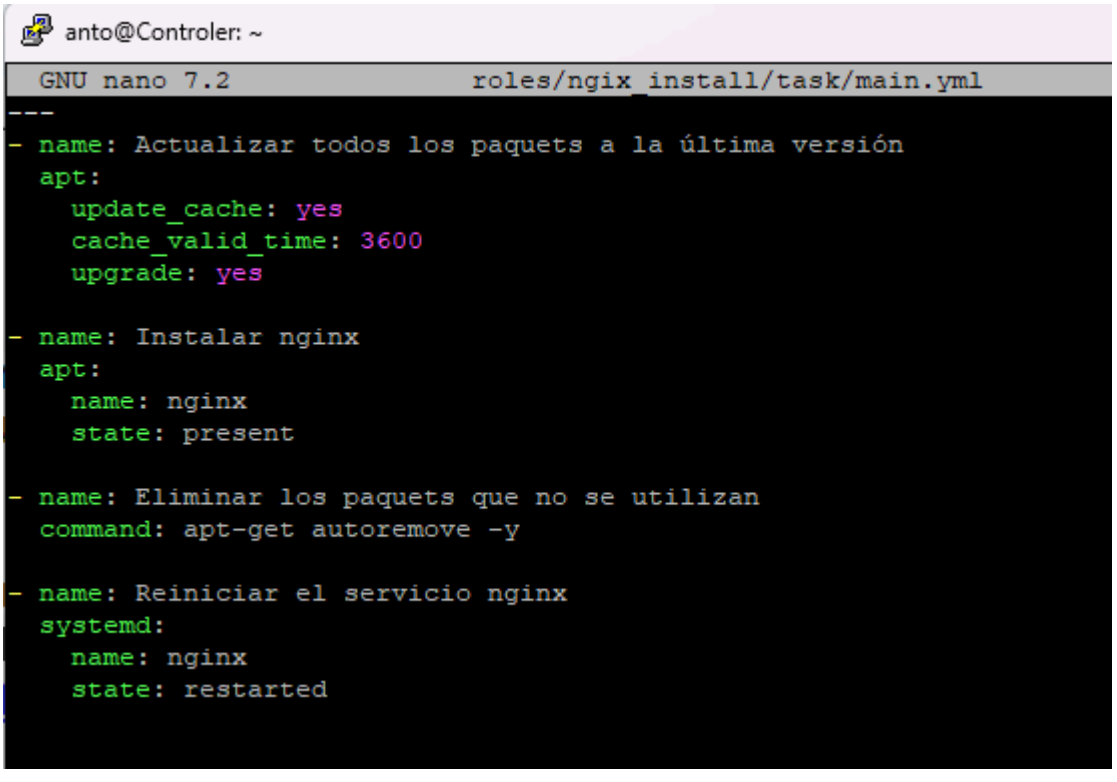
Es demana:

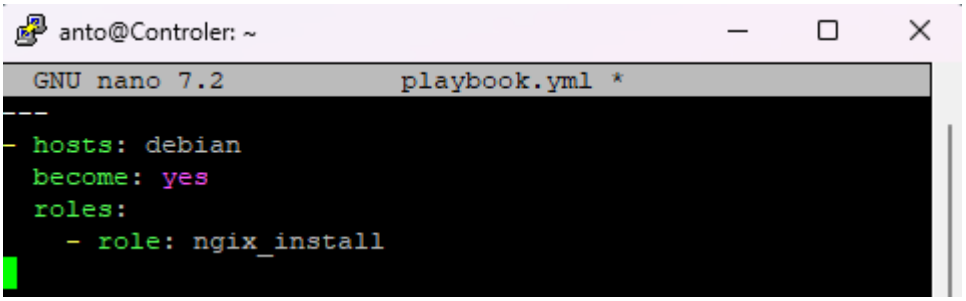
(10 punts) Pel role nginx_install voldrem que:

- actualitzi tots els paquets a l'última versió,
- faci un upgrade,
- instal·li nginx i esborri els paquets que no es facin servir
- l'update cache es farà cada hora (3600 segons) si el volem fer cada dia (86400) o cada setmana (604800), al mes (18748800)
- fes que es reinicii el servei ngix

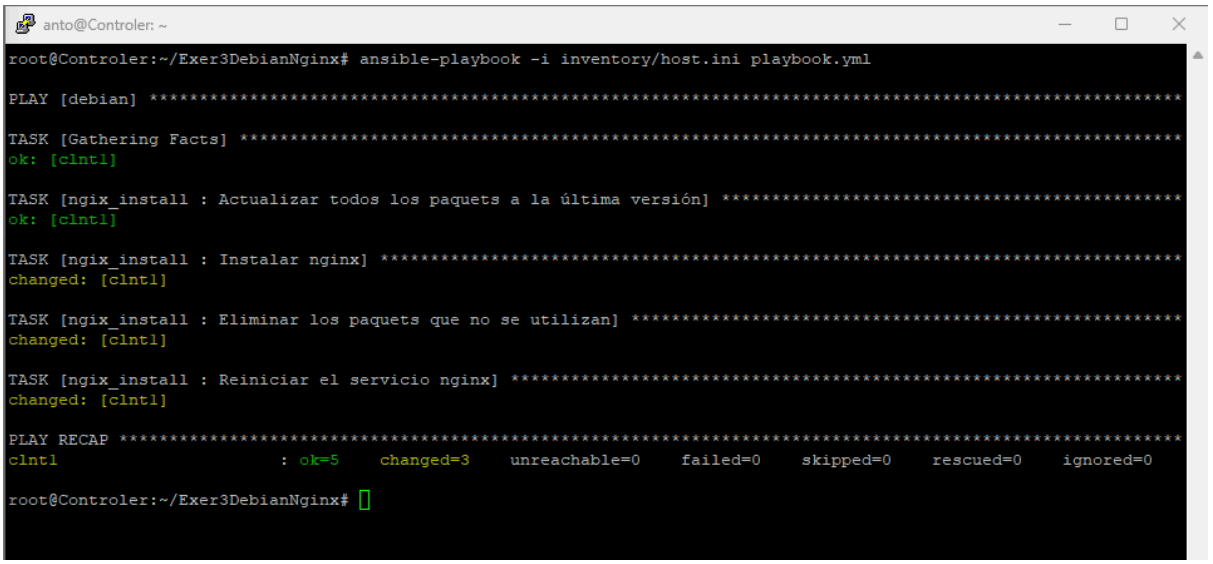
Executa el playbook amb els roles pertinents pel teu inventari.






anto@Controler: ~
GNU nano 7.2 playbook.yml *

- hosts: debian
become: yes
roles:
- role: ngix_install


anto@Controler: ~
root@Controler:~/Exer3DebianNginx# ansible-playbook -i inventory/host.ini playbook.yml
PLAY [debian] ***********
TASK [Gathering Facts] ***********
ok: [clnt1]
TASK [ngix_install : Actualizar todos los paquets a la última versión] ***********
ok: [clnt1]
TASK [ngix_install : Instalar nginx] ***********
changed: [clnt1]
TASK [ngix_install : Eliminar los paquets que no se utilizan] ***********
changed: [clnt1]
TASK [ngix_install : Reiniciar el servicio nginx] ***********
changed: [clnt1]
PLAY RECAP ***********
clnt1 : ok=5 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
root@Controler:~/Exer3DebianNginx#

.

(12 punts) Pel role de condicions (mira aquesta web i aquesta)

- Registrem/guandem dins d'una variable que es dirà ssh_file la ruta i el nom del fitxer que volem comprovar si existeix al node /root/.ssh/id_ecdsa.pub.Controller (o la key que hagim creat amb ssh-keygen (ed55199, rdsa …))
- Registrem/guandem dins d'una variable que es dirà ssh_folder la ruta /root/.ssh
- Comprova que existeix directori /root/.ssh al teu Node
- Mostrarem per pantalla “Folder exists” si existeix i “Folder NOT exists!!!” si no existeix
- Si no existeix /root/.ssh al teu Node l’haurem de crear mitjançant Ansible per owner root, group root i permisos 0700
- Comprova que existeix el fitxer /root/.ssh/id_ecdsa.pub.Controller (o la key que hagim creat amb ssh-keygen (ed55199, rdsa …)) al teu Node
- Mostrarem per pantalla el missatge “File exists” si existeix i “File NOT exists!!!” si no existeix
- Si no Existeix, copiarem el fitxer /root/.ssh/id_ecdsa.pub del nostre Controller a /root/.ssh/id_ecdsa.pub.Controller del nostre Node per owner root, group root i permisos 0644 (amb això podríem haver passat la nostra clau pública, però la necessitem abans …)

Executa el playbook amb els roles pertinents pel teu inventari.

anto@Controler: ~

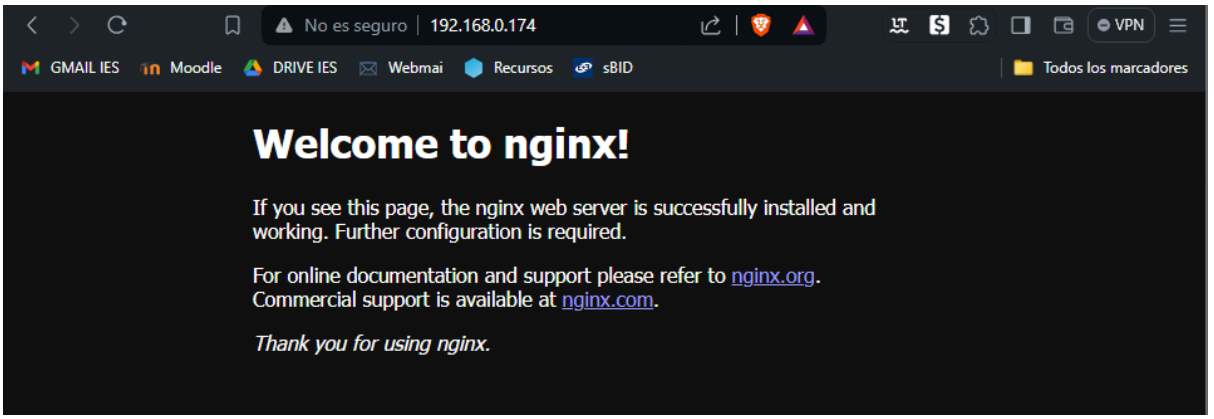
```
  GNU nano 7.2                                          playbook.yml *
---
- hosts: debian
  become: yes
  roles:
    - role: ngix_install
    - role: condiciones
```

anto@Controler: ~

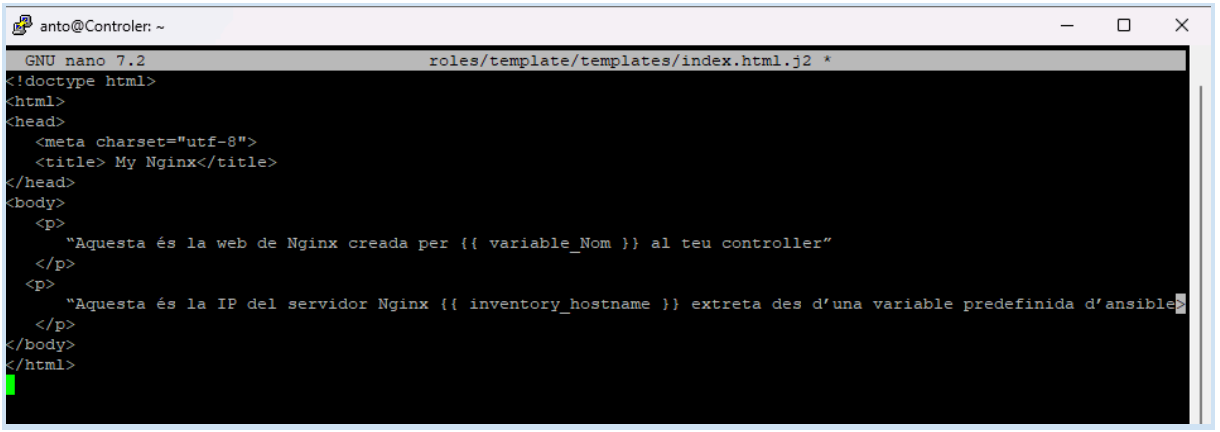
```
root@Controler:~/Exer3DebianNginx# ansible-playbook -i inventory/host.ini playbook.yml

PLAY [debian] ***********************************************************************************************************

TASK [Gathering Facts] **************************************************************************************************
ok: [clnt1]

TASK [ngix_install : Actualizar todos los paquets a la última versión] **************************************************
ok: [clnt1]

TASK [ngix_install : Instalar nginx] ************************************************************************************
ok: [clnt1]

TASK [ngix_install : Eliminar los paquets que no se utilizan] ***********************************************************
changed: [clnt1]

TASK [ngix_install : Reiniciar el servicio nginx] ***********************************************************************
changed: [clnt1]

TASK [condiciones : Registrar variables] ********************************************************************************
ok: [clnt1]

TASK [condiciones : Comprobar si existe el directorio] ******************************************************************
ok: [clnt1]

TASK [condiciones : Mostrar si el directorio existe] *********************************************************************
ok: [clnt1] => {
    "msg": "Folder exists"
}

TASK [condiciones : Mostrar si el directorio no existe] ******************************************************************
skipping: [clnt1]

TASK [condiciones : Crear el directorio si no existe] ********************************************************************
skipping: [clnt1]

TASK [condiciones : Comprobar si existe el archivo] **********************************************************************
ok: [clnt1]

TASK [condiciones : Mostrar si el archivo existe] ************************************************************************
ok: [clnt1] => {
    "msg": "File exists"
}

TASK [condiciones : Mostrar si el archivo no existe] *********************************************************************
skipping: [clnt1]

TASK [condiciones : Copiar el archivo si no existe] **********************************************************************
skipping: [clnt1]

PLAY RECAP **************************************************************************************************************
clnt1                      : ok=10   changed=2    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0

root@Controler:~/Exer3DebianNginx#
```

(9 punts) Role Template amb variables

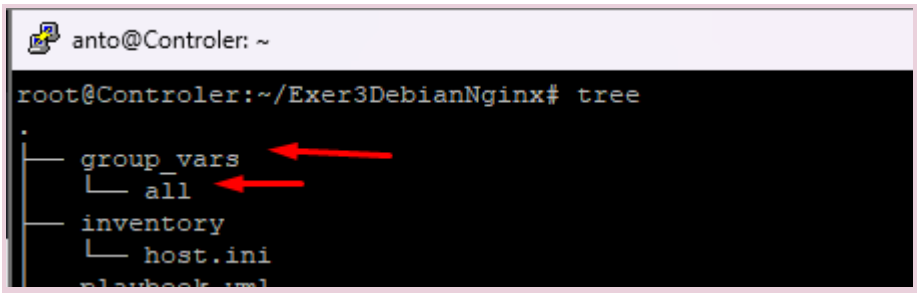
- Des d'un navegador web d'una màquina que tingui comunicació amb el node, escriu la IP del Node on has instal·lat Nginx i adjunta una captura que mostri la web per defecte de Nginx)

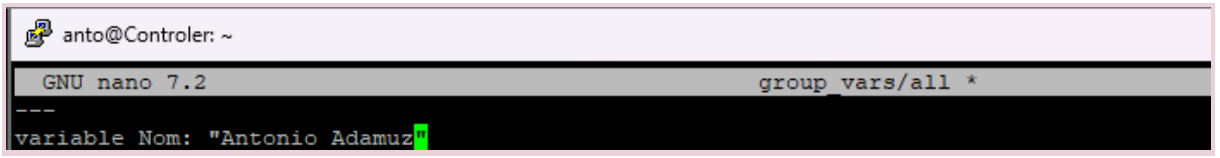


- Crea el fitxer index.html.j2 al teu controller amb el contingut i la ruta indicada just a sota
  NOTA: crea la ruta per posar el fitxer al directori indicat a ella, ja que més endavant demano fer-lo mitjançant un template i no un copy
  Aquí tens el contingut de index.html.j2 (aquesta seria la ruta feta servir per mi)
  #nano Exer3DebianNginx/roles/template/templates/index.html.j2

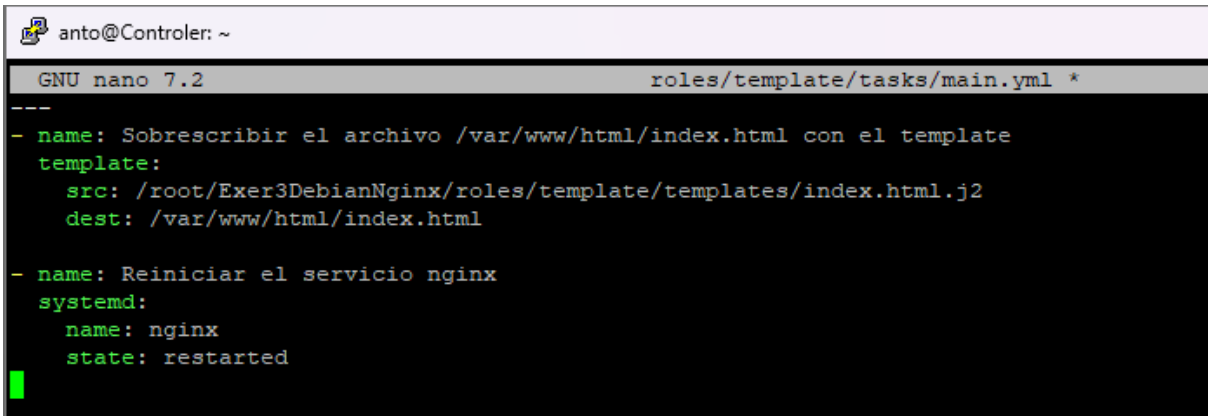


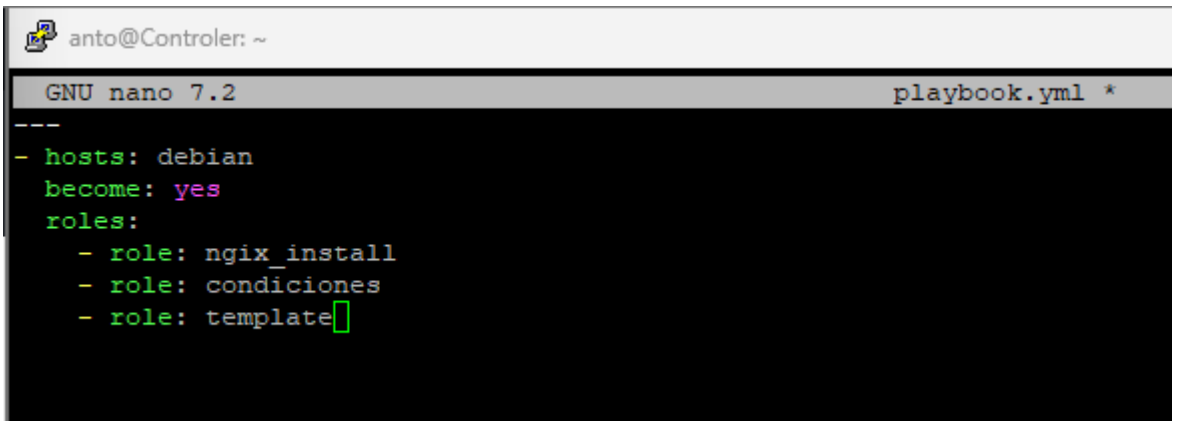
- la variable_Nom contindrà El Teu Nom i Cognom (si la posem al fitxer all li estem dient que la variable serà per tots els usuaris)
  #mkdir Exer3DebianNginx/group_vars
  #nano Exer3DebianNginx/group_vars/all

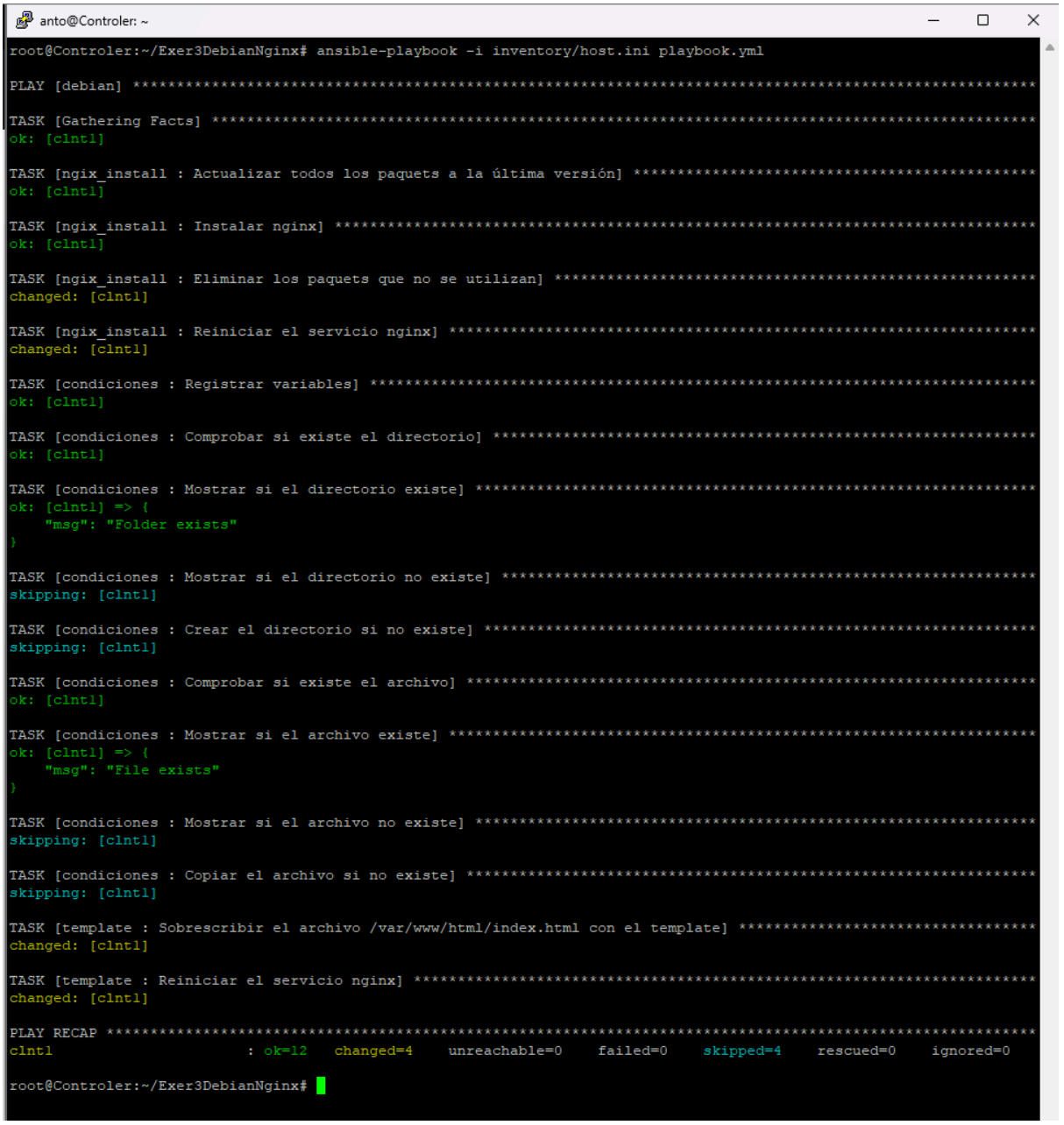




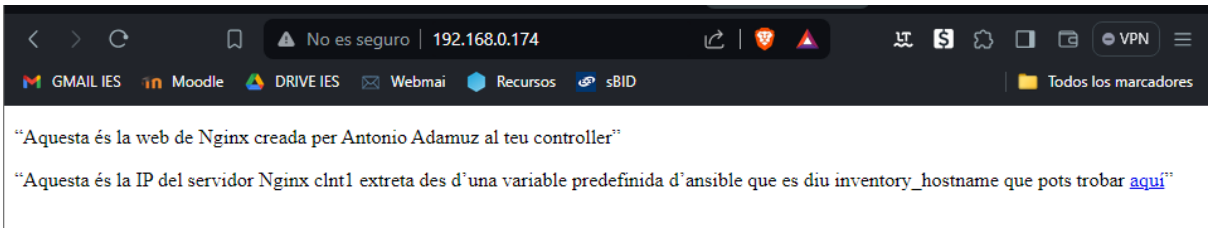
- Fent servir un template dins d'un role anomenat template, sobreescriu el fitxer /var/www/html/index.html del node on has instal·lat Nginx amb el fitxer index.html.j2 que acabes de crear (recorda que hem de fer que nginx reinicialitzi el servei per fer efectius els canvis de la web)







- Des del teu navegador escriu la IP del Node on has instal·lat Nginx (fes F5 per si la tens a caché) i adjunta una captura amb la sortida de la teva web.



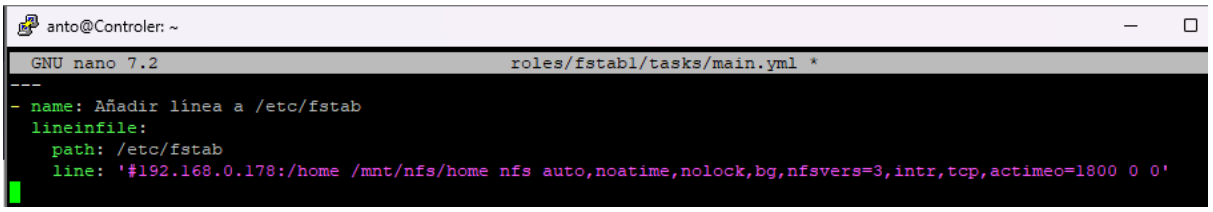
(4 punts) Afegir una línia a /etc/fstab

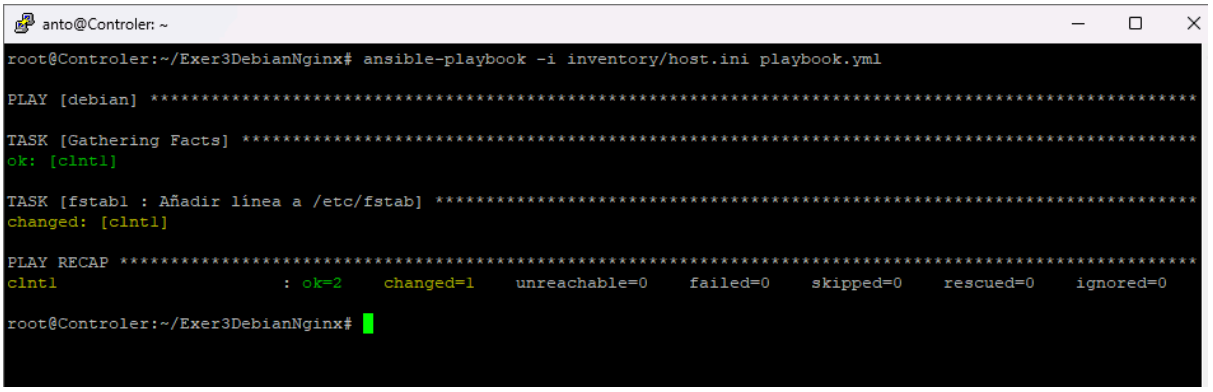
- fes servir un role que es digui fstab1 on fent servir el mòdul lineinfile d'ansible afegiràs la línia següent al teu Debian node. Adjunta el codi fet servir.

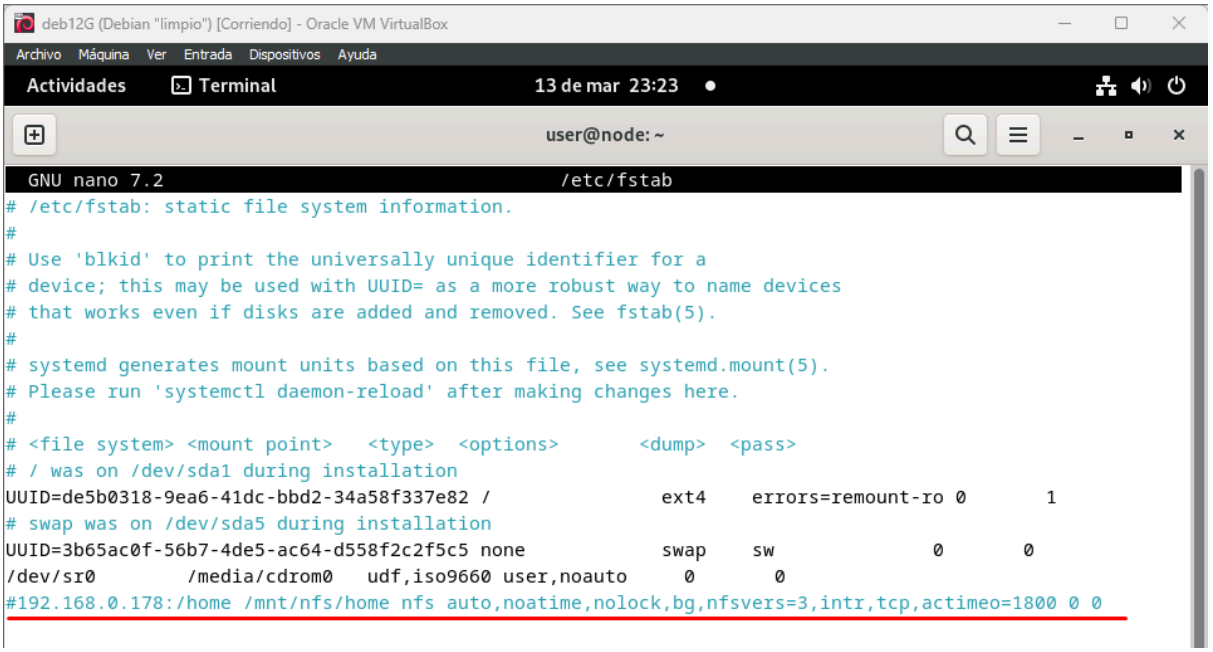
#192.168.1.175:/home /mnt/nfs/home nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0

NOTA: la línia l'has d'introduir amb el comentari per evitar errors

adjunta les captures que demostrin que has executat el role i ha funcionat







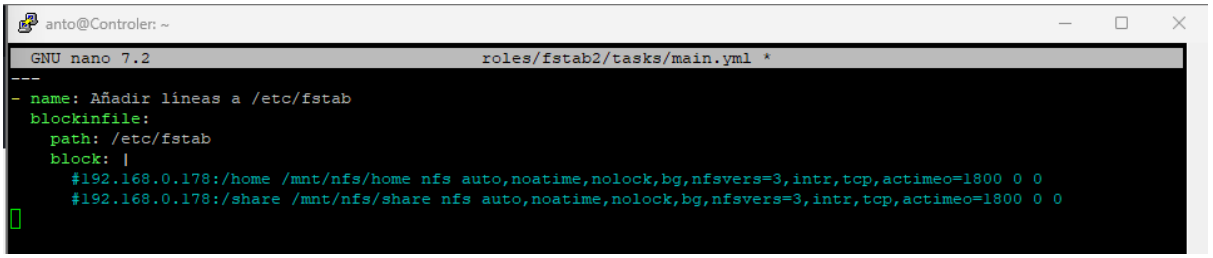
(8) Afegir dos línies a /etc/fstab

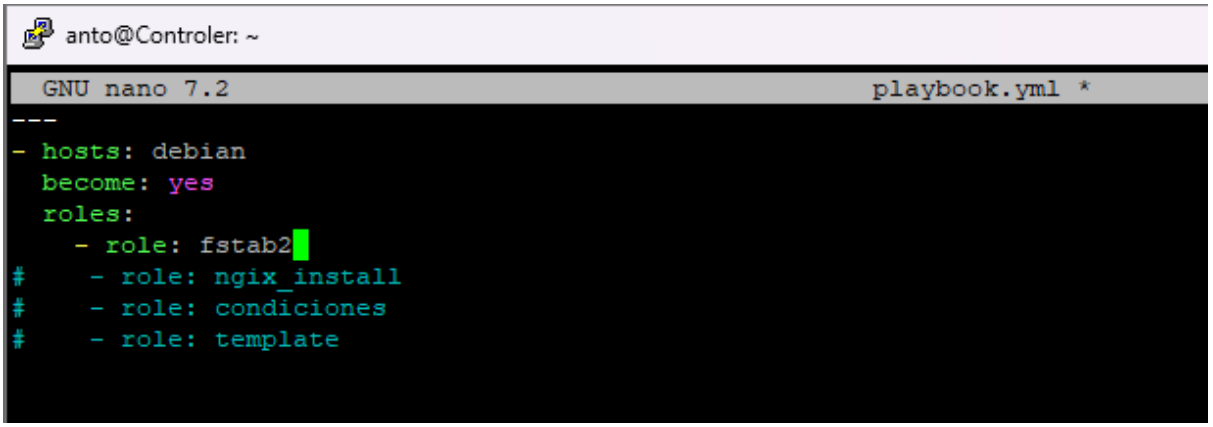
- comenta el role fstab1 anterior al teu playbook.yml i afegeix el role que crearem a continuació fstab2
- fes servir un role que es digui fstab2 on fent servir el [mòdul blockinfile](#) d'ansible afegiràs la línia següent al teu Debian node. Adjunta el codi fet servir.

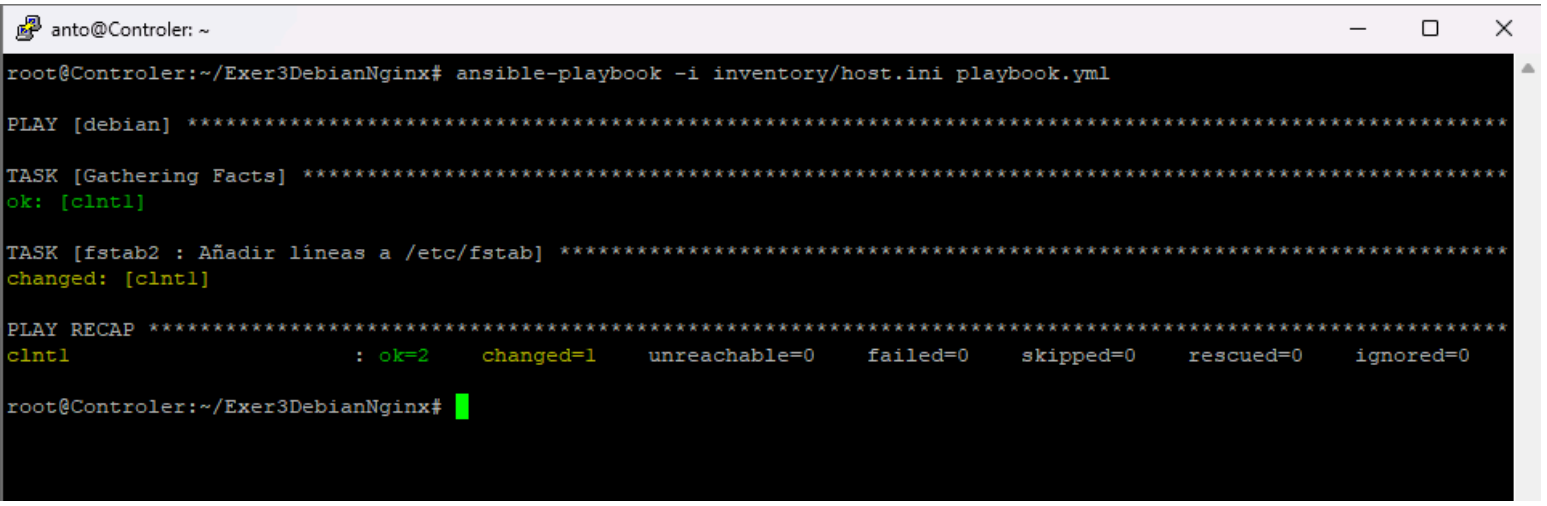
#192.168.1.175:/home /mnt/nfs/home nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
#192.168.1.175:/share /mnt/nfs/share nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0

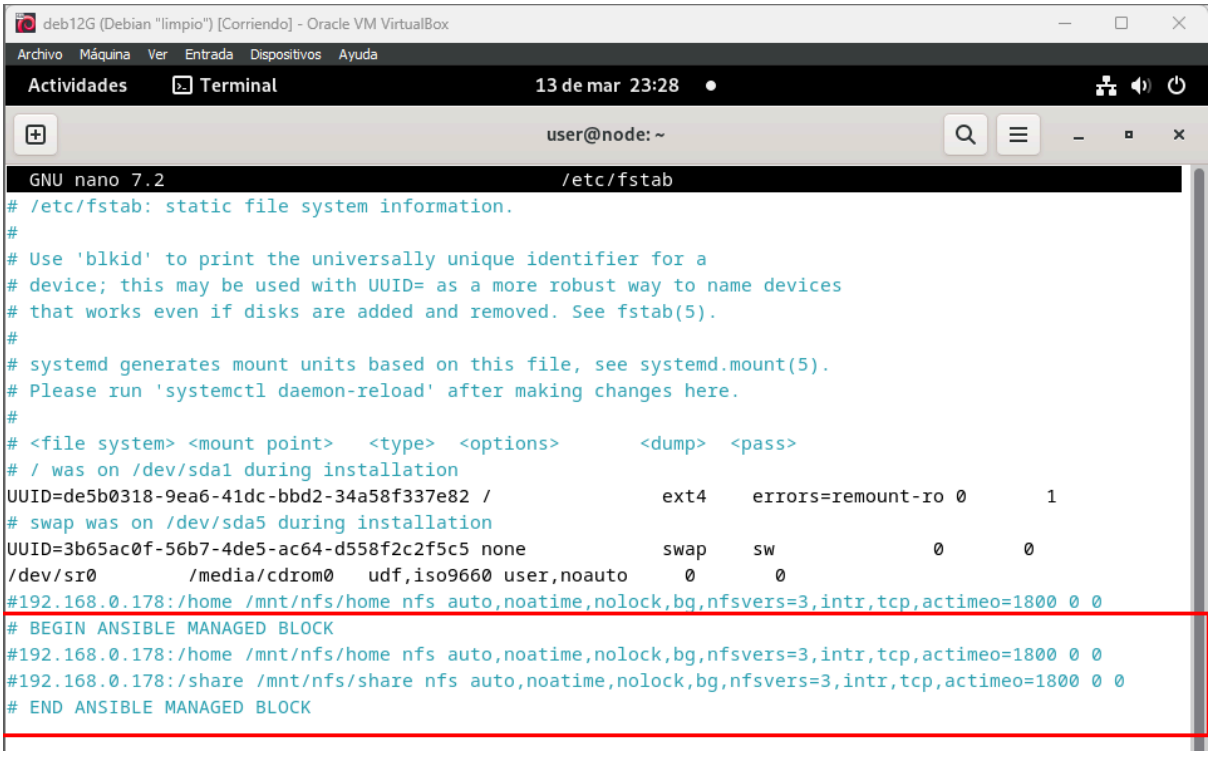
NOTA: les línies les has d'introduir amb el comentari per evitar errors
adjunta les captures que demostrin que has executat el role i ha funcionat
Accedeix al node i mira el contingut del fitxer /etc/fstab. Tens 2 cops la línia

#192.168.1.175:/home /mnt/nfs/home nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0









que hauríem de fer perquè quan torni a executar el blockinfile si estan afegides no les escrigui?
Un cop posat el marker, prova que executes el playbook i no afegeix les línies (adjunta la captura.)

anto@Controler: ~

```
  GNU nano 7.2                                roles/fstab2/tasks/main.yml *
---
- name: Añadir lineas a /etc/fstab
  blockinfile:
    path: /etc/fstab
    block: |
      #192.168.0.178:/home /mnt/nfs/home nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
      #192.168.0.178:/share /mnt/nfs/share nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
    marker: "# {mark} ANSIBLE MANAGED BLOCK"
```

anto@Controler: ~

```
root@Controler:~/Exer3DebianNginx# nano roles/fstab2/tasks/main.yml
root@Controler:~/Exer3DebianNginx# ansible-playbook -i inventory/host.ini playbook.yml

PLAY [debian] ******************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
ok: [clnt1]

TASK [fstab2 : Añadir lineas a /etc/fstab] *************************************************************************************
ok: [clnt1]

PLAY RECAP *********************************************************************************************************************
clnt1                      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@Controler:~/Exer3DebianNginx#
```

deb12G (Debian "limpio") [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal 13 de mar 23:30

user@node: ~

```
  GNU nano 7.2                                  /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=de5b0318-9ea6-41dc-bbd2-34a58f337e82 /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=3b65ac0f-56b7-4de5-ac64-d558f2c2f5c5 none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
#192.168.0.178:/home /mnt/nfs/home nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
# BEGIN ANSIBLE MANAGED BLOCK
#192.168.0.178:/home /mnt/nfs/home nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
#192.168.0.178:/share /mnt/nfs/share nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
# END ANSIBLE MANAGED BLOCK
```

Opcionals

Opcional 1. Busca informació sobre Ansible Galaxy i digues per què creus que la podriem fer servir.

Opcional 2. Posa en marxa Ansible Automation Platform. Pots descarregar la versió de proves aquí. Documenta la seva instal·lació i execució d'uns dels exercicis anteriors que has fet des d'aquesta Interfície.
Indica tots els links fets servir per realitzar la documentació.

Opcional 3. Investiga l'editor lapce.dev i compara'l a visual code.
Pots escriure a 5 línies alhora per exemple per comentar-les?
Pots reemplaçar una paraula a tot el fitxer?
explica més utilitats que creguis imprescindibles?