

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.

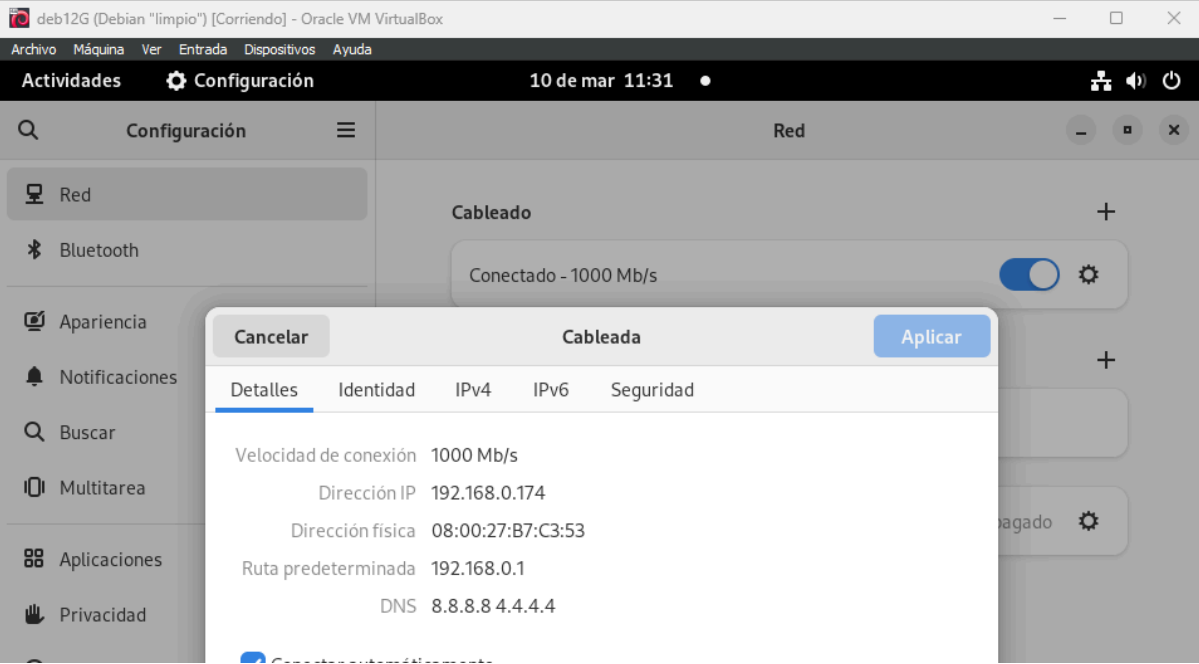
Adjunta la captura de les IPs (ip a), DNSs (cat /etc/resolv.conf), gateway (ip ro), nom de màquina, i cat /etc/apt/sources.list en el cas de Linux. Posa les captures corresponents de windows també. **(7 punts)**

Debian Server (Controler)

```
anto@Controler: ~  
root@Controler:~# ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:51:7d:38 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.0.178/22 brd 192.168.3.255 scope global enp0s3  
        valid_lft forever preferred_lft forever  
    inet6 fe80::a00:27ff:fe51:7d38/64 scope link  
        valid_lft forever preferred_lft forever  
root@Controler:~#
```

```
anto@Controler: ~  
root@Controler:~# cat /etc/resolv.conf  
nameserver 8.8.8.8  
root@Controler:~# hostname  
Controler  
root@Controler:~# ip ro  
default via 192.168.0.1 dev enp0s3 onlink  
169.254.0.0/16 dev enp0s3 scope link metric 1000  
192.168.0.0/22 dev enp0s3 proto kernel scope link src 192.168.0.178  
root@Controler:~# cat /etc/apt/sources.list  
#deb cdrom:[Debian GNU/Linux 12.2.0 _Bookworm_ - Official amd64 DVD Binary-1 with  
firmware 20231007-10:29]/ bookworm main non-free-firmware  
  
deb http://deb.debian.org/debian/ bookworm main non-free-firmware  
deb-src http://deb.debian.org/debian/ bookworm main non-free-firmware  
  
deb http://security.debian.org/debian-security bookworm-security main non-free-firmware  
deb-src http://security.debian.org/debian-security bookworm-security main non-free-firmware  
  
# bookworm-updates, to get updates before a point release is made;  
# see https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#_updates_and_backports  
deb http://deb.debian.org/debian/ bookworm-updates main non-free-firmware  
deb-src http://deb.debian.org/debian/ bookworm-updates main non-free-firmware  
root@Controler:~#
```

Debian12 (Node)



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

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Configuración 10 de mar 11:31

Red

Bluetooth

Apariencia

Notificaciones

Buscar

Multitarea

Aplicaciones

Privacidad

Cableado

Conectado - 1000 Mb/s

Cancelar Cableada Aplicar

Detalles Identidad IPv4 IPv6 Seguridad

Velocidad de conexión 1000 Mb/s

Dirección IP 192.168.0.174

Dirección física 08:00:27:B7:C3:53

Ruta predeterminada 192.168.0.1

DNS 8.8.8.8 4.4.4.4

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

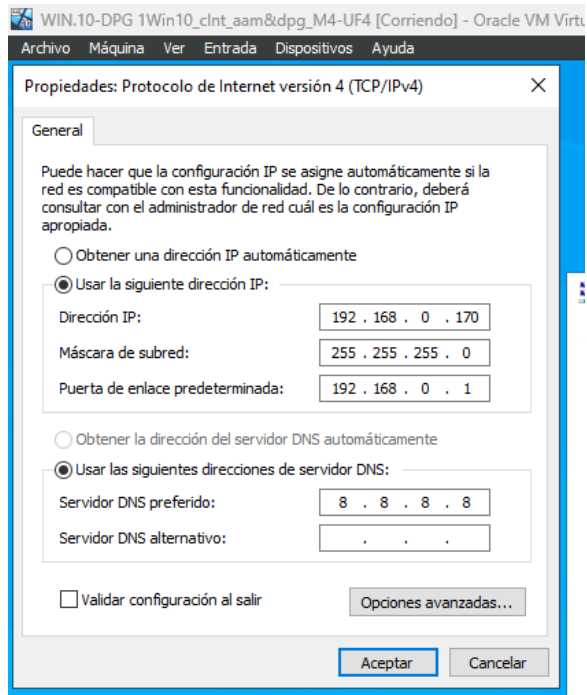
Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal 10 de mar 11:27

user@deb12G: ~

```
root@deb12G:~# hostname
node
root@deb12G:~#
```

Windows 10



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

Deb12C-Ansible [Corriendo] - Oracle VM VirtualBox

```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
root@Controler:~# apt install openssh-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libwrap0 openssh-sftp-server runit-helper
Paquetes sugeridos:
  molly-guard monkeysphere ssh-askpass ufw
Se instalarán los siguientes paquetes NUEVOS:
  libwrap0 openssh-server openssh-sftp-server runit-helper
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 584 kB de archivos.
Se utilizarán 2.327 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s_
```

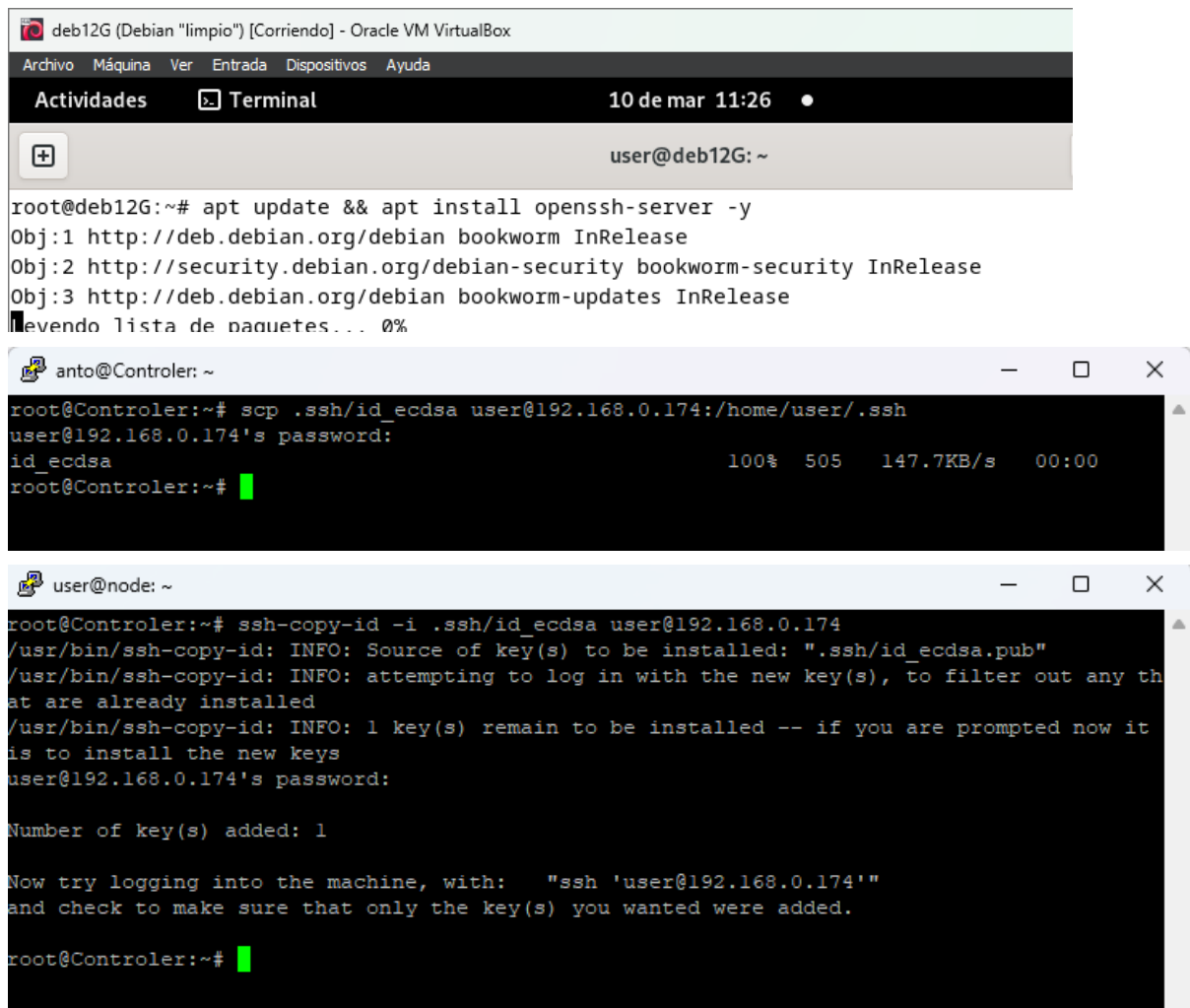
Deb12C-Ansible [Corriendo] - Oracle VM VirtualBox

```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
root@Controler:~# ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/root/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ecdsa
Your public key has been saved in /root/.ssh/id_ecdsa.pub
The key fingerprint is:
SHA256:V+18I9hC394LFCAQi0XGBBHmLa9fekMdtwJuV6sQdv4 root@Controler
The key's randomart image is:
+----[ECDSA 256]----+
|      =BBo.  .      |
|    o =..  .  .  .  |
|  + o   ....      |
|    o   ..+oo      |
|   +S..o =oo.     |
|   o.=. .o o.o    |
|  ..o.*....  .    |
|   .o*o=..  .  .  |
|   .o.ooE   .     |
+----[SHA256]-----+
root@Controler:~#
```

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):



The image displays three terminal windows from a VirtualBox environment. The top window is titled 'deb12G (Debian "limpio") [Corriendo] - Oracle VM VirtualBox' and shows the installation of OpenSSH server on a Debian 12 VM. The user is root, and the command 'apt update && apt install openssh-server -y' is executed. The output shows the update of package lists and the installation of openssh-server. The middle window is titled 'anto@Controler: ~' and shows the copying of the local SSH key to the remote node. The command 'scp .ssh/id_ecdsa user@192.168.0.174:/home/user/.ssh' is executed, and the output shows the successful transfer of the id_ecdsa key. The bottom window is titled 'user@node: ~' and shows the installation of the copied key on the remote node. The command 'ssh-copy-id -i .ssh/id_ecdsa user@192.168.0.174' is executed, and the output shows the successful installation of the key. The user is prompted to enter their password, and the output shows the successful installation of the key.

```
deb12G (Debian "limpio") [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 10 de mar 11:26
user@deb12G: ~
root@deb12G:~# apt update && apt install openssh-server -y
Obj:1 http://deb.debian.org/debian bookworm InRelease
Obj:2 http://security.debian.org/debian-security bookworm-security InRelease
Obj:3 http://deb.debian.org/debian bookworm-updates InRelease
Levendo lista de paquetes... 0%

anto@Controler: ~
root@Controler:~# scp .ssh/id_ecdsa user@192.168.0.174:/home/user/.ssh
user@192.168.0.174's password:
id_ecdsa 100% 505 147.7KB/s 00:00
root@Controler:~#

user@node: ~
root@Controler:~# ssh-copy-id -i .ssh/id_ecdsa user@192.168.0.174
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/id_ecdsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any th
at are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
is to install the new keys
user@192.168.0.174's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'user@192.168.0.174'"
and check to make sure that only the key(s) you wanted were added.

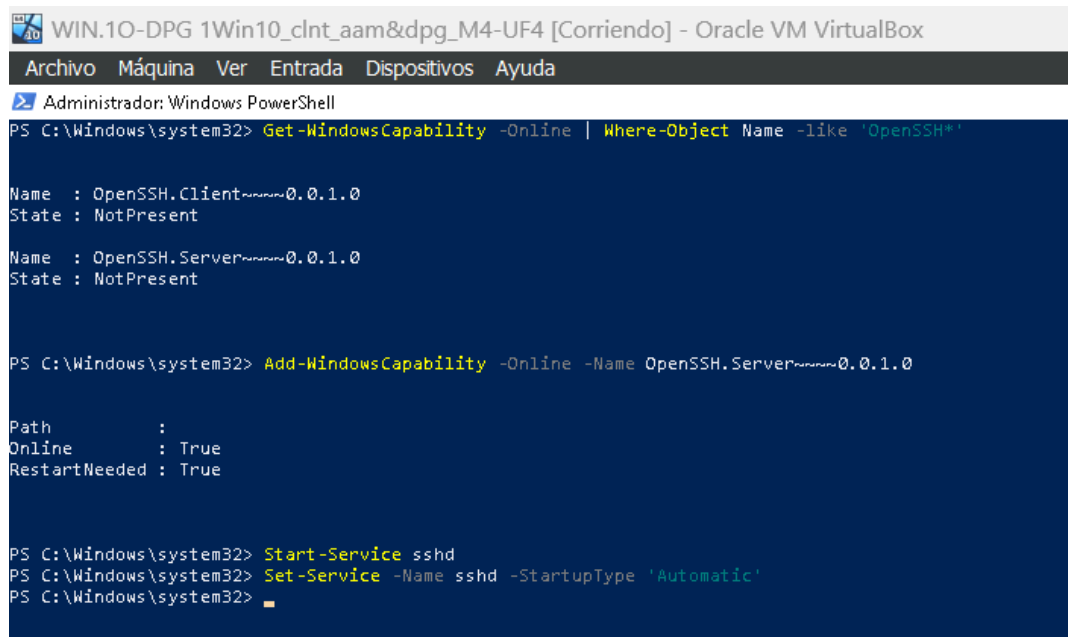
root@Controler:~#
```

```
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:



```
WIN.10-DPG 1Win10_clnt_aam&dpg_M4-UF4 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Administrador: Windows PowerShell
PS C:\Windows\system32> Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'

Name : OpenSSH.Client~~~~~0.0.1.0
State : NotPresent

Name : OpenSSH.Server~~~~~0.0.1.0
State : NotPresent

PS C:\Windows\system32> Add-WindowsCapability -Online -Name OpenSSH.Server~~~~~0.0.1.0

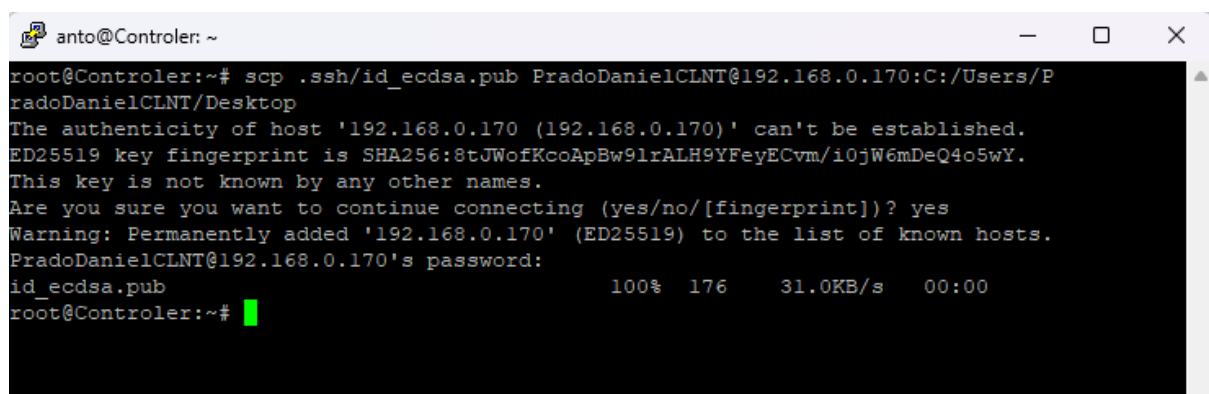
Path      :
Online    : True
RestartNeeded : True

PS C:\Windows\system32> Start-Service sshd
PS C:\Windows\system32> Set-Service -Name sshd -StartupType 'Automatic'
PS C:\Windows\system32>
```

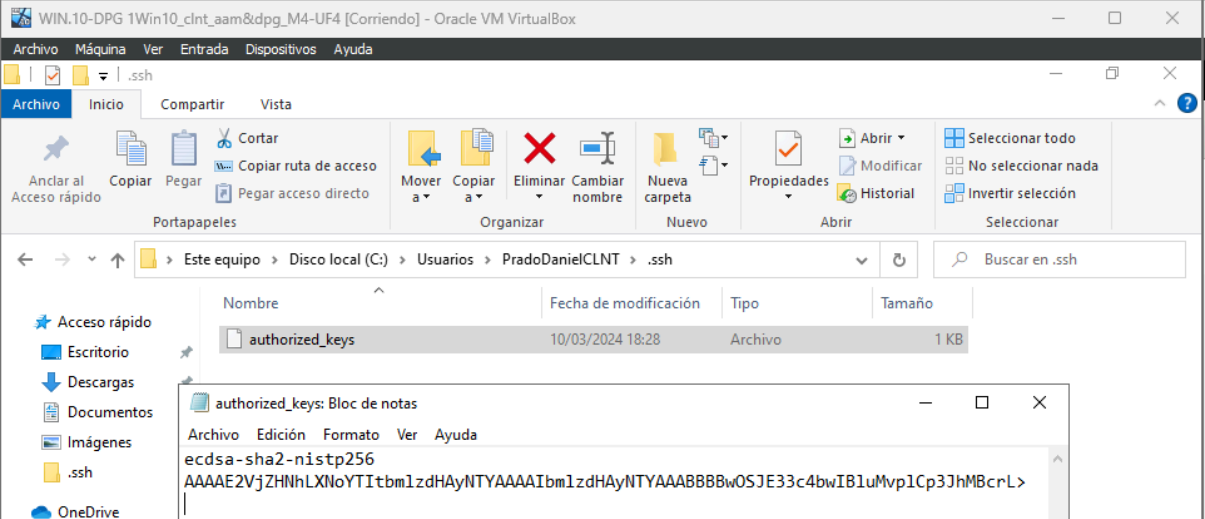
Copia la clau pública que has generat al teu Controller als 2 nodes. **(2 punts)**

→**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.



```
anto@Controler: ~
root@Controler:~# scp .ssh/id_ecdsa.pub PradoDanielCLNT@192.168.0.170:C:/Users/PradoDanielCLNT/Desktop
The authenticity of host '192.168.0.170 (192.168.0.170)' can't be established.
ED25519 key fingerprint is SHA256:8tJWofKcoApBw9lrALH9YFeyECvm/i0jW6mDeQ4o5wY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.170' (ED25519) to the list of known hosts.
PradoDanielCLNT@192.168.0.170's password:
id_ecdsa.pub                                100% 176    31.0KB/s   00:00
root@Controler:~#
```

```
Administrador: C:\Windows\system32\conhost.exe
root@Controler:~# ls -l .ssh/
total 16
-rw----- 1 root root 505 mar  5 16:41 id_ecdsa
-rw-r--r-- 1 root root 176 mar  5 16:41 id_ecdsa.pub
-rw----- 1 root root 364 mar  7 20:01 known_hosts
-rw-r--r-- 1 root root 142 mar  7 20:01 known_hosts.old
root@Controler:~#
```

```
Administrador: C:\Windows\system32\conhost.exe
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\PradoDanielCLNT>
```

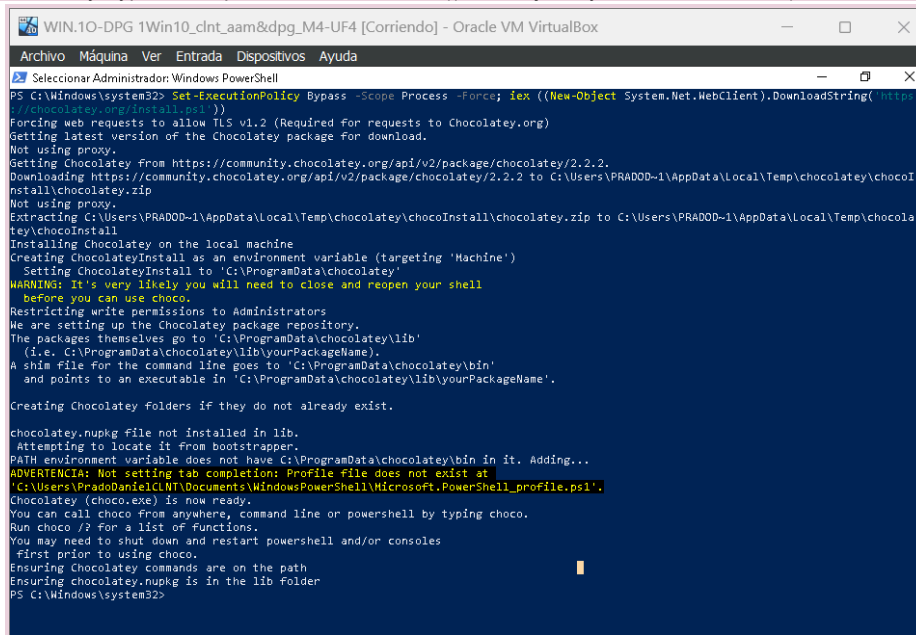
NOTA 2: [que és chocolatey](#) , [chocolatey en Ansible](#)

→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` , que se usa 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`.

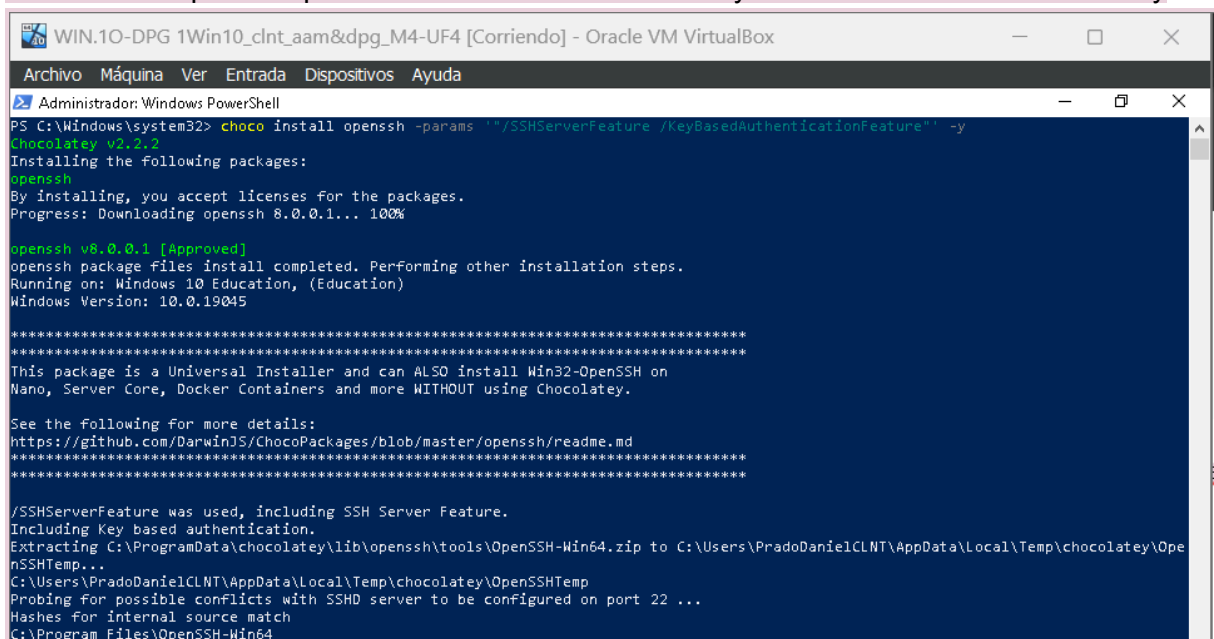
NOTA 3: comandes instal·lació chocolatey

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



```
PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
Forcing web requests to allow TLS v1.2 (Required for requests to Chocolatey.org)
Getting latest version of the Chocolatey package for download.
Not using proxy.
Setting Chocolatey from https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2
Downloading https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2 to C:\Users\PRADOD~1\AppData\Local\Temp\chocolatey\chocoI
nstall\chocolatey.zip
Extracting C:\Users\PRADOD~1\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip to C:\Users\PRADOD~1\AppData\Local\Temp\chocola
tey\chocoInstall
Installing Chocolatey on the local machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.
Creating Chocolatey Folders if they do not already exist.
chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
ADVERTENCIA: Not setting tab completion: Profile file does not exist at
'C:\Users\PradoDanielCLNT\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? For a list of Functions.
You may need to shut down and restart powershell and/or consoles
first prior to using choco.
Ensuring Chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\Windows\system32>
```

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



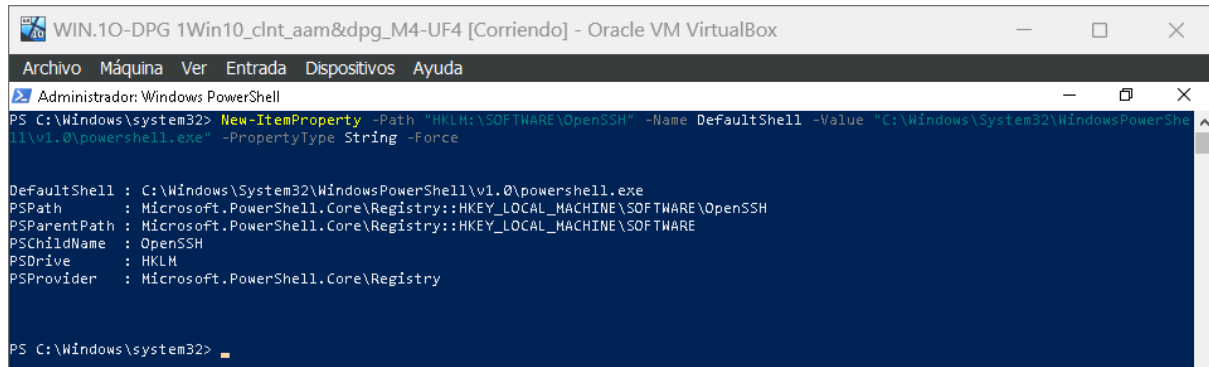
```
Administrator: Windows PowerShell
PS C:\Windows\system32> choco install openssh -params "/SSHServerFeature /KeyBasedAuthenticationFeature" -y
Chocolatey v2.2.2
Installing the following packages:
openssh
By installing, you accept licenses for the packages.
Progress: Downloading openssh 8.0.0.1... 100%
openssh v8.0.0.1 [Approved]
openssh package files install completed. Performing other installation steps.
Running on: Windows 10 Education, (Education)
Windows Version: 10.0.19045

*****
This package is a Universal Installer and can ALSO install Win32-OpenSSH on
Nano, Server Core, Docker Containers and more WITHOUT using Chocolatey.

See the following for more details:
https://github.com/DarwinJS/ChocoPackages/blob/master/openssh/readme.md
*****

/SSHServerFeature was used, including SSH Server Feature.
Including Key based authentication.
Extracting C:\ProgramData\chocolatey\lib\openssh\tools\OpenSSH-Win64.zip to C:\Users\PradoDanielCLNT\AppData\Local\Temp\chocolatey\Ope
nSSHTemp...
C:\Users\PradoDanielCLNT\AppData\Local\Temp\chocolatey\OpenSSHTemp
Probing for possible conflicts with SSHD server to be configured on port 22 ...
Hashes for internal source match
C:\Program Files\OpenSSH-Win64
```

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



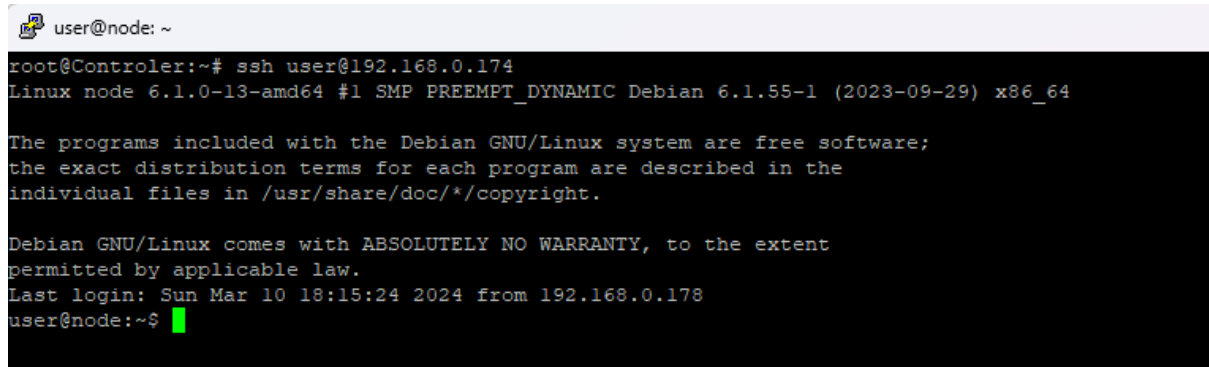
```
WIN.10-DPG 1Win10_clnt_aam&dpg_M4-UF4 [Corriendo] - Oracle VM VirtualBox
Administrador: Windows PowerShell
PS C:\Windows\system32> New-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell -Value "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -PropertyType String -Force

DefaultShell : C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\OpenSSH
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE
PSChildName  : OpenSSH
PSDrive      : HKLM
PSProvider   : Microsoft.PowerShell.Core\Registry

PS C:\Windows\system32>
```

Prova que funciona la connexió ssh des del teu controller als nodes fent servir la Key. (2 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 18:15:24 2024 from 192.168.0.178
user@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'inventari 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: ~  
root@Controler:~# mkdir Exer1DebianNode  
root@Controler:~# cd Exer1DebianNode  
root@Controler:~/Exer1DebianNode# tree  
.  
  
0 directories, 0 files  
root@Controler:~/Exer1DebianNode# exit
```

```
anto@Controler: ~  
root@Controler:~/Exer1DebianNode# mkdir -p inventory roles/clean/tasks roles/install_compressors/tasks roles/update_upgrade_clean/tasks  
root@Controler:~/Exer1DebianNode# touch inventory/hosts.ini playbook.yml roles/clean/tasks/main.yml roles/install_compressors/tasks/main.yml roles/update_upgrade_clean/tasks/main.yml  
root@Controler:~/Exer1DebianNode# tree  
.  
├── inventory  
│   └── hosts.ini  
├── playbook.yml  
└── roles  
    ├── clean  
    │   └── tasks  
    │       └── main.yml  
    ├── install_compressors  
    │   └── tasks  
    │       └── main.yml  
    └── update_upgrade_clean  
        └── tasks  
            └── main.yml  
  
9 directories, 5 files  
root@Controler:~/Exer1DebianNode#
```

```
anto@Controler: ~  
inventory/hosts.ini *  
[debian]  
192.168.0.174
```

```
anto@Controler: ~  
GNU nano 7.2 roles/clean/tasks/main.yml *  
---  
- name: Clean Up Unnecessary Packages and Cache  
  apt:  
    autoremove: yes  
    autoclean: yes
```

```
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

```
anto@Controler: ~  
    "purge": false,  
    "state": "present",  
    "update_cache": null,  
    "update_cache_retries": 5,  
    "update_cache_retry_max_delay": 12,  
    "upgrade": null  
  },  
  "stderr": "",  
  "stderr_lines": [],  
  "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\n",  
  "stdout_lines": [  
    "Reading package lists...",  
    "Building dependency tree...",  
    "Reading state information..."  
  ]  
}  
  
PLAY RECAP *****  
192.168.0.174      : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
root@Controler:~/Exer1DebianNode#
```

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

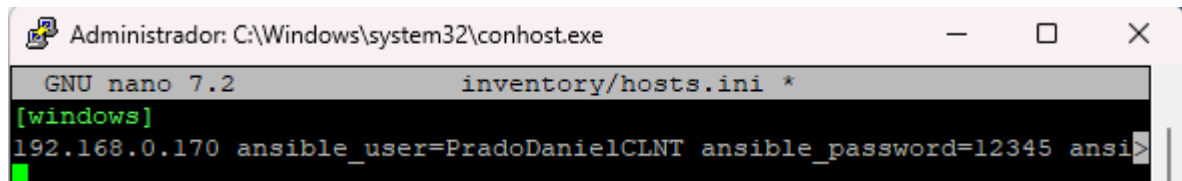
```
anto@Controler: ~  
root@Controler:~/Exer2WinNode# tree  
.  
├── inventory  
│   └── hosts.ini  
├── playbook.yml  
├── roles  
│   └── install_base  
│       └── task  
│           └── main.yml  
└──  
  
5 directories, 3 files  
root@Controler:~/Exer2WinNode#
```

```
anto@Controler: ~  
root@Controler:~# apt install sshpass  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
sshpass ya está en su versión más reciente (1.09-1+b1).  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.  
root@Controler:~#
```

Configuración con key (no me estaba funcionando)

```
anto@Controler: ~  
GNU nano 7.2 inventory/hosts.ini  
[windows]  
windows_cliente ansible_host=192.168.0.170 ansible_user=PradoDanielCLNT  
  
[all:vars]  
ansible_ssh_private_key_file=~/.ssh/id_ecdsa
```


Configuración con contraseña (me ha funcionado)



```
Administrador: C:\Windows\system32\conhost.exe
GNU nano 7.2 inventory/hosts.ini *
[windows]
192.168.0.170 ansible_user=PradoDanielCLNT ansible_password=12345 ansi>
```

Comando completo (no se llega a ver en la captura)

[windows]

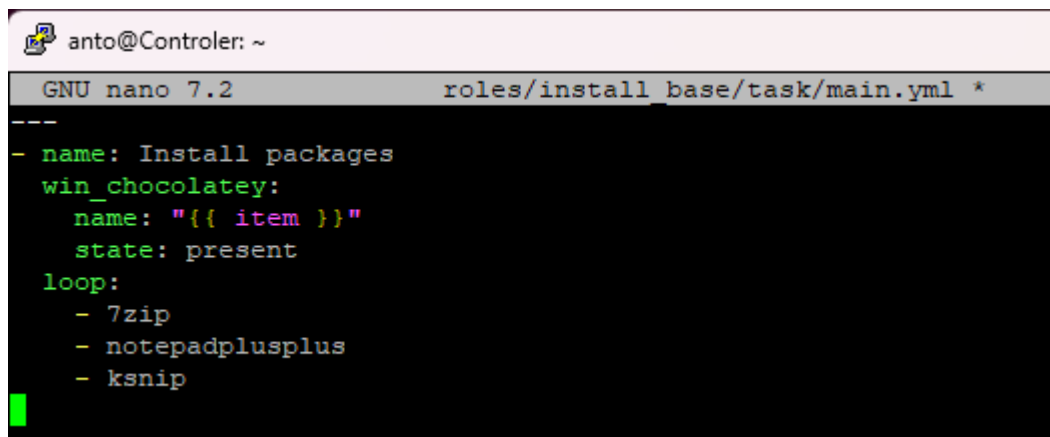
192.168.0.170 ansible_user=PradoDanielCLNT ansible_password=12345

ansible_connection=ssh ansible_shell_type=powershell

ansible_winrm_server_cert_validation=ignore



```
anto@Controler: ~
GNU nano 7.2 playbook.yml *
---
- hosts: windows
  roles:
    - install_base
```



```
anto@Controler: ~
GNU nano 7.2 roles/install_base/task/main.yml *
---
- name: Install packages
  win_chocolatey:
    name: "{{ item }}"
    state: present
  loop:
    - 7zip
    - notepadplusplus
    - ksnip
```

```
anto@Controler: ~  
l 1: rcvd close\r\nndebug2: channel 1: output open -> drain\r\nndebug2: chan_shutdown_read: chan  
nel 1: (i0 o1 sock 5 wfd 5 efd -1 [closed])\r\nndebug2: channel 1: input open -> closed\r\nndebu  
g2: channel 2: gc: user detached\r\nndebug2: channel 2: is dead\r\nndebug2: channel 2: garbage c  
ollecting\r\nndebug1: channel 2: free: client-session, nchannels 3\r\nndebug3: channel 2: status  
: The following connections are open:\r\n #1 mux-control (tl6 [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\nndebug2: channel 1: obuf empty\r\nnd  
ebug2: chan_shutdown_write: channel 1: (i3 o1 sock 5 wfd 5 efd -1 [closed])\r\nndebug3: mux_cli  
ent_read_packet: read header failed: Broken pipe\r\nndebug2: Received exit status from master 0  
\r\nndebug2: channel 1: output drain -> closed\r\nndebug2: channel 1: is dead (local)\r\nndebug2:  
channel 1: gc: notify user\r\nndebug3: mux_master_control_cleanup_cb: entering for channel 1\r\n\r\nndebug2: channel 1: gc: user detached\r\nndebug2: channel 1: is dead (local)\r\nndebug2: channe  
l 1: garbage collecting\r\nndebug1: channel 1: free: mux-control, nchannels 2\r\nndebug3: channe  
l 1: status: The following connections are open:\r\n #1 mux-control (tl6 [mux-control] nr0 i3  
/0 o3/0 e[closed]/0 fd 5/5/-1 sock 5 cc -1 io 0x00/0x00)\r\n\r\nndebug2: set_control_persist_ex  
it_time: schedule exit in 60 seconds\r\n\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 webserver 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):

```
anto@Controler: ~  
root@Controler:~/Exer3DebianNginx# tree  
.  
├── group_vars  
│   └── all  
├── inventory  
│   └── host.ini  
├── playbook.yml  
└── roles  
    ├── condiciones  
    │   └── tasks  
    │       └── main.yml  
    ├── nginx_install  
    │   └── tasks  
    │       └── main.yml  
    └── template  
        ├── tasks  
        │   └── main  
        ├── templates  
        └── index.html.j2  
  
12 directories, 6 files
```

```
anto@Controler: ~  
GNU nano 7.2 inventory/host.ini *  
[debian]  
clnt1 ansible_host=192.168.0.174
```

```
anto@Controler: ~  
GNU nano 7.2 playbook.yml  
---  
- hosts: debian  
  become: yes  
  roles:  
    - condiciones  
    - nginx_install  
    - template
```

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 reiniciï el servei nginx

Executa el playbook amb els roles pertinents pel teu inventari.

```
anto@Controler: ~
GNU nano 7.2 roles/nginx_install/tasks/main.yml
--
- name: Actualizar todos los paquetes a la última versión
  apt:
    update_cache: yes
    cache_valid_time: 3600
    upgrade: yes

- name: Instalar nginx
  apt:
    name: nginx
    state: present

- name: Eliminar los paquetes que no se utilizan
  command: apt-get autoremove -y

- name: Reiniciar el servicio nginx
  systemd:
    name: nginx
    state: restarted
```

```
anto@Controler: ~
GNU nano 7.2 roles/nginx_install/task/main.yml
---
- name: Actualizar todos los paquetes a la última versión
  apt:
    update_cache: yes
    cache_valid_time: 3600
    upgrade: yes

- name: Instalar nginx
  apt:
    name: nginx
    state: present

- name: Eliminar los paquetes que no se utilizan
  command: apt-get autoremove -y

- name: Reiniciar el servicio nginx
  systemd:
    name: nginx
    state: restarted
```

```
anto@Controler: ~  
GNU nano 7.2      playbook.yml *  
--  
- hosts: debian  
  become: yes  
  roles:  
    - role: nginx_install
```

```
anto@Controler: ~  
root@Controler:~/Exer3DebianNginx# ansible-playbook -i inventory/host.ini playbook.yml  
  
PLAY [debian] *****  
  
TASK [Gathering Facts] *****  
ok: [clnt1]  
  
TASK [nginx_install : Actualizar todos los paquetes a la última versión] *****  
ok: [clnt1]  
  
TASK [nginx_install : Instalar nginx] *****  
changed: [clnt1]  
  
TASK [nginx_install : Eliminar los paquetes que no se utilizan] *****  
changed: [clnt1]  
  
TASK [nginx_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, rsa ...)`)
- 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'hauré 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, rsa ...)`) 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                                roles/condiciones/tasks/main.yml
--
- name: Registrar variables
  set_fact:
    ssh_file: /root/.ssh/id_ecdsa.pub.Controller
    ssh_folder: /root/.ssh

- name: Comprobar si existe el directorio
  stat:
    path: "{{ ssh_folder }}"
  register: folder_stat

- name: Mostrar si el directorio existe
  debug:
    msg: "Folder exists"
  when: folder_stat.stat.exists

- name: Mostrar si el directorio no existe
  debug:
    msg: "Folder NOT exists!!!"
  when: not folder_stat.stat.exists

- name: Crear el directorio si no existe
  file:
    path: "{{ ssh_folder }}"
    state: directory
    owner: root
    group: root
    mode: '0700'
  when: not folder_stat.stat.exists

- name: Comprobar si existe el archivo
  stat:
    path: "{{ ssh_file }}"
  register: file_stat

- name: Mostrar si el archivo existe
  debug:
    msg: "File exists"
  when: file_stat.stat.exists

- name: Mostrar si el archivo no existe
  debug:
    msg: "File NOT exists!!!"
  when: not file_stat.stat.exists

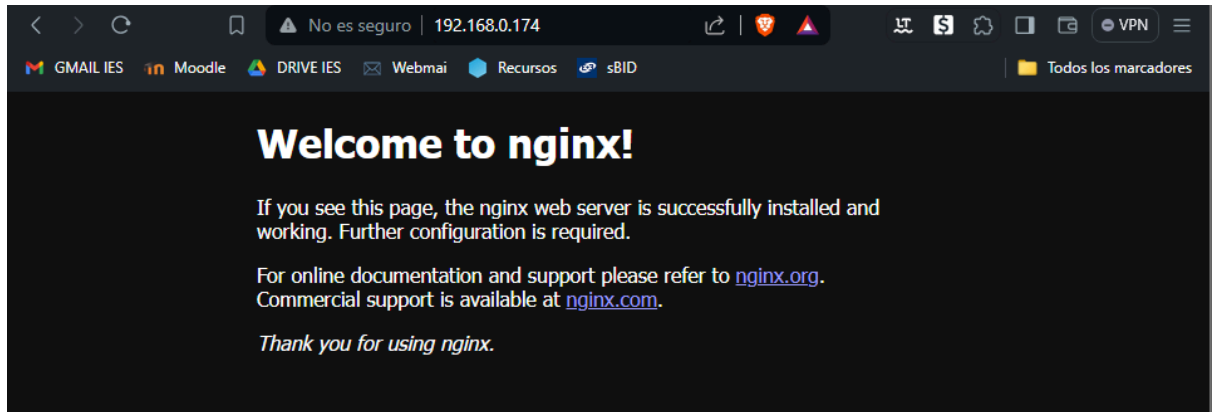
- name: Copiar el archivo si no existe
  copy:
    src: /root/.ssh/id_ecdsa.pub
    dest: "{{ ssh_file }}"
    owner: root
    group: root
    mode: '0644'
  when: not file_stat.stat.exists
```

```
anto@Controler: ~  
GNU nano 7.2                                playbook.yml *  
---  
- hosts: debian  
  become: yes  
  roles:  
    - role: nginx_install  
    - role: condiciones  
█
```

```
anto@Controler: ~  
root@Controler:~/Exer3DebianNginx# ansible-playbook -i inventory/host.ini playbook.yml  
  
PLAY [debian] *****  
  
TASK [Gathering Facts] *****  
ok: [clnt1]  
  
TASK [nginx_install : Actualizar todos los paquetes a la última versión] *****  
ok: [clnt1]  
  
TASK [nginx_install : Instalar nginx] *****  
ok: [clnt1]  
  
TASK [nginx_install : Eliminar los paquetes que no se utilizan] *****  
changed: [clnt1]  
  
TASK [nginx_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

```
anto@Controler: ~
GNU nano 7.2 roles/template/templates/index.html.j2 *
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title> My Nginx</title>
</head>
<body>
  <p>
    "Aquesta és la web de Nginx creada per {{ variable_Nom }} al teu controller"
  </p>
  <p>
    "Aquesta és la IP del servidor Nginx {{ inventory_hostname }} extreta des d'una variable predefinida d'ansible"
  </p>
</body>
</html>
```


- 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
```

```
anto@Controler: ~
root@Controler:~/Exer3DebianNginx# tree
.
├── group_vars
│   └── all
├── inventory
│   └── host.ini
└── playbook.yml
```

```
anto@Controler: ~
GNU nano 7.2 group_vars/all *
---
variable Nom: "Antonio Adamuz"
```

- Fent servir un template dins d'un role anomenat template, sobreesciu 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)

```
anto@Controler: ~
GNU nano 7.2 roles/template/tasks/main.yml *
---
- name: Sobrescribir el archivo /var/www/html/index.html con el template
  template:
    src: /root/Exer3DebianNginx/roles/template/templates/index.html.j2
    dest: /var/www/html/index.html

- name: Reiniciar el servicio nginx
  systemd:
    name: nginx
    state: restarted
```

```
anto@Controler: ~
GNU nano 7.2 playbook.yml *
---
- hosts: debian
  become: yes
  roles:
    - role: nginx_install
    - role: condiciones
    - role: template
```

```

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

PLAY [debian] *****

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

TASK [nginx_install : Actualizar todos los paquetes a la última versión] *****
ok: [clnt1]

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

TASK [nginx_install : Eliminar los paquetes que no se utilizan] *****
changed: [clnt1]

TASK [nginx_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]

TASK [template : Sobrescribir el archivo /var/www/html/index.html con el template] *****
changed: [clnt1]

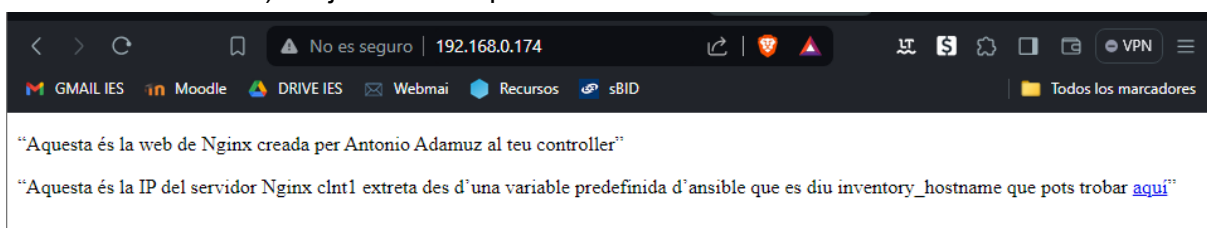
TASK [template : Reiniciar el servicio nginx] *****
changed: [clnt1]

PLAY RECAP *****
clnt1                : ok=12   changed=4   unreachable=0   failed=0   skipped=4   rescued=0   ignored=0

root@Controler: ~/Exer3DebianNginx#

```

- 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

```
anto@Controler: ~
GNU nano 7.2 roles/fstab1/tasks/main.yml *
---
- name: Afegir linea a /etc/fstab
  lineinfile:
    path: /etc/fstab
    line: '#192.168.0.178:/home /mnt/nfs/home nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0'
```

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

PLAY [debian] *****

TASK [Gathering Facts] *****
ok: [c1nt1]

TASK [fstab1 : Afegir linea a /etc/fstab] *****
changed: [c1nt1]

PLAY RECAP *****
c1nt1 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

root@Controler:~/Exer3DebianNginx#
```

```
deb12G (Debian "limpío") [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 13 de mar 23:23
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
```

(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
```

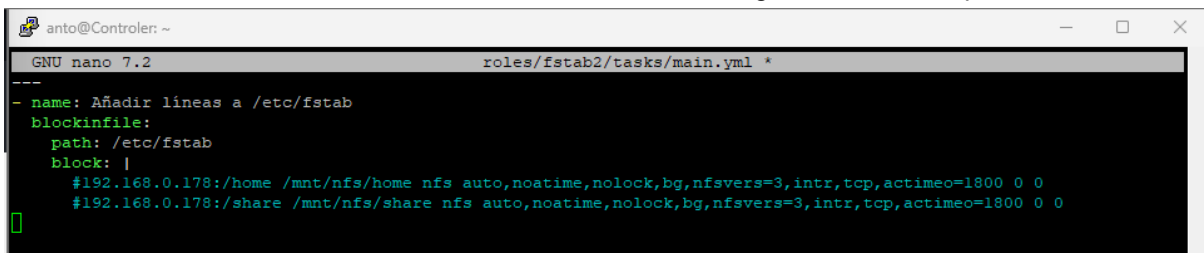
```
#192.168.1.175:/share /mnt/nfs/share nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
```

NOTA: les línies les has d'introduir amb el comentari per evitar errors

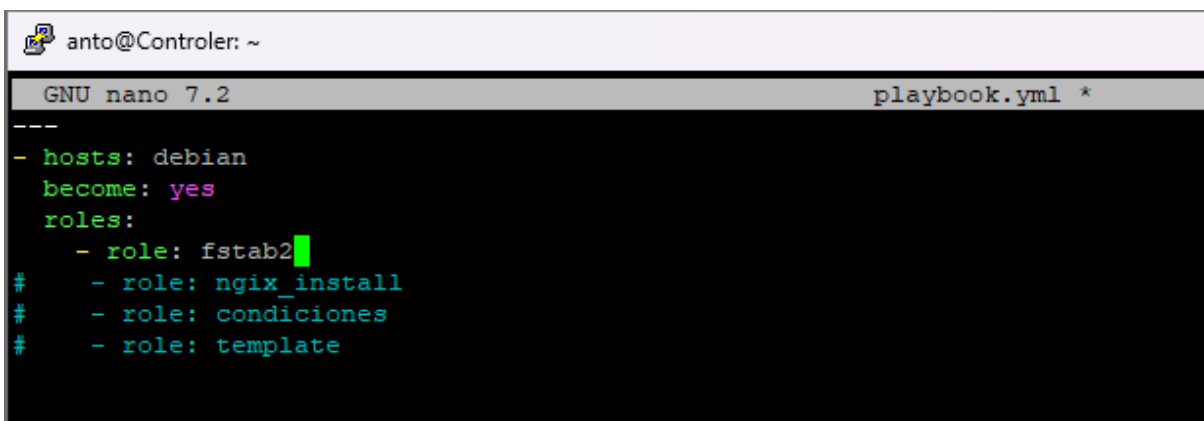
adjunta les captures que demostrin que has executat el role i ha funcionat

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
```



```
anto@Controler: ~
GNU nano 7.2 roles/fstab2/tasks/main.yml *
--
- name: Añadir líneas 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
[]
```



```
anto@Controler: ~
GNU nano 7.2 playbook.yml *
--
- hosts: debian
  become: yes
  roles:
    - role: fstab2
#    - role: nginx_install
#    - role: condiciones
#    - role: template
```

```
anto@Controler: ~  
root@Controler:~/Exer3DebianNginx# ansible-playbook -i inventory/host.ini playbook.yml  
PLAY [debian] *****  
TASK [Gathering Facts] *****  
ok: [clnt1]  
TASK [fstab2 : Añadir líneas a /etc/fstab] *****  
changed: [clnt1]  
PLAY RECAP *****  
clnt1 : ok=2 changed=1 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 23:28  
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
```

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 líneas 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 líneas 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 "limpío") [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 podríem 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?