

A thick dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date. Below the banner, several thin, curved lines in dark blue and light grey sweep upwards from the bottom left corner.

14-10-2025

Manuales de configuración de ansible

Práctica Profesional NRC-50-69370

Docente:

Julio Eduardo Valenzuela Diaz

Contenido

Introducción.....	2
Configuración del PC Esclavo.....	2
WinRM:	2
Verificación de la Configuración WinRM	3
Configuración de Permisos de Usuario de Dominio	3
Configuración Adicional de WinRM.....	5
Permitir Conexiones No Encriptadas	5
Configuración del PC Maestro	6
Instalación de Ubuntu desde Microsoft Store	6
Instalación de Windows Terminal	7
Configuración Inicial de Ubuntu	7
Actualización del Sistema Ubuntu	8
Instalación de Ansible y Python	9
Instalación de Librerías para WinRM.....	10
Configuración del Entorno de Desarrollo.....	10
Creación del Archivo de Inventario.....	11
Creación del Playbook de Prueba	12
Pruebas de Funcionamiento.....	12

Introducción

El Proyecto ORION (Optimización Remota e Inteligente de la Sala T-201) busca implementar un sistema de gestión centralizada para los computadores de la sala, permitiendo administrar múltiples equipos de forma remota desde una única estación de trabajo.

Arquitectura de la Solución

El sistema utiliza una arquitectura maestro-esclavo donde:

- **PCs Esclavos:** Equipos de la sala T-201 que reciben instrucciones
- **PC Maestro:** Equipo central que coordina y ejecuta las operaciones
- **WinRM:** Protocolo de Microsoft para la gestión remota de Windows
- **Ansible:** Herramienta de automatización que orquesta las tareas

Configuración del PC Esclavo

WinRM: Windows Remote Management es el servicio que permite la gestión remota de equipos Windows. Sin esta configuración, el PC Maestro no podría comunicarse con los esclavos para ejecutar tareas administrativas.

1. Ejecuta la configuración rápida de WinRM con el comando *winrm quickconfig*

El comando *winrm quickconfig* es la forma específica de Microsoft para configurar automáticamente WinRM. Este comando:

- Habilita el servicio WinRM
- Crea un listener HTTP en el puerto 5985
- Configura las reglas de firewall necesarias
- Establece la configuración básica de seguridad

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\windows\system32> winrm quickconfig
WinRM no está configurado para recibir solicitudes en este equipo.
Se deben realizar estos cambios:

Inicie el servicio WinRM.
Establezca el tipo de servicio WinRM en inicio automático aplazado.

¿Desea realizar estos cambios [y/n]? y
WinRM se ha actualizado para recibir solicitudes.

Se cambió el tipo de servicio WinRM correctamente.
Servicio WinRM iniciado.
WinRM no está configurado para permitir acceso remoto al equipo para administración.
Se deben realizar estos cambios:

Habilitar la excepción de firewall WinRM.

¿Desea realizar estos cambios [y/n]? y
WinRM se actualizó para administración remota.
Excepción de firewall WinRM habilitada.
PS C:\windows\system32>
```

Verificación de la Configuración WinRM

2. Verificar el estado del servicio WinRM con el comando *Get-Service WinRM*

```
Administrador: Windows PowerShell
PS C:\windows\system32> Get-Service WinRM

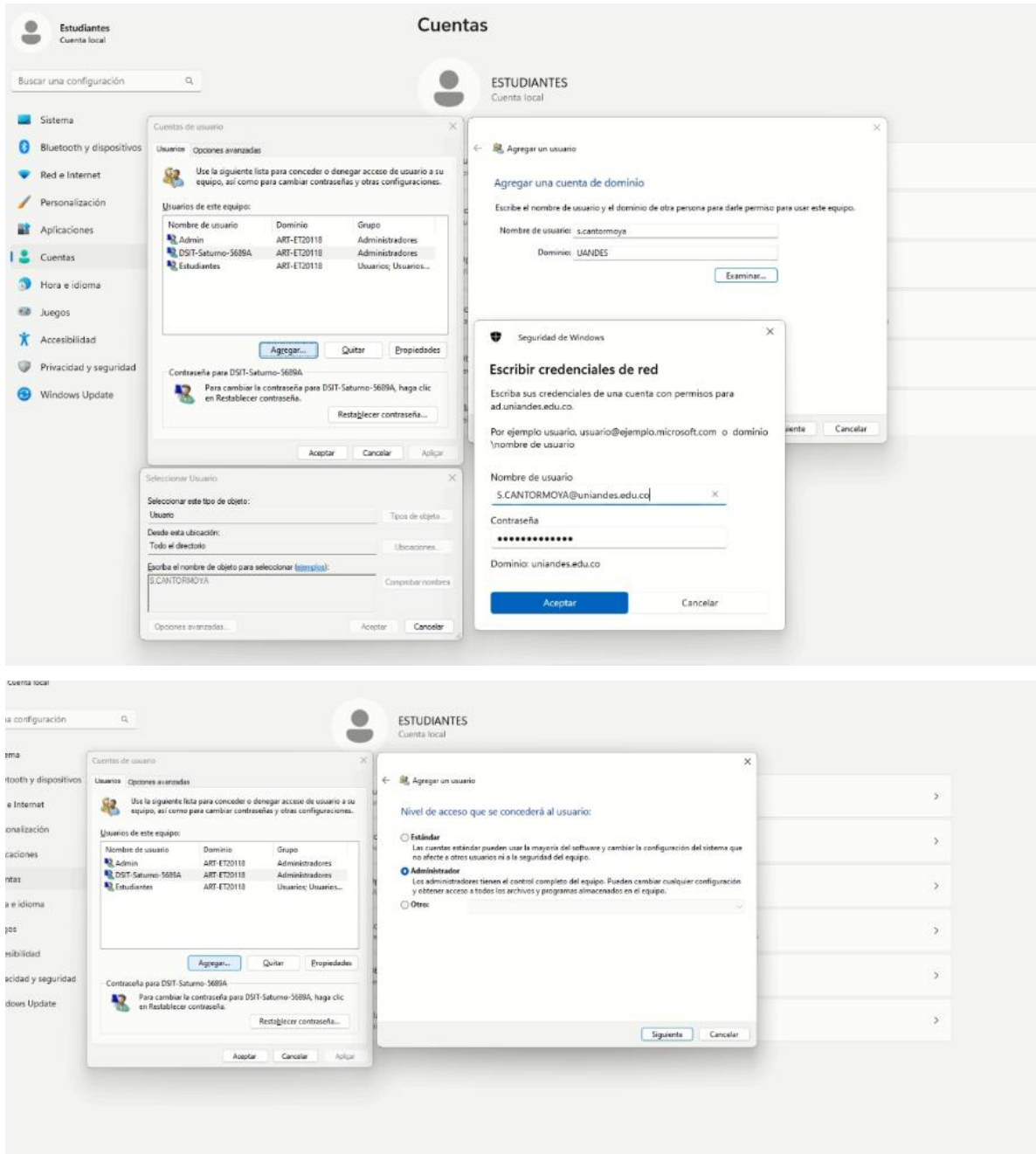
Status      Name      DisplayName
-----
Running     WinRM     Administración remota de Windows (W...
```

Configuración de Permisos de Usuario de Dominio

La herramienta netplwiz permite gestionar usuarios y grupos de forma gráfica. Es especialmente útil para técnicos que prefieren interfaces visuales sobre línea de comandos.

3. Procedimiento paso a paso:
 - Abrir la herramienta *netplwiz*:
 - Presiona Windows + R para abrir el cuadro de ejecución.
 - Escribe *netplwiz* y presiona Enter.
 - Damos clic en el botón agregar.
 - Colocamos el usuario como el nombre del dominio.
 - Buscamos el usuario dentro del directorio activo.

- Le asignamos el rol de administrador.



¿Por qué es necesario este paso?

El usuario de dominio debe ser administrador local en el PC esclavo para poder ejecutar comandos remotos a través de WinRM. Sin estos privilegios, la autenticación fallaría incluso con credenciales correctas.

Configuración Adicional de WinRM

4. Ejecuta el comando *winrm enumerate winrm/config/listener*

```
Administrador: Windows PowerShell
PS C:\windows\system32> winrm enumerate winrm/config/listener
Listener
  Address = *
  Transport = HTTP
  Port = 5985
  Hostname
  Enabled = true
  URLPrefix = wsman
  CertificateThumbprint
  ListeningOn = 127.0.0.1, 157.253.26.109, ::1, fe80::acf8:89fc:d6c9:37e9%9
PS C:\windows\system32> _
```

¿Qué hace este comando?

- Muestra todos los listeners de WinRM activos en el equipo
- Confirma que existe un listener en el puerto 5985 (HTTP)
- Verifica la configuración de seguridad del listener
- Permite identificar problemas de configuración

Permitir Conexiones No Encriptadas

5. Ejecutar el comando *winrm set winrm/config/service '@{AllowUnencrypted="true"}'*

```
Administrador: Windows PowerShell
PS C:\windows\system32> winrm set winrm/config/service '@{AllowUnencrypted="true"}'
Service
  RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;IU)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)
  MaxConcurrentOperations = 4294967295
  MaxConcurrentOperationsPerUser = 1500
  EnumerationTimeoutms = 240000
  MaxConnections = 300
  MaxPacketRetrievalTimeSeconds = 120
  AllowUnencrypted = true
  Auth
    Basic = true
    Kerberos = true
    Negotiate = true
    Certificate = false
    CredSSP = false
    CbtHardeningLevel = Relaxed
  DefaultPorts
    HTTP = 5985
    HTTPS = 5986
  IPv4Filter = *
  IPv6Filter = *
  EnableCompatibilityHttpListener = false
  EnableCompatibilityHttpsListener = false
  CertificateThumbprint
  AllowRemoteAccess = true
PS C:\windows\system32>
```

¿Qué hace este comando?

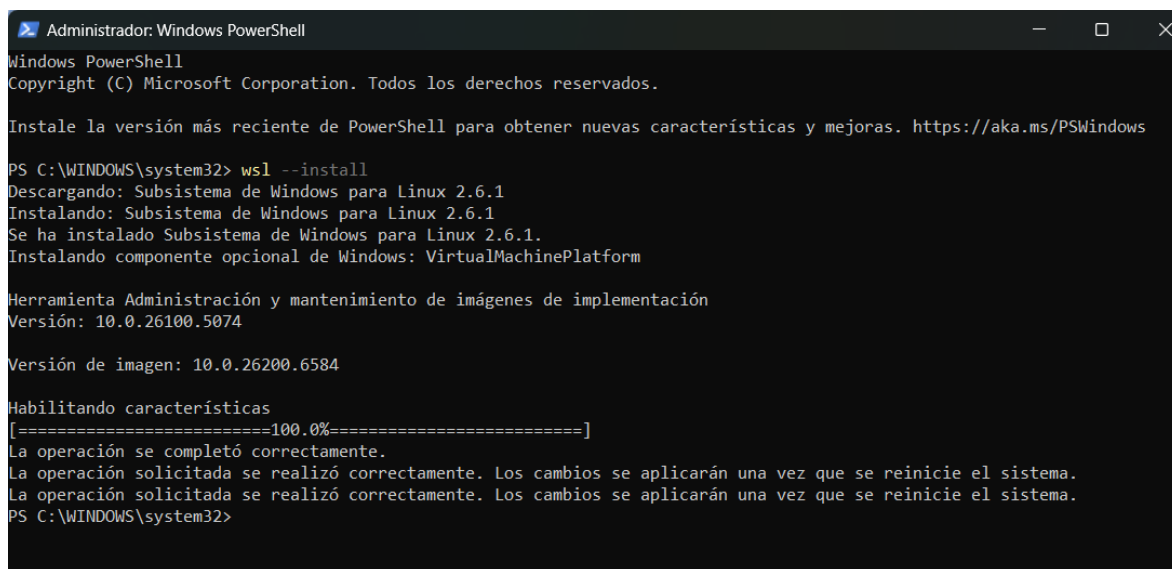
- Habilita la opción AllowUnencrypted en el servicio WinRM
- Permite que clientes se conecten sin requerir encriptación SSL
- Es necesario cuando se usa WinRM sobre HTTP (puerto 5985) sin certificados

¿Por qué es necesario en este contexto?

- Simplifica la configuración inicial en entornos de prueba
- Evita problemas de certificados SSL en la fase de desarrollo
- Permite que Ansible se conecte via HTTP sin complicaciones adicionales

Configuración del PC Maestro

1. Ejecuta el comando `wsl --install`



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

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --install
Descargando: Subsistema de Windows para Linux 2.6.1
Instalando: Subsistema de Windows para Linux 2.6.1
Se ha instalado Subsistema de Windows para Linux 2.6.1.
Instalando componente opcional de Windows: VirtualMachinePlatform

Herramienta Administración y mantenimiento de imágenes de implementación
Versión: 10.0.26100.5074

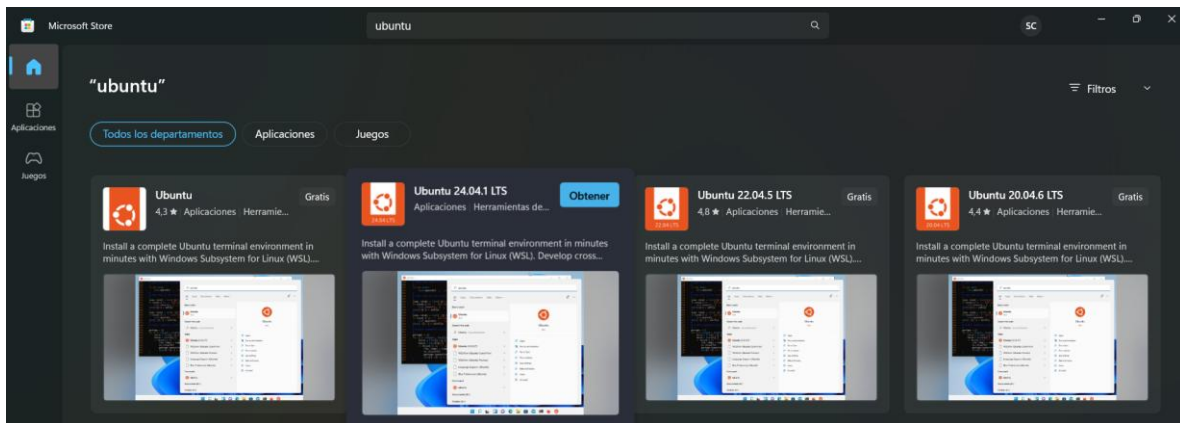
Versión de imagen: 10.0.26200.6584

Habilitando características
[=====100.0%=====]
La operación se completó correctamente.
La operación solicitada se realizó correctamente. Los cambios se aplicarán una vez que se reinicie el sistema.
La operación solicitada se realizó correctamente. Los cambios se aplicarán una vez que se reinicie el sistema.
PS C:\WINDOWS\system32>
```

Este comando instala WSL2 (Windows Subsystem for Linux versión 2) con la distribución Ubuntu por defecto. El sistema requerirá reiniciarse para completar la instalación.

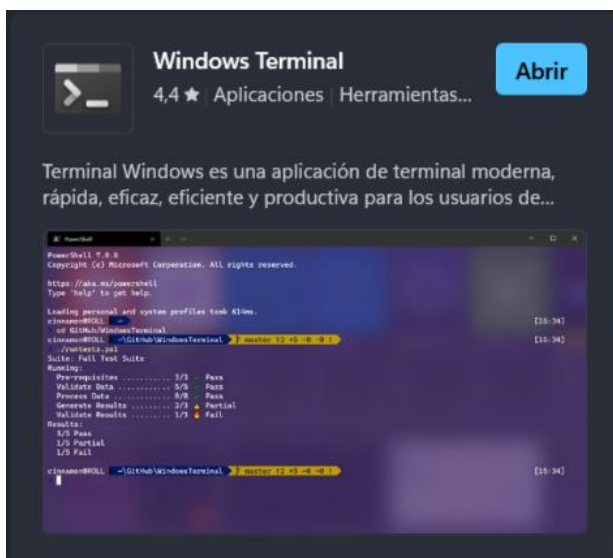
Instalación de Ubuntu desde Microsoft Store

2. Descargar la distribución de Linux de preferencia desde la Microsoft Store



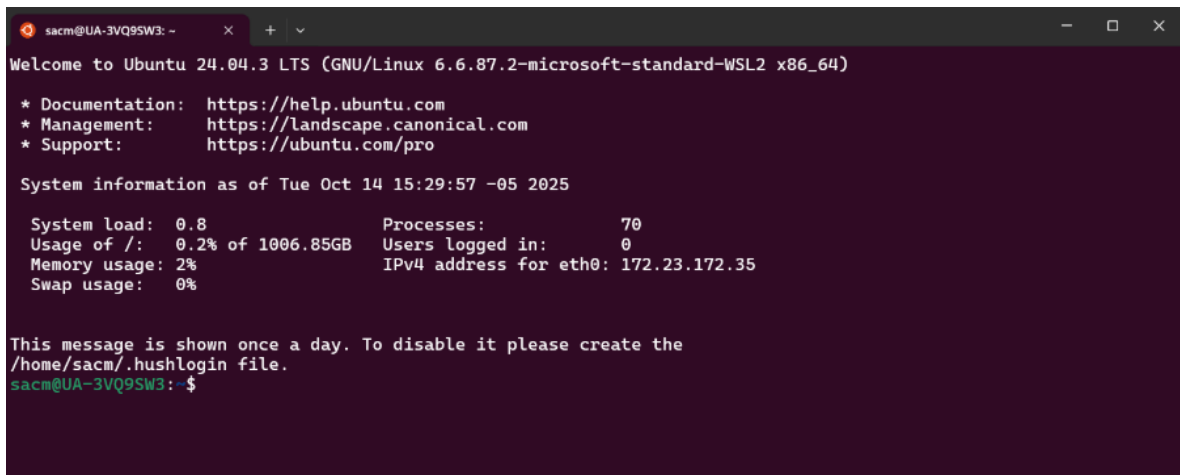
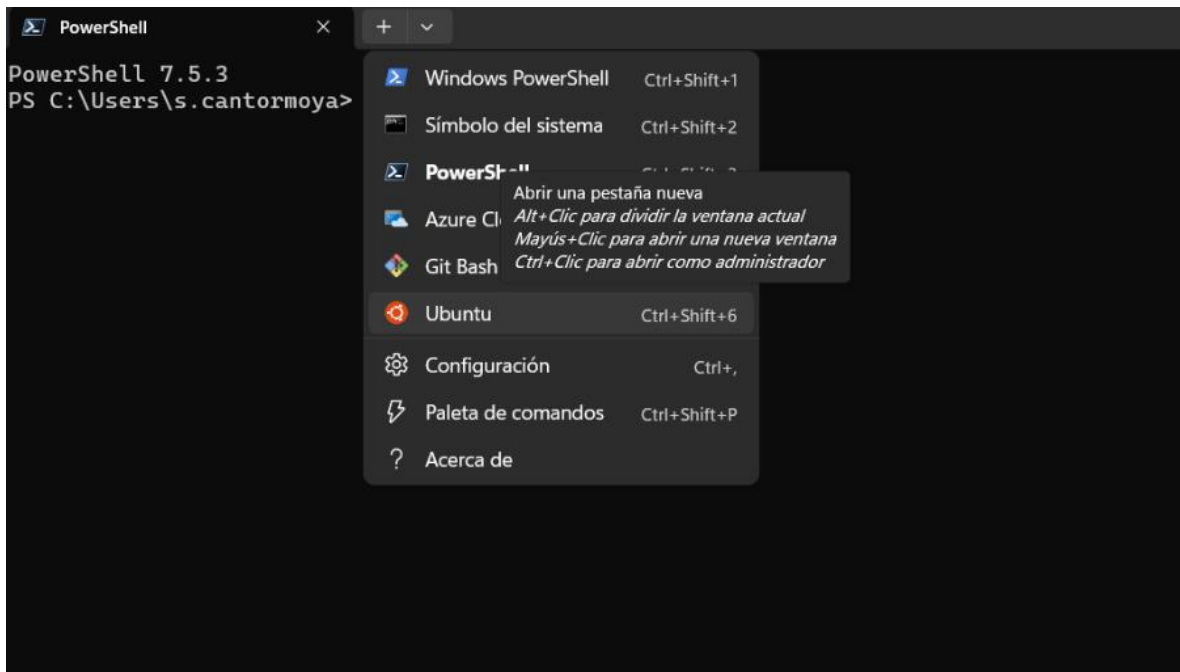
Instalación de Windows Terminal

3. Desde la Microsoft Store descarga Windows terminal para tener una terminal mas moderna que permite gestionar múltiples Shell



Configuración Inicial de Ubuntu

4. Configura Ubuntu
 - Abrir Windows Terminal.
 - Seleccionar Ubuntu de la lista de shells disponibles.
 - Esperar a que se complete la instalación inicial.
 - Cuando se solicite, crear un nombre de usuario y contraseña para el entorno Linux.



Actualización del Sistema Ubuntu

5. Ejecutar los siguientes comandos en la terminal de Ubuntu:
 - `sudo apt update`
 - `sudo apt upgrade -y`

```
sacm@UA-3VQ9SW3: ~$ sudo apt update
[sudo] password for sacm:
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1217 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1498 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [288 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [202 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [8748 B]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [882 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.3 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1490 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [196 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [18.1 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1978 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [301 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [378 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.2 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2084 kB]
Get:23 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [450 kB]
Get:24 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:25 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [520 B]
```

```
sacm@UA-3VQ9SW3: ~$ sudo apt upgrade -y
41 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libllvm17t64 libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
The following packages have been kept back:
  libgl1-amber-dri libglapi-mesa
The following packages will be upgraded:
  cloud-init dconf-gsettings-backend dconf-service dpkg dpkg-dev landscape-client landscape-common libc-bin
  libc-dev-bin libc-devtools libc6 libc6-dev libdconf1 libdpkg-perl libnss-systemd libpam-modules
  libpam-modules-bin libpam-runtime libpam-systemd libpam0g libsqlite3-0 libssl3t64 libsystemd-shared
  libsystemd0 libtiff6 libudev1 linux-libc-dev locales openssl python3-pip python3-xlrd snapd systemd
  systemd-dev systemd-hwe-hwdb systemd-resolved systemd-sysv systemd-timesyncd udev
39 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
Need to get 64.3 MB of archives.
After this operation, 3370 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 dpkg amd64 1.22.6ubuntu6.5 [1282 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libc-devtools amd64 2.39-0ubuntu8.6 [29.3 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libc6-dev amd64 2.39-0ubuntu8.6 [2125 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libc-dev-bin amd64 2.39-0ubuntu8.6 [20.4 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 linux-libc-dev amd64 6.8.0-85.85 [1921 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libc6 amd64 2.39-0ubuntu8.6 [3263 kB]
```

`sudo apt update`: Actualiza la lista de repositorios y paquetes disponibles

`sudo apt upgrade -y`: Instala todas las actualizaciones disponibles automáticamente (el flag -y confirma todas las preguntas)

Instalación de Ansible y Python

6. correr el comando `sudo apt install ansible python3-pip -y`

```
sacm@UA-3VQ9SW3:~$ sudo apt install ansible python3-pip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ansible is already the newest version (9.2.0+dfsg-0ubuntu5).
python3-pip is already the newest version (24.0+dfsg-1ubuntu1.3).
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2 libdrm-radeon1 libgl1-amd-gpu-dri libglapi-mesa libllvm17t64 libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
sacm@UA-3VQ9SW3:~$
```

Ansible: Motor de automatización que nos permitirá gestionar los PCs esclavos

python3-pip: Gestor de paquetes de Python para instalar librerías adicionales

Instalación de Librerías para WinRM

7. correr el comando `pip install pywinrm[credssp] requests-ntlm --break-system-packages`

```
sacm@UA-3VQ9SW3:~$ pip install pywinrm[credssp] requests-ntlm --break-system-packages
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests-ntlm in /usr/lib/python3/dist-packages (1.1.0)
Requirement already satisfied: pywinrm[credssp] in /usr/lib/python3/dist-packages (0.4.3)
Requirement already satisfied: requests-credssp>=1.0.0 in ~/.local/lib/python3.12/site-packages (from pywinrm[credssp]) (2.0.0)
Requirement already satisfied: cryptography in /usr/lib/python3/dist-packages (from requests-credssp>=1.0.0->pywinrm[credssp]) (41.0.7)
Requirement already satisfied: pypng>=0.5.0 in ~/.local/lib/python3.12/site-packages (from requests-credssp>=1.0.0->pywinrm[credssp]) (0.12.0)
Requirement already satisfied: requests>=2.0.0 in /usr/lib/python3/dist-packages (from requests-credssp>=1.0.0->pywinrm[credssp]) (2.31.0)
sacm@UA-3VQ9SW3:~$
```

Librerías instaladas:

- **pywinrm**: Permite a Python comunicarse con Windows via WinRM.
- **requests-ntlm**: Proporciona soporte para autenticación NTLM.
- **credssp**: Habilita el protocolo CredSSP para autenticación.

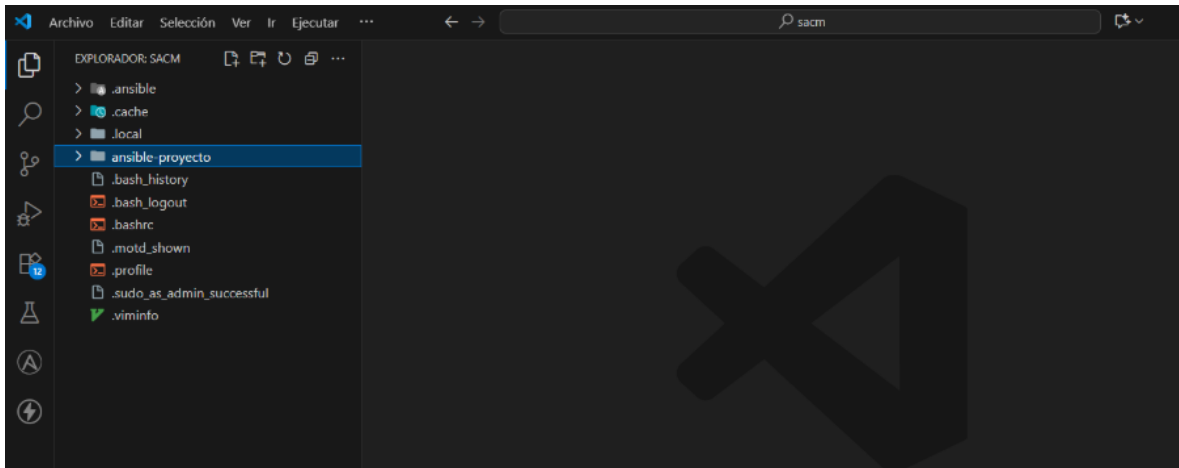
Configuración del Entorno de Desarrollo

6. usar el comando `code`.

Este comando abre Visual Studio Code con el directorio actual de Ubuntu, permitiendo una interfaz gráfica para gestionar archivos en lugar de usar solo línea de comandos.

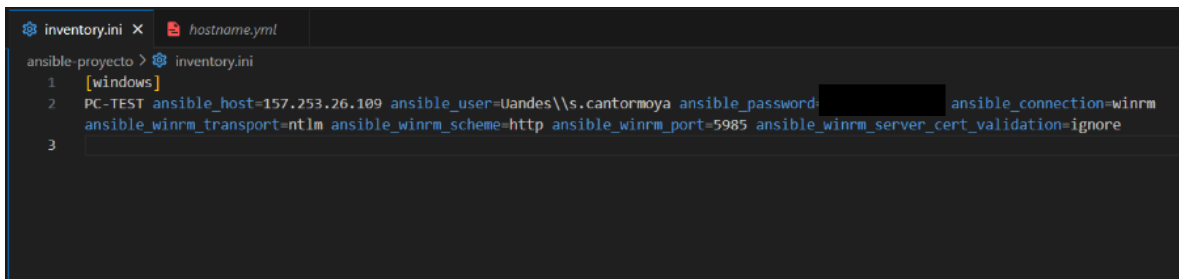
```
sacm@UA-3VQ9SW3:~$ code .
sacm@UA-3VQ9SW3:~$
```

- a) En VSCode, crear una nueva carpeta llamada **ansible-proyecto**.
- b) Esta carpeta contendrá todos los archivos necesarios para utilizar Ansible.



Creación del Archivo de Inventario

7. En la carpeta que acabamos de crear vamos a crear el archivo inventory.ini el cual será el inventario de todas nuestras maquinas esclavo

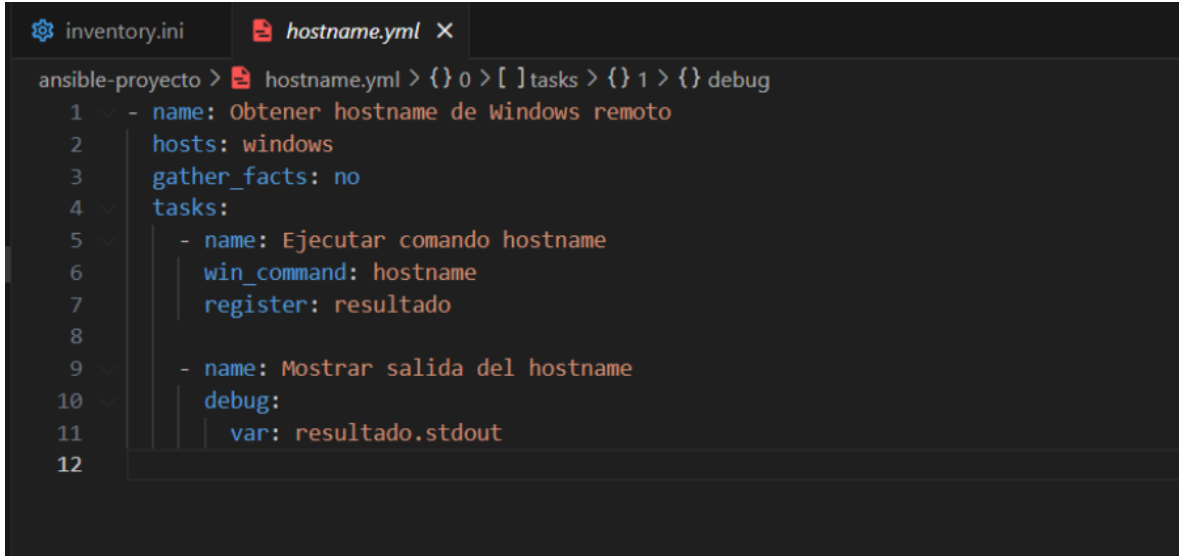


Variables explicadas:

- `ansible_user`: Usuario de dominio con formato DOMINIO/usuario.
- `ansible_password`: Contraseña del usuario.
- `ansible_connection=winrm`: Especifica que la conexión será via WinRM.
- `ansible_winrm_server_cert_validation=ignore`: Ignora validación de certificados SSL.

Creación del Playbook de Prueba

8. En la carpeta que acabamos de crear vamos a crear el archivo `hostname.yml` el cual contendrá todas las tareas o recetas que ejecutaremos en las maquinas.



```
ansible-proyecto > hostname.yml > {} 0 > [ ] tasks > {} 1 > {} debug
1 - name: Obtener hostname de Windows remoto
2   hosts: windows
3   gather_facts: no
4   tasks:
5     - name: Ejecutar comando hostname
6       win_command: hostname
7       register: resultado
8
9     - name: Mostrar salida del hostname
10      debug:
11        var: resultado.stdout
12
```

Explicación del playbook:

- `gather_facts: no`: Desactiva la recolección automática de información.
- `win_command: hostname`: Ejecuta el comando `hostname` en el equipo Windows.
- `register: hostname_result`: Guarda el resultado del comando en una variable.
- `debug`: Muestra el resultado en la salida.

Pruebas de Funcionamiento

9. Navegar al directorio del proyecto en la terminal de Ubuntu con el comando `cd ansible-proyecto`
10. Ejecuta el comando `ansible-playbook -i inventory.ini hostname.yml`

```
sacm@UA-3VQ9SW3: ~/ansible
sacm@UA-3VQ9SW3:~/ansible-proyecto$ ansible-playbook -i inventory.ini hostname.yml

PLAY [Obtener hostname de Windows remoto] *****

TASK [Ejecutar comando hostname] *****
changed: [PC-TEST]

TASK [Mostrar salida del hostname] *****
ok: [PC-TEST] => {
  "resultado.stdout": "ART-ET20118\r\n"
}

PLAY RECAP *****
PC-TEST : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignor
ed=0

sacm@UA-3VQ9SW3:~/ansible-proyecto$
```

Interpretación del resultado:

- ok=2: Las dos tareas se completaron exitosamente.
- changed=0: No se realizaron cambios en el sistema.
- unreachable=0: El equipo esclavo fue contactado exitosamente.
- failed=0: No hubo errores durante la ejecución.
- devolvió el hostname del equipo esclavo por medio de la IP.