

Full Installation Steps for Ansible with WinRM

Here are the full steps for installing Ansible with WinRM support (including the pywinrm dependency) on a Linux system (Ubuntu/Debian or CentOS/RHEL).

This will enable you to manage Windows servers using Ansible via WinRM.

1. Install Ansible on Linux

a. On Ubuntu/Debian

- Update system packages:

```
sudo apt update
```

- Install the required dependencies:

```
sudo apt install software-properties-common
```

- Add the Ansible PPA (Personal Package Archive):

```
sudo add-apt-repository ppa:ansible/ansible
```

- Update package list again:

```
sudo apt update
```

- Install Ansible:

```
sudo apt install ansible
```

- Verify Ansible installation:

```
ansible --version
```

b. On CentOS/RHEL/Fedora

- Install EPEL (Extra Packages for Enterprise Linux) repository:

For CentOS/RHEL 7:

```
sudo yum install epel-release
```

For CentOS/RHEL 8 and Fedora:

```
sudo dnf install epel-release
```

- Install Ansible:

For CentOS/RHEL 7:

```
sudo yum install ansible
```

For CentOS/RHEL 8 and Fedora:

```
sudo dnf install ansible
```

- Verify Ansible installation:

```
ansible --version
```

2. Install pywinrm Dependency

- Install pywinrm using pip:

```
sudo pip3 install pywinrm
```

- Install additional dependencies for NTLM authentication:

```
sudo pip3 install requests-ntlm
```

- Verify pywinrm installation:

```
pip3 show pywinrm
```

3. Configure Ansible to Manage Windows Hosts with WinRM

a. Modify Ansible Inventory File

- Create or modify your inventory file (e.g., hosts) to include your Windows server(s):

[windows]

winserver01 ansible_host=192.168.1.100 # Replace with the IP of your Windows server

[windows:vars]

ansible_user=Administrator # Replace with your Windows admin username

ansible_password=YourSecurePassword # Replace with your Windows password

ansible_connection=winrm

ansible_winrm_transport=ntlm

ansible_port=5985 # Use port 5985 for HTTP or 5986 for HTTPS

ansible_winrm_server_cert_validation=ignore # Set this to 'ignore' for self-signed certs

b. Test the Connection

- Run an Ansible ad-hoc command to test the connection:

```
ansible -i hosts winserver01 -m win_ping
```

4. Troubleshooting

- Ensure WinRM is enabled on the Windows machine:

Run the following PowerShell command on the Windows server to enable WinRM:

```
winrm quickconfig
```

- Check for network connectivity issues:

Ensure there are no firewalls blocking the communication between the Ansible control node and the Windows server on the WinRM ports (5985/5986).

- SSL certificate validation:

If you're using HTTPS (port 5986), ensure the Windows server has a valid SSL certificate or set

`ansible_winrm_server_cert_validation=ignore` if you're using a self-signed certificate.

- Verify that the correct Ansible modules are available:

```
ansible-doc -l | grep win
```

5. Optional: Install Ansible in a Virtual Environment (for isolation)

- Create and activate a virtual environment:

```
python3 -m venv ansible-env
```

```
source ansible-env/bin/activate
```

- Install Ansible and pywinrm in the virtual environment:

```
pip install ansible pywinrm requests-ntlm
```

- Run Ansible commands inside the virtual environment by activating it whenever needed.

6. Full Verification

- Create a test playbook (e.g., `test_playbook.yml`):

- name: Test Windows server connection

```
hosts: windows
```

```
tasks:
```

- name: Ping Windows Server

```
win_ping:
```

- Run the playbook:

```
ansible-playbook -i hosts test_playbook.yml
```

You should receive a "pong" response if everything is configured properly.