**Windows Ansible Connection by winrm**

**Step1: Create Two Ec2 Instances (Windows & Linux) on AWS.**

1. **Start Linux server with putty and execute below commands.**

1)Update Linux server

$ sudo apt-get update

2) upgrade os

$ sudo apt-get upgrade –y

3) Install Prerequisites for ansible installation

$ sudo apt install software-properties-common

4) Add the Ansible Personal Package Archive (PPA)

$ sudo add-apt-repository --yes --update ppa:ansible/ansible

5) Install Ansible

$ sudo apt install ansible

6) Verify the Installation

$ ansible --version

7) Install ‘python3-venv’

$ sudo apt install python3-venv

8)Create a Virtual Environment

$ python3 -m venv myenv

9) Activate the Virtual Environment

$ source myenv/bin/activate

10) Install pywinrm within the Virtual Environment

$ pip install pywinrm

11) Deactivate the Virtual Environment

$ deactivate

12) Create one directory

$ mkdir windows\_ansible

13)Create host file inside directory

$ vi hostfile

**Note:** Enter below host contain in this hostfile

[windows]

windows\_host ansible\_host=13.234.66.99

[windows:vars]

ansible\_user=Administrator

ansible\_password=0Gi9(v9S=L2$1A72P)y)wJ0\*jT68L-jM

ansible\_port=5985

ansible\_connection=winrm

ansible\_winrm\_transport=basic

ansible\_winrm\_scheme=http

ansible\_winrm\_server\_cert\_validation=ignore

Note: below Changes need in the host file.

Replace ‘windows\_host ansible\_host’ with your windows server host Ip.

Replace ‘ansible\_password’ with your windows RDP password.

1. **Start windows server with RDP and execute below commands.**

1)Enable PowerShell remoting

# Enable-PSRemoting –Force

$ winrm quickconfig

2) Set WinRM service startup type to automatic

$ Set-Service WinRM -StartupType 'Automatic'

3) Configure WinRM Service

$ Set-Item -Path WSMan:\localhost\Service\Auth\Certificate -Value $true

$ Set-Item -Path 'WSMan:\localhost\Service\AllowUnencrypted' -Value $true

$ Set-Item -Path 'WSMan:\localhost\Service\Auth\Basic' -Value $true

$ Set-Item -Path 'WSMan:\localhost\Service\Auth\CredSSP' -Value $true

4) Configure Trusted Hosts

$ Set-Item WSMan:\localhost\Client\TrustedHosts -Value "13.233.0.144" -Force

5) Set LocalAccountTokenFilterPolicy

$ New-ItemProperty -Name LocalAccountTokenFilterPolicy -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -PropertyType DWord -Value 1 –Force

6) Set Execution Policy to Unrestricted

$ Set-ExecutionPolicy Unrestricted –Force

7) Restart the WinRM service

$ Restart-Service WinRM

8) List the WinRM listeners

$ winrm enumerate winrm/config/Listener

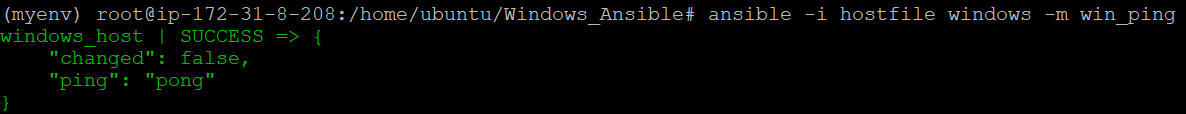
**Step2: Check Windows Ansible connection:**

**Note:** Open the Linux server where we have installed Ansible and check the connection by using the below command.

1. Check windows Ansible connection.

$ ansible -i hostfile windows -m win\_ping

**Our Expected result:**



**Step3: For test actual connection.**  
  
Create an Ansible playbook on a Linux server to **create** a directory and file on windows server, then execute that playbook by using the command below.

1. Ansible playbook

$ vi test\_playbook.yml  
  
---

- name: Create folder and file on Windows Server

hosts: windows\_host # Adjust to match the host or group name from your inventory file

gather\_facts: no # Disable gathering facts for Windows hosts

tasks:

- name: Create directory

win\_shell: New-Item -Path "C:\New folder" -ItemType Directory

become: yes # Run the command with elevated privileges (Administrator)

become\_user: Administrator # Specify the user with administrative rights

args:

creates: C:\New folder # Check if directory already exists

- name: Create file

win\_shell: Out-File -FilePath "C:\New folder\example.txt" -InputObject "Hello, World!"

become: yes

become\_user: Administrator

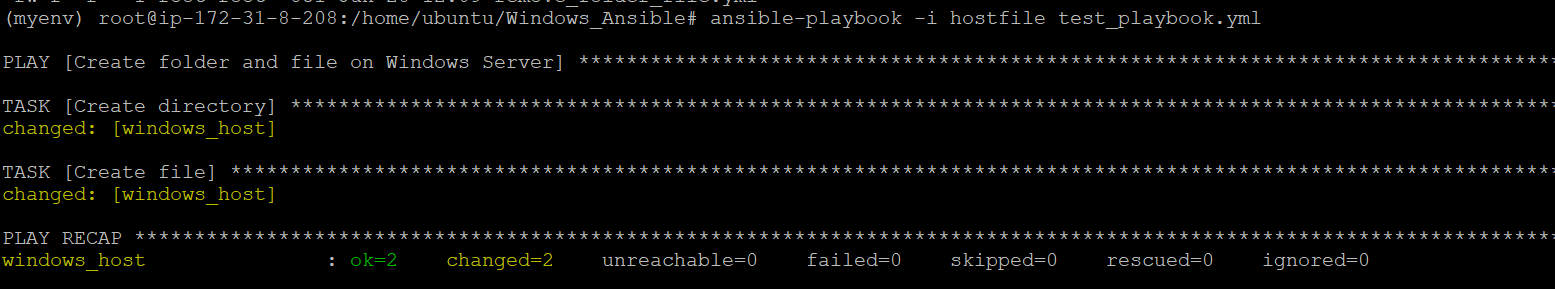
args:

creates: C:\New folder\example.txt # Check if file already exists

1. Execute ansible playbook

$ ansible-playbook –i hostfile test\_playbook.yml

Note: After executing the playbook successfully, you can check the results on your Windows server at the specified location.



**Step4: Remove created directory and file from windows.**

Create an Ansible playbook on a Linux server to **remove** a directory and file from windows server, then execute that playbook by using the command below.

1. Ansible playbook

$ vi remove\_folder\_file.yml

---

- name: Delete directory and file on Windows Server

hosts: windows\_host # Adjust to match the host or group name from your inventory file

gather\_facts: no # Disable gathering facts for Windows hosts

tasks:

- name: Delete file

win\_shell: Remove-Item -Path "C:\New folder\test\_folder\example.txt" -Force

become: yes # Run the command with elevated privileges (Administrator)

become\_user: Administrator # Specify the user with administrative rights

args:

removes: C:\New folder\test\_folder\example.txt # Check if file exists before deletion

- name: Delete directory

win\_shell: Remove-Item -Path "C:\New folder\test\_folder" -Recurse -Force

become: yes

become\_user: Administrator

args:

removes: C:\New folder\test\_folder # Check if directory exists before deletion

1. Execute ansible playbook

$ ansible-playbook –i hostfile remove\_folder\_file.yml

Note: After executing the playbook successfully, you can check the results on your Windows server at the specified location.

