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

$ 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.

2) 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 "35.154.198.224" -Force

---- Note: replace this ip with ansible server public Ip

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:

```
(myenv) root@ip-172-31-8-208:/home/ubuntu/Windows_Ansible# ansible -i hostfile windows -m win_ping
windows_host | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

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

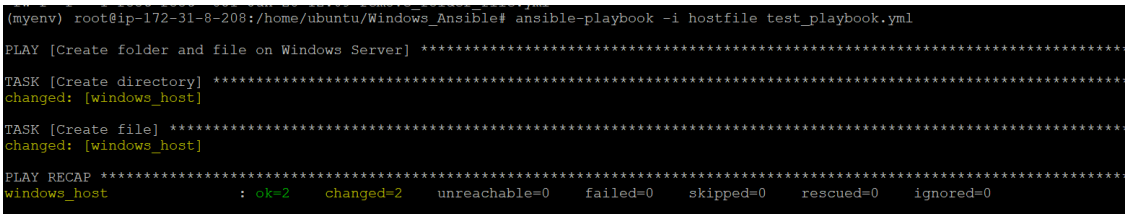
```
---
- name: Create folder and file on Windows desktop
 hosts: windows_host
 gather_facts: no

 tasks:
  - name: Create directory on desktop
    win_shell: New-Item -Path "\Desktop\NewFolder" -ItemType Directory
    become: yes
    become_user: Administrator

  - name: Create file inside the folder
    win_shell: Out-File -FilePath "\Desktop\NewFolder\example.txt" -InputObject
"Hello, World!"
    become: yes
    become_user: Administrator
```

2) 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 folder and file on Windows desktop
  hosts: 13.201.185.247
  gather_facts: no

  tasks:
    - name: Delete file inside the folder
      win_shell: Remove-Item -Path
"C:\Users\Administrator\Desktop\NewFolder\example.txt" -Force
      become: yes
      become_user: Administrator
      ignore_errors: yes  # Ignore errors if the file does not exist

    - name: Delete directory on desktop
      win_shell: Remove-Item -Path "C:\Users\Administrator\Desktop\NewFolder" -
Recurse -Force
      become: yes
      become_user: Administrator
      ignore_errors: yes  # Ignore errors if the folder does not exist
```

2) 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.