- Ansible Tips and Tricks: 10+ Ansible Examples to Save Time and Automate More Tasks
- Ansible Linux Users & Groups By Examples: 20+ Automation Examples on Linux Users and Groups Operation for Modern IT Infrastructure
- Ansible For PostgreSQL by Examples: 10+ Examples To Automate Your PostgreSQL database
- Ansible For Amazon Web Services AWS By Examples: 10+ Examples To Automate Your AWS Modern Infrastructure

demo

How to configure a Windows Host for Ansible connections.

- Create a user
- verify PowerShell & .NET
- setup WinRM
- create Inventory & Playbook

First of all, I'd suggest creating a user to run Ansible automation. This user needs to be Power User or have Administrative privileges in order to execute some Ansible code. Second, you need to verify that PowerShell and .NET versions, modern operating systems already meet the requirements. The most important part is to set up the WinRM. There is a great PowerShell script that sets up both HTTP and HTTPS listeners with a self-signed certificate and enables the Basic authentication option on the service. Once everything is done on the node you could configure the Ansible inventory on the Ansible Controller machine and run your first Ansible Playbook with the `` `win_ping` `` module to verify the successful configuration.

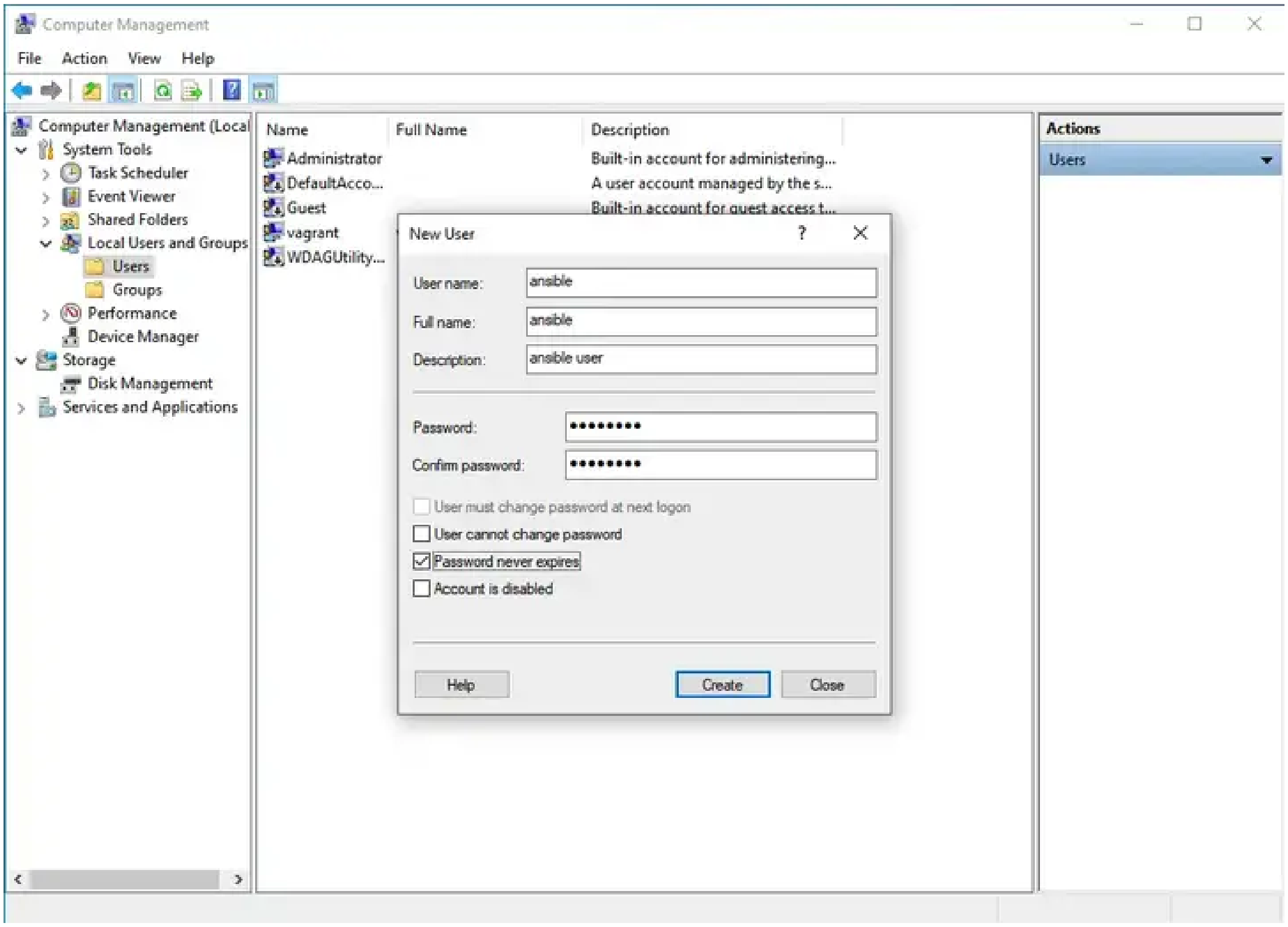
Windows node

Create an “ansible” user

1. open Computer Management (right-click from “This PC” > “Manage”)

2. New User (right-click from “Users” > “New User”)

User name: **ansible** Full name: **ansible** Description: **ansible user** Password and Confirm password: **SuperSecurePassword123@** Options: enable `` `Password never expires` `` disable `` `User must change password at next logon` ``



Name	Full Name	Description
Administrator		Built-in account for administering...
DefaultAcco...		A user account managed by the s...
Guest		Built-in account for guest access t...
vagrant		
WDAGUtility...		

User name: ansible

Full name: ansible

Description: ansible user

Password: •••••••••

Confirm password: •••••••••

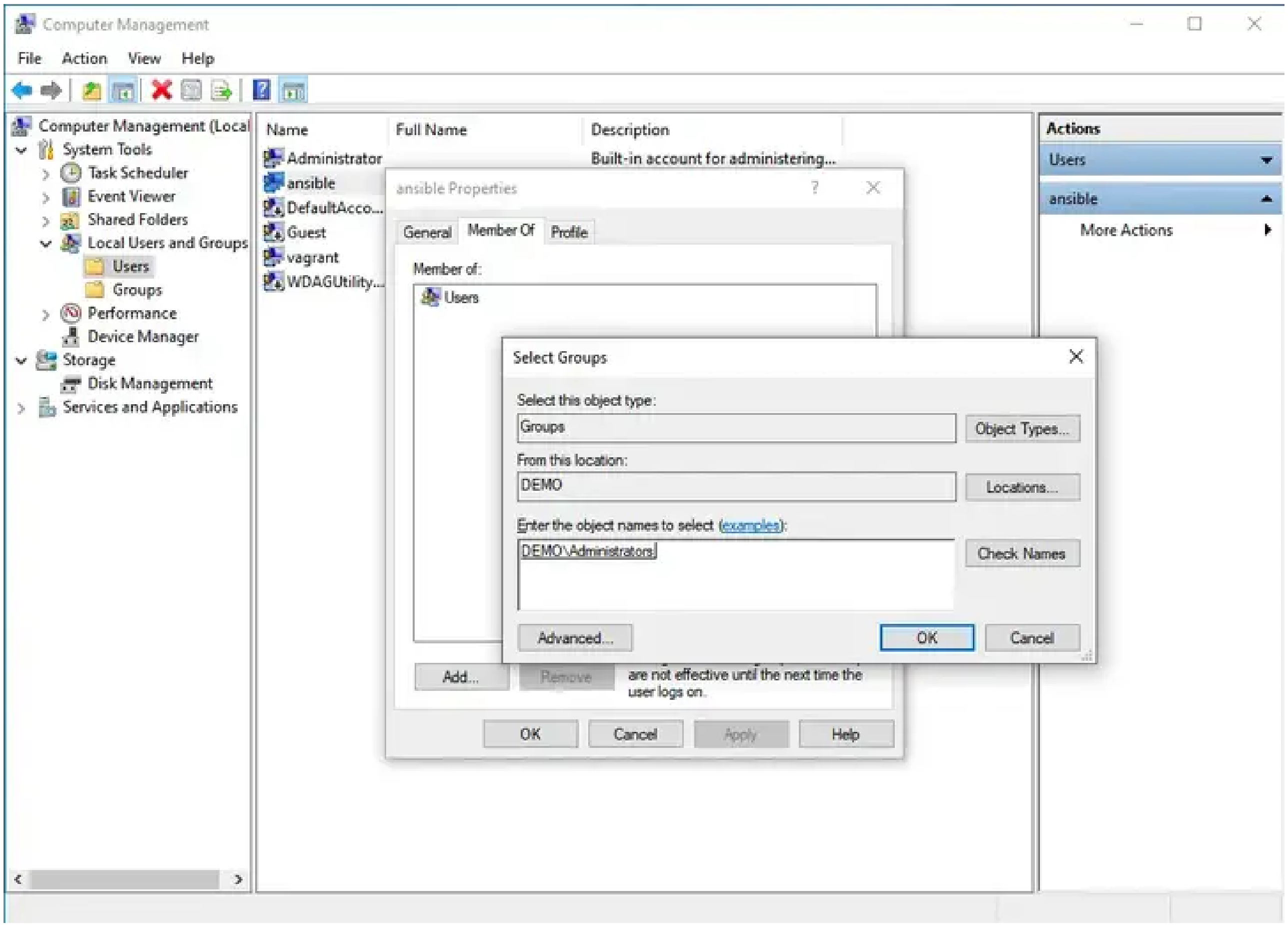
□ User must change password at next logon

□ User cannot change password

☑ Password never expires

□ Account is disabled

3. Add “ansible” user to “administrators” Group



Name	Full Name	Description
Administrator		Built-in account for administering...
ansible		
DefaultAcco...		
Guest		
vagrant		
WDAGUtility...		

Member of:
Users

are not effective until the next time the user logs on.

Select this object type:
Groups

From this location:
DEMO

Enter the object names to select (examples):
DEMO\Administrators

Verify PowerShell, .NET and set up WinRM

1. verify PowerShell version

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\vagrant> Get-Host | Select-Object Version
Version
-------
5.1.19041.1237
```

2. verify .NET version

```
PS C:\Users\vagrant> Get-ChildItem 'HKLM:\SOFTWARE\Microsoft\NET Framework Setup\
>>
PSChildName Version
----------- -------
```

```
Client      4.8.04084
Full        4.8.04084
Client      4.0.0.0
PS C:\Users\vagrant>
```

3. Verify WinRM not-configured

```
PS C:\Users\vagrant> winrm get winrm/config/Service
WSManFault
    Message = The client cannot connect to the destination specified in the reque
Error number:  -2144108526 0x80338012
The client cannot connect to the destination specified in the request. Verify tha
PS C:\Users\vagrant> winrm get winrm/config/Winrs
WSManFault
    Message = The client cannot connect to the destination specified in the reque
Error number:  -2144108526 0x80338012
The client cannot connect to the destination specified in the request. Verify tha
PS C:\Users\vagrant> winrm enumerate winrm/config/Listener
WSManFault
    Message = The client cannot connect to the destination specified in the reque
Error number:  -2144108526 0x80338012
The client cannot connect to the destination specified in the request. Verify tha
PS C:\Users\vagrant>
```

4. Setup WinRM

```
PS C:\Users\vagrant> [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityP
>> $url = "https://raw.githubusercontent.com/ansible/ansible/devel/examples/scrip
>> $file = "$env:temp\ConfigureRemotingForAnsible.ps1"
>>
>> (New-Object -TypeName System.Net.WebClient).DownloadFile($url, $file)
>>
>> powershell.exe -ExecutionPolicy ByPass -File $file
PS C:\Users\vagrant>
```

5. Verify WinRM configured

```
PS C:\Users\vagrant> winrm get winrm/config/Service
Service
    RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;IU)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;
    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:\Users\vagrant>
PS C:\Users\vagrant> winrm get winrm/config/Winrs
Winrs
    AllowRemoteShellAccess = true
    IdleTimeout = 7200000
    MaxConcurrentUsers = 2147483647
    MaxShellRunTime = 2147483647
    MaxProcessesPerShell = 2147483647
    MaxMemoryPerShellMB = 2147483647
    MaxShellsPerUser = 2147483647
PS C:\Users\vagrant> winrm enumerate winrm/config/Listener
Listener
    Address = *
    Transport = HTTPS
    Port = 5986
    Hostname = WIN10
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint = F4D065F8FC6EE18F1F0FF9533584955D0C9B8E59
    ListeningOn = 10.0.2.15, 127.0.0.1, 169.254.20.54, ::1, fe80::44a1:482d:5918:
PS C:\Users\vagrant> winrm enumerate winrm/config/Listener
Listener
    Address = *
    Transport = HTTPS
    Port = 5986
    Hostname = WIN10
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint = F4D065F8FC6EE18F1F0FF9533584955D0C9B8E59
    ListeningOn = 10.0.2.15, 127.0.0.1, 192.168.0.83, ::1, fe80::94b1:dc79:39cf:8
PS C:\Users\vagrant>
```

Ansible Controller

code

- inventory

```
[windows]
windows10 ansible_host=192.168.0.59
[windows:vars]
ansible_user=ansible
ansible_password=SuperSecurePassword123@
ansible_port=5986
ansible_connection=winrm
ansible_winrm_transport=basic
ansible_winrm_server_cert_validation=ignore
```

- win_ping.yml

```
---
```

```
- name: win_ping module demo
  hosts: windows
  become: false
  gather_facts: false
  tasks:
    - name: test connection
      ansible.windows.win_ping:
```

execution

```
ansible-pilot $ ansible-playbook -i windows/inventory windows/win_ping.yml
PLAY [win_ping module demo] *****************************************************
TASK [test connection] **********************************************************
ok: [windows10]
PLAY RECAP **********************************************************************
windows10                  : ok=1    changed=0    unreachable=0    failed=0    sk
ansible-pilot $
```

[code with ❤️ in GitHub]

Recap

Now you know Configure a Windows Host for Ansible. Subscribe to the YouTube channel, Medium, Website, Twitter, and Substack to not miss the next episode of the Ansible Pilot.

Academy

Learn the Ansible automation technology with some real-life examples in my

200+ Lessons Video Course

My book **Ansible By Examples**: 200+ Automation Examples For Linux and Windows System Administrator and DevOps