

[illegible]

Name	WinRM: Configure via Windows GUI
URL	https://attackdefense.com/challengedetails?cid=2023
Type	Services Exploitation: WinRM

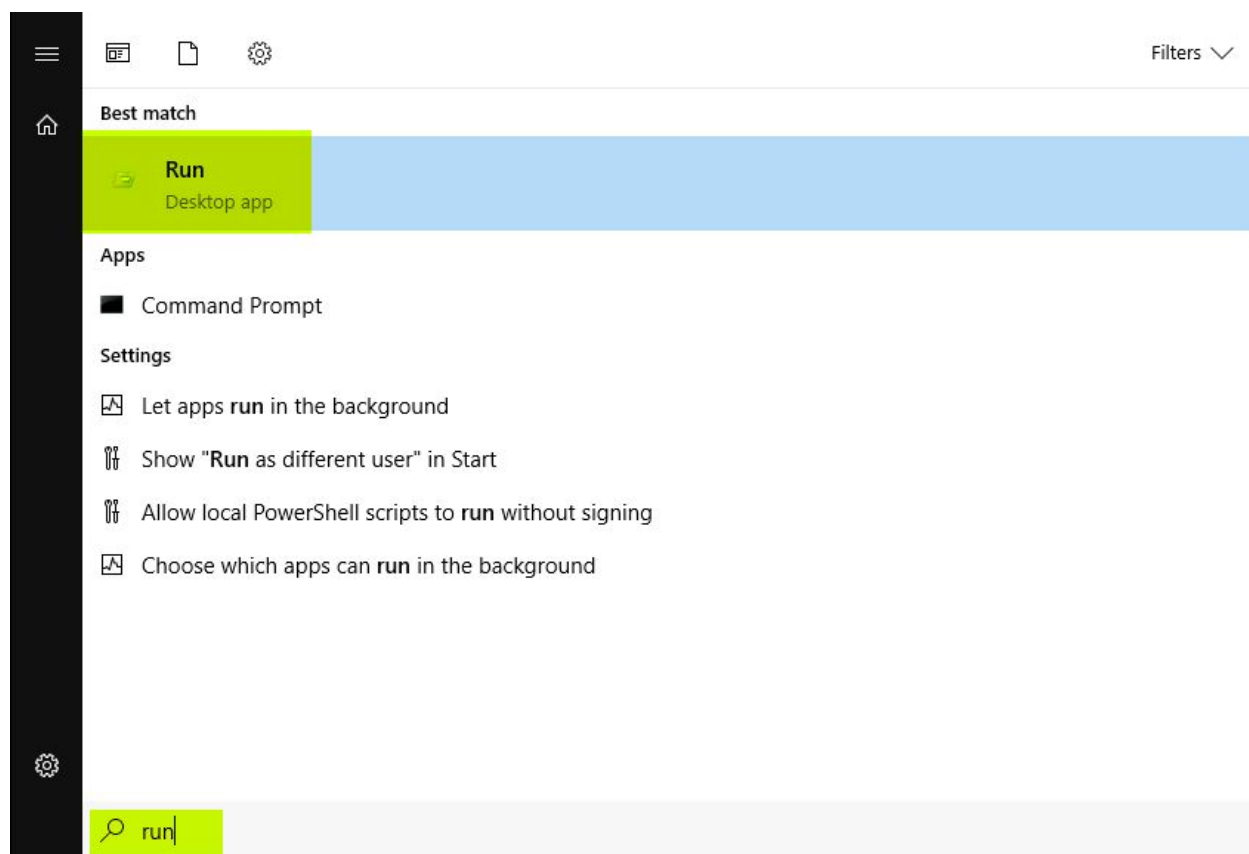
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Note: By default, if you are using Windows Server then, the WinRM service is already up and running. You need to configure the service in order to access it remotely. In this manual, we are demonstrating how to enable WinRM service and making necessary changes for learning purposes.

Configuration of WinRM

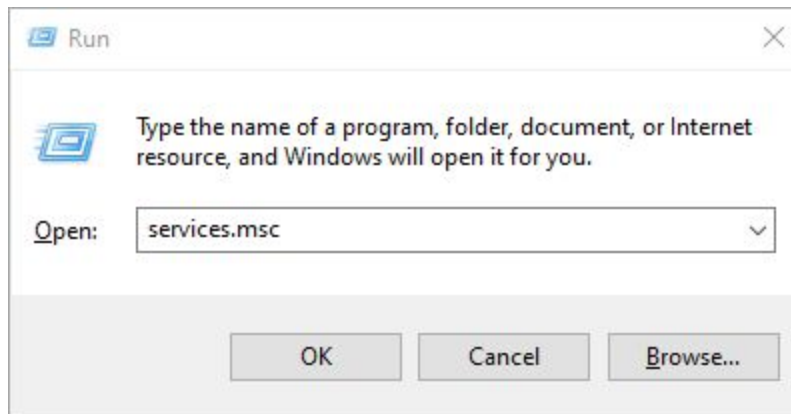
Note: Follow all the below steps on the “**Target Machine**” Also, The '**Windows + R**' would work on Linux systems. If you are using a Windows system, then it would conflict with your machine, and hence you won't be able to get the run prompt. You could manually open '**run**' by following the below steps:

1. Go to the Windows Start menu
2. Search for Run
3. Click on the Run App icon

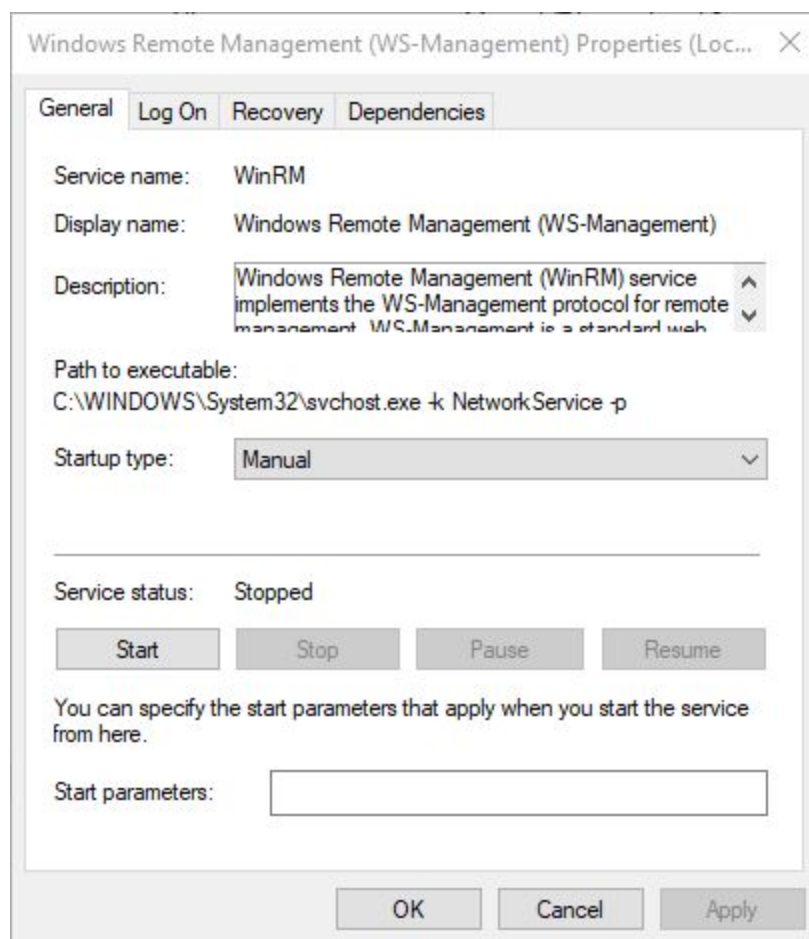


Step 1: Run powershell.exe to check for WinRM service status, if it's running or not. Open windows services

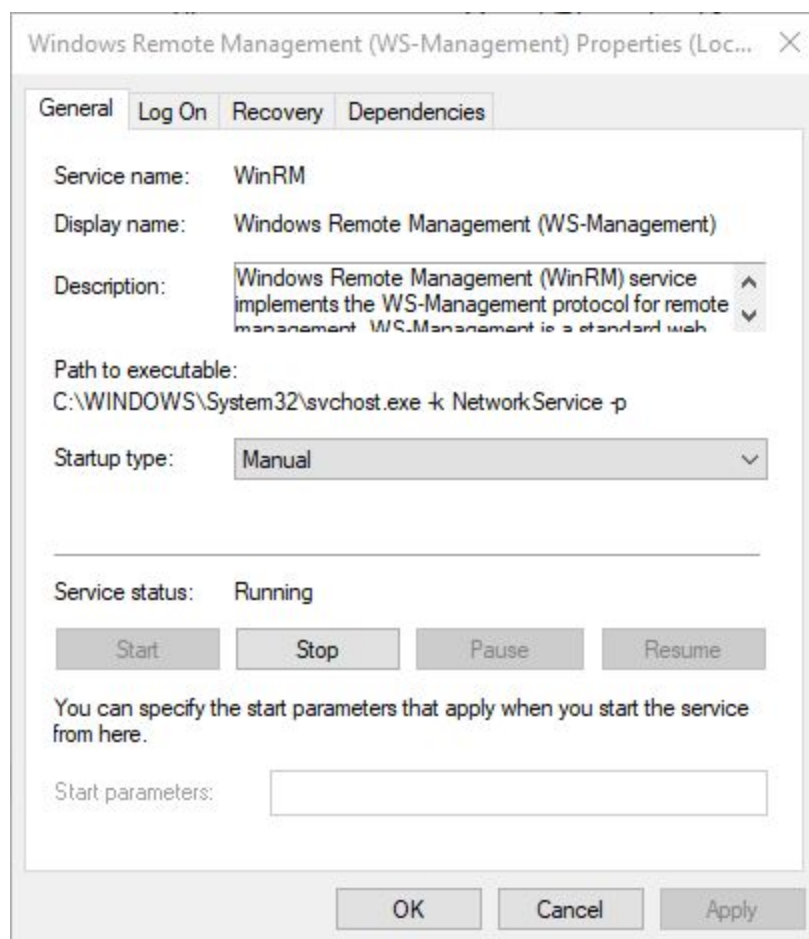
Command: Press Windows + R, type services.msc in Run dialog, and hit the Enter key to open it.



Step 2: Locate the “Windows Remote Management (WS-Management)” i.e WinRM service and check the status of the service.

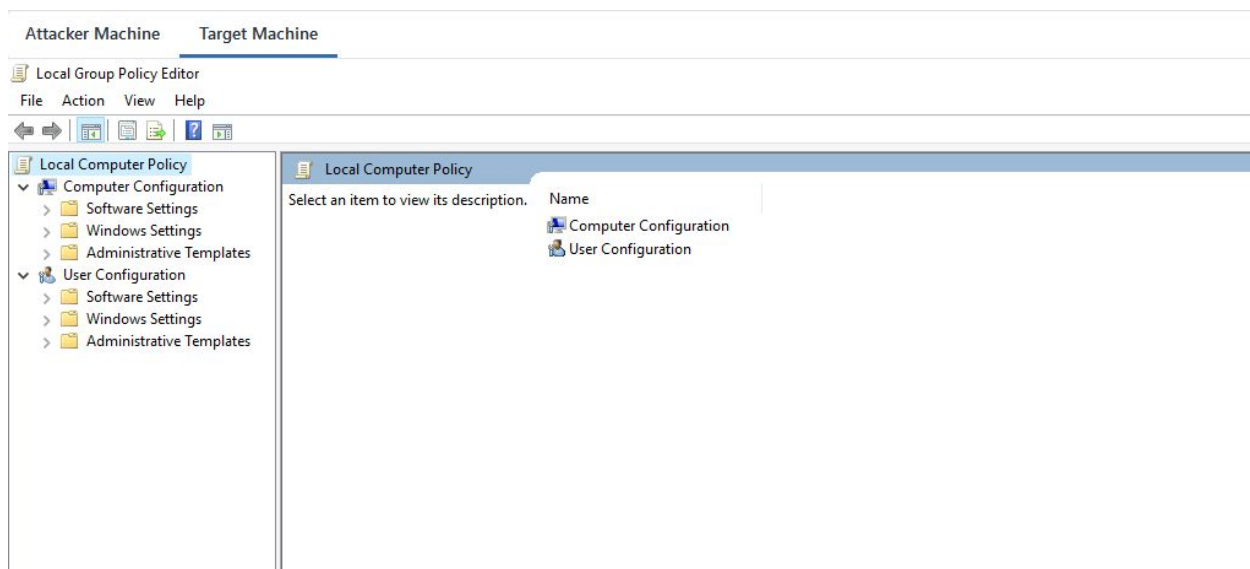
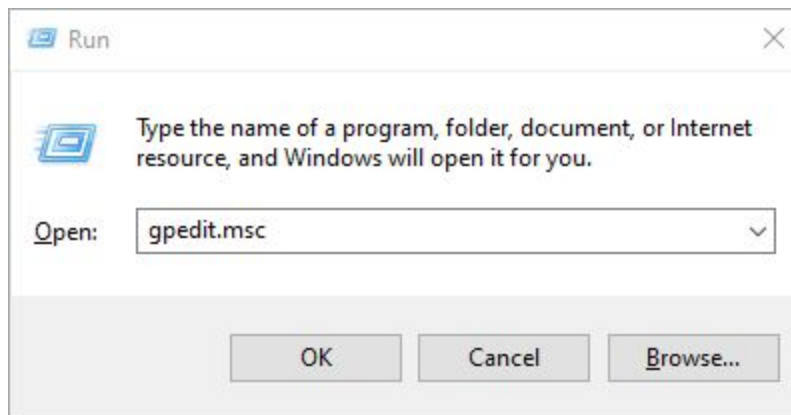


Step 3: The service is not running, Start the WinRM service and close the services.msc.

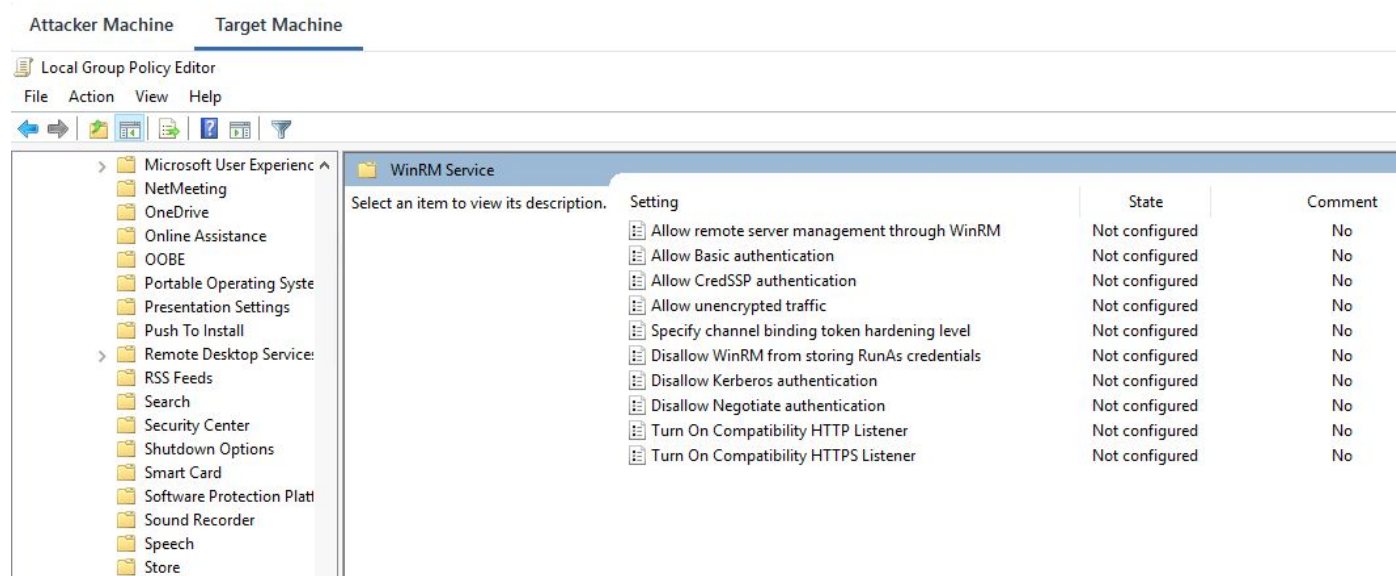


The winrm service is up and running. Now, we will configure the service.

Step 4: Press Windows + R, type **gpedit.msc** in the Run dialog, and hit the Enter key to open it.



Step 5: Expand the Menu tree as follows: **Computer Configuration > Administrative Templates > Windows Components > Windows Remote Management (WinRM) > WinRM Service.**



Step 6: Double click on “**Allow remote server management through WinRM**” and enable it.

Allow remote server management through WinRM

Previous Setting Next Setting

☐ Not Configured
 ☒ Enabled
 ☐ Disabled

Comment:

Supported on: At least Windows Vista

Options:

IPv4 filter: *

IPv6 filter: *

Syntax:

Type "*" to allow messages from any IP address, or leave the field empty to listen on no IP address. You can specify one or more ranges of IP addresses.

Example IPv4 filters:

2.0.0.1-2.0.0.20, 24.0.0.1-24.0.0.22

*

Help:

This policy setting allows you to manage whether the Windows Remote Management (WinRM) service automatically listens on the network for requests on the HTTP transport over the default HTTP port.

If you enable this policy setting, the WinRM service automatically listens on the network for requests on the HTTP transport over the default HTTP port.

To allow WinRM service to receive requests over the network, configure the Windows Firewall policy setting with exceptions for Port 5985 (default port for HTTP).

If you disable or do not configure this policy setting, the WinRM service will not respond to requests from a remote computer, regardless of whether or not any WinRM listeners are configured.

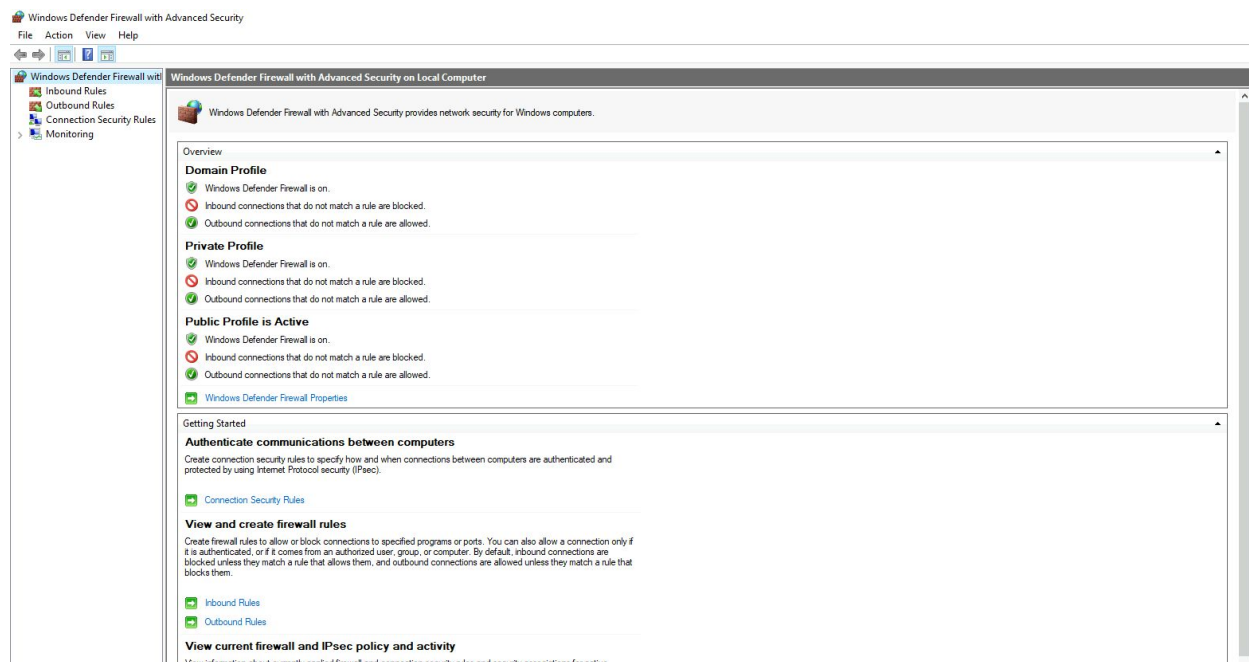
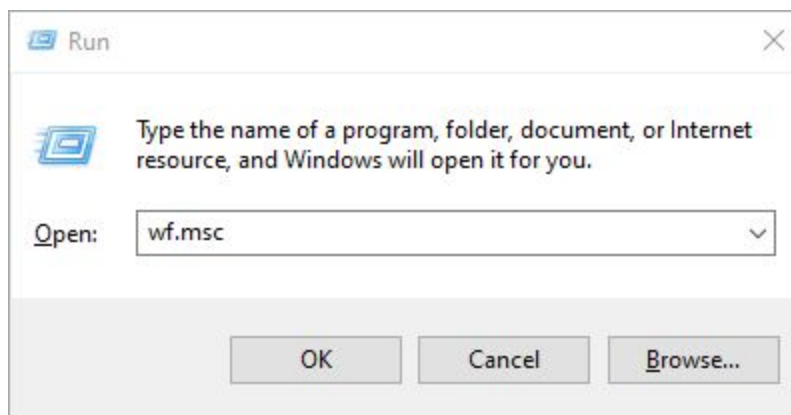
The service listens on the addresses specified by the IPv4 and IPv6 filters. The IPv4 filter specifies one or more ranges of IPv4 addresses, and the IPv6 filter specifies one or more ranges of IPv6 addresses. If specified, the service enumerates the available

OK Cancel Apply

Note: Remember to set options an asterisk symbol i.e “*”

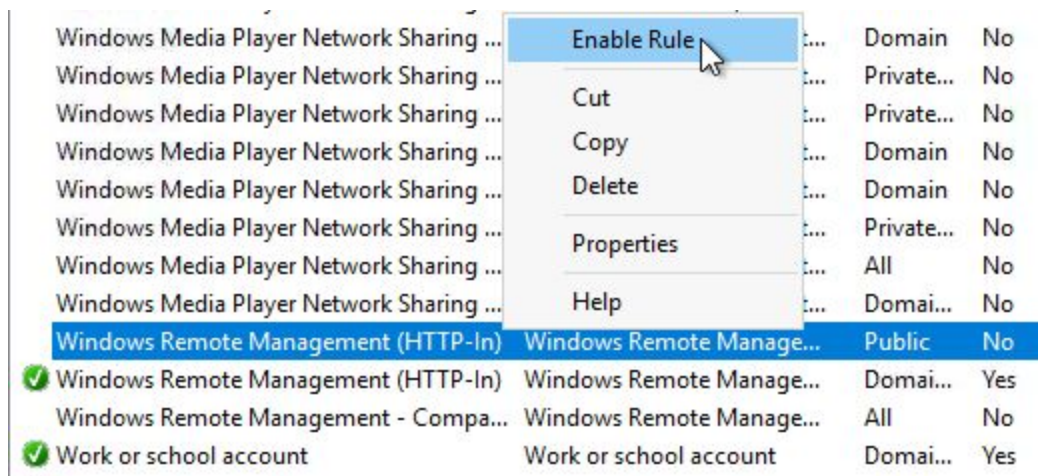
Click on “**Apply**” and “**ok**”

Step 7: Configure the Firewall rules to allow incoming connections for WinRM service. Press Windows + R, type **wf.msc** in the Run dialog, and hit the Enter key to open it.



Step 8: Expand the Menu tree as follows: Inbound Rules > Windows Remote Management (HTTP-In).

Right-click on **“Windows Remote Management (HTTP-In)”** and enable the rule.



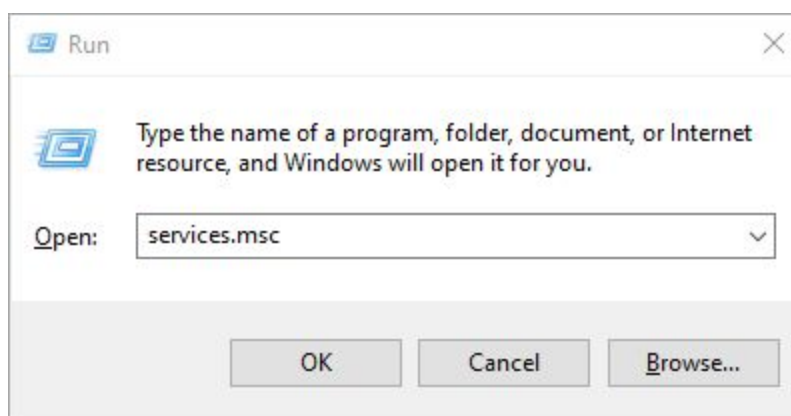
We have successfully configured the WinRM service on the remote server.

Note: Follow all the below steps on the “Attacker Machine”

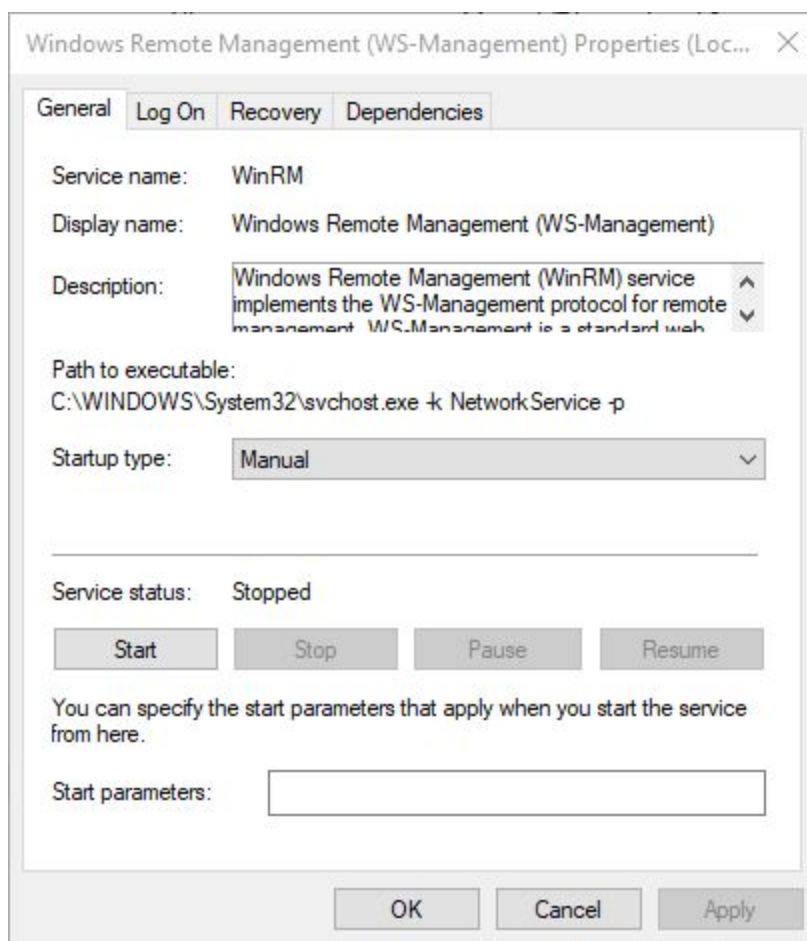
Step 9: We also need to allow clients (administration machines) to connect to the specific or all remote servers by modifying the “**TrustedHosts**”. Switch to the **client machine** and configure TrustedHosts to allow all remote servers.

Note: You cannot modify the TrustedHosts file if the WinRM service is not running.

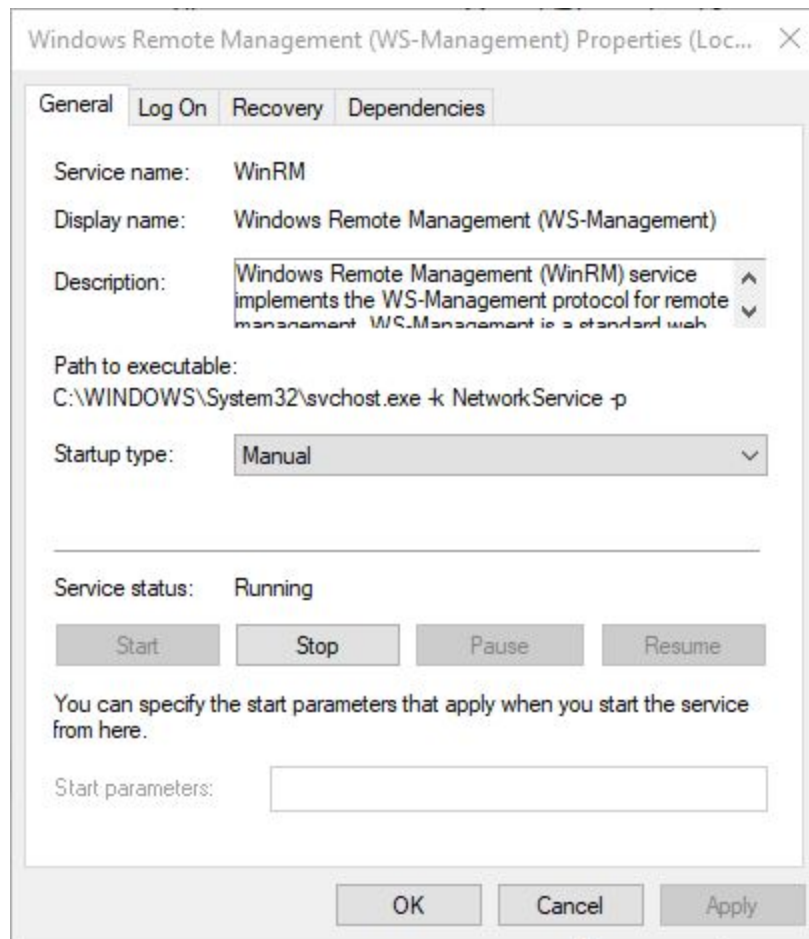
Command: Press Windows + R, type services.msc in Run dialog, and hit the Enter key to open it.



Step 10: Locate the “Windows Remote Management (WS-Management)” i.e WinRM service and check the status of the service.

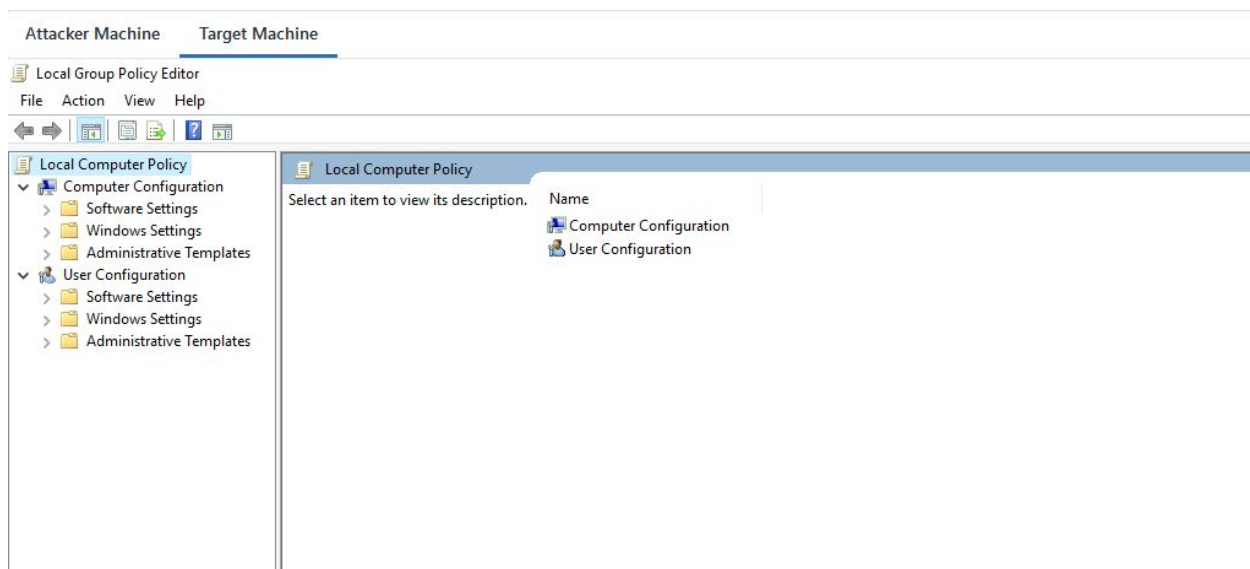
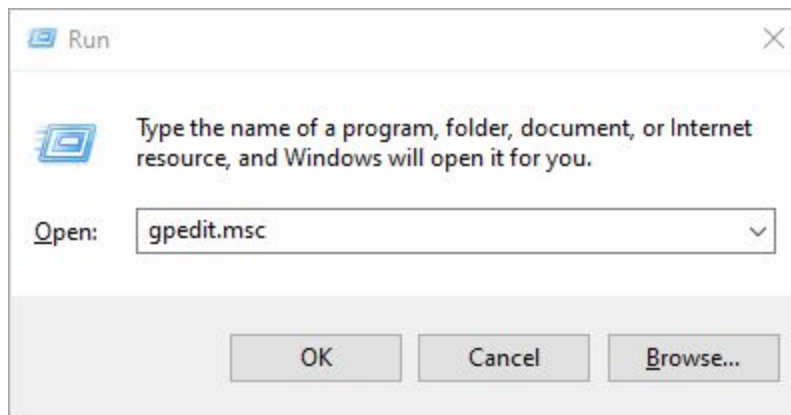


Step 11: The service is not running, Start the WinRM service and close the services.msc.

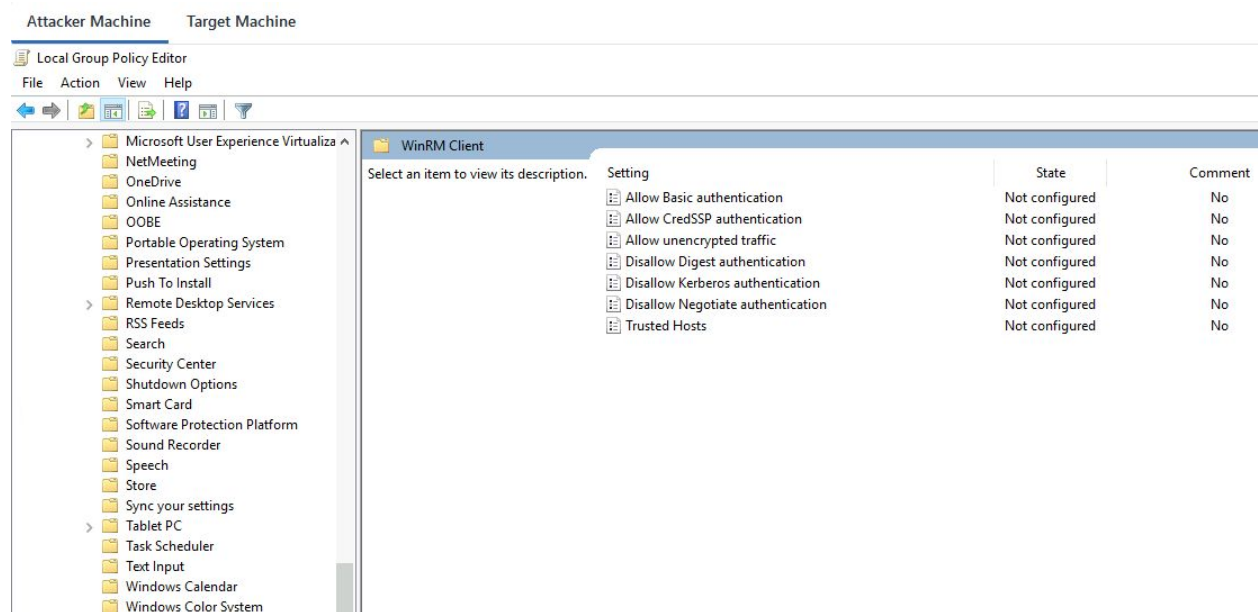


The winrm service is now running.

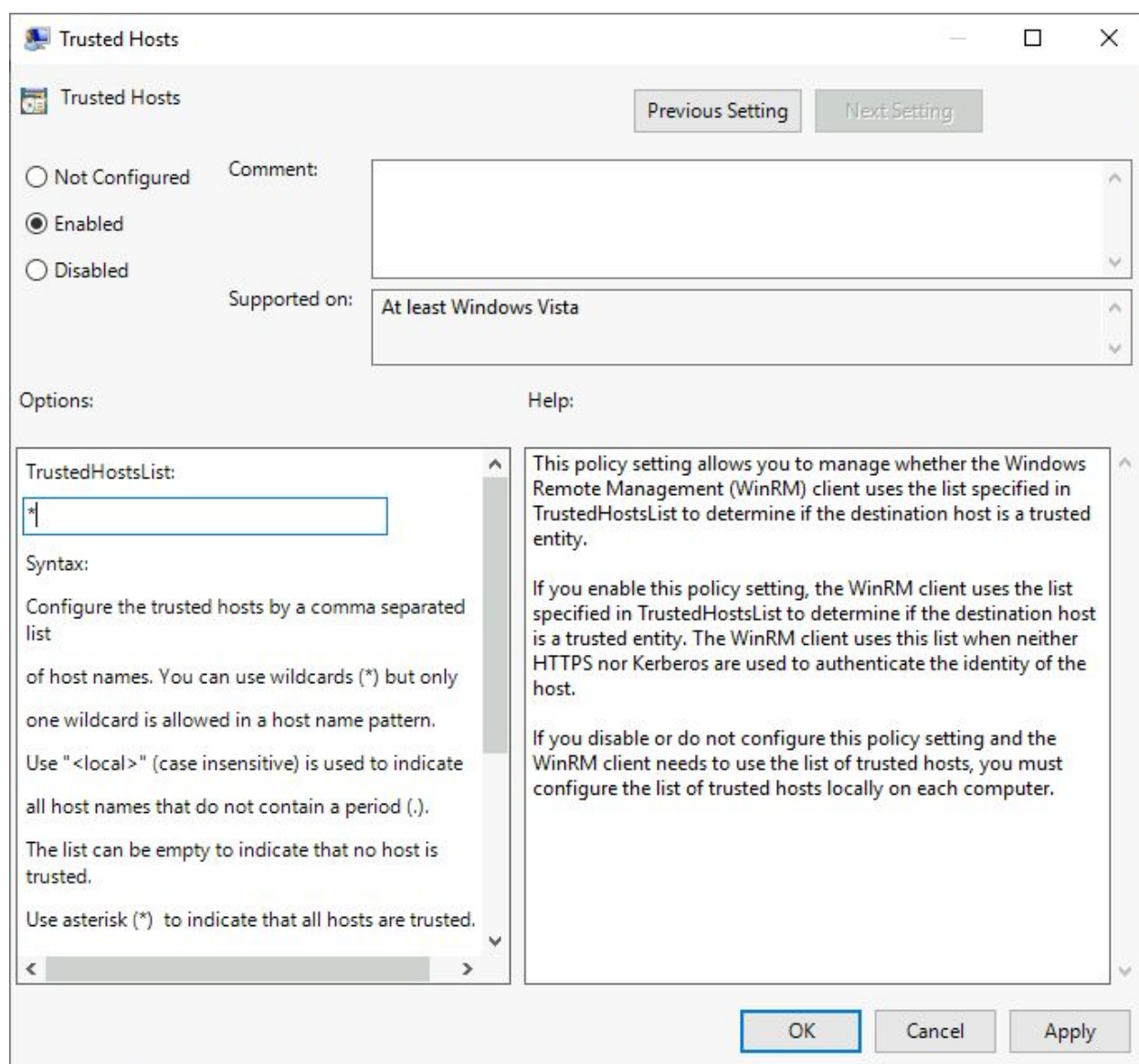
Step 12: Modifying the TrustedHosts file and allowing all remote hosts to connect. Press Windows + R, type **gpedit.msc** in the Run dialog, and hit the Enter key to open it.



Step 13: Expand the Menu tree as follows: **Computer Configuration > Administrative Templates > Windows Components > Windows Remote Management (WinRM) > WinRM Client.**



Step 14: Double click on “**Trusted Hosts**” and enable it.



Note: Remember to set options an asterisk symbol i.e “*”

We have modified **trustedhosts** and allowed any remote servers. Later, restart the WinRM service.

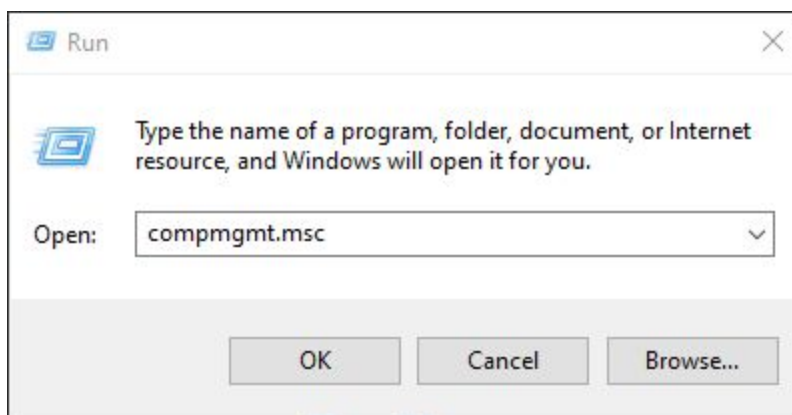
Note: It's always a good practice to mention the IP address or the computer name of the remote server. Also, you don't need to keep the WinRM service running on the client machine. We can turn it off after modifying the TrustedHosts.

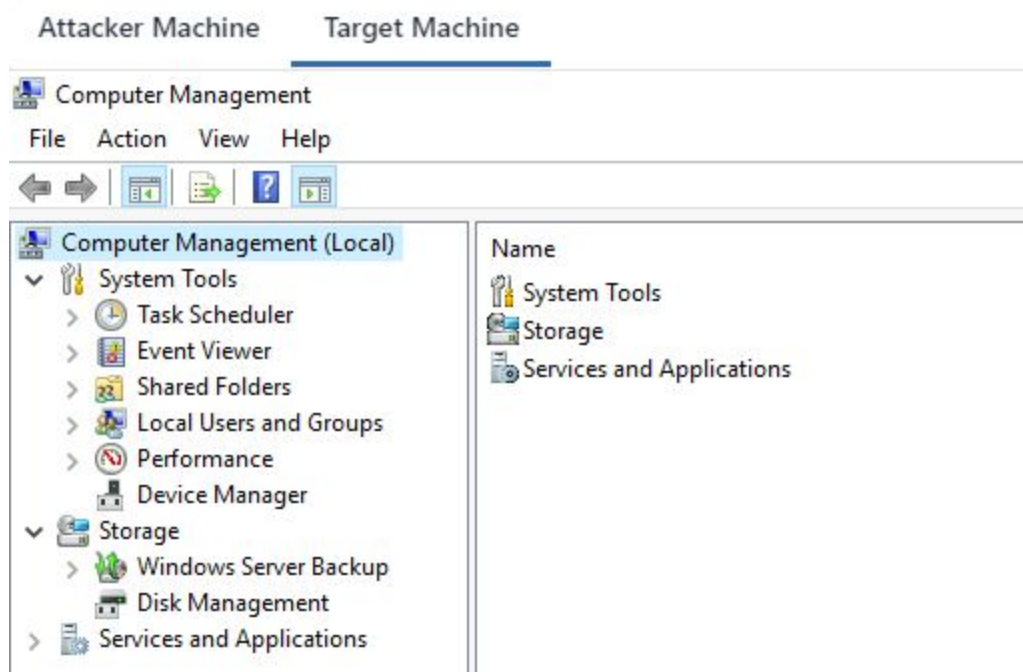
Creating a Demo User

Note: Follow all the below steps on the “Target Machine”

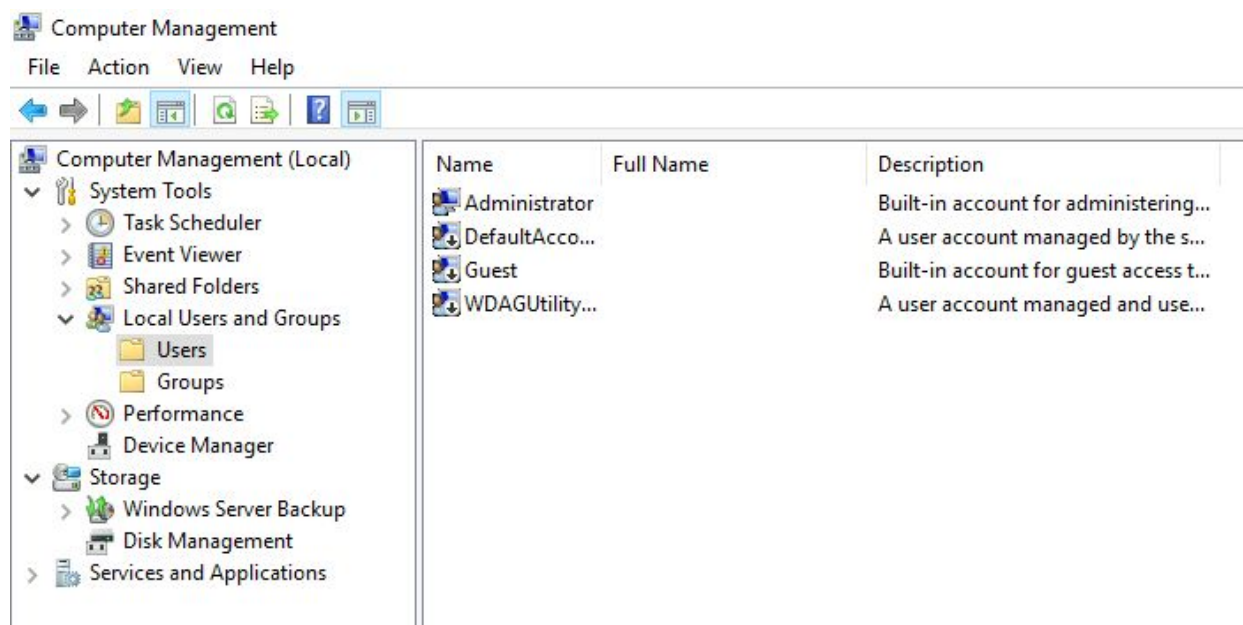
Step 1: We will create a demo user on the remote machine and the same user credentials we will use to execute commands from the client.

Modifying the TrustedHosts file and allowing all remote hosts to connect. Press Windows + R, type **compmgmt.msc** in the Run dialog, and hit the Enter key to open it.

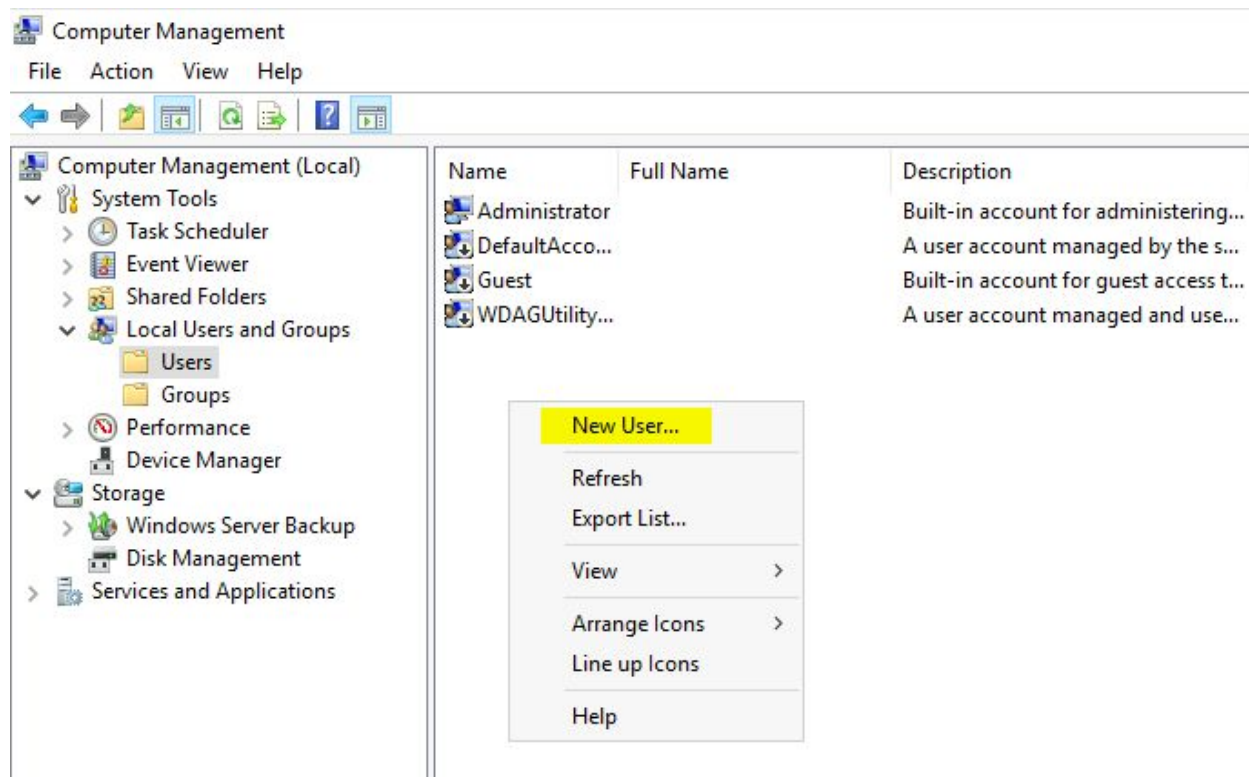


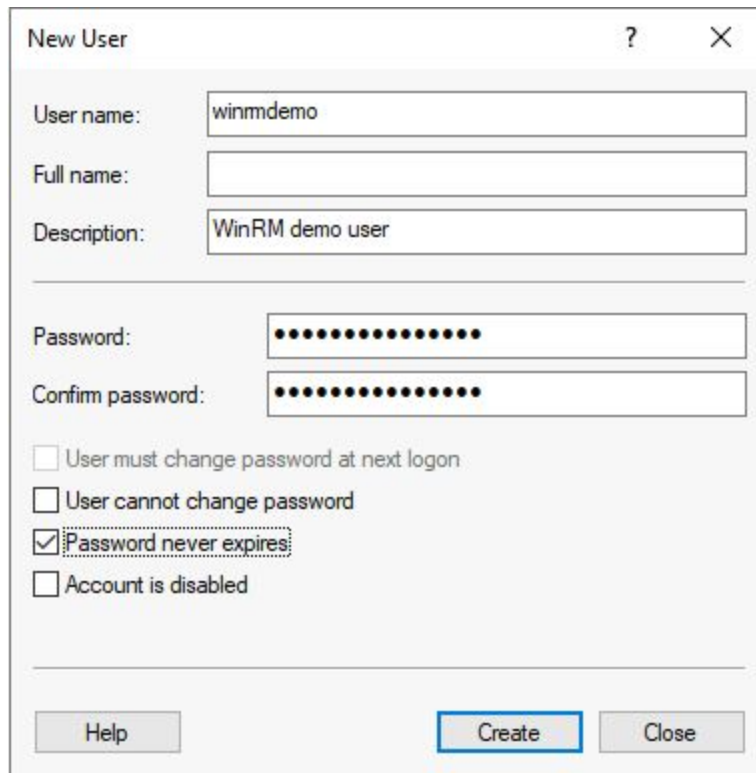


Step 2: Expand the Menu tree as follows: Local Users and Groups > Users



Step 3: We will create a demo user i.e **winrmdemo:password_123321** on the remote machine and the same user credentials we will use to execute commands.





The image shows a 'New User' dialog box with a red header bar. The dialog contains several input fields and checkboxes. The 'User name' field is filled with 'winrmdemo'. The 'Full name' field is empty. The 'Description' field is filled with 'WinRM demo user'. The 'Password' and 'Confirm password' fields are filled with 12 dots each. There are four checkboxes: 'User must change password at next logon' (unchecked), 'User cannot change password' (unchecked), 'Password never expires' (checked), and 'Account is disabled' (unchecked). At the bottom, there are three buttons: 'Help', 'Create' (highlighted with a blue border), and 'Close'.

New User ? X

User name: winrmdemo

Full name:

Description: WinRM demo user

Password:

Confirm password:

☐ User must change password at next logon

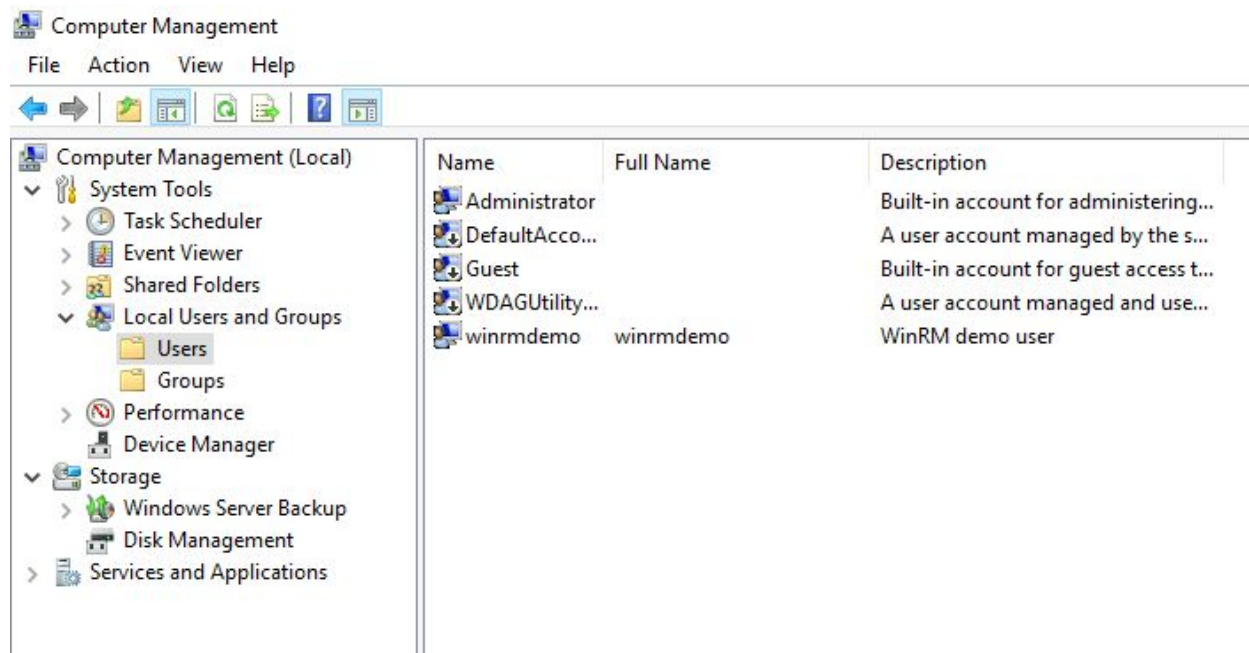
☐ User cannot change password

☒ Password never expires

☐ Account is disabled

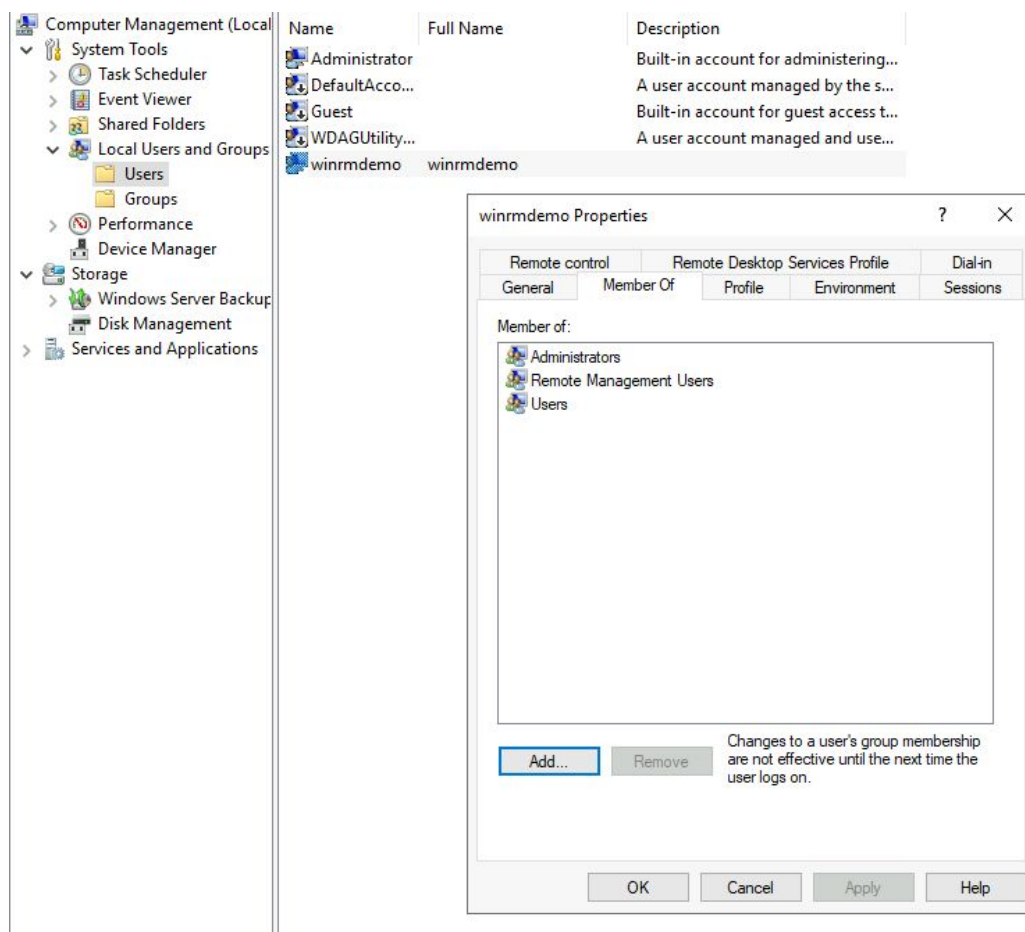
Help Create Close

After filling in the user details, click on “Create”



We have successfully, created a user i.e “winrmdemo”

Adding the user in the “**Administrators**” group and “**Remote Management Users**” group



Execute Commands On Remote Server

Note: Follow all the below steps on the “Attacker Machine”

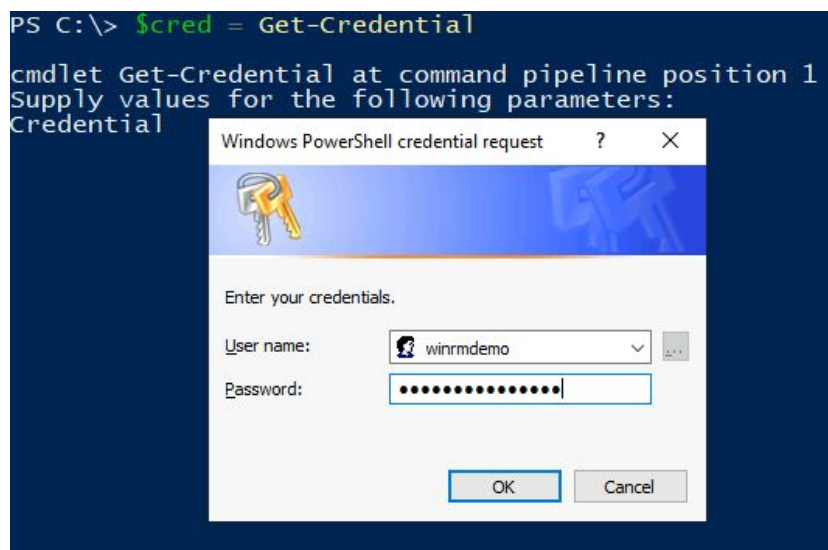
Step 1: We will execute the command using the “invoke-command” cmdlet to find all the running processes.

“The Invoke-Command cmdlet runs commands on a local or remote computer and returns all output from the commands, including errors.”

Command:

Store the target machine credentials in the \$cred variable.

\$cred = Get-Credential



```
PS C:\> $cred = Get-Credential

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
PS C:\> 
```

Step 2: Invoke the command `cmdlet` and check running processes.

Note: Please check the remote server IP address. By running the "**ipconfig**" command.

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : ap-southeast-1.compute.internal
    Link-local IPv6 Address . . . . . : fe80::fdf1:7a52:a039:fe85%4
    IPv4 Address. . . . . : 10.0.0.69
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.0.1
PS C:\Users\Administrator>
```

In my case, the remote server IP address is “10.0.0.69”

Command: Invoke-Command -ComputerName 10.0.0.69 -ScriptBlock {Get-Process} -Credential \$cred

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Invoke-Command -ComputerName 10.0.0.69 -ScriptBlock {Get-Process} -Credential $cred
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName	PSComputerName
130	9	15544	14252	0.11	4052	0	amazon-ssm-agent	10.0.0.69
258	14	6852	25560	0.77	4580	2	conhost	10.0.0.69
412	17	2184	5160	0.34	572	0	csrss	10.0.0.69
160	9	1656	4496	0.09	648	1	csrss	10.0.0.69
265	12	2132	6660	0.58	1248	2	csrss	10.0.0.69
372	15	3520	14996	0.13	4268	2	ctfmon	10.0.0.69
129	7	1440	5804	0.02	3972	0	dllhost	10.0.0.69
540	22	16628	38564	0.11	740	1	dwm	10.0.0.69
586	27	17900	60304	0.61	2104	2	dwm	10.0.0.69
1882	75	35840	110388	5.52	4676	2	explorer	10.0.0.69
49	6	1420	3608	0.00	948	0	fontdrvhost	10.0.0.69
49	6	1624	4128	0.03	952	1	fontdrvhost	10.0.0.69
49	7	2428	5928	0.06	1876	2	fontdrvhost	10.0.0.69
0	0	56	8	0.00	0	0	Idle	10.0.0.69
91	7	1180	4732	0.02	2684	0	LiteAgent	10.0.0.69
462	25	10112	42740	0.23	3368	1	LogonUI	10.0.0.69
1071	23	5316	14560	1.19	804	0	lsass	10.0.0.69
221	13	3016	10292	0.09	2952	0	msdtc	10.0.0.69
581	66	181032	183360	37.42	2816	0	MsMpEng	10.0.0.69
190	10	3336	9240	0.03	3468	0	NisSrv	10.0.0.69
887	62	167144	184220	12.94	1068	2	powershell	10.0.0.69
315	13	2484	11084	0.22	3348	2	rdpclip	10.0.0.69
0	7	504	70280	0.53	88	0	Registry	10.0.0.69
237	12	2504	12932	0.11	2144	2	RuntimeBroker	10.0.0.69
386	20	8820	26492	0.38	4128	2	RuntimeBroker	10.0.0.69
238	12	3456	14296	0.11	5036	2	RuntimeBroker	10.0.0.69
661	32	35268	78696	0.73	5000	2	SearchUI	10.0.0.69
515	11	4484	8948	0.86	784	0	services	10.0.0.69

We have successfully executed command on the remote server.

Checking members of the Administrators group.

Command: Invoke-Command -ComputerName 10.0.0.69 -ScriptBlock {net localgroup administrators} -Credential \$cred

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Invoke-Command -ComputerName 10.0.0.69 -ScriptBlock {net localgroup administrators} -Credential $cred
Alias name      administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members
-----
Administrator
winrmdemo
The command completed successfully.
PS C:\Users\Administrator> _
```

References:

- Installation and configuration for Windows Remote Management (<https://docs.microsoft.com/en-us/windows/win32/winrm/installation-and-configuration-for-windows-remote-management>)
- Enable-PSRemoting (<https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/enable-psremoting?view=powershell-7>)