WinRM Client Plugin

Plugin Information
View WinRM Client on the plugin site for more information.

This plugin's main goal is to provide WinRM Operations as Build Step.

About

Following Operations implemented:

1) Send-File Operation

2) Invoke-Command Operation

Job DSL:

```
job {
   steps {
       winRMClient {
           hostName(String hostName)
           credentialsId(String credentialsId)
           sendFile(String source, String destination, String configurationName)
           invokeCommand(String command)
       }
   }
}

For example:

freeStyleJob('WinRMClientJob') {
       steps {
         winRMClient {
               hostName('192.168.1.2')
               credentialsId('44620c50-1589-4617-a677-7563985e46e1')
               sendFile('C:\\test.txt','C:\\test', 'DataNoLimits')
               invokeCommand('dir')
         }
       }
}
```

Jenkins Pipeline:

```
 winRMClient credentialsId: '549bdb9b-0d36-4c21-948a-ca0a9ef6575f', hostName: '192.168.1.7',
 winRMOperations: [invokeCommand('dir'), sendFile(configurationName: 'DataNoLimits', destination: 'C:\\test',
source: 'C:\\test.txt')]
```

Configure WinRM to Use HTTP

Configure the WinRM host to enable communication with the PowerShell plug-in through the HTTP protocol.

Modify the WinRM configuration by running commands on the WinRM host machine. Can use the same machine as both the WinRM service and WinRM client.

Procedure:

- Run the following command to set the default WinRM configuration values.

  c:\> winrm quickconfig

- (Optional) Run the following command to check whether a listener is running, and verify the default ports.

  c:\> winrm e winrm/config/listener

  The default ports are 5985 for HTTP, and 5986 for HTTPS.

- Enable basic authentication on the WinRM service.

  Run the following command to check whether basic authentication is allowed.

  c:\> winrm get winrm/config

  Run the following command to enable basic authentication.

  c:\> winrm set winrm/config/service/auth @{Basic="true"}

- Run the following command to allow transfer of unencrypted data on the WinRM service.

  c:\> winrm set winrm/config/service @{AllowUnencrypted="true"}

- Enable basic authentication on the WinRM client.

  Run the following command to check whether basic authentication is allowed.

  c:\> winrm get winrm/config

  Run the following command to enable basic authentication.

  c:\> winrm set winrm/config/client/auth @{Basic="true"}

- Run the following command to allow transfer of unencrypted data on the WinRM client.

  c:\> winrm set winrm/config/client @{AllowUnencrypted="true"}

- If the WinRM host machine is in an external domain, run the following command to specify the trusted hosts.

  c:\> winrm set winrm/config/client @{TrustedHosts="host1, host2, host3"}

- Run the following command to test the connection to the WinRM service.

  c:\> winrm identify -r:http://winrm_server:5985 -auth:basic -u:user_name -p:password -encoding:utf-8

Configure WinRM to Use HTTPS

Configure the WinRM host to enable communication with the PowerShell plug-in through the HTTPS protocol.The WinRM host requires a certificate so that it can communicate through the HTTPS protocol. You can either obtain a certificate or generate one. For example, you can generate a self-signed certificate by using the Certificate Creation tool (makecert.exe) that is part of the .NET Framework SDK.

Procedure:

- Generate a self-signed certificate.

  The following command line contains example syntax for creating a certificate on the WinRM host by using makecert.exe.

  makecert.exe -r -pe -n "CN=host_name-3,O=organization_name" -e mm/dd/yyyy -eku 1.3.6.1.5.5.7.3.1 -ss my -sr localMachine -sky exchange -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12 certificate_name.cer

- Add the generated certificate by using the Microsoft Management Console.
  1. Run mmc.exe.
  2. Select File > Add/Remove Snap-in.
  3. From the list of available snap-ins, select Certificates and click Add.
  4. Select Computer account and click Next.
  5. Click Finish.
  6. Verify that the certificate is installed in Console Root > Certificates (Local Computer) > Personal > Certificates and Console Root > Certificates (Local Computer) > Trusted Root Certification Authorities > Certificates.

  If the certificate is not installed in the Trusted Root Certification Authorities and Personal folders, you must install it manually.

- Create an HTTPS listener by using the correct thumbprint and host name.

  The following command line contains example syntax for creating an HTTPS listener.

  winrm create winrm/config/Listener?Address=*+Transport=HTTPS @{Hostname="host_name";CertificateThumbprint="certificate_thumbprint"}

- Test the connection.

  The following command line contains example syntax for testing the connection.

  winrs -r:https://host_name:port_number -u:user_name -p:password hostname"}

Increase upload data size

- You need to create a new PSSessionConfiguration (this to not use the default one) in your remote computer:

Register-PSSessionConfiguration -Name DataNoLimits #or the name you like.

- Then configuring the parameter you want (in this case MaximumReceivedDataSizePerCommandMB and MaximumReceivedObjectSizeMB):

  Set-PSSessionConfiguration -Name DataNoLimits -MaximumReceivedDataSizePerCommandMB 500 -MaximumReceivedObjectSizeMB 500

- Then create the new session in your local Computer with the PSSessionConfiguration you need:

  $Session = New-PSSession -ComputerName MyRemoteComp -ConfigurationName DataNoLimits

Release 1.0 (04 March 2017)

- First public release