Ansible for Windows

**Ansible Information**

1. Ansible-base vs. Ansible-core
   1. Ansible distributes a minimalist package called ‘Ansible-core’, however, version 2.10 and lower it is called ‘Ansible-base’.
      1. Both copies contain the **Ansible language**, **runtime**, and a **very short list of core modules** and **other plugins**.
2. Ansible project v.3/4/5x naming confusion.
   1. Red Hat has messed up the naming scheme. If you read or hear Ansible project 3/4/5 and so on, this is a different package.
   2. Ansible project is an all in one package that includes more than just the core components. It includes a bundle of community plugins and modules with it.
      1. Since Ansible project 3 they removed Ansible-base naming and it is now considered as Ansible-core in Ansible project.
      2. In sum, Ansible project = Ansible-core + community modules/plugins

**Python Requirements**

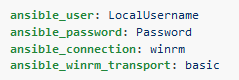
1. Ensure that you have Ansible and Python2 or Python3 installed.
   1. This can be done by Linux OS or WSL2 on a Windows machine.
   2. Ansible-core 2.11 will require soft dependency of python 3.8, Ansible-core 2.12 will have the requirement of python 3.8 or newer mandatory.
2. Python pip module
   1. By default, Python3 (3.4) and up pip module is included with the Python3 install.
   2. However, with Python2 you need to manually install pip.
3. Pywinrm module need to be installed.
   1. If pip is install, please use “pip list” to check if you have it already.
      1. Otherwise, you need to obtain it and install it.
   2. If you decide to go with Pywinrm **ONLY**, you can only do **basic**, **certificate**, and **NTLM auth**.

**Windows WinRM Authentication**

*WinRM is a management protocol used by Windows to remotely communicate with another server. Since Windows Server 2012, WinRM has been enabled by default, but in most cases extra configuration is required to use WinRM with Ansible. Ansible uses the python pywinrm package to communicate with Windows servers over WinRM. It is not installed by default with the Ansible package.*

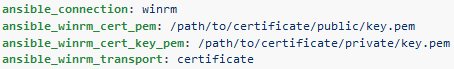
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Option** | **Local Accounts** | **AD Accounts** | **Cred Delegation** | **HTTP Encrypt** |
| Basic | Yes | No | No | No |
| Certificate | Yes | No | No | No |
| Kerberos | No | Yes | Yes | Yes |
| NTLM | Yes | Yes | No | Yes |
| CredSSP | Yes | Yes | Yes | Yes |

1. Basic Authentication
   1. Basic authentication is one of the simplest authentication options to use, but is also the most insecure. Basic authentication can only be used for local accounts (not domain accounts).
      1. The following example shows the inventory host vars configured for basic authentication:



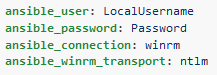
* + 1. Basic authentication is not enabled by default on a Windows host but can be enabled by running the following in PowerShell:
       1. “Set-Item -Path WSMan:\localhost\Service\Auth\Basic -Value $true”

1. Certificate
   1. Certificate authentication uses certificates as keys similar to SSH key pairs, but the file format and key generation process is different.
      1. The following example shows the inventory host vars configured for certificate authentication:

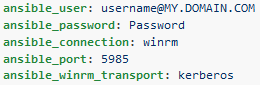


* + 1. Certificate authentication is not enabled by default on a Windows host but can be enabled by running the following in PowerShell:
       1. “Set-Item -Path WSMan:\localhost\Service\Auth\Certificate -Value $true”
  1. Active Directory Certificate Services is beyond of scope in this documentation but may be the best option to use when running in a domain environment.
     1. Generate a Certificate.
     2. Import a Certificate to the Certificate Store.
     3. Map the Certificate to an Account.

1. NTLM
   1. NTLM is an older authentication mechanism used by Microsoft that can support both local and domain accounts. NTLM is enabled by default on the WinRM service, so no setup is required before using it.
      1. NTLM is the easiest authentication protocol to use and is more secure than Basic authentication. If running in a domain environment, Kerberos should be used instead of NTLM.
      2. This example shows the inventory host variables configured to use NTLM authentication:



1. Kerberos
   1. Kerberos is the recommended authentication option to use when running in a domain environment. Kerberos supports features like credential delegation and message encryption over HTTP and is one of the more secure options that is available through WinRM. Kerberos requires some additional setup work on the Ansible host before it can be used properly.
      1. The following example shows host vars configured for Kerberos authentication:



* 1. Here are the other dependencies. Note, you need additional repos either online or offline to obtain and install these dependencies. **Please install pywinrm[kerberos] python package**.
     1. For RHEL/CentOS/etc:
        1. “sudo yum install gcc python-devel krb5-devel krb5-workstation python-devel”
        2. “pip install pywinrm[kerberos]”
  2. Ansible computer
     1. Edit “/etc/krb5.conf”
        1. Under “[libdefaults]”, add in “default\_realm = YourDomainNameALLInCAPS.com”
        2. Under “[realms]”, write as the following (this is case sensitive!!!!);

DomainNameALLInCAPS.COM = {

kdc = YourDomainController.DomainNameALLInLowerCase.com

admin\_server = YourDomainController.DomainNameALLInLowerCase.com

}

* + - 1. Under “[domain\_realm]”, write in the following;

.YourDomainNameLowerCase.com = DOMAINNAME.COM

YourDomainNameLowerCaseNoPeriod.com = DOMAINNAME.COM

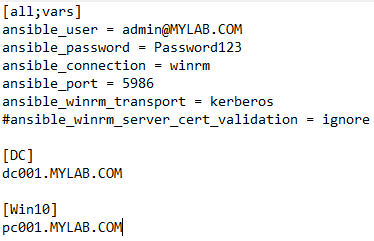
* + 1. Kerberos check!!
       1. Within your Ansible Unix/Linux environment type in the follow to see if Kerberos is set up correctly.
          1. “kinit Username@YourDomainName.com”

Enter the password and you should return nothing.

* + - * 1. “klist”

This should return a print out of a Kerberos ticket and its timestamp.

* 1. Ansible Inventory File
     1. Edit the Ansible inventory file to look something like this.



1. CredSSP
   1. CredSSP authentication is a newer authentication protocol that allows credential delegation. This is achieved by encrypting the username and password after authentication has succeeded and sending that to the server using the CredSSP protocol. CredSSP can be used for both local and domain accounts and also supports message encryption over HTTP.
      1. Because the username and password are sent to the server to be used for double hop authentication, ensure that the hosts that the Windows host communicates with are not compromised and are trusted.