**How to install and configure Ansible on Red Hat Enterprise Linux**

Red Hat Enterprise Linux 7.2 server which has been registered to the Red Hat Network for updates using

* **subscription-manager register --auto-attach**

The easiest way to install Ansible is by adding a third-party repository named EPEL (Extra Packages for Enterprise Linux), which is maintained over at http://fedoraproject.org/wiki/EPEL.

You can easily add the repo by running the following command

* **rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm**

No other software is required as Ansible utilizes SSH to interact with remote servers.

Now that we’ve added the EPEL repository, we’re ready to install Ansible! This can be done by running

* **yum -y install ansible**

on the command line. This will install a bunch of python dependencies during the process, but will only take around 30 seconds to complete.

Once the above has completed, you can confirm that Ansible is installed and ready to go by running

* **ansible --version.**

**Now let's see the steps to set up WinRM**

**Download the WinRM script on Windows 10 host**

WinRM can be installed using a script that can be uploaded with the name ConfigureRemotingForAnsible.ps1.

In our case, we have saved the file on the Desktop under the name ConfigureRemotingForAnsible.ps1

Step 2: Run the WinRM script on Windows 10 host

Next, run PowerShell as the Administrator

Run-PowerShell-as-Administrator

Navigate to the script location and run it. In this case, we have navigated to the Desktop location where we saved the script. Next, proceed and execute the WinRM script on the WIndows host:

Part 3: Connecting to Windows Host from Ansible Control Node

To test connectivity to the Windows 10 host, run the command:

# ansible winhost -m win\_ping

The output shows that we have indeed established a connection to the remote Windows 10 host from the Ansible Control node. This implies that we can now manage the remote Windows host using Ansible Playbooks. Let’s create a sample playbook for the Windows host system.

Part 4: Creating and running a playbook for Windows 10 host

In this final section, we shall create a playbook and create a task that will install Chocolatey on the remote host. Chocolatey is a package manager for Windows system. The play is defined as shown:

# vim chocolatey.yml

---

- hosts: winhost

gather\_facts: no

tasks:

- name: Install Chocolatey on Windows10

win\_chocolatey: name=procexp state=presen

Save and close the yml file. Next, execute the playbook as shown

# ansible-playbook chocolatey.yml