Module 4: Ansible on Cloud

Demo Document - 1

Demo: Configuration of Ansible on EC2 windows.

Problem Statement:

Configuration of Ansible on windows.

Solution Steps:

1. **Install ansible on control node(centos8)**

   - Install **python3** on centos8 as shown below:

     sudo dnf install python3

```
[centos@ip-172-31-22-243 ~]$ sudo dnf install python3
CentOS-8 - AppStream
CentOS-8 - Base
CentOS-8 - Extras
Dependencies resolved.
================================================================================
 Package                                Architecture            Version
================================================================================
Installing:
 python36                               x86_64                  3.6.8-2.module_el8.4
Upgrading:
 platform-python-pip                    noarch                  9.0.3-19.el8
 platform-python-setuptools             noarch                  39.2.0-6.el8
Installing dependencies:
 python3-pip                            noarch                  9.0.3-19.el8
 python3-setuptools                     noarch                  39.2.0-6.el8
Enabling module streams:
 python36                                                       3.6
```

   - Make python 3 the default by running below command.

     sudo alternatives –set python /usr/bin/python3

```
3. 18.118.84.234 (centos)
[centos@ip-172-31-22-243 ~]$ sudo alternatives --set python /usr/bin/python3
[centos@ip-172-31-22-243 ~]$
```

   - Verify the python version as shown below.

Python -version

```
[centos@ip-172-31-22-243 ~]$ python --version
Python 3.6.8
[centos@ip-172-31-22-243 ~]$
```

2. **Configure Virtual environment for running Ansible.**

sudo dnf install python3-virtualenv

```
[centos@ip-172-31-22-243 ~]$ sudo dnf install python3-virtualenv
Last metadata expiration check: 0:04:20 ago on Sun 20 Jun 2021 08:17:15 AM UTC.
Dependencies resolved.
================================================================================
 Package                    Architecture          Version
================================================================================
Installing:
 python3-virtualenv         noarch                15.1.0-19.module_el8.4.0+790+083e3d81
Upgrading:
 platform-python            x86_64                3.6.8-37.el8
 python3-libs               x86_64                3.6.8-37.el8
Installing dependencies:
 libpkgconf                 x86_64                1.4.2-1.el8
 pkgconf                    x86_64                1.4.2-1.el8
 pkgconf-m4                 noarch                1.4.2-1.el8
 pkgconf-pkg-config         x86_64                1.4.2-1.el8
 platform-python-devel      x86_64                3.6.8-37.el8
```

- Create virtual workspace by running **virtualenv env**

```
[centos@ip-172-31-22-243 ~]$ virtualenv env
Using base prefix '/usr'
New python executable in /home/centos/env/bin/python3.6
Not overwriting existing python script /home/centos/env/bin/python (you must use /home/centos
Installing setuptools, pip, wheel...done.
[centos@ip-172-31-22-243 ~]$
```

- Source it to activate. **source env/bin/activate**

```
[centos@ip-172-31-22-243 ~]$ source env/bin/activate
(env) [centos@ip-172-31-22-243 ~]$
```

3. **Install Ansible on Controller host.**

- Install ansible as shown below using **pip install ansible.**

```
(env) [centos@ip-172-31-22-243 ~]$ pip install ansible
Collecting ansible
  Downloading ansible-4.1.0.tar.gz (34.0 MB)
     |████████████████████████████████| 34.0 MB 74.1 MB/s
Collecting ansible-core<2.12,>=2.11.1
  Downloading ansible-core-2.11.1.tar.gz (6.1 MB)
     |████████████████████████████████| 6.1 MB 54.1 MB/s
Collecting jinja2
  Downloading Jinja2-3.0.1-py3-none-any.whl (133 kB)
     |████████████████████████████████| 133 kB 70.4 MB/s
Collecting PyYAML
  Downloading PyYAML-5.4.1-cp36-cp36m-manylinux1_x86_64.whl (640 kB)
     |████████████████████████████████| 640 kB 57.4 MB/s
Collecting cryptography
  Downloading cryptography-3.4.7-cp36-abi3-manylinux2014_x86_64.whl (3.2 MB)
     |████████████████████████████████| 3.2 MB 48.1 MB/s
Collecting packaging
```

- Check the ansible version after the installation using **ansible –version and** test using **ansible localhost -m ping**

```
(env) [centos@ip-172-31-22-243 ~]$ ansible --version
[DEPRECATION WARNING]: Ansible will require Python 3.8 or
05:13:41) [GCC 8.4.1 20200928 (Red Hat 8.4.1-1)]. This fe
setting deprecation_warnings=False in ansible.cfg.
ansible [core 2.11.1]
  config file = None
  configured module search path = ['/home/centos/.ansible
  ansible python module location = /home/centos/env/lib/p
  ansible collection location = /home/centos/.ansible/col
  executable location = /home/centos/env/bin/ansible
  python version = 3.6.8 (default, Mar 19 2021, 05:13:41)
  jinja version = 3.0.1
  libyaml = True
(env) [centos@ip-172-31-22-243 ~]$
```

```
(env) [centos@ip-172-31-22-243 ~]$ ansible localhost -m ping
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer
05:13:41) [GCC 8.4.1 20200928 (Red Hat 8.4.1-1)]. This feature
setting deprecation_warnings=False in ansible.cfg.
[WARNING]: No inventory was parsed, only implicit localhost is
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
(env) [centos@ip-172-31-22-243 ~]$
```

- Define windows hosts in ansible control node.

  Vim /etc/ansible/hosts

```
# vim /etc/ansible/hosts
```

- Hosts file look like below after the update.

  [winhost]

  192.168.0.14

  [winhost:vars]

  ansible_user: IEUser

  ansible_password: Passw0rd

  ansible_connection:winrm

  ansible_winrm_server_cert_validation=ignore

```
(env) [centos@ip-172-31-22-243 ansible]$ cat hosts
[winhost]
192.168.0.14

[winhost:vars]
ansible_user:IEUser
ansible_password:Passw0rd
ansible_connection:winrm
ansible_winrm_server_cert_validation=ignore
~
```

4. **Install Pywinrm on Ansible control node.**

   - Install **pywinrm** as shown below.

   - **pip install pywinrm**

```
(env) [centos@ip-172-31-22-243 ansible]$ pip install pywinrm
Collecting pywinrm
  Downloading pywinrm-0.4.2-py2.py3-none-any.whl (44 kB)
     |████████████████████████████████| 44 kB 2.1 MB/s
Collecting xmltodict
  Downloading xmltodict-0.12.0-py2.py3-none-any.whl (9.2 kB)
Collecting requests>=2.9.1
  Downloading requests-2.25.1-py2.py3-none-any.whl (61 kB)
     |████████████████████████████████| 61 kB 9.8 MB/s
Collecting six
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Collecting requests-ntlm>=0.3.0
  Downloading requests_ntlm-1.1.0-py2.py3-none-any.whl (5.7 kB)
Collecting certifi>=2017.4.17
```

5. **Configuring Windows Host**

   Windows host system needs to fulfill the below requirements.

   - Windows system should be windows 7 or later.

   - .NET framework 4.0 later.

- Windows PowerShell version 3.0

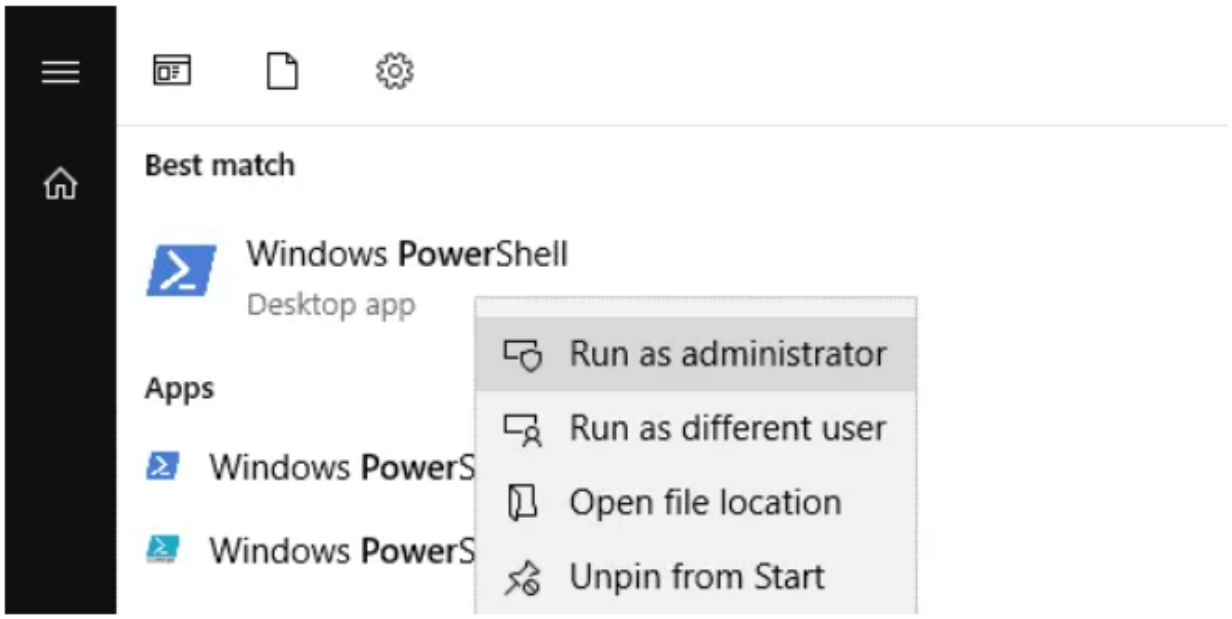
6. **Download the WinRM Script on Windows10 Host.**

- WinRM can be installed using the script.

- Save the below script as **ConfigureRemotingForAnsible.ps1** in the downloads or any location.

powershellscript.txt

- Run the above downloaded WinRM script on windows 10 host as shown below using administrator.



- Run the script **.\ConfigureRemotingForAnsible.ps1** on windows host as shown below.

```
PS C:\> .\ConfigureRemotingForAnsible.ps1
Self-signed SSL certificate generated; thumbprint: F66E8A714EA34B207CFB5D076F2D180E04816F90


wxf                   : http://schemas.xmlsoap.org/ws/2004/09/transfer
a                     : http://schemas.xmlsoap.org/ws/2004/08/addressing
w                     : http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
lang                  : en-US
Address               : http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
ReferenceParameters   : ReferenceParameters

Ok.


PS C:\>
```

7. **Connecting to windows host from ansible node.**

Test the connectivity to the windows host 10 by running below command.

ansible winhost -m win_ping

```
# ansible winhost -m win_ping
```

```
(env) [linuxtechi@centos-8 ~]$ ansible winhost -m win_ping
192.168.43.147 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```