Ansible with Windows

Ansible Training

Agenda

- How to connect
- How to use

How to Connect

Ansible Training

WinRM

- To connect to Windows systems, WinRM (Windows Remote Management) protocol is the must used one
- This protocol is well established on Windows world
- Uses HTTP/S transport layer to communicate
- Is a standard option making it easy to be configured and to be used

WinRM on Linux

- Since the control node should be a Linux system you need to use an implementation of WinRM on Linux
- Python have a package that implements this protocol and needs to be installed on control node to run Ansible commands

WinRM Authentication

- Several methods are allowed to make authentication over WinRM
  - Basic (username, password)
  - Certificates
  - NTLM
  - Kerberos
  - CredSSP
- Depending on the protcol you may use local accounts or Active Directory Domain accounts

WinRM Authentication

Option	Local Accounts	Active Directory Accounts	Credential Delegation	HTTP Encryption
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

Windows SSH

- There is an experimental feature to use OpenSSH for Windows connection
- This rely on previous OpenSSH installation and configuration in Windows Managed Nodes
- Since is an experimental feature is not recommended to use in production environments

How to Use

Ansible Training

How to use

- Every concept works the same way when running on Linux
- You just need to adapt small changes to make it compatible

Minor changes

- Windows related modules: Ansible.Windows — Ansible Documentation
- Shift from `/` as the path separator to `\`
- Adjust inventory **`host_vars`**/**`group_vars`** to add WinRM related properties

Samples

Ansible Training

Copy Files

```yaml
- name: Copy Files
  hosts: win
  tasks:
  - name: Copy File
    win_copy:
      src: /tmp/file.txt
      dest: C:\ansible_examples\
```

Install IIS

```yaml
---
- name: Installing IIS
  hosts: server
  gather_facts: false
  tasks:
    - name: Install IIS Web-Server with sub features and management
tools ansible.windows.win_feature:
        name: Web-Server
        state: present
        include_sub_features: true
        include_management_tools: true
      register: iis_install
      notify: Reboot

  handlers:
    - name: Reboot when Web-Server feature requires it
      ansible.windows.win_reboot:
      when: iis_install.reboot_required
      listen: Reboot
```

Install Apache MSI

```yaml
- name: Installing Apache MSI
  hosts: win

  tasks:
    - name: Download the Apache installer
      win_get_url:
        url: https://archive.apache.org/dist/httpd/binaries/win32/httpd-2.2.25-win32-x86-no_ssl.msi
        dest: C:\ansible_examples\httpd-2.2.25-win32-x86-no_ssl.msi

    - name: Install MSI
      win_package:
        path: C:\ansible_examples\httpd-2.2.25-win32-x86-no_ssl.msi
        state: present
```

Demo: Ansible in Windows

Ansible Advanced