Practical Part Day 2

Prerequisites:

For the practical part you would need 3 VMs.

1 Control Machine (CentOS), 2 Client Machines (Windows Server) with winrm preconfigured.

Task:

For Windows hosts define a user and a password which can be used for Ansible Server to connect to hosts.

These user and password considered as secrets should be defined as variables in Inventory file vars.

Use group_vars method to set secrets variables values, which should be used in Inventory file. Group_vars folder should be in a same folder where next playbook .yaml file will be stored.

```
  GNU nano 2.3.1           File: group_vars/WINDOWS_SERVER

---
ansible_user                        : Administrator
ansible_password                    : 11041976Ykv
ansible_port                        : 5986
ansible_connection                  : winrm
ansible_winrm_server_cert_validation : ignore
```

Encrypt secrets data in group_vars my means of Ansible-Vault and setting a password for secrets decryption.

```
[root@centos7 windows]# ansible-vault encrypt group_vars/WINDOWS_SERVER
New Vault password:
Confirm New Vault password:
Encryption successful
```

```
[root@centos7 windows]# cat group_vars/WINDOWS_SERVER
$ANSIBLE_VAULT;1.1;AES256
33623664326232363030623461643262616231306234653466616637626233396230643566626366
30313661343030353238636238313031373938366164663 00a336465303361333737323937613365
39363937303136326335663465373137616466393437313935656133336561643932333037383837
31616639323030343 70a6333663630663836623635373635643036396563323964663232343663 63
34313761313430383463666230333032613931653262313534396463323431343132343564636132
62373264613662313965396337363931653039616664323232643835636130313962336234343732
30373161633131633063613765343661313462663630316130356666666335623537636466353361
34393465646533393263323061366534643563613835326462383239636436313163386139386539
33636566373064656630333566333937343838356431653066376138383866353137386537393932
36396132306434353737373466366262376562393439623633623761623464643635373737346465
63613433653662366232643235376638633937646431396531323035366646413565363938633166
62363635333666633665316263303365663738636561303834373064366466353261633939616337
63323331346665366163613237663034653631383034303736613831336138393230613661643363
30616461616630353861333130383336333234666261633235396363633261653337336233396433
34663062353832646331396335663163333565666463336536326635626334383630383932393266
65356666343461326538383134393930663930396666633332363739343331623638393131653732
3736
```

Using an Ansible Role, which can be taken from Ansible Galaxy by means of ansible-galaxy install, write a playbook which will run Windows updates for categories:

- "CriticalUpdates"
- "SecurityUpdates"
- "Updates".

```
  GNU nano 2.3.1            File: vars/main.yml

---

patching_windows_reboot: True
patching_windows_reboot_timeout: 7200
patching_windows_categories:
  - "CriticalUpdates"
  - "SecurityUpdates"
  - "Updates"
```

```
  GNU nano 2.3.1              File: tasks/main.yml

---

- name: Install Windows updates
  win_updates:
    category_names: "{{ patching_windows_categories }}"
    reboot: "{{ 'yes' if patching_windows_reboot else 'no' }}"
    reboot_timeout: "{{ patching_windows_reboot_timeout }}"
  register: win_updates_result
  failed_when: not patching_windows_reboot and win_updates_result.failed is defined and $

- name: Reboot and Retry
  block:

    - name: Reboot Windows to retry update installation
      win_reboot:
        reboot_timeout: "{{ patching_windows_reboot_timeout }}"

    - name: Install Windows Updates Retry
      win_updates:
        category_names: "{{ patching_windows_categories }}"
        reboot: "{{ 'yes' if patching_windows_reboot else 'no' }}"
        reboot_timeout: "{{ patching_windows_reboot_timeout }}"
```

For this purpose an available role sparknsh.patching_window role can be used.

```
[root@centos7 windows]# ansible-galaxy install sparknsh.patching_windows
- downloading role 'patching_windows', owned by sparknsh
- downloading role from https://github.com/sparknsh/ansible-role-patching-windows/archive
/v1.0.3.tar.gz
- extracting sparknsh.patching_windows to /root/.ansible/roles/sparknsh.patching_windows
- sparknsh.patching_windows (v1.0.3) was installed successfully
```

When running a play book by ad-hoc command a secret should be decrypted by means of vault-password-file argument.

```
[root@centos7 windows]# ansible-playbook playbook.yml -i inventory --vault-passw
ord-file pass.txt

PLAY [Test playbook] ***********************************************************

TASK [Gathering Facts] *********************************************************
ok: [windows2016]

TASK [sparknsh.patching_windows : Install Windows updates] *********************
ok: [windows2016]

TASK [sparknsh.patching_windows : Reboot Windows to retry update installation] *
**
changed: [windows2016]

TASK [sparknsh.patching_windows : Install Windows Updates Retry] ***************
ok: [windows2016]

PLAY RECAP *********************************************************************
windows2016                : ok=4    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```