



redhat.

# GÉRER WINDOWS AVEC ANSIBLE



Michael Lessard  
Architecte de Solutions sénior

 [michaellessard](#)

Février 2018

A

# Agenda

- Qu'est ce qu'Ansible fait pour Windows ?
- Historique
- Comment Ansible travaille avec Windows
- Modules windows disponibles
- Méthodes d'authentification
- Prérequis
- Validation
- Exemples de playbook

# Qu'est ce que Ansible fait pour Windows ?



Avec le support Windows natif d'Ansible, vous pouvez :

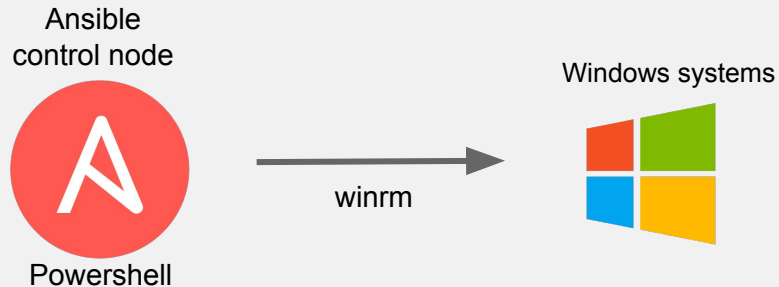
- Récupérer les faits des machines Windows
- Installer et désinstaller des MSIs
- Activer et désactiver les fonctionnalités Windows
- Démarrer, arrêter, et gérer des services Windows
- Créer et gérer des usagers et des groupes locaux ou AD
- Gérer des paquetages Windows via [Chocolatey package manager](#)
- Gérer et installer des mises à jour Windows
- Récupérer des fichiers d'un site distant
- Pousser et exécuter vos scripts PowerShell

# Historique

- Modules Ansible pour Windows
  - V 1.8 : 10
  - V 1.9 : 14
  - V 2.0 : 30
  - V 2.1 : 37
  - V 2.2 : 42
  - V 2.3 : 54
  - V 2.4 : 74
  - V 2.5 (beta) : ~80

# Comment Ansible travaille avec Windows ?

Les modules Ansible pour Windows sont écrits en powershell et exécuter au travers winrm (Windows Remote Management)



# MODULES WINDOWS

`win_acl` - Set file/directory/registry permissions for a system user or group  
`win_acl_inheritance` - Change ACL inheritance  
`win_chocolatey` - Manage packages using chocolatey  
`win_command` - Executes a command on a remote Windows node  
`win_copy` - Copies files to remote locations on windows hosts  
`win_defrag` - Consolidate fragmented files on local volumes.  
`win_disk_image` - Manage ISO/VHD/VHDX mounts on Windows hosts  
`win_dns_client` - Configures DNS lookup on Windows hosts  
`win_domain` - Ensures the existence of a Windows domain.  
`win_domain_controller` - Manage domain controller/member server state for a Windows host  
`win_domain_group` - creates, modifies or removes domain groups  
`win_domain_membership` - Manage domain/workgroup membership for a Windows host  
`win_domain_user` - Manages Windows Active Directory user accounts  
`win_dotnet_ngen` - Runs ngen to recompile DLLs after .NET updates  
`win_dsc` - Invokes a PowerShell DSC configuration  
`win_environment` - Modifies environment variables on windows hosts.  
`win_eventlog` - Manage Windows event logs  
`win_eventlog_entry` - Write entries to Windows event logs  
`win_feature` - Installs and uninstalls Windows Features on Windows Server  
`win_file` - Creates, touches or removes files or directories.  
`win_file_version` - Get DLL or EXE file build version  
`win_find` - return a list of files based on specific criteria  
`win_firewall` - Enable or disable the Windows Firewall  
`win_firewall_rule` - Windows firewall automation  
`win_get_url` - Fetches a file from a given URL  
`win_group` - Add and remove local groups  
`win_group_membership` - Manage Windows local group membership  
`win_hotfix` - install and uninstalls Windows hotfixes  
`win_iis_virtualdirectory` - Configures a virtual directory in IIS.  
`win_iis_webapplication` - Cowin\_acl - Set file/directory/registry permissions for a system user or group  
`win_iis_webapppool` - configures an IIS Web Application Pool  
`win_iis_webbinding` - Configures a IIS Web site.  
`win_iis_website` - Configures a IIS Web site.  
`win_lineinfile` - Ensure a particular line is in a file, or replace an existing line using a back-referenced regular expression.  
`win_mapped_drive` - maps a network drive for a user  
`win_msg` - Sends a message to logged in users on Windows hosts.

`win_msi ***(D)**` - Installs and uninstalls Windows MSI files  
`win_nssm` - NSSM - the Non-Sucking Service Manager  
`win_owner` - Set owner  
`win_package` - Installs/uninstalls an installable package  
`win_pagefile` - Query or change pagefile configuration  
`win_path` - Manage Windows path environment variables  
`win_ping` - A windows version of the classic ping module  
`win_power_plan` - Changes the power plan of a Windows system  
`win_psexec` - Runs commands (remotely) as another (privileged) user  
`win_psmodule` - Adds or removes a Powershell Module.`win_rabbitmq_plugin`  
`win_reboot` - Reboot a windows machine  
`win_reg_stat` - returns information about a Windows registry key or property of a key  
`win_regedit` - Add, change, or remove registry keys and values  
`win_region` - Set the region and format settings  
`win_regmerge` - Merges the contents of a registry file into the windows registry  
`win_robocopy` - Synchronizes the contents of two directories using Robocopy  
`win_route` - Add or remove a static route.  
`win_say` - Text to speech module for Windows to speak messages and optionally play sounds  
`win_scheduled_task` - Manage scheduled tasks  
`win_security_policy` - changes local security policy settings  
`win_service` - Manages Windows services  
`win_share` - Manage Windows shares  
`win_shell` - Execute shell commands on target hosts.  
`win_shortcut` - Manage shortcuts on Windows  
`win_stat` - returns information about a Windows file  
`win_tempfile` - Creates temporary files and directories.  
`win_template` - Templates a file out to a remote server.  
`win_timezone` - Sets Windows machine timezone  
`win_toast` - Sends Toast windows notification to logged in users on Windows 10 or later hosts  
`win_unzip` - Unzips compressed files and archives on the Windows node  
`win_updates` - Download and install Windows updates  
`win_uri` - Interacts with webservices  
`win_user` - Manages local Windows user accounts  
`win_user_right` - Manage Windows User Rights  
`win_wait_for` - Waits for a condition before continuing  
`win_wakeonlan` - Send a magic Wake-on-LAN (WoL) broadcast packet  
`win_webpicmd` - Installs packages using Web Platform Installer command-line

# À venir (Ansible 2.5)

- Améliorations avec become
- win\_updates: gestion multi-reboot, blacklist
- win\_certificate
- win\_xml
- win\_disk\_management
- Windows Nano server



# MÉTHODES D'AUTHENTIFICATION

# Méthodes d'authentification

[http://docs.ansible.com/ansible/latest/intro\\_windows.html#authentication-options](http://docs.ansible.com/ansible/latest/intro_windows.html#authentication-options)

Option	Local Account	Active directory Account	Credential Delegation
Basic	Yes	No	No
Certificate	Yes	No	No
Kerberos	No	Yes	Yes
NTLM	Yes	Yes	no
CredSSP	Yes	Yes	Yes

Autres options :

- OpenSSH pour Windows (<https://github.com/PowerShell/Win32-OpenSSH>)
- pywinrm secure sans certificat SSL (beta)

# PRÉREQUIS

# Prérequis LINUX (CREDSSP)

- Une machine Linux avec Ansible 2.4
  - Pour RHEL/Centos/Fedora : `yum install python2-winrm python2-requests`
  - Dans `group_vars/windows.yaml` , ajoutez ce-ci :
    - `ansible_user: Administrator`
    - `ansible_password: somepassword`
    - `ansible_port: 5986`
    - `ansible_connection: winrm`
    - `ansible_winrm_server_cert_validation: ignore`
    - `ansible_winrm_transport: credssp`
  - Des modules python additionnels :
    - `pip install "pyOpenSSL>=17.3.0"` (il y a un bogue voir l'exemple complet)
    - `pip install "pywinrm[credssp]"`

# Prérequis WINDOWS (CREDSSP)

- Windows 7 sp1 ou Windows 2008 sp1 +
- Powershell 3 (mais 5 est requis pour certains modules)  
<https://github.com/jborean93/ansible-windows/blob/master/scripts/Upgrade-PowerShell.ps1>
- Configurer CredSSP
  - Exemple : Windows 2016
    - Télécharger le script suivant :  
<https://github.com/ansible/ansible/blob/devel/examples/scripts/ConfigureRemotingForAnsible.ps1>
    - Démarrer powershell (run as administrator)
      - `.\ConfigureRemotingForAnsible.ps1 -CertValidityDays 3650 -EnableCredSSP`

# VALIDATION

# Validation

## PING

```
# ansible windows -i hosts -m win_ping
34.229.11.47 | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
```

## RÉCUPÉRER LES FAITS

```
# ansible windows -i hosts -m setup
```

# EXEMPLES DE PLAYBOOK



# Installe Firefox avec Chocolatey

```
---  
- name: Install Firefox using Chocolatey  
  hosts: all  
  tasks:  
    - name: Install Firefox  
      win_chocolatey:  
        name: firefox  
        state: present
```

# Mise à jour et redémarrage si nécessaire

```
---  
- name: Update  
  hosts: all  
  tasks:  
    - name: update windows  
      win_updates:  
        register: update_result  
  
    - debug: var=update_result  
  
- name : reboot if required  
  win_reboot:  
    when: update_result.reboot_required
```

# Créer un usager local

```
---  
- name: Create a user  
  hosts: all  
  tasks:  
    - name: Ensure user bob is present  
      win_user:  
        name: bob  
        password: B0bP4ssw0rd  
        state: present  
        groups:  
          - Users
```

# Become

```
---  
- name: Disable Zune Music and Zune Video appx  
  win_shell: |  
    Get-AppxPackage -name "Microsoft.ZuneMusic" | Remove-AppxPackage  
    Get-AppxPackage -name "Microsoft.ZuneVideo" | Remove-AppxPackage  
  become: yes  
  become_user: Administrator
```

# ipconfig

```
---  
- name: ipconfig  
  hosts: windows  
  tasks:  
    - name: run ipconfig  
      win_command: ipconfig  
      register: ipconfig  
  
    - debug: var=ipconfig
```

# stat

```
---
- name: Validate presence of win.ini
  hosts: windows
  tasks:
    - name: test stat module on file
      win_stat: path="C:/Windows/win.ini"
      register: stat_file

    - debug: var=stat_file

    - name: check stat_file result
      assert:
        that:
          - "stat_file.stat.exists"
          - "not stat_file.stat.isdir"
          - "stat_file.stat.size > 0"
          - "stat_file.stat.md5"
```

# D'autres exemples de playbook

***Dag Wieers***

<https://github.com/crombeen/ansible>

# DÉMONSTRATION : EXEMPLE COMPLET AVEC CREDSSP



# Windows

:: Exécute powershell comme un administrateur

```
c:\> Invoke-WebRequest -OutFile ansible.ps1
https://raw.githubusercontent.com/ansible/ansible/devel/examples/scripts/ConfigureRemotingForAnsible.ps1
c:\> .\ansible.ps1 -CertValidityDays 3650 -EnableCredSSP
```

Pour configurer plusieurs systèmes windows : win\_psexec

# Linux (RHEL)

:: Installer et configurer Ansible et les requis

```
# subscription-manager repos --enable rhel-7-server-extras-rpms
# rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
# yum install ansible python2-winrm python2-requests python2-pip
# pip uninstall pyOpenSSL
# rm -rf /usr/lib64/python2.7/site-packages/OpenSSL/
# pip install "pyOpenSSL>=17.3.0"
# pip install "pywinrm[credssp]"

# mkdir ansible-windows ; cd ansible-windows
# mkdir group_vars ; cd group_vars
# vim windows.yaml
ansible_user: Administrator
Ansible_password: somepassword
ansible_port: 5986
ansible_connection: winrm
ansible_winrm_server_cert_validation: ignore
ansible_winrm_transport: credssp
```

```
# cd .. ; vim inventory
[windows]
54.86.171.5

# vim ansible.cfg
[defaults]
warnings = False
gathering = smart
ansible_winrm_server_cert_validation = ignore

# ansible windows -i inventory -m win_ping
34.235.166.197 | SUCCESS => {
  "changed": false,
  "failed": false,
  "ping": "pong"
}
```

```
# vim install_firefox.yaml
---
- name: Install Firefox using Chocolatey
hosts: all
tasks:
  - name: Install Firefox
    win_chocolatey:
      name: firefox
      state: present

# ansible-playbook -i inventory install_firefox.yaml
PLAY [Install firefox using Chocolatey ] *****

TASK [Gathering Facts] *****
ok: [34.235.166.197]

TASK [Install Firefox] *****
[WARNING]: Chocolatey was missing from this system, so it was installed during this task run.

changed: [34.235.166.197]

PLAY RECAP *****
34.235.166.197      : ok=2    changed=1    unreachable=0    failed=0
```



EXTRA

# BASIC AUTH

# Prérequis LINUX (BASIC AUTH)

- Une machine Linux avec Ansible 2.4
  - Pour RHEL/Centos/Fedora : `yum install python2-winrm python2-requests`
  - Dans `group_vars/windows.yaml`, ajoutez ce-ci :
    - `ansible_user: Administrator`
    - `ansible_password: somepassword`
    - `ansible_port: 5985`
    - `ansible_connection: winrm`
    - `ansible_winrm_server_cert_validation: ignore`

Note : Windows subsystem for Linux (WSL) peut être utilisé, mais est non supporté par Microsoft et Red Hat

# Prérequis WINDOWS (BASIC AUTH)

- Windows 7 sp1 ou Windows 2008 sp1 +
- Powershell 3 (mais 5 est requis pour certains modules)  
<https://github.com/jborean93/ansible-windows/blob/master/scripts/Upgrade-PowerShell.ps1>
- Configurer le mode d'authentification
  - Exemple : Windows 2016
    - Winrm est présent mais non configuré
      - `winrm set winrm/config/service/auth @{Basic="true"}`
      - `winrm set winrm/config/service @{AllowUnencrypted="true"}`
    - Ouvrir le port 5985 du pare-feu