

# PILOTER LES MACHINES WINDOWS

## PRÉREQUIS WINDOWS

- Windows est un système d'exploitation non POSIX
  - Différences majeures => modules spécifiques
  - L'hôte Windows doit
    - être version 7, 8.1, 10, Server 2008, 2008 R2
    - PowerShell 3.0
    - .NET 4.0
    - WinRM activé

# WINRM

- WinRM => Windows Remote Management
- Service Windows permettant de gérer l'ordinateur depuis le réseau
- Disposer d'un compte utilisateur est indispensable
- Protocole SOAP (transport HTTP)
- Deux composants principaux du service WinRM à paramétrer
  - *listener*
  - *service*

## ECOUTEURS WINRM (LISTENERS)

Les services WinRM écoutent les requêtes sur un ou plusieurs ports.

Pour afficher les écouteurs en cours d'exécution sur le service WinRM :

```
winrm enumerate winrm/config/Listener
```

## Exemple de retour possible

```
Address = *  
Transport = HTTP  
Port = 5985  
Hostname  
Enabled = true  
URLPrefix = wsman  
CertificateThumbprint  
ListeningOn = 10.0.2.15, 127.0.0.1, 192.168.56.155, ::1,  
fe80::5efe:10.0.2.15%6, fe80::5efe:192.168.56.155%8,  
fe80::ffff:ffff:fffe%2, fe80::203d:7d97:c2ed:ec78%3,  
fe80::e8ea:d765:2c69:7756%7
```

- **Transport:** HTTP ou HTTPS
- **Port:** Le port d'écoute
  - par défaut 5985 pour HTTP et 5986 pour HTTPS
  - correspond à la variable hôte *ansible\_port*
- **URLPrefix:** Le préfixe de l'URL du WS
  - par défaut wsman
  - Si modifié, la variable hôte *ansible\_winrm\_path* doit être définie
- **CertificateThumbprint:** Si HTTPS, il s'agit de l'empreinte numérique du certificat

# CONFIGURATION DU SERVICE

Si le service n'est pas actif, il faut exécuter en tant qu'Administrateur :

```
winrm quickconfig
```

ou

```
winrm quickconfig -transport:https
```

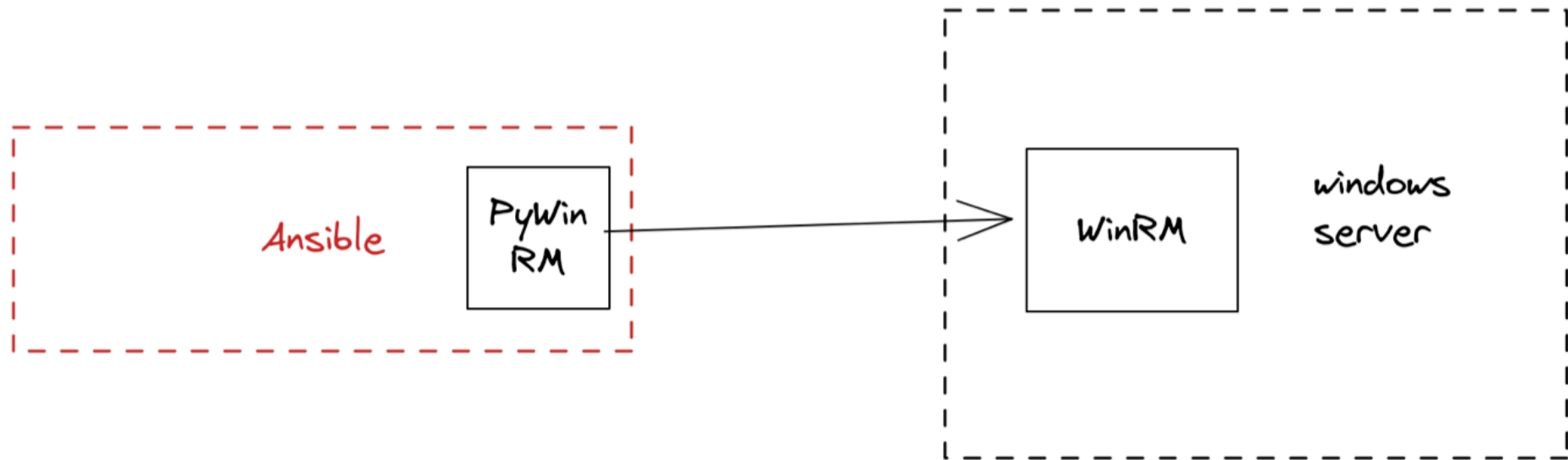
## Côté Ansible, pour chaque machine cible Windows :

```
ansible_host: 192.168.1.40
ansible_user: Edouard Durand
ansible_password: secret
ansible_connection: winrm
ansible_winrm_transport: ntlm
ansible_port: 5985
ansible_winrm_scheme: http
```

# PYWINRM

PyWinRM est un **client Python** pour le service Windows Remote Management (WinRM).

Il vous permet d'invoquer des commandes sur des machines Windows cibles à partir de n'importe quelle machine pouvant exécuter **Python**.





# INSTALLATION PYWINRRM

Tout simplement depuis pip

```
pip install pywinrm
```

Utilisation du module `ansible.builtin.winrm`

[https://docs.ansible.com/ansible/latest/collections/ansible/builtin/winrm\\_connection.html](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/winrm_connection.html)

# ANSIBLE ET POWERSHELL

Lancement de commandes powershell depuis le module du core d'Ansible `win_powershell`

```
- name: Run basic PowerShell script
  ansible.windows.win_powershell:
    script: |
      echo "Hello World"
```

# ANSIBLE ET CHOCOLATEY

Chocolatey est un gestionnaire de paquets pour windows.  
Un module est disponible dans le core d'Ansible pour l'utiliser :

```
- name: Install git
win_chocolatey:

name: git

state: present

- name: Upgrade installed packages
win_chocolatey:

name: all

state: latest

- name: Install notepadplusplus version 6.6
```