ANSIBLE
AUTOMATING WINDOWS

Phil Griffiths
Domain Solution Architect
Red Hat UK

James Bailey
SCCM Specialist
Worldpay
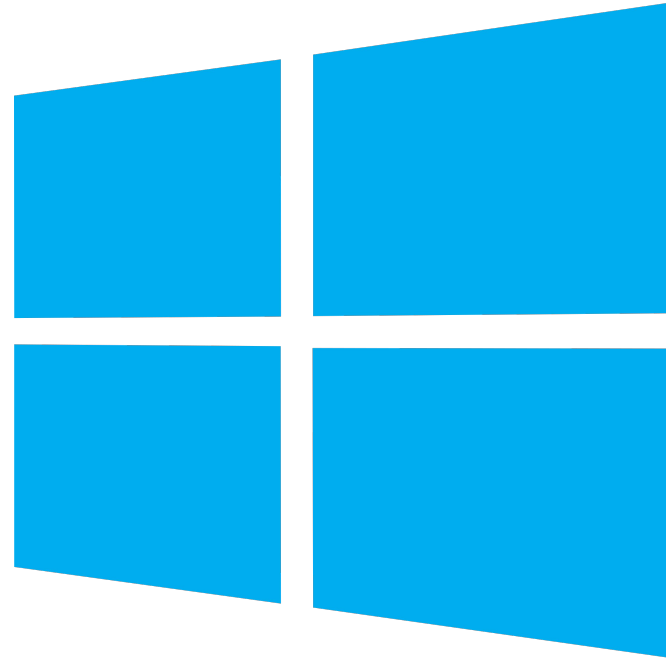
redhat

Ansible & Windows 101

LINUX

LIKE

LINUX

WINDOWS

LIKE

WINDOWS

- Windows 7, 8.1, and 10

- Windows Server 2008, 2008 R2, 2012, 2012 R2, and 2016

- Ansible requires PowerShell 3.0 or newer and at least .NET 4.0

- A WinRM listener/service should be created and activated

redhat.

ANSIBLE

The script ConfigureRemotingForAnsible.ps1 can be used to set up the basics

This sets up **HTTP(S)** listeners with a self-signed certificate and enables the *Basic* authentication

Other Authentication methods:

- SSL Certs
- NTLM
- Kerberos
- CredSSP
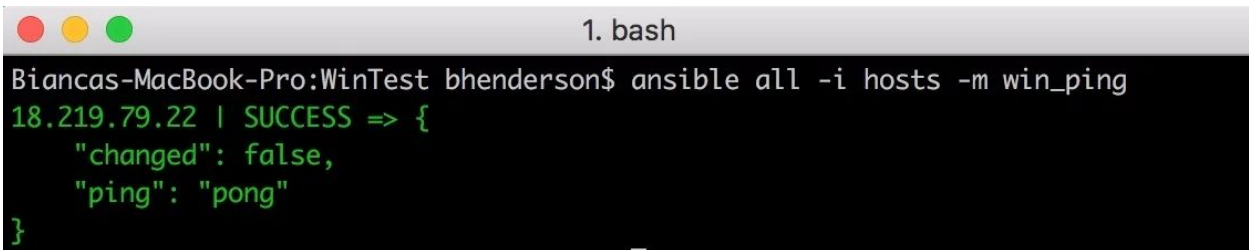
Ansible uses the pywinrm package to communicate with Windows servers over WinRM

redhat.

Windows support relies on a few standard variables to indicate:

- username
- password
- connection type

These variables are most easily set up in the inventory file, but can be set in *host_vars/ group_vars*

```
ansible_user: LocalUsername
ansible_password: Password
ansible_connection: winrm
ansible_winrm_transport: basic
ansible_winrm_server_cert_validation: ignore
```

```
● ● ●                              1. bash

Biancas-MacBook-Pro:WinTest bhenderson$ ansible all -i hosts -m win_ping
18.219.79.22 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Windows Modules

89 | 350+

Powershell DSC resources

```
$ ansible --version
2.7.5
$ ansible-doc -l | ^grep win_ | wc -l
```

Out of the box:

- Gather facts on Windows hosts
- Install and uninstall MSIs, Updates
- Enable and disable Windows Features
- Start, stop, and manage Windows services
- Create and manage local and AD users and groups
- Manage Windows packages via the Chocolatey package manager
- Push and execute any PowerShell scripts you write

Extend functionality:

- write your own modules in PowerShell

Ansible users have written modules for managing:

- Filesystem ACLs
- Windows Firewall
- Hostname, AD membership
- and more...

PLAYBOOK EXAMPLE: WINDOWS

```
- hosts: new_servers
  tasks:
  - name: ensure common OS updates are current
    win_updates:
      reboot: yes

    register: update_result


  - name: ensure domain membership
    win_domain_membership:
      dns_domain_name: contoso.corp
      domain_admin_user: '{{ domain_admin_username }}'
      domain_admin_password: '{{ domain_admin_password }}'
      state: domain
    register: domain_result


  - name: reboot and wait for host if domain change require it
    win_reboot:
    when: domain_result.reboot_required


  - name: ensure common tools are installed
    win_chocolatey:
      name: '{{ item }}'
    with_items: ['sysinternals', 'googlechrome']
```