



# MANAGE WINDOWS WITH ANSIBLE

**The what, the why and the how?**

Roland Wolters  
Senior Product Marketing Manager  
2019-05-08



# WHO AM I?

- Roland Wolters
- @liquidat
- > 10 years experience
- Consultant
- Project Manager
- Solution Architect
- Product Marketing Manager

# WHO AM I?

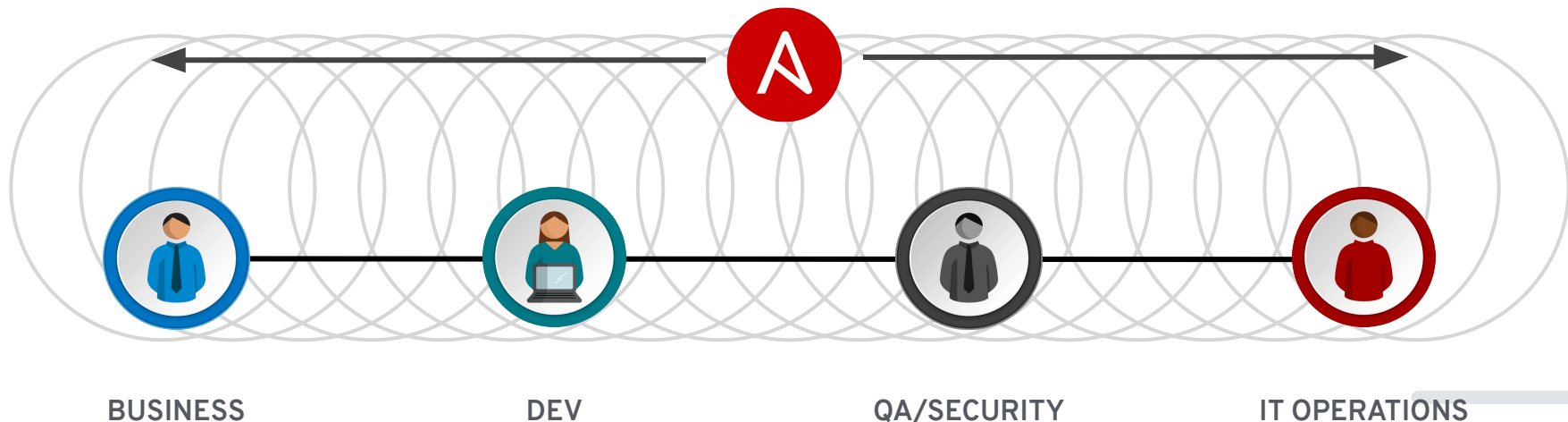
**Ansible Ninja**



# WHO AM I?



# ANSIBLE IS THE UNIVERSAL LANGUAGE



Ansible is the first **automation language** that can be read and written across IT.

Ansible is the only **automation engine** that can automate the entire application lifecycle and continuous delivery pipeline.

# ANSIBLE AUTOMATES TECHNOLOGIES YOU USE

## CLOUD

AWS  
Azure  
Digital Ocean  
Google  
OpenStack  
Rackspace  
**+more**

## OPERATING SYSTEMS

RHEL and Linux  
UNIX  
Windows  
**+more**

## VIRT & CONTAINER

Docker  
VMware  
RHV  
OpenStack  
OpenShift  
**+more**

## STORAGE

NetApp  
Red Hat Storage  
Infinidat  
**+more**

## WINDOWS

ACLs  
Files  
Packages  
IIS  
Regedit  
Shares  
Services  
Configs  
Users  
Domains  
**+more**

## NETWORK

Arista  
A10  
Cumulus  
Bigswitch  
Cisco  
Cumulus  
Dell  
F5  
Juniper  
Palo Alto  
OpenSwitch  
**+more**

## DEVOPS

Jira  
GitHub  
Vagrant  
Jenkins  
Bamboo  
Atlassian  
Subversion  
Slack  
Hipchat  
**+more**

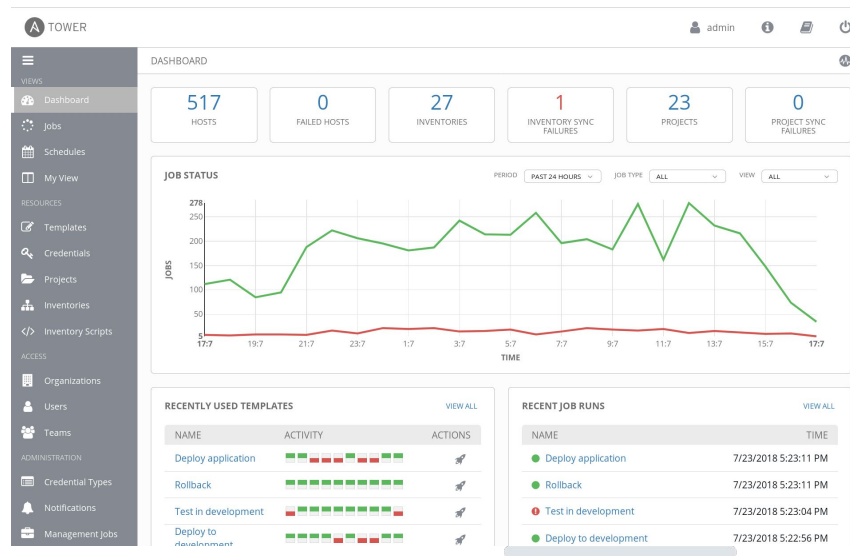
## MONITORING

Dynatrace  
Airbrake  
BigPanda  
Datadog  
LogicMonitor  
Nagios  
New Relic  
PagerDuty  
Sensu  
StackDriver  
Zabbix  
**+more**

# ANSIBLE TOWER

Ansible Tower is an **enterprise framework** for controlling, securing and managing your Ansible automation – with a **UI and RESTful API**.

- **Role-based access control**
- **Deploy** entire applications with **push-button deployment** access
- All automations are **centrally logged**
- Works with Linux nodes, networking devices - and of course Windows nodes



# ANSIBLE WINDOWS AUTOMATION

Use Ansible to deploy and manage Windows systems and applications.

**70+**

Windows Modules



# PLAYBOOK EXAMPLE: WINDOWS

```
- hosts: new_servers
```

```
tasks:
```

```
- name: install critical updates
```

```
  win_updates:
```

```
    category_names: CriticalUpdates
```

```
- name: add a domain user
```

```
  win_domain_user:
```

```
    name: somebody
```

```
    upn: somebody@mydomain.local
```

```
    groups:
```

```
      - Domain Admins
```

SO HOW DOES IT WORK?

# NOT SSH

- WinRM (HTTP-based remote shell protocol)
- Non-interactive logon
- Different connection plugin
- `pywinrm`

# STILL NOT SSH

- PSRP support since Ansible 2.7
- Faster, better
- File transfer
- Requires **pypsrp**

... Microsoft OpenSSH?

# POWERSHELL

- Unlike Python, "just there" on modern Windows
- We can use .NET
- Powershell 3+, Windows 7/Server 2008 RC2+

# INVENTORY

- Windows has its own connection type
- Variable in inventory must be set
- Similar to other target platforms

# INVENTORY EXAMPLE: WINDOWS

```
[windows]  
mssqlserver.example.com  
iisserver.example.com
```

```
[windows:vars]  
ansible_user=ansible  
ansible_password=password  
ansible_connection=winrm
```

**OR**

```
ansible_connection=psrp
```

# WHAT CAN WE DO?



# COMMANDS & SCRIPTS

# WINDOWS COMMAND

- Simply executes a command
- Not run through shell → no shell variables, no shell specific commands
- Quite secure
- No real idempotency

# WINDOWS COMMAND

- name: run a cmd command

win\_command: cmd.exe /c mkdir C:\temp

- name: run a vbs script

win\_command: cscript.exe script.vbs

- name: run from specific folder, skip when condition already met

win\_command: wbadmin -backupTarget:C:\backup\

args:

chdir: C:\somedir\

creates: C:\backup\

# WINDOWS SHELL

- Executes within a PowerShell
- Use PowerShell commands, variables, etc.
- Even multi-line scripts possible
- Less secure!
- No real idempotency

# WINDOWS SHELL

- name: run command through the shell

```
win_shell: Write-Host Hello world
```

- name: run multi-lined shell commands

```
win_shell: |
```

```
    $value = Test-Path -Path C:\temp
```

```
    if ($value) {
```

```
        Remove-Item -Path C:\temp -Force
```

```
    }
```

```
    New-Item -Path C:\temp -ItemType Directory
```

# SCRIPT

- Works on Linux and Windows
- Transfers and executes a script
- Local copy can still be templated!
- Only use in cases where the other modules don't work
- No real idempotency

# SCRIPT

- name: run a script  
script: /tmp/myscript.bat

# SOFTWARE MANAGEMENT



# APPLICATION INSTALLATION

## WAYS TO INSTALL SOFTWARE

### **win\_package**

The default module to install MSI or EXE

### **win\_chocolatey**

If possible, use Chocolatey! A package management framework for Windows - like the app stores on mobile phones, homebrew or the repositories on Linux distributions. Community driven.

### **win\_feature**

Installs or uninstalls Windows Roles or Features on Windows Server using the Add/Remove-WindowsFeature Cmdlets on Windows 2008 R2 and Install/Uninstall-WindowsFeature Cmdlets on Windows 2012.

### **win\_update**

Manage updates: install KBs, install all updates from a certain category and blacklist what does not fit your current setup.

### **win\_hotfix**

Install or remove windows hotfixes.

# APPLICATION INSTALLATION WITH WIN\_PACKAGE

- name: Install Visual C thingy

win\_package:

path: [http://download.microsoft.com/.../vcredist\\_x64.exe](http://download.microsoft.com/.../vcredist_x64.exe)

product\_id: '{CF2BEA3C-26EA-32F8-AA9B-331F7E34BA97}'

arguments:

- /install
- /passive
- /norestart

# APPLICATION INSTALLATION WITH WIN\_CHOCOLATEY

- name: Install multiple packages

win\_chocolatey:

name:

- procexp
- putty
- windirstat

state: present

# WINDOWS UPDATES

- Basic, synchronous updates - **win\_updates**
- Uses configured source (Windows Update/WSUS)
- (New in 2.5): transparent SYSTEM + auto reboot

# WINDOWS UPDATES

- name: install critical updates except blacklisted  
win\_updates:  
category\_names: CriticalUpdates  
reboot: yes # <--- new in 2.5!  
blacklist: # <--- new in 2.5!  
- KB4056892

# WINDOWS HOTFIX

- Install/Remove specific hotfixes
- Only does local installations, get hotfix beforehand
- Uses the DISM powershell module
- Thus Powershell version  $\geq 4$ , only on Windows 8+ and Server 2012+ installed by default

# WINDOWS HOTFIX

- name: install hotfix validating Identifier

win\_hotfix:

hotfix\_identifier:

Package\_for\_KB3172729~31bf3856ad364e35~amd64~~6.3.1.0

source:

C:\temp\windows8.1-kb3172729-x64\_e8003822a7ef4705cbb65623b72fd3ce  
c73fe222.msu

state: present

# REBOOTS

- `win_reboot` action makes managed reboots trivial
- `wait_for_connection` is just the second half



# REBOOTS

```
# Apply updates and reboot if necessary
```

```
- win_updates:
```

```
  register: update_result
```

```
- win_reboot:
```

```
  when: update_result.reboot_required
```

```
# Reboot a slow machine that might have lots of updates to apply
```

```
- win_reboot:
```

```
  shutdown_timeout: 3600
```

```
  reboot_timeout: 3600
```

# CONFIGURATION MANAGEMENT & SERVICES

# REGISTRY

- Manage individual key/value (win\_regedit)
- Manage idempotent bulk import (win\_regmerge)

# REGISTRY

- name: ensure registry value

win\_regedit:

path: HKLM\Software\Microsoft\Windows

name: SomeValueName

value: 0x12345

- name: merge registry data

win\_regmerge:

path: ComplexRegData.reg

# ACLS

- More granular than Linux permissions
- SDDL?!
- More like SELinux ACLs

# ACLs

- name: ensure owner recursively

win\_owner:

path: C:\Program Files\SomeApp

user: Administrator

recurse: true

- name: ensure complex ACLs

win\_acl:

path: C:\Temp

user: Users

rights: ReadAndExecute,Write,Delete

inherit: ContainerInherit,ObjectInherit

# SERVICES

- **win\_service** looks/acts like Linux service module
- Provides fine control over complex service behavior config in Windows SCM (who/what/when/how)

# SERVICES

- name: ensure IIS is running

win\_service:

name: W3Svc

state: running

- name: ensure firewall service is stopped/disabled

win\_service:

name: MpsSvc

state: stopped

start\_mode: disabled



# DOMAINS & CREDENTIALS

# DOMAINS

- Enterprise identity management
- Makes auth complex
- Ansible can do "throwaway" domains easily
- Promote/depromote Domain Controllers
- Joining/leaving domain is simple
- Manage basic domain objects

# DOMAINS

- name: create a domain

win\_domain:

dns\_domain\_name: mydomain.local

safe\_mode\_password: ItsASecret

- name: add a domain user

win\_domain\_user:

name: somebody

upn: somebody@mydomain.local

groups:

- Domain Admins

# BECOME

- Run with full privileges that are available to remote user
- Uses `runas` user
- Ansible  $\geq 2.5$ , else UAC and `SeTcbPrivilege`
- `become_user`: local or domain user account, local service accounts like System or NetworkService

# BECOME

- win\_whoami:
- win\_whoami:  
become: yes
- win\_whoami:  
become: yes  
become\_user: System

# DSC, AYNONE?

# WHAT ABOUT DSC?

## CONFIGURATIONS

- Declarative PowerShell scripts
- Define and configure instances of resources
- DSC will simply “make it so”
- Idempotent

## RESOURCES

- "Make it so" part of DSC
- Contain the code
- Files, Windows processes, VM running in Azure, etc.

# WHY USE ANSIBLE & DSC TOGETHER?

- Embed DSC in a broader automation approach
- Cover more than just Windows
- Tasks which are not idempotent by design



# WHY USE ANSIBLE & DSC TOGETHER?

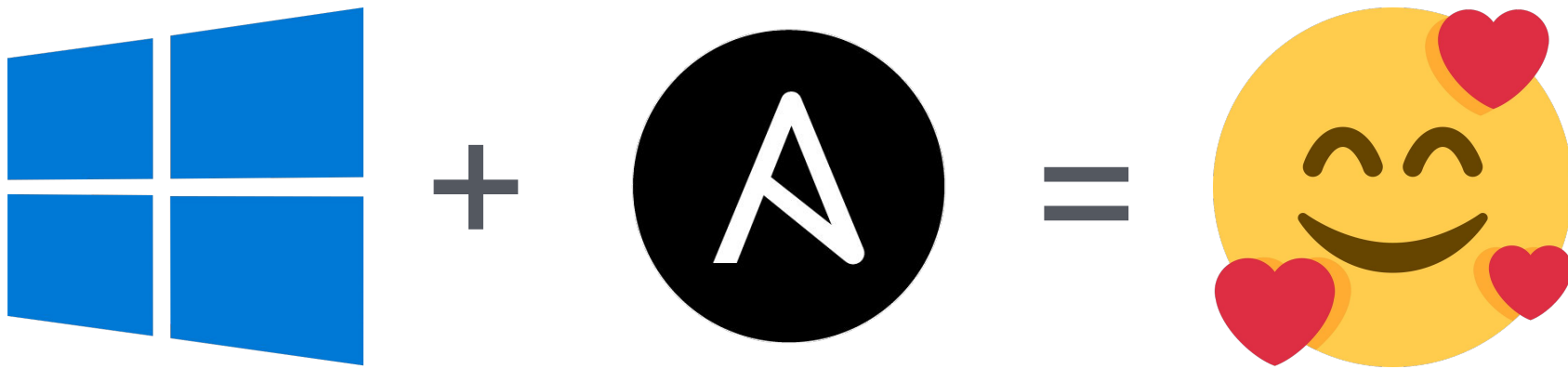
- Use Tower features via Ansible to govern DSC execution
- Manage DSC resources
- Free form module

# ANSIBLE & DSC - WE HAVE MODULES FOR THAT!

- name: install xDNSServer DSC module on target  
win\_psmodule:  
    name: xDnsServer
- name: create DNS zone  
win\_dsc:  
    resource\_name: xDnsServerPrimaryZone  
    name: createdbyansible.com

# WRAP-UP

# WRAP-UP



Windows is a first class citizen within the Ansible ecosystem!

# DO NEXT

## GET STARTED

- ✓ Try Tower for free  
[ansible.com/tower-trial](https://ansible.com/tower-trial)
- ✓ Three steps to start right off  
[ansible.com/get-started](https://ansible.com/get-started)
- ✓ Want to learn more?  
[ansible.com/resources](https://ansible.com/resources)

## FOCUS ON WINDOWS

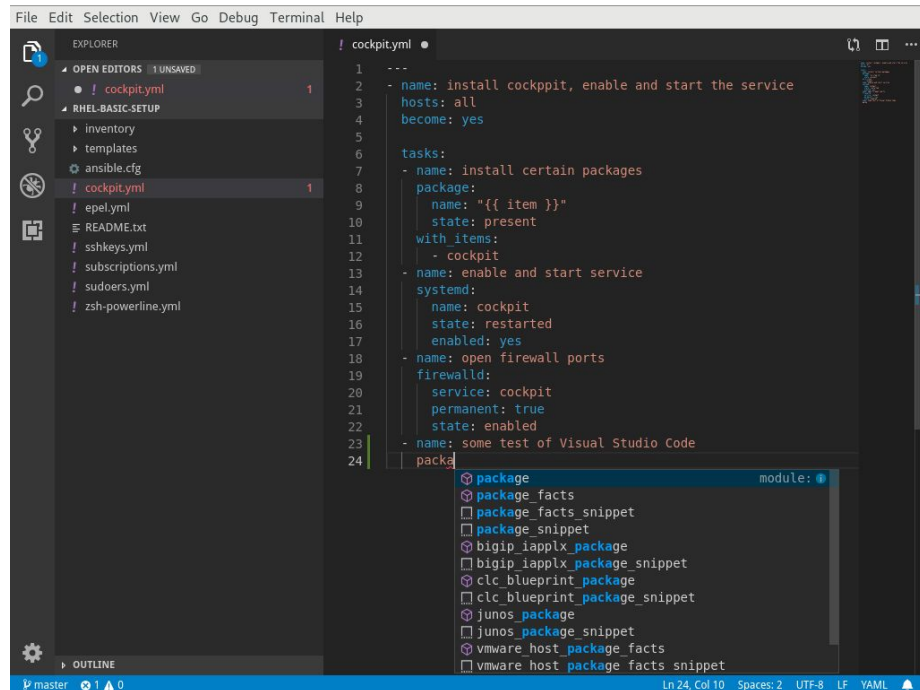
- ✓ Connect to your hosts  
[ansible.com/blog/connecting-to-a-windows-host](https://ansible.com/blog/connecting-to-a-windows-host)
- ✓ Check out roles for Windows platform  
on [galaxy.ansible.com](https://galaxy.ansible.com)
- ✓ Start with Ansible and Azure  
[docs.microsoft.com/en-us/azure/ansible/](https://docs.microsoft.com/en-us/azure/ansible/)

TIME FOR DEMO?

# USE VISUAL STUDIO CODE

**Lightweight** but **powerful** open source editor.  
And a rich Ansible extension is available -  
provided by Microsoft.

- Code completion
- Syntax highlighting
- Run playbooks



RED HAT  
**SUMMIT**

THANK YOU



[linkedin.com/company/Red-Hat](https://www.linkedin.com/company/Red-Hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/RedHatinc](https://www.facebook.com/RedHatinc)



[twitter.com/RedHat](https://twitter.com/RedHat)