



**Red Hat**

**RED HAT FORUM | September 10 2019**

**Manage Windows with Ansible**

**Roland Wolters**

**Red Hat Product Marketing**

# Who am I?

Roland Wolters

@liquidat

> 10 years experience

Consultant

Project Manager

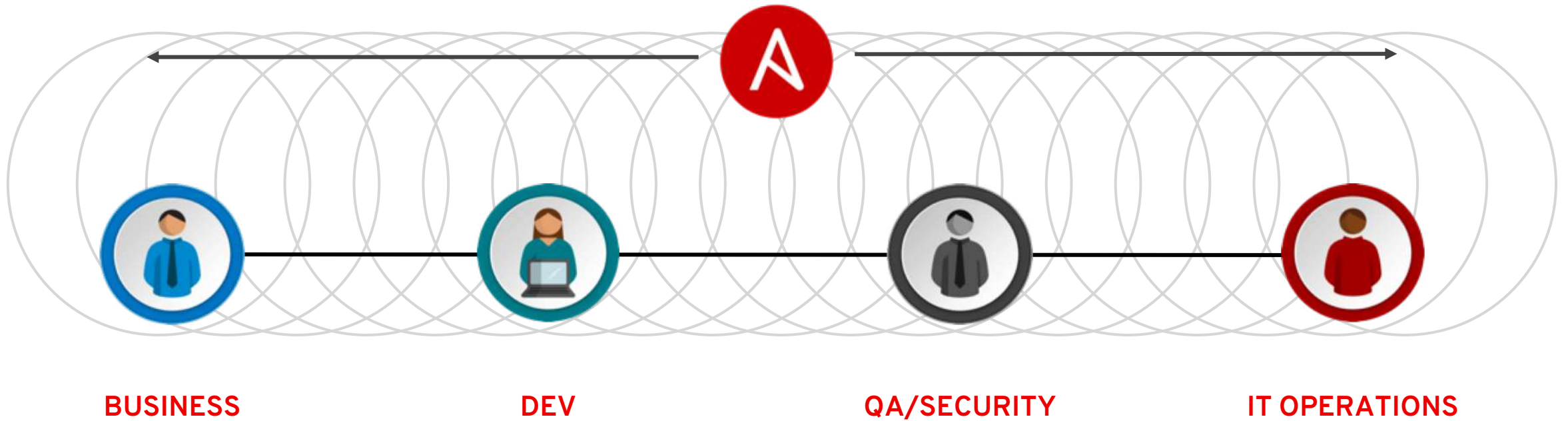
Solution Architect

Product Marketing Manager

# 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. Including different platforms like Microsoft Windows.

# 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

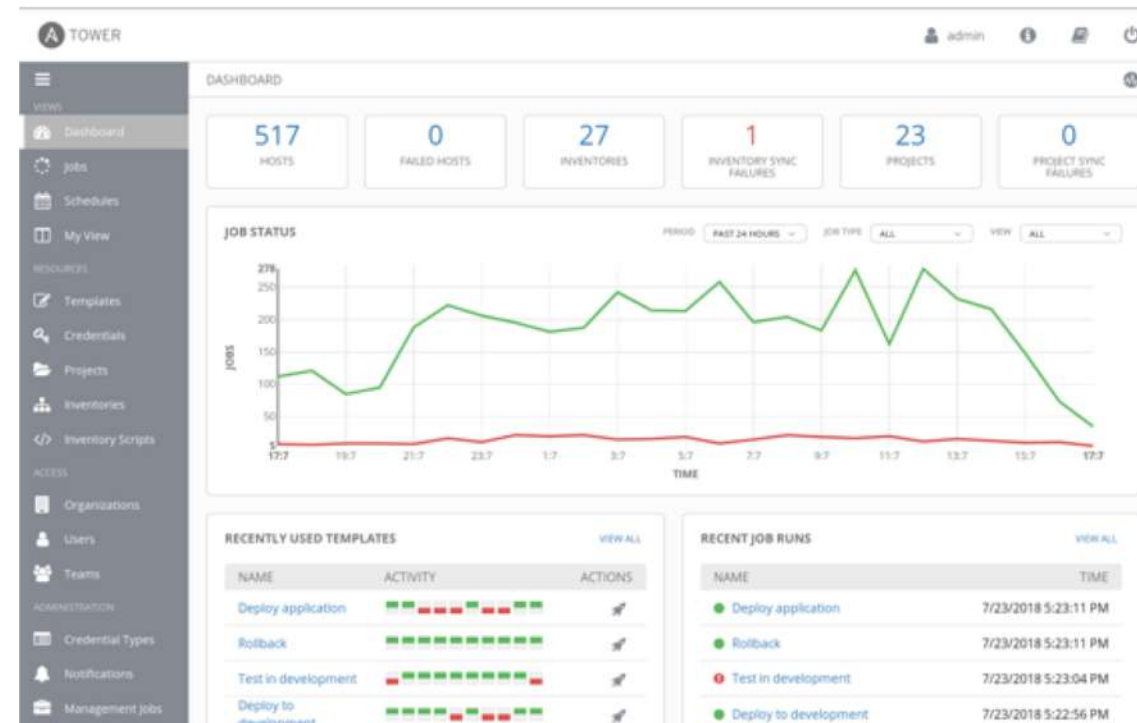
## MONITORING

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: ensure IIS is running

**win\_service:**

name: W3Svc

state: running

- 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

Requires `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_connection=winrm
```

OR

```
ansible_connection=psrp
```



What We Can Do  
Next Week!

Monday:

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\  
args:

creates: C:\backup\  
args:

# 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

Tuesday:

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 Feature

- name: Install IIS

  - win\_feature:

    - name: Web-Server

    - state: present

- name: Install IIS with sub features and management tools

  - win\_feature:

    - name: Web-Server

    - state: present

    - include\_sub\_features: yes

    - include\_management\_tools: yes

# 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

# 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
```

Wednesday:

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

Thursday:

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

# Authentication

Option	Local Accounts	Active Directory	Credential Delegation
Basic	Yes	No	No
Certificate	Yes	No	No
NTLM	No	Yes	Yes
Kerberos	Yes	Yes	No
CredSSP	Yes	Yes	Yes

# Authentication

Pick the one right for your use case

Some are more secure; some are easier to set up

Be aware of weaknesses of each type

# Authentication

ansible\_user: username@MY.DOMAIN.COM

ansible\_password: Password

ansible\_connection: winrm

ansible\_winrm\_transport: kerberos

ansible\_connection: winrm

ansible\_winrm\_cert\_pem: /path/to/certificate/public/key.pem

ansible\_winrm\_cert\_key\_pem: /path/to/certificate/private/key.pem

ansible\_winrm\_transport: certificate

Friday:

DSC, Anyone?

# 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



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/)

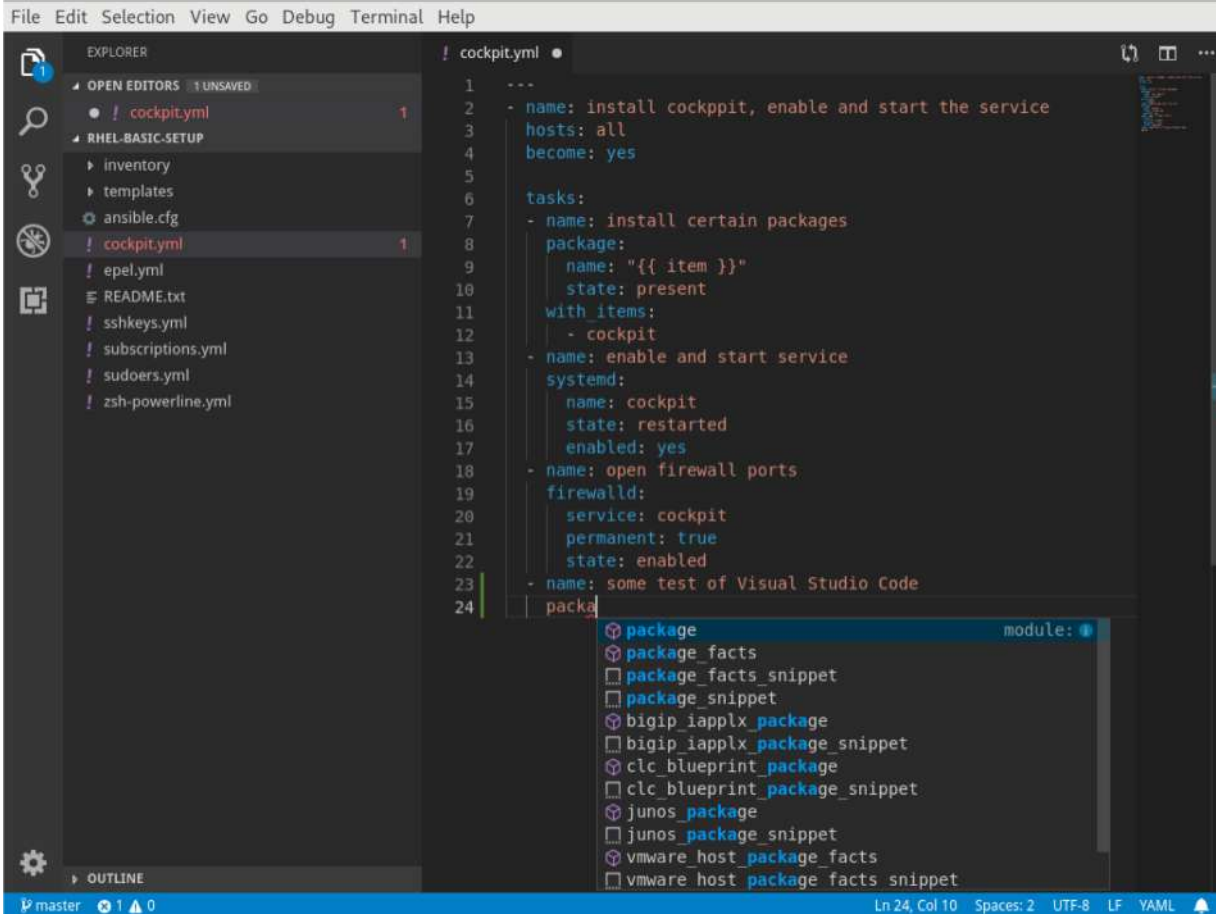


Hint:  
How To Code  
Best!

# 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
- Shows Microsoft's commitment to Ansible



The screenshot shows the Visual Studio Code interface with an Ansible playbook named 'cockpit.yml' open. The left sidebar shows the Explorer view with a file tree for 'RHEL-BASIC-SETUP' containing files like 'inventory', 'templates', 'ansible.cfg', 'cockpit.yml', 'epel.yml', 'README.txt', 'sshkeys.yml', 'subscriptions.yml', 'sudoers.yml', and 'zsh-powerline.yml'. The main editor area shows the following Ansible playbook content:

```
1 ---
2 - name: install cockpit, enable and start the service
3   hosts: all
4   become: yes
5
6   tasks:
7     - name: install certain packages
8       package:
9         name: "{{ item }}"
10        state: present
11      with_items:
12        - cockpit
13    - name: enable and start service
14      systemd:
15        name: cockpit
16        state: restarted
17        enabled: yes
18    - name: open firewall ports
19      firewallld:
20        service: cockpit
21        permanent: true
22        state: enabled
23  - name: some test of Visual Studio Code
24  packa
```

A code completion dropdown menu is visible at the bottom right, showing suggestions for 'package' and various Ansible modules like 'package\_facts', 'package\_snippet', 'bigip\_iapplx\_package', etc.





**Red Hat**

**RED HAT FORUM | September 10 2019**  
**Running AI, ML and analytics workloads**  
**on OpenShift using open data hub**

**Codrin Bucur, Principal Architect, Red Hat**  
**Michael Clifford, Data Scientist, Red Hat**  
**Sherard Griffin, Sr. Manager, Software Engineering,**  
**Red Hat**

# Thank you!

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



[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)