



Manage Windows with Ansible

Dmitry Alekhin
Solution Architect



ANSIBLE

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
Regedits
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

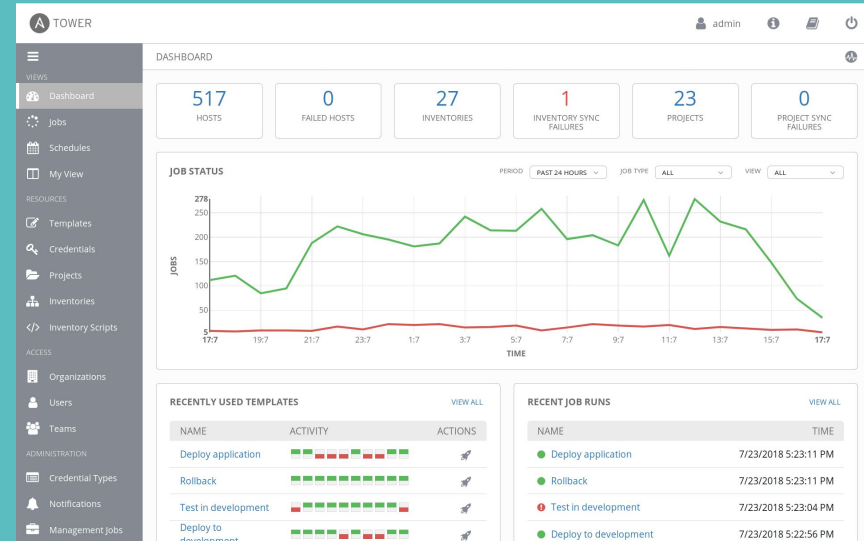
MONITORING

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.

100+

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
```

#ANSIBLEAUTOMATES

SO HOW DOES IT WORK?



ANSIBLE

First Way: WINRM

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

Second way: PSRP

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

Powershell

- Always in modern Windows Operating Systems
- Can use .NET
- Powershell 3+, Windows 7/Server 2008 RC2+

INVENTORY FILE

- 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
```

#ANSIBLEAUTOMATES

COMMANDS & SCRIPTS



ANSIBLE

WINDOWS SHELL

- 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:

chdir: C:\somedir\
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

SCRIPT

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

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_updates	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

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
```



# 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

- 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

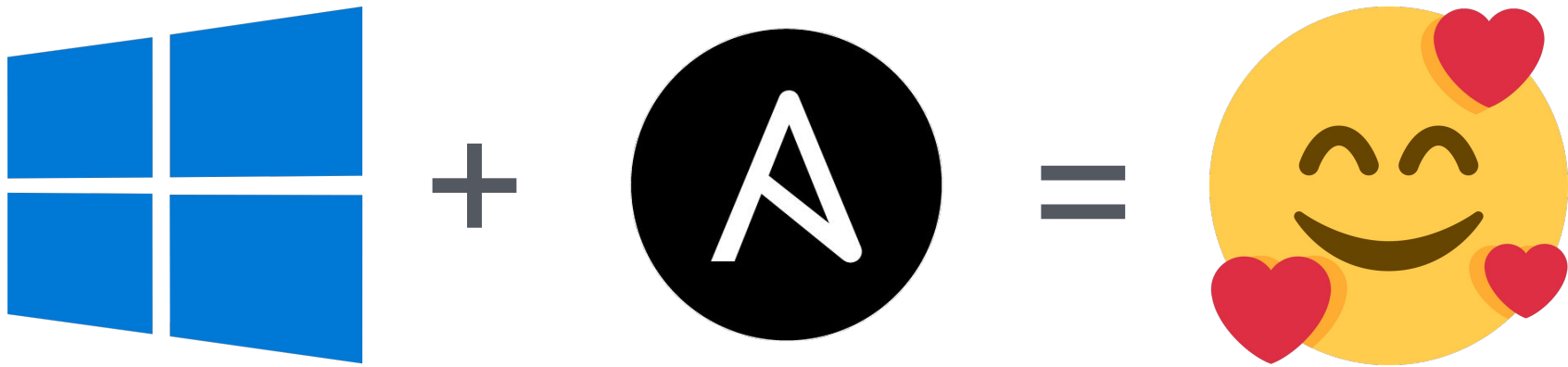
#ANSIBLEAUTOMATES

# WRAP-UP



ANSIBLE

# WRAP-UP



Windows is a first class citizen within the Ansible ecosystem!

#ANSIBLEAUTOMATES

# HOW TO CODE?

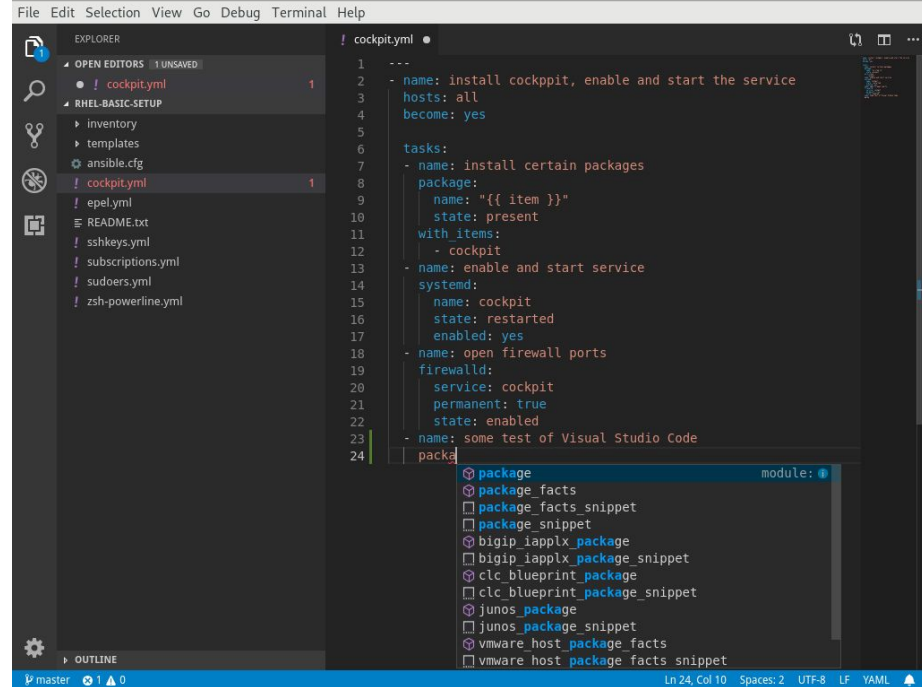


ANSIBLE

# 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



# 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/)

**Thank you!**