Red Hat
Ansible Automation Platform

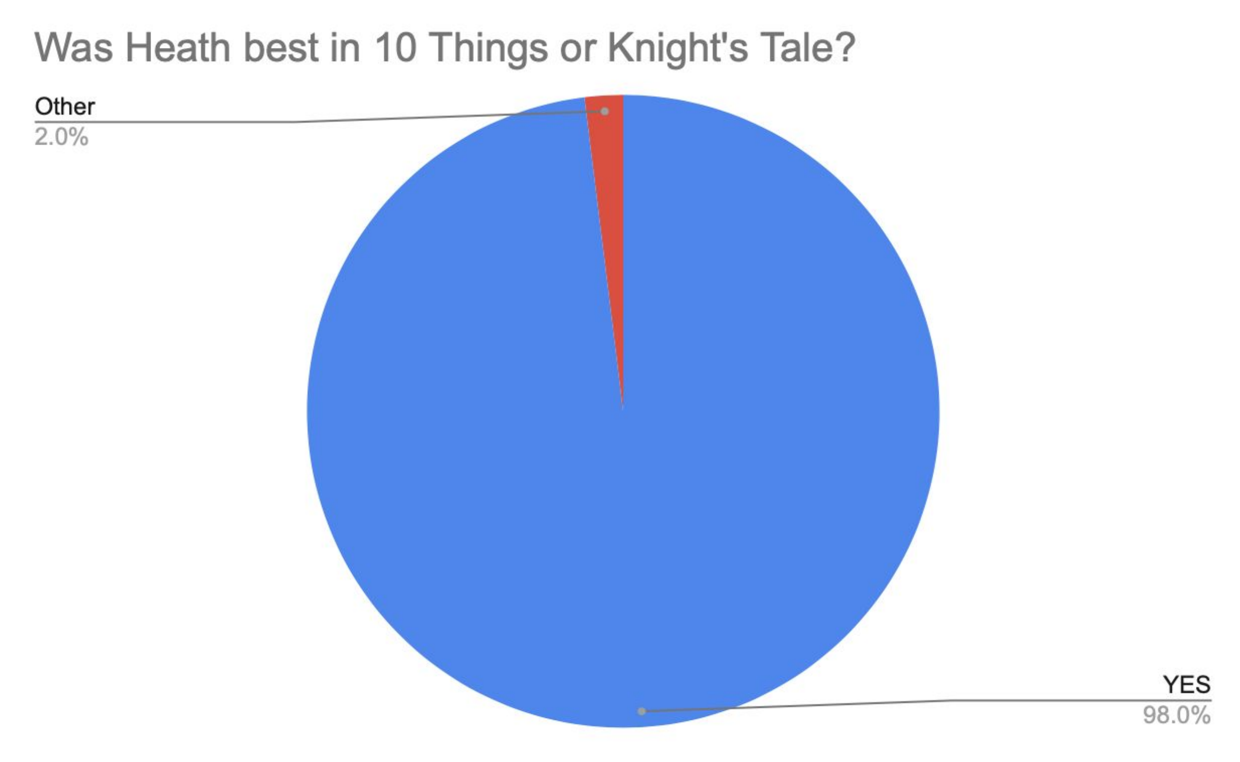
ANSIBLE AUTOMATES

10 Things I Hate About You: Manage Windows like Linux with Ansible

Colin McNaughton
Technical Marketing Manager - Ansible Automation

Red Hat

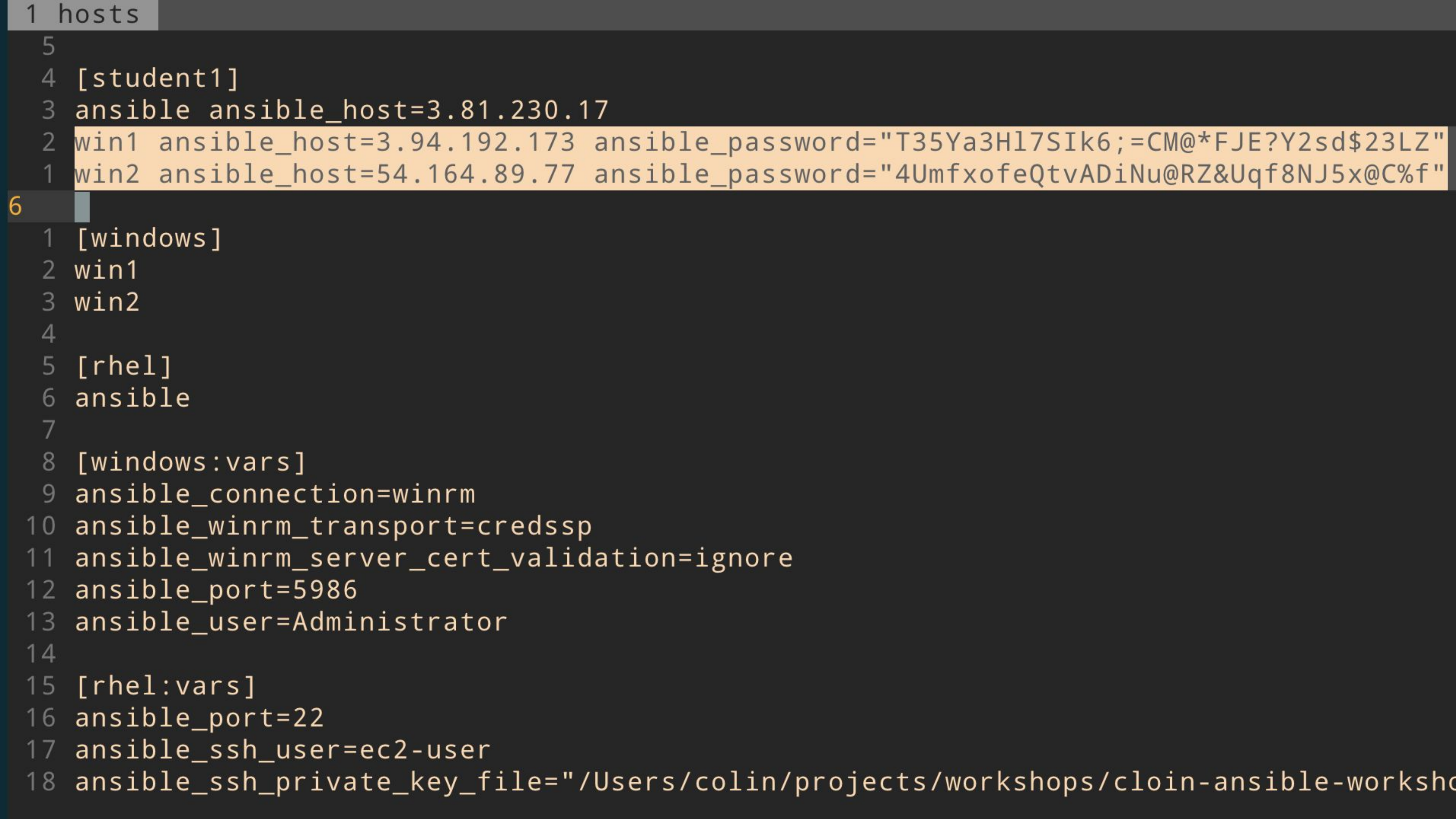
Who am I?


Was Heath best in 10 Things or Knight's Tale?

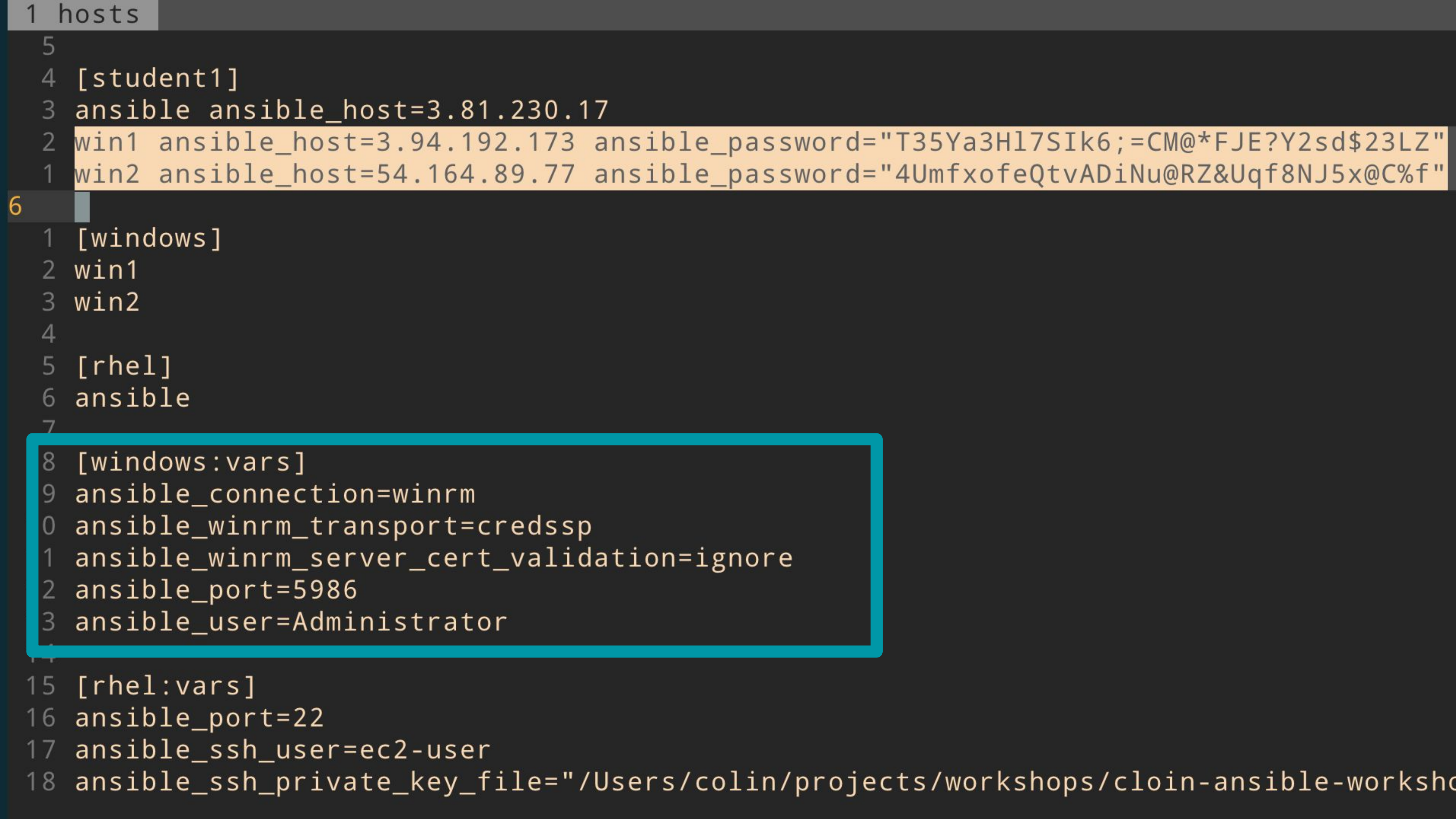
Not SSH

- WinRM (HTTP-based remote shell protocol)
- Non-interactive logon
- Different connection plugin
- What about Microsoft OpenSSH?

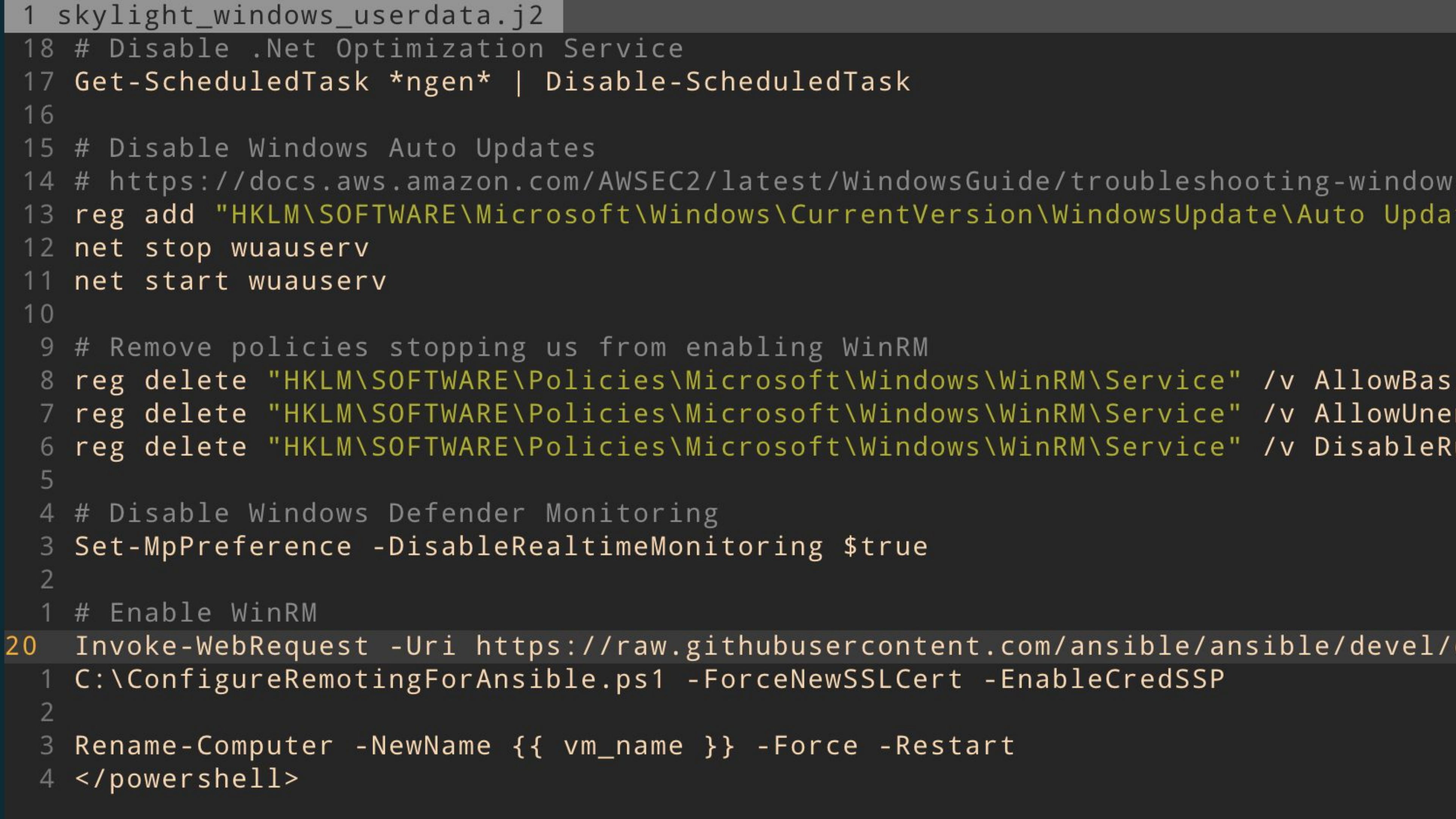
A little bit closer now: WinRM Connectivity



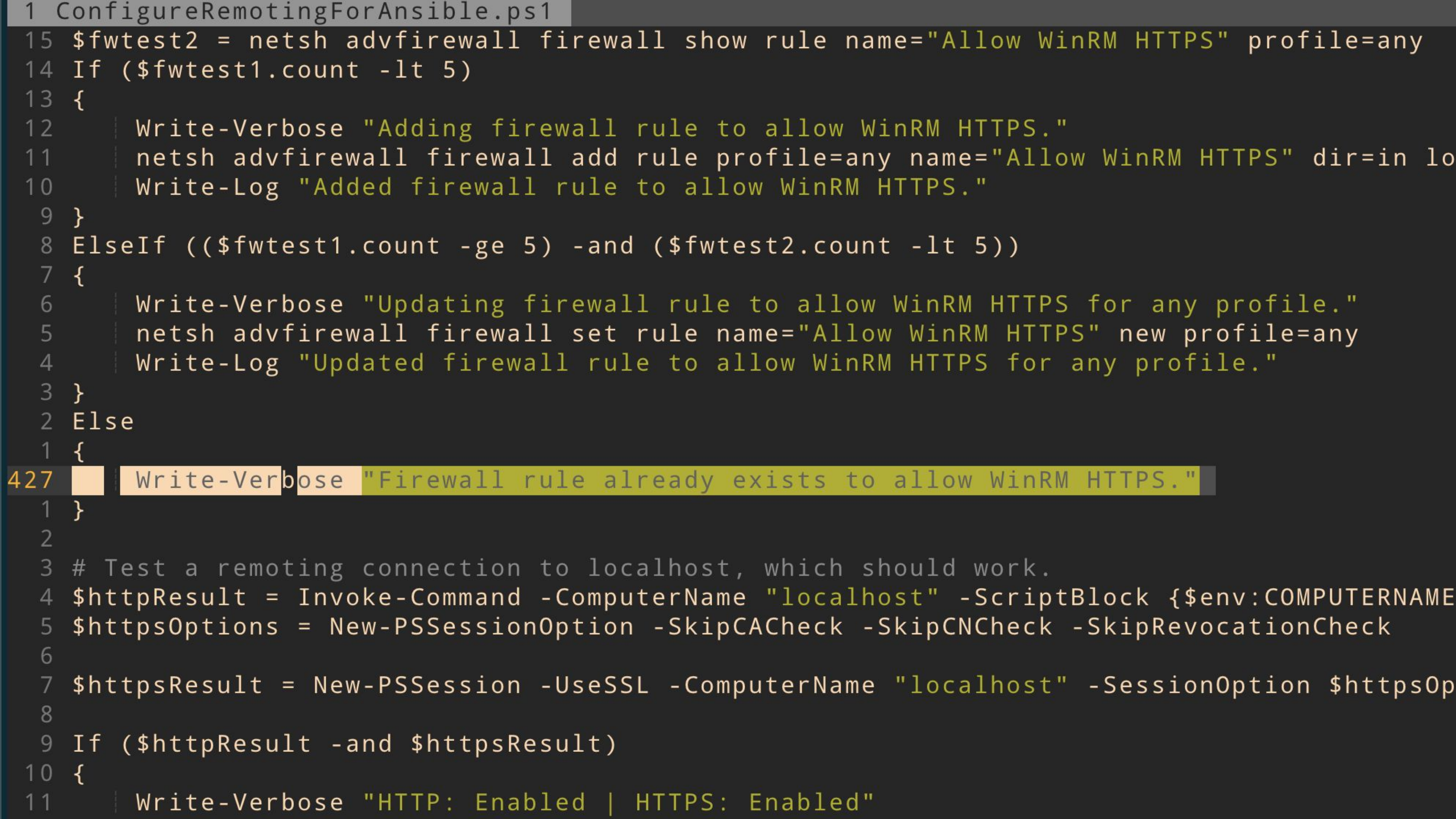
```

[student1]
ansible ansible_host=3.81.230.17
win1 ansible_host=3.94.192.173 ansible_password="T35Ya3Hl7SIk6;=CM@*FJE?Y2sd$23LZ"
win2 ansible_host=54.164.89.77 ansible_password="4UmfxofeQtvADiNu@RZ&Uqf8NJ5x@C%f"

[windows]
win1
win2

[rhel]
ansible

[windows:vars]
ansible_connection=winrm
ansible_winrm_transport=credssp
ansible_winrm_server_cert_validation=ignore
ansible_port=5986
ansible_user=Administrator

[rhel:vars]
ansible_port=22
ansible_ssh_user=ec2-user
ansible_ssh_private_key_file="/Users/colin/projects/workshops/cloin-ansible-worksho
```



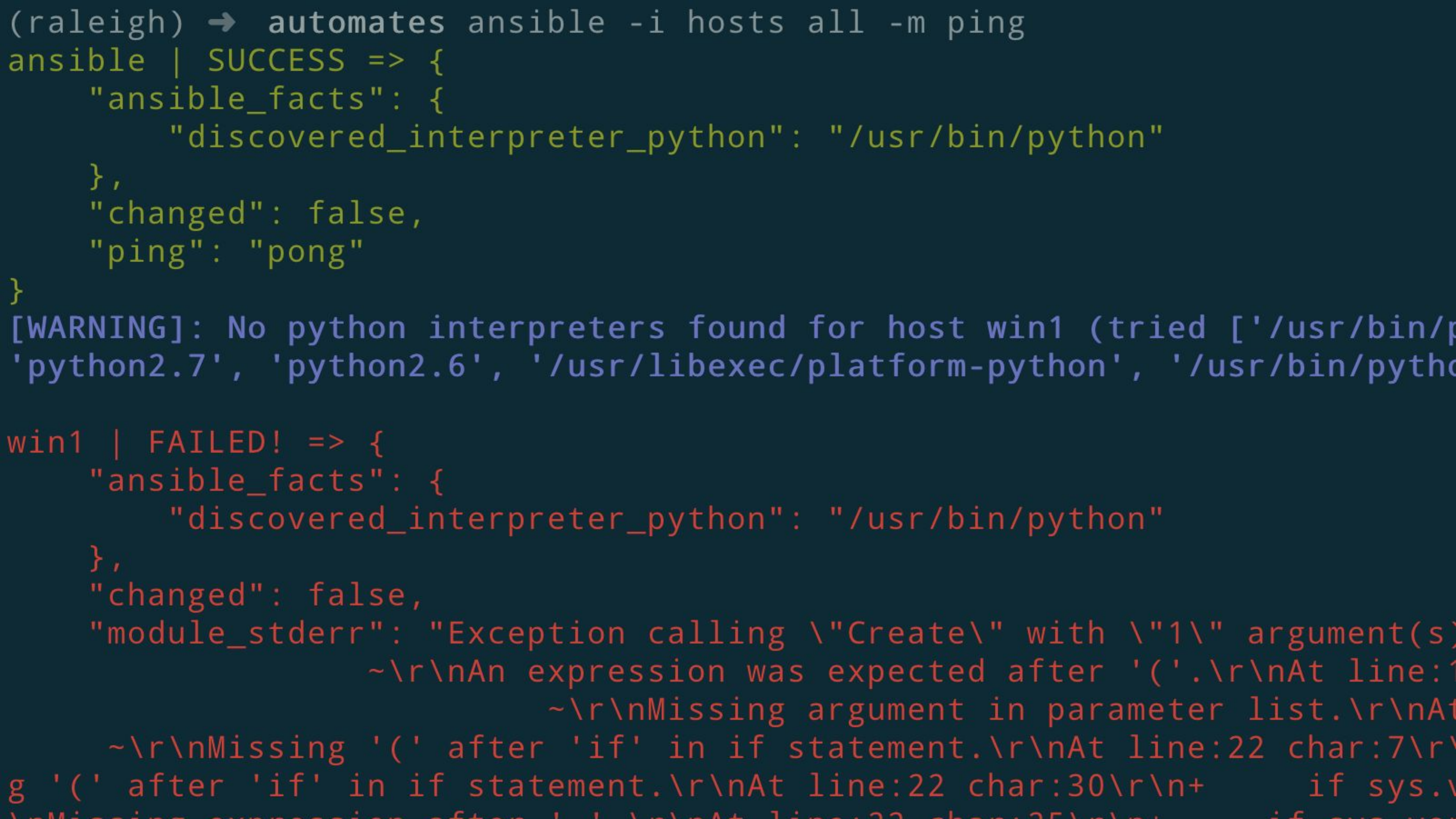
```

[student1]
ansible ansible_host=3.81.230.17
win1 ansible_host=3.94.192.173 ansible_password="T35Ya3Hl7SIk6;=CM@*FJE?Y2sd$23LZ"
win2 ansible_host=54.164.89.77 ansible_password="4UmfxofeQtvADiNu@RZ&Uqf8NJ5x@C%f"

[windows]
win1
win2

[rhel]
ansible

[windows:vars]
ansible_connection=winrm
ansible_winrm_transport=credssp
ansible_winrm_server_cert_validation=ignore
ansible_port=5986
ansible_user=Administrator

[rhel:vars]
ansible_port=22
ansible_ssh_user=ec2-user
ansible_ssh_private_key_file="/Users/colin/projects/workshops/cloin-ansible-worksho
```



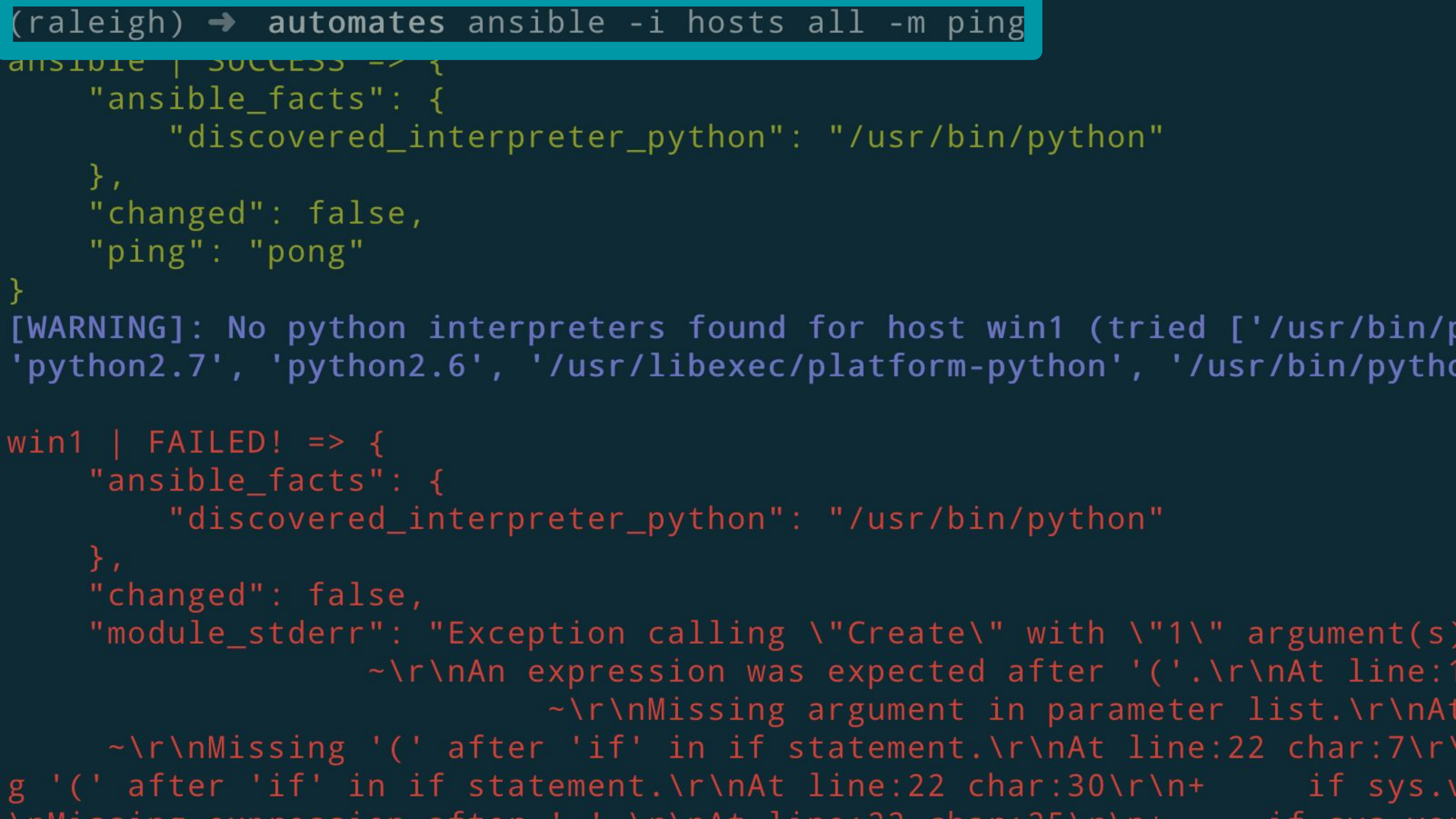
```
# Disable .Net Optimization Service
Get-ScheduledTask *ngen* | Disable-ScheduledTask

# Disable Windows Auto Updates
# https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/troubleshooting-window
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Upda
net stop wuauserv
net start wuauserv

# Remove policies stopping us from enabling WinRM
reg delete "HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service" /v AllowBas
reg delete "HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service" /v AllowUne
reg delete "HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service" /v DisableR

# Disable Windows Defender Monitoring
Set-MpPreference -DisableRealtimeMonitoring $true

# Enable WinRM
Invoke-WebRequest -Uri https://raw.githubusercontent.com/ansible/ansible/devel/
C:\ConfigureRemotingForAnsible.ps1 -ForceNewSSLCert -EnableCredSSP

Rename-Computer -NewName {{ vm_name }} -Force -Restart
</powershell>
```



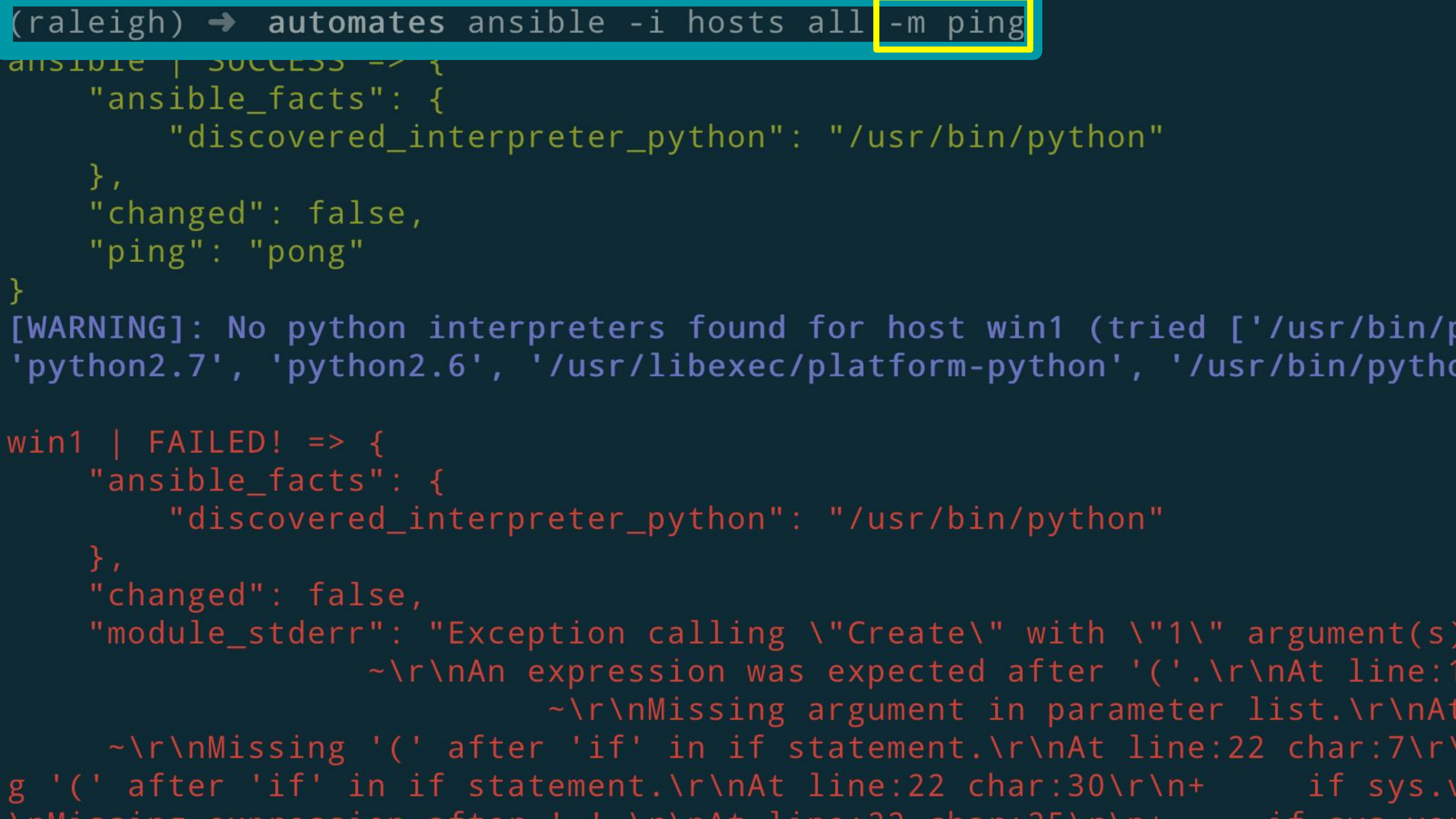
```
$fwtest2 = netsh advfirewall firewall show rule name="Allow WinRM HTTPS" profile=any
If ($fwtest1.count -lt 5)
{
    Write-Verbose "Adding firewall rule to allow WinRM HTTPS."
    netsh advfirewall firewall add rule profile=any name="Allow WinRM HTTPS" dir=in lo
    Write-Log "Added firewall rule to allow WinRM HTTPS."
}
ElseIf (($fwtest1.count -ge 5) -and ($fwtest2.count -lt 5))
{
    Write-Verbose "Updating firewall rule to allow WinRM HTTPS for any profile."
    netsh advfirewall firewall set rule name="Allow WinRM HTTPS" new profile=any
    Write-Log "Updated firewall rule to allow WinRM HTTPS for any profile."
}
Else
{
    Write-Verbose "Firewall rule already exists to allow WinRM HTTPS."
}

# Test a remoting connection to localhost, which should work.
$httpResult = Invoke-Command -ComputerName "localhost" -ScriptBlock {$env:COMPUTERNAME
$httpsOptions = New-PSSessionOption -SkipCACheck -SkipCNCheck -SkipRevocationCheck

$httpsResult = New-PSSession -UseSSL -ComputerName "localhost" -SessionOption $httpsOp

If ($httpResult -and $httpsResult)
{
    Write-Verbose "HTTP: Enabled | HTTPS: Enabled"
```



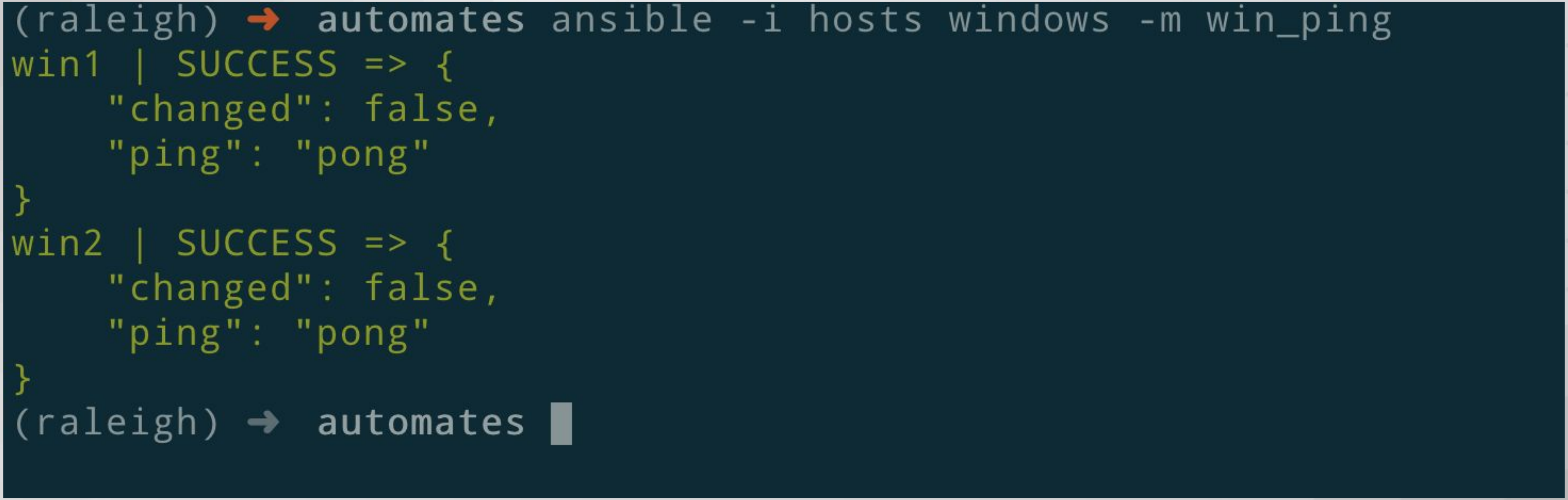
```
(raleigh) ➜  automates ansible -i hosts all -m ping
ansible | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: No python interpreters found for host win1 (tried ['/usr/bin/p
'python2.7', 'python2.6', '/usr/libexec/platform-python', '/usr/bin/pytho

win1 | FAILED! => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "module_stderr": "Exception calling \"Create\" with \"1\" argument(s
                  ~\r\nAn expression was expected after '('.\r\nAt line:1
                           ~\r\nMissing argument in parameter list.\r\nAt
     ~\r\nMissing '(' after 'if' in if statement.\r\nAt line:22 char:7\r
g '(' after 'if' in if statement.\r\nAt line:22 char:30\r\n+     if sys.v
\nMissing expression after '.'.\r\nAt line:22 char:35\r\n+     if sys.ve
```



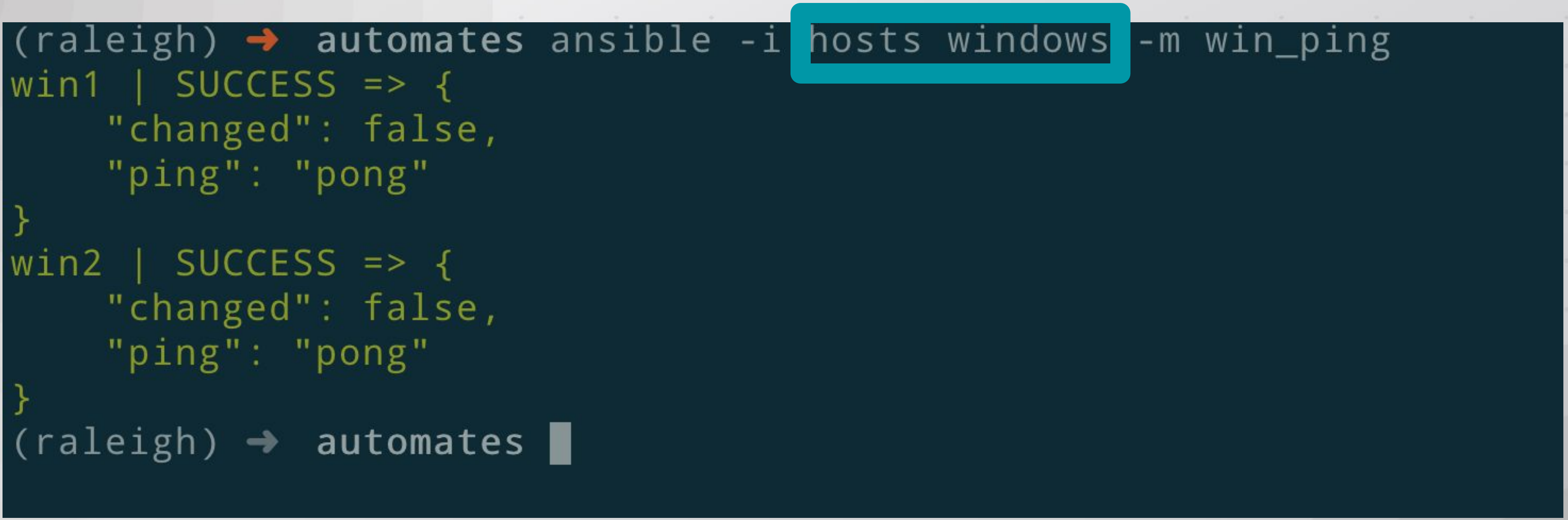
```
(raleigh) ➜  automates ansible -i hosts all -m ping
ansible | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: No python interpreters found for host win1 (tried ['/usr/bin/p
'python2.7', 'python2.6', '/usr/libexec/platform-python', '/usr/bin/pytho

win1 | FAILED! => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "module_stderr": "Exception calling \"Create\" with \"1\" argument(s
                  ~\r\nAn expression was expected after '('.\r\nAt line:1
                           ~\r\nMissing argument in parameter list.\r\nAt
     ~\r\nMissing '(' after 'if' in if statement.\r\nAt line:22 char:7\r\
g '(' after 'if' in if statement.\r\nAt line:22 char:30\r\n+     if sys.\
```



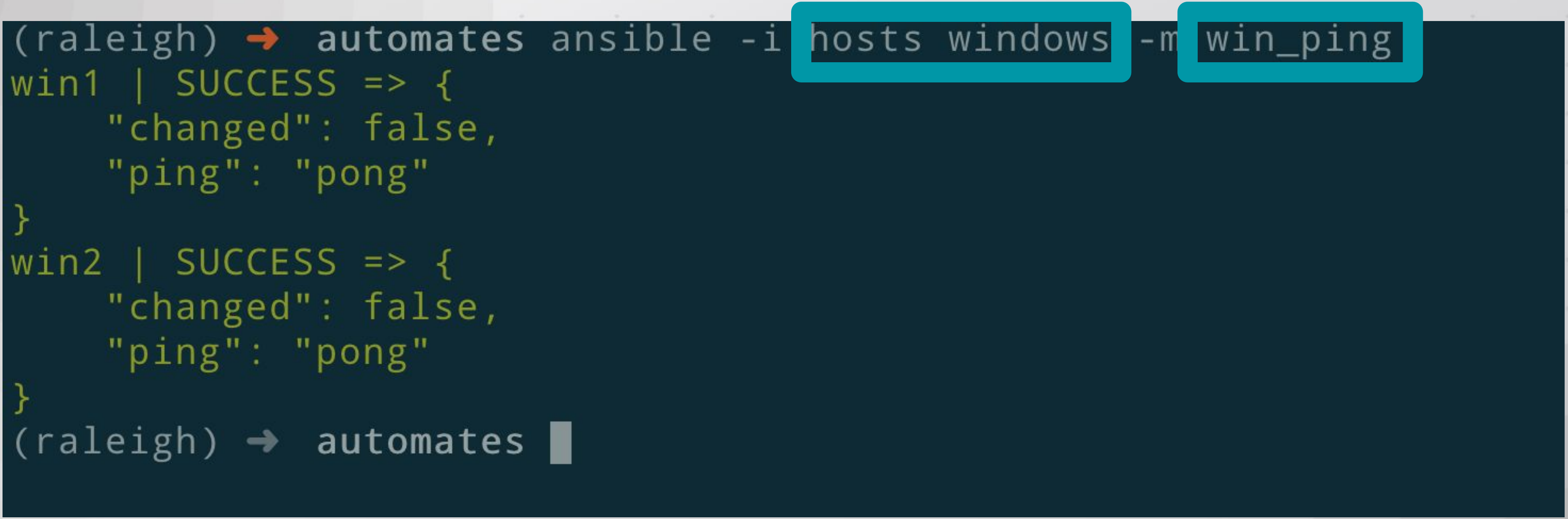
```
(raleigh) ➜  automates ansible -i hosts all -m ping
ansible | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: No python interpreters found for host win1 (tried ['/usr/bin/p
'python2.7', 'python2.6', '/usr/libexec/platform-python', '/usr/bin/pytho

win1 | FAILED! => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "module_stderr": "Exception calling \"Create\" with \"1\" argument(s
                  ~\r\nAn expression was expected after '('.\r\nAt line:1
                          ~\r\nMissing argument in parameter list.\r\nAt
     ~\r\nMissing '(' after 'if' in if statement.\r\nAt line:22 char:7\r
g '(' after 'if' in if statement.\r\nAt line:22 char:30\r\n+     if sys.\
\nMissing expression after '.'.\r\nAt line:22 char:35\r\n+     if sys.ve
```



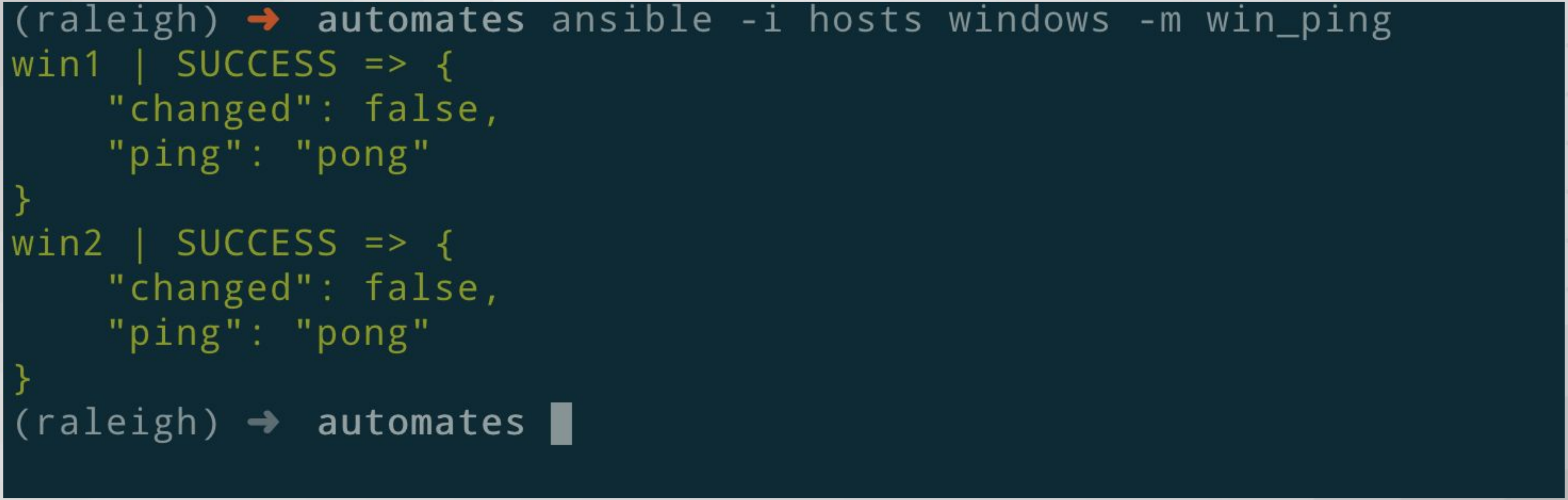
```
(raleigh) ➜  automates ansible -i hosts windows -m win_ping
win1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
win2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
(raleigh) ➜  automates 
```



```
(raleigh) ➜  automates ansible -i hosts windows -m win_ping
win1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
win2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
(raleigh) ➜  automates 
```



```
(raleigh) ➜  automates ansible -i hosts windows -m win_ping
win1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
win2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
(raleigh) ➜  automates
```



```
(raleigh) ➜  automates ansible -i hosts windows -m win_ping
win1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
win2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
(raleigh) ➜  automates 
```

Powershell

- Unlike Python, "just there" on modern Windows
- We can use .NET
- Powershell 3+, Windows 7/Server 2008+
  - Experimental PSCore/6/7 support
- Access to the DSC universe via `win_dsc`

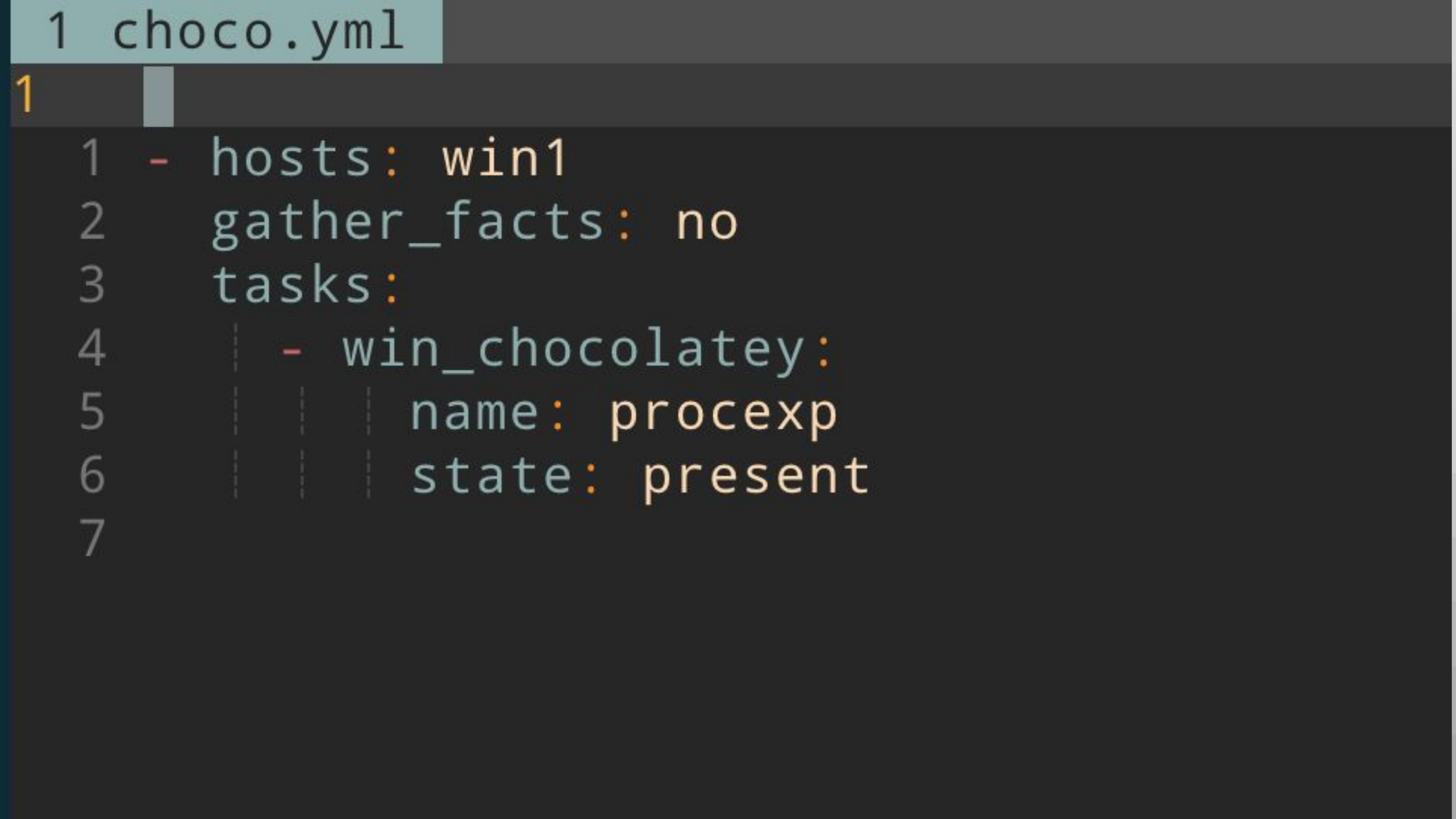
Powershell / DSC

```
- win_psmodule: # install xDNSServer DSC module on target
   name: xDnsServer
- win_dsc: # create DNS zone
   resource_name: xDnsServerPrimaryZone
   name: createdbyansible.com
- win_dsc: # create DNS record
   resource_name: xDnsRecord
   name: test
   zone: createdbyansible.com
   target: 1.2.3.4
   type: ARecord
```

App Install/Maintenance

- win_chocolatey !
- win_package if you must
- ~~`shell: c:\temp\setup.exe /quiet /dostuff`~~

A little bit closer now: win_chocolatey module



```yaml
- hosts: win1
  gather_facts: no
  tasks:
    - win_chocolatey:
        name: procexp
        state: present
```

Reboots, oh the reboots...

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

Windows Update

- Basic, synchronous updates
- Uses configured source (Windows Update/WSUS)
- Transparent SYSTEM + auto reboot

Windows Update

```
- win_updates:
  category_names: CriticalUpdates
  reboot: yes
  blacklist:
  - KB4056892
```

IIS

- Modules for managing websites, webapps, apppools, virtual dirs, etc.

IIS

```
- win_iis_website:
    name: Default Web Site
    physical_path: C:\Inetpub\WWWRoot

- win_iis_webapp:
    site: Default Web Site
    name: OrchardCMS
    physical_path: C:\Inetpub\WWWRoot\Orchard
```

Registry

- Manage individual key/value (win_regedit)
- Manage idempotent bulk import (win_regmerge)

Registry

```
- win_regedit:
   path: HKLM\Software\Microsoft\Windows
   name: SomeValueName
   value: 0x12345


- win_regmerge:
   path: ComplexRegData.reg
```

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

```
# ensure IIS is running
- win_service:
  name: W3Svc
  state: running

# ensure firewall service is stopped/disabled
- win_service:
  name: MpsSvc
  state: stopped
  start_mode: disabled
```

Domains

- Windows' way of doing enterprise identity
- Makes auth complex
- Ansible can do "throwaway" domains easily
- Promote/depromote DCs
- Joining/leaving domain is simple
- Manage basic domain objects

Domains

```
# create a domain
- win_domain:
   dns_domain_name: mydomain.local
   safe_mode_password: ItsASecret

# add a domain user
- win_domain_user:
   name: somebody
   upn: somebody@mydomain.local
   groups:
   - Domain Admins
```

ACLs

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

ACLs

```
O:BAG:S-1-5-21-328427983-2845905853-4261175022-
513D:AI(A;OICIID;FA;;;SY)(A;OICIID;FA;;;BA)(A;O
IC
ID;LC;;;BU)(A;CIID;DC;;;BU)(A;OICIIOID;GA;;;CO)
```

ACLs

```
- win_owner:
   path: C:\Program Files\SomeApp
   user: Administrator
   recurse: true

- win_acl:
   path: C:\Temp
   user: Users
   rights: ReadAndExecute,Write,Delete
   inherit: ContainerInherit,ObjectInherit
```

Wrapup

Questions?

#ANSIBLEFEST2019

THANK YOU

youtube.com/AnsibleAutomation

facebook.com/ansibleautomation

linkedin.com/company/Red-Hat

twitter.com/ansible