

Windows Configuration Management with Puppet

Why let the Linux admins have all the fun?



Gene Liverman

SRE @ Puppet

Formerly SysAdmin @ University of West Georgia

Windows, Linux, Mac, & Solaris

Storage, Firewalls, Load Balancers, etc.

Agenda

Better together

Chocolatey

Patching

Modules

Packer

Better Together

(DSC, Hyper-V, SCCM, WMI) + Puppet

DSC in PowerShell vs DSC via Puppet

Write less code to do the same thing

simple_dsc_example.ps1	simple_puppet_example.pp
<pre>1 Configuration Example 2 { 3 Import-DscResource xPSDesiredStateConfiguration 4 5 xArchive SampleCompressArchive 6 { 7 Path = "c:\foo\logs" 8 Destination = "c:\foo\logbackup" 9 CompressionLevel = "Optimal" 10 DestinationType = "File" 11 } 12 13 File ExampleTextFile 14 { 15 Ensure = 'Present' 16 Type = 'File' 17 DestinationPath = 'c:\foo.txt' 18 Contents = 'example text' 19 } 20 } 21</pre>	<pre>1 dsc_xarchive { 'SampleCompressArchive': 2 dsc_path => 'c:/foo/logs', 3 dsc_destination => 'c:/foo/logbackup', 4 dsc_compressionlevel => 'Optimal', 5 dsc_destinationtype => 'File', 6 } 7 8 dsc_file { 'ExampleTestFile': 9 dsc_ensure => 'Present', 10 dsc_type => 'File', 11 dsc_destinationpath => 'c:/foo.txt', 12 dsc_contents => 'example text', 13 } 14</pre>

Migrate resources to Puppet

Get the same result fewer lines of code

package-dsc-vs-puppet.pp

```
1  dsc_package{ 'python':
2    · dsc_ensure      => 'Present',
3    · dsc_name        => 'Python 2.7.10',
4    · dsc_productid   => 'E2B51919-207A-43EB-AE78-733F9C6797C2',
5    · dsc_path        => 'C:\\python.msi',
6  }
7
8  package { 'python':
9    · ensure => present,
10   · source => 'C:\\python.msi',
11 }
12
```

Puppet + DSC

DSC doesn't do everything Puppet does

- **No real reporting**
- **No rebooting as needed**
- **No package management**
- **Doesn't keep historical data on changes performed**

DSC does cover things Puppet doesn't

- **Hyper-V is one example**

You'll likely run into a situation where either Puppet or DSC doesn't cover the entirety of the problem all by itself. Instead of trying to make one solution fit all, you can mix and match them.

Paraphrasing the "Managing Windows with Puppet Enterprise" white paper

Layering Puppet Enterprise on top of DSC

Gain rich management capabilities including:

- **Node classification**
- **Status of nodes across your infrastructure**
- **Global access control**
 - Authentication can be via AD credentials
- **Tracking and reporting change**
 - This simplifies compliance auditing

Puppet, DSC, & Hyper-V

The xHyper-V module

- **xVHD** manages VHDs in a Hyper-V host.
- **xVhdFile** manages files or directories in a VHD.
- **xVMHost** manages Hyper-V host settings.
- **xVMNetworkAdapter** manages VMNetadapters attached to a Hyper-V virtual machine or the management OS.
- **xVMProcessor** manages Hyper-V virtual machine processor options.
- **xVMSwitch** manages virtual switches in a Hyper-V host.

Puppet + WMI

- **Custom facts**
 - Starting out with Puppet custom facts
 - <https://glennsarti.github.io/blog/puppet-ruby-facts/>
- **Integrate with existing SCCM**
 - Query client configuration such as location
 - Module: `thoward/sccm_application`
 - Installs or uninstalls packages advertised by SCCM

Chocolatey

aka apt-get / yum for Windows

Package management for Windows

- **What is Chocolatey?**
- **Public repo: chocolatey.org**
- **Private repo: Chocolatey Server**
 - Module: `chocolatey/chocolatey_server`

Common use cases for Chocolatey

- **FOSS Puppet agent**
- **Monitoring & Backup agents**
- **Log shipping**
- **PowerShell / WMF, .Net, NodeJS**
- **Browsers & plugins**
 - Chrome & Firefox
 - Adobe, Flash, Java
- **Utilities such as Git, 7-Zip, & Notepad++**

Patching

WSUS + strict change windows + Puppet

Patching scenario

- **Install & manage Chocolatey**
- **Install NuGet package provider for PowerShell**
- **Install PSWindowsUpdate PowerShell module**
- **Define a maintenance window**
- **Configure automated reboots during the maintenance window**
- **Install PowerShell and WMF**
- **Configure WSUS-related settings on the client**
- **Apply all non-interactive updates**



Demo Time: Code Sample

A Puppet manifest for patching Windows

Code sample (video)

win_patching.pp

```
1 class { 'chocolatey':  
2   · log_output => true,  
3 }  
4  
5 chocolateyfeature { 'autouninstaller':  
6   · ensure => enabled,  
7 }  
8  
9 exec { 'Install NuGet package provider':  
10   · command    => '$(Install-PackageProvider -Name NuGet -Force)',  
11   · onlyif     => '$(if((Get-PackageProvider -Name NuGet  
·     -ListAvailable -ErrorAction SilentlyContinue) -eq $null) { exit  
·     0 } else { exit 1 })',  
12   · provider   => powershell,  
13   · log_output => true
```

Modules

Common tasks made easier and repeatable

Modules pulled in by puppetlabs/windows

Vox Pupuli, voxpupuli.org

- puppet/download_file
- puppet/windows_env
- puppet/windowsfeature

Puppet, puppet.com

- puppetlabs/acl
- puppetlabs/chocolatey
- puppetlabs/dsc

- puppetlabs/iis
- puppetlabs/powershell
- puppetlabs/reboot
- puppetlabs/registry
- puppetlabs/stdlib
- puppetlabs/wsus_client

Modules for common Windows tasks

A sample of what's on the Puppet Forge

Backups:

- thespain/veeamagent (WIP)

FOSS Puppet:

- theforeman/puppet

Logs:

- genebean/nxlog
- puppet/windows_eventlog

Monitoring:

- genebean/zabbixagent

- icinga/icinga2
- puppet/nsclient
- sensu/sensu

Others:

- puppet/sslcertificate
- puppet/windows_firewall
- puppetlabs/azure
- puppetlabs/aws

Packer

Templated virtual machines

Packer resources

Packer: <https://www.packer.io>

Blogs:

- <http://www.hurryupandwait.io/blog/creating-windows-base-images-for-virtualbox-and-hyper-v-using-packer-boxstarter-and-vagrant>

Templates:

- <https://github.com/puppetlabs/puppetlabs-packer>
- <https://github.com/mwrock/packer-templates>

References and resources

- **Blog posts on Windows from Puppet:**
 - <https://puppet.com/blog-tags/powershell-dsc>
 - <https://puppet.com/blog-tags/windows>
- **Blog post on custom facts:** <https://glennsarti.github.io/blog/puppet-ruby-facts/>
- **Chocolatey:** <https://chocolatey.org>
- **PowerShell xHyper-V:** <https://github.com/PowerShell/xHyper-V>
- **Puppet Forge:** <https://forge.puppet.com>
- **White paper:**
 - <https://puppet.com/resources/whitepaper/managing-windows-with-puppet-enterprise>

What we talked about

DSC, WMI, SCCM

Package management with Chocolatey

Patching and rebooting during change windows

Managing common tasks with Puppet modules

Building images with Packer

Go have some fun automating your Windows configurations

Let Puppet do the repetitive things while you do play with new stuff

Gene Liverman

gene.liverman@puppet.com

genebean.github.io/talks

github.com/genebean

[@technicalissues](#)