# Server Build Out Procedures V2 – Highly scripted

## Create New Server

Edit variables in d:\chef\CreateNewServer.ps1

Open ChefDK PS prompt (as administrator)

Set-Location d:\chef

.\CreateNewServer.ps1

<#

CreateNewServer.ps1

Generates a test kitchen instance framework of files and configures it through updates to the files

It appears that I run into problems if I try to run kitchen converge at the end of this script. Will run it manually for now.

#>

# Setup Server Instance Values

# Edit these lines before each run

$NewComputerName = 'Server16'

$NewComputerIP = '192.168.0.116'

$NewComputerDNS = '192.168.0.110,192.168.0.101'

# Setup paths and starting location

# The source files have been customized to the point simple subsitutions is all that is required

$ChefKitchenRoot = 'D:\Chef'

$SourceFiles = 'D:\chef\SourceFiles'

set-location $ChefKitchenRoot

# Use Chef Generate to create the framework with locations for both templates and files

chef generate cookbook $NewComputerName

chef generate template $NewComputerName server-info.txt

###ToDo###

# This line generates an unwanted file

chef generate file $NewComputerName Scripts

# Copy Templates

Copy-Item -Path "$SourceFiles\templates\server-info.txt.erb" -Destination "$ChefKitchenRoot\$NewComputerName\templates"

# Edit and Copy Scripts

# vmscript1.ps1

# Substitute the new computer name

Copy-Item -Path "$SourceFiles\files\vmscript1.ps1" -Destination "$ChefKitchenRoot\$NewComputerName\files\default"

(Get-Content "$ChefKitchenRoot\$NewComputerName\files\default\vmscript1.ps1").replace('[NewServer]', $NewComputerName) |

Set-Content "$ChefKitchenRoot\$NewComputerName\files\default\vmscript1.ps1"

# vmscript2.ps1

# Substitute the new IP address and DNS Search Order

Copy-Item -Path "$SourceFiles\files\vmscript2.ps1" -Destination "$ChefKitchenRoot\$NewComputerName\files\default"

(Get-Content "$ChefKitchenRoot\$NewComputerName\files\default\vmscript2.ps1").replace('[IPAddress]', $NewComputerIP) |

Set-Content "$ChefKitchenRoot\$NewComputerName\files\default\vmscript2.ps1"

(Get-Content "$ChefKitchenRoot\$NewComputerName\files\default\vmscript2.ps1").replace('[DNSSearchOrder]', $NewComputerDNS) |

Set-Content "$ChefKitchenRoot\$NewComputerName\files\default\vmscript2.ps1"

# MyVMCommands.psm1

# A PowerShell module with custom tools

Copy-Item -Path "$SourceFiles\files\myvmcommands.psm1" -Destination "$ChefKitchenRoot\$NewComputerName\files\default"

# HostScript1.ps1

# Substitute the new VM name

Copy-Item -Path "$SourceFiles\files\HostScript1.ps1" -Destination "$ChefKitchenRoot\$NewComputerName"

(Get-Content "$ChefKitchenRoot\$NewComputerName\HostScript1.ps1").Replace('[NewVMName]', "$NewComputerName`\_2012R2") |

Set-Content "$ChefKitchenRoot\$NewComputerName\HostScript1.ps1"

# HostScript2.ps1

# Substitute the new VM name

Copy-Item -Path "$SourceFiles\files\HostScript2.ps1" -Destination "$ChefKitchenRoot\$NewComputerName"

(Get-Content "$ChefKitchenRoot\$NewComputerName\HostScript2.ps1").Replace('[NewVMName]', "$NewComputerName`\_2012R2") |

Set-Content "$ChefKitchenRoot\$NewComputerName\HostScript2.ps1"

# Kitchen Files (Kitchen.yml)

# Substitute the new computer name

Copy-Item -Path "$SourceFiles\KitchenRecipe\.kitchen.yml" -Destination "$ChefKitchenRoot\$NewComputerName"

(Get-Content "$ChefKitchenRoot\$NewComputerName\.kitchen.yml").Replace('[NewServer::default]', "[$NewComputerName::default]") |

Set-Content "$ChefKitchenRoot\$NewComputerName\.kitchen.yml"

# Default recipe

# Substitute the new computer name

Copy-Item -Path "$SourceFiles\KitchenRecipe\default.rb" -Destination "$ChefKitchenRoot\$NewComputerName\recipes"

(Get-Content "$ChefKitchenRoot\$NewComputerName\recipes\default.rb").Replace('[Server]', $NewComputerName) |

Set-Content "$ChefKitchenRoot\$NewComputerName\recipes\default.rb"

From (ChefDK) PS prompt set location d:\chef\serverXX

Kitchen Converge

---

driver:

  name: hyperv

  parent\_vhd\_folder: D:\HyperVResources\VMs\BaseBox

  parent\_vhd\_name: BaseBox.vhdx

  vm\_switch: ExternalSwitch

  memory\_startup\_bytes: 2GB

provisioner:

  name: chef\_zero

transport:

  password: H0rnyBunny

platforms:

  - name: windows-2012r2

suites:

  - name: default

    run\_list:

      - recipe[NewServer::default]

    attributes:

#

# Cookbook:: [Server]

# Recipe:: default

#

# Copyright:: 2017, The Authors, All Rights Reserved.

# Create new directories in the new machine

directory 'C:\temp'

directory 'C:\Scripts'

directory 'C:\Program Files\WindowsPowerShell\Modules\MyVmCommands'

# Copy the Template

template 'C:\temp\server-info.txt' do

  source 'server-info.txt.erb'

end

# Copy the files

# Must be VERY careful that case matches file, script and recipe!!

cookbook\_file 'C:\Program Files\WindowsPowerShell\Modules\MyVmCommands\MyVmCommands.psm1' do

    source 'myvmcommands.psm1'

end

cookbook\_file 'C:\Scripts\VMScript1.ps1' do

    source 'vmscript1.ps1'

end

cookbook\_file 'C:\Scripts\VMScript2.ps1' do

    source 'vmscript2.ps1'

end

## Customize the VM and Server (scripted)

1. HostScript1.ps1
2. Vmscript1.ps1
3. HostScript2.ps1
4. Vmscript2.ps1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# HostScript1

$VMNname = '[NewVMName]' # Substitute here

# Rename the VM

Rename-VM -Name default-windows-2012r2 -NewName $VMNname

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# VMScript1

# Settings

$ServerName = '[NewServer]' # Substitute here

# Rename the computer

Rename-Computer -NewName $ServerName

# Rename the existing (External) NIC

Get-NetAdapter | Rename-NetAdapter -NewName ExternalNIC

# Shutdown VM so a Host script can add the next NIC

Stop-Computer

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# HostScript2

$VMNname = '[NewVMName]' # Substitute here

# Add VM Network Adapter

Add-VMNetworkAdapter -VMName $VMNname -SwitchName InternalSwitch

# Start VM

Get-VM -Name $VMNname | Start-VM

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# VM Script 2

# Settings

$IP = '[IPAddress]' # Substitute here

$DNS = '[DNSSearchOrder]' # Substitute here

# Rename new Internal NIC.

Get-NetAdapter | Where-Object Name -ne 'ExternalNIC' | Rename-NetAdapter -NewName InternalNIC

# Set InternalNIC static IP address

Get-NetAdapter | Where-Object Name -eq 'InternalNIC' | New-NetIPAddress -PrefixLength 24 -IPAddress $IP

# Set DNS on InternalNIC

Set-InternalDNS $DNS

# Disable External NIC

Set-MyVmNetwork -Toggle InternalOnly

# Enable Remote Desktop

Enable-RemoteDesktop

# Set the firewall for PS Remoting

Get-NetFirewallPortFilter | ?{$\_.LocalPort -eq 5985 } | Get-NetFirewallRule | ?{ $\_.Direction –eq "Inbound" -and $\_.Profile -eq "Public" -and $\_.Action –eq "Allow"} | Set-NetFirewallRule -RemoteAddress "Any"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Bootstrap new computer to Chef

Run from ChefDK server, Server9

knife bootstrap windows winrm 192.168.0.1XX --winrm-user coatelab\administrator --winrm-password 'xxxxxxxxx' --node-name serverX.coatelab.com --run-list 'role[web]' --msi-url <http://server8.coatelab.com/chef-client-12.18.31-1-x64.msi>

Role: Web

chef-client::default (Sched Task go every 5 min)  
chef-client::delete\_validation

learn\_chef\_iis-0.4.1::default

Default Recipe

#

# Cookbook Name:: learn\_chef\_iis

# Recipe:: default

#

# Copyright (c) 2016 The Authors, All Rights Reserved.

# Add IIS Feature

powershell\_script 'Install IIS' do

  code 'Add-WindowsFeature Web-Server'

  guard\_interpreter :powershell\_script

  not\_if '(Get-WindowsFeature -Name Web-Server).Installed'

end

# Start the IIS Service

service 'w3svc' do

  action [:enable, :start]

end

# Set Permissions

directory 'c:\inetpub\wwwroot' do

  rights :read, 'IIS\_IUSRS'

  recursive true

  action :create

end

# Copy a sample template file

template 'c:\inetpub\wwwroot\Default.htm' do # ~FC033

  source 'Default.htm.erb'

end

# Copy a sample file

cookbook\_file 'C:\Scripts\NewScript.ps1' do

    source 'NewScript.ps1'

end

Daves-Test-Cookbook -0.1.0

Default Recipe

#

# Cookbook:: Daves-Test-Cookbook

# Recipe:: default

#

# Copyright:: 2017, The Authors, All Rights Reserved.

#

# A registry hack to prevent Server Manager from starting at logon

registry\_key 'HKEY\_LOCAL\_MACHINE\Software\Microsoft\ServerManager' do

  values [{

    :name => 'DoNotOpenServerManagerAtLogon',

    :type => :dword,

    :data => 1

  }]

  action :create

end

ad-join

Library Cookbook

JoinCoatelabDomain

Default Recipe

#

# Cookbook Name:: JoinCoatelabDomain

# Recipe:: default

#

# Copyright 2017, YOUR\_COMPANY\_NAME

#

# All rights reserved - Do Not Redistribute

#

include\_recipe 'ad-join::default'

# domain\_join 'Server10' do

# domain\_join node['hostname'] do

#The parameter in the first line is ignored if the server is not to be renamed

domain\_join node['hostname'] do

  domain 'coatelab.com'

  domain\_user 'coateds'

  domain\_password 'H0rnyBunny'

  ou nil

  action :join

end

test\_powershell -0.1.1

Default Recipe

#

# Cookbook:: test\_powershell

# Recipe:: default

#

# Copyright:: 2017, The Authors, All Rights Reserved.

# This is the Cheffy way to make it indempotent

powershell\_script 'make\_test\_file' do

    guard\_interpreter :powershell\_script

    code 'New-Item -Type File -Path c:/scripts/cheftestfile.txt'

    not\_if 'test-path c:/scripts/cheftestfile.txt'

end

# This is the PowerShell way to make it indempotent

# powershell\_script 'make\_test\_file' do

#     code <<-EOH

#     $File = 'c:/scripts/cheftestfile.txt'

#     If (!(Test-Path $File)) {New-Item -Type File -Path $File}

#     EOH

# end

# This works

powershell\_script 'set\_dns' do

    code <<-EOH

    Set-InternalDNS '192.168.0.110,192.168.0.101'

    EOH

end

powershell\_script 'check\_wins' do

    guard\_interpreter :powershell\_script

    code 'Set-Content -Path c:/scripts/cheftestfile.txt -Value 1'

    only\_if 'get-service WINS'

end