# Build out Server9 in the lab – High Detail description

## Use Chef to generate a cookbook

From d:\Chef using ChefDK, run the following commands. (Substitute ‘Server9’ for [ServerName])

* chef generate cookbook [ServerName]
* chef generate template [ServerName] server-info.txt
* chef generate file [ServerName] Scripts

## Customize the cookbook

Template

Erb contents:

fqdn: <%= node['fqdn'] %>  
hostname: <%= node['hostname'] %>  
platform: <%= node['platform'] %> - <%= node['platform\_version'] %>  
cpu count: <%= node['cpu']['total'] %>

Having trouble saving the file with new contents… Must edit as administrator

This is a good example of using Attributes when built, this file is in the C:\Temp folder of the VM and will read something like this:

fqdn: Server8.sea.corp.expecn.com  
hostname: Server8  
platform: windows - 6.3.9600  
cpu count: 2

File

Copy PowerShell scripts to D:\chef\[ServerName]\files\default.

* VMScript1.ps1
* VMScript2.ps1
* MyVMCommands.psm1

These files are located in this repository. The first two are called from the new VM to customize it during the install/config process. The third file is a PS Module and contains functions to be called during normal operations of the VM in the lab.

Kitchen.yml

---

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[[ServerName]::default]

attributes:

Recipes Default.rb

#

# Cookbook:: Server9

# Recipe:: default

#

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

# Create new directoies 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

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

## Converge to build the VM

Copy HostScript1 and 2 to root of new Server D:\Chef\[Servername]

kitchen converge

## Post Converge Scripts/Configuration/Chef-Client

Host Script 1

* Must be edited
* Rename the VM

VM Script 1

* Must be edited
* Rename the Computer
* Rename the included NIC (ExternalNIC)
* Stop the VM OS

Host Script 2

* Must be edited
* Adds a second NIC (InternalSwitch)
* Starts the VM

VM Script 2

* Must be editied
* Rename Internal NIC, Statically address it
* Configure to run internal NiC only using PS function in module that should be there
* Configure RDP
* Join the Domain

New process…

* 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 ChefDK client  
  knife bootstrap windows winrm 192.168.0.107 --winrm-user coatelab\administrator --winrm-password 'xxxxxxxxx' --node-name server7.coatelab.com --run-list 'role[web]' --msi-url <http://server2.coatelab.com/chef-client-12.18.31-1-x64.msi>

Automation Goals:

* Parameterize Scripts so that can be called with ComputerName, IP Address etc
* Move PS Remoting Firewall configuration into VMScript2
* Move Join Domain procedure into a Cookbook?

## Chef Bootstrap, Role and Cookbooks configuration

Role: Web

* chef-client::default  
  Modify this cookbook to make Sched Task go every 5 min
* chef-client::delete\_validation
* learn\_chef\_iis::default