**Workstation Setup**

<https://docs.chef.io/install_dk.html>

1. Install ChefDK (Workstation)
2. chef verify – confirms chef dk install is correct (Workstation)
3. Install knife-windows ruby gem, chef gem install knife-windows -V
4. Install chef server (use script)
5. access Chef by name or IP address from browser, log in (see script for creds)
6. Select Administration tab, select organization, select Starter Kit
7. Download starter kit
8. Extract to c:\Users\<username>\
9. Setup Chef repo – use starter kit, extract
10. cd to c:\users\<username>\chef-repo
11. knife ssl fetch (to copy down certificate from chef server
12. knife ssl check – verify cert
13. knife client list – verify everthing is ok.
14. move chefdk references to beginning of windows path variable
15. Need to confirm firewall is disabled on node
16. run windows bootstrap command

knife bootstrap windows winrm 62.193.13.72 -x '.\Administrator' -P Interactive2014 -N "dr-2015-01"

These steps I tried when using jump server:

1. Need to set HTTP\_PROXY variable to admin hub proxy, [**http://85.15.29.86:8080**](http://85.15.29.86:8080/)
2. update chef-repo\.chef\knife.rb (had to change to IP address in adminhub)
3. Need to exit out of powershell to activate proxy settings from above

Class Notes:

Note all commands are not idempotent, you may need to code that in.

Order of parameters in resource does not matter

research notification handlers pdf page 252

recommendation – write recipe to delete validation.pem

file, cookbookfile, - create a cross plateform delete cookbook

can use shorthand for default action from resource directory default is create directory

search – node and databags are the useful searches

chef create cookbook

knife cookbook create ip-logger

chef upload cookbook

knife cookbook upload ip-logger

chef update run\_list

knife node run\_list add windowsnode 'recipe[ip-logger]', this will run the default recipe to run another recipe[cookbook:recipe\_name]

instructor email: [elon@chef.io](mailto:elon@chef.io)

Path variables on my workstation:

changed:

C:\opscode\chef\bin;C:\opscode\chef\embedded\bin;C:\opscode\chefdk\bin;C:\Python27\;C:\ProgramData\Oracle\Java\javapath;C:\Program Files (x86)\RSA SecurID Token Common;C:\Program Files (x86)\AMD APP\bin\x86\_64;C:\Program Files (x86)\AMD APP\bin\x86;C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCLS Client\;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Program Files\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files\Perforce;C:\Program Files (x86)\Perforce;C:\Program Files\Interactive Intelligence\;C:\Program Files (x86)\Interactive Intelligence\;C:\Program Files (x86)\Interactive Intelligence\ININ Trace Initialization\;C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\DTS\Binn\;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Python27\Scripts;C:\Program Files\OpenVPN\bin;C:\Users\Toby.Matherly\AppData\Local\GitHub\;C:\ProgramData\chocolatey\bin

current:

C:\Python27\;C:\ProgramData\Oracle\Java\javapath;C:\Program Files (x86)\RSA SecurID Token Common;C:\Program Files (x86)\AMD APP\bin\x86\_64;C:\Program Files (x86)\AMD APP\bin\x86;C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCLS Client\;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Program Files\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files\Perforce;C:\Program Files (x86)\Perforce;C:\Program Files\Interactive Intelligence\;C:\Program Files (x86)\Interactive Intelligence\;C:\Program Files (x86)\Interactive Intelligence\ININ Trace Initialization\;C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\DTS\Binn\;C:\opscode\chef\bin;C:\opscode\chef\embedded\bin;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Python27\Scripts;C:\Program Files\OpenVPN\bin;C:\Users\Toby.Matherly\AppData\Local\GitHub\;C:\ProgramData\chocolatey\bin;C:\opscode\chefdk\bin

chef server commands:

create new user

sudo chef-server-ctl user-create jojo Jojo Baldeo [jojo.baldeo@inin.com](mailto:jojo.baldeo@inin.com) pw

sudo chef-server-ctl user-create julius Julius DelaCruz [juluis.delacruz@inin.com](mailto:juluis.delacruz@inin.com) pw

sudo chef-server-ctl user-create loven Loven Torella [loven.torella@inin.com](mailto:loven.torella@inin.com) pw

add user to org

sudo chef-server-ctl org-user-add caas jojo

sudo chef-server-ctl org-user-add caas julius

sudo chef-server-ctl org-user-add caas loven

**Run chef client remotely from windows**

Copy a file to multiple servers:

PS C:\chef> Get-ADComputer -Filter 'Name -like "NPilot-\*"' | ForEach-Object { $test=$\_.Name; Copy-Item .\validation.pem -Destination "\\$test\c$\chef" }

run first-boot script:

Get-ADComputer -Filter 'Name -like "NPilot-\*"' | ForEach-Object { invoke-command -ComputerName $\_.Name -ScriptBlock {chef-client -j "c:\chef\first-boot.json" } }

dir command:

PS C:\chef> Get-ADComputer -Filter 'Name -like "NPilot-\*"' | ForEach-Object { invoke-command -ComputerName $\_.Name -ScriptBlock {dir c:\chef\validation.pem } }

**Knife backup**

knife backup export -c ~/chef-repo/.chef/knife.rb -D ~/chef-backup/

create script: chef\_server\_permissions.rb

sudo chmod a+x chef\_server\_permissions.rb

knife backup restore -c ~/chef-repo/.chef/knife.rb -D ~/chef-backup/ -y

knife search node 'platfor?:ubuntu'

**compare backup to current environment:**

knife diff --chef-repo-path ~/chef-repo/chef-backup

**Commands that allow knife windows to work:**

# must be run from command line and not powershell

winrm set winrm/config/service/auth @{Basic="true"}

winrm set winrm/config/service @{AllowUnencrypted="true"}

winrm set winrm/config/client/auth @{Basic="true"}

winrm set winrm/config/client @{AllowUnencrypted="true"}