|  |  |
| --- | --- |
|  | Chef Doc |

|  |  |  |
| --- | --- | --- |
|  | Document Reference: | None |
|  | Current Version: | 1.0 |
|  | Version Date: | 24/10/2018 |
|  | Status | Final |
|  | Author: | Sebastien Chartrand |
|  |

Table of Contents

[1 Document Control 1](#_Toc529191876)

[1.1 Document Control/Change History 1](#_Toc529191877)

[2 Bootstrap / install client 1](#_Toc529191878)

[3 Chef Server user access 1](#_Toc529191879)

[3.1 List all users 1](#_Toc529191880)

[3.2 List organization 1](#_Toc529191881)

[3.3 Create user 1](#_Toc529191882)

[3.4 Starter kit 1](#_Toc529191883)

[4 Git 1](#_Toc529191884)

[5 Install Chef Development kit 1](#_Toc529191885)

[5.1 Generate cookbook template 1](#_Toc529191886)

[5.2 Create an attribute (Variable) 1](#_Toc529191887)

[5.3 Create a template 1](#_Toc529191888)

[5.4 Generate recipe 1](#_Toc529191889)

[6 Kitchen 1](#_Toc529191890)

[6.1 Test with inspec 1](#_Toc529191891)

[7 Knife and berks 1](#_Toc529191892)

[8 Scan and chef automate server 1](#_Toc529191893)

[9 Role and custom attributes 1](#_Toc529191894)

[10 Best practices 1](#_Toc529191895)

# Document Control

## Document Control/Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Comment** | **Editor** |
| 0.1 | 2018-06-08 | Draft | Sebastien Chartrand |
| 1.0 | 2018-08-02 | Final |  |

# Bootstrap / install client

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

# note sudo will want to run with a password, sudoers nopass permission, does not seem to work.

knife bootstrap 10.10.20.55 -x UserName -P 'YourPassword' --sudo -N rtx0l06

# Chef Server user access

## List all users

chef-server-ctl user-list

## List organization

chef-server-ctl org-list

## Create user

chef-server-ctl user-create UserName Sebastien Chartrand sebastien.chatrand@fujitsu.com 'Mypass'

Attach to org has an admin

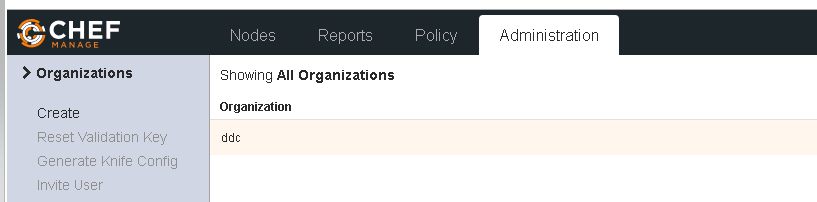
chef-server-ctl org-user-add rtx UserName –admin

## Starter kit

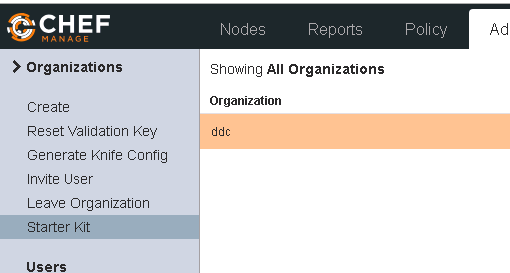
The Starter is a preconfigure tar balled directory that has your chef config and preconfigured. You get fetch it at this location

Ex : <https://1.1.1.1/organizations/ddc/getting_started> = <https://chefserver/organizations/orgname/getting_started>

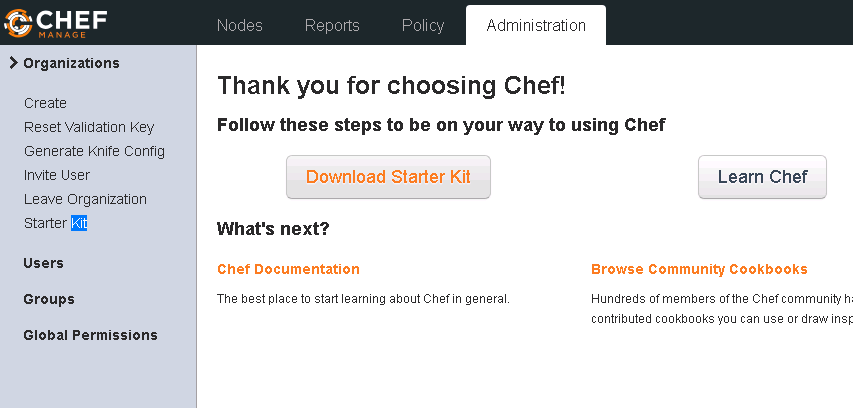
* Click on the Administration tab, Then your organization



Choose Starter kit



* Then Download Starter kit



Extract the downloaded file with the utility of your likening. It will extract a chef-repo directory.

[chartrse@rtxchwk01 ~]$ cd chef-repo/

The hidden. chef directory is where your chef server key and knife config resides.

[chartrse@rtxchwk01 chef-repo]$ cat .chef/

chartrse.pem knife.rb syntaxcache/ trusted\_certs/

[chartrse@rtxchwk01 chef-repo]$ cat .chef/knife.rb

# See https://docs.getchef.com/config\_rb\_knife.html for more information on knife configuration options

current\_dir = File.dirname(\_\_FILE\_\_)

log\_level :info

log\_location STDOUT

node\_name "chartrse"

client\_key "#{current\_dir}/chartrse.pem"

chef\_server\_url "https://rtxlchp01.rtxlab.local/organizations/rtx"

cookbook\_path ["#{current\_dir}/../cookbooks"]

[chartrse@rtxchwk01 chef-repo]$ ls

cookbooks README.md roles

[chartrse@rtxchwk01 chef-repo]$

# Git

Add all files to the commit

git add -A

Commit and comment

git commit -s -m "First commit"

Add repository has origin in this example we used GitHub

git remote add origin <https://github.com/aink99/chef_doc.git>

Push your change

git push -u origin master

Add only one file commit and push

touch foobar

git add foobar

git commit -m "test commit foobar"

git push

# Install Chef Development kit

sudo rpm -Uvh <https://packages.chef.io/files/stable/chefdk/3.3.23/el/7/chefdk-3.3.23-1.el7.x86_64.rpm>

**Note there’s no best practice you chose multiple cookbook with one recipe or a cookbook with multiple recipe.**

**Ex you could create a cookbook per tower ex fai\_linux\_os and put multiple Linux recipe for some linux os and another one more specific fai\_linux\_apache**

## Generate cookbook template

Configure your get settings (Only if you have not done it before)

git config --global user.email "sebastien.chartrand@fujitsu.com"

git config --global user.name "Sebastien Chartrand

Create a directory and generate your cookbook

mkdir chef

cd chef

chef generate cookbook fai\_linux\_baseline

## Create an attribute (Variable)

This test coobook is for managing the password length in /etc/login.defs

Within your newly created cookbook generate an attribute called password

chef generate attribute password

cd attributes

vi attributes/password.rb

add the following line and save

default['fai\_linux\_baseline']['password\_length'] = '14'

## Create a template

Generate a template and copy or redirect the content of /etc/login.defs

chef generate template . login.defs

cat /etc/login.defs >> templates/login.defs.erb

modify the PASS\_MIN\_LEN like this so that it uses the attribute

PASS\_MIN\_LEN **<%= node['fai\_linux\_baseline']['password\_length'] %>**

Any other change in that template file will be changed on the node.

## Generate recipe

chef generate recipe . password\_policy

cd recipes/

edit password\_policy.rb

vim password\_policy.rb

add the following to use your template

template '/etc/login.defs' do

source 'login.defs.erb'

owner 'root'

group 'root'

mode '0655'

end

Has best practice include your new recipes to the default one

Edit default.rb

vi recipes/default.rb

and add your recipe

include\_recipe 'fai\_linux\_baseline::password\_policy'

# Kitchen

Kitchen starts a small chef server (called chef zero )on from your chef development kit station and , can test the change. Driver can be hypervirtual-box(with the vagrant driver), docker etc.

In this example, we don’t have any cloud based VM a so will used the dokken driver (Docker image + systemd)

Install the docker and dokken kitchen driver

chef gem install kitchen-docker

chef gem install kitchen-dokken

Edit the yml file

vi .kitchen.yml

add the following

---

driver:

name: dokken

privileged: true # because Docker and SystemD/Upstart

provisioner:

name: dokken

transport:

name: dokken

verifier:

name: inspec

platforms:

#- name: ubuntu-18.04

# driver:

# image: dokken/ubuntu-18.04

# pid\_one\_command: /bin/systemd

# intermediate\_instructions:

# - RUN /usr/bin/apt-get update

- name: centos-7

driver:

image: dokken/centos-7

pid\_one\_command: /usr/lib/systemd/systemd

suites:

- name: default

run\_list:

- recipe[fai\_linux\_baseline::default]

#- recipe[test\_cookbook::default]

Create your docker dokken image

chartrse@rtxchwk01 fai\_linux\_baseline]$ kitchen create

-----> Starting Kitchen (v1.23.2)

-----> Creating <default-centos-7>...

Creating kitchen sandbox at /home/chartrse/.dokken/kitchen\_sandbox/8f42d15416-default-centos-7

Creating verifier sandbox at /home/chartrse/.dokken/verifier\_sandbox/8f42d15416-default-centos-7

Building work image..

Creating container 8f42d15416-default-centos-7

Finished creating <default-centos-7> (0m4.33s).

-----> Kitchen is finished. (0m8.02s)

Login to your image

[chartrse@rtxchwk01 fai\_linux\_baseline]$ kitchen login

[root@dokken /]# logout

Converge (apply) your change (Note if you have not created your kitchen, converge will create it for you )

[chartrse@rtxchwk01 fai\_linux\_baseline]$ kitchen converge

-----> Starting Kitchen (v1.23.2)

-----> Converging <default-centos-7>...

Creating kitchen sandbox in /home/chartrse/.dokken/kitchen\_sandbox/8f42d15416-default-centos-7

Preparing dna.json

Resolving cookbook dependencies with Berkshelf 7.0.6...

Removing non-cookbook files before transfer

Preparing validation.pem

Preparing client.rb

Starting Chef Client, version 14.5.33

Creating a new client identity for default-centos-7 using the validator key.

resolving cookbooks for run list: ["fai\_linux\_baseline::default"]

Synchronizing Cookbooks:

- fai\_linux\_baseline (0.1.0)

Installing Cookbook Gems:

Compiling Cookbooks...

Converging 1 resources

Recipe: fai\_linux\_baseline::password\_policy

\* template[/etc/login.defs] action create (up to date)

Running handlers:

Running handlers complete

Chef Client finished, 0/1 resources updated in 02 seconds

Finished converging <default-centos-7> (0m8.70s).

-----> Kitchen is finished. (0m12.82s)

## Test with inspec

Edit your recipe test

vi test/integration/default/password\_policy\_test.rb

Add the following resource called file. Here are making sure that the line that starts with PASS\_MIN\_LEN then we use a world card and make sure line that end with 14

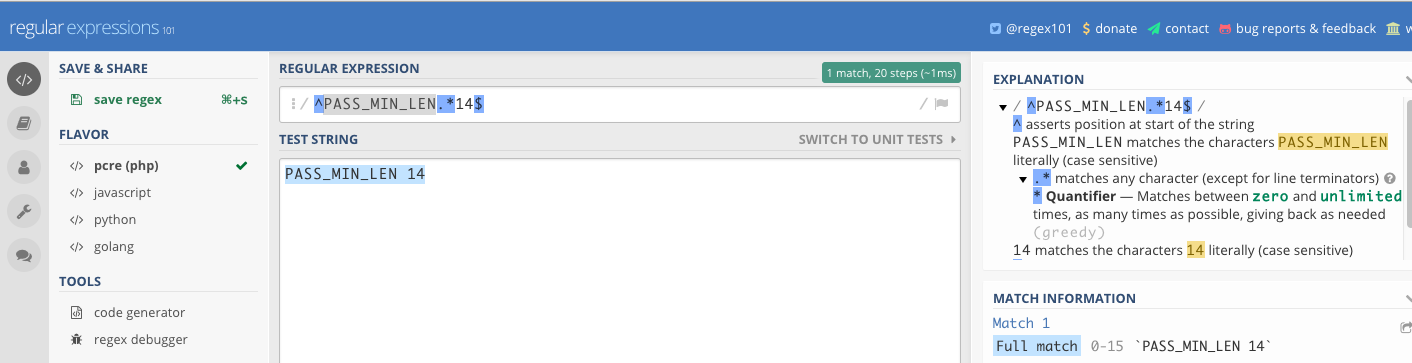
describe file('/etc/login.defs') do

its('content') { should match /^PASS\_MIN\_LEN.\*14$/ }

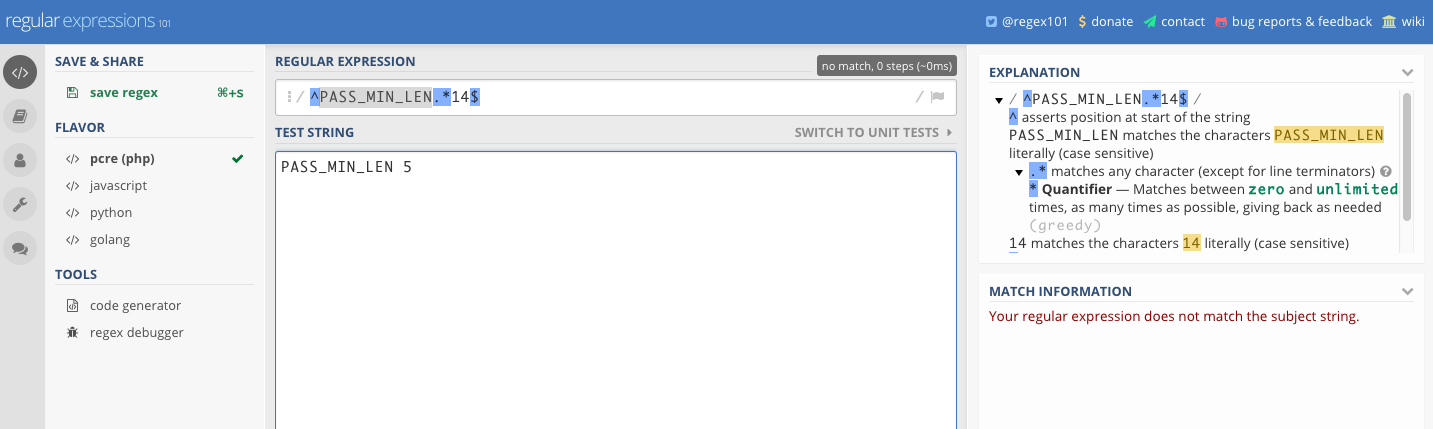
end

You can test your regex on on the following site : <https://regex101.com/>

Match:



No match:



You can now add this part to your kitchen yaml file:

suites:

- name: default

run\_list:

- recipe[fai\_linux\_baseline::default]

#- recipe[test\_cookbook::default]

verifier:

inspec\_tests:

- test/integration/default

attributes:

Kitchen test , builds converge, test then destroy kitchen ex:

chartrse@rtxchwk01 fai\_linux\_baseline]$ kitchen test

-----> Starting Kitchen (v1.23.2)

-----> Cleaning up any prior instances of <default-centos-7>

-----> Destroying <default-centos-7>...

Deleting kitchen sandbox at /home/chartrse/.dokken/kitchen\_sandbox/8f42d15416-default-centos-7

Deleting verifier sandbox at /home/chartrse/.dokken/verifier\_sandbox/8f42d15416-default-centos-7

Finished destroying <default-centos-7> (0m11.61s).

-----> Testing <default-centos-7>

-----> Creating <default-centos-7>...

Creating kitchen sandbox at /home/chartrse/.dokken/kitchen\_sandbox/8f42d15416-default-centos-7

Creating verifier sandbox at /home/chartrse/.dokken/verifier\_sandbox/8f42d15416-default-centos-7

Building work image..

Creating container 8f42d15416-default-centos-7

Finished creating <default-centos-7> (0m3.86s).

-----> Converging <default-centos-7>...

Creating kitchen sandbox in /home/chartrse/.dokken/kitchen\_sandbox/8f42d15416-default-centos-7

Preparing dna.json

Resolving cookbook dependencies with Berkshelf 7.0.6...

Removing non-cookbook files before transfer

Preparing validation.pem

Preparing client.rb

Starting Chef Client, version 14.5.33

Creating a new client identity for default-centos-7 using the validator key.

resolving cookbooks for run list: ["fai\_linux\_baseline::default"]

Synchronizing Cookbooks:

- fai\_linux\_baseline (0.1.0)

Installing Cookbook Gems:

Compiling Cookbooks...

Converging 1 resources

Recipe: fai\_linux\_baseline::password\_policy

\* template[/etc/login.defs] action create

- update content in file /etc/login.defs from 11fa3a to a3c20f

--- /etc/login.defs 2016-11-04 18:24:33.000000000 +0000

+++ /etc/.chef-login20181029-52-7f51mg.defs 2018-10-29 18:52:35.848648200 +0000

@@ -23,8 +23,8 @@

# PASS\_WARN\_AGE Number of days warning given before a password expires.

#

PASS\_MAX\_DAYS 99999

-PASS\_MIN\_DAYS 0

-PASS\_MIN\_LEN 5

+PASS\_MIN\_DAYS 5

+PASS\_MIN\_LEN 14

PASS\_WARN\_AGE 7

#

- change mode from '0644' to '0655'

Running handlers:

Running handlers complete

Chef Client finished, 1/1 resources updated in 02 seconds

Finished converging <default-centos-7> (0m13.54s).

-----> Setting up <default-centos-7>...

Finished setting up <default-centos-7> (0m0.00s).

-----> Verifying <default-centos-7>...

Loaded tests from {:path=>".data.home.chartrse.chef-repo.cookbooks.fai\_linux\_baseline.test.integration.default"}

Profile: tests from {:path=>"/data/home/chartrse/chef-repo/cookbooks/fai\_linux\_baseline/test/integration/default"} (tests from {:path=>".data.home.chartrse.chef-repo.cookbooks.fai\_linux\_baseline.test.integration.default"})

Version: (not specified)

Target: docker://f2391af73f083027c305ced06224641693e2babcb362e5297b1a5ec6aed7be0c

File /etc/login.defs

✔ content should match /^PASS\_MIN\_LEN.\*14/

Test Summary: 1 successful, 0 failures, 0 skipped

Finished verifying <default-centos-7> (0m3.70s).

-----> Destroying <default-centos-7>...

Deleting kitchen sandbox at /home/chartrse/.dokken/kitchen\_sandbox/8f42d15416-default-centos-7

Deleting verifier sandbox at /home/chartrse/.dokken/verifier\_sandbox/8f42d15416-default-centos-7

Finished destroying <default-centos-7> (0m10.93s).

Finished testing <default-centos-7> (0m43.73s).

-----> Kitchen is finished. (0m46.66s)

[chartrse@rtxchwk01 fai\_linux\_baseline]$ kitchen list

Instance Driver Provisioner Verifier Transport Last Action Last Error

default-centos-7 Dokken Dokken Inspec Dokken <Not Created> <None>

[chartrse@rtxchwk01 fai\_linux\_baseline]$

# Knife and berks

Knife ssh gets is information from chef server, when the secure bit it set to on the node, you need an extra attribute. This is in order to make it harder for the attacker in case you chef server get comprise. Ex: you need to specify -a ipaddress. ohai gathers the info IP , number of CPU and sends it to the chef server so it can be used .For example the platform

chartrse@rtxchwk01 fai\_linux\_patch]$ ohai |grep -i '"platform":'

[2018-10-26T15:59:12-05:00] INFO: The plugin path /etc/chef/ohai/plugins does not exist. Skipping...

"platform": "oracle",

To upload your cookbook to your chef server use:

[chartrse@rtxchwk01 fai\_linux\_baseline]$ berks upload

Uploaded fai\_linux\_baseline (0.2.0) to: 'https://rtxlchp01.rtxlab.local/organizations/rtx'

If you run the command again without changing your version metadata.rb , the server will notice and won’t upload your cookbook.

[chartrse@rtxchwk01 fai\_linux\_baseline]$ berks upload

Skipping fai\_linux\_baseline (0.1.0) (frozen)

[chartrse@rtxchwk01 fai\_linux\_baseline]$

This on reason to use berks upload over knife upload. Berks add an extra protection will knife upload could bypass it

To fetch any cookbook dependencies, use berks install:

[chartrse@rtxchwk01 chef-svrm1]$ berks install

Resolving cookbook dependencies...

Fetching 'chef-svrm1' from source at .

Fetching cookbook index from https://supermarket.chef.io...

Using chef-sugar (4.1.0)

Using firewall (2.6.5)

Installing hostsfile (3.0.1)

Using chef-svrm1 (0.1.0) from source at .

[chartrse@rtxchwk01 chef-svrm1]$

This will fetch all dependencies that are in the metadata.rb

[chartrse@rtxchwk01 chef-svrm1]$ grep depends metadata.rb

depends 'hostsfile'

depends 'firewall'

This is how to attaches a recipe to the node (runlist) . It will be executed the next time the node performs the command chef-client

knife node run\_list add rtxlks01 'recipe[fai\_linux\_baseline]'

To force a node to check and execute it’s run list perform the following command:

knife ssh 'name:rtx0l0\*' -x UserName -P 'Password' 'sudo chef-client' -a ipaddress

You can place regex or wildcards in 'name:nodname'.

Here are some commands to performs against the node

* list the nodes

chartrse@rtxchwk01 chef-svrm1]$ knife node list

RTXLTST01.RTXLAB.Local

RTXLTST03.RTXLAB.Local

RTXLTST04.RTXLAB.Local

RTXLTST05.RTXLAB.Local

RTXLTST06.RTXLAB.Local

RTXLTST07.RTXLAB.Local

RTXLTST10.rtxlab.local

RTXLTST11.rtxlab.local

TestNode

rtx0l06

rtxchwk01.rtxlab.local

rtxlchr04.rtxlab.local

rtxlchs01.rtxlab.local

rtxlchw01.rtxlab.local

rtxles01.rtxlab.local

rtxles02.rtxlab.local

rtxles03.rtxlab.local

rtxlks01

rtxltst12.rtxlab.local

* Show node information

[chartrse@rtxchwk01 chef-svrm1]$ knife node show rtx0l06

Node Name: rtx0l06

Environment: \_default

FQDN:

IP:

Run List: recipe[cis-rhel], recipe[fai\_linux\_olscan], recipe[fai\_linux\_patch]

Roles:

Recipes: cis-rhel, cis-rhel::default, fai\_linux\_olscan, fai\_linux\_olscan::default, fai\_linux\_patch, fai\_linux\_patch::default, cis-rhel::aide, aide::default, cis-rhel::at, cis-rhel::auditd, auditd::rules, auditd::default, cis-rhel::core\_dumps, os-hardening::limits, cis-rhel::cron, cron::default, cis-rhel::firewalld, firewall::default, cis-rhel::grub, cis-rhel::kernel\_modules, cis-rhel::login\_banners, cis-rhel::login\_defs, os-hardening::login\_defs, cis-rhel::network\_packet\_remediation, sysctl::default, sysctl::service, cis-rhel::ntp, ntp::default, cis-rhel::pam, os-hardening::pam, cis-rhel::partitions, cis-rhel::rsyslog, rsyslog::client, rsyslog::default, cis-rhel::packages\_services, cis-rhel::ssh, ssh-hardening::default, ssh-hardening::server, ssh-hardening::client, cis-rhel::sysctl, os-hardening::sysctl, sysctl::apply, cis-rhel::syslog-ng, cis-rhel::useradd, cis-rhel::minimize\_access, os-hardening::minimize\_access, os-hardening::suid\_sgid, audit::default, audit::inspec, fai\_linux\_patch::rh\_yumall

Platform: oracle 6.10

Tags:

* Delete a node

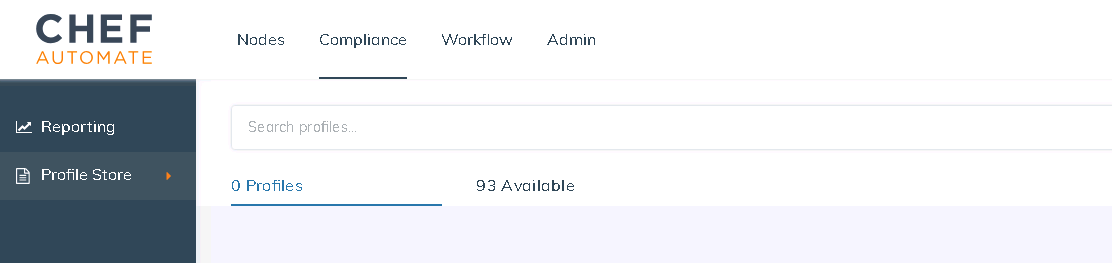
[chartrse@rtxchwk01 chef-svrm1]$ knife node delete TestNode

Do you really want to delete TestNode? (Y/N) N

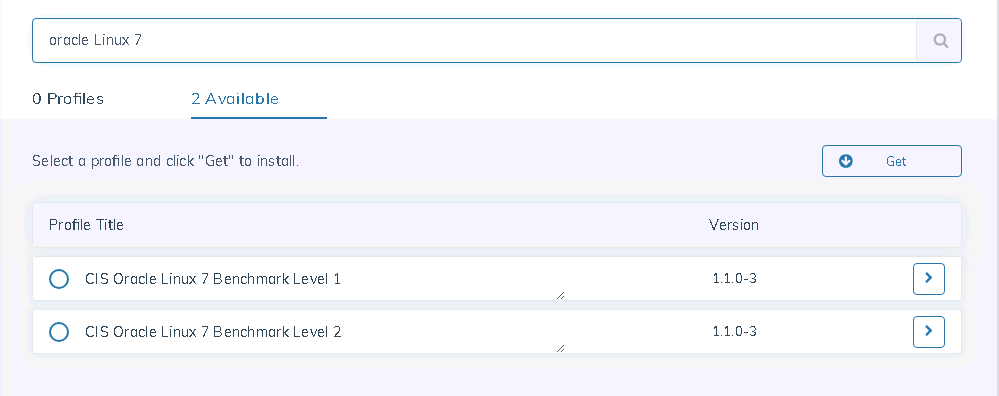
You said no, so I'm done here.

# Scan and chef automate server

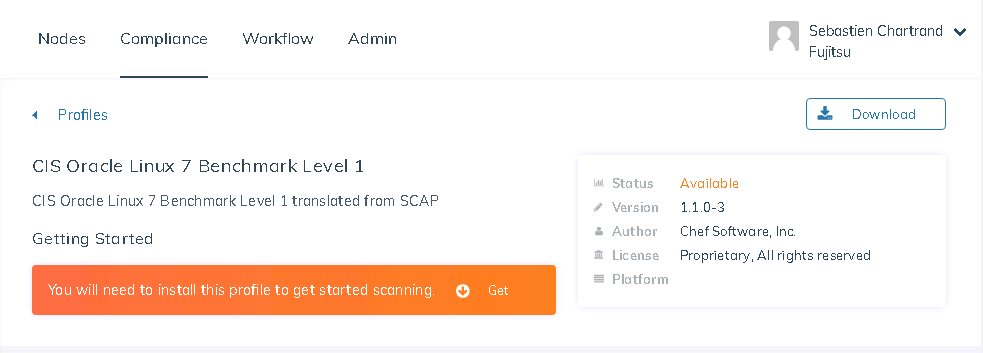
Go to <https://198.19.50.76/viz/#/welcome>, compliance, profiles and then profile store.



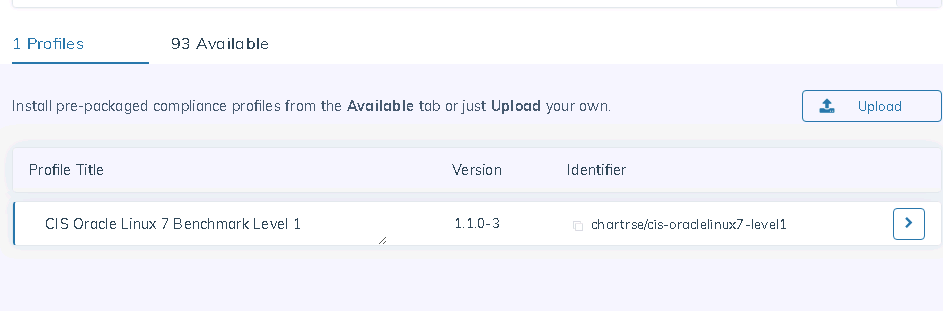
Then chose Available search something like oracle Linux 7



Select cis Oracle Linux Benchmark level 7 and click on the orange get button.



Now if you head back to Compliance, it will be shown under Profiles.



Do a chef generate cookbook

chef generate cookbook fai\_linux\_olscan

cd fai\_linux\_olscan

Edit the default recipe

vi recipes/default.rb

and add include the default audit cookbook

include\_recipe 'audit::default'

Edit your metadata.rb

Vi metadata.rb

And add this dependency

depends 'audit'

Generate a new attribute

Chef generate attribute auditprofile

and add the following to attributes/auditprofile.rb

default['audit']['reporter'] = 'chef-server-automate'

default['audit']['profiles'].push(

# Profile from Chef Compliance

{

'name': 'FAI\_Orl\_Lnx\_6\_CIS\_L1\_test',

'compliance': 'chartrse/cis-ol6-level1-server'

},

)

# Role and custom attributes

knife role edit linux-chef-client

{

"name": "linux-chef-client",

"description": "",

"json\_class": "Chef::Role",

"default\_attributes": {

"chef\_client": {

"interval": 300,

"splay": 60

}

},

"override\_attributes": {

},

"chef\_type": "role",

"run\_list": [

"recipe[chef-client::default]"

],

"env\_run\_lists": {

}

}

[chartrse@rtxchwk01 chef-repo]$ knife role show linux-chef-client

chef\_type: role

default\_attributes:

chef\_client:

interval: 300

splay: 60

description:

env\_run\_lists:

json\_class: Chef::Role

name: linux-chef-client

override\_attributes:

run\_list: recipe[chef-client::default]

[chartrse@rtxchwk01 chef-repo]$

[chartrse@rtxchwk01 chef-repo]$ knife node run\_list add rtxol07 "role[linux-chef-client]"

[chartrse@rtxchwk01 chef-repo]$ knife node show rtxol07

Node Name: rtxol07

Environment: \_default

FQDN: rtxol07.rtxlab.local

IP: 10.10.20.56

Run List: role[linux-chef-client]

Roles: linux-chef-client

Recipes: chef-client, chef-client::default, chef-client::service, chef-client::systemd\_service

Platform: oracle 7.4

Tags:

# Best practices

Best Practice Guidelines - Chef

Thursday, February 22, 2018

1:00 PM

**Nodes**

All nodes to be bootstrapped in lowercase for the FQDN of the node in all environment

It is best practice and also Automate and Chef Server search is case sensitive and will make it easier to look for a node in any environment.

Windows Bootstrap process aligns this requirement for future chef-client deployments for all three chef servers in (DDC, SDC, TOR)

**Cookbook Naming Convention**

All cookbook names will also be lowercase following below structure:

Structure:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Header |  |  |  |  |  |  |  | Variable |  |  |  |  |  |  |
| g | g | g | \_ | d | d | d | d | d | d | d | d | d | d | d |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

***Header portion***

ggg -- Usage determinant (fai for generic cookbook used across all environments, cmn/sdc/tor for specific environments or domains)

***Variable portion***

dddddddd -- use anything descriptive to define purpose of the cookbook, name of the application or tool etc..(example for a SysEdge install agent cookbook suggested name would be fai\_install\_sysedge\_agent).  Separate each word with a underscore ( \_ )

Example for a generic cookbook used across all environments

* fai\_bootstrap
* fai\_wsus\_client

Examples for specific cookbooks - below are cookbooks for CMN domain nodes to install specific tools/agents and will only be used for nodes in cmn environment

* cmn\_install\_te\_agent
* cmn\_install\_av\_agent

Also if any cookbook can be used for cross-platform deployment its best to have single cookbook catering for different platform than creating platform specific cookbooks - this way we can minimize the number of cookbooks to maintain and support if there are changes needed. (Example the Tripwire request)

**Cookbook Approvals**

Following the best practice any new cookbook that gets pushed to the workflow has to be approved by peers and not by the person who created the cookbook.  If the cookbook is already in use and assigned to nodes - if approved by peers will ensure changes are reviewed to avoid production outage or issues.  If this is practically not possible then get your lead/manager to approve the cookbook if they have access and if the changed code was already tested in lab.

Note:  Cookbooks are not available until they are delivered - so if the cookbook was approved make sure it passes all the phases and is also pushed through the Deliver to be utilized on the nodes.  There is no restriction on Deliver to be done by peer's, this can be initiated by the modifier of the cookbook.

**BEST PRACTICE:**

Make sure to name the cookbook as descriptive as possible to identify what its purpose is and what the intended usage would be.  Separate the sections in the cookbook name with a underscore ( \_ ) and always populate the README.md with description, usage and comment appropriately in your recipes the code and what each block of code does.

Created with Microsoft OneNote 2016.