Date:3/6/2017

Task:

step1:

Using Chef Install Apache

Step 2:

Enable Apache on boot

Step 3:

Deploy our code on Apache

Step 4:

Verify

**#Installing httpd**

package 'httpd' do

action :install

End

**#Remove httpd**

package 'httpd' do

action :remove

end

**#Starting & enabling httpd**

service 'httpd' do

action [:enable,:start]

End

**#Create index.html on apache**

File ‘/var/www/html/index.html’ do

Content ‘<h1>Chef Rocks</h1>’

End

To change the version of cookbook

Metadata.rb

**To uninstall apache**

Sudo yum remove httpd

**Custom Recipes: chef run list multiple recipes**

Breaking recipes Default rb into multiple rb

install\_apache.rb

enable\_apache.rb

deploy\_code.rb

-- Updating a run list of a node

Knife node run\_list set ‘node1’ ‘recipe[myapache::install\_apache],recipe[myapache::enable\_apache],recipe[myapache::deploy\_code]’

Note:

In runlist the order of recipes are important

How to remove a run list?

Term:

Converging 3 resources - executing the resources

**Recipe Dependencies:**

**One recipe dependent on another recipe.**

**Include\_recipe ‘install\_apache’**

**Service ‘httpd’ do**

**Action [:enable,:start]**

**End**

**How to test cookbooks?**

**-- Create .kitchen.yml**

**kitchen driver create kitchen-name vagrant**

**Here vagrant - to create vm on top of it converge the cookbook**

**-- Where we have kitchen.yml it is under cookbook .**

**--**

**kitchen converge**

**Kitchen login // to verify**

**Serverspec: Framework for Unit Testing**

**-- To Test resources of cookbook**

**-- httpd installed or not**

**Serverspec.org site**

**Dependency management:**

**-- community cookbooks**

**-- transitive dependency**

**Eg: To install java and tomcat using chef**

**We can write our own code otherwise there are community cookbooks available in chef**

**Supermarket just use them as dependency.**

**Berkshelf**

**-- As a configuration file.**

**Eg: install maven**

**We can mention in berksfile directly**

**And also in metadata.rb**

**depends ‘maven’, ‘~> 4.0.1’**

**Integration with chef:**

1. **Chef developers push latest cookbook changes to GIT**
2. **Latest modifications in the GIT triggers a job in jenkins**

**- this job uploads the cookbooks to chef server**

**- It calls chef-client on chef nodes**

**-- Jenkins wants use knife command for interacting with chef server**

**-- We need to install chef development kit on jenkins m/c**

**Steps:**

**Goto Download chefDk**

**Install ChefDk in CentOS**

**Rpm -ivh rpm file**

**Copy the rpm URL:**

**Step -1:**

**Sudo wget**

**Step 3:**

**-- Create job in jenkins under this workspace create .chef foloder knife .rb**

**And username.pem**

**var/lib/jenkins**

**Push chef code to git repo**

**Should not upload .chef it have password**

**Should not upload in open area.**

**Attributes:**

**Node Specific data**

**Ohai:**

**-- Its is a tool.**

**-- Install on node**

**-- Ohai is installed on the chef node when it is bootstraped.**

**-- Collect node specific information**

**-- node deatils,software details.**

**-- which collects hardware specifi data,software specific data.**

**-- ohai collect the attributes in the form of JSON**

**--**

**commands:**

**ohai**

**ohai | grep "platform"**

**ohai | grep "hostname"**

**ohai | grep "ipaddress"**

**Use:**

**make recipe generic**

**case default []**

**Using Node attributes in the recipe**

**package\_name=’apach2’**

**Case node[‘platform\_family’]**

**When ‘rhel’ then(optinal)**

**Then**

**package\_name=’httpd’**

**Else**

**package\_name=’apache2’**

**End**

**Package package\_name do**

**Action :install**

**End**

**Custom Attributes:**

**We can define custom attributes**