Notes for Chef

- *Installing a package can be done by…*

```
package “<packageName>” do
   action [:install | upgrade | remove ]
end
```

- *Starting a service can be done by…*

```
service “<serviceName>” do
   action [:start | enable | disable | restart | stop ]
end
```

- *Managing a file/directory by…*

```
file “<fileName>” do
   action [:create | create_if_missing | delete | touch ]
end

directory “<directoryName>” do
   action [:create | delete ]
end
```

- *Running a command by…*

```
execute “<commandName>” do
   command “<commandString>”
   path [“/bin”,”/usr/bin”,”/opt/local/bin”]
   user “<aUser>”
end
```

- *Creating a new cookbook can be done by…*
    - cd ~/chef-repo
    - knife cookbook create <name>
    - cd cookbook/name
    - mkdir files templates (making files and templates available)
    - cd cookbook/<name>/recipes/
    - vi default.rb (embed logic) for example….
    - knife cookbook upload <name> (uploading cookbook to the server)

```
   package "ntp" do
```

```
          action [:install]
     end

     remote_file "Copy service file" do
          path "/etc/init.d/someService"
          source "file:///home/someService"
          owner 'root'
          group 'root'
          mode 0755
     end

     template "/etc/ntp.conf" do
          source "ntp.conf.erb"
          variables(
               :ntp_server => "time.nist.gov" )
          notifies :restart, "service[ntpd]"
     end

     service "ntpd" do
          action [:enable,:start]
     end


     # Install
     package "apache2" do
          action :install
     end

     # Service
     service "apache2" do
          supports :restart => :true
          action [:enable, :start]
     end
```

Static files to be deployed

Static files to be deployed are placed in the files/ directory and referenced via

```
     cookbook_file "/server/path/to/file.ext" do
       source "filename.ext"
       owner "root"
       group "root"
       mode 00600
       action :create_if_missing
     end
```

Template files to be deployed

Template files to be deployed are placed in the templates/ directory and referenced via

```
template "/etc/ntp.conf" do
     source "ntp.conf.erb"
     variables(
          :varName => "This is a variable",
          :varNameEx => “This is another variable” )
     notifies :restart, "service[ntpd]"
end
```

With the template file using variables replaced using ERB

```
<%= @varName %>
<%= @varNameEx %>
```

Running the cookbooks

- *Running the cookbooks…*
    - `cd ~chef-repo/cookbooks`
    - Load the cookbooks into Chef
    - `knife cookbook upload <recipeName>`
    - **Create role**
    - `cd ~/chef-repo/roles`
    - vi roleName.json

```
{
     “name” : “roleName”,
     “default_attributes” : {
     },
     “run_list” : [
          “recipe[recipeName]”,
          “recipe[recipeName::anotherRecipe]”
     ]
}
```

- Import the role via `knife role from file <rolename>.json`
- Create node and bootstrap it with the role/recipe just created
  `knife bootstrap <nodeName> -x <user> –P <pwd> –N <nodeAlias> -r “role[roleName] | recipe[recipeName]”`
- `knife node show <nodeName>`
- `ssh nodeName, then sudo chef-client`

Creating a node

- *Creating nodes…*

```
knife bootstrap <fullNodeName|IPADDR> --sudo –x
<sshUser> -P <sshPwd> -N “<nodeAlias>”
```

Creating an environment

- *Creating environments…*

    - `cd ~chef-repo/`
    - `mkdir environments`
    - `cd environments`
    - `vim dev.rb production.rb`

```
# dev.rb
name “dev”
description “Dev environment”
cookbook “cookbookName”, “=0.2.0”
cookbook “cookbookName2”, “=0.1.1”

# production.rb
name “Production”
description “Prod environment”
cookbook “cookbookName”, “=0.1.0”
cookbook “cookbookName2”, “=0.1.0”
```

    - Load into Chef
    - `knife environment from file dev.rb`
    - `knife environment from file production.rb`
    - `knife environment list [-l]`
    - `knife environment show dev`
    - `knife environment show production`
    - Assign to a node
    - `knife node edit <nodeName>`
    - Edit “chef_environment” to have “dev”
    - `knife node show <nodeName>`

Creating a data bag

- *Creating data bags…*
    - `cd ~chef-repo/`
    - `mkdir data_bags`
    - `cd data_bags`
    - `mkdir <databagName>`
    - `cd <databagName>`
    - `vim databag1.json databag2.json`

```
# databag1.json
{
  “key1” : “value1”
  “key2” : “value2”
}

# databag2.json
{
  “key1” : “value1”
  “key2” : “value2”
```

```
 }

 ◦ knife data bag from file databag1 databag1.json
 ◦ knife data bag from file databag2 databag2.json
 ◦ cd ~chef-repo/
 ◦ cd cookbooks/<name>/recipes
 ◦ vim default.rb

data_bag(“databag1”).each do |site|
   key_data = data_bag_item(“databag1”, site)
   site_name = key_data[“key1”]

   template “string#{site_name}”
      source “template_txt.erb”
      variables {
          vartxt => key_data[“key1”],
          vartxtA => key_data[“key2”]
         }
    end
end

 ◦ vim ../templates/template_txt.erb

<% if @varText == “key1” %>
    VartxtA = <%= @vartxtA %>
<% end %>
```

Notifies

- Order of precedence is determined by order in file.
- Notifies can be use to notify another resource to take action once this resource is done, A=>B
- Subscribes tells a resource to take action once a resource it has subscribed to takes action, A<=B

```
template "/etc/chef/server.rb" do
  source "server.rb.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, "service[chef-solr]", :delayed
  notifies :restart, "service[chef-solr-indexer]", :delayed
  notifies :restart, "service[chef-server]", :delayed
end
```

Attributes

- *Attributes can be defined at the cookbook level and the role level…*

  - For cookbooks...

```
cd ~/chef-repo
cd cookbooks/<name>/attributes
vim default.rb

# default.rb
default["<name>"]["dir"]          = "/etc/apache2"
default["<name>"]["listen_ports"] = [ "80","443" ]
…
```

- For roles…

```
cd ~/chef-repo
cd roles/
vim <role>.json

{
     “name” : “roleName”,
     “default_attributes” : {
          "<name>”: {
                "key": value"
                   },
          "<name>”: {
                "key": value"
                   }
     },
     “run_list” : [
          “recipe[recipeName]”,
          “recipe[recipeName::anotherRecipe]”
     ]
     "env_run_lists": {
         "production": [
           "recipe[nginx::config_prod]"
         ],
         "testing": [
           "recipe[nginx::config_test]"
         ]
     }
}
```

Other commands

Other commands are…

- To show os attributes

```
ohai
```

- To difference cookbooks…

```
knife diff <cookbookName>
```

- To edit node run list …

  ```
  knife node run_list add “recipe[xxx,cccc,ddd]”
  knife node run_list remove “recipe[xxx,cccc,ddd]”
  ```

- To search for items of interest (Solar) (supports wildcards)…

  ```
  knife search [client | environment | node | role ]
  “key:value” [-a <attribute>]
  ```

- To download cookbooks …

  ```
  knife cookbook site search <string>
  knife cookbook site download <cookbookName>
  knife cookbook site install <cookbook>
  ```

- To install recipe from client (run on client) …

  ```
  chef-client
  ```

- To list nodes and show attributes …

  ```
  Knife node list
  Knife client list
  Knife node show <name> [-l] [-Fj] [-a <attribute>]
  ```