**Installing MongoDB using DevOps Tools**

1. Create cookbooks for 3 main parts of MongoDB

Mongod

Mongos

Mongocfg

1. Create the following folders in each cookbook

chef - Revision 20376: /trunk/cookbooks/mongod

..

attributes/

files/

recipes/

templates/

1. Under the attributes create common.yaml which contains all the variables and values.

# this is where common attributes go which are shared across all environments

# mostly things like roleuser, deployment dir, etc.

app\_name: mongod

deploy\_dir: mongo

role\_user: mongod

base\_dir: /data/svc/mongo

conf\_file: /data/svc/mongo/conf/mongod.conf

sysconf\_file: /etc/sysconfig/mongod

db\_dir: /data/svc/mongo/db

log\_dir: /data/svc/mongo/log/

log\_file: /data/svc/mongo/log/mongod.log

#storage\_engine: mmapv1

storage\_engine: wiredTiger

logappend: true

days\_to\_keep\_logs: 7

1. Create another file called hosts.yaml and add the following code

mongod-shard1:

replica\_set: replicaSet1

node\_group: "mongod-shard1"

mongod-shard2:

replica\_set: replicaSet2

node\_group: “mongod-shard2"

mongod-shard3:

replica\_set: replicaSet3

node\_group: "mongod-shard3"

mongod-shard4:

replica\_set: replicaSet4

node\_group: "mongod-shard4"

1. Under Recipes, Create a ruby file called default.rb

require 'nv\_helpers'

class Chef::Recipe

include NvHelpers::NodeGroupHelper

end

# Make sure we have the appropriate users

localuser "mongod" do

action :add

end

localuser "munin" do

action :add

end

#####################

# Dependencies

#####################

#Install MongoDB

script "install modeling packages" do

interpreter "bash"

user "root"

code <<-EOH

yum install mongodb-org-3.2.6 -y

EOH

end

include\_recipe "base::default"

#####################

# ulimit

#####################

ulimit\_types = %W(hard soft)

item\_types = %W(nofile nproc)

ulimit\_types.each do |ulimit\_type|

item\_types.each do |item\_type|

eharmonyops\_limitsconf "mongod-#{item\_type}-#{ulimit\_type}" do

target "mongod"

type ulimit\_type

item item\_type

value "8192"

action :add

end

end

end

#item\_types = %W(memlock)

#ulimit\_types.each do |ulimit\_type|

# item\_types.each do |item\_type|

# eharmonyops\_limitsconf "mongod-#{item\_type}-#{ulimit\_type}" do

# target "mongod"

# type ulimit\_type

# item item\_type

# value "64"

# action :add

# end

# end

#end

# NOTE: munin-node uses port 4949, which needs to be open on the

# monitored system, so the agent can access this data source.

# See http://mms.10gen.com/help/install.html#hardware-monitoring-with-munin-node.

yum\_package "munin-node"

# Disable separate init script for munin-node: we will launch it manually

# from the init script installed by this cookbook.

script "disable munin-node at startup" do

interpreter "bash"

user "root"

code "chkconfig munin-node off"

end

#####################

# Post-Install Cleanup

#####################

script "post-install cleanup" do

interpreter "bash"

user "root"

code <<-EOH

chmod a+r #{node[:config]['base\_dir']}/bin/\*.js

chmod a+rx #{node[:config]['base\_dir']}/bin/\*.sh

chkconfig --del mongod

if [ -f /etc/init.d/mongod ]; then rm /etc/init.d/mongod; fi

EOH

end

#####################

# Cron Section

#####################

# Log Rotate

cron "Daily rotation of mongod logs." do

user node[:config][:role\_user]

minute "0"

hour "0"

command "#{node[:config]['base\_dir']}/bin/logrotate.sh"

end

1. Under template add scripts to create sharding and replicaset which was explained in original installation document.

logrotate.sh

rs\_add\_secondaries.js

rs\_get\_primary.js

rs\_get\_primary.sh

rs\_initiate.js

rs\_set\_primary.js

rs\_set\_primary.sh

1. Add mongo configuration file with following code.

#replica set

<% if node[:config][node[:service]] && node[:config][node[:service]].has\_key?('replica\_set') %>

replSet=<%= node[:config][node[:service]]['replica\_set'] %>

<% end %>

#where to log

logpath=<%= node[:config]['log\_file'] %>

#log overwritten or appended to

logappend=<%= node[:config]['logappend'] %>

#fork and run in background

fork=true

#path to data files

dbpath=<%= node[:config]['db\_dir'] %>

storageEngine=<%= node[:config]['storage\_engine'] %>