Chefguidelines

Contents

Overview

Jessica's intern project on using Chef for automating NetApp Cluster Data ONTAP management.

A copy of the poster is located at https://wikid.netapp.com/wikid/images/6/65/Jessica_Ko_Poster.pdf .

The code is located at **`/u/vijaye/git/chef_repo.git/`**. After cloning the repo, the custom cookbook is located at **`/cookbooks/vsim`**.

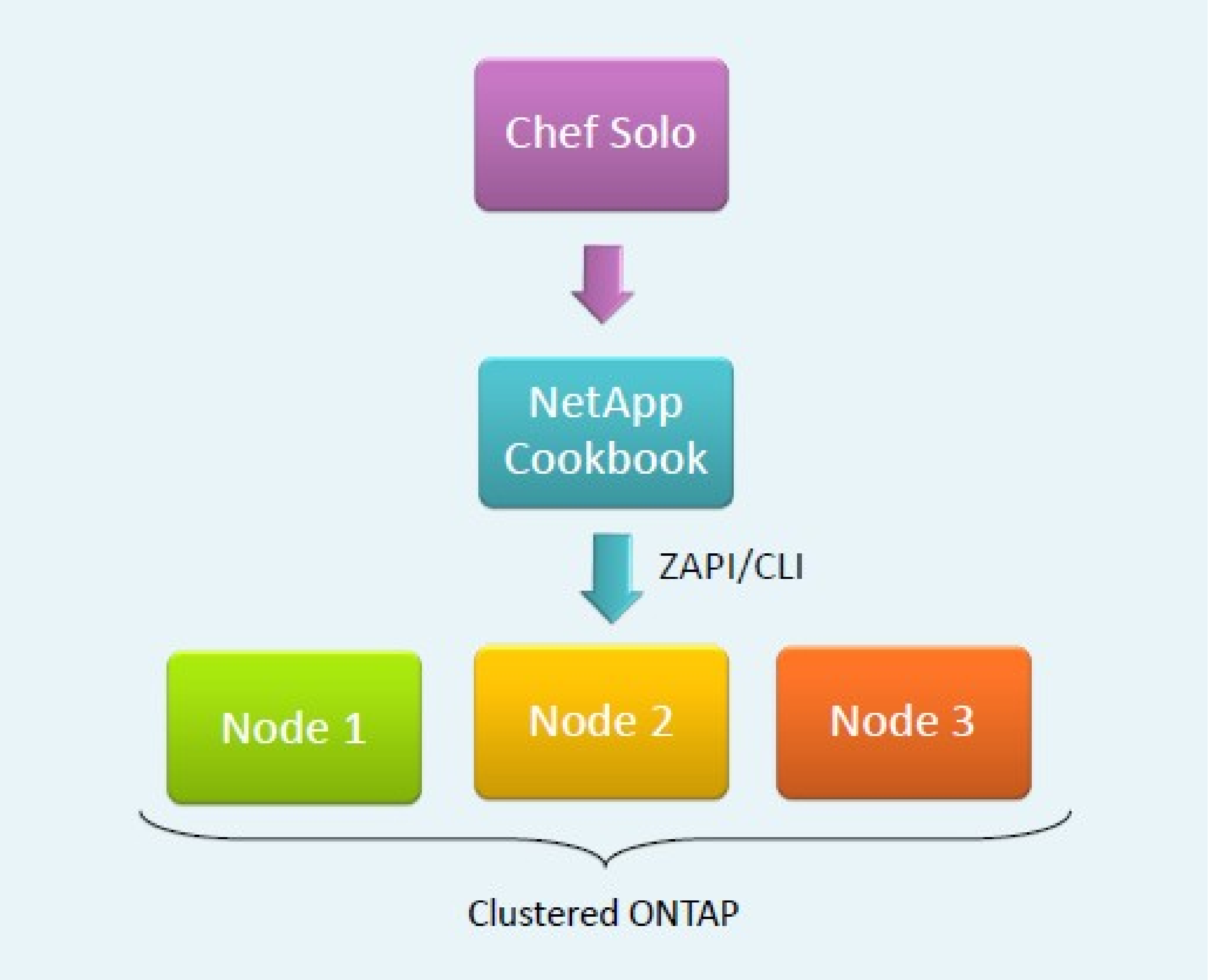
Useful Links

Chef website (https://www.chef.io/)

Chef documentation (http://docs.chef.io/)

Tutorials on how to use Chef (https://learn.chef.io/)

Chef Overview



1. Chef Solo is an executable that runs locally on the computer. (`chef-client -z` or `chef-solo` can run this mode) Chef server is also available for running Chef outside of the workstation.
2. The executable calls the appropriate recipes in the cookbook to run.
3. Through ZAPI or CLI, the recipe will be run on each of the nodes.

Cookbooks

Many cookbooks can reside in the same directory. Cookbooks contain some default settings, recipes, and much more.

For my cookbook, I use the following folders in the cookbook:

- attributes - default settings that can be changed and overriden
- libraries - contains the files from for the ZAPI commands
- providers - code for how the actions should run for objects (resources)
- recipes - files that contain an ordered list of commands to run
- resources - define the available actions and attributes of objects

Set Up

- Download Chef.
- Change the attributes in `/cookbooks/netapp/attributes/default.rb, /cookbooks/vsim/attributes/default.rb` , and /cookbooks/aws/attributes/default.rb
- Add appropriate filepaths for `solo.rb` .

VSIM Cookbook (Custom Cookbook)

Resources and Actions Available

`vsim`

- setup - builds the VSIM
- ha_mode - restart nodes and enable ha mode
- teardown - delete items from vsim folder

`disk`

- assign - assign owner to disks
- remove - remove owner from disks

`aggregate`

- create - make an aggregate on a specified node
- delete - remove aggregate
- relocation - move one aggregate from one node to another node
- rename - change the name of an aggregate
- state - bring an aggregate online or offline
- add - add disks to an aggregate

`cf`

- takeover - starts a takeover of the partner
- giveback - giveback resources

Note: Aggregate's create is slightly different than the aggregate resource in the NetApp cookbook because `node-name` attribute didn't seem to work. The aggregate resource in the vsim cookbook handles the `node-name` attribute differently. However, the delete action is the same in both cookbooks.

Recipes

Recipes, which are in cookbooks, are files that contain an ordered list of commands to run.

A few recipes are available at `/cookbooks/vsim/recipes`.

- aggrcreate.rb - assigns 5 disks to aggregate `aggrdemoA` on one node and assigns 5 disks to aggregate `aggrdemoB` on another node
- aggrdelete.rb - delete `aggrdemoA` and `aggrdemoB`
- aggrmodify.rb - aggrdemoB to offline, relovate aggrdemoA to node2, rename aggrdemoA to new_aggrdemoA, and add 5 disks to new_aggrdemoA

- diskassign.rb - assigns an owner for 5 disks on each node
- diskremove.rb - removes ownership on disks **`VMw-1.15`** and **`VMw-1.19`** (these can be changed)
- giveback.rb - only the giveback action
- ha_mode.rb - runs the **`ha_mode`** action for the **`vsim`** resource
- setup.rb - runs the **`setup`** action for the **`vsim`** resource; currently uses profile **`vsimcha2n12`**
- takeover.rb - only the takeover action

AWS Cookbook

This cookbook builds off of the VSIM cookbook.

AWS HA Setup

Remember to assign the appropriate attributes first. The following steps are based on these steps.

Step 1: Create mediator LUNs

The recipe is `mediatorLUN_create.rb` and the run_list is `mediatorLUNcreate.json`. This will prompt for a password once (need to access cycl servers).

Step 2: Create AWS/HA VSIMs

The recipe is `vsim_setup.rb` and the run_list is `vsimsetup.json`. This will prompt for a password once (need to access cycl servers). The default profile in the recipe is vsimcha2n12, but it can be changed in the recipe. After this step, need to wait for some time for the nodes to boot up.

Step 3: Setup HA mode

The recipe is `ha_mode.rb` and the run_list is `ha_mode.json` . After this step, "cf status" seems to not show that the nodes are in ha mode, but after completing all the steps the nodes seem to be in ha mode by checking (cf status on the filer).

Step 4: Setup ISCSI session

The recipe is `iscsi.rb` and the run_list is iscsi`.json` . This will prompt for a password twice for the diag user.

Step 5: Reboot

The recipe is reboot`.rb` and the run_list `reboot.json` . This will prompt for a password once for the admin user. After running this recipe, wait for both nodes to reboot.

Step 6: Enable HA

The recipe is enable_ha`.rb` and the run_list is `enable_ha.json` . After these steps, check "cf status" on the filer for confirmation.

Step 7: Set up mirrored aggregate

The recipe `setup_mirror_aggr.rb` and the`run_list setup_mirror_aggr.json` will run the following actions. The recipe will disable autoassign for the disks, remove the owner from the spare disks, assign disks to the appropriate node for pool1, and mirror the aggregate. This will prompt for the admin password once.

Step 8: Setup mediator disks

The recipe is `setup_mediator_disk.rb` and the run_list is setup_mediator_disk`.json` .This will assign the two 0f.* disks appropriately to the mediator.

How To

How to Run Recipes

This is one way out of many ways for how to run recipes. (Sometimes keys need to be deleted when making a new vsim for ssh to work correctly.)

In a json file, specify in order of which recipes to run in the **`run_list`**. Some examples can be located in **`/demo.`**

The command to run is

```
chef-client -z -c <config file, ex. solo.rb> --minimal-ohai -j <where the runlist is>
```

- **`-z`** is to specifiy the solo mode to run locally
- **`-c`** is for the config file
- **`--minimal-ohai`** runs minimal Ohai plugins for a fast run (this is optional)
- -j location of a file where some attributes can be defined such as a **`run_list`**

The following command will run the hamode recipe to enable HA mode on the nodes.

```
chef-client -z -c solo.rb --minimal-ohai -j run_vsimhamode.json
```

How to Add Resources

To add custom resources, add a file with the name of the new resource to resources and providers folders. The file in the resources folder will specify the attributes and actions of the new resource. The file in the providers folder will be what code is run when the action is run.

How to Write Recipes

Recipes can be written with blocks like the following. Only one attribute is shown, but many attributes can be listed in the recipe. Looking at the files in resources is helpful for determining what kind of attributes exist. Most of the inputs available for that ZAPI command will be an available attribute.

```
<cookbook name>_<resource name> <name of resource object> do

    <attribute name> <value>

    action <action name> (This is optional because there are default actions too)

end
```

An example of creating an aggregate called "aggrdemoA" of 5 disks.

```
vsim_aggregate "aggrdemoA" do
    disk_count 5
    action :create
end
```

How to Use Resource Names

Table showing what the resource name should be of each aggregate and action.

VSIM Cookbook

Resource	Action	Meaning of Name
vsim	setup, ha_mode, teardown	No meaning
	reboot	"all" for all nodes or node-name of one node
disk	assign	Node to assign disk on
	remove_owner	No meaning
	create_pool_spare	
	auto_assign	"all" for all nodes or node-name of one node
aggregate	create	Name of the aggregate
	delete	
	state	

	add	
	mirror	
	relocation	Destination node name of the relocation
	rename	Original aggregate name
cf	takeover	Node performing the action on the partner node
	giveback	
	mode	node name for the action to be acted on
	service_enable	

AWS Cookbook

Resource	Action	Meaning of Name
mediatorLUN	create	No meaning
setup	setup_iscsi	No meaning
	setup_mediator_disk	

Demo Videos

- VSIM setup
- Enable HA mode
- Disk and aggregates
- VSIM teardown