vcluster Command Structure

Seo-Young Noh, Dada Huang*

National Institute of Supercomputing & Networking
Korea Institute of Science and Technology Information
Youseong, Daejeon, 305-806, Korea
{rsyoung, huang_dada}@kisti.re.kr

September 5, 2013

Abstract

This report provides the command scheme for vcluster.

1 TODO List

- Need to articulate this article
- Need to add outputs of commands
- Sort commands in alphabetical order

2 Overall View of Commands

A vcluster command consists of **Command Category** and **Real Command**. Command category indicates where a real command is belonging to. For example, we have to use plugman as a command category when handling plugin related works. Following shows command categories implemented (or to be implemented) in vcluster.

---

cloudman	command
plugman	command

*Additional authors will be listed depending on their contributions.

```
vmman               command
[vclman]            command
...                 ...
```

There is a special command category called `vclman` which can be omitted. Types of commands belonging to this category are including configurations, `start` and `stop` of subsystems of `vcluster`.

3 Commands included in **vclman** Category

Commands in this category are including configuration, settings, starting and ending modules. `vclman` is generally obmitted. You can print out all commands in this category with `--help` option.

```
  vclman
    --help      list up commands which belong to this category,
                all commands and usages
```

3.1 **check_q**

This command checks the status of a queue of a batch system. Since this command requests data from an underlying batch system, a batch system plugin has to be loaded first. Like the other command, `--help` shows the usages and possible options.

```
vclmann
    check_q --help
      [ --help |
        -h | --type=hold
        -i | --type=idle
        -r | --type=running ]
```

`check_q` command prints out the status of a queue. It can be used with an option to extract specific jobs. For example, `-r` option returns the number of running jobs while `-i` and `-h` return the number of idle jobs and holding jobs respectivley. Without an option, it shows all information. The following shows examples of `check_q` command.

```
vcluster> vclman check_q

vcluster> check_q

vcluster> check_q -r
vcluster> check_q --type=running
```

```
vcluster> check_q -i
vcluster> check_q --type=idle

vcluster> check_q -h
vcluster> check_q --type=hold
```

3.2 check_p

This command checks the status of a pool of a batch system. Since this command requests data from an underlying batch system, a batch system plugin must be loaded first.

```
vclmann
    check_p
      [ --help |
        -r | --type=running
        -i | --type=idle
        -t | --type=trouble ]
```

`--help` option shows the usage of this command. `check_p` command prints out the status of a pool. It can be used with an option to extract specific systems. For example, `-r` option returns the number of running machines while `-i` returns the number of idle. `-t` option shows how machines are in trouble, which is not in service. The following shows some examples of this command.

```
vcluster> vclman check_p

vcluster> check_p

vcluster> check_p -r
vcluster> check_p --type=running

vcluster> check_p -i
vcluster> check_p --type=idle

vcluster> check_p -t
```

```
vcluster> check_p --type=trouble
```

3.3 start

This command makes a subsystem to start its service. For example, `vmman`, virtual manager, has to be running before creating virtual machines.

```
vclmann
    start
      [ --help |
        vmman | ]
```

`--help` option shows the usage of this command and the other options which can be used in this command.

```
vcluster> vclman start vmman

vcluster> start vmman
```

3.4 stop

This command makes a subsystem to stop its service. For example, `vmman`, virtual manager, has to be running before creating virtual machines. In order to stop this service, use `stop` command.

```
vclmann
    stop
      [ --help |
        vmman | ]
```

`--help` option shows the usage of this command and the other options which can be used in this command.

```
vcluster> vclman stop vmman

vcluster> stop vmman
```

4 Commands included in **vmman** Category

All virtual machines are managed by `vmman` (Virtual Machine Manager). Virtual machines are created through a cloud plugin and such virtual machines should be tracked by `vmman`.

Like the other categories, `--help` option provides information on commands that belong to this category.

```
vmman
  --help     list up commands that belong to this category,
             all commands and usages
```

4.1 `list`

This command lists all virtual machines managed by `vmman`. Please make sure not all virtual machines are running on a same cloud; they can exist on different cloud systems. Like the other command, `--help` shows the usages and possible options.

```
vmman
    list
      [ --help |
        [ -a | --type=available
          -r | --type=running
          -s | --type=suspended]
        -n CLOUD_NAME | --name=CLOUDNAME
        -refresh [CLOUD_NAME]
        ]
```

Without any option, it lists up all virtual machines. Option `-a` shows how many virtual machines are available. Option `-r` extracts only running virtual machines while `-s` does suspended virtual machines. It can also specify a cloud with `-n` or `--name` option.

when there is an option "-refresh",vcluster would connect to clouds and get the real time vmss informations then refresh the vm container

The below shows examples of `list` command.

```
vcluster> vmman list

vcluster> vmman list -a

vcluster> vmman list -r

vcluster> vmman list -s

vcluster> vmman list -n amazon

vcluster> vmman list --name=amazon

vcluster> vmman list -refresh fermicloud
```

4.2 `create`

This command creates a virtual machine in a cloud system. It can specify a cloud to create a virtual machine or bunch of virtual machines. Without any option, it creates a virtual machine in a default cloud. This command has to communicate with an underlying cloud plugin in order to generate proper command for the cloud. Since a new virtual machine has to be created, `vmman` has to be involved in order to keep track of the newly created virtual machine.

We have to decide whether this command should be included in `vmman` category rather than `vclman` category. I currently put this command in `vmman` category because this command anyway has to communicate with `vmman`.

```
vmman
    create
      [ --help ] |
      [ -n NUM_VMS | --num=NUM_VMS ] |
      { -c CLOUDNAME | --name=CLOUDNAME } [-n NUM_VMS | --num=NUM_VMS ]
```

The below shows some examples of this command.

```
vcluster> vmman create --help

vcluster> vmman create

vcluster> vmman create -n 2

vcluster> vmman create -c fermicloud -n 2
```

4.3 **destroy**

This command remove a specified virtual machine from cloud.

```
vmman
    destroy
      [ --help ] |
      [ID]
```

4.4 **suspend**

This command suspend a specified virtual machine from cloud.

```
vmman
    suspend
      [ --help ] |
      [ID]
```

4.5 `start`

This command starts a suspended virtual machine,it can only performed on a VM whose status is suspend

```
vmman
    start
      [ --help ] |
      [ID]
```

4.6 `show`

This command lists the detail informations of a virtual machine,including internal id in cloud,private and public ip address,launch time,etc.

```
vmman
    show
      [ --help ] |
      [ID]
```

5 Commands of `plugman` Category

All commands after `plugman` are plugin related ones. Such commands are including load, unload, list of plugins. There are two types of plugins which are batch plugin and cloud plugin, respectively. We will discuss plugin related commands in the following subsections.

Like a general Linux command, `--help` option shows the usages of `plugman` and commands.

```
  plugman
    -help     list up all options and usages
```

5.1 load

This command loads a plugin or a bunch of plugins. Since one batch system plugin is only allowed at the same time, the structure of `load` command depends on the type of plugin.

```
plugman
    load
      -c PLUGIN... | --type=cloud PLUGIN...
      -b BATCH_PLUGIN | --type=batch PLUGIN
```

The options `-c` and `--type=cloud` are identical. These options are saying that we are about to load cloud type plugin(s). Like the cloud type options, we can use `-b` and `--type=batch` options for a batch type plugin. Please note that unlike a cloud type plugin, only one batch plugin should be provided.

Examples for this command are as below:

```
vcluster> plugman load -c plugin-1
vcluster> plugman load --type=cloud plugin-1

vcluster> plugman load -c plugin-1 plugin-2 plugin-3
vcluster> plugman load --type=cloud plugin-1 plugin-2 plugin-3

vcluster> plugman load -b plugin-1
vcluster> plugman load --type=batch plugin-1

vcluster> plugman load -b plugin-1 plugin-2 plugin-3
vcluster> plugman load --type=batch plugin-1 plugin-2 plugin-3
```

5.2 unload

This command unloads a plugin or a bunch of plugins. This command unlike `load` command does not indicate the type of plugin(s) to be unloaded.

```
plugman
```

```
    unload PLUGIN...
```

Below shows an example of this command.

```
vcluster> plugman unload plugin-1
vcluster> plugman unload plugin-1 plugin-2
```

5.3 list

This command lists up all designated plugins. Option `--help` shows the usage of this command and options in detail.

```
plugman
    list --help
```

Since `vcluster` does not have `register` command, it retrieves all plugins under a specified plugin directory. When listing up plugins currently being used, option `-l` or `--loaded` can be used at the end of command.

```
plugman
    list
      -c | --type=cloud
      -b | --type=batch
      -l | --loaded
```

You may combine `-c` and `-l` options to show up all loaded cloud plugins. The following shows examples of this command.

```
vcluster> plugman list -c
vcluster> plugman list --type=cloud

vcluster> plugman list -b
```

```
vcluster> plugman list --type=batch

vcluster> plugman list -l
vcluster> plugman list --loaded

vcluster> plugman list -l -c
vcluster> plugman list -l --type=cloud
vcluster> plugman list --loaded -c

vcluster> plugman list -l -b
vcluster> plugman list -l --type=batch
vcluster> plugman list --loaded -b
```

TODO: when listing up all plugins under a directory, the output should explicitly mention that the outputs are coming from a directory, not from memory.

5.4 `info`

This command prints the information about a plugin. It will be used to retrieve detailed information about the plugin. TODO: plugin interface needs to provide this feature. We may need to introduce a structure containing required fields for this command.

```
plugman
    info PLUGIN
```

Below shows an example of this command.

```
vcluster> plugman info plugin-1
```

---

6 Commands included in `cloudman` Category

This command category is related with the management of cloud systems. It handles the information about cloud systems which will be used in `vcluster`. In order to create a virtual machine, `vcluster` must load cloud systems first. In general, this command category handles `cloudelement` defined in a configuration file.

Like a general Linux command, `--help` option shows the usages of `cloudman` and commands.

```
cloudman
  -help     list up all options and usages
```

6.1 `register`

This command registers cloud systems defined in a configuration file as `cloudelement`.

```
cloudman
    register <cloudsystem config file>
```

It registers all cloud systems defined in `<cloudsystem config file>`. All cloud systems are defined as `cloudelemet` in the configuration file. The following shows an example. TODO: we have to finalize all necessary fields in cloudelement.

```
[cloudelement]
type = private
name = mycloud
interface = proxy-opnenebula
endpoint = 150.183.234.2
max = 20

[cloudelement]
type = private
name = yourcloud
```

```
interface = proxy-openstack
endpoint = 150.183.234.3
max = 20

[cloudelement]
type = public
endpoint = http://fcl002.fnal.gov:4567/
accesskey = rsyoung
secretkey = <rsyoung's secretkey>
instancetype = m1.test
image = ami-00000171
max = 10
version = 2011-05-15
signatureversion = 2
signaturemethod = HmacSHA256

[cloudelement]
type = public
endpoint = https://ec2.amazonaws.com/
accesskey = <your ec2 access key>
secretkey = <your ec2 secret key>
instancetype = m1.large
keyname = seoyoungnoh_key
image = ami-00000171
max = 30
version = 2011-05-15
signatureversion = 2
signaturemethod = HmacSHA256
rsyoung@diogenes:~/developments/vcluster$
```

Below shows an example of this command.

```
vcluster> cloudman register cloudsystems.conf
```

6.2 load

This command loads a cloud system which has been unloaded. The cloud system must be defined in cloud system configuration file. When `cloudman` registers the configuration file, the cloud system was loaded at the first time,

but at a certain point, it was intentionally unloaed in order to prevent to launch virtual machines by `vmman`.

```
cloudman
    load
      CLOUDNAME...
```

If a cloud system has been already loaded, then this command does not affect the cloud system information; otherwise, the cloud system will be activated and `vmman` can create virtual machines.

Examples for this command are as below:

```
vcluster> cloudman load fermicloud
vcluster> cloudman load fermicloud gcloud
```

6.3 unload

This command unloads a cloud system which has been already loaded.

```
cloudman
    unload CLOUD...
```

Below shows an example of this command.

```
vcluster> plugman unload cloud-1
vcluster> plugman unload cloud-1 cloud-2
```

6.4 list

This command lists up all registered clouds,inclouding loaded and unloaded clouds. Option `--help` shows the usage of this command and options in detail.

```
cloudman
        list
          -l | --loaded
          -u | --unloaded
```