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

July 3, 2013

Abstract

This report provides the command scheme for vcluster.

1 TODO List

- Need to articulate this article
- Need to add outputs of commands

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
vmman             command
```

[*] Additional authors will be listed depending on their contributions.

```
    [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 vcluster.

3 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, respectivley. We will discuss plugin related commands in the following subsections.

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

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

3.1 **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. Pleas 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
```

3.2 **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
```

3.3 **list**

This command lists up all designated plugins. 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
```

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

3.4 info

This command prints the information about a plugin. It will be used to retrieve detail 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
```