Plugin API

Functions

get_versions()

Returns all versions of Hadoop that could be used with the plugin. It is responsibility of the plugin to make sure that for each version it has available required images, configs, whatever else it needs to create the Hadoop cluster.

Returns: list of strings - Hadoop versions

Example Return Value: (“Apache Hadoop 1.1.1”, “CDH 3”, “HDP 1.2”)

get_configs(hadoop_version)

Lists all configs supported by plugin with descriptions, defaults and node process for which this config is applicable.

Returns: list of **config**s

Example Return Value: ((“heap size”, “512”, true, “jt”))

get_supported_node_types(hadoop_version)

Lists of all supported NodeTypes for a given Hadoop version.

Returns: list of strings - node types

Example Return Value: (“mgmt”, “jt+nn”, “tt+dn”)

validate_cluster(cluster_description)

Validates **user_input**s in a given **cluster_description**. Returns empty list if all inputs are correct. Otherwise for each incorrect input function should return **validation_error** with meaningful content.

For each **user_input** the function must check that it is applied to the correct Node Type, i.e. that corresponding **config.applicable_node_processes** and Node Type has one of the node processes in common. Also function should check that the provided value is correct for the given **config**.

Returns: list of **validation_error**s

*Example Return Value***:** ((“heap size”), (“Heap size is a required field and must be specified”), (“mapred.task.timeout”, “The parameter must be int”))

get_infra(cluster_description)

Gets specifications for VMs required for cluster. Basically cluster description already contains specification for cluster nodes - number of VMs for each flavor. And in general case, plugin should just copy specs from **cluster_description** to its **vm_request**s, but that is not a strict rule. When returning **vm_request**s, plugin

- must specify image for VMs
- could change VMs specs in any way it needs. For instance, plugin can ask for additional VMs for the management tool.

*Returns***:** list of **vm_request**s

Example Return Value: [(jt+nn", "m1.medium", "ubuntu-12.10", 1, {"set_root_pswd"="qwerty", "generate_keys"=True), ("tt+dn", "m1.small", "ubuntu-12.10", 10, {"set_root_pswd"="qwerty", "generate_keys"=True)]

configure_cluster(cluster_description, vm_specs)

Configures cluster on provisioned by savanna VMs. In this function plugin should perform all actions like adjusting OS, installing required packages (including Hadoop, if needed), configuring Hadoop, etc.

Returns: None

start_cluster(cluster_description, vm_specs)

Start already configured cluster. This method is guaranteed to be called only on cluster which was already prepared with **configure_cluster(...)** call.

Returns: None

scale_cluster(cluster_description, remaining_vm_specs, new_vm_specs, delete_vm_specs)

Scales cluster - adds/removes nodes to/from active cluster. This function should configure new nodes and attach them to cluster. Additionally here plugin can make some cleanup on VMs that will be deleted.

Returns: None

on_terminate_cluster(cluster_description, vm_specs)

When user terminates cluster, Savanna simply shuts down all the cluster VMs. This method is guaranteed to be invoked before that, allowing plugin to do some clean-up.

Returns: None

Objects

All fields are strings unless specified otherwise

config

Specifies a single key-value pair. Both key and value are strings.

config_name
default_value
is_optional: boolean
applicable_node_processes: list of strings

user_input

Value provided by user for a specific config.

config_name
value

vm_group_description

Specifies group of VMs within a cluster.

node_type
flavor
configs: list of **user_input**s
count: int

cluster_description

Contains all relevant info about cluster. This object is is provided to the plugin for both cluster creation and scaling.

cluster_name
cluster_configs: list of **user_input**s
hadoop_version
vm_groups: list of **vm_group_description**s

It is guaranteed that cluster contains exactly one group with node_type == 'mgmt'. It is also guaranteed that cluster either contains one group with node_type == 'jt+nn', or one group 'jt' and one group 'nn'. In all these cases group's count is exactly 1.

validation_error

Describes what is wrong with one of the values provided by user.
config_name
error_message

vm_request

Specifies VMs which plugin requests from Savanna.

flavor
image
count: int
configs: list of **config**s

vm_specs

Specifies VMs created by Savanna for plugin

flavor
image
count: int
configs: list of **config**s
servers: list of **server**s

server

Specifies a single VM created for cluster

ip
compute_host
auth type (password / key) ?
credentials ?

Image Registry API

A component helping plugin find suitable images by some criteria. All search is based on tags. A tag is just a string. Each image could have several tags simultaneously.

set_description(image, os_description, hadoop_version, extra)

Sets human-readable information for image, for example, “Ubuntu 13.04 x86_64, Apache Hadoop 1.1.1, Java 1.7u21”
Returns: None

tag_image(image, tags)

Adds tags to image

Returns: None

untag_image(image, tags)

Removes tags from image

Returns: None

get_image_tags(image)

Queries all tags for the given image

Returns: list of strings - image tags

get_image_by_tags(image, tags)

Queries images having all of the specified tags

Returns: list of strings - images' ids

VM Manager

A pack of low-level helpers to help plugin interact with vms

execute(command)

Executes command in shell on VM via ssh (non-interactive)

copy_to_vm(filename)

Copies file to VM via ssh

copy_from_vm(filename)

Copies file from vm via ssh

Additionally we are thinking about adding some helpers for some high-level actions, for example:

install package (using apt-get/yum ?)