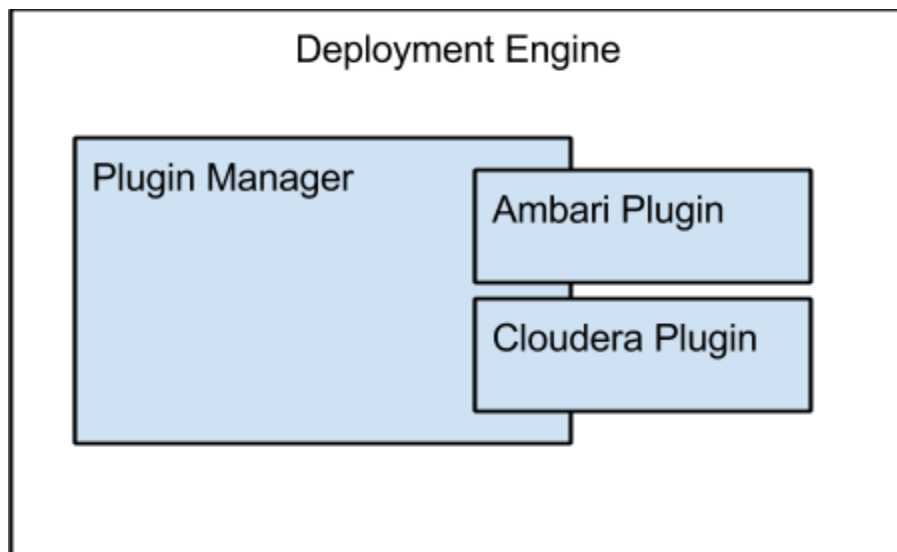


Savanna Deployment Engine

The Savanna Deployment Engine is identified as a pluggable component responsible for deploying Hadoop on provisioned VMs while leveraging Hadoop management solutions such as Ambari. This document proposes an approach for implementing such a component. We provide a more in-depth view of the components that comprise the Deployment Engine, the APIs they expose, and the sequence of events involved in common Hadoop cluster operations.

Deployment Engine Architecture

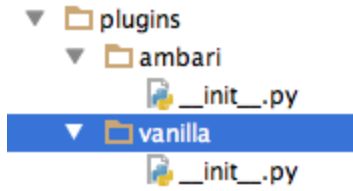


The Deployment Engine has the following components:

- Plugin Manager - an element responsible for loading and registering Hadoop Management plugins.
- Hadoop Management Plugin(s) - responsible for exposing a set of functions to the Savanna controller that allow for the creation, modification, and control of Hadoop clusters.

Plugin Manager

The Deployment Engine Plugin Manager loads and registers management plugins. These plugins are packaged as python modules and are deployed to an agreed upon location. To ascertain which plugins are available for Hadoop cluster management, the plugin manager simply iterates thru the modules it finds under the well known location.



In addition, the plugin manager initializes the plugins, allowing them to allocate resources required for the execution of management operations etc. During shutdown, the plugin manager iterates thru the set of loaded plugins and invokes a shutdown method, providing the opportunity to release any held resources and execute any cleanup they may require.

Hadoop Management Plugins

The management plugins are responsible for the actual interactions with the back end management systems. To that end, each plugin will expose the following methods:

Method Name	Inputs	Outputs	Description
init()	<ul style="list-style-type: none"> savanna_ctx - a context object that provides access to the savanna controller's public services and data. 	Void	This method is invoked during the initial load of the plugin. The plugin will generally cache the provided context object.
validate_cluster()	<ul style="list-style-type: none"> node_template - provides access to the configuration items associated with the hadoop cluster to provision. 	result - true if the cluster is valid, false otherwise	Validates the configuration choices for the indicated Hadoop cluster.
create_cluster()	<ul style="list-style-type: none"> cloud_ctx - an object providing access to the provisioned VMs and allowing for the execution of operations on those VMs node_template - object providing access to configuration items for the Hadoop cluster being 	cluster_info - an object with attributes describing the created cluster, such as cluster name, host names, configured services, etc.	This method is invoked to initiate the Hadoop cluster provisioning subsequent to the provisioning of the VMs by the Savanna VM Provisioning component.

	created. This object is opaque to the controller and unique to the plugin being invoked.		
add_hosts()	<ul style="list-style-type: none"> • cloud_ctx - see above • node_template - an object specifying the characteristics of the hosts to add. The plugin can infer the appropriate operation to execute to expand the Hadoop cluster (e.g. if hostname(s) are specified, add nodes to those hosts) 	host_info - an object containing information about the host(s) added to the cluster.	Allows for the addition of Hadoop servers to an existing cluster. The specified VMs should have already been provisioned by the VM Provisioning component and have the resources available to host a Hadoop instance.
remove_hosts()	<ul style="list-style-type: none"> • cloud_ctx - see above • node_template - in this case, the node spec would specify the nodes to remove from the cluster (either by type or specific host) 	True if the operation was a success, false otherwise	Removes the specified hosts/nodes from the Hadoop cluster. The hosts removed are indicated by the information provided in the node_spec.
get_management_URIs()	<ul style="list-style-type: none"> • cluster_name - the name of the cluster for which the management URI is requested (Optional) 	URI(s) - if no value is provided for cluster_name, all known management URIs for the given tenant are returned. If a value is provided, the the management URI specific to that cluster will be returned.	Returns the URI(s) that provide access to the known management servers (console, REST interface etc) associated with the tenant.
get_cluster_status()	<ul style="list-style-type: none"> • cluster_name - the 	cluster_status	Returns the health

	cluster for which status information is requested (Optional)	- a JSON object providing the current health of the hadoop cluster. If no cluster name is provided, this method will return an aggregated status for all known clusters associated with the tenant.	of the tenant's Hadoop clusters.
exec_resource_action()	<ul style="list-style-type: none"> • resource_request - an HTTP request object that specifies the resource, verb, headers, credentials etc for execution against the Hadoop management server. 	response - an HTTP response object containing the headers, contents etc of the response from the management server.	Invokes a REST request against the management server. This method can parse the URI to determine the appropriate backend management server (host and port), restructure the URI to account for vendor specific resources etc.
get_node_template()	<ul style="list-style-type: none"> • cluster_name - the name of the cluster for which the cluster spec is requested. 	node_template - a node template containing a cluster spec describing the details of the given cluster.	Returns the cluster_spec for the given cluster (contained by a node template). As an example, this method may be invoked during the execution of a Savanna management request for cloning a cluster.
shutdown()	<ul style="list-style-type: none"> • cluster_name - the name of the cluster being shutdown/removed. 	Void	Allows the plugin to perform any clean up required during a cluster shutdown or

			removal.
--	--	--	----------

Management Objects

The following section provides detailed descriptions (function, APIs, etc) for the management associated objects leveraged by the Hadoop management plugins.

Savanna Context

The savanna context is leveraged by plugins to interact with the Savanna management server. It is expected that the server will provide access to certain required facilities such as persistence store access, HTTP request invocation capability, a credential store(?) etc.

The API exposed by the Savanna Context is:

Method Name	Inputs	Outputs	Description
persist()	<ul style="list-style-type: none"> key - the key for storing the given value value - the object to persist 	Void	Persists the given value and associates it to the provided key
lookup()	<ul style="list-style-type: none"> key - the key to the requested value 	value - the value to which the key is mapped	Returns the value mapped to the provided key
exec_http_request()	<ul style="list-style-type: none"> request - an HTTP request object that provides the URI, headers, credentials etc required for invoking the outbound request 	response - the response from the invocation	Invokes the provided request and returns the response
get_credential()*	<ul style="list-style-type: none"> key - a key that maps to a stored credential (e.g. username/password) 	credential - an opaque credential object that can be leveraged for secure invocations	Provides access to credentials managed by the Savanna server
get_logger()	<ul style="list-style-type: none"> None 	logger - returns the	Returns a logger

		savanna controller logger.	appropriate for logging server level log updates.
--	--	----------------------------	---

* optional or under consideration

Cloud Context

The cloud context provides information concerning the cluster being managed (e.g. the VMs provisioned for a cluster creation request) as well as access to the assets associated with the given cluster, most importantly the VM instances.

The cloud context exposes the following functionality:

Method Name	Inputs	Outputs	Description
get_servers()	None	server[] - an array of server instances, each providing a means of interaction with the given server instance	Returns an array of server instances that allow for direct interaction with the VMs in the cluster.
get_logger()	None	logger - a logger associated with the specific cluster	Returns a logger appropriate for logging cluster-specific log messages.

In addition, the cloud context provides access to the following attributes:

- **vm_image[]** - the images associated with this cluster
- **flavor[]** - the flavors associated with this cluster
- **node_count** - the number of nodes in the cluster

Server

The server object provides a means for interacting directly with a VM/Server instance. The Hadoop management plugin, in the process of executing its various functions, requires mechanisms for installing packages, executing shell commands etc directly on the target VMs. It is via the server reference that the plugin can perform such operations.

The Server exposes the following methods:

Method Name	Inputs	Outputs	Description
execute()	<ul style="list-style-type: none"> • command - the text of the command to execute • prompts[] - an array of 	result - contains a boolean status that is true if the command executed successfully, false	This method allows the plugin to execute a command on the server.

	responses to prompts if the command to execute requires such interaction (optional)	otherwise. Also provides access to the associated streams (stdout,stderr)	
install()	<ul style="list-style-type: none"> • package_name - the name of the package to install 	result - True if the package installed successfully, false otherwise	Performs an installation of the indicated package on the server.
open_file()	<ul style="list-style-type: none"> • file_path - the full path to the file requested • mode - specifies how the file system should open the file (read, write, append, random access etc) 	file_handle - a reference to the file	Provides a reference to the file allowing for file-based operations (write, delete, modify)

In addition, the Server exposes the following attributes:

- **public_ip** - the public IP of the server
- **private_ip** - the private IP of the server
- **hostname** - the host name of the server
- **vm_image** - the image associated with the server
- **flavor** - the VM flavor associated with the server
- **host_id** - an identifier for the physical host to which the server is associated
- **rack_id** - an identifier designating the rack to which this server is associated.

NOTE: There may be additional metadata items provided to allow the Hadoop provider to properly configure the hadoop cluster across virtualized assets.