

OpenStack Neat – Dynamic Consolidation of Virtual Machines: Blueprint

Anton Beloglazov

26th of July 2012

Contents

1	Summary	2
2	Release Note	2
3	Rationale	2
4	User stories	2
5	Assumptions	2
6	Design	2
7	Implementation	3
7.1	UI Changes	3
7.2	Code Changes	4
7.3	Migration	4
8	Test/Demo Plan	4
9	Unresolved issues	4
10	BoF agenda and discussion	4

1 Summary

A server template consists of a base image plus the definitions of configuration metadata. For example, a server template might include an Apache HTTP server; the metadata would include the server name, location of the HTML root directory, and tuning parameters. Glance stores the template in its registry; Nova, when creating a new server from the template, would validate the required metadata and configure the internal applications directly.

The metadata could also be used to drive automatically-generated web interfaces to solicit the configuration metadata.

Server templates could greatly increase the flexibility and usability of compute clouds; rather than creating a “bare” server and configuring it manually, this could allow users to prepopulate applications in a server image and configure them automatically.

2 Release Note

This section should include a paragraph describing the end-user impact of this change. It is meant to be included in the release notes of the first release in which it is implemented. (Not all of these will actually be included in the release notes, at the release manager’s discretion; but writing them is a useful exercise.)

It is mandatory.

3 Rationale

4 User stories

5 Assumptions

Glance “stores” the server template and metadata map; Nova must “implement” the server template.

6 Design

This is just one possible design for this feature (keep that in mind). At its simplest, a server template consists of a core image and a “metadata map”. The metadata map defines metadata that must be collected during server creation and a list of files (on the server) that must be modified using the defined metadata.

Here is a simple example: let's assume that the server template has a Linux server with Apache HTTP installed. Apache needs to know the IP address of the server and the directory on the server that contains the HTML files.

The metadata map would look something like this:

```
metadata {
  IP_ADDRESS;
  HTML_ROOT : string(1,255) : "/var/www/";
}
map {
  /etc/httpd/includes/server.inc
}
```

In this case, the `{{{metadata}}}` section defines the metadata components required; the `{{{map}}}` section defines the files that must be parsed and have the metadata configured. Within the `{{{metadata}}}` section, there are two defined items. `{{IP_ADDRESS}}` is a predefined (built-in) value, and `{{HTML_ROOT}}` is the root directory of the web server.

For `{{HTML_ROOT}}`, there are three sub-fields: the name, the data type, and (in this case) the default value. The token `{{{required}}}` could be used for items that must be supplied by the user.

When the server is created, a (as-yet-undefined) process would look at the files in the `{{{map}}}` section and replace metadata tokens with the defined values. For example, the file might contain:

```
<VirtualHost {{IP_ADDRESS}}:*>
  DocumentRoot "{{HTML_ROOT}}";
</VirtualHost>
```

7 Implementation

This section should describe a plan of action (the “how”) to implement the changes discussed. Could include subsections like:

7.1 UI Changes

Should cover changes required to the UI, or specific UI that is required to implement this

7.2 Code Changes

Code changes should include an overview of what needs to change, and in some cases even the specific details.

7.3 Migration

Include:

- data migration, if any
- redirects from old URLs to new ones, if any
- how users will be pointed to the new way of doing things, if necessary.

8 Test/Demo Plan

This need not be added or completed until the specification is nearing beta.

9 Unresolved issues

This should highlight any issues that should be addressed in further specifications, and not problems with the specification itself; since any specification with problems cannot be approved.

10 BoF agenda and discussion

Use this section to take notes during the BoF; if you keep it in the approved spec, use it for summarising what was discussed and note any options that were rejected.