Ganeti remote API

Documents Ganeti version 1.2

1. Introduction

Ganeti supports a remote API for enable external tools to easily retrieve information about a cluster's state. The remote API daemon, `ganeti-rapi`, is automatically started on the master node if the `--enable-rapi` parameter is passed to the `configure` script. Alternatively you can start it manually. By default it runs on TCP port 5080, but this can be changed either in `.../constants.py` or via the command line parameter `-p`. SSL support can also be enabled by passing command line parameters.

> **Note:** Ganeti 1.2 only supports a limited set of calls, all of them read-only. The next major version will have support for write operations.

2. Protocol

The protocol used is JSON (http://www.json.org/) over HTTP designed after the REST (http://en.wikipedia.org/wiki/Representational_State_Transfer) principle.

3. Usage examples

You can access the API using your favorite programming language as long as it supports network connections.

3.1. Shell

```
wget -q -O - http://CLUSTERNAME:5080/info
```

3.2. Python

```
import urllib2
f = urllib2.urlopen('http://CLUSTERNAME:5080/info')
print f.read()
```

3.3. JavaScript

Note: While it's possible to use JavaScript, it poses several potential problems, including browser blocking request due to non-standard ports or different domain names. Fetching the data on the webserver is easier.

```
var url = 'http://CLUSTERNAME:5080/info';
var info;

var xmlreq = new XMLHttpRequest();
xmlreq.onreadystatechange = function () {
  if (xmlreq.readyState != 4) return;
  if (xmlreq.status == 200) {
    info = eval("(" + xmlreq.responseText + ")");
    alert(info);
  } else {
    alert('Error fetching cluster info');
  }
  xmlreq = null;
};
xmlreq.open('GET', url, true);
xmlreq.send(null);
```

4. Resources

4.1. /

/ resource.

Method	Description
GET	Show the list of mapped resources. Returns: A dictionary with 'name' and 'uri' keys for each of them.

4.2. /info

Cluster info.

Method	Description

Method	Description
GET	Returns cluster information. Example: { "config_version": 3, "name": "cluster1.example.com", "software_version": "1.2.4", "os_api_version": 5, "export_version": 0, "master": "node1.example.com", "architecture": ["64bit", "x86_64"], "hypervisor_type": "xen-3.0", "protocol_version": 12 }

4.3. /instances

/instances resource.

Method	Description
GET	Returns a list of all available instances. Returns: A dictionary with 'name' and 'uri' keys for each of them. Example: [{ "name": "web.example.com", "uri": "\/instances\/web.example.com" }, { "name": "mail.example.com", "uri": "\/instances\/mail.example.com" }]

4.4. /instances/[instance_name]

/instances/[instance_name] resources.

Method	Description
GET	Send information about an instance.

4.5. /instances/[instance_name]/tags

/instances/[instance_name]/tags resource.

Manages per-instance tags.

Method	Description
GET	Returns a list of instance tags. Example: ["tag1", "tag2", "tag3"]

4.6. /nodes

/nodes resource.

Method	Description

Method	Description
GET	Returns a list of all nodes. Returns: A dictionary with 'name' and 'uri' keys for each of them. Example: [{ "name": "node1.example.com", "uri": "\/instances\/node1.example.com" }, { "name": "node2.example.com", "uri": "\/instances\/node2.example.com" }]

4.7. /nodes/[node_name]

/nodes/[node_name] resources.

Method	Description
GET	Send information about a node.

4.8. /nodes/[node_name]/tags

/nodes/[node_name]/tags resource.

Manages per-node tags.

Method	Description
GET	Returns a list of node tags. Example: ["tag1", "tag2", "tag3"]

4.9. /os

/os resource.

Method	Description
GET	Return a list of all OSes. Can return error 500 in case of a problem. Example: ["debian-etch"]

4.10. /tags

/tags resource.

Manages cluster tags.

Method	Description
GET	Returns a list of all cluster tags. Example: ["tag1", "tag2", "tag3"]

4.11. /version

/version resource.

This resource should be used to determine the remote API version and to adapt clients accordingly.

Method	Description
GET	Returns the remote API version.