oVirt

Utilizing oVirt's extensibility to fit your needs

Doron Fediuck
Red Hat

Infrastructure.Next
February 2014

Agenda

oVirt

- Part 1 – Consuming oVirt
  - oVirt API
  - oVirt SDK
  - SNMP (New in 3.4!)

- Part 2 – Extending oVirt
  - VDSM hooks
  - Scheduling API
  - UI Plugins

oVirt API

What can I do via API?

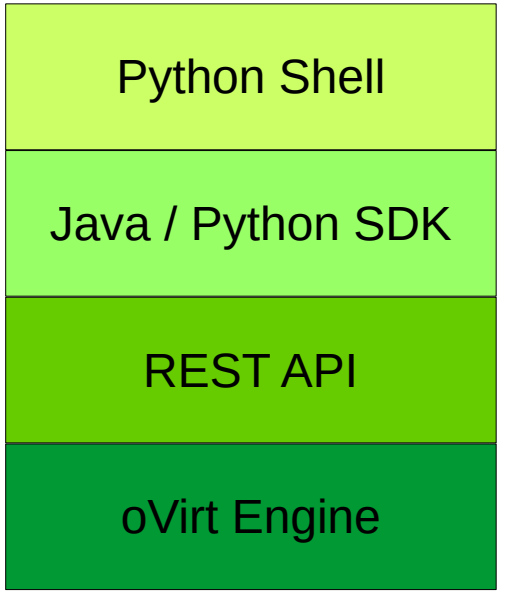
oVirt

- Access via REST/SDK/Shell
- Infrastructure configuration
- Virtual machine (VM) configuration and management
  - Networking for the Guest, Virtual disks, VM properties
- User permissions management
- Advanced operations not available in the GUI

API Concepts

oVirt

- All APIs integrate through the oVirt engine
- All types of APIs are based on web services interface
- Backward and forward compatibility
- Secure access
- Session-based access



API methods

oVirt

REST

`https://host:port/api/vms`

Returns:

- XML/JSON/...

```
<vm id="aee0dbce-1591-44d4-9052-c2209b3e45b8" href="/api/vms/aee0dbce-1591-44d4-9052-c2209b3e45b8">
     <name>Austin</name><actions>
     <link rel="shutdown" href="/api/vms/aee0dbce-1591-44d4-9052-c2209b3e45b8/shutdown"/>
      <link rel="start" href="/api/vms/aee0dbce-1591-44d4-9052-c2209b3e45b8/start"/>
.....................
```

SDK (Python/Java)

`api.vms.list()`

Returns:

- list of VM objects

Shell

list vms

Returns:

- Formatted text

```
id                    : 18df94a7-048f-4306-9cfd-a74e8ea3b907
name                  : Boston
description           : Main service for Boston
cluster-id            : 99408929-82cf-4dc7-a532-9d998063fa95
cpu-topology-cores    : 2
cpu-topology-sockets  : 1
```

oVirt SDK

oVirt SDK

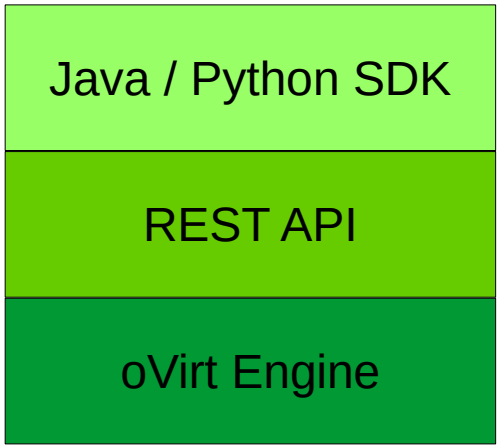
oVirt

- Mainly used for integration or advanced automation
- Object oriented
- Current bindings
  - Java - http://www.ovirt.org/Java-sdk
  - Python - http://www.ovirt.org/Python-sdk
  - libgovirt (GObject wrapper for the oVirt REST API) - https://github.com/GNOME/libgovirt
  - rbovirt: ruby binding for the oVirt REST API- https://github.com/abenari/rbovirt

oVirt SDK - Concepts

oVirt

- Complete protocol abstraction
- Full compliance with the oVirt API architecture
- Self descriptive, intuitive and easy to use
- Auto-generated
- Auto-completion*



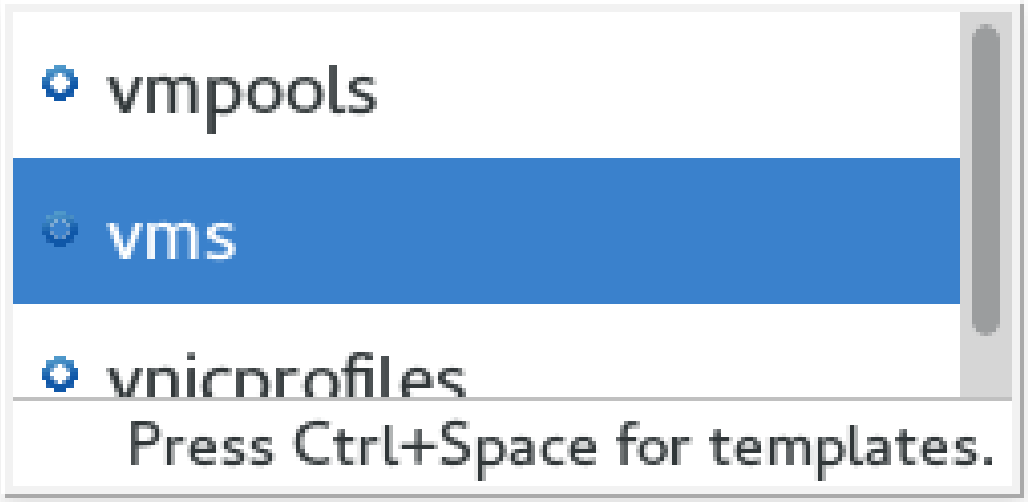
* On supported environments

oVirt SDK – Example: Basics

```
from ovirtsdk.api import API

api = API(url='http://localhost:8080',
          username='user@domain',
          password='password')
```

oVirt SDK – Example: Basics

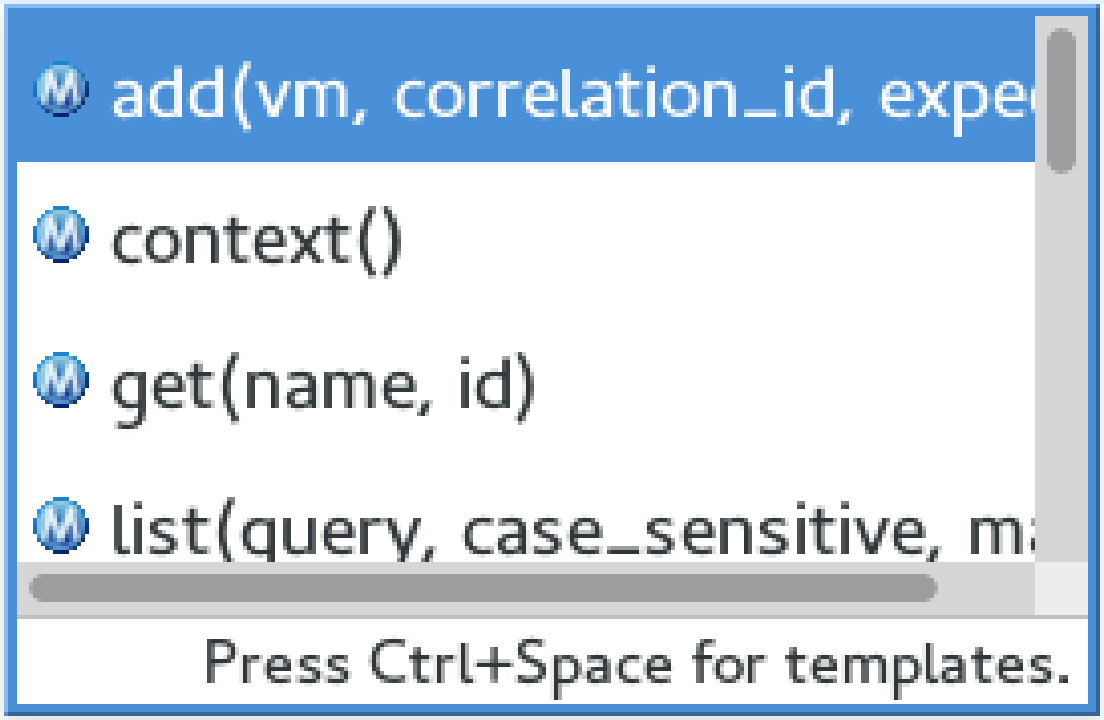
```
from ovirtsdk.api import API

api = API(url='http://localhost:8080',
          username='user@domain',
          password='password')

api.v
```



oVirt SDK – Example: Basics

oVirt

```
from ovirtsdk.api import API

api = API(url='http://localhost:8080',
          username='user@domain',
          password='password')

api.vms.
```



oVirt SDK Example: Adding a VM

oVirt

```
from ovirtsdk.api import API
from ovirtsdk.xml import params

api = API(url='http://localhost:8080',
          username='user@domain',
          password='password')

cluster = api.clusters.get(name='Default')
template = api.templates.get(name='RHEL7_0')
param = params.VM(name='RHEL_VM1',
                  cluster=cluster,
                  template=template,
                  memory=4*1024**3)
vm1 = api.vms.add(param)
```

SNMP support

oVirt SNMP

oVirt

- Coming in oVirt 3.4
- oVirt generated events may be sent as traps

  Allows integration with monitoring systems such as Nagios, BMC Patrol, HP OpenView, etc.
- Raw example

```
SNMPv2-MIB::snmpTrapOID.0 = OID: SNMPv2-SMI::enterprises.2312.13.1.0.30
SNMPv2-SMI::enterprises.2312.13.1.0.30.0 = STRING: "User admin@internal logged in."  SNMPv2-SMI::enterprises.2312.13.1.0.30.1 = STRING: "NORMAL" SNMPv2-SMI::enterprises.2312.13.1.0.30.2 = STRING: "alertMessage"   SNMPv2-SMI::enterprises.2312.13.1.0.30.3 = STRING: "2014-01-12 07:14:22.576"
```

oVirt SNMP

oVirt

Trap structure

OID	Type	Value
SNMPv2-MIB::snmpTrapOID.0	STRING	event message
SNMPv2-MIB::snmpTrapOID.1	STRING	Severity (NORMAL, WARNING or ERROR)
SNMPv2-MIB::snmpTrapOID.2	STRING	Type (ALERT_MESSAGE or RESOLVED_MESSAGE)
SNMPv2-MIB::snmpTrapOID.3	STRING	Log time

oVirt SNMP

oVirt

- Every event type will have '**Up**'
  - VM_MIGRATION_START
- Some have '**Down**'
  - VM_MIGRATION_DONE

More: http://www.ovirt.org/Features/engine-snmp

Agenda

oVirt

- Part 1 – Consuming oVirt
  - Introduction
  - oVirt API
  - oVirt SDK
  - SNMP (New in 3.4!)

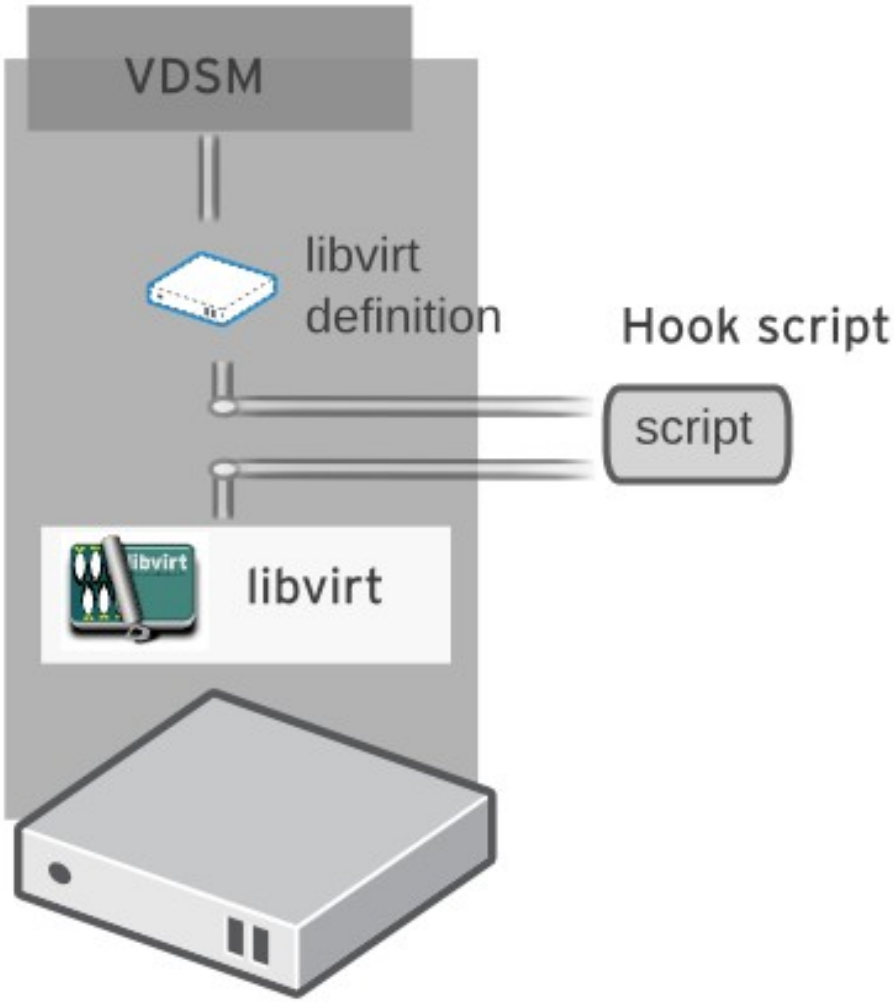
- Part 2 – Extending oVirt
  - VDSM hooks
  - Scheduling API
  - UI Plugins

VDSM Hooks

Hooks

oVirt

- VDSM manages a hypervisor
- “Hook” mechanism for customization
  - Allows administrator to define scripts to modify VM/VDSM operation
    - Extend or modify VM configuration
    - Run different system scripts



Hooks

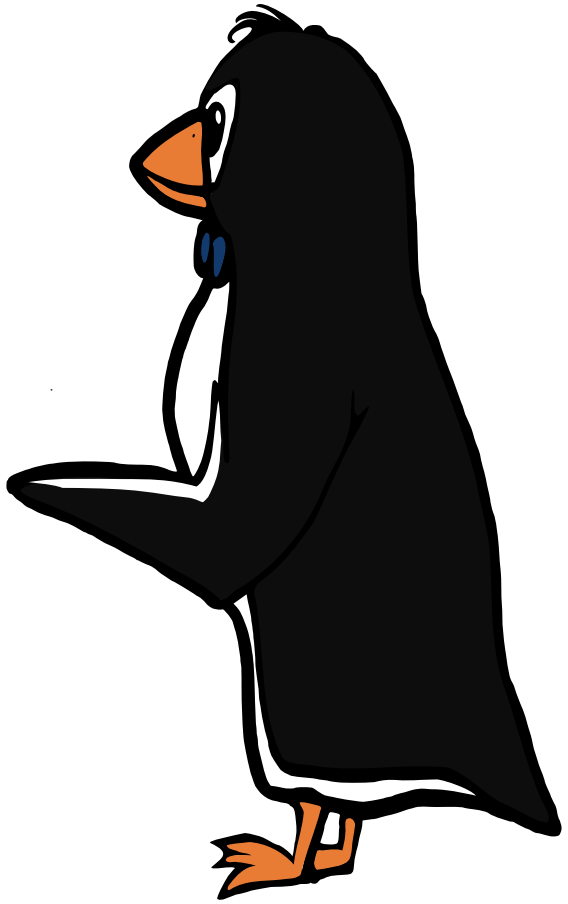
oVirt

- Hook scripts are called at specific events
- Hooks can modify a virtual machines XML definition before VM start
- Hooks can run system commands – eg. Apply firewall rule to VM
- More info:
  - **http://www.ovirt.org/Vdsm_Hooks**
  - **http://www.ovirt.org/VDSM-Hooks_Catalogue**

Hook Entry Points

oVirt

- Lifecycle events where you can apply hooks
  - VDSM (management agent) start
  - VDSM stop
  - VM start
  - VM stop
  - VM migration in/out
  - VM Pause
  - VM Continue
  - VM Hibernate
  - VM resume from hibernate
  - VM set ticket
  - NIC hotplug / hotunplug
  - On host networking configuration change



Hook Example

oVirt

```
import os
import hooking

def removeMacSpoofingFilter(interface):
    for filterElement in interface.getElementsByTagName('filterref'):
        if isMacSpoofingFilter(filterElement):
            interface.removeChild(filterElement)

def isMacSpoofingFilter(filterElement):
    """
    Accept a filter DOM element
    and checks if it's a mac spoofing filter
    """
    filterValue = filterElement.getAttribute('filter')
    return filterValue == 'vdsm-no-mac-spoofing'

if __name__ == '__main__':
    if hooking.tobool(os.environ.get('ifacemacspoof')):
        domxml = hooking.read_domxml()
        interface, = domxml.getElementsByTagName('interface')
        removeMacSpoofingFilter(interface)
        hooking.write_domxml(domxml)
```

Scheduling API

Scheduling API

oVirt

The need - **construct user-defined scheduling policy**

Re: [Users] How to define max number of running VMs on a host?

....

I have 4 graphic workstations with 3 graphic cards on each. I wanna passthrough graphic cards to the VMs one by one, since one workstation has only 3 cards, I must limit the number of running VM on a host to 3.

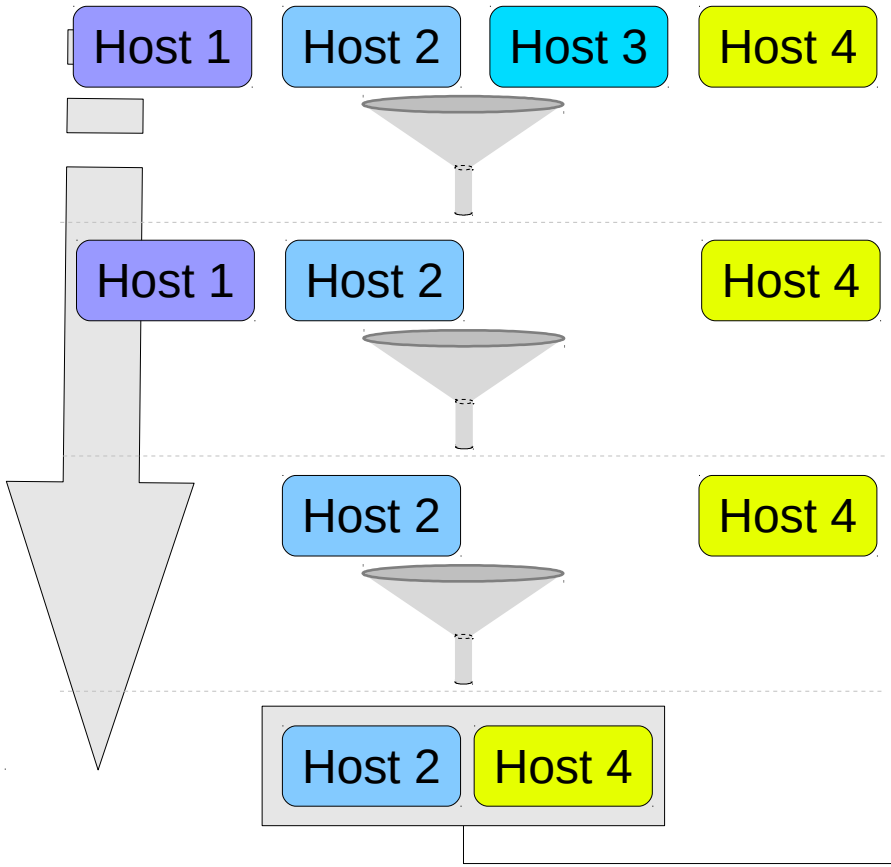
Scheduling Mechanism

oVirt

- Scheduling policy consists of
  - Filter(s)
  - Weight(s)
  - Load balancing module
- Python supported external modules
- Assign the desired policy for a cluster
- More info:
  - http://goo.gl/senjQA - Sample policy units catalog
  - http://www.ovirt.org/External_Scheduler_Samples

Scheduling Mechanism

oVirt

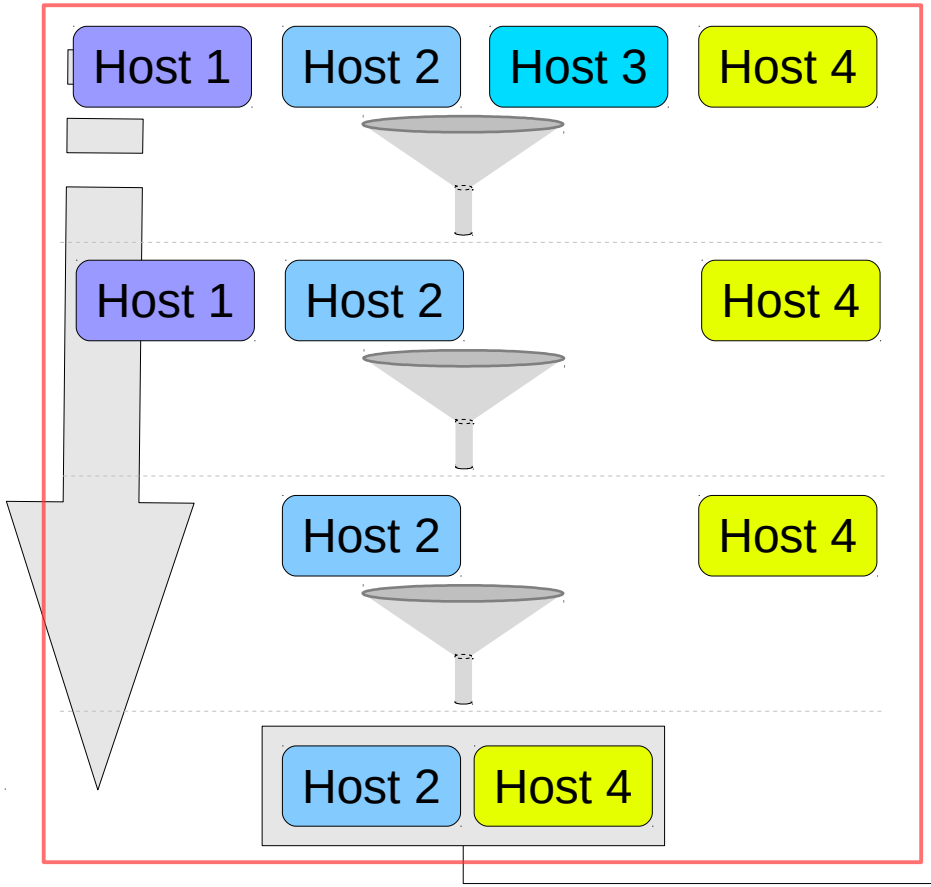


	func 1	func 2	sum
Factor	5	2	
Host 2	10	2	54
Host 4	3	12	**39***

***Host 4 sum: 3*5+12*2 = 39**

Filter Module

oVirt

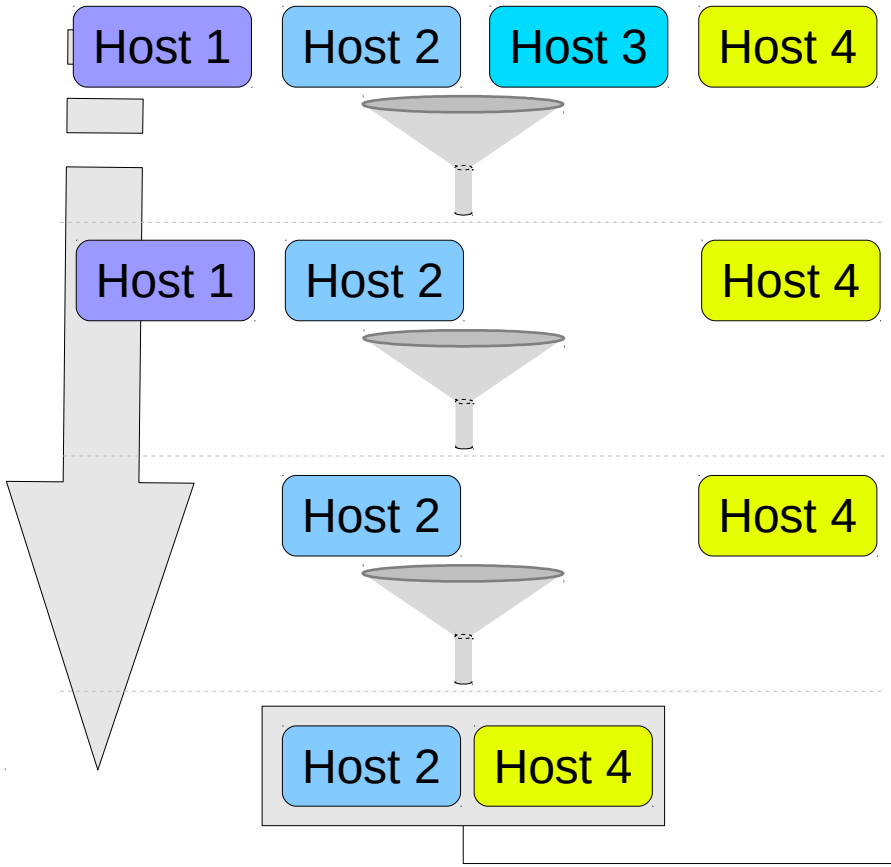


	func 1	func 2	sum
Factor	5	2	
Host 2	10	2	54
Host 4	3	12	**39***

***Host 4 sum: 3*5+12*2 = 39**

Weight Module

oVirt



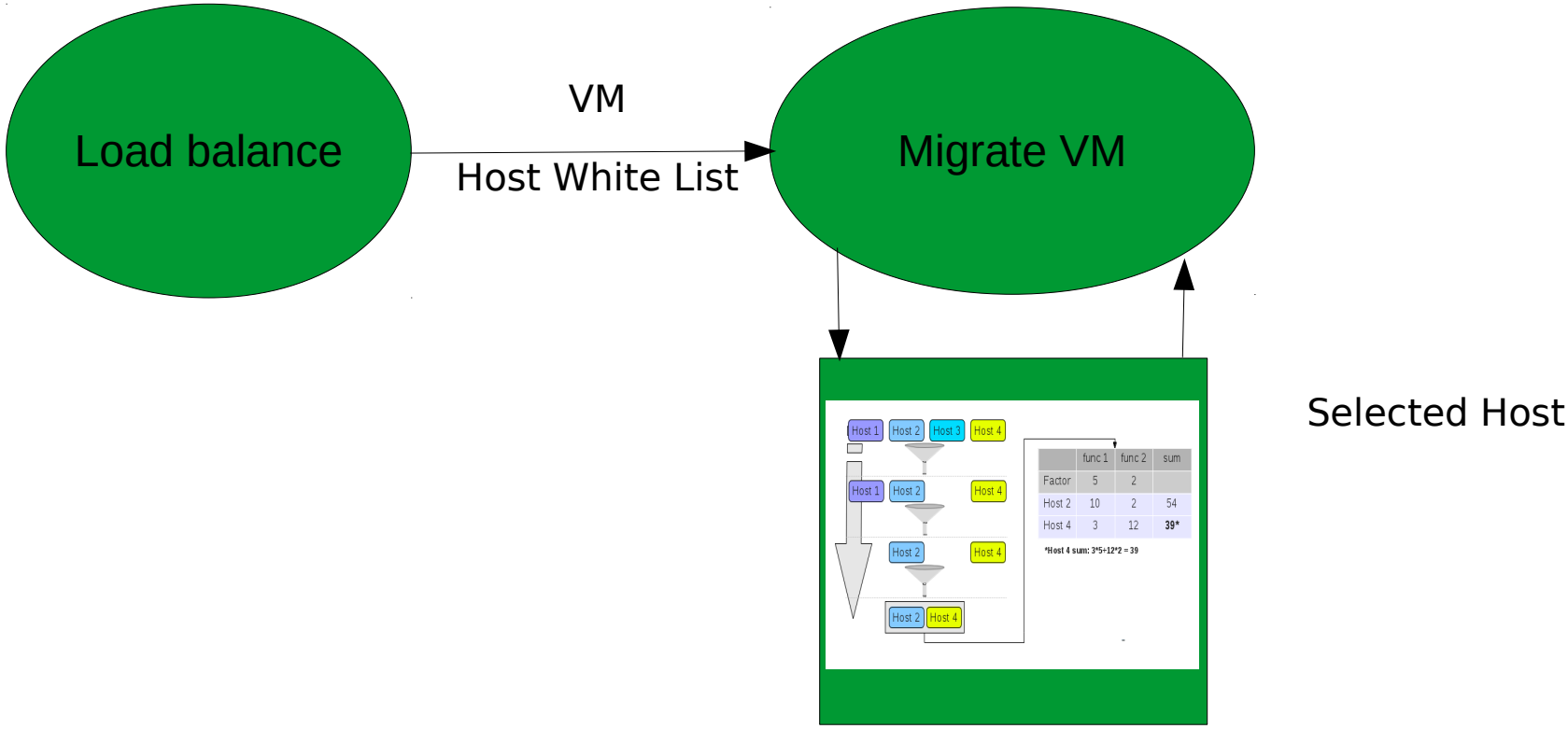
	func 1	func 2	sum
Factor	5	2	
Host 2	10	2	54
Host 4	3	12	**39***

***Host 4 sum: 3*5+12*2 = 39**

Load Balancing

oVirt

- Triggers a scheduled task to determine which VM needs to be migrated
- A single load balancing logic is allowed per cluster



Let's go back to the example

oVirt

Re: [Users] How to define max number of running VMs on a host?

....

I have 4 graphic workstations with 3 graphic cards on each. I wanna
passthrough graphic cards to the VMs one by one, since one workstation
has only 3 cards, I must limit the number of running VM on a host to 3.

Filter: filters out hosts with number running of vms > 3

Filter Example

oVirt

```
class max_vms():
    '''returns only hosts with less running vms then the maximum'''

    #What are the values this module will accept, used to present
    #the user with options
    properties_validation = 'maximum_vm_count=[0-9]*'

    def do_filter(self, hosts_ids, vm_id, args_map):
        #open a connection to the rest api
        try:
            connection = API(url='http://host:port',
                             username='user@domain', password='')
        except BaseException as ex:
            #letting the external proxy know there was an error
            print >> sys.stderr, ex
            return

        #get our parameters from the map
        maximum_vm_count = int(args_map.get('maximum_vm_count', 100))

        #get all the hosts with the given ids
        engine_hosts = \
            connection.hosts.list(
                query=" or ".join(["id=%s" % u for u in hosts_ids]))

        #iterate over them and decide which to accept
        accepted_host_ids = []
        for engine_host in engine_hosts:
            if(engine_host and
                    engine_host.summary.active < maximum_vm_count):
                accepted_host_ids.append(engine_host.id)
        print accepted_host_ids
```

Filter Example

oVirt

```
class max_vms():
    '''returns only hosts with less running vms then the maximum'''

    #What are the values this module will accept, used to present
    #the user with options
    properties_validation = 'maximum_vm_count=[0-9]*'

    def do_filter(self, hosts_ids, vm_id, args_map):
        #open a connection to the rest api
        try:
            connection = API(url='http://host:port',
                             username='user@domain', password='')
        except BaseException as ex:
            #letting the external proxy know there was an error
            print >> sys.stderr, ex
            return

        #get our parameters from the map
        maximum_vm_count = int(args_map.get('maximum_vm_count', 100))

        #get all the hosts with the given ids
        engine_hosts = \
            connection.hosts.list(
                query=" or ".join(["id=%s" % u for u in hosts_ids]))

        #iterate over them and decide which to accept
        accepted_host_ids = []
        for engine_host in engine_hosts:
            if(engine_host and
                    engine_host.summary.active < maximum_vm_count):
                accepted_host_ids.append(engine_host.id)
        print accepted_host_ids
```

Filter Example

oVirt

```
class max_vms():
    '''returns only hosts with less running vms then the maximum'''

    #What are the values this module will accept, used to present
    #the user with options
    properties_validation = 'maximum_vm_count=[0-9]*'

    def do_filter(self, hosts_ids, vm_id, args_map):
        #open a connection to the rest api
        try:
            connection = API(url='http://host:port',
                             username='user@domain', password='')
        except BaseException as ex:
            #letting the external proxy know there was an error
            print >> sys.stderr, ex
            return

        #get our parameters from the map
        maximum_vm_count = int(args_map.get('maximum_vm_count', 100))

        #get all the hosts with the given ids
        engine_hosts = \
            connection.hosts.list(
                query=" or ".join(["id=%s" % u for u in hosts_ids]))

        #iterate over them and decide which to accept
        accepted_host_ids = []
        for engine_host in engine_hosts:
            if(engine_host and
                    engine_host.summary.active < maximum_vm_count):
                accepted_host_ids.append(engine_host.id)
        print accepted_host_ids
```

Filter Example

oVirt

```
class max_vms():
    '''returns only hosts with less running vms then the maximum'''

    #What are the values this module will accept, used to present
    #the user with options
    properties_validation = 'maximum_vm_count=[0-9]*'

    def do_filter(self, hosts_ids, vm_id, args_map):
        #open a connection to the rest api
        try:
            connection = API(url='http://host:port',
                             username='user@domain', password='')
        except BaseException as ex:
            #letting the external proxy know there was an error
            print >> sys.stderr, ex
            return

        #get our parameters from the map
        maximum_vm_count = int(args_map.get('maximum_vm_count', 100))

        #get all the hosts with the given ids
        engine_hosts = \
            connection.hosts.list(
                query=" or ".join(["id=%s" % u for u in hosts_ids]))

        #iterate over them and decide which to accept
        accepted_host_ids = []
        for engine_host in engine_hosts:
            if(engine_host and
                    engine_host.summary.active < maximum_vm_count):
                accepted_host_ids.append(engine_host.id)
        print accepted_host_ids
```

Filter Example

oVirt

```
class max_vms():
    '''returns only hosts with less running vms then the maximum'''

    #What are the values this module will accept, used to present
    #the user with options
    properties_validation = 'maximum_vm_count=[0-9]*'

    def do_filter(self, hosts_ids, vm_id, args_map):
        #open a connection to the rest api
        try:
            connection = API(url='http://host:port',
                             username='user@domain', password='')
        except BaseException as ex:
            #letting the external proxy know there was an error
            print >> sys.stderr, ex
            return

        #get our parameters from the map
        maximum_vm_count = int(args_map.get('maximum_vm_count', 100))

        #get all the hosts with the given ids
        engine_hosts = \
            connection.hosts.list(
                query=" or ".join(["id=%s" % u for u in hosts_ids]))

        #iterate over them and decide which to accept
        accepted_host_ids = []
        for engine_host in engine_hosts:
            if(engine_host and
                    engine_host.summary.active < maximum_vm_count):
                accepted_host_ids.append(engine_host.id)
        print accepted_host_ids
```

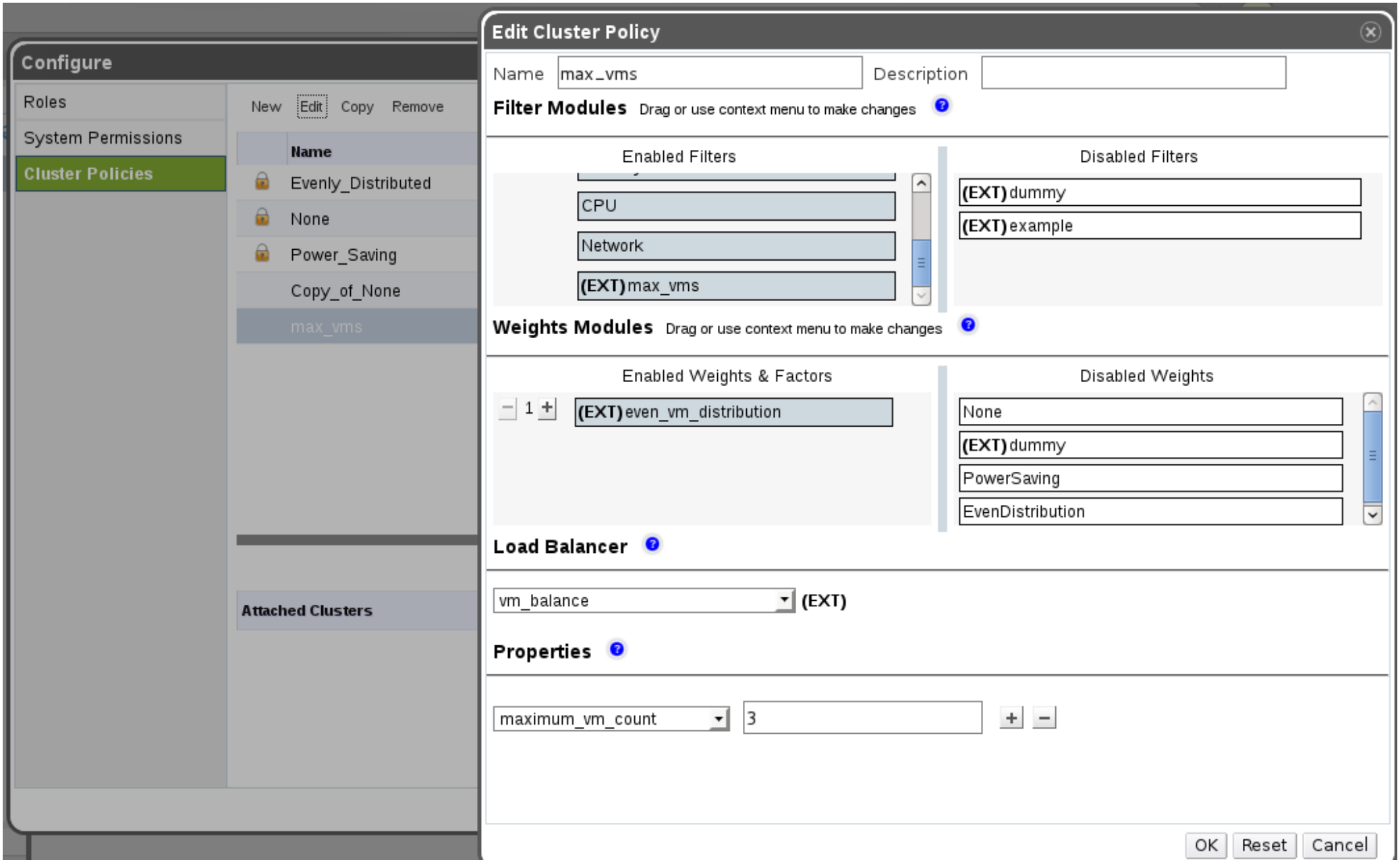
External Policy Units

oVirt

- Scheduler proxy is scanning directory /usr/share/ovirt-scheduler-proxy/plugins for python source files
- Analyze for filter / weight / balance functions
- Cache results
- Expose source files as external policy units

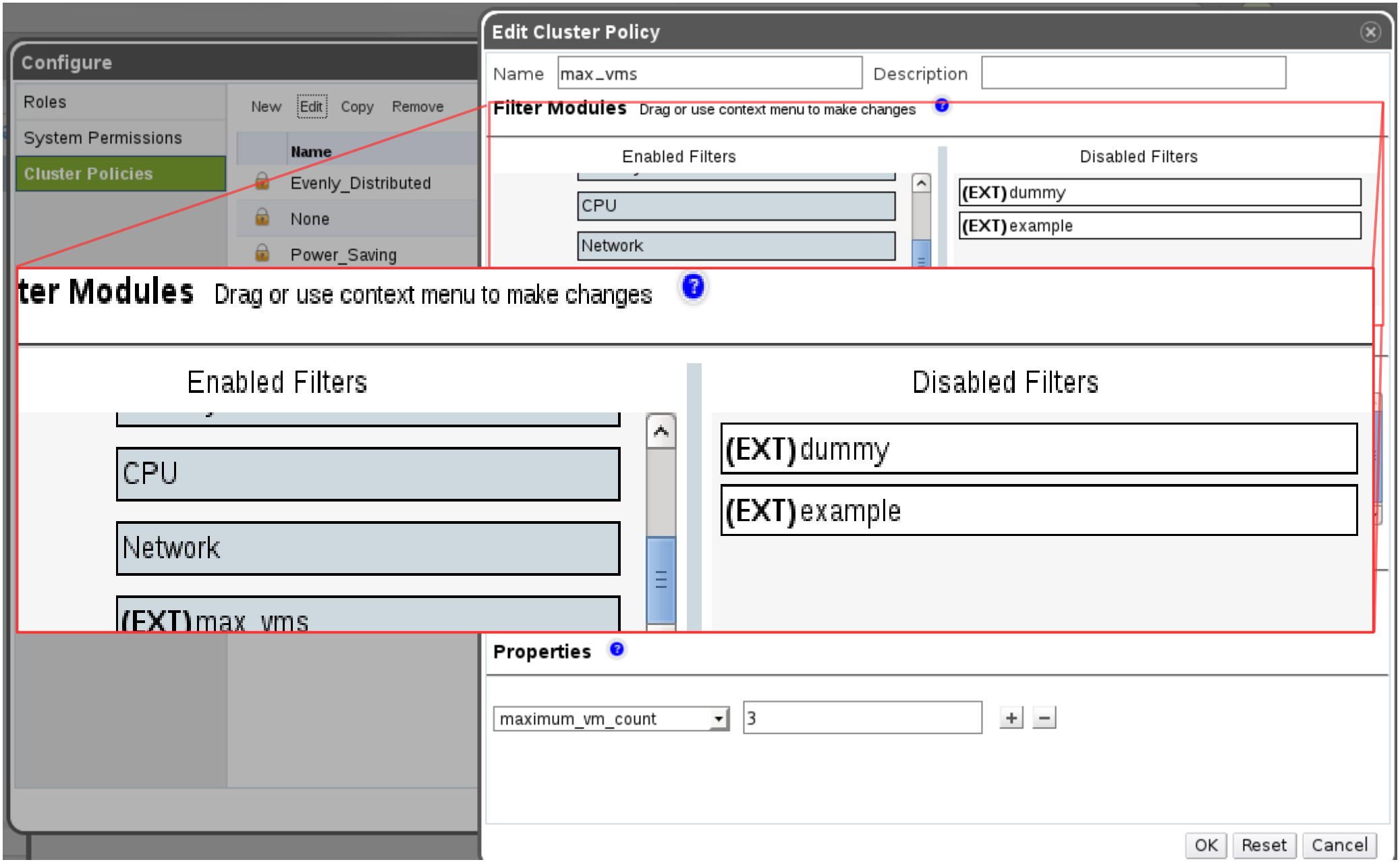
Cluster Policy Management

oVirt



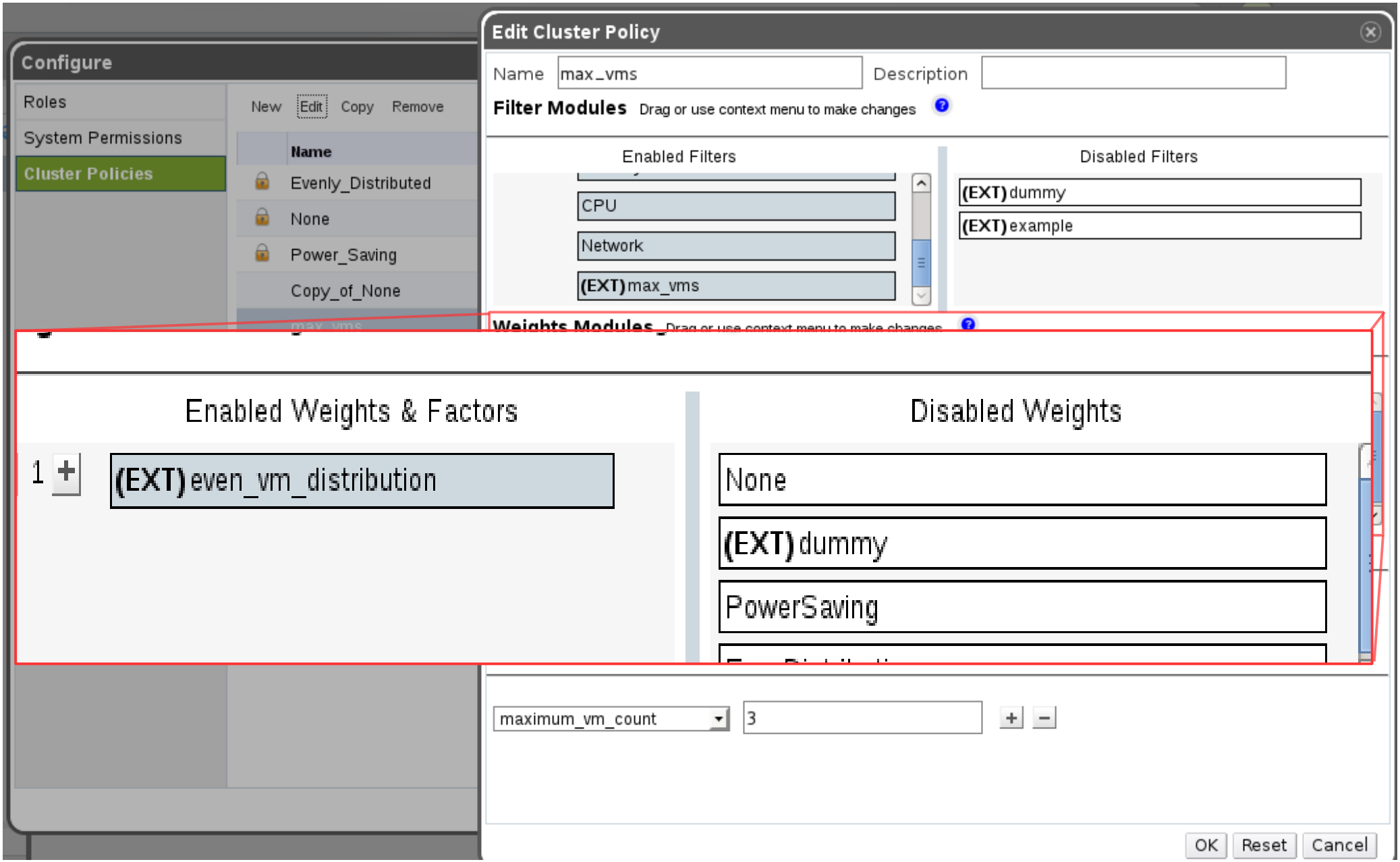
Cluster Policy Management

oVirt



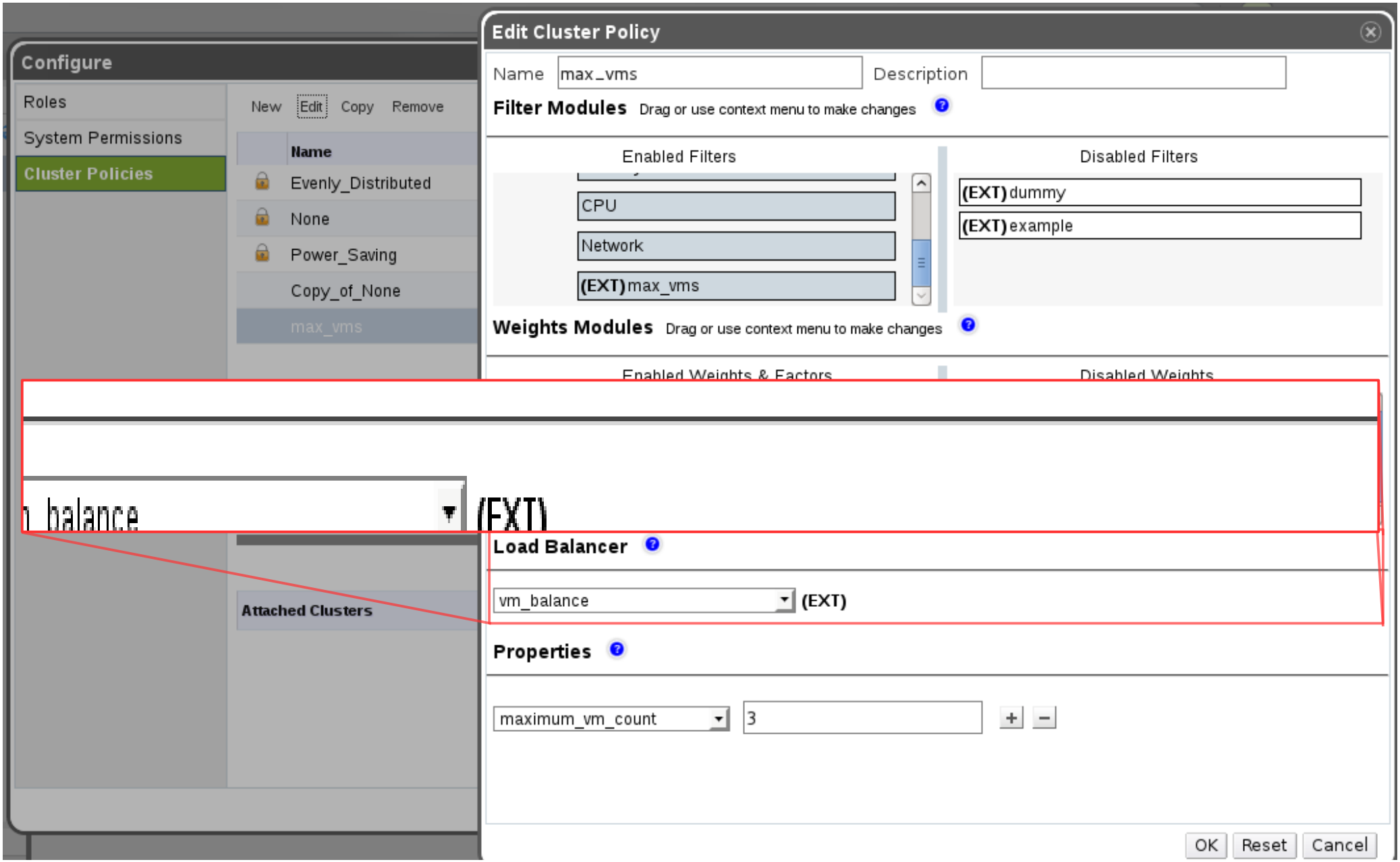
Cluster Policy Management

oVirt



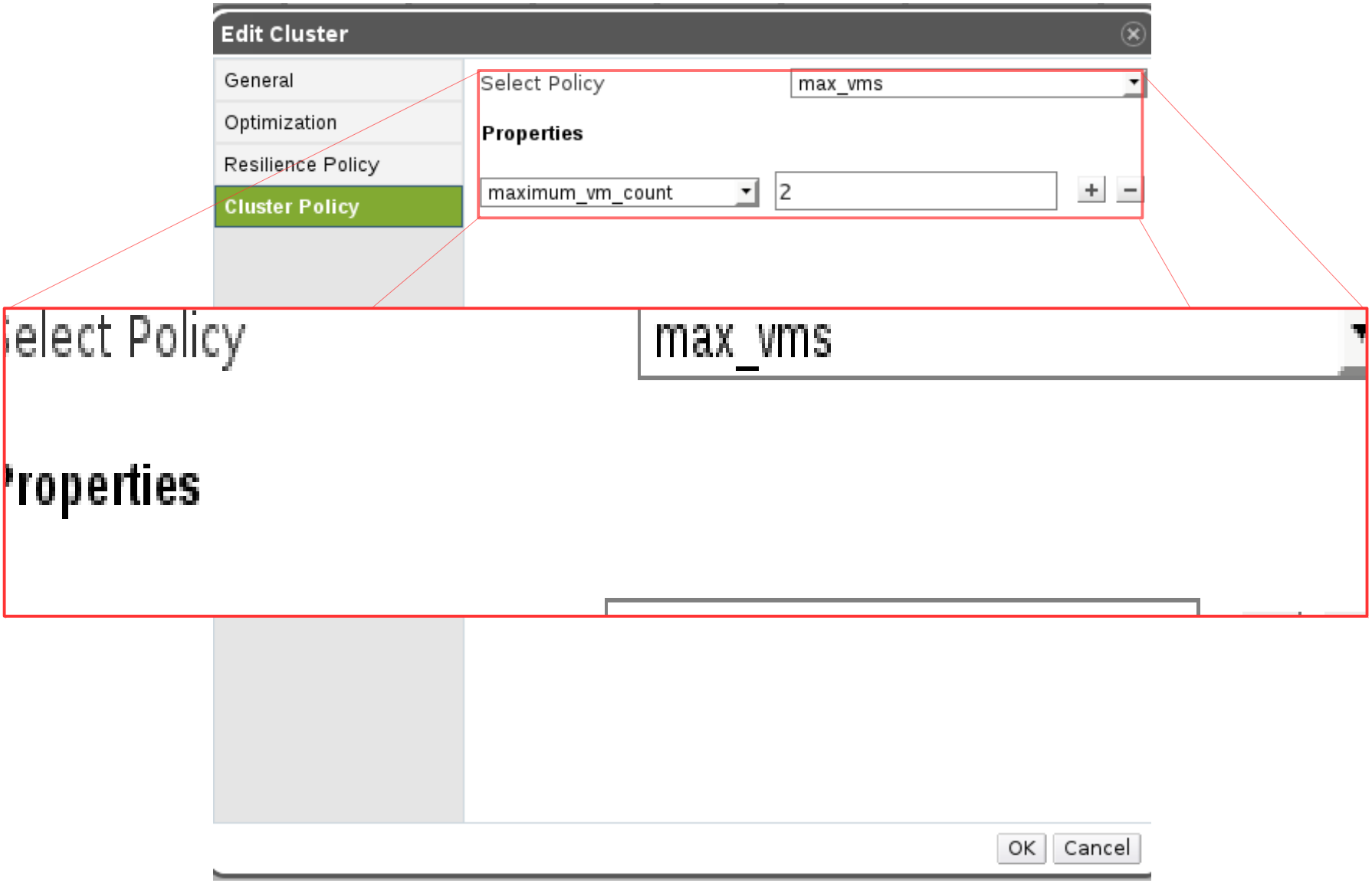
Cluster Policy Management

oVirt



Apply Cluster Policy

oVirt

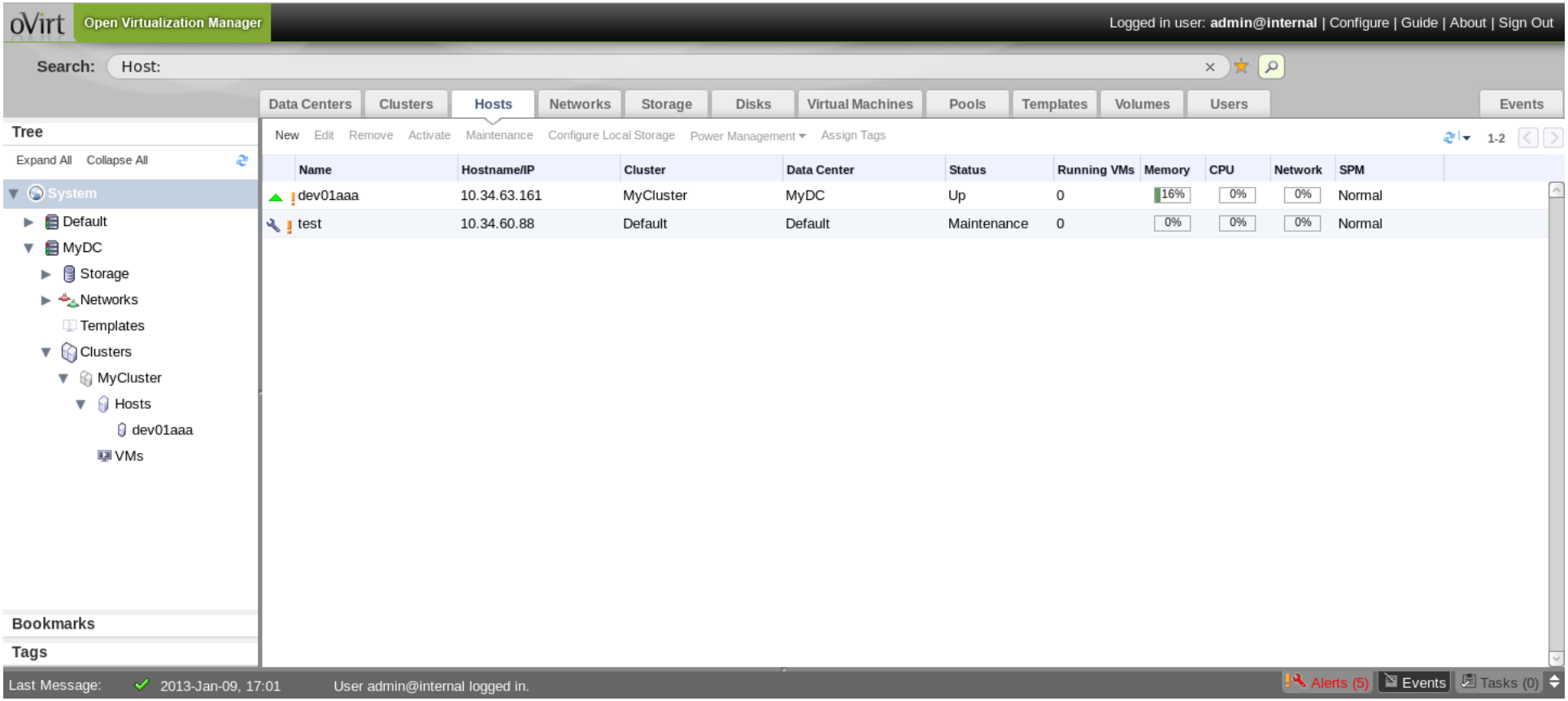


UI Plugins

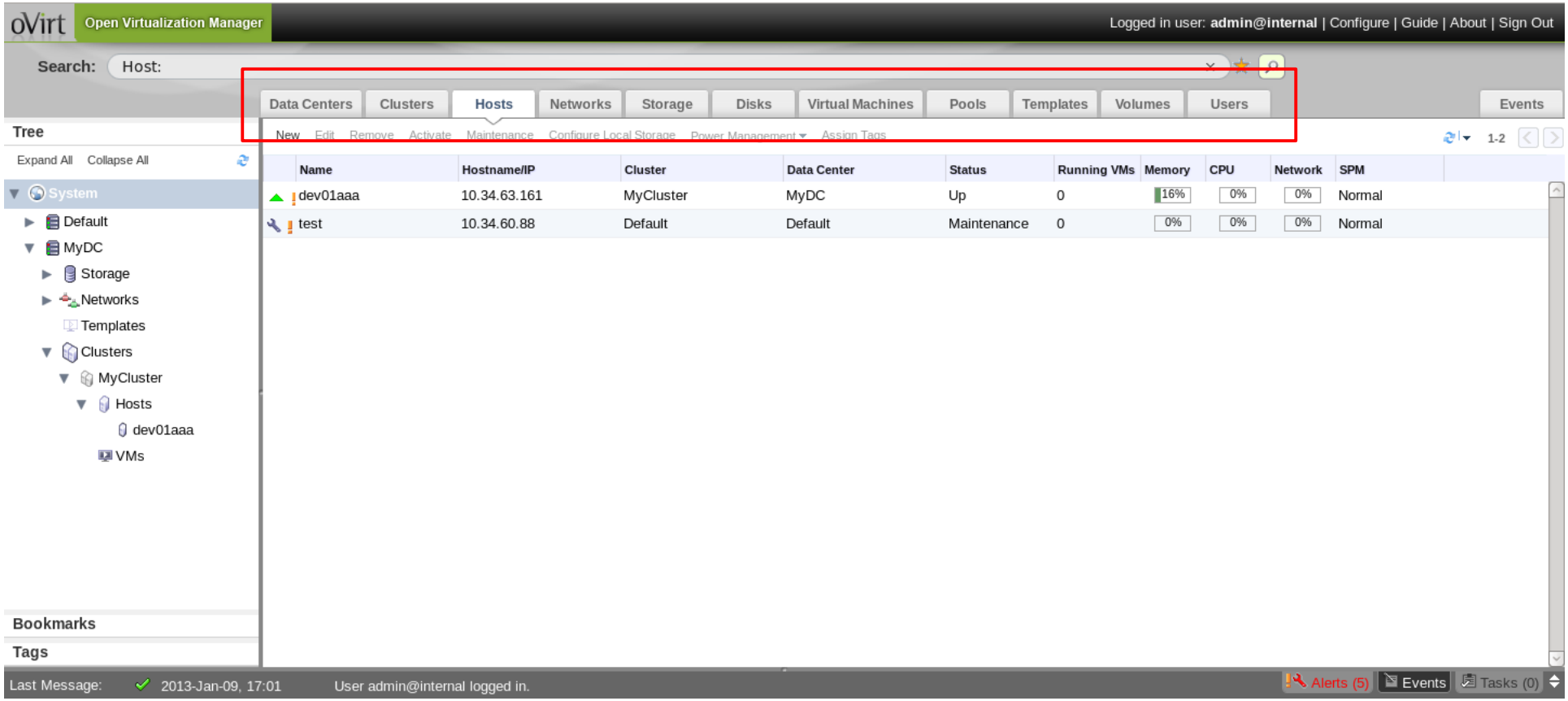
Web Admin user interface

oVirt

Extend oVirt Web Admin user interface

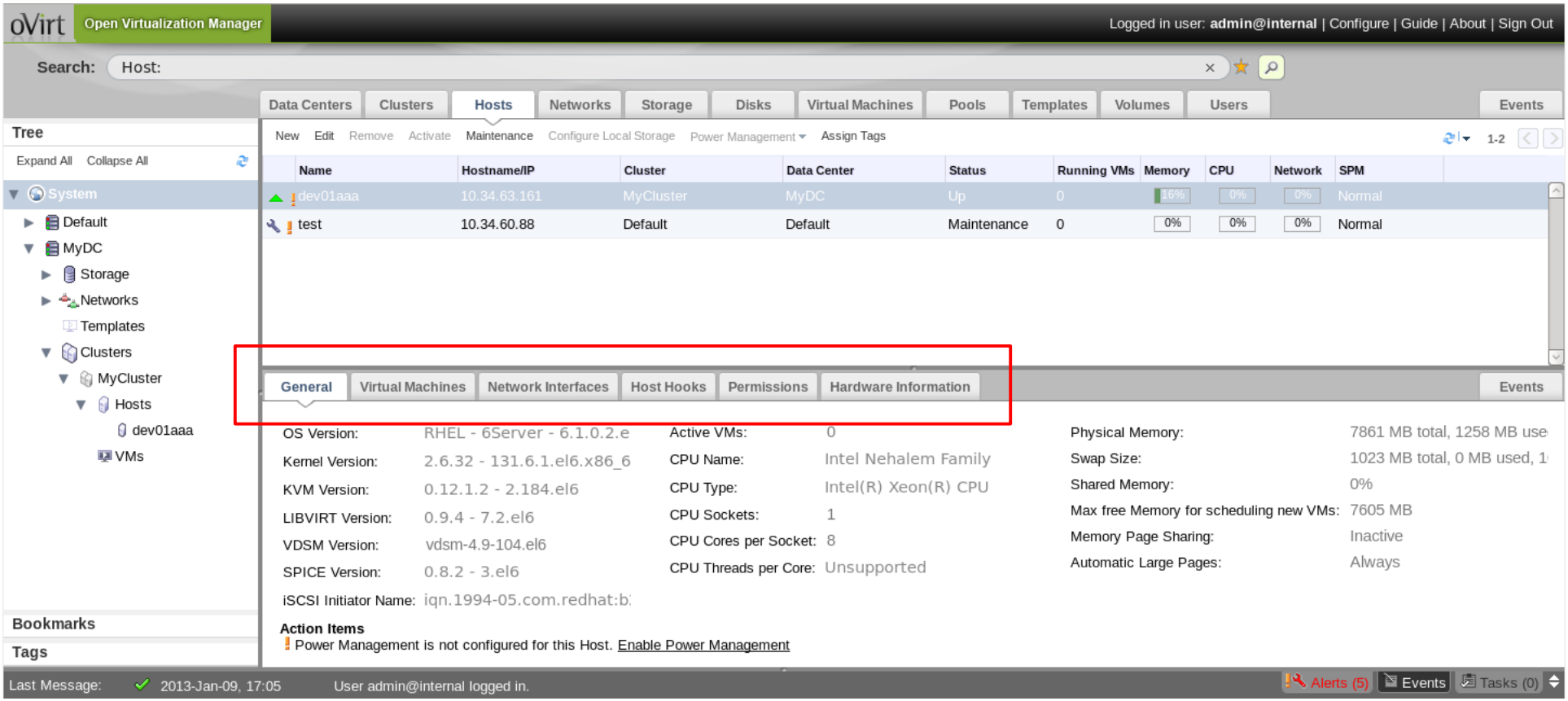


Web Admin user interface

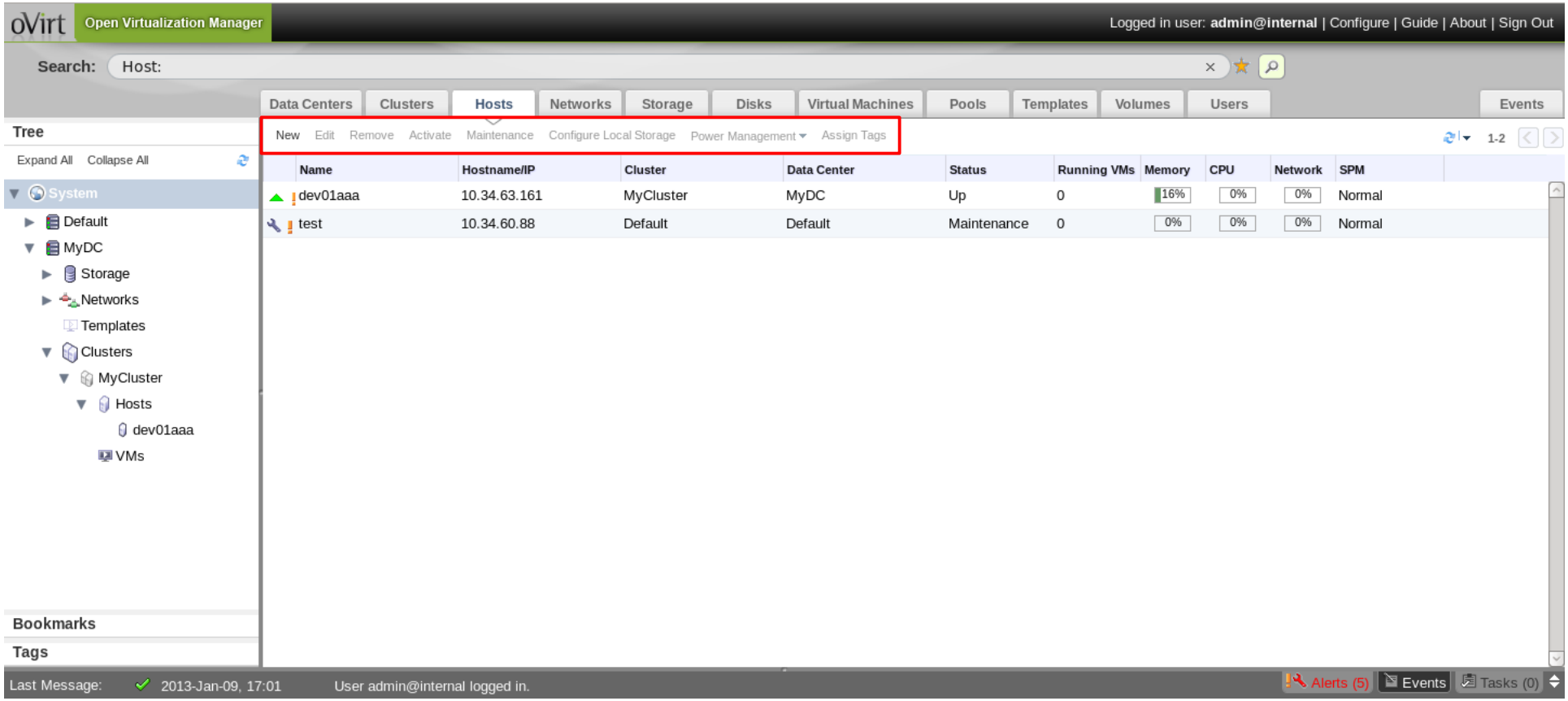


Web Admin user interface

oVirt

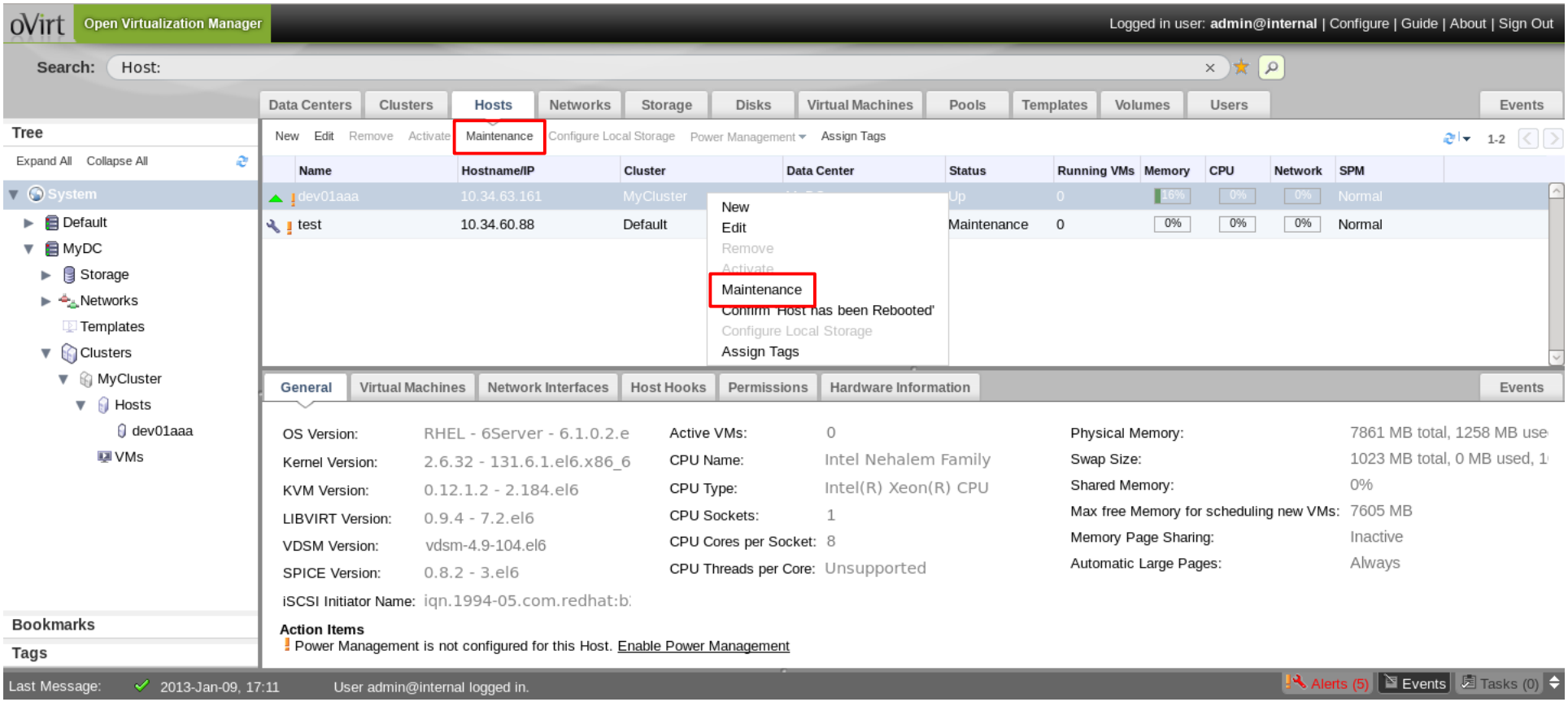


Web Admin user interface



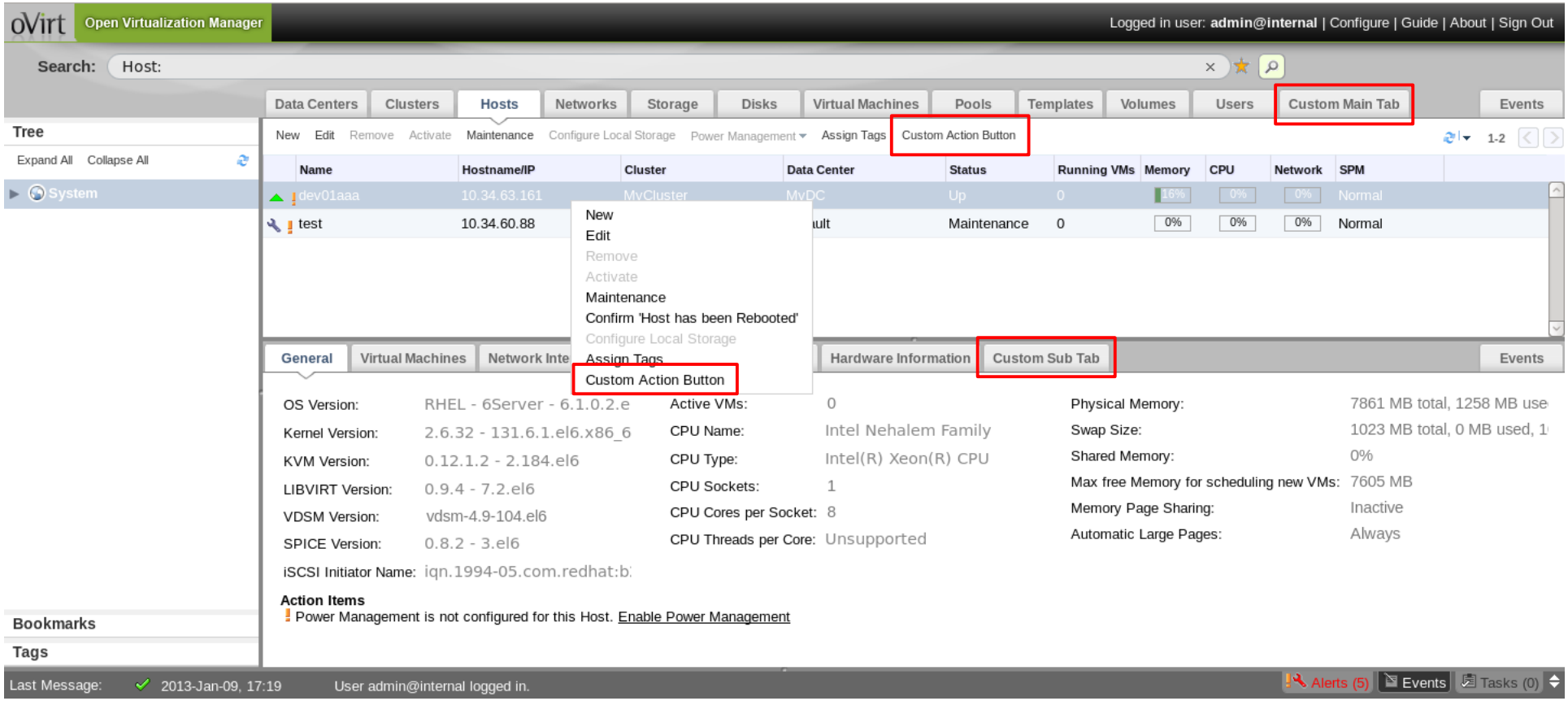
Web Admin user interface

oVirt



What's currently possible

oVirt



UI plugin basics

oVirt

- Plugin host page
  - Hosts actual plugin code (JavaScript)
    `/usr/share/ovirt-engine/ui-plugins/<resourcePath>/<hostPage>.html`
- Plugin descriptor
  - Meta-data + default configuration
    `/usr/share/ovirt-engine/ui-plugins/<descriptorName>.json`
- Plugin user configuration
  - Override default configuration, tweak runtime behavior
    `/etc/ovirt-engine/ui-plugins/<descriptorName>-config.json`

Writing UI plugins

oVirt

```
<!DOCTYPE html>
<html>
<head>
    <!-- Fetch additional resources if necessary -->
    <script type="text/javascript" src="jquery-min.js"></script>

    <!-- Actual plugin code -->
    <script>
        // Access plugin API from iframe context
        var api = parent.pluginApi('myPlugin');

        // Register plugin event handler functions
        api.register({
            UiInit: function() {
                api.addMainTab('Foo Tab', 'foo-tab', 'http://foo.com/');
            }
        });

        // Tell plugin infrastructure that we are ready
        api.ready();
    </script>

</head>
<body> <!-- HTML body is intentionally empty --> </body>
</html>
```

Plugin descriptor

oVirt

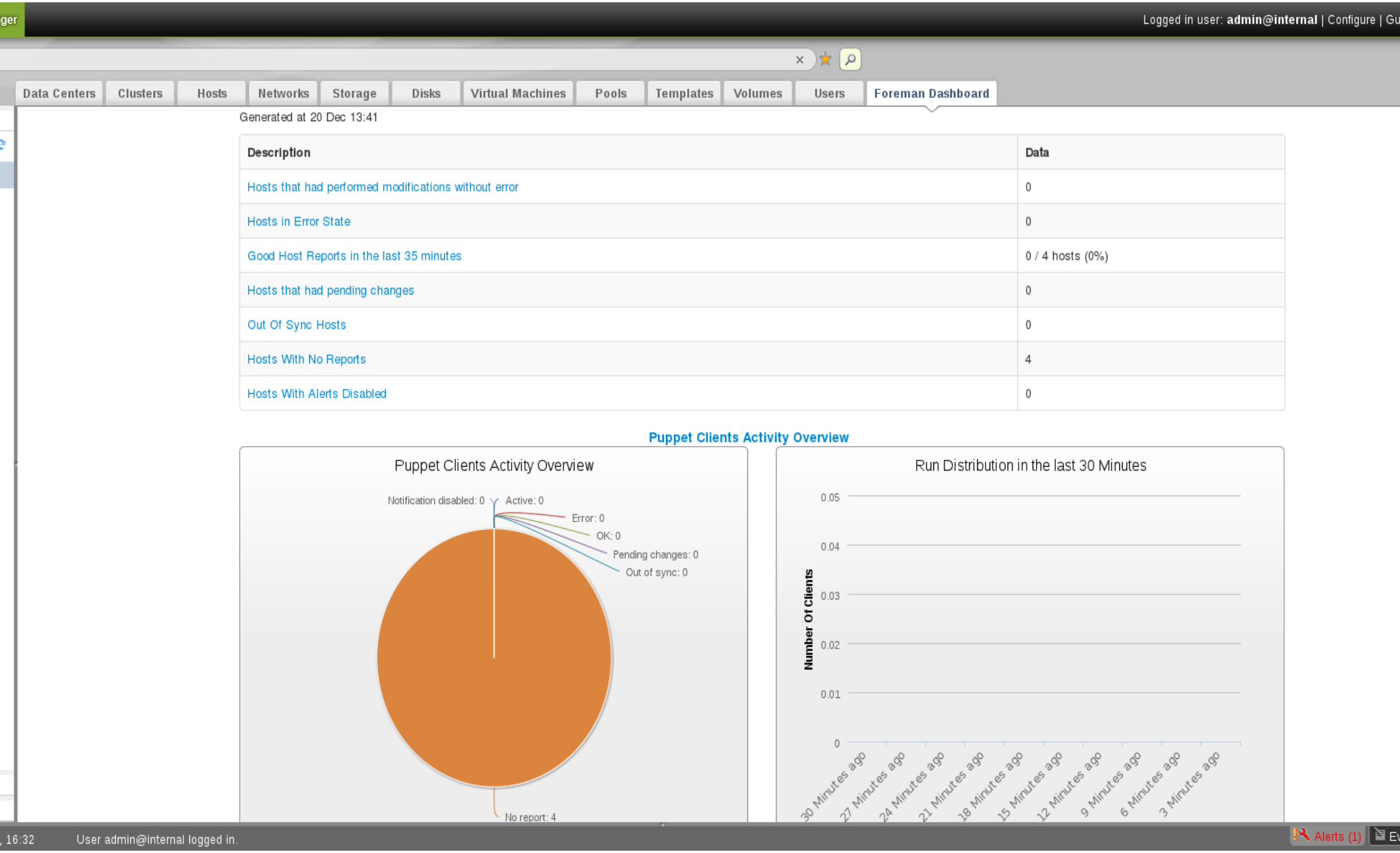
Meta-data + default configuration

/usr/share/ovirt-engine/ui-plugins/<descriptorName>.json

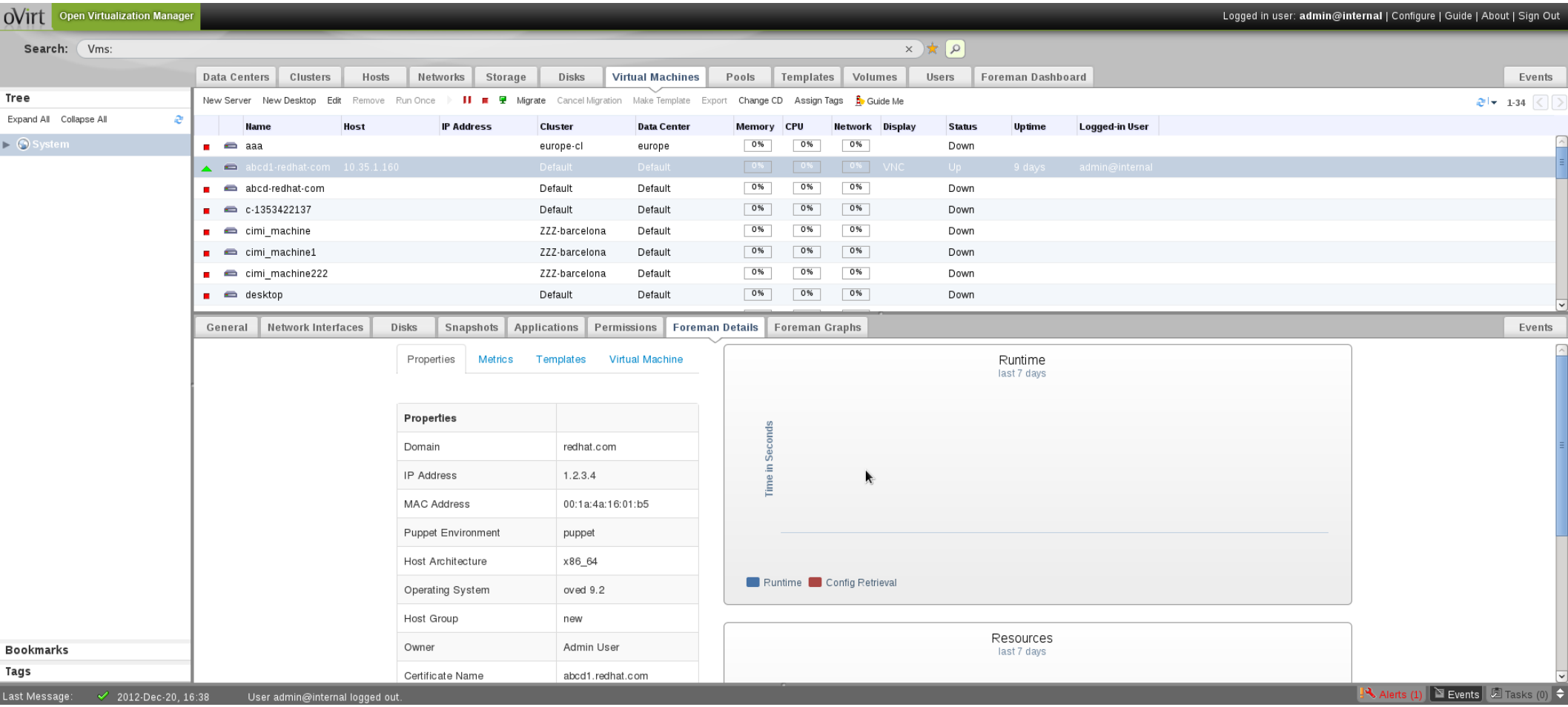
```
{

    // A name that uniquely identifies the plugin (required)
    "name": "foo",

    // URL of plugin host page that invokes the plugin code (required)
    "url": "/webadmin/webadmin/plugin/foo/start.html",

    // Default configuration object associated with the plugin (optional)
    "config": { "band": "ZZ Top", "classic": true, "score": 10 },

    // Path to plugin static resources (optional)
    // Used when serving plugin files through PluginResourceServlet
    // This path is relative to /usr/share/ovirt-engine/ui-plugins
    "resourcePath": "foo-files"

}
```

UI-Plugin: Foreman main tab

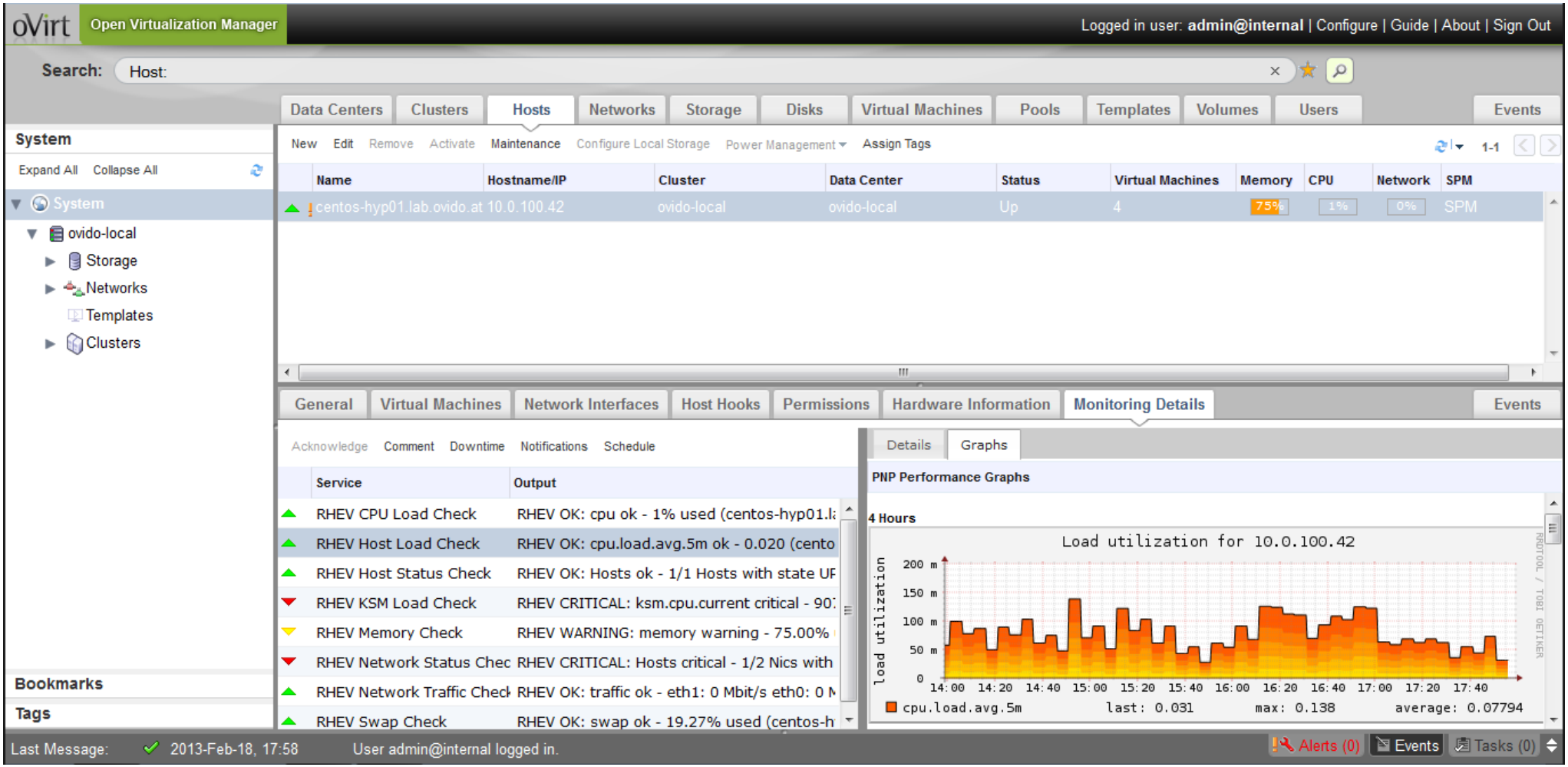
oVirt



UI-Plugin: VM Foreman Details



UI-Plugin: oVirt Monitoring sub-tab (Nagios)

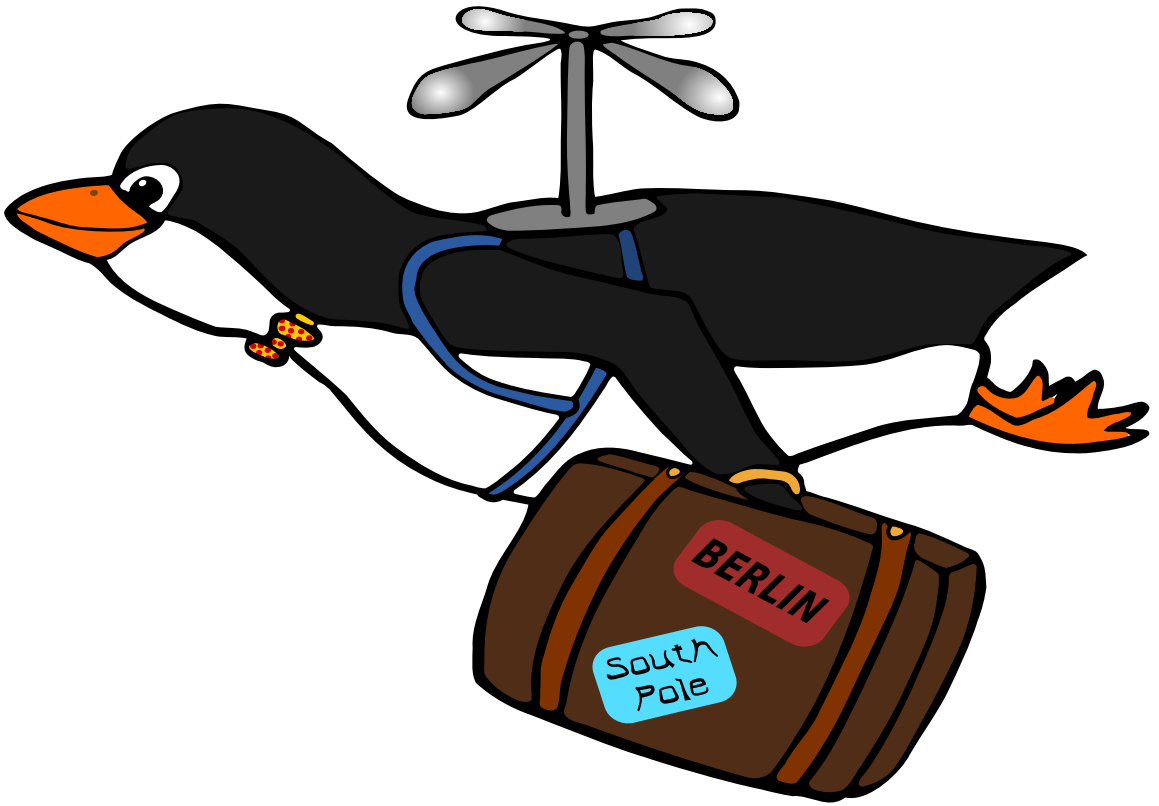


More info at: https://labs.ovido.at/monitoring/wiki/ovirt-monitoring-ui-plugin

UI Plugins

oVirt

Scary demo....



Summary

oVirt

- Consuming oVirt
  - oVirt API
  - oVirt SDK
  - SNMP

- Extending oVirt
  - VDSM hooks
  - Scheduling API
  - UI Plugins

Questions?

THANK YOU!

http://www.ovirt.org
http://www.ovirt.org/Category:SLA

http://lists.ovirt.org/mailman/listinfo
vdsm-devel@lists.fedorahosted.org

#ovirt irc.oftc.net

doron@redhat.com