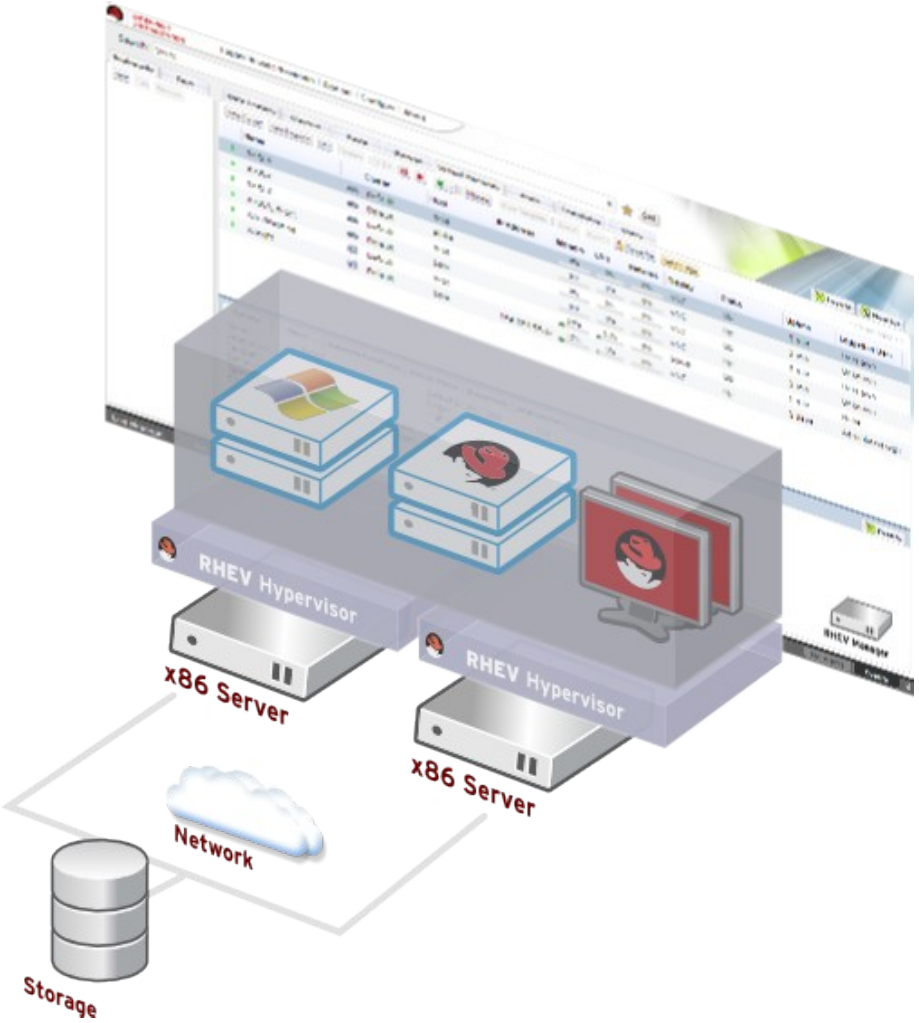
oVirt Architecture

Itamar Heim
Director, RHEV-M Engineering, Red Hat

oVirt Engine

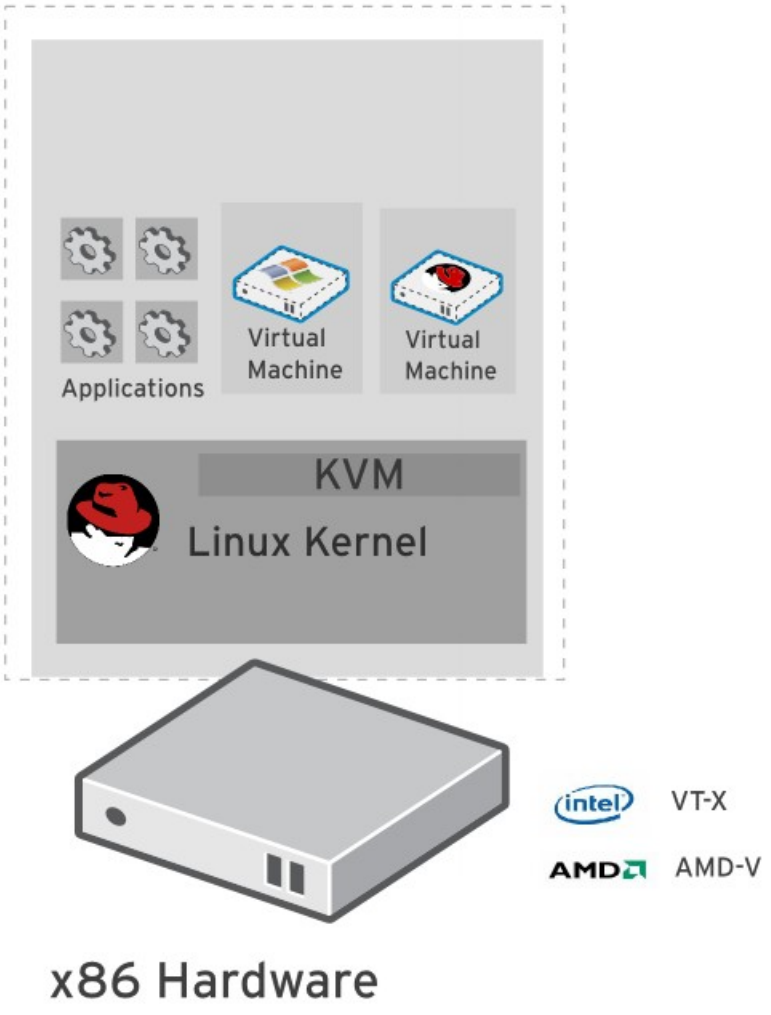
Large scale, centralized management for server and desktop virtualization

Based on leading performance, scalability and security infrastructure technologies



Kenrel-based Virtual Machine (KVM)

- Included in Linux kernel since 2006
- Runs Linux, Windows and other operating system guests
- Advanced features
  - Live migration
  - Memory page sharing
  - Thin provisioning
  - PCI Pass-through
- KVM architecture provides high “feature-velocity” – leverages the power of Linux

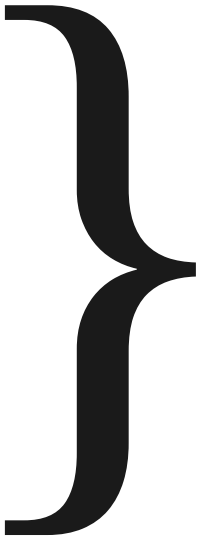


Linux as a Hypervisor?

- What makes up a hypervisor ?
  - Hardware management
  - Device drivers
  - I/O Stack
  - Resource Management
  - Scheduling
  - Access Control
  - Power Management
  - Memory Manager
  - Device Model (emulation)
  - Virtual Machine Monitor

Linux as a Hypervisor?

- What makes up a hypervisor ?
  - Hardware management
  - Device drivers
  - I/O Stack
  - Resource Management
  - Scheduling
  - Access Control
  - Power Management
  - Memory Manager
  - Device Model (emulation)
  - Virtual Machine Monitor



Operating System Kernel

Linux as a Hypervisor?

How well does Linux perform as a hypervisor?
Isn't Linux a general purpose operating system?

Linux is architected to scale from the smallest embedded systems through to the largest multi-socket servers

- From cell phones through to mainframes

KVM benefits from mature, time tested infrastructure

- Powerful, scalable memory manager
- Robust security infrastructure
- High performance network stack
- Versatile storage infrastructure – iSCSI, FC, NAS, multipath, etc
- Rich ecosystem of supported hardware systems

Linux as a Hypervisor?

oVirt

How well does Linux perform as a hypervisor?
Isn't Linux a general purpose operating system?

Over the last 4 years features have been added to Linux to provide a
better infrastructure for a hypervisor

- Scheduler enhancements
  Improved scalability and reduced latency
- Enhancements to memory manager
  Advanced features such as memory page sharing and compression
- Improvements to Block I/O subsystem
  - Better performance, automated alignment, etc

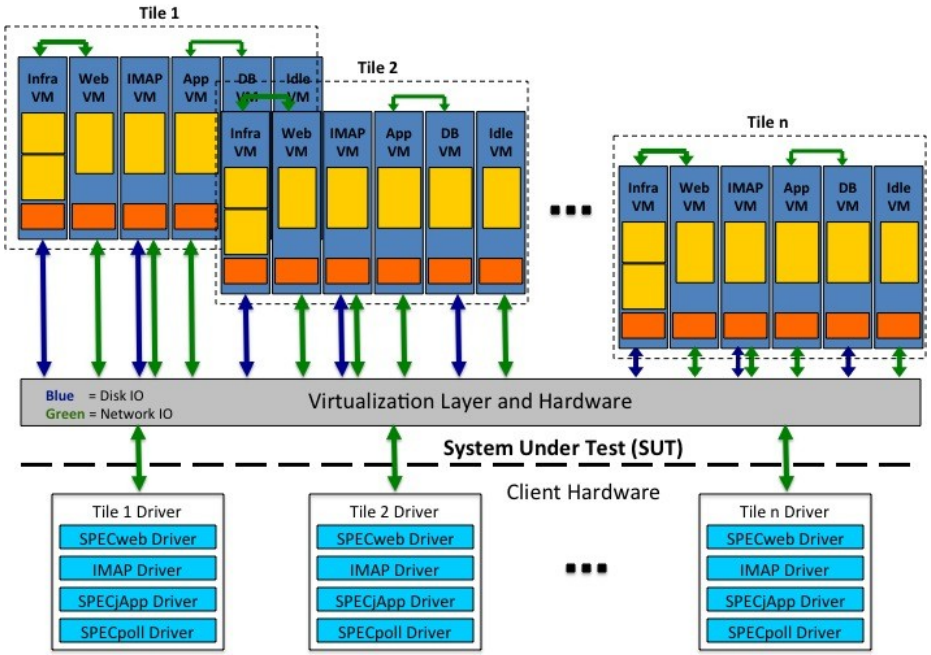
Red Hat Enterprise Virtualization
Performance and Scalability

SPECvirt_sc2010

Vendor neutral virtualizaztion benchmarks
Comprised of application specific benchmarks running inside “tiles”

Each tile runs 6 virtual machines

- Application Server
- Database Server
- Mail Server
- Web Server
- Infrastructure Server
- Idle Server



Each VM runs a benchmark, eg SpecWeb, SPECjAppServer,SPECmail and must meet specifi

Red Hat Enterprise Virtualization
Performance and Scalability

SPECvirt_sc2010

KVM leads the pack in 2, 4, 8 socket systems for SPECvirt

Including the largest benchmark results with over 400 Vms

Score : 7067 @ **432 VMs** (72 tiles)

Processor: Intel Xeon E7-4870 (80 cores, 8 chips, 10 cores/chip, 2 threads/core)
Memory: 2 TB (128 x 16 GB, Quad Rank x4 PC3-8500 CL7 ECC DDR3 1066MHz LP RDIMM)

http://www.spec.org/virt_sc2010/

Red Hat Enterprise Virtualization
Competitive Landscape

Test Center Scorecard	Management	Performance	Reliability	Scalability	Installation	Overall Score
	25%	20%	20%	20%	15%	
Citrix XenServer 5.6.1	7	8	8	7	9	7.7 GOOD
	25%	20%	20%	20%	15%	
Microsoft Windows Server 2008 R2 Hyper-V	8	8	9	8	7	8.1 VERY GOOD
	25%	20%	20%	20%	15%	
Red Hat Enterprise Virtualization for Servers 2.2	8	8	8	9	9	8.4 VERY GOOD
	25%	20%	20%	20%	15%	
VMware vSphere 4.1	9	9	9	9	9	9.0 EXCELLENT

- InfoWorld “shootout” 2011
  - Independent analysis of leading virtualization platforms
  - After <18 months Red Hat has overtaken Citrix & Microsoft in performance and functionality

http://bit.ly/virtshootout

Security

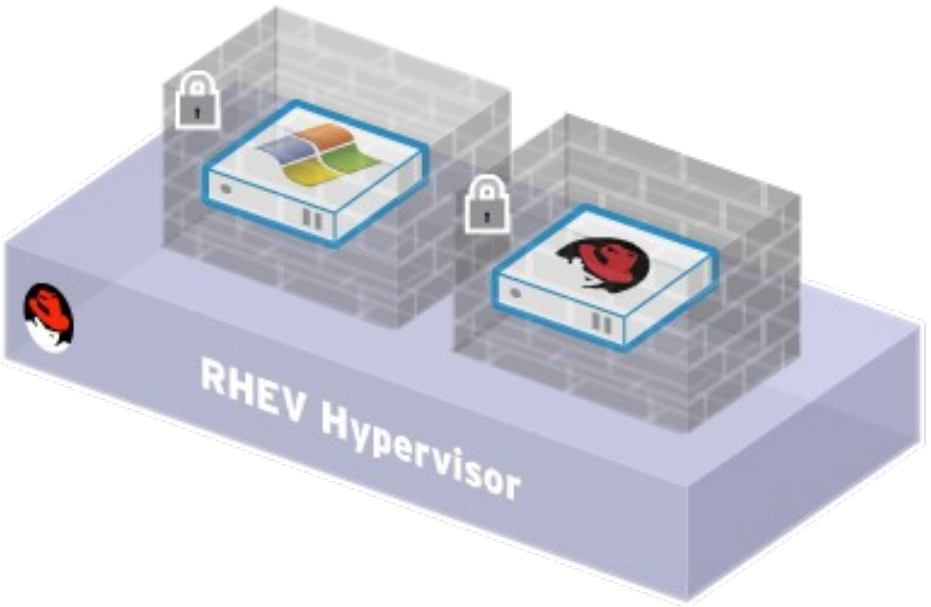
oVirt inherits the security features of Linux

SELinux security policy infrastructure

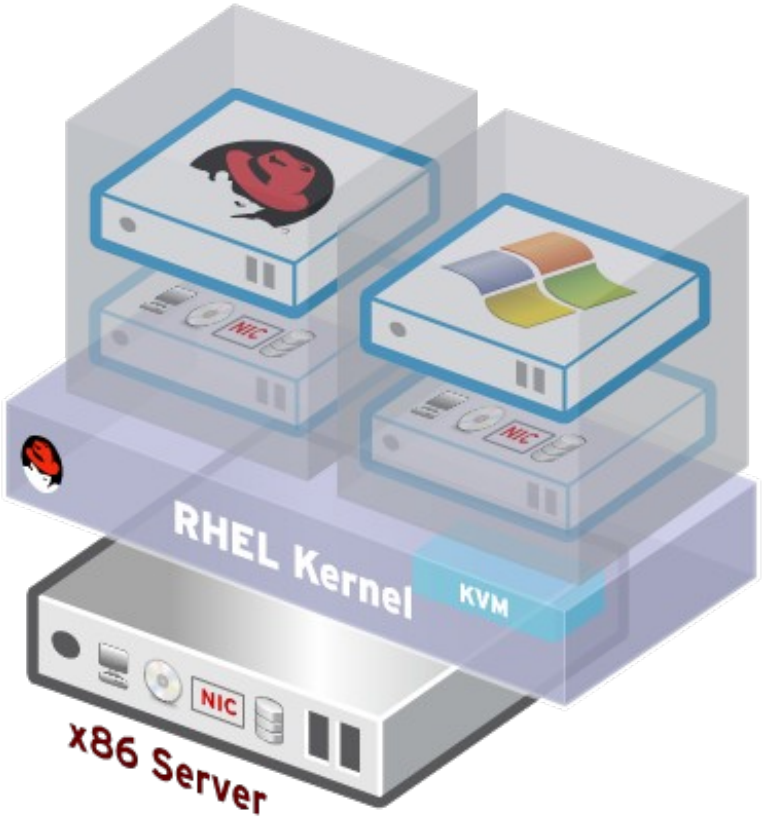
- Provides protection and isolation for virtual machines and host
- Compromised virtual machine cannot access other VMs or host

sVirt Project

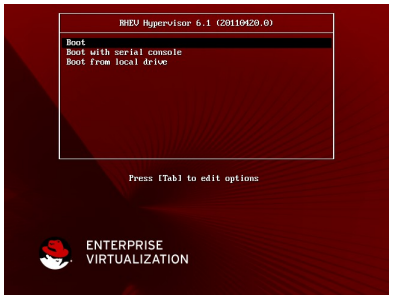
- Sub-project of NSA's SELinux community. Provides “hardened” hypervisors
- Multilevel security. Isolate guests
- Contain any hypervisor breaches

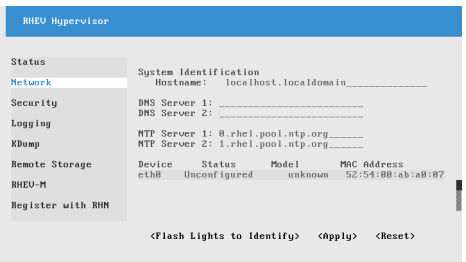


oVirt Node



- Standalone hypervisor
  - Small footprint < 100MB
    - Customized 'spin' of Fedora + KVM
    - 'Just enough' Fedora to run virtual machines
    - Runs on all RHEL hardware with Intel VT/AMD-V CPUs
  - Easy to install, configure and upgrade
    - PXE boot, USB boot, CD or Hard drive





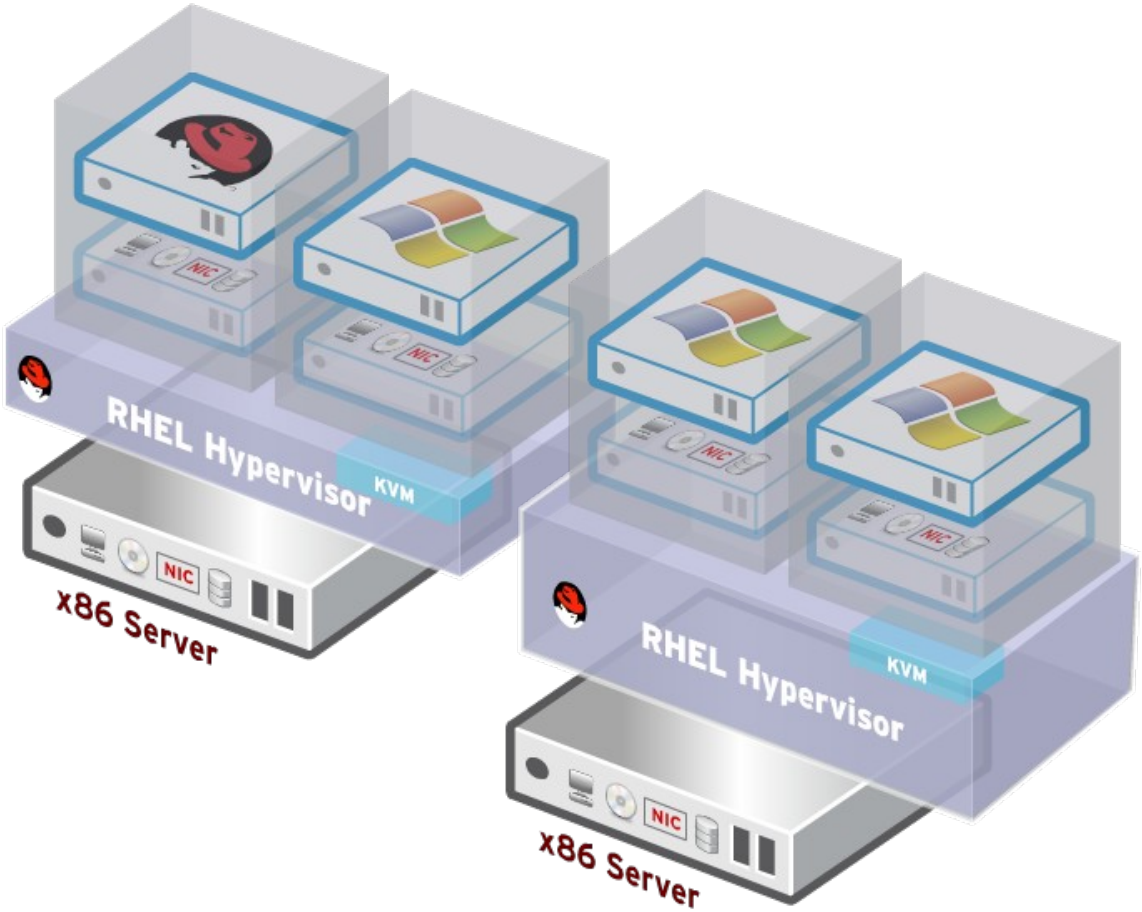
oVirt Node vs. Full Host

oVirt Node

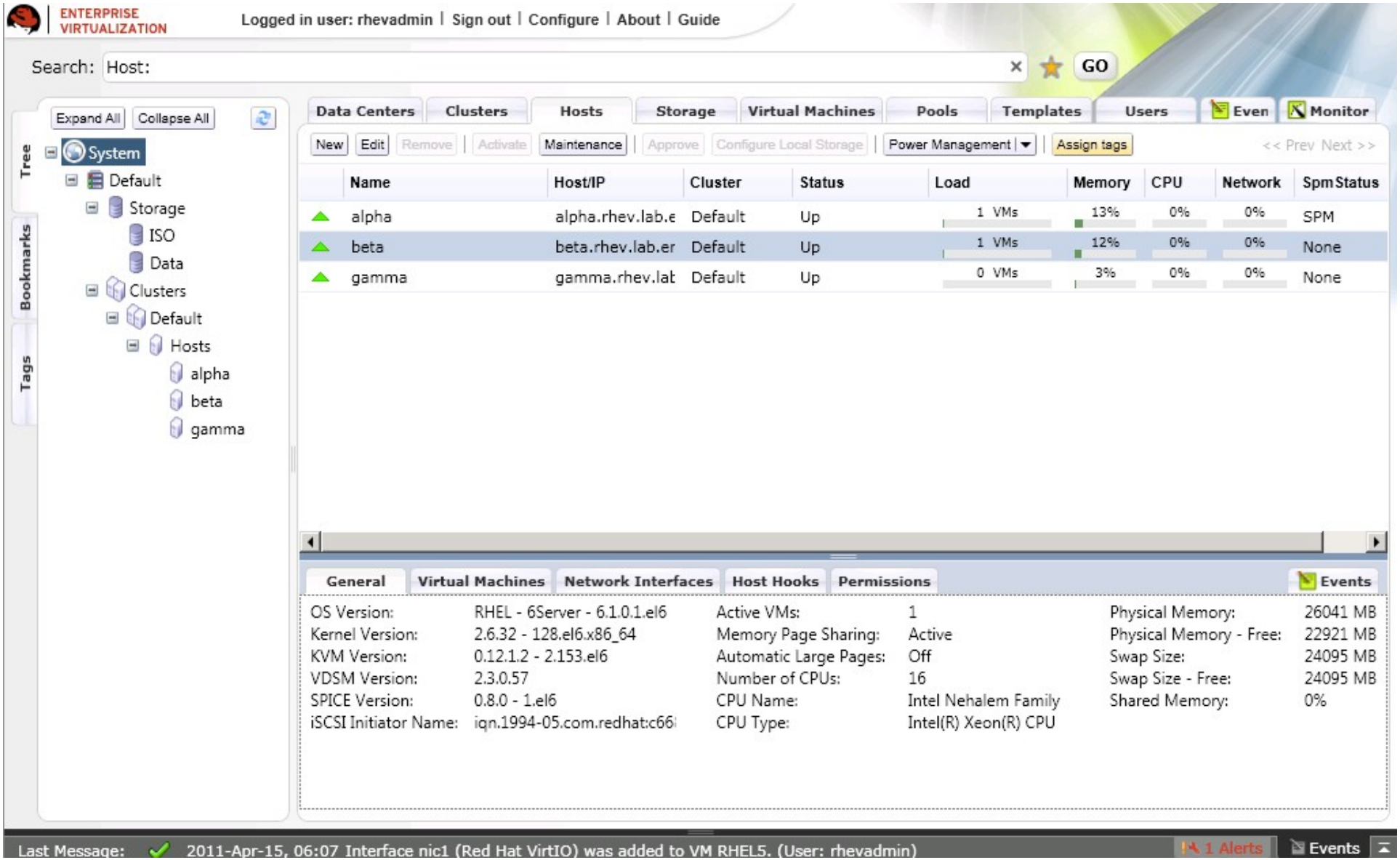
- Less than 100 MB
- Pre-configured, no Linux skills needed.

Full Host

- Flexible
- Add monitoring agents, scripts etc. Leverage existing Fedora infrastructure.
- Hybrid mode capable



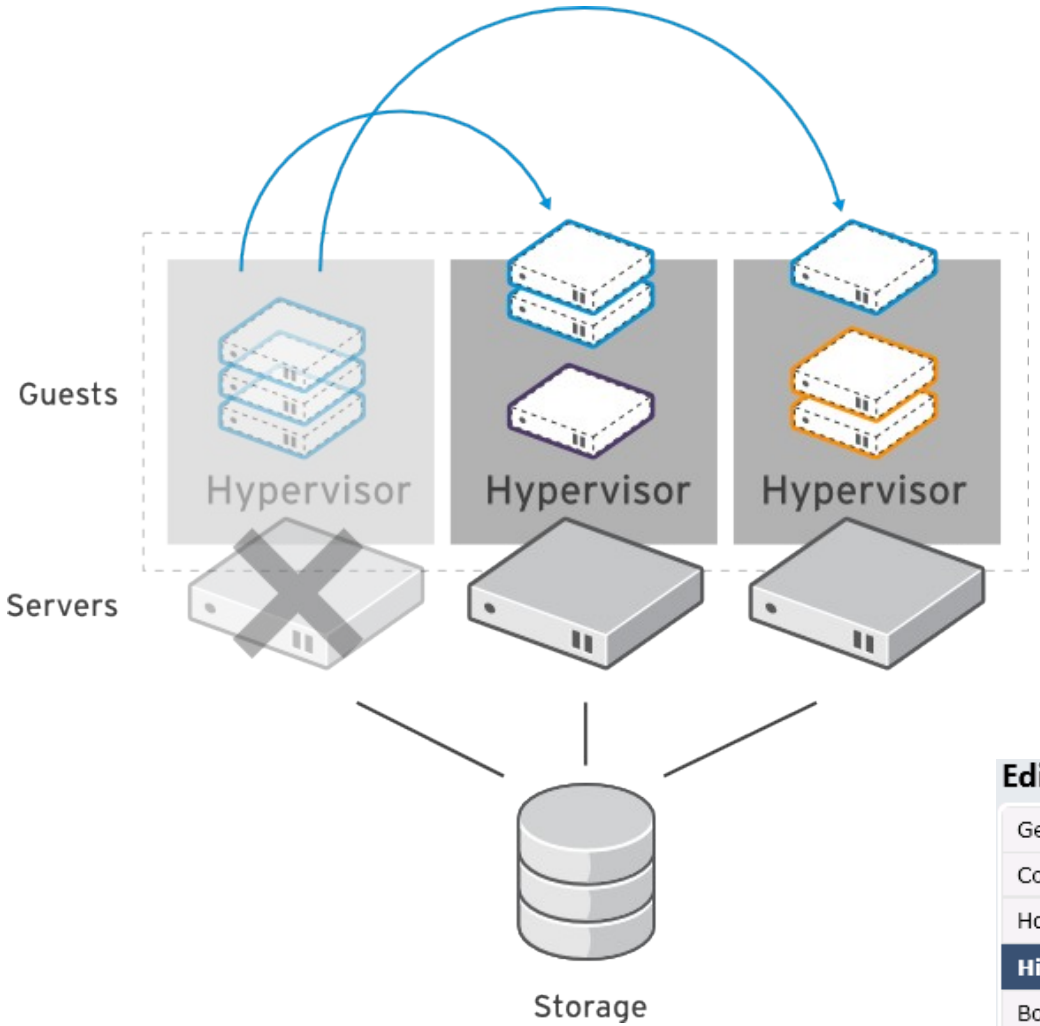
Ovirt Engine



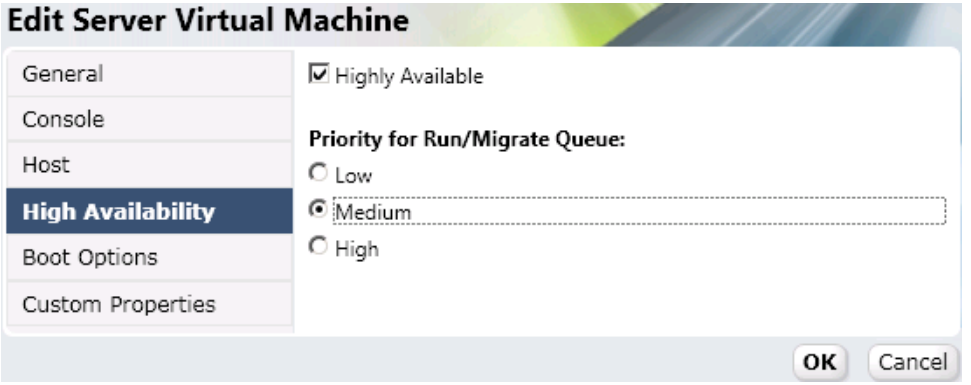
Management Features

Feature	Description
High Availability	Restart guest VMs from failed hosts automatically on other hosts
Live Migration	Move running VM between hosts with zero downtime
System Scheduler	Continuously load balance VMs based on resource usage/policies
Power Saver	Concentrate virtual machines on fewer servers during off-peak hours
Maintenance Manager	No downtime for virtual machines during planned maintenance windows. Hypervisor patching
Image Management	Template based provisioning, thin provisioning and snapshots
Monitoring & Reporting	For all objects in system – VM guests, hosts, networking, storage etc.
OVF Import/Export	Import and export VMs and templates using OVF files
V2V	Convert VMs from VMware and RHEL/Xen to RHEV

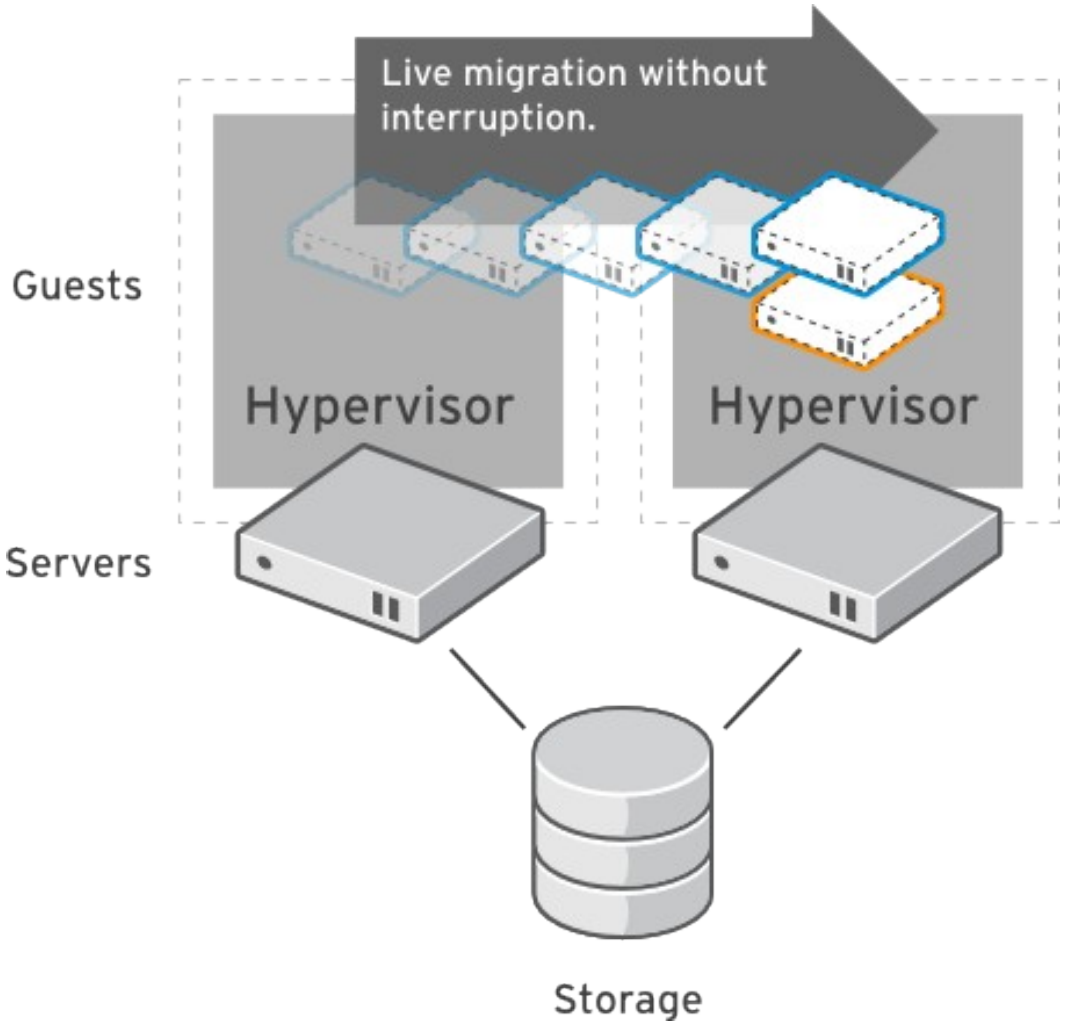
High Availability



- Build a highly available enterprise infrastructure
- Continually monitor host systems and virtual machines
- Automatically restart virtual machines in case of host failure
  - Restart virtual machine on another node in the cluster
- Use live migration to “fail-back” a VM to it's original host when the server is restored

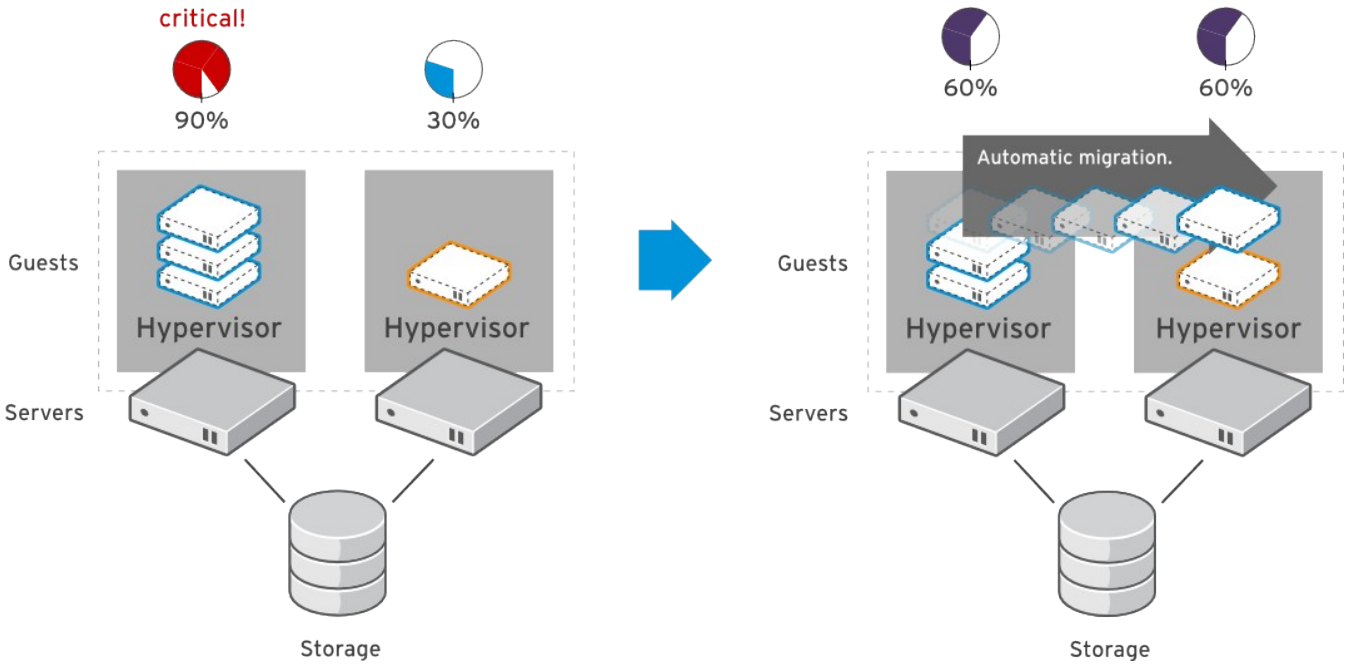


Live Migration



- Dynamically move virtual machines between hosts
  - No service interruption
  - Applications continue to run
- Migrate even I/O intensive workloads such as databases
- Perform hardware maintenance without application downtime
- Dynamically balance workloads between host systems

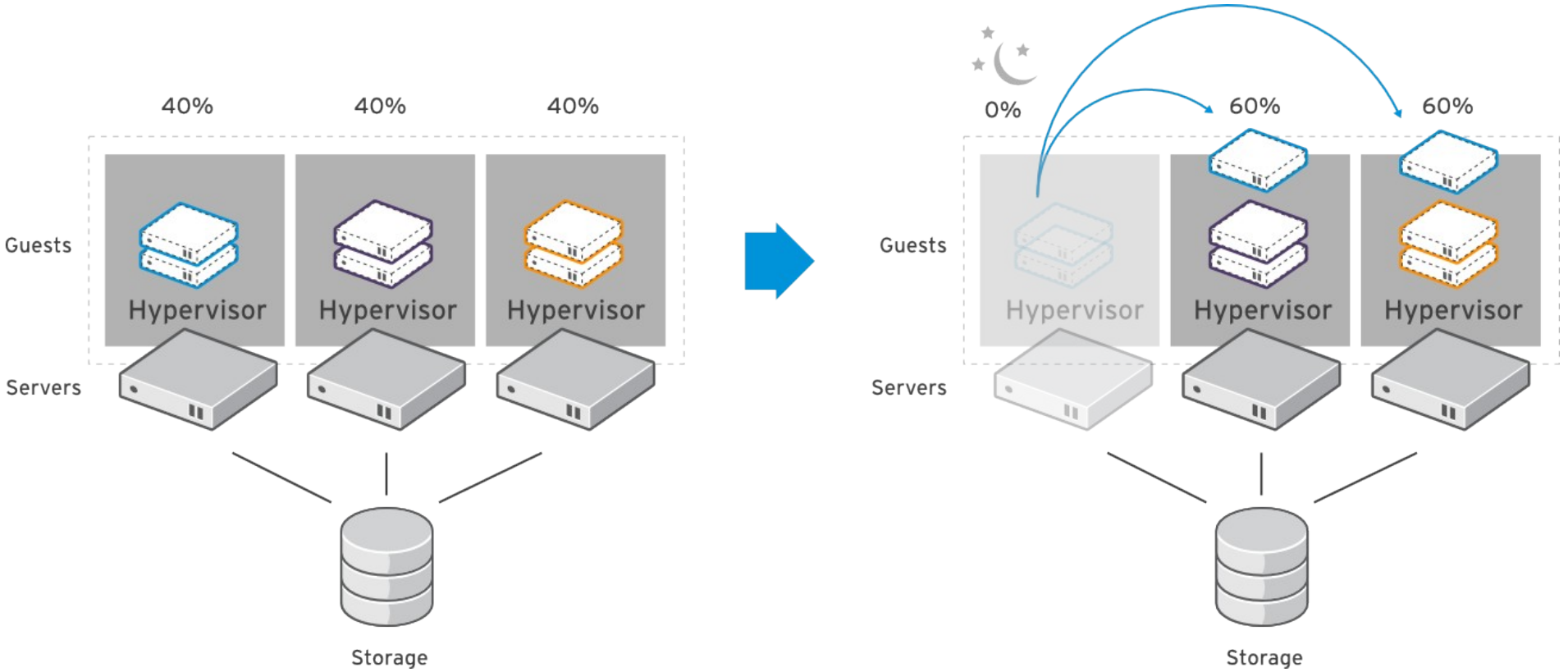
System Scheduler



- Dynamically balance workloads in the data center.
- Automatically live migrate virtual machines based on resources
- Define custom policies for distribution of virtual machines

Maintain consistent resource usage across the enterprise data center

Power Saver



Define policies to optimize workload on a fewer number of servers during “off-peak” hours

Management Features

Feature	Description
High Availability	Restart guest VMs from failed hosts automatically on other hosts
Live Migration	Move running VM between hosts with zero downtime
System Scheduler	Continuously load balance VMs based on resource usage/policies
Power Saver	Concentrate virtual machines on fewer servers during off-peak hours
Maintenance Manager	No downtime for virtual machines during planned maintenance windows. Hypervisor patching
Image Management	Template based provisioning, thin provisioning and snapshots
Monitoring & Reporting	For all objects in system – VM guests, hosts, networking, storage etc.
OVF Import/Export	Import and export VMs and templates using OVF files
V2V	Convert VMs from VMware and RHEL/Xen to RHEV

Virtual Desktop Infrastructure (VDI)

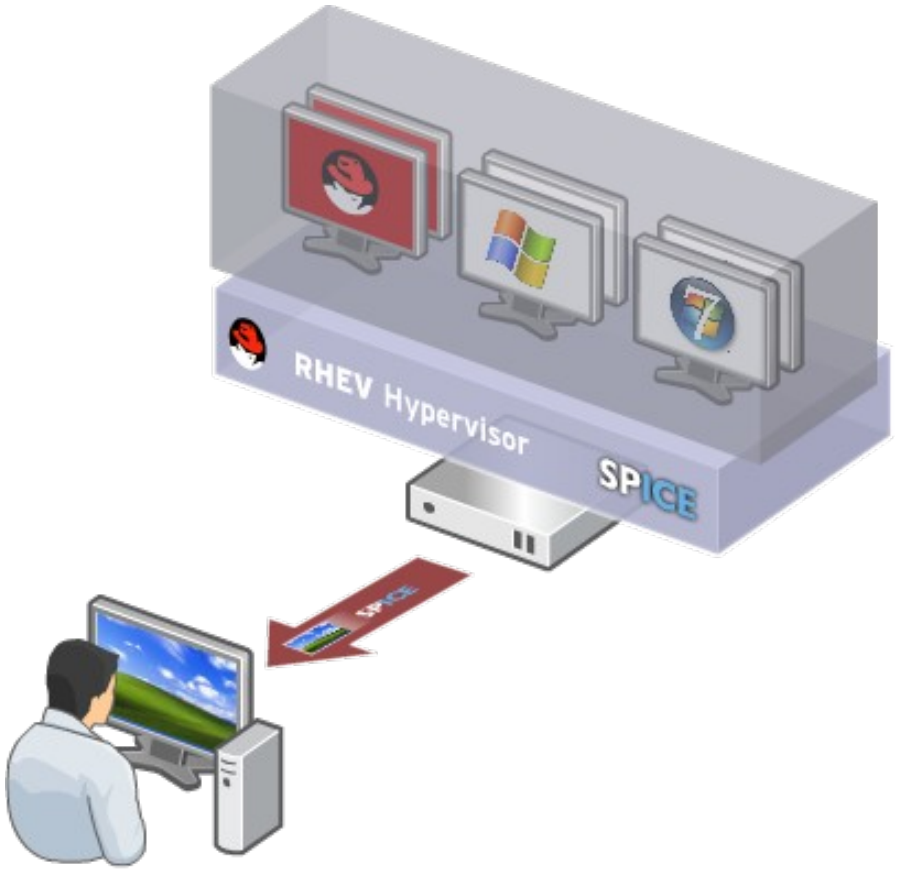
oVirt

Centralized management, security and policy enforcement

Virtual desktops with user experience of a physical PC

- Multiple monitors
- HD quality video
- Bi-directional audio/video for VoIP or video-conferencing
- Smartcard support
- USB support

Industry leading density of virtual desktops/server



Red Hat Enterprise Virtualization
RHEV 3.0 Key Initiatives

- Move from proprietary to open technologies
- Remove dependency on Windows
  (But maintain interoperability with Windows)
- Deliver new features and releases in parallel
- Build Open Source community project around open virtualization

History

- Qumranet
  - KVM
  - SPICE
  - **SolidICE --> RHEV-M C# --> RHEV-M Java --> oVirt**
- C# --> Java
  - using automatic conversion approach for core and UI[1]
- VDSM
- oVirt Node

[1] http://lpeer.blogspot.com/2010/04/switching-from-c-to-java.html

Things have changed

Things have evolved

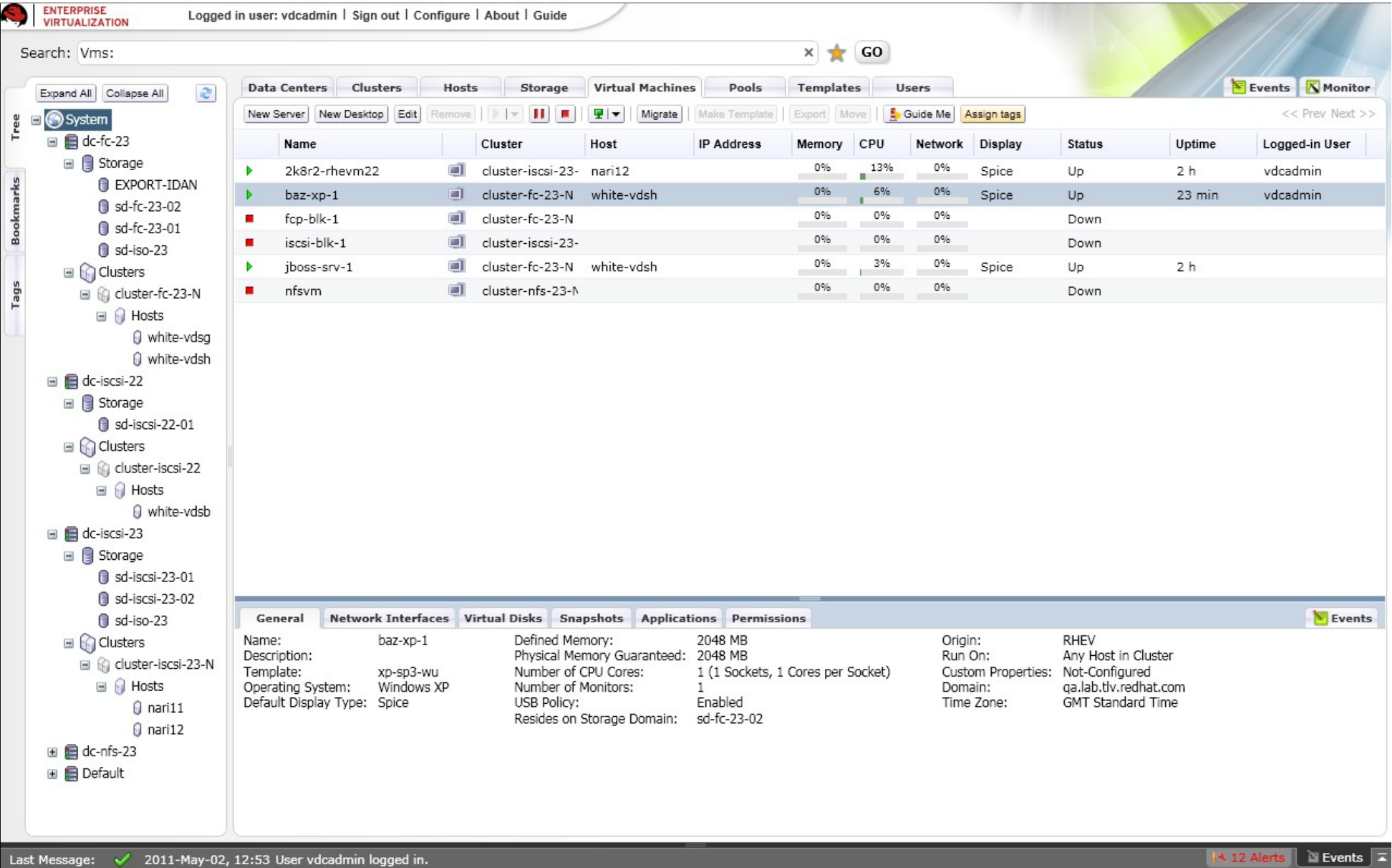
There are a lot of good ideas

There is a lot to refactor/change/do

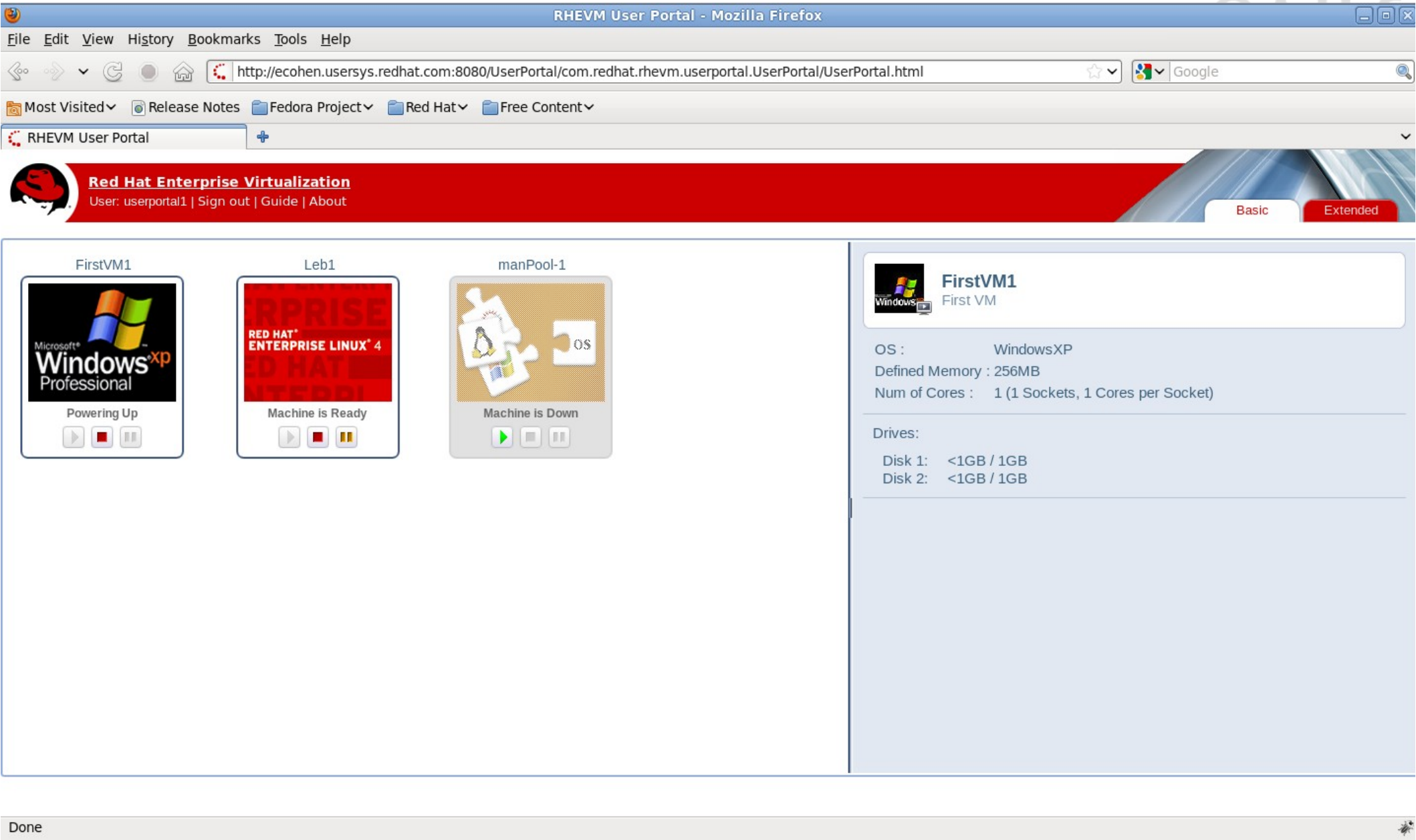
This is where we are

Let's get to work...

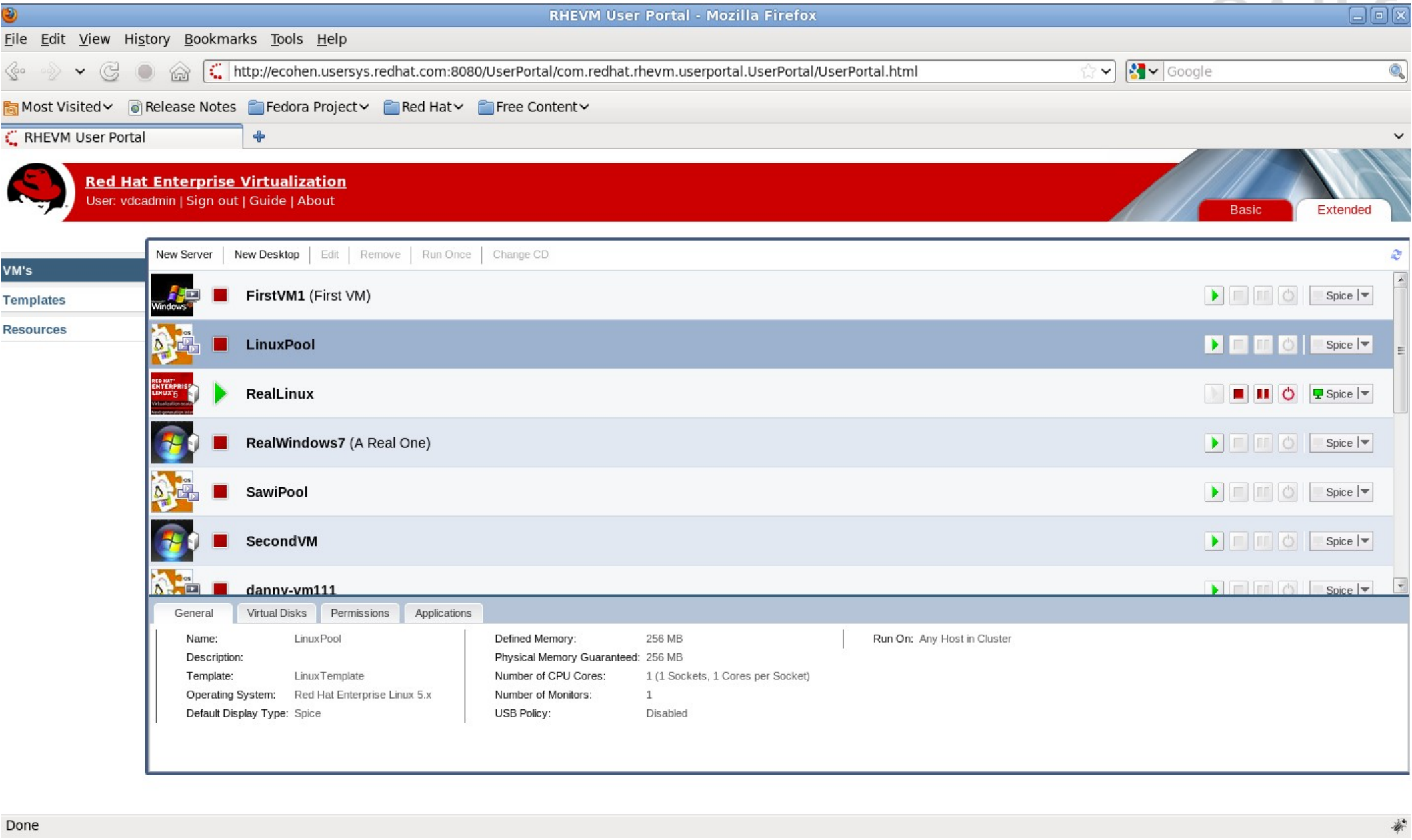
Admin Portal



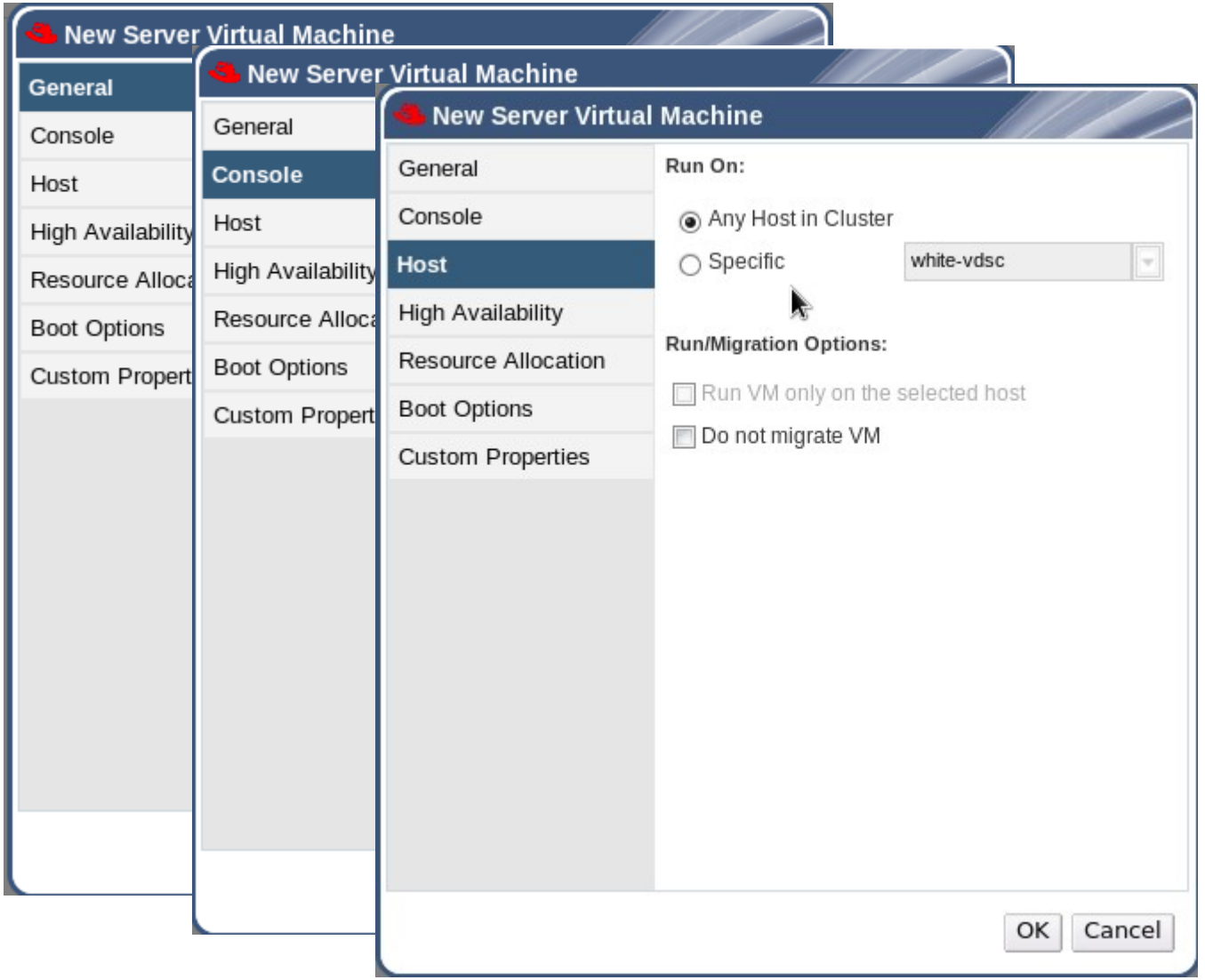
User Portal


RHEVM User Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://ecohen.usersys.redhat.com:8080/UserPortal/com.redhat.rhevm.userportal.UserPortal/UserPortal.html
Google
Most Visited
Release Notes
Fedora Project
Red Hat
Free Content
RHEVM User Portal
Red Hat Enterprise Virtualization
User: userportal1 | Sign out | Guide | About
Basic
Extended
FirstVM1
Leb1
manPool-1
Powering Up
Machine is Ready
Machine is Down
FirstVM1
First VM
OS : WindowsXP
Defined Memory : 256MB
Num of Cores : 1 (1 Sockets, 1 Cores per Socket)
Drives:
Disk 1: <1GB / 1GB
Disk 2: <1GB / 1GB
Done

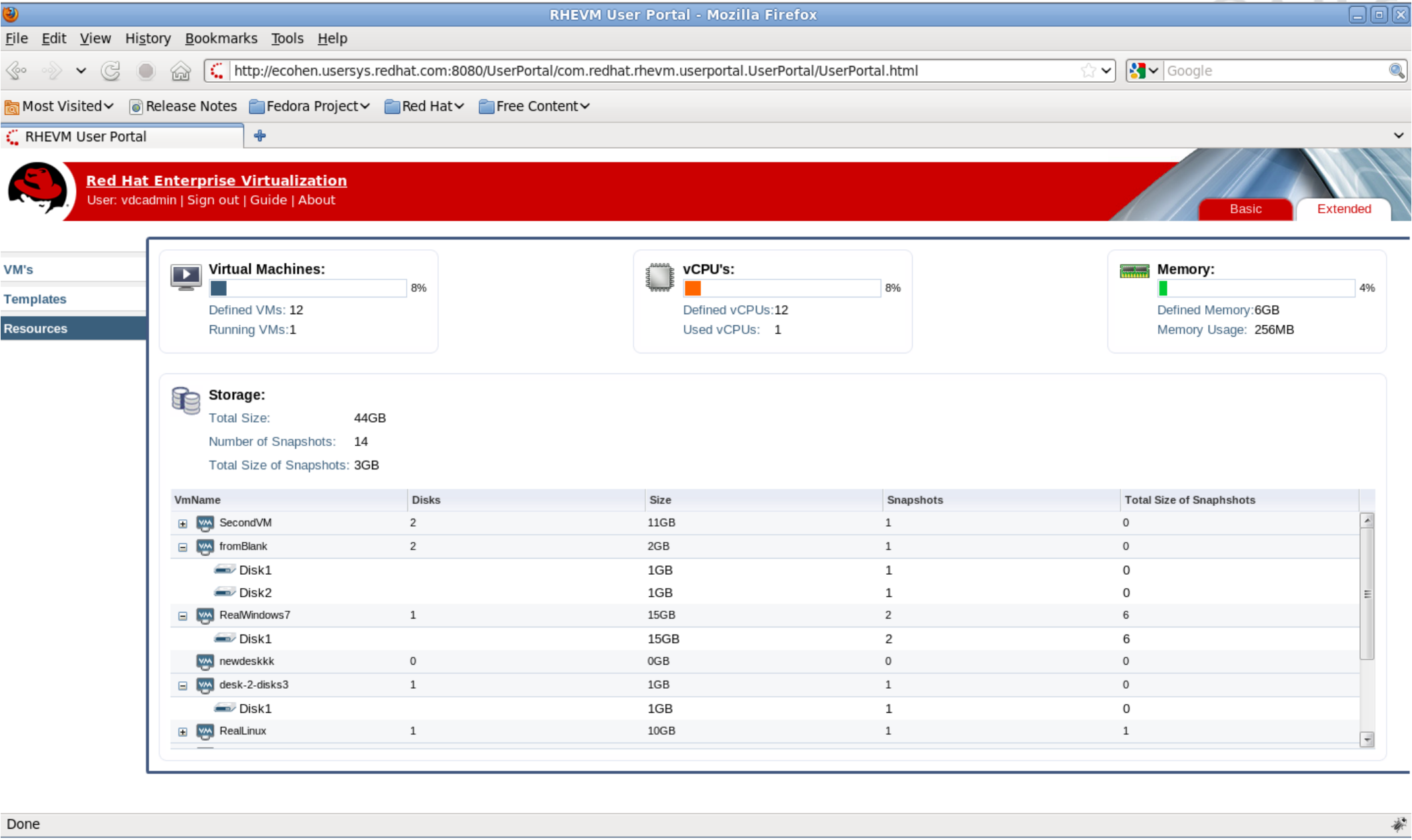
Power User Portal - VM's



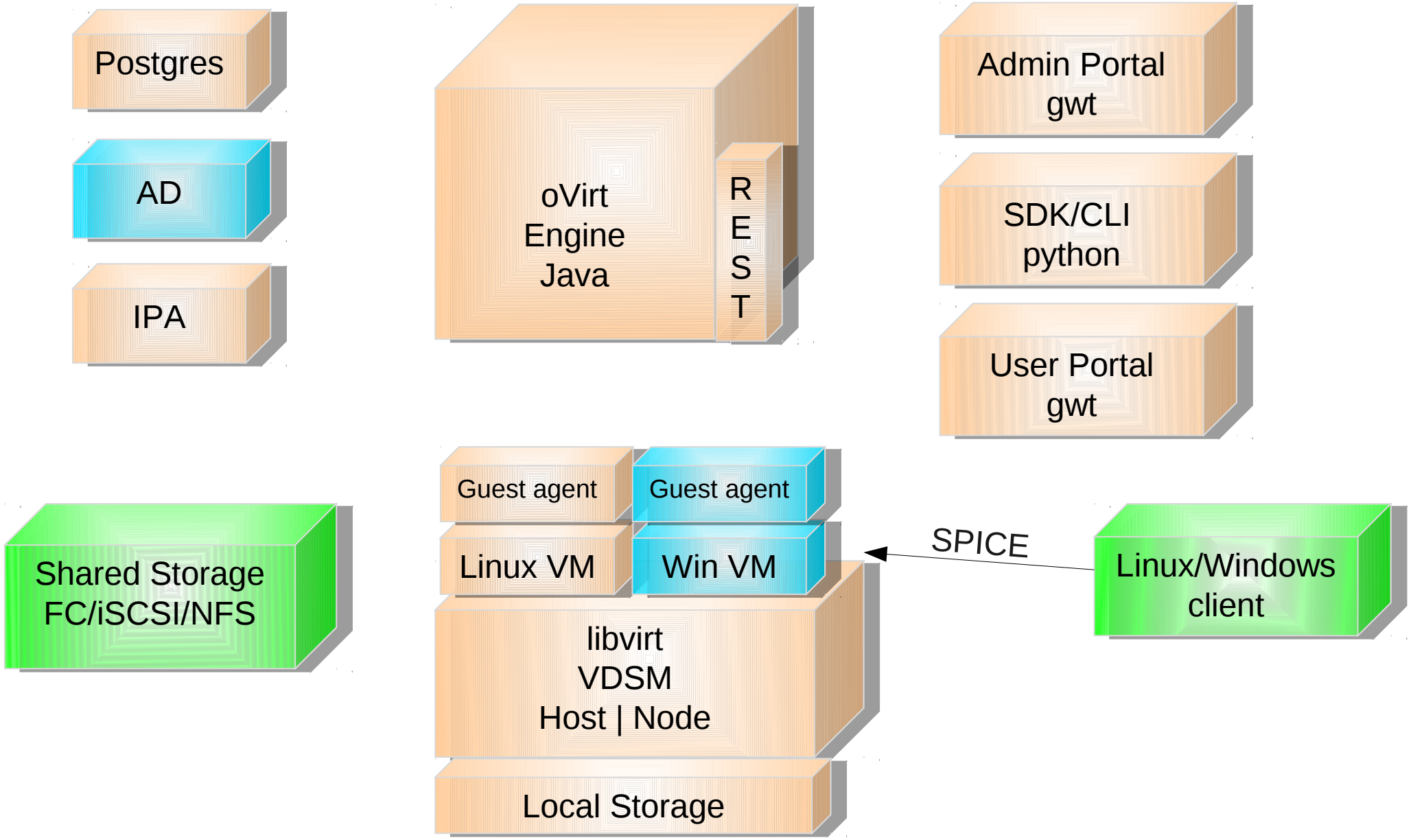
Add Virtual Machine



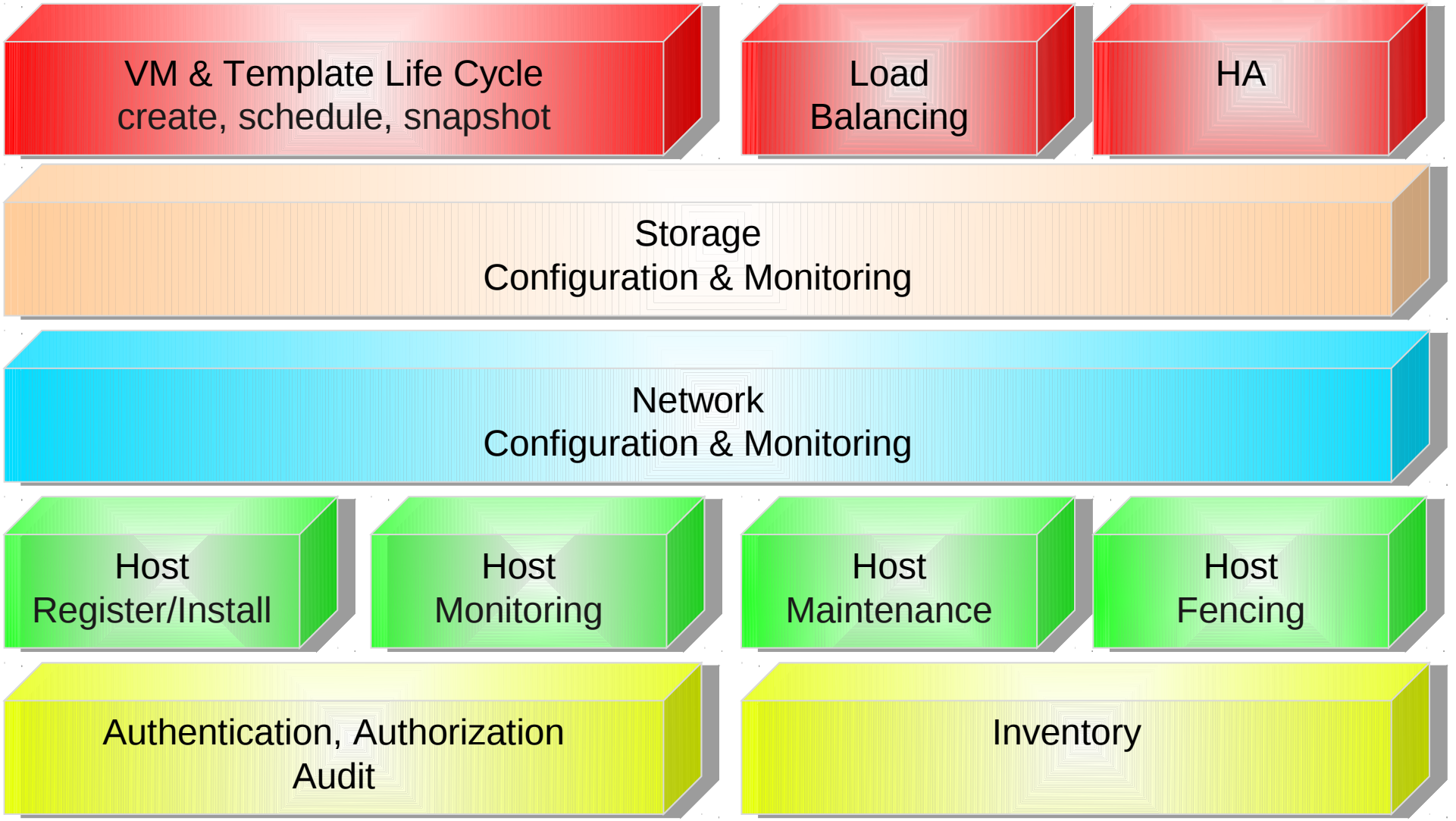
Power User Portal - Resources



oVirt High Level Architecture



Engine Core (Backend)



Authentication

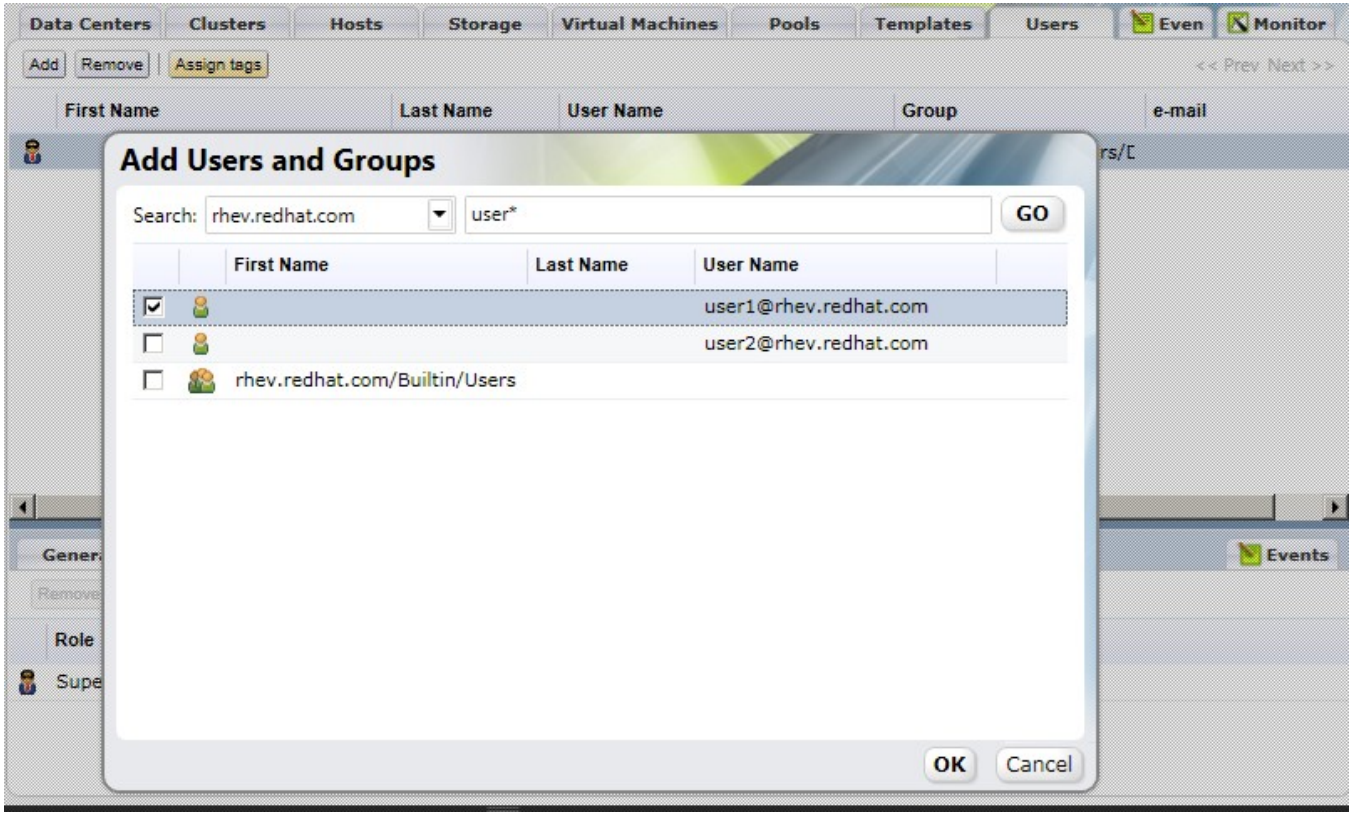
- Builtin user admin@internal
- AD, IPA integration
  - Kerberos authentication
  - LDAP - user info, group membership
  - Multiple domains, trusts, etc.
  - Cached for searches, not for login
- Next
  - Open LDAP (patch ready)
  - Internal users (picketlink?)
  - Linux users?

Multi Level Admin

- Users
- Groups
- Roles
- Permissions

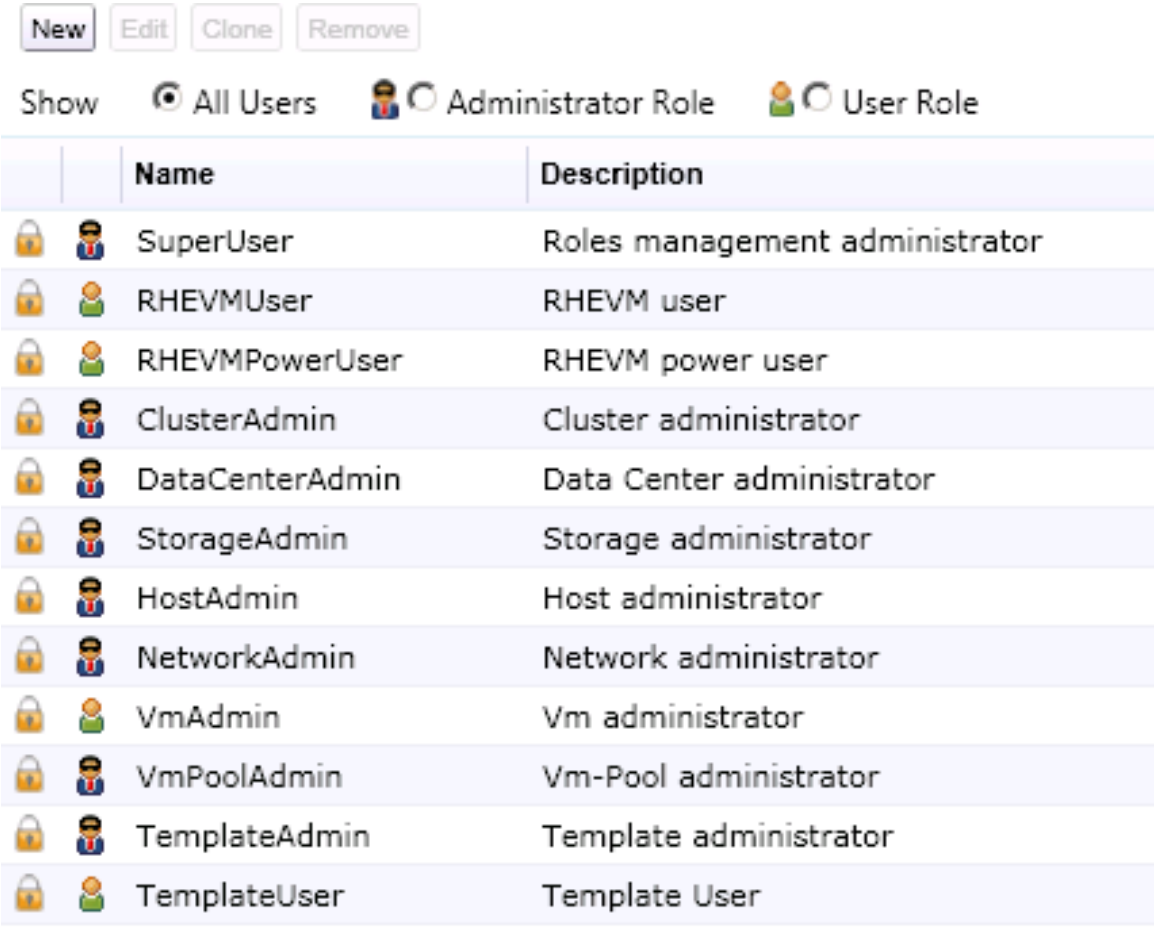
Multi Level Admin

- **Users**
- **Groups**
- Roles
- Permissions



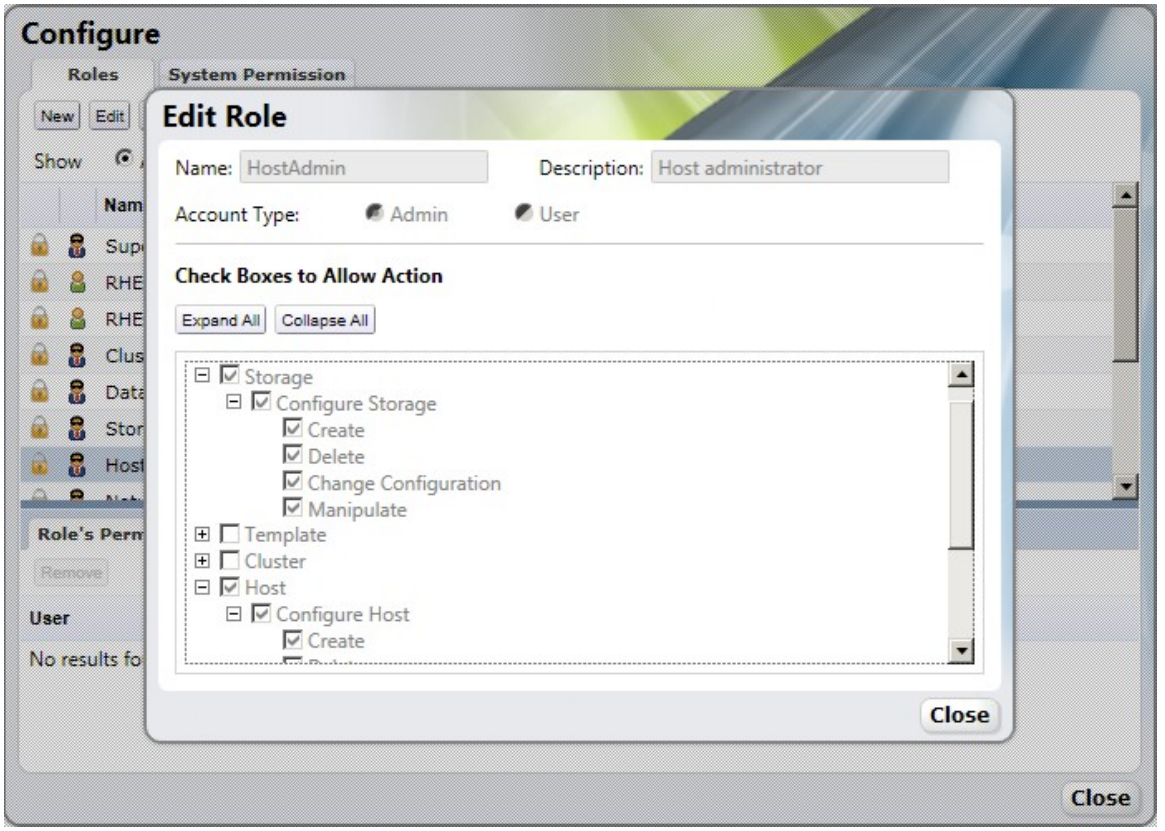
Multi Level Admin

- Users
- Groups
- **Roles**
- Permissions



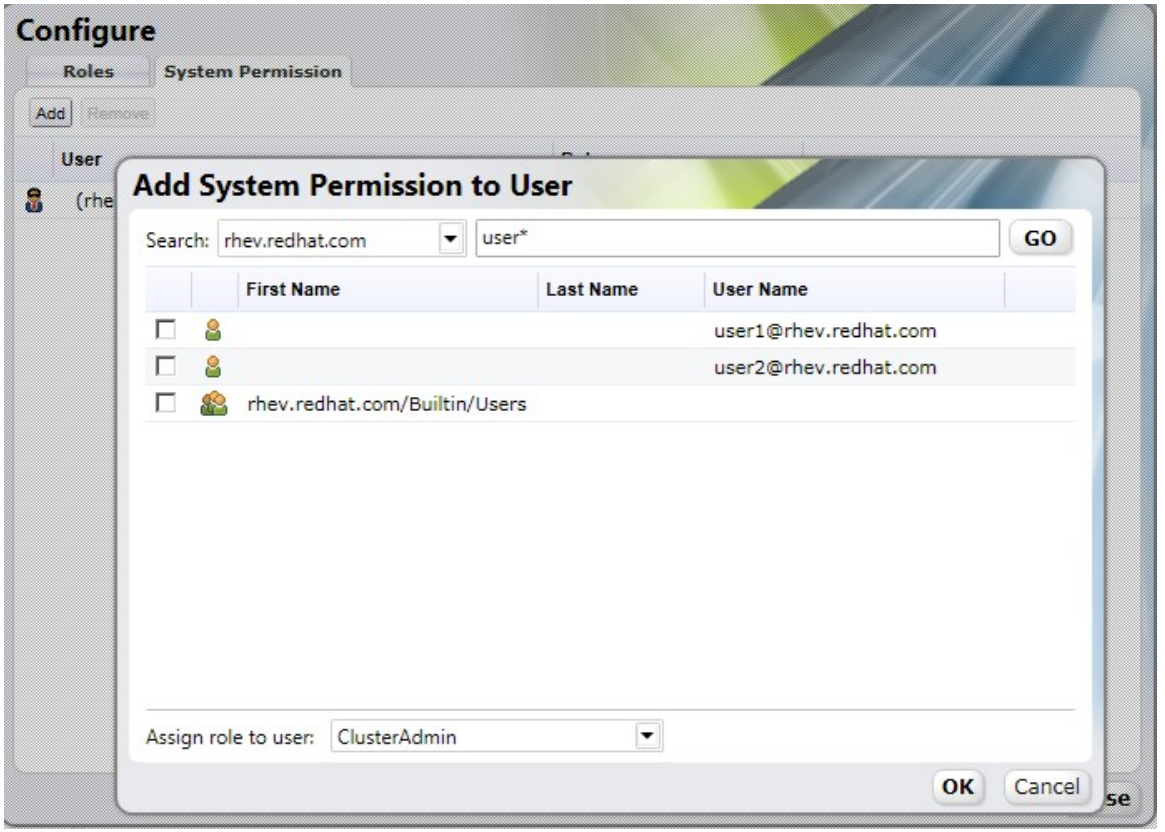
Multi Level Admin

- Users
- Groups
- **Roles**
- Permissions



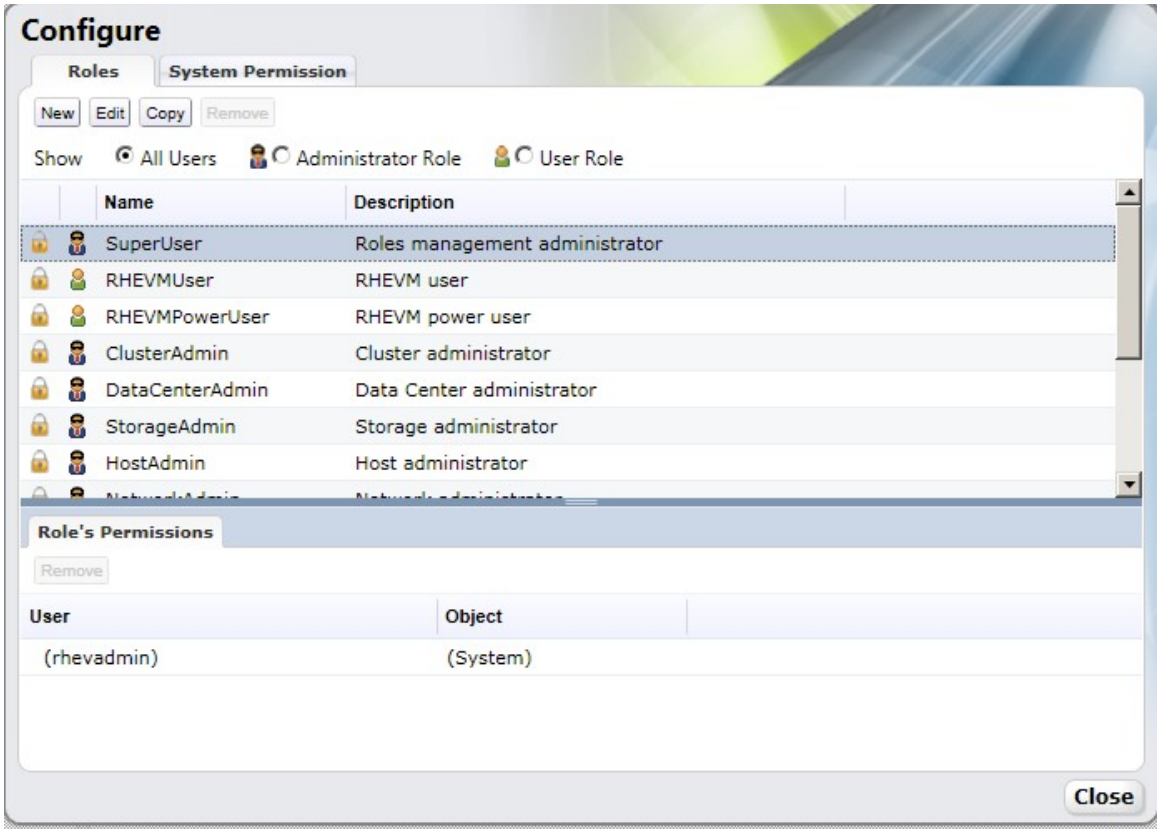
Multi Level Admin

- Users
- Groups
- Roles
- **Permissions**



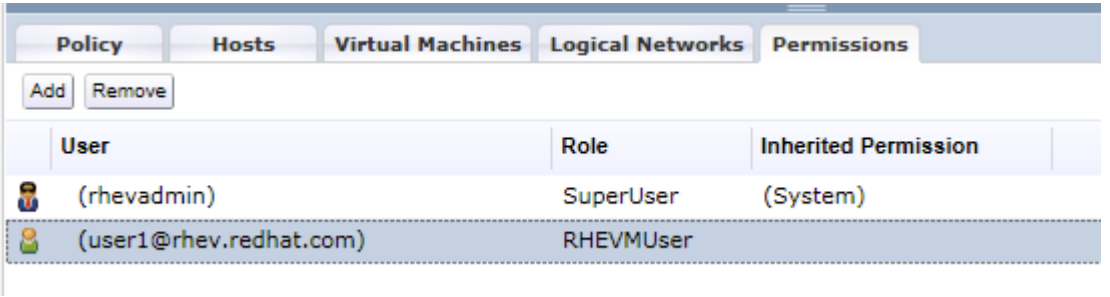
Multi Level Admin

- Users
- Groups
- Roles
- **Permissions**



Multi Level Admin

- Users
- Groups
- Roles
- **Permissions**



Database

- Moved from SQL Server to Postgres
- JDBC based
- Next
  - Hibernate
  - Scheme upgrade management

REST API

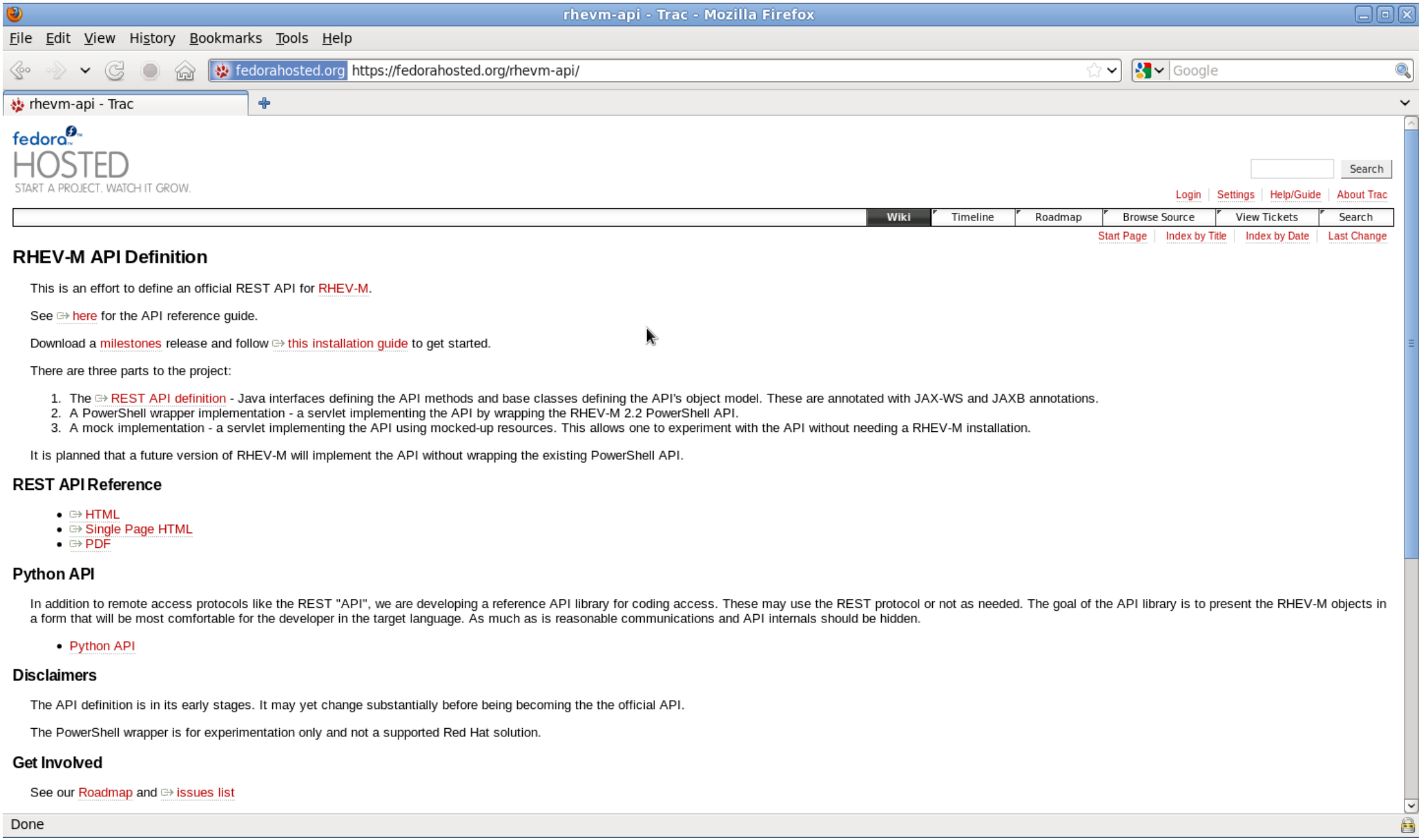
- New RESTful API for integration with oVirt Engine
  - REST interface exposed for all API functions
  - Developed in upstream RHEV-M API project (before oVirt)

RHEVM-API Upstream Project

Community project to deliver RESTful API for RHEV 2.2

https://fedorahosted.org/rhevm-api/

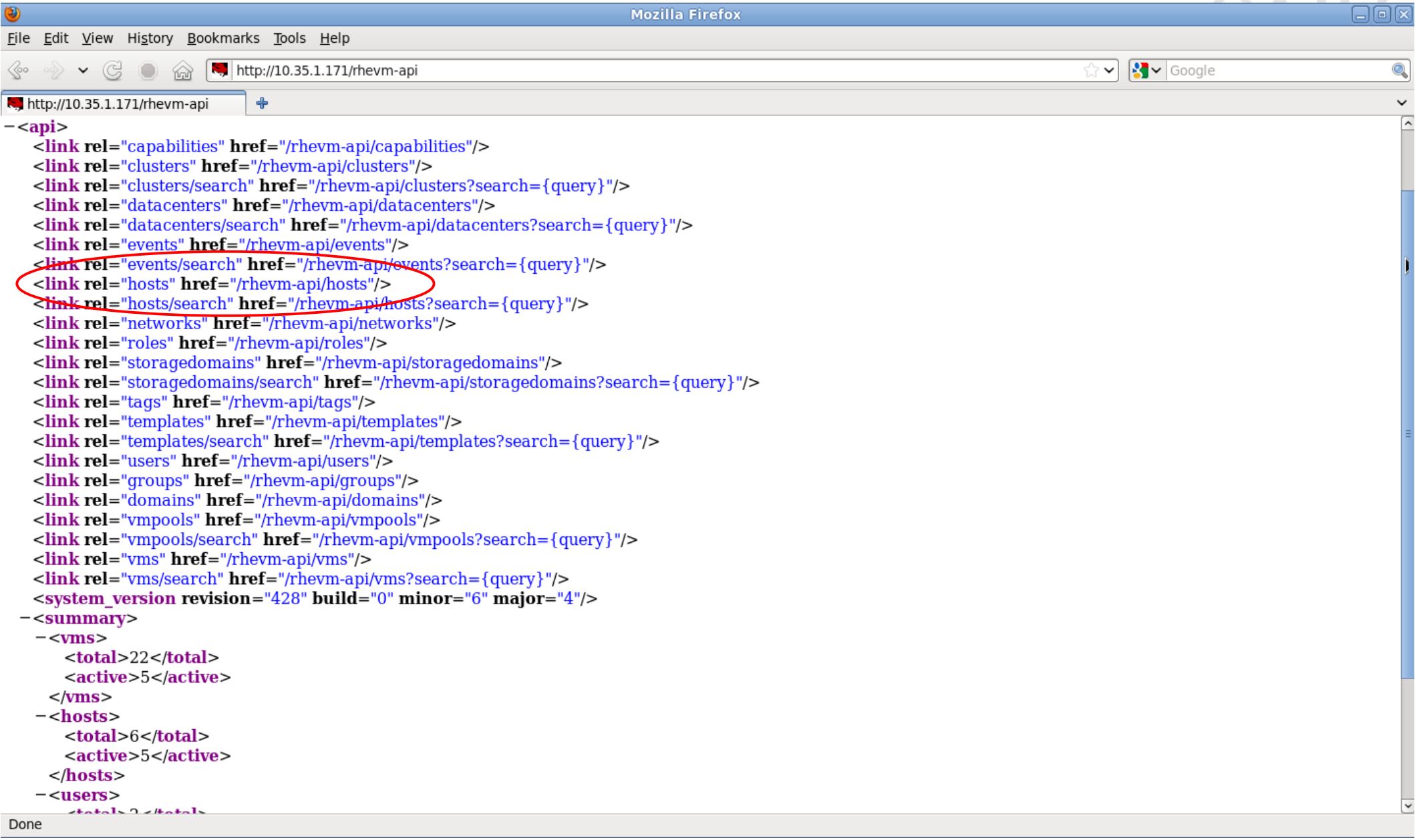
- Provides preview of 3.0 RESTful API
  - Draft implementation of new API
  - Runs on RHEL 2.2 wraps PowerShell
    - Allows early testing of API for customers and partners
    - 3.0 Implementation based on Java backend engine
  - Will be consolidated into oVirt


rhevm-api - Trac - Mozilla Firefox
File Edit View History Bookmarks Tools Help
fedorahosted.org https://fedorahosted.org/rhevm-api/
Google
rhevm-api - Trac
fedora HOSTED
START A PROJECT. WATCH IT GROW.
Search
Login | Settings | Help/Guide | About Trac
Wiki Timeline Roadmap Browse Source View Tickets Search
Start Page | Index by Title | Index by Date | Last Change
RHEV-M API Definition
This is an effort to define an official REST API for RHEV-M.
See here for the API reference guide.
Download a milestones release and follow this installation guide to get started.
There are three parts to the project:
1. The REST API definition - Java interfaces defining the API methods and base classes defining the API's object model. These are annotated with JAX-WS and JAXB annotations.
2. A PowerShell wrapper implementation - a servlet implementing the API by wrapping the RHEV-M 2.2 PowerShell API.
3. A mock implementation - a servlet implementing the API using mocked-up resources. This allows one to experiment with the API without needing a RHEV-M installation.
It is planned that a future version of RHEV-M will implement the API without wrapping the existing PowerShell API.
REST API Reference
HTML
Single Page HTML
PDF
Python API
In addition to remote access protocols like the REST "API", we are developing a reference API library for coding access. These may use the REST protocol or not as needed. The goal of the API library is to present the RHEV-M objects in a form that will be most comfortable for the developer in the target language. As much as is reasonable communications and API internals should be hidden.
Python API
Disclaimers
The API definition is in its early stages. It may yet change substantially before being becoming the the official API.
The PowerShell wrapper is for experimentation only and not a supported Red Hat solution.
Get Involved
See our Roadmap and issues list
Done

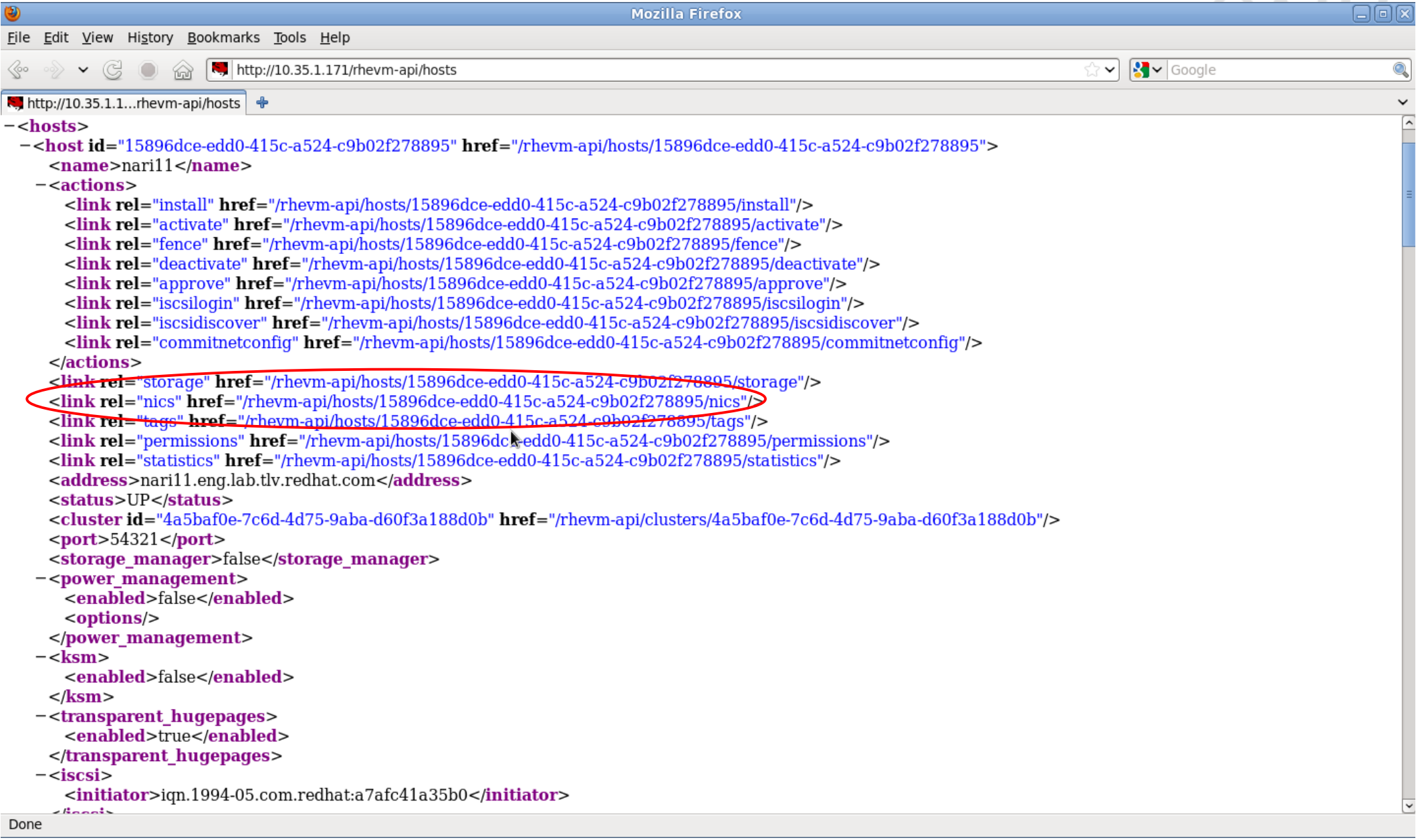
RESTful Web Service

- Stands for Representational State Transfer
- Modeling entity actions around HTTP verbs
  - GET
  - PUT
  - POST
  - DELETE
- Still uses 'actions' for some state changes
- Self describes – entity navigation and actions

Welcome

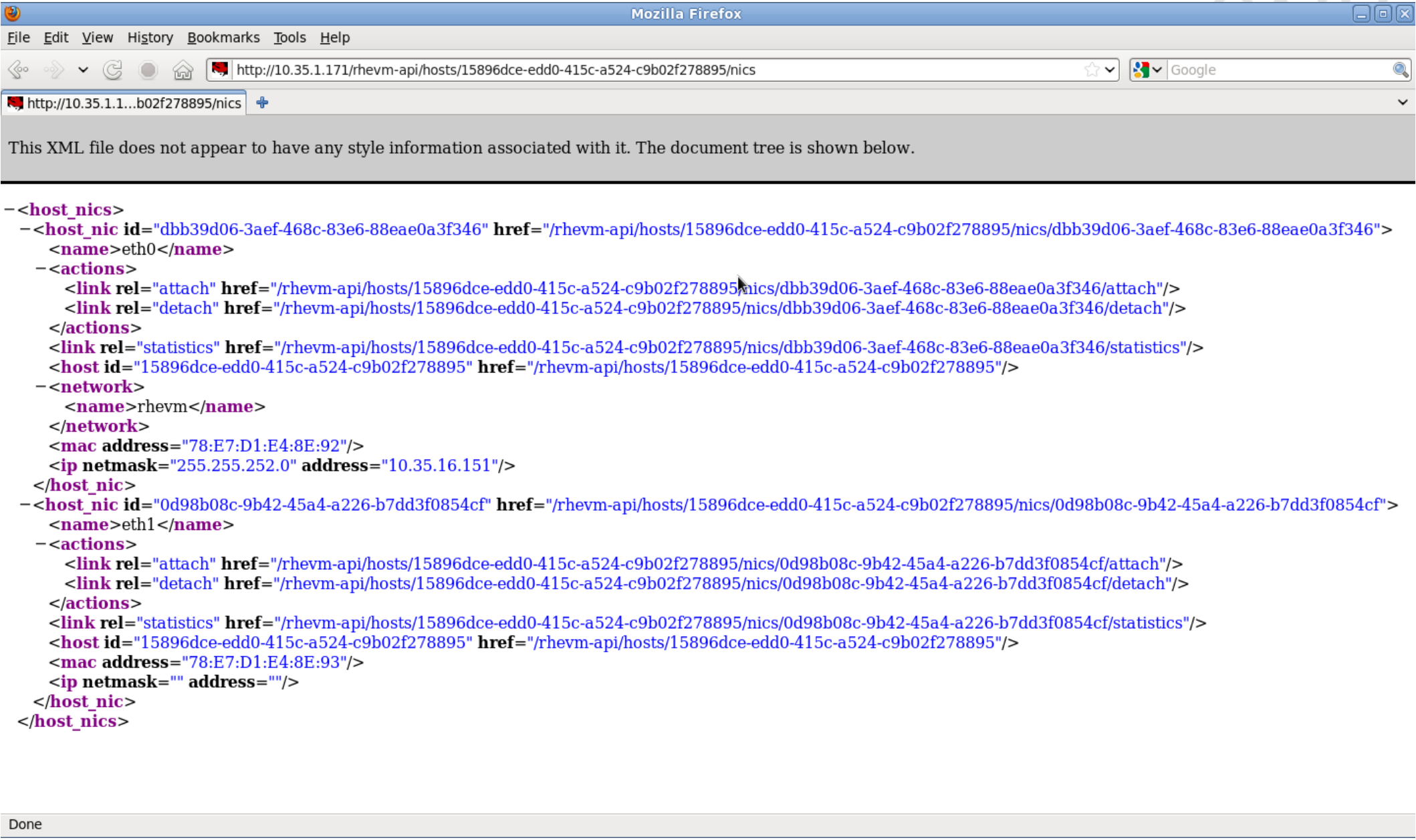

Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://10.35.1.171/rhevm-api
Google
http://10.35.1.171/rhevm-api
−<api>
<link rel="capabilities" href="/rhevm-api/capabilities"/>
<link rel="clusters" href="/rhevm-api/clusters"/>
<link rel="clusters/search" href="/rhevm-api/clusters?search={query}"/>
<link rel="datacenters" href="/rhevm-api/datacenters"/>
<link rel="datacenters/search" href="/rhevm-api/datacenters?search={query}"/>
<link rel="events" href="/rhevm-api/events"/>
<link rel="events/search" href="/rhevm-api/events?search={query}"/>
<link rel="hosts" href="/rhevm-api/hosts"/>
<link rel="hosts/search" href="/rhevm-api/hosts?search={query}"/>
<link rel="networks" href="/rhevm-api/networks"/>
<link rel="roles" href="/rhevm-api/roles"/>
<link rel="storagedomains" href="/rhevm-api/storagedomains"/>
<link rel="storagedomains/search" href="/rhevm-api/storagedomains?search={query}"/>
<link rel="tags" href="/rhevm-api/tags"/>
<link rel="templates" href="/rhevm-api/templates"/>
<link rel="templates/search" href="/rhevm-api/templates?search={query}"/>
<link rel="users" href="/rhevm-api/users"/>
<link rel="groups" href="/rhevm-api/groups"/>
<link rel="domains" href="/rhevm-api/domains"/>
<link rel="vmpools" href="/rhevm-api/vmpools"/>
<link rel="vmpools/search" href="/rhevm-api/vmpools?search={query}"/>
<link rel="vms" href="/rhevm-api/vms"/>
<link rel="vms/search" href="/rhevm-api/vms?search={query}"/>
<system_version revision="428" build="0" minor="6" major="4"/>
−<summary>
−<vms>
<total>22</total>
<active>5</active>
</vms>
−<hosts>
<total>6</total>
<active>5</active>
</hosts>
−<users>
Done

Hosts Collection




```
−<hosts>
  −<host id="15896dce-edd0-415c-a524-c9b02f278895" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895">
    <name>nari11</name>
    −<actions>
      <link rel="install" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/install"/>
      <link rel="activate" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/activate"/>
      <link rel="fence" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/fence"/>
      <link rel="deactivate" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/deactivate"/>
      <link rel="approve" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/approve"/>
      <link rel="iscsilogin" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/iscsilogin"/>
      <link rel="iscsidiscover" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/iscsidiscover"/>
      <link rel="commitnetconfig" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/commitnetconfig"/>
    </actions>
    <link rel="storage" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/storage"/>
    <link rel="nics" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/nics"/>
    <link rel="tags" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/tags"/>
    <link rel="permissions" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/permissions"/>
    <link rel="statistics" href="/rhevm-api/hosts/15896dce-edd0-415c-a524-c9b02f278895/statistics"/>
    <address>nari11.eng.lab.tlv.redhat.com</address>
    <status>UP</status>
    <cluster id="4a5baf0e-7c6d-4d75-9aba-d60f3a188d0b" href="/rhevm-api/clusters/4a5baf0e-7c6d-4d75-9aba-d60f3a188d0b"/>
    <port>54321</port>
    <storage_manager>false</storage_manager>
    −<power_management>
      <enabled>false</enabled>
      <options/>
    </power_management>
    −<ksm>
      <enabled>false</enabled>
    </ksm>
    −<transparent_hugepages>
      <enabled>true</enabled>
    </transparent_hugepages>
    −<iscsi>
      <initiator>iqn.1994-05.com.redhat:a7afc41a35b0</initiator>
```

Host networks collection



Create a Virtual Machine from a Template

POST http://10.35.1.1/rhevm-api/vms

```
<vm>
  <name>my_new_vm</name>
  <cluster id="99408929-82cf-4dc7-a532-9d998063fa95" />
  <template id="00000000-0000-0000-0000-000000000000" />
</vm>
```

```
curl -v -u "vdcadmin@qa.lab.tlv.redhat.com"
   -H "Content-type: application/xml"
   -d '<vm><name>my_new_vm</name><cluster id="99408929-82cf-4dc7-
       a532-9d998063fa95" /><template id="00000000-0000-0000-0000-
       000000000000"/></vm>'
   'http://10.35.1.1/rhevm-api/vms'
```

Changing a property

PUT http://10.35.1.1/rhevm-api/vms/2496a177-e7c8-4f82-bf3d-2d0f73444990

```
<vm>
  <name>test_vm_new_name</name>
</vm>
```

```
echo "<vm><name>test_vm_new_name1</name></vm>" >  /tmp/upload.xml
curl -v -u "vdcadmin@qa.lab.tlv.redhat.com"
  -H "Content-type: application/xml"
  -T /tmp/upload.xml
  'http://10.35.1.1/rhevm-api/vms/2496a177-e7c8-4f82-bf3d-2d0f73444990'
```

Adding a Virtual Disk

POST
http://10.35.1.1/rhevm-api/vms/2496a177-e7c8-4f82-bf3d-2

```
<disk>
  <storage_domain id="3e1c96f0-8667-4a80-9689-af1337395dea" href="/rhevm-
api/storagedomains/3e1c96f0-8667-4a80-9689-af1337395dea" />
  <size>1073741824</size>
  <type>system</type>
  <interface>virtio</interface>
  <format>raw</format>
  <sparse>true</sparse>
  <bootable>true</bootable>
  <wipe_after_delete>false</wipe_after_delete>
  <propagate_errors>false</propagate_errors>
</disk>
```

- curl -v -u "vdcadmin@qa.lab.tlv.redhat.com"
  -H "Content-type: application/xml"
  -d '<disk>...</disk>' http://...

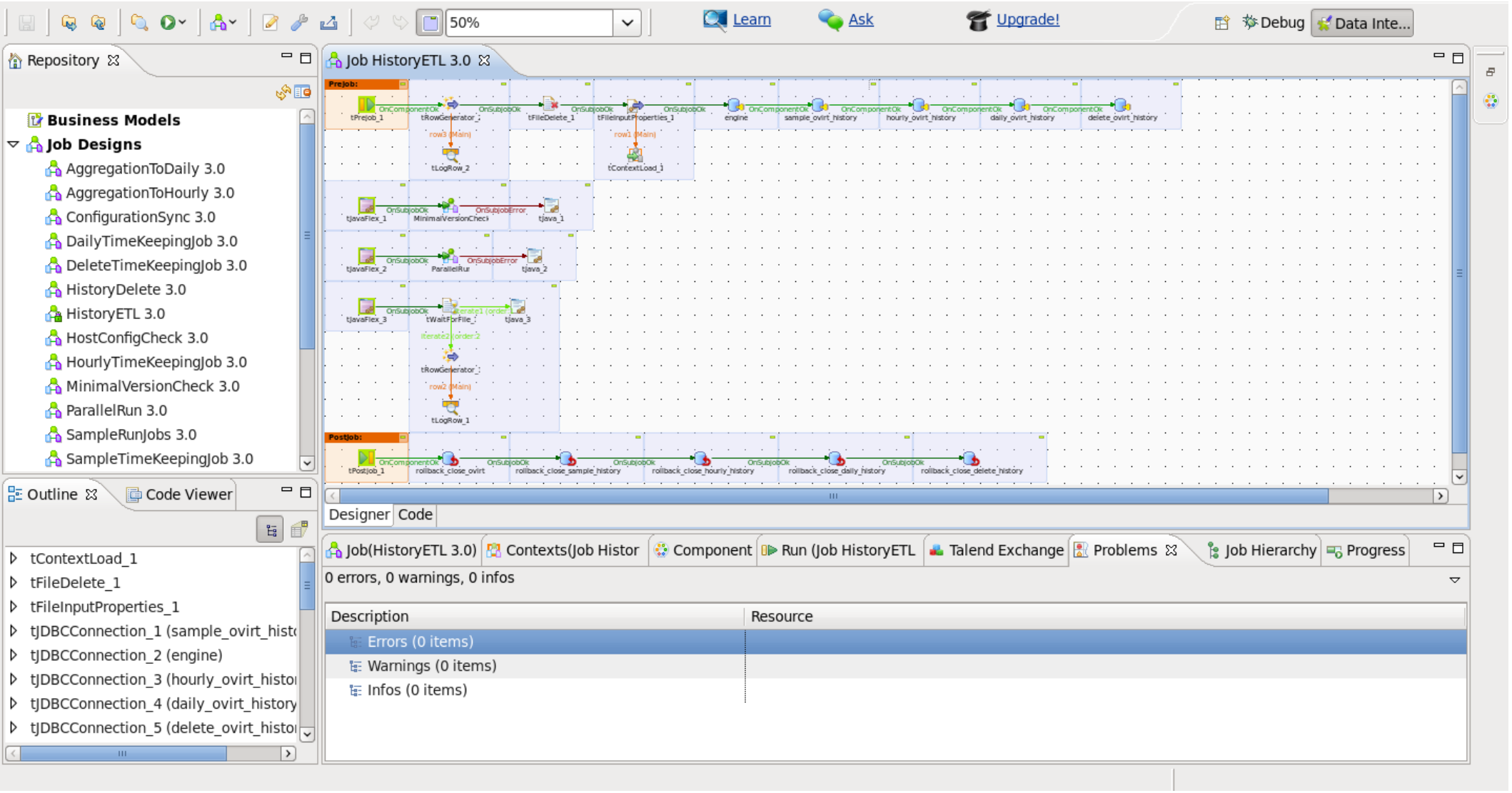
What Else?

- Data warehouse
- Reports (based on jasperforge.org)
- Tools
  - Notifications
  - Config
  - Iso uploader
  - Log collector

oVirt Data Warehouse

- ETL based on talendforge.org
- Periodic polling from operational DB
- Types of data
  - Config with version tracking
  - Statistics – aggregated hourly/daily
- API is view based

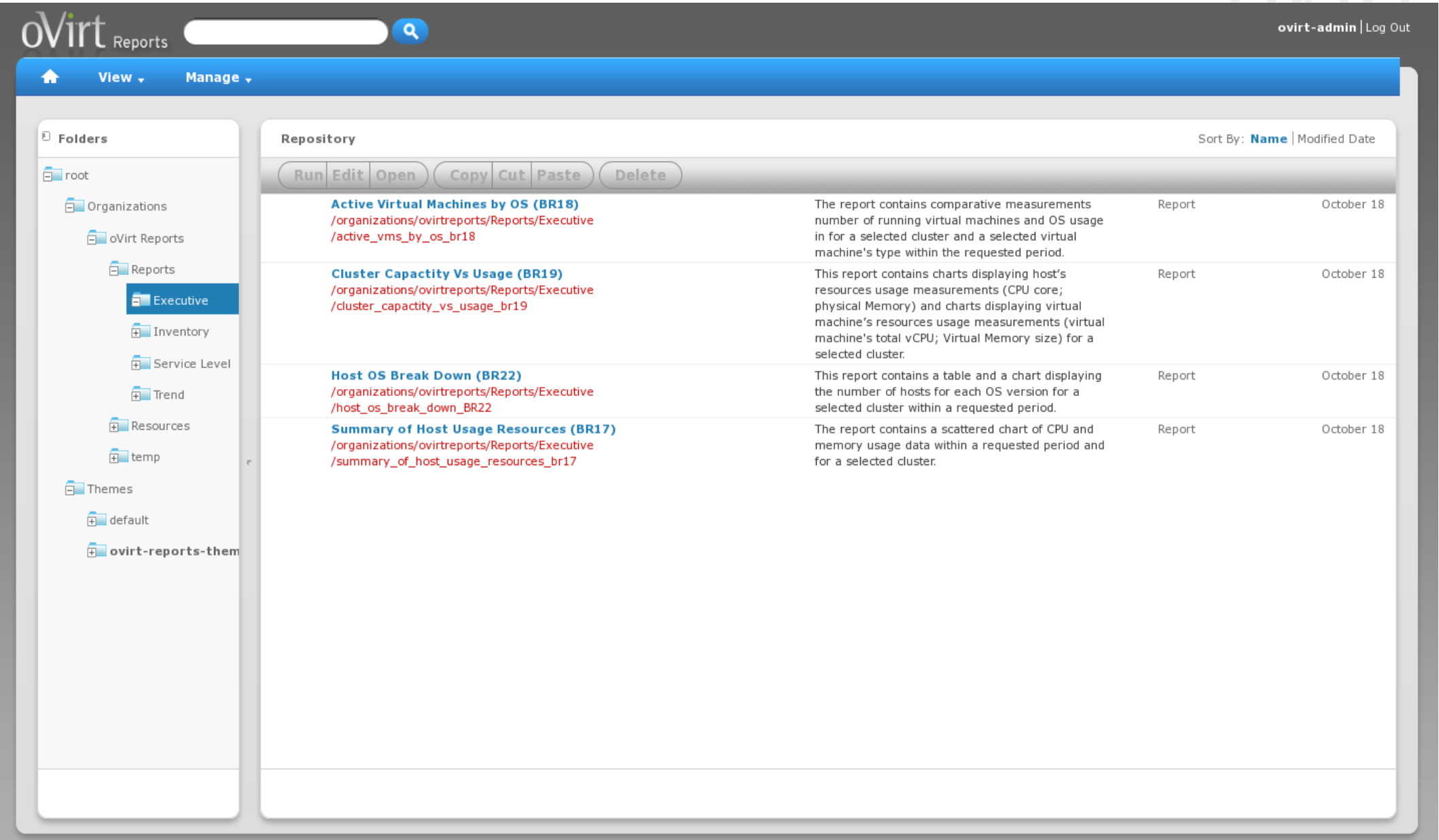
Talend Studio



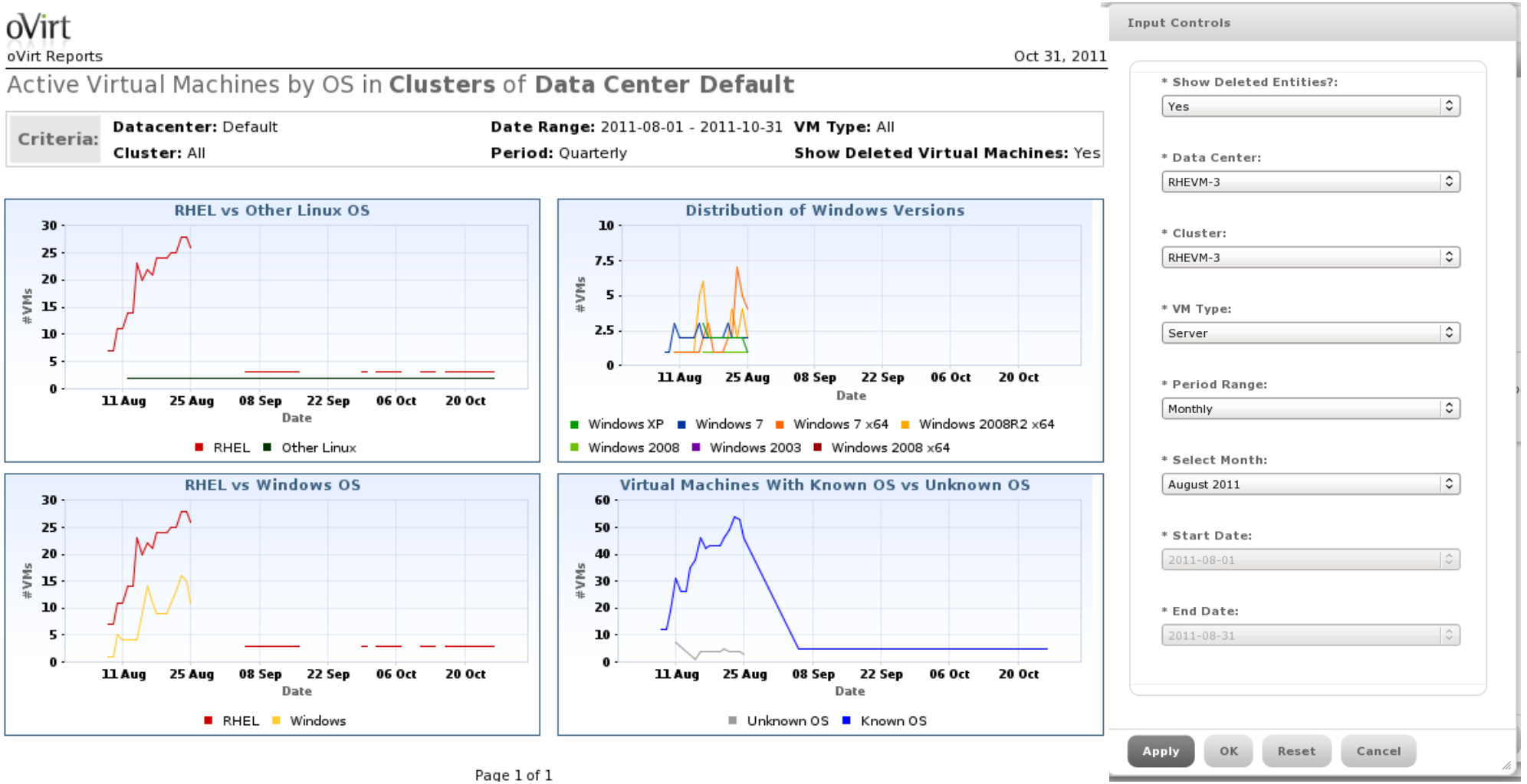
oVirt Reports

- Jasper allows to import/export reports definitions
- Rich reporting engine
  - Report scheduling
  - Filters
  - Export to various formats
  - Report creation studio
- Next
  - Integrated in web admin

oVirt Reports

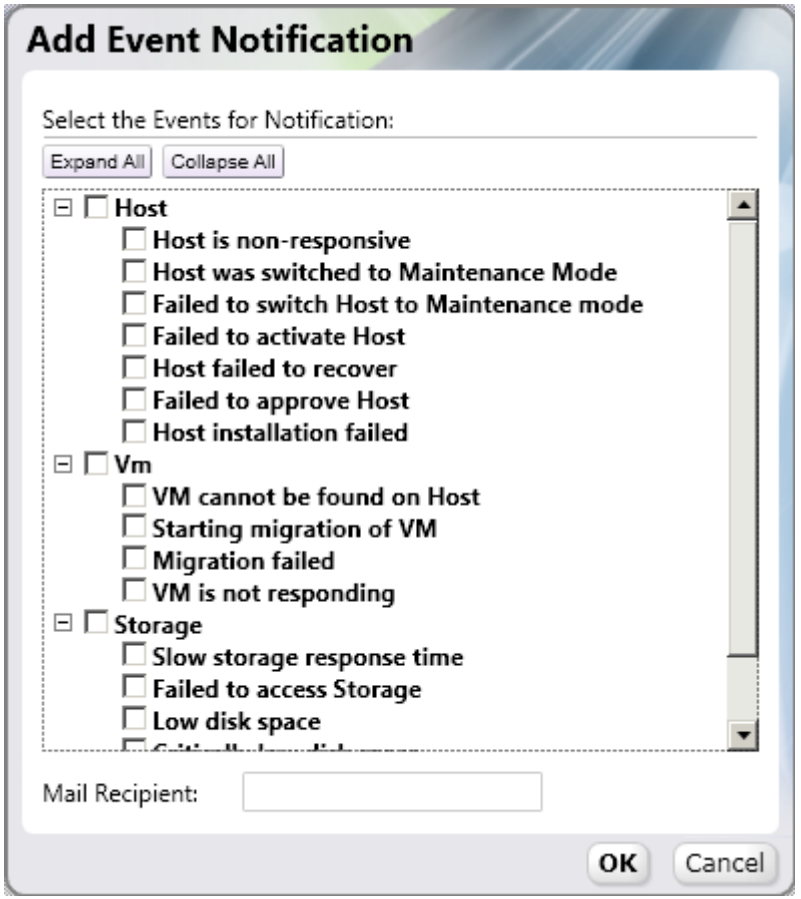


oVirt Reports



Notification Service

- oVirt allows registration to certain audit events
- The notification service sends emails per audit message to relevant users
- Also monitors engine itself



Configuration tool

- The configuration utility allows changing oVirt advanced configuration options
- Sample commands
  - engine-config --list
  - engine-config --get <key_name>
  - engine-config -all
  - engine-config --set <key_name>=<value>
- Special config for authentication domains: manage-domains

ISO Uploader

- Iso uploader is a utility to upload iso files to the iso domain, to allow bootstrapping guests from them
- Admin can just copy the files to the iso domain
- Supports both scp and nfs based copies
- Integrates with the REST API to allow using storage domain name instead of specific NFS path

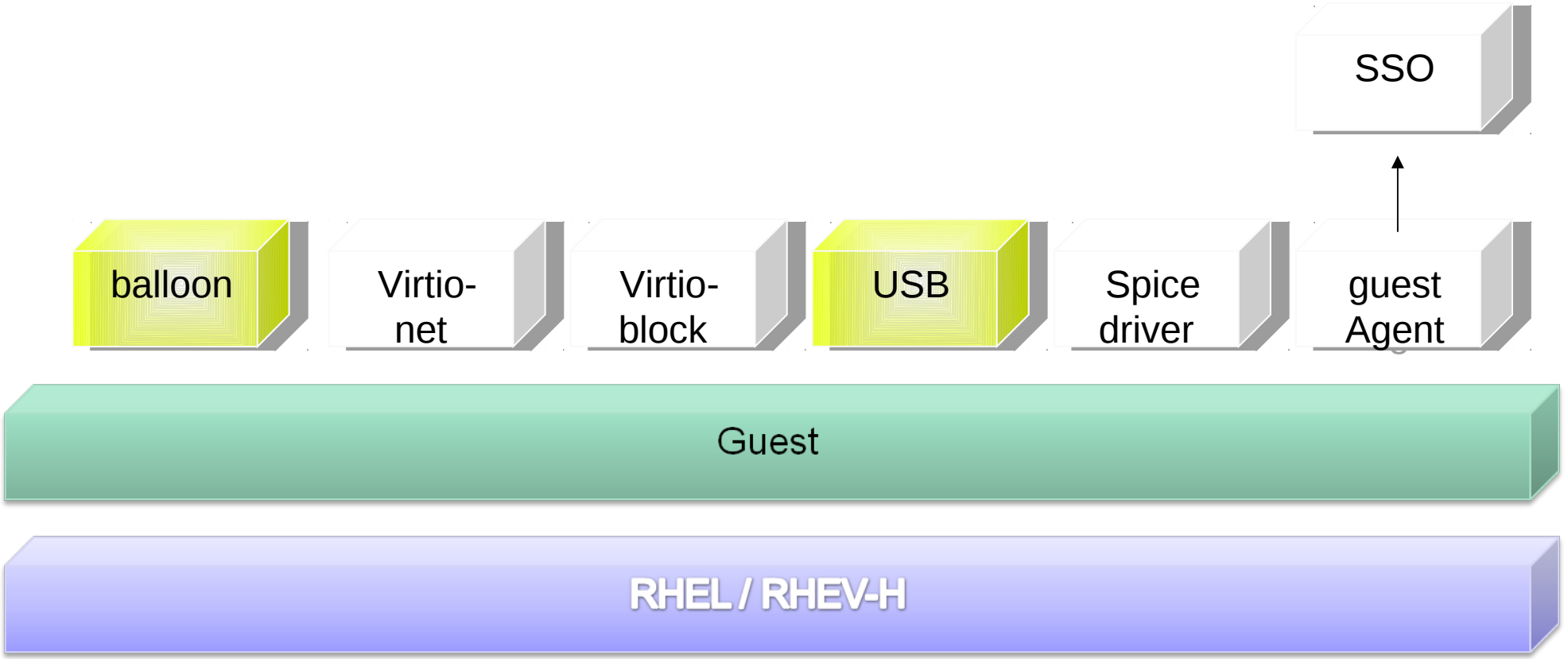
Log Collector

- The log collector utility helps collecting logs and configuration data for troubleshooting
- Written as a linux script launching sos plugins
- Collects the data from engine and nodes

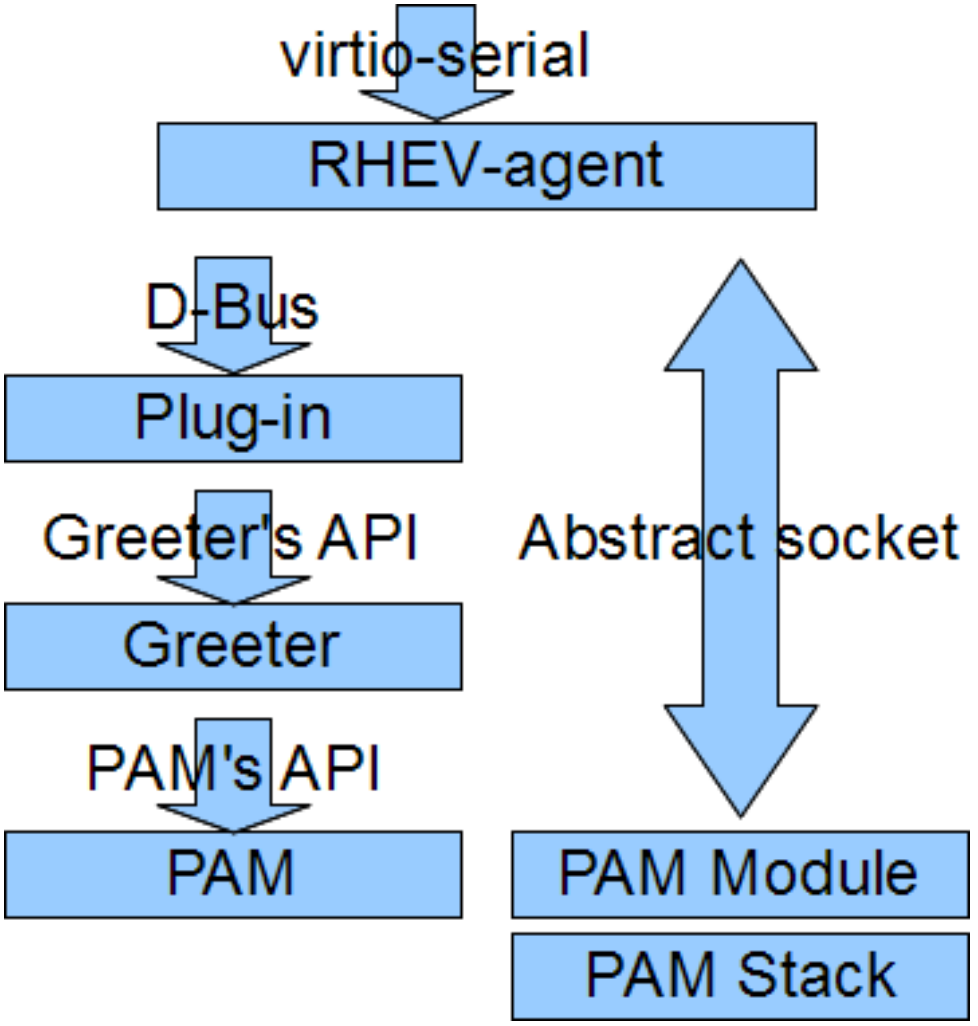
oVirt Guest Agent

- The guest agent provides additional information to oVirt Engine, such as guest memory usage, guest ip address, installed applications and sso.
- Python code, available for both linux and windows guests
- Communication is done over virtio-serial
- SSO for windows is based on a gina module for XP and a credential provider for windows 7
- SSO for RHEL 6 is based on a PAM module with support for both KDE and Gnome

Guest


SSO
balloon
Virtio-net
Virtio-block
USB
Spice driver
guest Agent
Guest
RHEL / RHEV-H

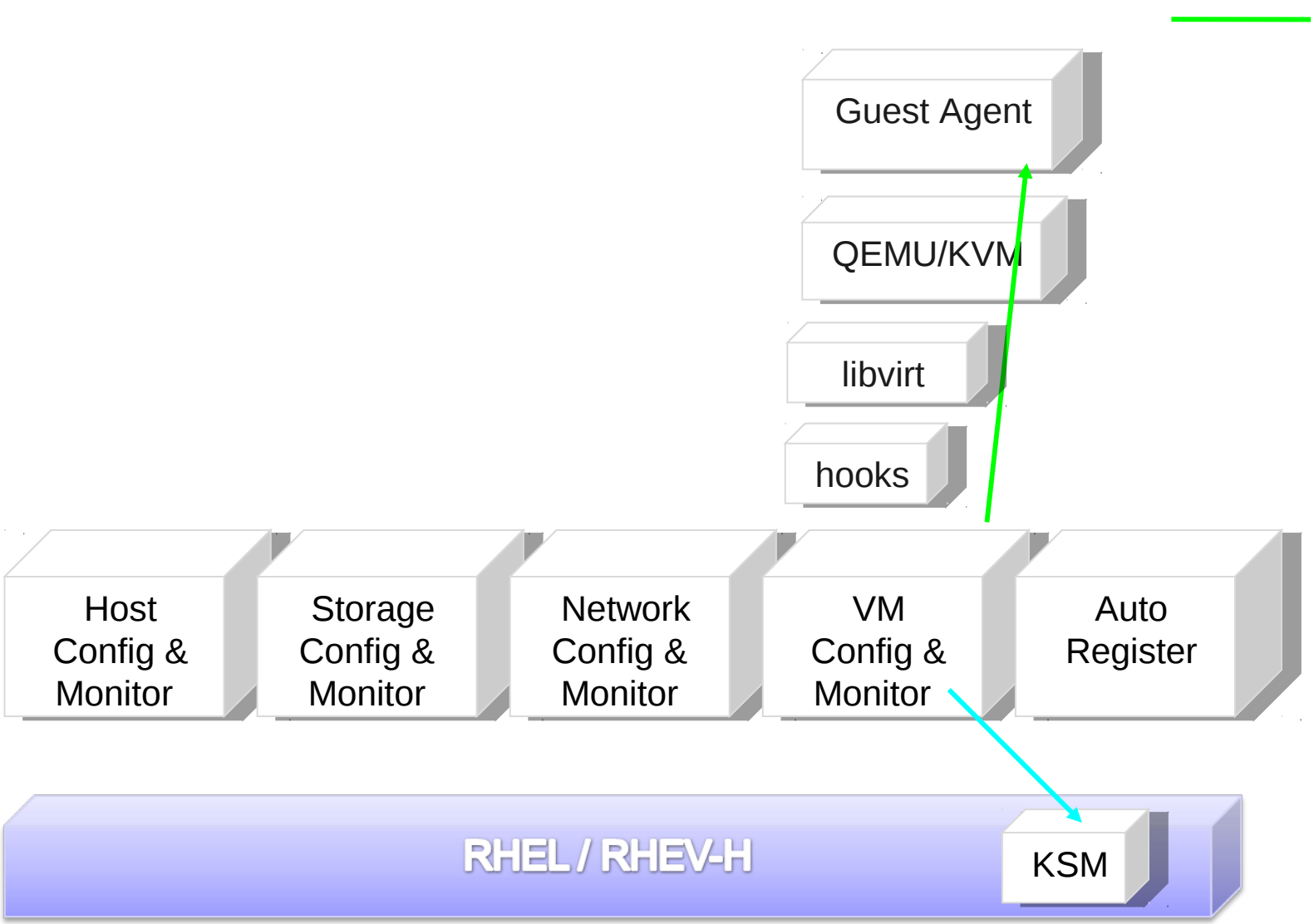
RHEV-M Guest Agent - SSO for RHEL



oVirt Host Agent - VDSM

- Covers all functionality required by oVirt Engine
- Configures host, networking and shared storage
- Uses libvirt for VM life cycle operations

oVirt Host Agent - VDSM



oVirt Storage

- VDSM manages a Storage Pool, comprised of Storage Domains
- **Storage Pool -** a VM repository that contains meta data about storage domains, storage tasks, VMs, locks, etc.
- **Storage Domain** - a disk image repository
- **Disk Image -** a collection of volumes (chain of snapshots)
- **Volume** - stored as files in NFS, and as Logical Volumes for FC/iScsi
- Thin provisioning for SAN supported (storage mailbox based)

Storage Pool Manager

The SPM runs on an arbitrary host (chosen by oVirt Engine)

oVirt Engine requires SPM to be running in order to add storage

If SPM host dies/disappears, RHEV-M causes SPM to start on a different host

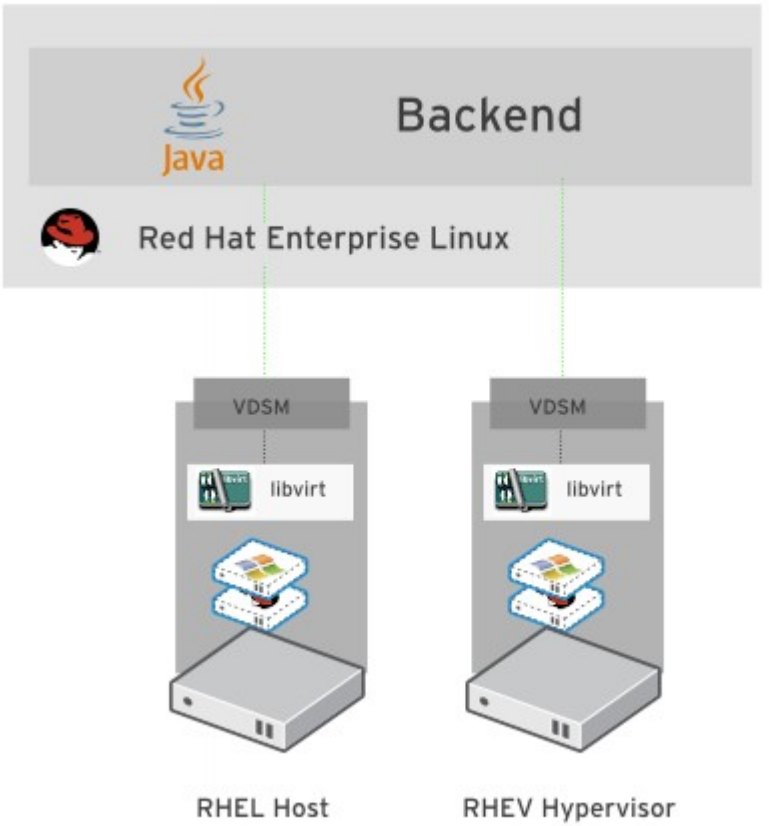
oVirt Storage “Clustering”

- A Storage Pool is implemented as a managed cluster
- Manager is oVirt Engine, running on a node external to the hosts using the storage pool
- Heartbeats and fencing are used in case of node failures
- Storage based leased locks used as another layer of protection
- Clustering wise - VMs are mostly single reader/writer - locks mostly needed to handle failures
- Can easily create a cluster of >100 nodes

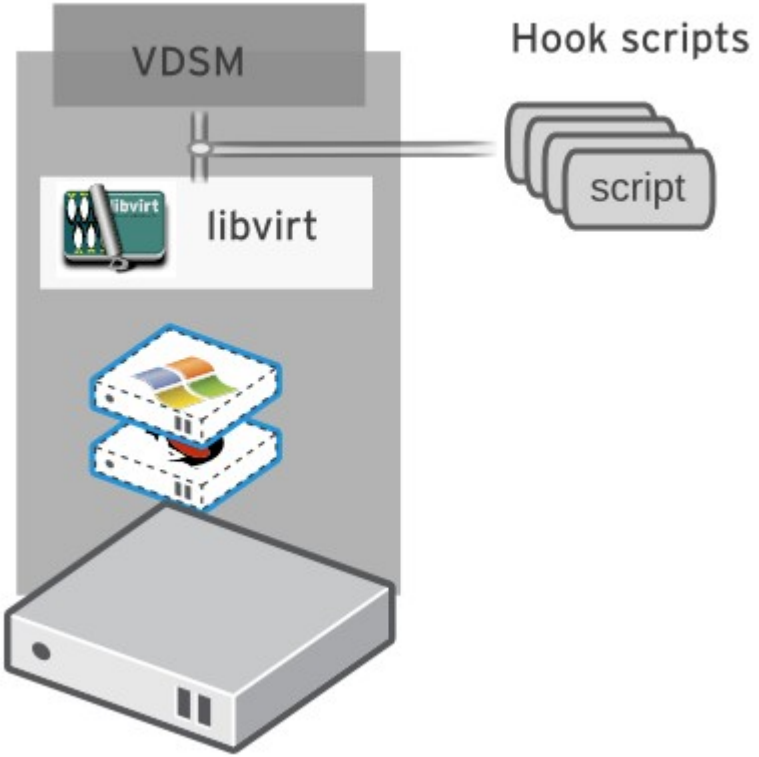
Hooks

- “Hook” mechanism for customization
  - Allows administrator to define scripts to modify VM operation
    - eg. Add extra options such as CPU pinning, watchdog device, direct LUN access, etc
  - Allows oVirt to be extended for new KVM features before full integration is done
  - An easy way to test a new kvm/libvirt/linux feature

Hooks


Java
Backend
Red Hat Enterprise Linux
VDSM
libvirt
VDSM
libvirt
RHEL Host
RHEV Hypervisor

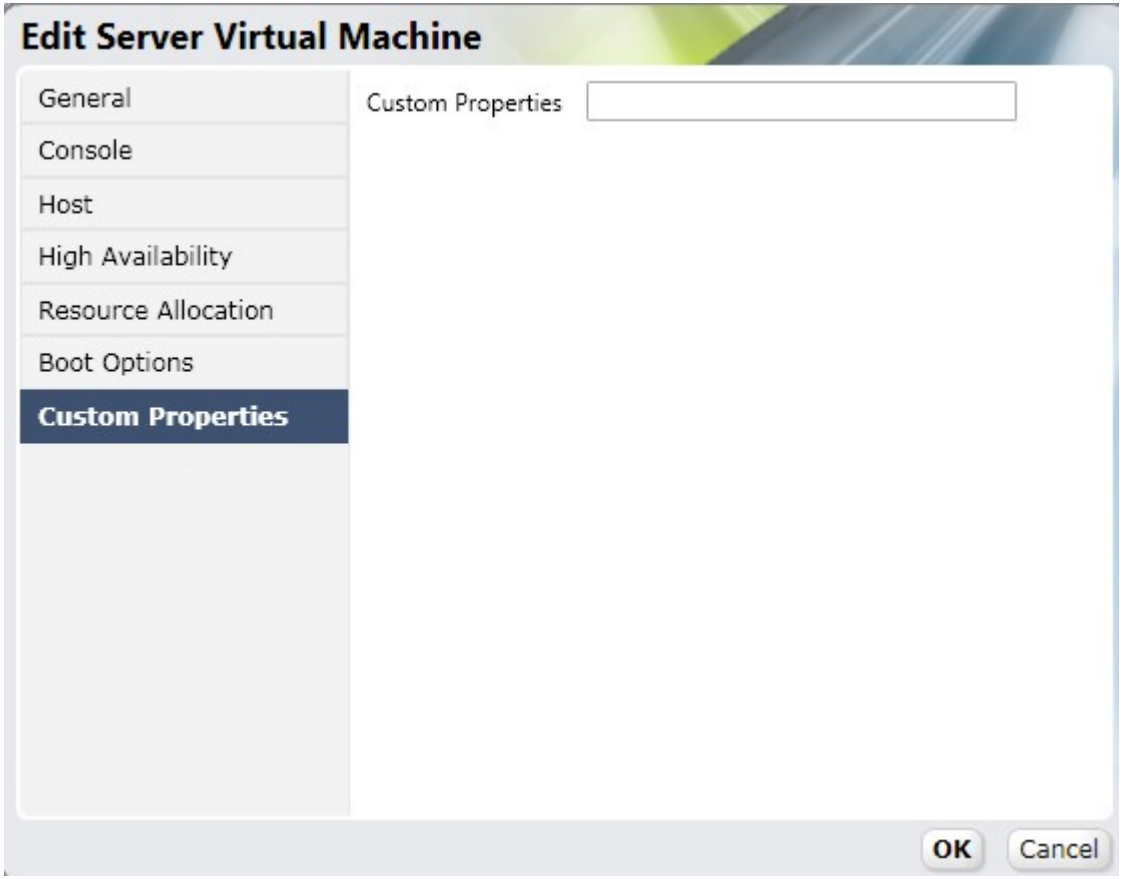
Hooks



Hooks

- Hook scripts are called at specific VM lifecycle events
  - VDSM (management agent) Start
  - Before VM start
  - After VM start
  - Before VM migration in/out
  - After VM migration in/out
  - Before and After VM Pause
  - Before and After VM Continue
  - Before and After VM Hibernate
  - Before and After VM resume from hibernate
  - On VM stop
  - On VDSM Stop
- Hooks can modify a virtual machines XML definition before VM start
- Hooks can run system commands – eg. Apply firewall rule to VM

Hooks


Edit Server Virtual Machine
General
Console
Host
High Availability
Resource Allocation
Boot Options
Custom Properties
Custom Properties
OK
Cancel

Hooks

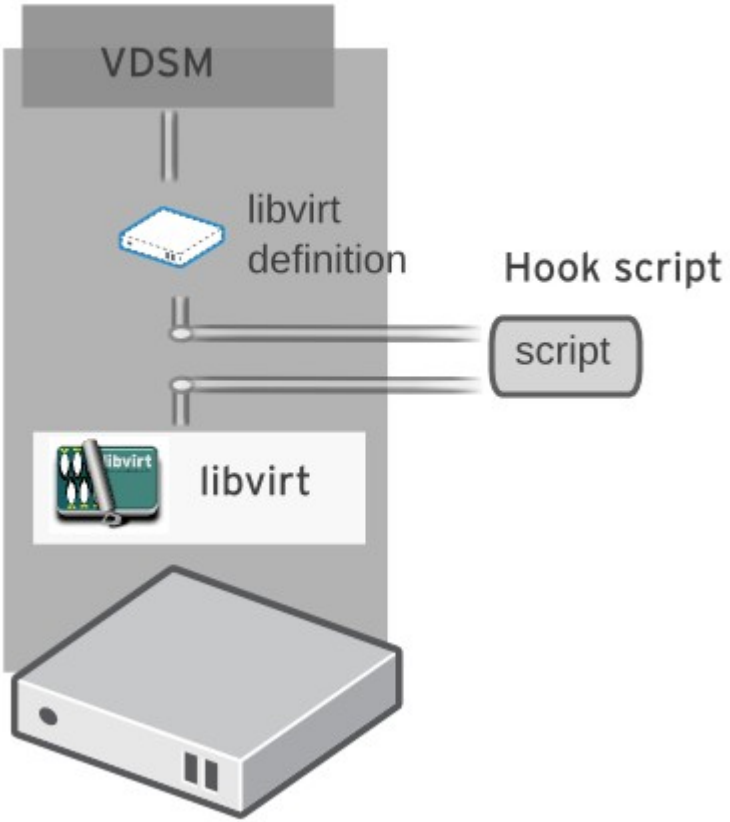
Hooks installed in /usr/libexec/vdsm/hooks

```
[root@host1 ~]# cd /usr/libexec/vdsm/hooks/
[root@host1 hooks]# ls -l
total 68
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 after_vdsm_stop
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 after_vm_cont
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 after_vm_dehibernate
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 after_vm_destroy
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 after_vm_hibernate
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 after_vm_migrate_destination
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 after_vm_migrate_source
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 after_vm_pause
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 after_vm_start
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 before_vdsm_start
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 before_vm_cont
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 before_vm_dehibernate
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 before_vm_hibernate
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 before_vm_migrate_destination
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 before_vm_migrate_source
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 before_vm_pause
drwxr-xr-x. 2 root root 4096 Apr 12 03:55 before_vm_start
[root@host1 hooks]# _
```

General | Virtual Machines | Network Interfaces | Host Hooks | Permissions

Event Name	Script Name	Property Name	Property Value
before_vm_start	10_faqemu	md5	2c352c04ecf994

Hooks



Hooks

```
#!/usr/bin/python

import os
import sys
import hooking
import traceback

'''
pincpu usages
=============
pincpu="0" (use the first cpu)
pincpu="1-4" (use cpus 1-4)
pincpu="^3" (dont use cpu 3)
pincpu="1-4,^3,6" (or all together)
'''

if os.environ.has_key('pincpu'):
    try:
        domxml = hooking.read_domxml()

        vcpu = domxml.getElementsByTagName('vcpu')[0]

        if not vcpu.hasAttribute('pincpu'):
            sys.stderr.write('pincpu: pinning cpu to: %s\n' % os.environ['pincpu']);
            vcpu.setAttribute('cpuset', os.environ['pincpu'])
            hooking.write_domxml(domxml)
        else:
            sys.stderr.write('pincpu: cpuset attribute is present in vcpu, doing nothing\n')
    except:
        sys.stderr.write('pincpu: [unexpected error]: %s\n' % traceback.format_exc())
        sys.exit(2)
```

```
<domain type='kvm'>
<name>win2k8r2</name>
  <uuid>aaa123ed-dce2-02ce-c90f-836e37cbaaa2</uuid>
  <memory>5242880</memory>
  <currentMemory>5242880</currentMemory>
  <vcpu cpuset='0'>1</vcpu>
```

Sample Hooks

- CPU pinning
- SR/IOV
- Smart card
- Direct LUN
- Hugepages
- Promiscuous mode network interface
- Cisco VN-Link
- Fileinject
- Floppy
- Hostusb
- Isolatedprivatevlan
- Numa
- Qos
- Scratchpad
- smbios

On the Horizon - Infra

- Engine – JBoss AS 7, modular lighter engine
- Engine – custom hooks
- Engine – vdsm communication protocol and transport
- API – non admin api
- Reports – integrated in web admin
- Code cleanups, refactoring, unitests, etc

On the Horizon - Features

- Live snapshots
- Live storage migration
- Quotas
- Hot plug
- Multiple storage domains
- Shared disks
- iScsi disk
- Shared file system support
- Storage array integration
- Qbg/Qbh
- virt-resize, pv-resize
- Progress bars
- Stable pci addresses
- Network types
- Backup API
- SLA
- SDM
- Many many more...

THANK YOU !

http://www.ovirt.org