Fedora Cumulus

Commodity Cloud Computing

A joint project between the oVirt virtualization and Fedora teams.

Mike McGrath <mmcgrath@redhat.com>

Jan Mark Holzer <jmh@redhat.com>

Bryan Kearney <bkearney@redhat.com>

Perry Myers <pmyers@redhat.com>

Hugh Brock <hbrock@redhat.com>

Michael DeHaan <mdehaan@redhat.com>

Brian Stevens <btstevens@redhat.com>

Document version: 0.3

Date: 2008-10-31

Table of Contents

Abstract

Cloud Computing is sort of a buzzword in technology at present. Red Hat itself has no cloud computing solutions to offer. This project aims to put together a proof of concept of such a product.

The goals are to understand the operational aspects involved in running a cloud and to identify areas where Red Hat can add value by either better/tighter integration with existing product capabilities, or by starting new developments.

Once the basic infrastructure is in place and some level of self service is implemented it can also be used to invite partners to participate in various areas of operating and building a cloud or even running their own speed tests through demos, etc.

This project itself will not produce a sellable product but will produce the blueprints for a product. The deliverables are a cloud known as Fedora Cumulus and a white paper on how it was built.

> Cumulus – A fairweather cloud that, with the right conditions, can turn generate an incredibly disruptive force.

The management software used to control the cloud will be oVirt. Part of this project is designed to harden this software for use in enterprise environments. We will take volunteers and developers from the community and give them access to guests on the cloud as well as to part of the oVirt interface. The cloud will be advertised as shell and work space, a place to do proof of concepts or development. We will not be allowing people to run production level sites on this cloud, at least not at first. This will allow us to schedule downtime to make drastic changes to the cloud while we're developing and not set a false impression of availability to our users. By using actual people we're putting actual load on the cloud and not generated load.

Introduction

Fedora

Fedora does not shy away from adopting new technologies early and helping usher them to maturity. Partnering with the oVirt team should prove beneficial to both teams. We will want all of the oVirt bits into Fedora prior to the actual roll out as it is a Fedora Infrastructure policy. The appliance itself doesn't have to be there but all the bits required to build it need to be.

oVirt

oVirt[1] is a newcomer to the virtualization field and combines many other Red Hat backed projects including Cobbler, FreeIPA and libvirt. Red Hat engineers have designed most of these technologies. oVirt is to provide an open standards based management platform for enterprise "clouds". With its management API being based on libvirt it is (too a large extent) HyperVisor agnostic and leaves the possibility open for other virtualization technologies to be managed using a uniform interface.
There is also a possibly to use oVirt as a management interface into Amazon's AWS/EC2 environment.

Technical Details

Terms

Cloud – An abstraction layer for an architectural pattern that allows virtual cpu and storage resources to be provided as an on-demand service. Users of the cloud will also be able delegate operational tasks and privileges to other users in the same cloud user group (Company, Division, Group, individual user)

Cloud computing - An approach to IT that involves the creation and deployment of services and applications over the internet, thereby providing location agnostic/independent compute resources supported by a centralized computing infrastructure (with an opportunity to connect and merge clouds over time)

Virtualization – For the purposes of this document virtualization refers specifically to KVM and hardware virtualization.

Guest – A virtual host running on an oVirt node. (KVM based hypervisor)

Node – The physical machine that the guests run on.

User – For the purposes of this document a user is someone who would log in to a guest and use it. Users may also log in to ovirt to manage their subset of hosts, but not manage the cloud or underlying hardware. Users in the cloud will also be able delegate operational tasks and privileges to other users in the same cloud user group (Company, Division, Group, individual user)

oVirt Administrator – Admins will be logging in to the physical hardware and storage to manage it. The Admin will also be responsible for the cloud itself using oVirt to manage users and who gets what access to what users.

Shared Storage – Block level shared storage. Users can create nfs shares and the like on their own, but when we say shared storage we are referring specifically to block level devices.

1 http://ovirt.org/

Hardware

Virtualization Hardware

The focus of Fedora Cumulus at this time is entirely based around commodity hardware. For our chosen virtualization servers we are picking the IBM x3550 1U[2]. These servers can be fitted with two quad core processors giving us 8 physical CPU cores per machine. These Xeon processors all have the vmx flag and can support hardware virtualization. They can take up to 32G of ram and allow for 2 expansion PCI-E slots. Additionally Fedora Infrastructure has standardized on these machines for most future purchases and the team will be familiar with them. Red Hat has a good contract with IBM and CDW, the service is top notch and we get better then normal pricing. Often purchases can be grouped with other teams to further reduce costs.

Storage Hardware

We want as close to SAN functionality without the SAN price. To do this we'll be purchasing commodity hardware and using network block device software. The hardware itself will likely be two x3550's similar to our virtualization hardware but each with an externally attached scsi storage tray or from two x3650's[3] that is filled with internal storage. The x3650's are 2U servers that can take up to 8 2.5” SAS drives. We'll take a measurable performance hit by using this over a SAN but the actual impact on our target users will probably not be noticeabl to most. Storage will be largely dependent on exactly how many guests we're looking to support.

Storage servers, at a minimum, will be setup to do live replication via iscsi and software raid (write-mostly option). At best we'll have a highly available iscsi solution. The implementation details here have not been completed.

Network

In the ideal setup we would have three switches:

- One switch dedicated to storage.
- One switch dedicated to management, live migration, etc.
- One switch dedicated to traffic the guests produce.

The first two switches could easily be combined in to one switch. Each of the servers we are purchasing come with two network interfaces. In order to go to three switches we'd need to purchase additional cards. Some further investigation on this is needed.

Cloud Implementation

Each of our x3550 servers (or virtualization servers) will be set up in a diskless fashion via pxeboot. Ideally we'll be building custom 'appliance node' that will have full Fedora Account System[4] connectivity and have Fedora's monitoring tools. This will add some additional overhead to traditional installs which can automatically be updated and managed like the rest of the environment. Each update to the operating system (configuration or package / security updates) will require a rebuild of the appliance. The system itself will be placed in Fedora's “Value Added” type of servers meaning Fedora

2 http://www-03.ibm.com/systems/x/hardware/rack/x3550/index.html
3 http://www-03.ibm.com/systems/x/hardware/rack/x3650/index.html
4 https://admin.fedoraproject.org/accounts/

Cumulus will not be affected by Fedora's release cycle. It also means that Fedora Cumulus will be put at a lower priority then Fedora's critical path.

Our primary storage servers, to be used to store our actual guests, will also be managed by the Fedora Infrastructure team with puppet. It will have an operating system installed on a small portion of its local disks. The rest of the disks will be combined using RAID 6 and exported via iscsi[5] or AOE[6]. At present iscsi is the front runner as Red Hat has backed it in the past, Fedora Infrastructure has used it for years and it already has support in oVirt. oVirt will not manage these boxes but will manage the iscsi interface they expose.

Full implementation of storage is TBD

Each of our virtualization servers will then mount the storage server. This block device will be separated into multiple logical volumes using lvm. oVirt will manage the lvm side of things including proper locking so clvm isn't needed. See figure 1.0 for more information, it is subject to change.

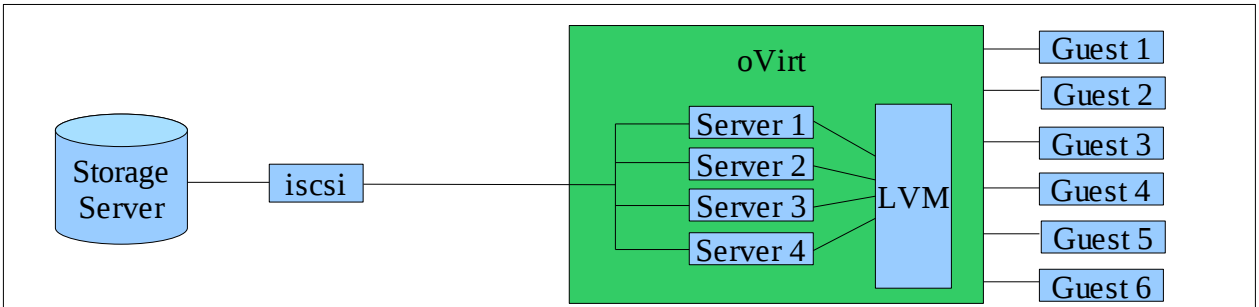


Figure 1.0

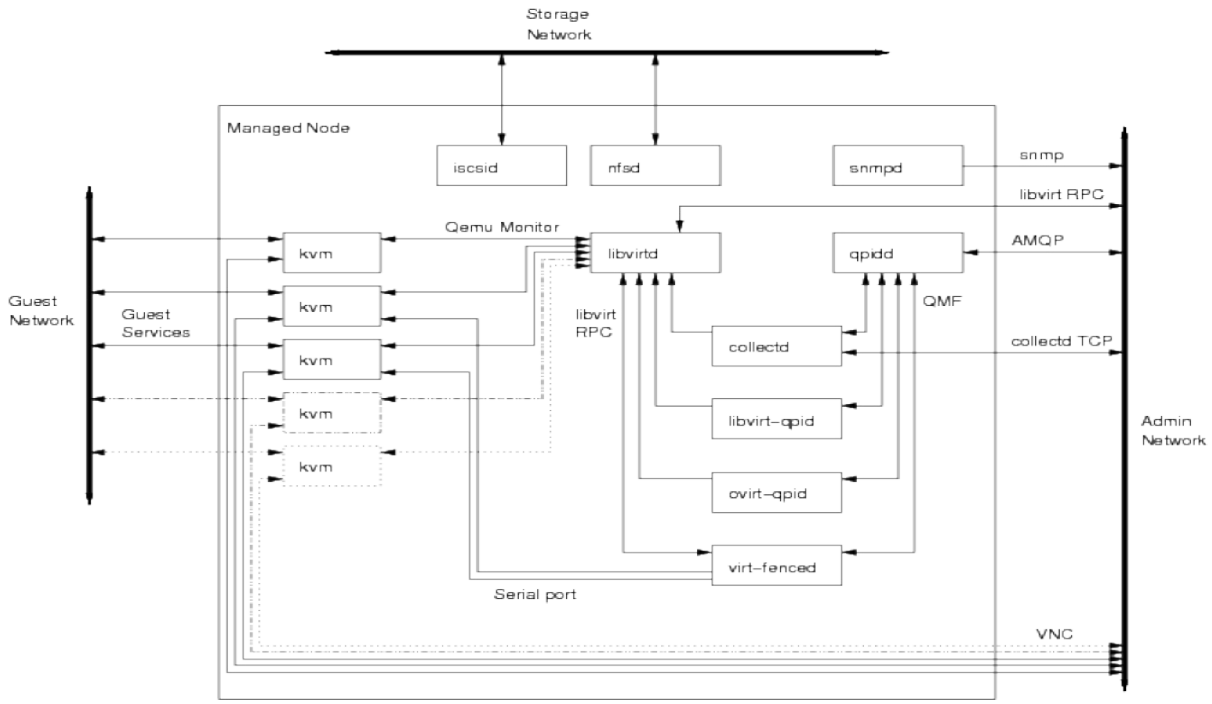
oVirt

oVirt is very much a wildcard in this project. It is still under aggressive development. There are concerns (see below) about whether or not oVirt can do a full implementation as designed above. It is released as an appliance image which should work on our servers out of the box. The oVirt milestones page seems to indicate it is possible to install oVirt in a non-appliance fashion but is not yet documented. This would be ideal.

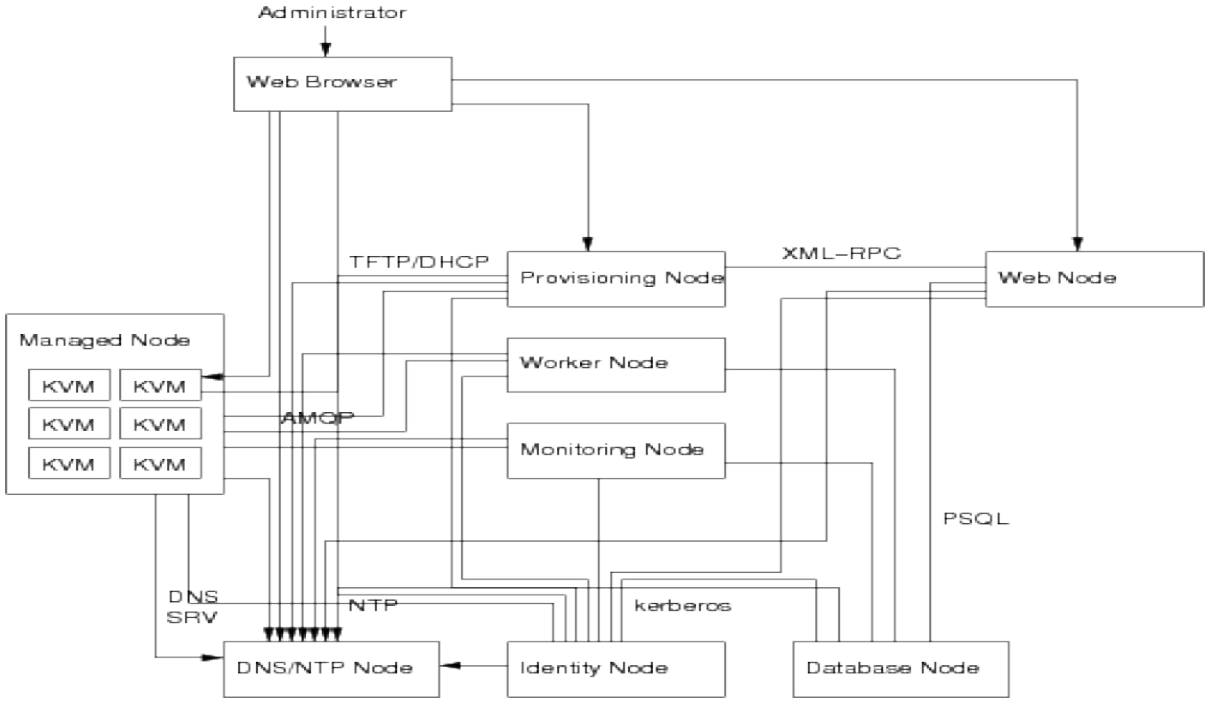
The Fedora Infrastructure team will get the initial ovirt install up and running. This includes installation of the oVirt Server Suite node (admin node). We'll be adding some additional users to a newly created sysadmin-ovirt group that will allow us to give oVirt developers and other interested parties access to these machines to help troubleshoot issues.

5 http://www.open-iscsi.org/
6 http://sourceforge.net/projects/aoetools/

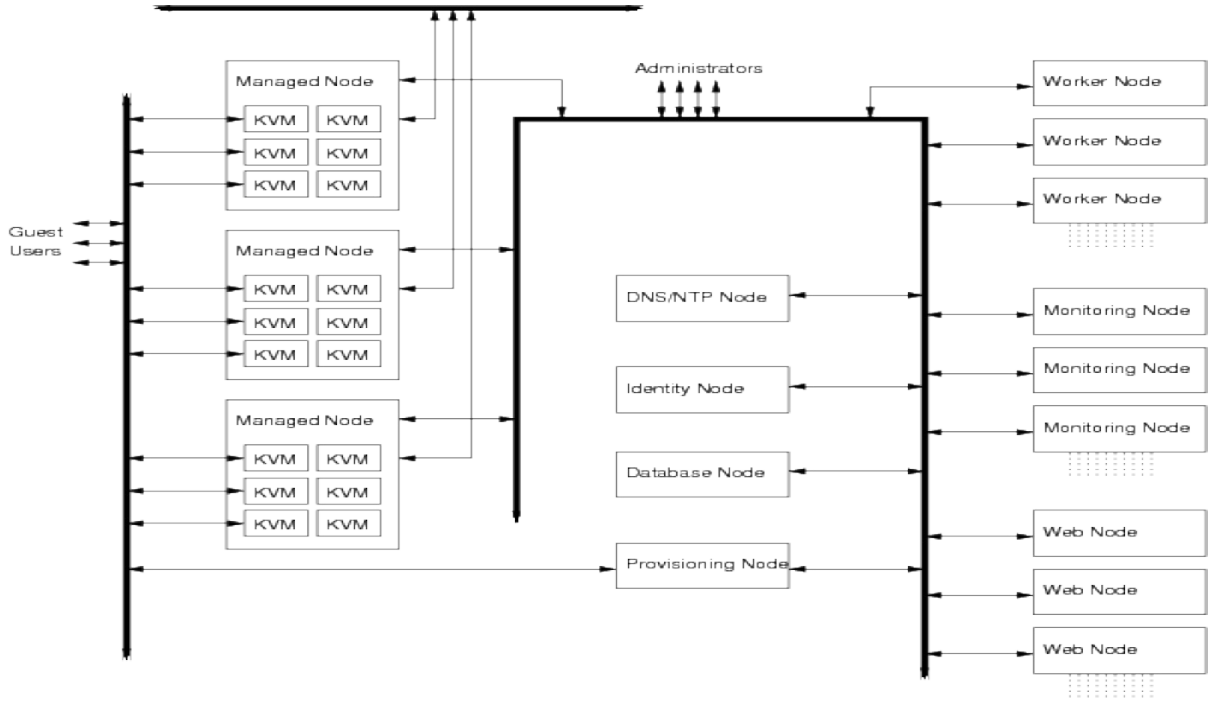
oVirt node architecture



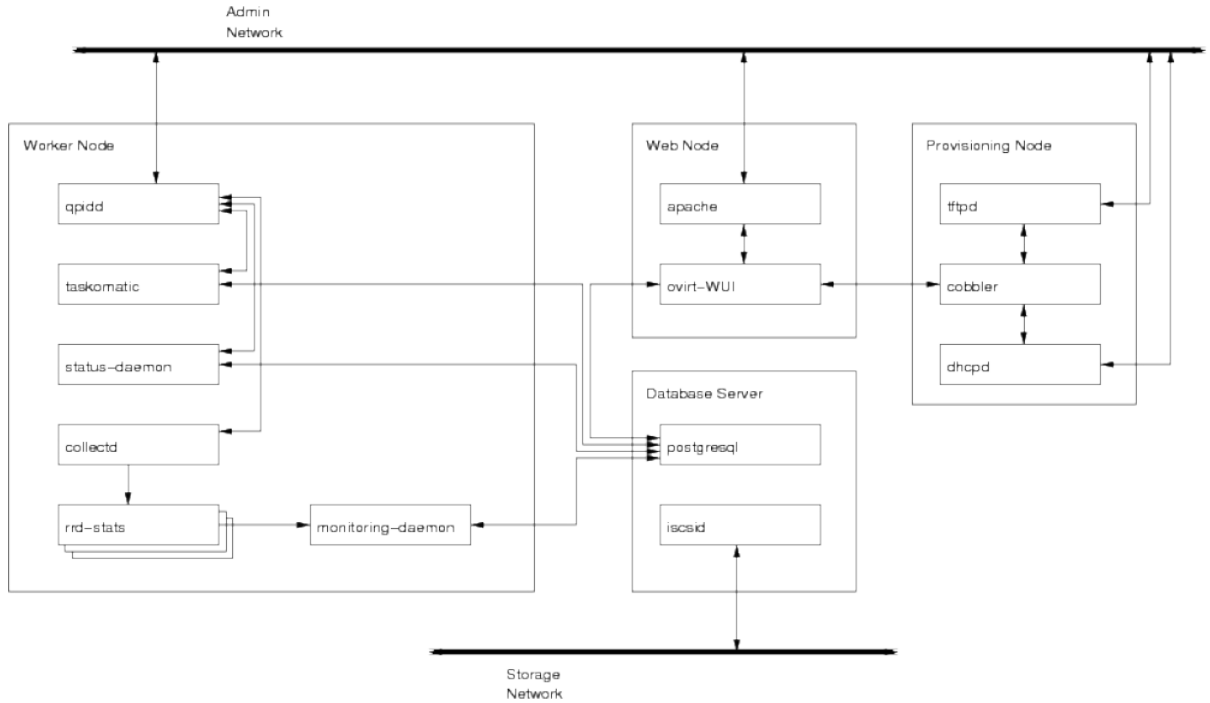
oVirt physical architecture



oVirt logical architecture



oVirt admin architecture



Timeline

- Suggested oVirt release / Feature freeze late December – Early January.
- Budget Approval (TBD)
- Quotes 2-4 days
- Purchase / shipping 7-10 days
- Installation 20-40 days (PHX2 is a particularly busy place right now, better to estimate high. Red Hat's IT team will do this, likely Jonathan Pickard. If there's budget to send Mike McGrath to do it that may speed things up)
- OS installation and storage configuration 2 days
- oVirt installation and configuration 5 days
- Initial small group deployment for testing 14 days
- Final public rollout and announcement

Use Cases

The use cases below describe offerings that Fedora Cumulus will offer. Many can be re-branded into other types of solutions to directly compete with other commonly found products.

1. Single Guest – A user has requested a stand alone guest. The user may log in to this guest and have root access.
    - An admin will create the guest, the user will verify access and at that point be in charge of all maintenance of the guest.
    - No username will exist in the oVirt interface.
    - Troubleshooting of this guest will be done by the Admins for booting or network issues.
2. Multiguest – A user or group will have access to multiple guests including access to the oVirt interface. The user will only have access to guests that belong to their multiguest group. Users will have root access on this guest. This may be marketed as “team lead manages a group of guests for their team”
    - An admin will create the guests and delegate access to the user.
    - User will have access to ovirt and will be charged with its maintenance. This includes rebooting, etc.
    - Admins will decide how many resources are available to the user and allow them to manage it themselves. For example, 1G RAM per guest, but allow the User to give 1.5G to one guest and only .5G to another.
3. Cluster – A user will have access to multiple guests including access the oVirt interface. This offering also includes some shared storage solutions. Users will only have access to guests that belong to their multiguest group. Users will have root access on their guests.
    - An admin will create the guests and delegate access to the user.
    - User will have access to oVirt and will be charged with its maintenance. This includes rebooting, etc.

- Admins will decide how many resources are available to the user and allow them to manage it themselves. For example, 1G RAM per guest, but allow the User to give 1.5G to one guest and only .5G to another.
- Admins will create shared storage devices and work with users to get them deployed. The user will use clvm or gfs or whatever clustered storage solution they'd like.

4. Managed – Users will have a stand alone guest. The user may log in to this guest but not have root access to it. A defined set of services will be provided and made available upon request.
    - Services TBD but likely include blogs, wikis, etc.
    - Services to be configured via puppet. User will not have a view into this, typical access will be via a web page.
    - Admins will be in charge of full troubleshooting of the guest in the event of issues not related to service configuration.
5. MRG[7] – Grid solution – Re-branded cluster solution.
    - An admin will create the guests and delegate access to the user.
    - User will have access to oVirt and will be charged with its maintenance. This includes rebooting, etc.
    - Admins will decide how many resources are available to the user and allow them to manage it themselves. For example, 1G RAM per guest, but allow the User to give 1.5G to one guest and only .5G to another.
    - Admins will create shared storage devices and work with users to get them deployed. The user will use clvm or gfs or whatever clustered storage solution they'd like.
    - Specially designed for users wanting to use an MRG grid.

White Paper

A large part of this project is not to prove that it can be done, but to show that others can do it. Writing a white paper will allow for this. We will focus on simplicity in our implementation and in the white paper. When we see 3/6ths, we need to reduce it to 1/2. Keeping large environments like this simplified will be make or break for Red Hat's future in cloud computing. This is especially true while Red Hat is not in the cloud hosting business. If it seems complicated to make and manage a cloud, it is more likely that organizations will just pay someone else to do it.

The white paper will provide full implementation details including what hardware we used, what software, etc. We will also run some basic performance tests on disk IO, computational details, etc. I've not seen many published metrics about stolen time and how busy guest A affects busy guest B when they are hosted on the same physical server.

The whitepaper(s) could also be used as implementation guides for GPS in customer engagements

We probably also want to involve the reference architecture group (Vijay Trehan) to document the build of the infrastructure and cloud. They have resources dedicated to these efforts and have already created a number of architecture documents for EC2, virtualization/HA etc ...

As we deploy the cloud and learn we should also document specific configuration and application use cases (ie how to implement a database guest, compute guest , web server guest/farm etc.)

7 http://www.redhat.com/mrg/

Given the open nature of the project and number of people involved we could also consider a cloud blog which we update on a regular basis

Concerns / Assumptions

Fedora has successfully used virtualization for years. This particular project is unique in that it will provide a comprehensive management solution in the middle of everything. Below are a list of questions, predicted pitfalls and other such concerns that need to be addressed.

1. FAS integration. FAS (Fedora Account System) is a custom built accounting system. The oVirt appliance uses kerberos by default. This seems to be bypassable at the Apache layer using mod_auth_pgsql. It is unclear what side effects this will have.
2. oVirt storage management.
3. oVirt's web interface is still designed to be run locally. When run remotely via a proxypass or other such methods, it sometimes misbehaves or acts very sluggishly. Some of this may be fixed in the next version.
4. It is still unclear if this project is officially approved. If it is, exactly how much budget has been approved for it. This concern will take care of itself but it's worth tracking.
5. The milestones listed from oVirt's[8] site do not list any actual time frames or dates. This makes it difficult to determine when full readiness will be available.
6. IP address space is still questionable. Getting a small group of IP's is easy, getting hundreds is more difficult. This is not a blocker to implementation but without it usability will be difficult as people would have to ssh either from inside Red Hat or through an externally available gateway server.
7. Legal agreement. I'll work with Tom (Fedora's Legal liaison) to come up with some text for everyone to agree to that keeps Fedora / Red Hat out of liability.
8. Worry of vendor lock (forced architecture)
9. Lack of oVirt management UI capabilities as identified in our first meeting , need oVirt UI designer
10. Overall priority of requirements coming from the cloud project into the oVirt (and other) team
11. User logins will be done via weak authentication – Username / password or via ssh key.
12. Will users log in and create their own guest once admins have assigned RAM, disk space, etc. Or will admins create the guest? Perhaps offer both? Something to discuss further.

Management

Much of this project is in implementation details and engineering. The project itself is small enough that it only involves a few people in a couple of teams. At worst we'd need some conflict resolution between teams. At a minimum we'd need general oversight via progress reports, demos, etc.

More resources to the oVirt team may help ease some pains we'd see in implementation and bug reporting / feature requests. This is completely out of scope for the project itself but would have a positive impact on oVirt as a product. Some blocking items from other teams include IPA's lack of cert management. This means oVirt does not have cert management and has complicated things.

8 http://ovirt.org/milestones.html

Future Solutions

Once this project gets going we should consider more broad implications of becoming a cloud aware company. This includes proper marketing materials, sales pitches, training, etc. All of which are far outside the scope of this project. Management will determine based on the success of this project how viable it is for the broader Red Hat family of products.

Budget

Below are several of the options we've looked at. The san solutions are still listed though are largely not being considered because of a preference for commodity hardware. It has been noted that Red Hat's IT department may be able to pay for some of these items. These details have yet to be worked out.

Facilities

Hosting space, power, etc is all provided by Red Hat. This space costs someone, but it won't be us. This also includes bandwidth, routers and other related network equipment. The exception to this may be switching equipment. We'll need to discuss that with the network team and see if they want us in their core switching system or a dedicated one.

Option 1 (SAN):

- Blade Center x 14 blades
- 16G RAM
- 1x146G HD
- SAN connectivity
- 2 quad core processors
- Total bladecenter cost: $98,808+tax/shipping
- SAN cost $15K/Terabyte
- Total Cost: 2T + blades = $128,808+tax/shipping
- Host estimate: 224 (1G RAM/host)

Option 1.1 (SAN):

- Blade Center x 5 blades
- 16G RAM
- 1x146G HD
- SAN connectivity
- 2 quad core processors
- Total bladecenter cost: $45,456+tax/shipping
- SAN cost $15K/Terabyte

- Total Cost: 1T + blades = $60,456+tax/shipping
- Host estimate: 80 (1G RAM/host)

Option 1.2

- Blade Center + 5 blades
- 16G
- 1x146G HD
- Storage via iscsi:
    - 2X IBM X3650 (2U)
    - 4T usable space
    - $13,452.00+tax/shipping
- Total Cost: 5 servers + storage = $40,768.05
- Host estimate: 80 (1G RAM/host)

Option 2 (SAN):

- Blade Center x 14 blades
- 32G RAM
- 1x146G HD
- SAN connectivity
- 2 quad core processors
- Total bladecenter cost: $117,012+tax/shipping
- SAN cost $15K/Terabyte
- Total Cost: 4T + blades = $177,012+tax/shipping
- Host estimate: 448 (1G RAM/host)

Option 2.1 (SAN):

- Blade Center x 5 blades
- 32G RAM
- 1x146G HD
- SAN connectivity
- 2 quad core processors
- Total bladecenter cost: $51,957+tax/shipping

- SAN cost $15K/Terabyte
- Total Cost: 1T + blades = $66,957+tax/shipping
- Host estimate: 160 (1G RAM/host)

Option 3:

- IBM X3550 (1U)
- 16G RAM
- 2x73G HD
- No San
- 2 quad core processors
- Cost / server = $4,876.00+tax/shipping
- Storage via iscsi:
    - 2X IBM X3650 (2U)
    - 4T usable space
    - $13,452.00+tax/shipping
- Total Cost: 14 servers + storage = $81,716.00
- Host estimate: 224 (1G RAM/host)

Option 3.1 (SAN):

- IBM X3550 (1U)
- 16G RAM
- 2x73G HD
- SAN cost $15K/Terabyte
- 2 quad core processors
- Cost / server = $7,286.00+tax/shipping
- Total Cost: 14 servers + 2T storage = $132,004.00
- Host estimate: 224 (1G RAM/host)

Option 3.2:

- IBM X3550 (1U)
- 16G RAM
- 2x73G HD

- No San
- 2 quad core processors
- Cost / server = $4,876.00+tax/shipping
- Storage via iscsi:
    - IBM X3650 (2U)
    - 4T usable space
    - $13,452.00+tax/shipping
- Total Cost: 5 servers + storage = $37,832
- Host estimate: 80 (1G RAM/host)

Option 4:

- IBM X3550 (1U)
- 32G RAM
- 2x73G HD
- No San
- 2 quad core processors
- Cost / server = $6,275.00+tax/shipping
- Storage via iscsi:
    - 2X IBM X3650 (2U)
    - 4T usable space
    - $13,452.00+tax/shipping
- Total Cost: 14 servers + storage = $101,302.00
- Host estimate: 448 (1G RAM/host)

Option 4.1 (SAN):

- IBM X3550 (1U)
- 32G RAM
- 2x73G HD
- SAN cost $15K/Terabyte
- 2 quad core processors
- Cost / server = $8,685.00+tax/shipping
- Total Cost: 14 servers + 4T storage = $181,590.00
- Host estimate: 448 (1G RAM/host)