

Managed Clouds and Why SMB Customers Don't Chase Unicorns

If we look at the economy and distribution of wealth, we can see that there are some interesting statistics about what many call the 1%. Stretch a little further into the top 10% and you will find what they refer to as the champagne glass effect. In other words, the top percentile of earners have a disproportionate percentage of the overall wealth. So what does this mean to what I'm talking about here?

Is There a Champagne Glass Effect in IT?

Gartner has a well laid out graph to illustrate the overall spend of technology that is available here: <http://www.gartner.com/technology/metrics/>



source: <http://www.gartner.com/technology/metrics/>

This is less of the illustration of a champagne glass effect with the graph than it is showing the distribution of spending by industry. As you can imagine, and as could be correlated by other data, there are certain industries and company sizes that spend more on services and IT resources.

What is different about the spending at the "enterprise" level, is that it is often done on hardware, software, and cloud resources, with a strong reliance on internal IT resources to manage those resources. While we often focus on those enterprise success stories with high trust and amplified feedback loops between the IT and business teams, what about the rest of us?

Small to Medium Business: The 99%

In using the economic champagne glass as the example to illustrate the 1% versus the 99%, we can take the same view of technology spending and reliance on internal resources for technology development and management. I don't want say that everything in the SMB space can be painted with the same broad brush, but there are hundreds upon thousands of businesses ranging in size, and they will consume and provision IT resources much differently than the 1%.

We can even expand to the 90%/10% range really and still capture a lot of environments which are still heavily invested in the traditional IT model.

SMB Doesn't Chase Unicorns

The elusive unicorn of IT is that internal IT staffer, or contractor, or even an entire team of people who are working magic in the technology backing the organization. That magic, is really not magical at all. But the ability to attain that seamless end-to-end success in providing IT resources to meet business needs is just as elusive as a unicorn when you aren't staffed to be able to handle the tasks.

Cloud is often touted as the answer. It isn't. Cloud is an enabling technology and methodology to deploy business applications into, but requires the ability to create, deploy, and manage these applications into that cloud environment. With many companies (especially SMB) this is as elusive as

the fabled unicorn that so many seem to hold in high regard. So, what is the SMB customer supposed to do when they aren't prepared to staff up with internal resources in hopes of building these environments?

Managed Cloud

To many cloud pundits and DevOps advocates, this is the antithesis of cloud. That being said, I'm a cloud advocate, but I also have the view of being inside the SMB market through experience and through my exposure in social networks, conferences, and various community organizations.

If managed cloud is often frowned upon by many leaders in the cloud industry, the question is why?

As a fan and customer of [Rackspace](#), I have really loved their choice to heavily leverage the managed cloud as a key offering. For many SMB customers, this is precisely what we need. Some regard it as training wheels to get into the cloud, and if that is the case then I don't think that is such a bad thing.

I've made a note of the [great article by John Engates, CTO of Rackspace, here at VirtualizationSoftware.com](#) and through conversations with other Rackers like Cody Bunch and Ken Hui (recently moved to EMC), you can see that the presence at VMUG conferences is really highlighting their plan to show their distinct capability to run big VMware and OpenStack clouds for their customers.



Mike Kavis at Cloud Technology Partners wrote an article on the move by Rackspace here that talks about the concept and the shift towards more managed provisioning versus the panacea of self-provisioned IaaS.

More companies are rising up with the approach of bringing managed services as an offering to differentiate themselves from the big IaaS (Infrastructure-as-a-Service) providers like cloud juggernaut Amazon, as well as Google and Microsoft who are climbing the market share charts quickly behind them. It isn't that IaaS isn't good, but the point is that the market for IaaS business is shrouded by these three major players.

SMB + High Touch = High Trust

The path to adoption of more cloud resources and DevOps practices in the SMB market place is one that will be paved by the managed cloud route in my opinion. Organizations which have not got the resources nor the experience to create and consume services in cloud environments will find that the managed cloud path is one that leads to comfort with the new ways in cloud IT.

Is this a guaranteed path? No. Is this right for every company? No. That's right, I said it. At this point in time and for a long time to come, public and private cloud environments, managed or self-provisioned, are not going to be appropriate for many organizations.

In a year or two we can reassess the landscape a bit and make a judgement on things then. For now, I like that managed cloud feeling. It's nice when someone else is on pager duty for you ☐

[Rackspace Private Cloud - Updated to OpenStack Grizzly Release!](#)

✘ Just in case you didn't spot this in the Twitter stream yesterday, there was some really cool news from the team at Rackspace! The Rackspace Private Cloud software is now updated with the Grizzly release of OpenStack.

(http://www.rackspace.com/knowledge_center/article/about-rackspace-private-cloud).

Dubbed the Rackspace Private Cloud v4.0, this is a great way to get your OpenStack cloud up and running with significantly less configuration required. The installation scripts will deploy onto Ubuntu 12.04 and CentOS 6.3 and includes the Chef server with cookbooks for simple management and deployment once you build your environment.

Scripts versus ISO

One of the really neat things that Rackspace did in the past was to build a standalone ISO to be able to deploy the Rackspace Private Cloud which was running the Folsom release of OpenStack. This ISO is still available, but for now, the Grizzly release is only available with the scripted deployment.

If you still want to use the standalone ISO, you can continue to use my series on deploying OpenStack with VMware Workstation ([Getting Started](#), [Setting up the All-In-One](#), [Adding Cinder](#)) and you can still download the archived versions of the ISO, and also get to the OpenCenter (version 3.0) installation through links in the Rackspace Knowledge Center - http://www.rackspace.com/knowledge_center/article/rackspace-private-cloud-software-archived-versions

What's new with this Release?

The use of Grizzly will add all of the new features of OpenStack of course, but along with that are new Chef cookbooks packaged into the build which make your continued management of OpenStack Grizzly much easier; Especially for those who are just dipping their toes into the OpenStack waters.

All of the core services are included (Nova, Horizon, Swift, Cinder, Keystone, Neutron, Glance) as well as Ceilometer which is the metering tool that is most likely slated for core project inclusion with the upcoming Havana release of the OpenStack software in Q4 of 2013.



Image courtesy of Rackspace:

http://www.rackspace.com/cloud/private/openstack_software/

HA all the way!

Another really cool feature in the Rackspace 4.0 release is the use of HA features of RabbitMQ and database services. The use of HA design in the build gives us another edge up over building from

scratch because there are some challenges for many to build out the underlying services in a multi-master configuration.

You have to ensure that your node deployments are built with HA in mind of course, but the steps saved in designing the HA database and message queuing are appreciated because of the time savings you receive from these features being pre-configured.

Thank You Rackspace for another great way to deploy OpenStack and I encourage everyone to take a look at this and reach out to the Rackspace crew. All that you have to do is to click the Download button on the Rackspace Private Cloud page

here: http://www.rackspace.com/cloud/private/openstack_software/ and you will receive contact information in email from the Rackspace support team.

[OpenStack Lab on VMware Workstation - Adding Cinder volume features](#)

✘ Now that we have our [OpenStack lab](#) up and running, we want to click around and kick the tires on all of the features. One of the first things that we will find is that the All-In-One is really an “almost” All-In-One. What I mean by that, is that there are services available which aren’t configured, and require a little more care and feeding.

Where is my Cinder?

The Block Storage feature of OpenStack is appropriately (or perhaps tongue-in-cheek) named Cinder. This service allows you to create block storage in your OpenStack cloud and attach to virtual machine instances. While the administration dashboard has the Volumes tab and appears to be all set to go, it requires a few extra steps to get Cinder volumes up and running.

Testing our Cinder volumes out of the box

Before we do anything, I wanted to show you how it doesn’t work. That sounds strange, but I’m stubborn and I like to prove things out, and if you are like me, you probably clicked through the dashboard provisioning some goodies and then realized that something wasn’t right with the Volumes page.

First, log in to your All-In-One dashboard as Demo (user account created during the deployment):



In your dashboard (aka Glance) click on the **Volumes** tab and click the **Create Volume** button:



In the Create Volume options box, fill in the name of your volume and a description first:



Next, scroll down and fill in the Size (GB) field. I'm using a 1 GB volume for my test:



In a matter of a few seconds we see the results of our volume creation:



Yup...it failed. Booooo! But that's ok, because as I mentioned at the start of the post, we have to configure Cinder and the volume space before this would work.

Adding the Cinder foundation

We need to do the following to enable and configure Cinder for our All-In-One build. These steps are:

1. Create a disk drive to store our Cinder volumes
2. Add Cinder components to the OpenStack build
3. Create the virtual storage volume and assign it to our Cinder volume pool
4. Restart the newly installed services
5. Test Cinder volume creation

Although we have a large disk assigned to the All-In-One (200 GB in my case), we still need another disk to use for the Cinder volumes. Remember that our 200 GB disk is for the virtual machines only. Additional volumes will be stored in our dedicated Cinder volumes.

For cleanliness, I'm going to shut down my OpenStack instance and add the disk to my VMware Workstation. We edit the settings and click on the **Add** button:



Choose **Disk** and click **Next**:



We choose to **Create a new virtual disk** and click **Next**:



Take the default which is **SCSI** and click **Next**:



For fun, I've set it at 40 GB and as always I choose the **Store virtual disk as a single file**:



I'll name the disk appropriately as **OPENSTACK01-cinder.vmdk** so I remember which one it is:



Now that we have our new disk added, we start up our All-In-One VM. Once we are up and running we will SSH into the machine to configure the new Cinder volume. You will need a client such as PuTTY, or whatever your SSH client of choice is.

Log in as the user you have created during the initial build. Once we are logged in, we will elevate our privileges to configure everything:



Type in **sudo su -** which puts you into the console with su privileges and the full environment. Next you type **source openrc** which loads the environment variables for our OpenStack configuration that we need for the other script processes:



Before we add new things, run the **knife node show <your node name>** which shows the results like the screen below. You see under the Run List that there is no Cinder mentioned:



Now we type **knife node run_list add <your node name> 'role[cinder-all]'** which will add the Cinder roles to the Run List:



Now we type **chef-client** which executes the cookbook recipe to create and initialize the Cinder services which are cinder-volume, cinder-api and cinder-scheduler.

This step takes a while once you start it, so you can leave it running for 5 minutes or so:



Once the installation process completed, you will see a screen like the one below with the last message being **Report Handlers Complete:**



Now we create the volume by using the **pvcreate /dev/sdb** command:



Next we create the volume group using the **vgcreate cinder-volumes /dev/sdb** command:



Finally we have to restart the Cinder services with the following commands:

service cinder-volume restart

service cinder-api restart

service cinder-scheduler restart



Phew! We did it! Now we head back to our dashboard to retry the process of creating a volume. Just as before, log in as **demo**, go to the **Volumes** tab and click on the **Create Volume** button:



In a few seconds you will see the console working away on your volume creation and finally it shows our new volume with a status of **Available**:



We did it!

Now you can deploy Cinder volumes as a part of your All-In-One lab deployment. This will come in handy later ☐