Building the Case for OpenStack – Value at Every Level

There was a particularly exciting article on the OpenStack Superuser site this week (http://superuser.openstack.org/articles/how-to-introduce-openstack-in-your-organization) which highlights some interesting notes about OpenStack adoption. One of the challenges of OpenStack adoptions is the concept of selling the idea of adopting an open source platform where some enterprises are lacking in confidence on supporting and operating one. There are many thoughts on how to do this, and the OpenStack Superuser blog shows how Best Buy, MercadoLibre, and Workday had success in doing so.

Seeing the Value Proposition at All Levels

This is where I like to spread out the story a bit and think of the specific value that we can attach to what OpenStack can do for our organization. This is targeted at companies from small to enterprise and all sized in between. While not every company has the depth in the org chart as others, and we have some with silos, and some with flat hierarchy, the theme is pretty similar regardless.

Here are some quick notes on the way I like to tell the story of the value of OpenStack.

CIO

It's no surprise that the pitch to the CIO is about better operational practices at lower costs. Enabling the business to achieve efficiency though the use of technology is part of the role of the CIO. We have a responsibility to use technology to drive business, and the innovation in the business will effectively drive technology to innovate with it.

IT Management

A happy CIO is made by happy IT management teams. We are looking from the top down in the list here for a reason. As we start with the upper management to see the high-level, holistic layer of managing IT infrastructure, we start to move down the org chart and see that the IT management teams will gain the benefit from having a system like OpenStack in place. Leveraging the capabilities of the IaaS platform, and additional integration with PaaS and other abstracted service offerings is all part of enabling a more competent IT practice.

Development Teams

We keep hearing about the benefits of DevOps, and with good reason. OpenStack is a powerful enabler for DevOps methods and with the use of the OpenStack APIs, a development team can use programmatic methods to request IT resources. The self-service web UI also gives a simple way to deliver access to the full service stack for your development teams. By reducing the time to deliver services, you reduce the time to market of application development. This is the heart of what DevOps is building towards. Regardless of the level of DevOps integration, the access created by OpenStack for developers rivals nearly any competitive product in the market place.

Operations Teams

Better operational practices come as a part of cloud adoption. Whether public or private cloud

deployments are used, the resulting processes that are put into place can trigger a powerful evolution for an organization. When DevOps methods are in place, its a win-win for both the operations and development team. Reducing the amount of time needed on a day-to-day basis by operations staff will allow them to put their efforts towards creating better, more innovative processes to further enable the business.

Customers

As a customer of a company that has embraced OpenStack, you have some assurance that the company is fond of innovation. It doesn't have to mean that being an OpenStack user requires being in a bleeding edge technology adoption cycle. Using OpenStack opens the door for better development and operational practices that ultimately lead to value for the customer. Better value translates to better pricing, better product availability, and higher satisfaction. This is the ultimate reason that we do what we do with technology, which is to enable better customer experience though better business process.

OpenStack Community

The side effect for greater OpenStack adoption is that the OpenStack community benefits. This can come in the way of developers providing code and enhancements upstream, through training and documentation, and in a somewhat more intangible way, in the way of having more user stories about successful OpenStack cloud deployments.

Having worked among the OpenStack community in a variety of ways, this really is one of the most powerful leading indicators of the success of OpenStack. More users correlates to wider adoption and a continued growth of the platform.

No cloud is too big or too small. With OpenStack clouds, we love them all

Why Kanban Makes me Better at Life

If you've been following along some of the #vDM30in30 content I've been doing lately, you'll notice a trend towards some non-technical content. I know that over the years I've been known as the PowerShell person, then the VMware person, then the OpenStack person. The reason that I've been changing the focus along the way was because I was really focused on what my tagline says: **People, Process, Technology**.

Technology is really the simpler portion of it all, because managing the people and process part (myself included) is really where the difficult work is.

What is Kanban?

First we will start with the obvious question, which is how do you pronounce Kanban? I use the pronunciation like Khan-bahn. When in doubt, just think of this:

Now that we've cleared that up, we can get to the fun stuff! What exactly is Kanban? Kanban comes from the Japanese word for signboard. I've learned about Kanban from The Phoenix Project and the reading I've done from Gene Kim, Eli Goldratt, and other leaders in the DevOps movement and agile software development, and manufacturing.

The origination of its use was from Toyota to create what many would call the JIT (Just In Time) delivery method where the supply of product was aligned to meet the specific demand. There is a great article on Wikipedia that gives a high level view of what the manufacturing influence was of this methodology (http://en.wikipedia.org/wiki/Kanban).

Why Kanban?

I don't run a manufacturing plant, or a supermarket, so why would I use Kanban? Great question, and the answer is that my time is my product, and my output is my work. The real core factor in why Kanban is of absolute importance to me: workload visualization. If you actually follow the Kanban (development) link (http://en.wikipedia.org/wiki/Kanban_(development)) in the Wikipedia article, you will see how this changes the application for what I use it for.

Kanban in software and infrastructure development is aligned beautifully with agile software development, and scrum practices which I've learned to really love over the last few years. Key feature to how Kanban works in this is the use of small batches with an iterative approach. Let's first see how the visualization portion works.

Start with 3 Columns: Todo, Doing, Done

I keep a personal Kanban board which has 3 columns in which I place my work. I use Post-It Notes at home, and I also use LeanKit for my online management of my tasks. Each task gets put into what we usually call the backlog, which is the Todo column. Tasks that are underway are then moved into the Doing column, and if you haven't already guessed where we are going next, the completed tasks go into the Done column.

After tasks are in the Done column, I also take time to review the work and make sure that it is completed to full satisfaction. This means that I move it out of the Done column to an Archive board for reference later on if I need to. The reason for the Done column is to ensure that we always evaluate the results. For the scrum aficionados, this is what we call the sprint retrospective.

Sprint versus Marathon

Sprinting in the running sense is an all-out effort over a short distance. This is the same way that we have to manage tasks. Rather than a marathon method where we look toward the end of a long run (this is the usual waterfall project methodology), we treat our work as smaller, more manageable portions of time called sprints. These can be from 2 days up to 3 weeks, but usually sit in the 2-3 week range.

By looking at tasks in a 2-3 week range, you don't get overwhelmed by the amount of work ahead. This also helps to let us slip by a small amount of time on a task and we simply move that task into the next sprint. If we were tracking on a 6 month schedule, you can let a lot slip to the end of the time period and then you'd be in real trouble!

Setting Limits: WIP FTW!

Work in Progress (WIP) is the amount of time that you limit yourself to in order to prevent getting

overloaded. My WIP limit is 3 items. This means that I will not have more than 3 simultaneous tasks on the go at a time. Some tasks can be quick and others not so much. This means that I can mix up the workload to make sure that I keep tasks flowing through. Remember, that this is all about visualizing the flow of tasks.

Better at Life?

The title does say that it makes me better at life, and what I mean by that is that I've applied the Kanban practice for home and work tasks. This has helped me to keep more on-track with things both in and out of the office.

This is all really quick stuff, and I have to say that Gene Kim does a much better job at explaining things, so I'll leave you with 2 things to take away as homework to find out more about Kanban.

1. Read Personal Kanban (http://www.personalkanban.com/pk/personal-kanban-the-book/#sthash.dnAvdGyQ.dpbs) because it's really good for bringing Kanban practices into the home and also at the office.
2. Watch this great webinar on YouTube that I attended with Gene Kim presented by the folks at LeanKit. Gene is a great speaker and does a phenomenal job of bringing across the value in what Kanban can do for you.

DevSecOps – Why Security is Coming to DevOps

With so many organizations making the move to embrace DevOps practices, we are quickly highlighting what many see as a missing piece to the puzzle: Security. As NV (Network Virtualization) and NFV (Network Function Virtualization) are rapidly growing in adoption, the ability to create programmable, repeatable security management into the development and deployment workflow has become a reality.

Dynamic, abstracted networking features such as those provided by OpenDaylight participants, Cisco ACI, VMware NSX, Nuage Networks and many others, are opening the doors to a new way to enable security to be a part of the application lifecycle management (ALM) pipeline. When we see the phrase Infrastructure-as-Code, this is precisely what is needed. Infrastructure configuration needs to extend beyond the application environment and out to the edge.

NFV: The Gateway to DevSecOps

Network virtualization isn't the end-goal for DevSecOps. It's actually only a minor portion. Enabling traffic for L2/L3 networks has been a major step in more agile practices across the data center. Both on-premises and cloud environments are already benefitting from the new ways of managing networks programmatically. Again, we have to remember that data flow is really only a small part of what NV has enabled for us.

Moving further up the stack to layers 4-7 is where NFV comes into play. From a purely operational perspective, NFV has given us the same programmatic, predictable deployment and management that we crave. Using common configuration management tools like Chef, Puppet, and Ansible for our regular data center management is now extensible to the network. This also seems like it is the raison d'être for NFV, but there is much more to the story.

NFV can be a confusing subject because it gets clouded as being L2/L3 management when it is really about managing application gateways, L4-7 firewalls, load balancers, and other such features. NFV enables the virtualization of these features and moving them closer to the workload. Since we know that

NV and NFV are Security Tools, not Networking Tools

When we take a look at NV and NFV, we have to broaden our view to the whole picture. All of the wins that are gained by creating the programmatic deployment and management seem to be mostly targeting the DevOps style of delivery. DevOps is often talked about as a way to speed application development, but when we move to the network and what we often call the DevSecOps methodology, speed and agility are only a part of the picture.

The reality is that NV and NFV are really security tools, not networking tools. Yes, that sounds odd, but let's think about what it is that NV and NFV are really creating for us.

When we enable the programmatic management of network layers, we also enable some other powerful features which include auditing for both setup and operation of our L2-L7 configurations. Knowing when and how our entire L2-L7 environments have changed is bringing great smiles to the faces of InfoSec folks all over, and with good reason.

East-West is the new Information Superhighway

Well, East-West traffic in the data center or cloud may not be a superhighway, but it will become the most traffic-heavy pathway over the next few years and beyond. As scale-out applications become the more common design pattern, more and more data will be traveling between virtualized components on behind the firewalls on nested, virtual networks.

There are stats and quotes on the amount of actual traffic that will pass in this way, but needless to say it is significant regardless of what prediction you choose to read. This is also an ability that has been accelerated by the use of NV/NFV.

Whatever the reasons we attach to how DevSecOps will become a part of the new data center and cloud practice, it is absolutely coming. The only question is how quickly we can make it part of the standard operating procedures.

Just when you thought you were behind the 8-ball with DevOps, we added a new one for you. Don't worry, this is all good stuff and it will make sense very soon. Believe me, because I'll be helping you out along the journey.

DiscoPosse Review: Learning Chef

I’m a big fan of Chef, and I always enjoy finding more reading resources for me and for those who are looking to adopt this great tool. The Learning Chef book is a great guide for people who have limited or no experience with this really powerful configuration management framework.

This truly is the soup to nuts (see what I did there?) guide to installing and using Chef for configuration management. This book takes you from the very beginning of explaining what configuration management is, and launching into a nicely illustrated installation guide for Chef. The good stuff is just beginning!

The book is laid out very nicely in clear sections, with focus on Ruby syntax, code samples, runbook authoring, testing, and environment creation. I would absolutely recommend this for readers getting started with Chef and even those with some familiarity. As a big fan of O’Reilly book formats and the great authors who create them, I am always happy when I have them available for learning and for future reference.

Click the image below to visit the O’Reilly site to order your copy, and I hope that you enjoy this book as much as I do