

OpenStack Summit Day 1 Quick Notes

It's quite an amazing to be here in Paris watching the OpenStack Summit unfold in front of us. With #vDM30in30 in full flight, this seemed like the perfect way to bring some of my experience from the City of Light.



The backdrop of the Champs-Élysées is like a perfect stage for me as a long time cyclist and fan of the Tour de France. While I won't be riding the cobble stones along the base of l'Arc de Triomphe, I will be walking the route, but the focus of what is happening in Paris is all OpenStack right now!

Focus on Enterprise, and Operations

While some of the most compelling use-cases for OpenStack stemmed from the ability to enable developers to simply manage environments in the same way that AWS has become the de facto tool of choice for developers on a public cloud platform. OpenStack has found its stride with many of the development-focused environments, but one of the areas that organizations have questioned whether OpenStack is ready is for the traditional enterprise.

When we say OpenStack for enterprise, we aren't talking about Netflix, eBay, LinkedIn and those types of organizations. While they are enterprise by definition because of their size, enterprise in the more "traditional" sense includes everything from soup to nuts as far as infrastructure. If you look at companies that are heavy on file-sharing, but lighter on rapid application development, you will see how making the move to OpenStack doesn't have the same draw that it may for other companies.

Beyond the type of workload as far as file sharing and such, the classic discussion around HA (High Availability) and live migration is once again at the fore when browsing potential incubation projects for the Kilo Design Summit.

Orchestrate All the Things!

Heat is getting lots of...well, heat. I've seen a lot of great work around using Heat, leveraging for external configuration management systems, and much more. Triple-O has many administrators building out some powerful orchestration recipes, and as Docker continues to draw attention, we are seeing more about the orchestration to build and scale applications.

Again, these are powerful DevOps enabling features, but the focus is on building up the strength on the operations side.

A Neat Little Package...or Distro in this Case

Want a private OpenStack cloud? Not a problem! The distributions are being developed at a decent pace, and the work being done by folks like Cloudscaling (now a part of EMC), Rackspace, Metacloud (now a part of Cisco), Piston, and Mirantis to name a few, is really laying the groundwork for a greater adoption by enterprise customers.

Despite all that we promote not the power of the OpenStack ecosystem, it will hold much more weight in many people's minds when more easily deployable, upgradeable, and fully supported distributions are available. If anything, we should regard this as a win. An OpenStack customer, regardless of how they got there, is an OpenStack customer just the same ☐

LTS FTW!

Discussions are happening on creating LTS (Long Term Support) editions of OpenStack. It is still early on in the process, but this could also be an important step. Major revisions on a 6-month cycle can be challenging for some organizations to think of adopting. It's not that there is no capability to make the updates work, but there are changes in process that will need to be in place to let companies stay up to date with limited disruption.

If we see LTS OpenStack become an option, I think that it could be an interesting and positive option to help more people enable their business using the powerful open cloud environment.

There will be much more excitement tomorrow at OpenStack Summit, and if you want to prepare for the next steps, it's a good time to tap into the excitement and start looking at how OpenStack can be a part of driving your business.

The Case for Containers: Why Docker is Coming on Strong

You can't go too far without seeing the word Docker among technology circles these days. This product/concept/company is getting some significant airtime over the course of the last few months, and there is no sign of that slowing down.

So, what exactly is the big deal around Docker? It's all about the containers. Well, mostly about the containers. Let's have a look at what the container concept is all about and how Docker comes into play among it.

The Container Concept

It's been said written in many forms, but for the TL;DR version, here is my quick summary:

Containers are encapsulated application environments that contain the necessary libraries to drive applications, with a common, open interface to run on top of a container engine such as Docker Engine. The container is lightweight as a result, and runs in isolation to provide the best flexibility and portability. The engine can run on multiple platforms, including bare-metal, which allows the containers to be deployed into on-premises and public clouds using the same application builds.

Based on this quick description, we can easily see why it's attractive. We have to be very clear though in what problem is being solved by containers. This allows us to attach the solution to the problem. Technology for technology's sake is not the right reason to deploy something.

The Problem

Application development and deployment creates a number of challenges. Included among them are:

- Application dependencies (libraries, hooks into the OS etc.)
- Consistency (many apps = many potential configurations)
- Programmability (repeatable, scriptable, measurable)
- Policy (business rules and other sometimes intangible dependencies)

These problems have been handled in a number of ways up to now. Some of them are clear constraints, but others are actually a collection of constraints which could include a lot of business processes.

Now that we've seen some of the problems, let's see what containers can do to move towards some solutions.

What Containers Solve

Based on the requirements above, we can now look at how containers are able to solve the problems, or at least get us closer to solving them. Remember that the real driver to anything we build a technology for is to answer an existing problem that exists to reduce cost, or increase the value in some way for the business.

Application Dependencies

Remember how Java was all about "write once, run anywhere"? While it's conceptually true, there are lots of issues around library and dependency versions when running Java. The same holds true for Ruby, Python, and many more languages. By moving the libraries for the language runtimes into the container, you isolate it from other application environments which reduces version issues and collisions with fighting for runtime access and contention.

Consistency

You've been there before. It's the moment that you deploy an application to a production server with a set of instructions handed to you by someone, and half way through you hit a snag of some kind.

Maybe the server is different, or maybe the configuration of the application or folder structure is not the same as the development environment. Sometimes it may have even been developed for another OS!

When adding the container as a consistent, predictable wrapper for the application environment, we ensure that all of the underlying requirements are already met. The dependencies are completely encapsulated inside the container.

Programmability

I've done my fair share of application deployments using typed out instructions, or worse, just verbal ones which invariably didn't end up being the actual steps needed to complete the deployment.

Making the application deployment programmable with a common configuration process is as close to a guarantee as you can ever get that the application will look identical in the development environment, as it will in QA, and ultimately in production where it matters the most.

Programmable configuration and deployment allows you to use your configuration management tool of choice to deploy your apps, such as Puppet, Chef, SCORCH, SCCM, vCAC and many more.

Policy

After years in financial services environments, and dozens of IT audits, I can assure you that integrating policy management into your application deployment strategy is of absolute importance.

Although containers don't wholly cover policy as a part of what they do, they will provide assurance that the deployment from one environment to another is more audit friendly.

We will leave policy for now, but in a later blog post I'm going to go much further into how policy management and containers come together.

What Containers Won't Fix

I like Docker, and I like containers as a deployment model, but before you run out and tell the CIO that you're changing to a "containers first" model, let's understand some of the adoption challenges you may face.

Here are a few points to think about:

- New tools don't fix old applications
- Cultural acceptance inside your organization
- Performance
- Full manageability

Have you ever heard the phrase "money won't fix money problems". The message in that phrase is that by simply throwing resources at the situation won't fix the core issue that caused the problem in the first place.

Adopting a new development and deployment model is not just a flip of a switch. There are many factors, most importantly ones wrapped around people and processes, that can impede changing to a container deployment model.

The cultural changes we talk about are similar to those for enabling a DevOps culture at your organization. There are significant changes to the process and people management that will lead toward better application development and deployment practices. This also holds true for containers to a degree.

Note that you don't have to be a DevOps shop to use containers, nor will using containers ensure that DevOps methodologies are in place. It can be a great pairing, but the culture shift has to be there to embrace containers as a platform

Performance and Manageability

I've separated these two out from the other two constraints, and paired them together for a reason. Performance management and overall management are particular challenges with containers as they are today.

Containerizing your applications is only dealing with the deployment and packaging. There are no baked in tools with container technologies that ensure performance or provide a robust management tool to understand your application environment.

Docker, for example, is meant to help the container. It lacks a monitoring tool that natively ensures that you are getting optimal performance for your application. It is designed so that you can easily

deploy more of your application containers, provided you've designed your application to be able to horizontally scale.

On management, there is also a lack of a robust management tool that can provide an overall view of the performance and topology of your Docker environment.

New tools are coming up that are attempting to take on these tasks, but there is still much more development in these areas to be done. It will come in time, especially as enterprise customers make the foray into using Docker as a part of their IT practice.

Study Up, it's Coming

The important thing about Docker and container concepts is that they are inevitably going to play a role, perhaps a significant one, in organizations.

I recommend that you take some time to do some reading, and I'll be preparing some Docker 101 goodness to help you along the journey.

Time to start to containerize!

[#vDM30in30 - 30 Post in 30 Days and Why it's Important](#)

Recently, I saw a post by Greg Ferro, also known as [@EtherealMind](#) on Twitter, who blogs at [EtherealMind.com](#) with some really great information specializing on networking, but with much more than that on his site. The post was called the [30 blogs in 30 days challenge](#) which I've seen in the past, and I love the inspiration that it gives to drive some excellent content and creativity.

Community, Virtual Design Master, and the #vDM30in30

I've been lucky to be a part of some incredible technology community groups. Besides being a co-leader of the Toronto VMUG and an occasional vBrownBag contributor, and over the last 2 years I've been working with [Angelo Luciani](#), [Melissa Palmer](#), and now Jonathan Frappier on the Virtual Design Master event. Part of the real fun we have had with Virtual Design Master is the amazing competitors, supporters, and contributors who have banded together to make this a premiere online reality event for IT professionals.

Once I saw the 30 blogs in 30 days from Greg, I immediately jumped in and said I'd love to be a part of it. What I really loved that happened immediately afterwards, was that many of the #vDM community jumped in and pledged to be a part of it as well.

What Will the Blogs Be About?

This is a great question. The answer is, anything! This is simply a challenge to ourselves to put our creative minds to work and flex our writing muscles with a fun, and exciting challenge.

In fact, you can kind of cheat and introduce yourself a little bit with your first post. Why not? All that matters is that you can prove to yourself that you can either begin writing, or improve on your existing writing skills. I call it writing muscle for a reason, because it needs exercise to make it stronger. There is no better training than kicking off the post-Halloween 30 days this way.

Follow Along Every Day in November

You can track the progress on Twitter (<https://twitter.com/search?f=realtime&q=%23vDM30in30>), and we will also create a list of the participants along with Greg to show our support for what he started with us.

If you want to join in, please feel free to do so. Even if you lose traction a little along the way, this is a great initiative to be a part of and it will help greatly with your writing skills.

Thank you Greg! Thank you [Cody Bunch](#) who also joined in and triggered me seeing the challenge. And thank you to the great community members who are supporting us, and building on their skills and creating great content to share with everyone.

Welcome to the [#vDM30in30!](#)

Toronto VMUG Half-Day Session - Rocking the Community with PernixData

We had a big day planned for the Toronto VMUG community for the half-day session and it was above and beyond my expectations. Community was the theme in every way as we

Kicking it off with Angelo Luciani

Angelo Luciani led us out with a preview of some VMware content including the public beta for vSphere. Remember, vSphere betas are like fight club: the first rule of vSphere betas is that nobody talks about the vSphere beta. While it may seem daunting to be under NDA (Non-Disclosure Agreement), it is a lot of fun to participate in beta programs, and you can help shape the next iteration of VMware's flagship products. To sign up for the vSphere beta program you can [click here](#).



He also chatted about the exciting work that is happening at the Virtual Design Master event this year. Make sure to check out the Virtual Design Master because we are ramping up in a big way now as Challenge 3 is about to be announced on Thursday July 24th at 9:00 PM Eastern live at <http://virtualdesignmaster.com/live>.

Community Presentation - Joel Gibson

Joel is one of our community members, and also a competitor in this year's Virtual Design Master competition. What is also great about Joel is that he was one of the folks who made use of the mentoring program to put the User in User Group that was run last year by a number of VMUG members and bloggers. A very cool idea that led to the Feed Forward program in VMUG this year.



Joel did a very interactive troubleshooting session with the audience as he highlighted some issues that he has run into both in his lab and at the office. What made it great was the very interactive nature. Joel has a great relaxed presentation style and encouraged questions and tips from the audience.

Feedback was great on the session and we are proud to have Joel among our membership.

PernixData - Re-think Storage Performance

For this event we were very excited to host PernixData with a presentation by Andy Daniel ([@vNephologist](#)) on the flagship PernixData FVP offering. Now in their second year since launch, the product is at version 1.5 and I predict some great things coming from the PernixData team.

The presentation was also very interactive, and triggered lots of questions and comments that led to a great conversation between Andy and the audience. This is precisely the kind of presentation that makes our VMUG attendees happy to be here, and to come back for.



Andy was also in town with Kirk Arrowood and it is always great to chat with them on what's happening at PernixData, and in the industry at large.

We also were proud to be able to finalize the addition of PernixData as a supporter of the Virtual Design Master competition. Andy and Kirk were excited about the event, and once again they showed how community oriented they are both personally and professionally.

Community Presentation - Mike Preston

If you don't already follow Mike online ([@mwpreston](#) on Twitter, [www.mwpreston.net](#)) then I encourage you to do so now. Mike is a great presenter, and has been a great help in the leadership at the Toronto VMUG. I'm pleased to count Mike among my friends, and I know I can always lean on him for some great tips on vCenter Orchestrator among many other VMware and virtualization technologies.



Mike gave a great presentation on the value of community, and how he has used community and online social media resources and blogs to help him in his career. Again, the theme of community was strong and was greatly appreciated by the audience.

Community Presentation - Eric Wright...hey, that's me!

This was a fun chance to be able to reprise my presentation which I gave at the Cloud Expo in New York recently on behalf of Pluralsight to go along with my Introduction to OpenStack course.

I really love giving this presentation and it was fun to share with the Toronto VMUG group about OpenStack, open source technologies, and the opportunities to work with VMware and OpenStack.



And yes, I was wearing socks with clouds on them.



All of the presentations are available at Angelo Luciani's site www.VirtuWise.com here.

Thanks to all the great members, community presenters, and to the VMUG organization for continuing to create an environment for us to deliver great content to our members with a distinct community focus which is unmatched by other events.