March 20, 2018

Aptira (https://aptira.com)

SUSE Expert Days (https://aptira.com/suse-expert-days/)

We recently teamed up with our good friends at SUSE (https://aptira.com/case-studies/case-study-swinburne-university-technology/) to build a very high-performing and scalable storage landscape at a fraction of the cost than with traditional storage systems for the Swinburne University of Technology (http://www.swinburne.edu.au/).

The challenge that the IT team at Swinburne were facing is how to accommodate the ever-growing need for performance and capacity while sticking to a tight budget. With SUSE Enterprise Storage we have been able to deploy a compelling and affordable solution to support Swinburne's storage needs. Their new storage platform is fast and flexible, meaning the IT team at Swinburne can support the needs of researchers more effectively. You can read more about this solution here, (https://aptira.com/case-studies/case-study-swinburne-university-technology/) or you could come see us at the SUSE Expert Days (https://www.suse.com/promo/expert-days/) in Melbourne next week where we will be diving into this solution in more detail.

Taking place in more than 80 cities worldwide, the SUSE Expert Days tour offers a half day of technical discussions, presentations and demos featuring SUSE engineers and experts. The presentations focus on technology topics that are relevant and compelling to IT professionals, centered on this year's theme: Open. Redefined.

Our session is from 10am (just after the SUSE storage demo). Stick around to chat with us during the break so you can learn more about how we work with SUSE to deliver storage solutions to meet a vast range of requirements. Check out the full agenda (http://events.suse.com/events/2018-apj-suse-expert-days-melbourne/agenda-68e988c761bb47bdb71d791cbdcabcbd.aspx) here, and don't forget to register (http://events.suse.com/events/2018-apj-suse-expert-days-melbourne/registration-68e988c761bb47bdb71d791cbdcabcbd.aspx). See you there!

WHEN

Thursday, March 22, 2018
8:30 AM – 1:30 PM
E. Australia Time

WHERE

Rydges Melbourne
186 Exhibition St
Melbourne 3000
Australia

The post SUSE Expert Days (https://aptira.com/suse-expert-days/) appeared first on Aptira (https://aptira.com).

by Jessica Field at March 20, 2018 06:19 AM (https://aptira.com/suse-expert-days/)

Rackspace Developer Blog (https://developer.rackspace.com)

Using multiple NIC cards on OpenStack instances for additional IP addresses (https://developer.rackspace.com/blog/Using-multiple-NIC-cards-on-OpenStack-instances-for-additional-IP-addresses)

In my previous series of articles (https://blog.rackspace.com/a-users-look-at-openstack-networking-part-5) on OpenStack networking, I mentioned that there are ways of doing things in the virtual world that can't be done in the physical world. Because of that, we shouldn't let the physical server limitations control our thinking in the virtual world -- but often we do.

Introduction

When a physical server needs multiple public Internet Protocol (IP) addresses, we add these IPs onto one network interface controller (NIC) card because a limited number of NIC card slots are available in a chassis. These limitations do not exist in the virtual world, where you can add a virtual NIC card for each additional IP needed. The limitation for virtual servers isn't physical slots -- instead, it's the number of available PCI bus addresses. This limitation occurs when a web server supports multiple domains, some of which have separate public IPs.

Each public IP is a floating IP address on the public cloud network. This floating IP must be mapped to an IP on the server so that traffic to the floating IP is directed to the server. The difference in the cloud is that the new IP can reside on a new NIC card as opposed to being added to an existing NIC card.

Routing problem

But this presents a newfound problem. Running separate NIC cards on the same subnet has been not recommended in the past, principally due to a lack of understanding of policy-based routing that was built into the kernel network stack over the last few years. Many people don't know that by using policy-based routing, it is quite simple to configure multiple NIC cards on the same subnet.

The problem occurs in the following scenario:

- A packet comes in on one interface (for example, interface A)
- The outgoing response packet gets routed out by another interface, which is connected to the same subnet (for example, interface B).
- The first routing table entry for the subnet is not for the interface on which the packet first arrived (in this case, interface A).

As a result, the outgoing reply packet has an Ethernet-frame MAC address of the interface from which the packet exits (interface B), which causes a problem. The issue is that OpenStack port security causes this outgoing packet to be dropped as it comes out of the instance. OpenStack does not have a port using an IP/MAC address combination, and OpenStack does not allow a port to change either its MAC address or its IP address.

Solution: Policy-based routing

Policy-based routing provides additional options to decide how a packet may be routed. Decisions can be based on a number of parameters. The typical router makes routing decisions based on the IP destination address. In some cases, you may need to route packets to interfaces based on different criteria, depending not only on the destination addresses but also on other packet fields such as source address, IP protocol, transport protocol ports, or even packet payload. The kernel's implementation of this type of routing is called *Policy Routing*.

In this application, we want to route packets based on the packet `source address` to ensure that the packets go out of the interface (or NIC card) that uses the packet's `from` IP address. If we don't do this, the system uses the interface with the first entry in the routing main table. This first entry may or may not be the interface on which the packet came in, which could cause the IP/MAC address relationship to be incorrect.

By default, Linux uses a minimum of three separate routing tables, named main, default, and local (main id=254, default id=253, and local id=255). For packets going to external addresses, the kernel looks in the main routing table (id=254) and finds the first routing match to the desired subnet, and it uses only that first entry. As a result, all packets from the NIC card use that same routing entry from the main table (id=254). Since the Ethernet frame information is added by the *outgoing* interface, the MAC address of the packet is from the outgoing interface (rather than the incoming interface) and communication fails. This can be fixed only by using policy-based routing. To do this, we need to create separate tables for each NIC card on the subnet. These tables include a route and a rule for each IP or NIC card, which cause packets to be routed out of the correct interface.

Policy-based routing example

Let's consider how to configure a system to use policy-based routing by doing the following tasks:

- Create routing tables populated with routes and rules for each NIC card.
- Add a route in the main table for each NIC card.

In this case, let's setup each interface to use static addresses rather than OpenStack's DHCP-based addressing system.

For recent Debian-based systems, the directory `/etc/network/interfaces.d` has a file for each interface, so we just need to set the static IP and to add four lines to the interface configuration file for policy-based routing. After setting the static IP and network information in the file, add the following details. Don't forget to change the sections between angle brackets(`<` and `>`) to contain the appropriate information for the interface:

```
post-up ip route add <subnet CIDR> dev <interface name> src <assigned IP address> table <table name or number>
post-up ip route add default via <subnet gateway> dev <interface name> table <table name or number>
post-up ip rule add from <assigned IP address> table <table name or number>
post-up ip route add <subnet CIDR> dev <interface name> src <assigned IP address>
```

Let's illustrate with the following example. A new interface is to be set up on the `10.20.0.0/24` network using this information:

```
device name - ens4
device IP - 10.20.0.4
subnet CIDR - 10.20.0.0/24
subnet gateway - 10.20.0.1
file name: /etc/network/interface.d/ens4.cfg
```

After we fill in the information, `/etc/network/interface.d/ens4.cfg` contains:

```
auto ens4
iface ens4 inet static
address 10.20.0.4
network 10.20.0.0
netmask 255.255.255.0
broadcast 10.20.0.255
post-up ip route add 10.20.0.0/24 dev ens4 src 10.20.0.4 table 4
post-up ip route add default via 10.20.0.1 dev ens4 table 4
post-up ip rule add from 10.20.0.4 table 4
post-up ip route add 10.20.0.0/24 dev ens4 src 10.20.0.4
```

Naming or numbering routing tables

Routing tables can be identified by either a number or a name. Here we used numbers, but to name them, use a unique table name followed by a unique number in the `/etc/iproute2/rt_tables` file. To avoid editing the file, use numbers for each table reference. It might be helpful to use the subnet part of the IP address of the NIC card as the table number or for NIC cards. If you have interfaces on different subnets, then add part of the base address to the table number to ensure that the table number is unique. The Linux kernel can handle up to 2**31 (2147483648) tables.

Conclusion

As you can see, it is quite easy to configure policy-based routing for these types of applications. Using separate NIC cards for each additional IP is a virtual-world alternative to assigning multiple IPs to a single NIC card.

Rackspace is the world's leading OpenStack service provider. To learn more about OpenStack, attend a public training session or schedule a private training class with our experts. For a complete schedule of class offerings, please visit the Rackspace Training page.

March 20, 2018 12:00 AM (https://developer.rackspace.com/blog/Using-multiple-NIC-cards-on-OpenStack-instances-for-additional-IP-addresses)

March 19, 2018

Cameron Seader (http://blog.seader.us/search/label/OpenStack)

SUSE is the trusted source for your Cloud Foundry PaaS (http://blog.seader.us/2018/03/suse-is-trusted-source-for-your-cloud.html)

With the recent release of the SUSE Cloud Application Platform (https://www.suse.com/products/cloud-application-platform/)based on Cloud Foundry (https://www.cloudfoundry.org/) and Kubernetes (https://kubernetes.io/), you might be wondering what benefits SUSE brings to these open source projects. Let me share with you some details about being a trusted source and enterprise-ready.

SUSE has been a major player in the open source industry for over 25 years, and our longstanding success is rooted deeply in this circle of trust. SUSE knows open source. And we know what it means to be enterprise ready. We have repeatedly and successfully turned open source technologies into powerful enterprise-class software solutions you use today. Key to our success, we have an innovative software management stack that is enterprise grade, we have a powerful build model that enhances our abilities to deliver enterprise-ready software, and we deliver enhanced security protection across the whole stack. But most importantly our engineersare trained and certified and ready to serve you.

With SUSE Cloud Application Platform, SUSE brings this model of trusted, enterprise-grade, open source software to your application delivery teams. You get a complete, open source solution with everything needed to accelerate application delivery, including SUSE Cloud Foundry (https://github.com/SUSE/scf), Stratos UI (https://github.com/cloudfoundry-incubator/stratos), SUSE CaaS Platform (https://www.suse.com/products/caas-platform/) (our Kubernetes distribution), and SUSE Enterprise Storage (https://www.suse.com/products/suse-enterprise-storage/). This is a first of its kind built and running on Kubernetes. If you missed what all that means read this blog post about Applying the Cloud Foundry workflow to Kubernetes (https://www.suse.com/c/applying-cloud-foundry-workflow-kubernetes/).

SUSE Linux Enterprise Server is at the core of the technologies we have created. The package management has been enhanced for containers. We have built an enterprise-grade container host OS called MicroOS which utilizes the enhancements. Updates for MicroOS are released on a continuous delivery model as transactional updates. These updates are atomic, don't influence the running system, and can be rolled back in the event you need to. The system can be manually rebooted to activate the changes that were applied from any updates, or be set up to reboot automatically on a scheduled basis through the rebootmgr tool. Everything delivered is signed and verified from SUSE sources. These features make MicroOS an ideal infrastructure for running Kubernetes, by addressing key reliability, availability, serviceability (RAS) and security requirements forany enterprise environment. You can read further at the openSUSE Kubic project portal (https://en.opensuse.org/Portal:Kubic)which is the upstream project for Container as a Service Platform.

Many developers today are comfortable building containers on linux variants that are not enterprise hardened, but those same containers will most likely be unacceptable in production environments. When you use SUSE Cloud Application Platform, you can be assured that your application is built on containers using SUSE Linux Enterprise base images, and you know that your container will make it out of dev/test and into production without any trouble. We utilize our powerful build model and the Open Build Service (http://openbuildservice.org/) to build our base container images using KIWI (https://opensuse.github.io/kiwi/). These container images are built, signed, and verified in the Open Build Service, and then each image is signed and readied for a public/private notary.

Moving up the SUSE Cloud Application Platform stack, you'll see how we've carried our trademark enterprise-grade value further, into SUSE Cloud Foundry. Signed and readied SUSE images are used as our base for the SUSE Cloud Foundry Fissile (https://github.com/SUSE/fissile)Stem Cell that runs both the Cloud Foundry application stack and Build Packs. On top of all of that, these OCI compliant images can be used to implement your application, or a third party application, without the hassle of stripping the base imagedown and recreating it.

The SUSE build model, utilizing the Open Build Service and other open source software givesus the advantage of having a fully secured, tested, signed, and verified delivery of the entire SUSE Cloud Application Platform, from source to image to notary and into your hands. These sources go through hundreds of quality assurance models daily in our openQA (https://openqa.opensuse.org/)tool as part of our pipeline delivery.

And, while we're on the absolutely critical topic of security, let's recognize that there's more to that than secure images. The whole SUSE Cloud Application Platform solution has been designed and delivered with security in mind. We fully support and integrate Apparmor on the container host (MicroOS). We also support the implementation of UEFISecure Boot, cryptographicallyhashing of all files, as well as a read-only root file system. Further hardening can be applied by following our hardening guide for SUSE Linux Enterprise Server (https://www.suse.com/documentation/sles-12/singlehtml/book_hardening/book_hardening.html).

Finally, and perhaps most importantly, we are here to help. We have trained and certified support engineers ready to jump in on a moment's notice to dig you out of trouble. That's part of our core mission (https://www.suse.com/brandcentral/suse/corporate.php#corporate)here at SUSE. We have available many different support offerings from dedicated to semi-dedicated premium engineers that can work directly with your teams.

SUSE is the trusted source for your Cloud Foundry PaaS. Our complete solution will give you everything you need to streamline lifecycle management of traditional and new cloud native applications. This platform facilitates DevOps process integration to accelerate innovation, improve IT responsiveness, and maximize return on investment.

Have a lot of fun!

by Cameron Seader (noreply@blogger.com) at March 19, 2018 11:59 PM (http://blog.seader.us/2018/03/suse-is-trusted-source-for-your-cloud.html)

Chris Dent (https://anticdent.org/)

Placement Container Playground 5 (https://anticdent.org/placement-container-playground-5.html)

This is the fifth in a series of posts about experimenting with OpenStack's (https://www.openstack.org/) placement service (https://developer.openstack.org/api-ref/placement/) in a container. In the previous episode (https://anticdent.org/placement-container-playground-4.html), I made an isolated container that persists data to itself work in kubernetes. In there I noted that persisting data to itself rather takes the joy and functionality out of using kubernetes: You can't have replicas, you can't autoscale, you lose all your data.

I spent some of yesterday and today resolving those issues and report on the results here: an autoscaling placement service that persists data to a postgresql server running `$elsewhere`.

The code for this extends the same branch of placedock (https://github.com/cdent/placedock/tree/self-contained-authless) as playground 4 (https://anticdent.org/placement-container-playground-4.html) and continues to use minikube (https://github.com/kubernetes/minikube). I gave up trying to get things to work on linux with the `kvm` or `kvm2` drivers. I should probably try the `none` driver at some point, but for now this work has been happening on a mac.

There are two main chunks to make this work:

- Adapting the creation of the container and the creation and syncing of the database so that the database can be outside the container.
- Tweaking the kubernetes bits to get a horizontal pod autoscaler (https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/) working.

Note: This isn't a tutorial on using kubernetes or placement, it's more of a trip report about the fun stuff I did over the weekend. If you try to follow this exactly for managing a placement service, it's not going to work very well. Think of this as a conversation starter. If you're interested in this stuff, let's talk. I recognize that my writing on this topic has become increasingly incoherent as I've gone off into the weeds of discovery. I will write a summary of all the playgrounds once they have reached a natural conclusion. For now, even in the weeds, they've taught me a lot.

Database Tweaking

In playground 4, the database is established when building the container: every container gets its own sqlite db sitting there ready and waiting for run time. This does not work if we want to use a remote db and we want multiple containers talking to the same db. Therefore the `sync.py` script, which creates the database tables is copied into the container at build time, but not actually run until run time.

At run time, it gets the database connection url from an environment variable, `DB_STRING` . If it's not set, a default is used. We can define the value in the kubernetes `deployment.yaml` .

But wait, the container had only been running the uwsgi process. How do we get it to use the environment variable, run `sync.py` , and only once that's done, start up the uwsgi server? Turns out we can replace the existing docker `CMD` with a script that does all that stuff. In the `Dockerfile` the end is adjusted to:

```
ADD startup.sh /
CMD ["sh", "-c", "/startup.sh"]
```

and `startup.sh` is:

```
DB_STRING=${DB_STRING:-sqlite:////cats.db}

# Do substitutions in the template.
sed -e "s,{DB_CONNECTION},$DB_STRING," < /etc/nova/nova.conf.tmp >
/etc/nova/nova.conf

# establish the database
python3 /sync.py --config-file /etc/nova/nova.conf

# run the web server
/usr/sbin/uwsgi --ini /placement-uwsgi.ini
```

It would surprise me not one iota if there aren't cleaner ways than that, but that way worked.

The result of this is that each time a container starts, it connects to the database described in `$DB_STRING` and tries to create and update the tables. If they are already created, it's happy. If something else is in the midst of versioning the database an exception is caught and ignored.

I had a `postgresql` server running on a nearby VM, so I used that. For the time being I simply added the necessary connection drivers to the container at build time, but if it was required to be super dynamic, then the python driver code could be installed at runtime. Being super dynamic is not really in scope for my experiments.

After doing all that, I adjusted my deployment to have 4 replicas and made sure things worked. And it did. Onward.

Kubernetes Tweaking

Having 4 containers, either doing nothing or being overloaded, is not really taking advantage of some of the best stuff about kubernetes. What we really want, to be both more useful and more cool, is to create and destroy the placement pods as needed. This is done with a Horizontal Pod Autoscaler (https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/). You tell it the minimum and maximum number of pods you're willing to accept and a metric for determining the percent of resource consumption that is the boundary between OK and overloaded.

Here's the `autoscaler.yaml` that works for me:

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: placement-deployment
  namespace: default
spec:
  maxReplicas: 10
  minReplicas: 1
  scaleTargetRef:
    apiVersion: extensions/v1beta1
    kind: Deployment
    name: placement-deployment
  targetCPUUtilizationPercentage: 50
```

While this is a relatively simple concept and the tooling is straightforward it took me quite some time to get this to work. I've been using `minikube 0.25.0` (the latest release as of this writing) but running it at the maximum version of kubernetes that it supports (v1.9.0). This leads to some conflicts.

In older versions, the expected way to manage autoscaling and metrics is to use heapster (https://github.com/kubernetes/heapster). Minikube includes an addon for this, but as installed it does not present a "rest" API for the information. That's okay for some unclear number of kubernetes versions back, but is not with v1.9.0.

Modern kubernetes has the Resource Metrics API (https://github.com/kubernetes/community/blob/master/contributors/design-proposals/instrumentation/resource-metrics-api.md). heapster can support that, but only if it is started with a particular flag. The other option is to start the kubernetes-controller with a particular flag so that autoscaling doesn't use the metrics API. I preferred to stay modern.

Unreleased `minikube` adds the metrics server as an addon (https://github.com/kubernetes/minikube/tree/master/deploy/addons/metrics-server). I was able to copy that code into my minikube setup and establish the service.

Next I discovered that my placement deployment needed to describe a resource limit in order for the autoscaling to work. In hindsight this is obvious. The autoscaling is done based on a percentage of a limit the deployment sets for itself. For instance if you say that say things should be scaled up when resource usage hits `50%` , kubernetes says "50% of *what*?".

In my case that meant adjusting the containers section of `deployment.yaml` :

```
  containers:
  - name: placement
    image: placedock:1.0
    env:
    - name: DB_STRING
      value: postgresql+psycopg2://cdent@192.168.1.76/placement?client_encoding=utf8
    ports:
    - containerPort: 80
    # We must set resources for scaling to work.
    resources:
      requests:
        cpu: 250m
```

That last stanza is saying "we request 1/4 core worth of cpu". So now, the autoscaler is expressing when cpu utilization hits 50% (of 1/4 core), scale.

Note: If you're following along and using `minikube` with `docker-machine` keep in mind that the default "machine" is pretty small so you need to keep the resource request for each individual container pretty small or you will soon overwhelm the machine. I had `cpu` above set to `1000m` initially. Starting new pods was slow enough that they never became ready.

When I finally got this working it was fun to watch (literally with `watch kubectl get hpa`). If you're starting from scratch it can take a while for everything to warm up and be running, but eventually you'll see low usage and low replicas (this output is wide, you may need to scroll):

```
NAME                   REFERENCE                         TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
placement-deployment   Deployment/placement-deployment   0% / 50%  1         10        1          1m
```

To load up the deployment and make it scale (assuming there's a bit of data in the database, like the one resource provider that the `gabbi-run` in `bootstrap.sh` will create) I did this:

```
export PLACEMENT=$(minikube service placement-deployment --url)
ab -n 100000 -c 100 -H 'x-auth-token: admin' $PLACEMENT/resource_providers
```

After a while the target usage raises above the `50%` target and more replicas are created.

```
NAME                   REFERENCE                         TARGETS    MINPODS   MAXPODS   REPLICAS   AGE
placement-deployment   Deployment/placement-deployment   109% / 50% 1         10        3          8m
```

When the `ab` was done and resource usage settled, all but one of the containers were terminated. That one was working as expected:

```
curl -H 'x-auth-token: admin' $PLACEMENT/resource_providers |json_pp
```

Even though I know that's exactly how it is supposed to work, it's still pretty cool. What next? I need to add forbidden traits (http://specs.openstack.org/openstack/nova-specs/specs/rocky/approved/placement-forbidden-traits.html) support to the placement service, but after that I will likely revisit this stuff, either for more scale fun (https://anticdent.org/placement-scale-fun.html) or the cat management mentioned in playground 4 (https://anticdent.org/placement-container-playground-4.html).

As always, please leave a comment or otherwise contact me if you have questions, there's something I've done weirdly, or you're doing similar stuff and there's an opportunity for us to collaborate.

by Chris Dent at March 19, 2018 07:30 PM (https://anticdent.org/placement-container-playground-5.html)

OpenStack Superuser (http://superuser.openstack.org)

OpenStack community updates for new contributors (http://superuser.openstack.org/articles/openstack-community-updates-for-new-contributors/)

New contributors were the focus of many discussions the week of the Project Teams Gathering (PTG (https://www.openstack.org/ptg/)) in a snow-beaten Dublin, Ireland. Almost every day there was a topic or a group meeting to discuss how to make it easier for people to come into our community and start contributing.

> #OpenStack (https://twitter.com/hashtag/OpenStack?src=hash&ref_src=twsrc%5Etfw) PTGs are always unique, and this one doesn't disappoint! #SnowpenStack (https://twitter.com/hashtag/SnowpenStack?src=hash&ref_src=twsrc%5Etfw) pic.twitter.com/GYuhFi5Ky0 (https://t.co/GYuhFi5Ky0)
>
> — Thierry Carrez (@tcarrez) March 1, 2018 (https://twitter.com/tcarrez/status/969201205690929152?ref_src=twsrc%5Etfw)

First Contact SIG

The First Contact SIG (https://wiki.openstack.org/wiki/First_Contact_SIG) met bright and early Monday morning to take stock of the existing resources to help new contributors. We discussed everything with the aim of getting everyone aware of the tools and groups that exist so that our SIG members can share them with new community members.

There was a discussion about IRC channels (https://wiki.openstack.org/wiki/IRC) and what to do with #openstack-101. To many of you, #openstack-101 might seem like the right place to go when you are getting started and have basic questions. However, to the majority of the world '101' doesn't have the meaning it does to US-centric minds. In an effort to better direct those basic questions, we've decided to get rid of the channel and roll those discussions into the #openstack-dev channel. Besides the name, #openstack-101 had other issues with low traffic and not many people answering questions or directing traffic. By removing the old channel, questions will get more attention and hopefully we'll keep more people engaged.

The other larger action to emerge from our discussions was to establish a weekly meeting. While sifting through some of the less-known contributor resources like ask.openstack.org, (http://ask.openstack.org) we decided that certain areas there and in Gerrit should be monitored routinely to make sure people don't fall through the cracks. We now host weekly meetings (http://eavesdrop.openstack.org/) in #openstack-meeting at 0800 UTC to go over a few standing items and any other new ones that come up. Join us (http://eavesdrop.openstack.org/#First_Contact_SIG_Meeting) if there's a resource you want to know more about, get more involved, or think we should pay closer attention to!

> I don't know about you @OpenStack (https://twitter.com/OpenStack?ref_src=twsrc%5Etfw) but I think our #SnowpenStack (https://twitter.com/hashtag/SnowpenStack?src=hash&ref_src=twsrc%5Etfw) #PTG (https://twitter.com/hashtag/PTG?src=hash&ref_src=twsrc%5Etfw) game night was a massive success 🙂 pic.twitter.com/4yke8xEGTb (https://t.co/4yke8xEGTb)
>
> — Kendall Nelson (@knelson92) March 2, 2018 (https://twitter.com/knelson92/status/969515964176969734?ref_src=twsrc%5Etfw)

Contributor Guide

The Contributor Guide (https://docs.openstack.org/contributors/) starred in two team discussions: the First Contact SIG and the Docs/i18n. In both rooms, we did walk-throughs and took exhaustive notes on all the things that were missing, needed expanding upon or updating.

After these discussions, I added the items discussed to our task tracker so that throughout the week people could pick up items and get them done in their spare time. Since then, there's been an influx of patches from many people that hadn't previously worked on for the Contributor Guide. While it still has a lot of growing to do, the path ahead is more clearly marked.

> These are my people. I've walked through fire with them, they have my back … And we can gather around a piano #WeAreOpenStack (https://twitter.com/hashtag/WeAreOpenStack?src=hash&ref_src=twsrc%5Etfw) pic.twitter.com/YoxKfJZvzN (https://t.co/YoxKfJZvzN)
>
> — Monty Taylor @ 🇺🇸 (@e_monty) February 28, 2018 (https://twitter.com/e_monty/status/968658891507732481?ref_src=twsrc%5Etfw)

OpenStack Upstream Institute

With the growth and development of the Contributor Guide, the way the team run Upstream Institute (http://superuser.openstack.org/articles/kickstart-knowledge-openstack-upstream-institute/) has the opportunity to evolve even further.

The goal has always been to continually evolve and be more hands-on and interactive for students. The Contributor Guide offers an even more engaged approach to getting all the basic accounts and tools set up than the slides we have used previously. As the Guide continues to grow and cover more information, we are able to remove many slides that cover the same material. During the trainings, we will now walk through the guide taking breaks to do exercises to make sure students have completed sections successfully.

Discussions at the PTG were primarily around what we need to do to continue development of the guide and what changes we need to make to the slides to get things prepared for the upcoming training at the Vancouver Summit (https://www.openstack.org/summit).

If you'd like to participate, more on the free, daylong training here. (https://docs.openstack.org/upstream-training/upstream-training-summit.html)

Kendall Nelson is the OSF's upstream developer advocate. You can find her on IRC at diablo_rojo or on Twitter. (https://twitter.com/knelson92)

The post OpenStack community updates for new contributors (http://superuser.openstack.org/articles/openstack-community-updates-for-new-contributors/) appeared first on Superuser (http://superuser.openstack.org).

by Kendall Nelson at March 19, 2018 03:55 PM (http://superuser.openstack.org/articles/openstack-community-updates-for-new-contributors/)

Giulio Fidente (http://giuliofidente.com/)

Ceph integration topics at OpenStack PTG (http://giuliofidente.com/2018/03/ceph-integration-topics-at-openstack-ptg.html)

I wanted to share a short summary of the discussions happened around the Ceph integration (in TripleO) at the OpenStack PTG (https://www.openstack.org/ptg/).

ceph-{container,ansible} branching

Together with John Fulton (http://blog.johnlikesopenstack.com/) and Guillaume Abrioux (https://github.com/guits) (and after PTG, Sebastien Han (http://www.sebastien-han.fr/)) we put some thought into how to make the Ceph container images and ceph-ansible releases fit better the OpenStack model; the container images and ceph-ansible are in fact loosely coupled (not all versions of the container images work with all versions of ceph-ansible) and we wanted to move from a "rolling release" into a "point release" approach, mainly to permit regular maintenance of the previous versions known to work with the previous OpenStack versions. The plan goes more or less as follows:

> - ceph-{container,ansible} should be released together with the regular ceph updates
> - ceph-container will start using tags and stable branches like ceph-ansible does

The changes for the ceph/daemon docker images are visible already: https://hub.docker.com/r/ceph/daemon/tags/ (https://hub.docker.com/r/ceph/daemon/tags/)

Multiple Ceph clusters

In the attempt to support better the "edge computing" use case, we discussed adding support for the deployment of multiple Ceph clusters in the overcloud.

Together with John Fulton (http://blog.johnlikesopenstack.com/) and Steven Hardy (http://hardysteven.blogspot.com/) (and after PTG, Gregory Charot) we realized this could be done using multiple stacks and by doing so, hopefully simplify managament of the "cells" and avoid potential issues due to orchestration of large clusters.

Much of this will build on Shardy's blueprint to split the control plane, see spec at: https://review.openstack.org/#/c/523459/ (https://review.openstack.org/#/c/523459/)

The multiple Ceph clusters specifics will be tracked via another blueprint: https://blueprints.launchpad.net/tripleo/+spec/deploy-multiple-ceph-clusters (https://blueprints.launchpad.net/tripleo/+spec/deploy-multiple-ceph-clusters)

ceph-ansible testing with TripleO

We had a very good chat with John Fulton (http://blog.johnlikesopenstack.com/), Guillaume Abrioux (https://github.com/guits), Wesley Hayutin (https://github.com/weshayutin) and Javier Pena (https://jpenatech.wordpress.com/) on how to get tested new pull requests for ceph-ansible with TripleO; basically trigger an existing TripleO scenario on changes proposed to ceph-ansible.

Given ceph-ansible is hosted on github, Wesley's and Javier suggested this should be possible with Zuul v3 and volunteered to help; some of the complications are about building an RPM from uncommitted changes for testing.

Move ceph-ansible triggering from workflow_tasks to external_deploy_tasks

This is a requirement for the Rocky release; we want to migrate away from using workflow_tasks and use external_deploy_tasks instead, to integrate into the "config-download" mechanism.

This work is tracked via a blueprint and we have a WIP submission on review: https://blueprints.launchpad.net/tripleo/+spec/ceph-ansible-external-deploy-tasks (https://blueprints.launchpad.net/tripleo/+spec/ceph-ansible-external-deploy-tasks)

We're also working with Sofer Athlan-Guyot on the enablement of Ceph in the upgrade CI jobs and with Tom Barron on scenario004 to deploy Manila with Ganesha (and CephFS) instead of the CephFS native backend.

Hopefully I didn't forget much; to stay updated on the progress join `#tripleo` on freenode or check our integration squad status at: https://etherpad.openstack.org/p/tripleo-integration-squad-status (https://etherpad.openstack.org/p/tripleo-integration-squad-status)

by Giulio Fidente at March 19, 2018 02:32 AM (http://giuliofidente.com/2018/03/ceph-integration-topics-at-openstack-ptg.html)

March 17, 2018

Chris Dent (https://anticdent.org/)

Placement Container Playground 4 (https://anticdent.org/placement-container-playground-4.html)

Update: If you want to experiment with the exact code that was used to write this blog posting it is the code at this commit (https://github.com/cdent/placedock/commit/7aff711511be0bb747ec8a7b4a00fcc18a58f882). The branch (https://github.com/cdent/placedock/tree/self-contained-authless) has since evolved to address some of scaling issues described within. A post about that is forthcoming.

This is the fourth in a series of posts about experimenting with OpenStack's (https://www.openstack.org/) placement service (https://developer.openstack.org/api-ref/placement/) in a container. In the previous episode (https://anticdent.org/placement-container-playground-3.html), I was working on further limiting the modules being imported into the service. Today I report on two recent efforts:

- Running placement standalone with its own sqlite database within the container, and without keystone auth.
- Running the resulting container in kubernetes (https://kubernetes.io/) with a reasonable HTTP setup, as a learning exercise.

- Standalone Throwaway Storage (https://anticdent.org/feeds/openstack.atom.xml#standalone-throwaway-storage)
- Now With Kubernetes! (https://anticdent.org/feeds/openstack.atom.xml#now-with-kubernetes)

These two goals are somewhat at odds with one another: Having the database in the container is useful for throwaway testing but defeats the usefulness of using kubernetes: there's no shared storage so a deployment with four replicas is really 4 different services. So the outcome here is to run only one container, but sufficient things were learned so that if/when there is a *Playground 5* it should be possible to scale to infinity and beyond.

For the time being this work in being done on a branch of placedock (https://github.com/cdent/placedock/tree/self-contained-authless). I eventually intend to consolidate this back to an intermediate image from which different variants (uwsgi protocol based shared storage, for use with a running devstack; http-based shared storage; http-based throwaway storage) can be built. As a reminder: at no point should this `Dockerfile` or any of the associated code be considered correct. It's fodder for experimentation and learning.

Standalone Throwaway Storage

I intend, at some point, to write a somewhat tongue in cheek exploration of using the placement service to manage the care and feeding of cats. Something that operates as both a tool for getting my mind right about what placement is, in its essence, as well as a kind of tutorial for coming at it from a different angle. To make it useful I want it to be easy for the reader to play along. A container is one way to accomplish that but by default placement wants to share a database with nova, use mysql, and engage with the keystone identity service. That's heavy.

Making a lighter container required some changes:

1. Getting rid of the VOLUME that was sharing configuration settings into the container. Better to just copy them in.
2. Set the database connection string to `sqlite` and a file, local to the container.
3. Figure out how to create the database tables.

The third step proved time consuming. As one of the original goals was to limit the amount of nova modules being imported, it was not simply a case of calling the same code (https://github.com/openstack/nova/blob/b6c00f24d4b611746775c4a6c0edd4789ac8b408/nova/cmd/manage.py#L800-L821) used by the `nova-manage api_db sync` command. Instead calling the `migrate` module directly was the way to go. Teasing out the correct arguments required some trial and error.

Then it turned out that one of the migrations (https://github.com/openstack/nova/blob/b6c00f24d4b611746775c4a6c0edd4789ac8b408/nova/db/sqlalchemy/api_migrations/migrate_repo/versions/014_keypair imports `nova.objects.keypair` to get at a static string constant. This results in all the nova objects being imported, setting off a cascade of dependencies.

Simply deleting the offending migration doesn't work because the migration system will not accept gaps in the numbered files. So I dynamically rewrite the file to have an empty `upgrade` method.

Another, more complex-in-action but tidier-in-result, operation might be to copy and renumber the migration files of only those tables that are used by the placement service. I didn't try this because I was in a hurry and we're going to have to address placement database management as part of extraction (http://lists.openstack.org/pipermail/openstack-dev/2018-March/128004.html). This will likely change much of how things are done, including the base schema.

The resulting database migration code is in sync.py (https://github.com/cdent/placedock/blob/self-contained-authless/sync.py) and it works. The resulting container, fronted by either apache2 or nginx behaves as desired. I ran many concurrent requests against it and discovered a bug with some database sync checks (https://bugs.launchpad.net/nova/+bug/1756151).

Now With Kubernetes!

Once I got that working, I wanted to make it work with kubernetes. minikube (https://github.com/kubernetes/minikube) makes this pretty easy and the instructions associated with minikube and kubernetes itself (https://kubernetes.io/docs/tutorials/stateless-application/hello-minikube/) are good. If you're on Linux, and using nested virtualization you may run into difficulties. I had enough that I decided to think about that some other time, only after I learned a lot more about virsh and esxi that I did before today.

As explained above, the main reason is for learning, because the scaling possibilities are lost by the throwaway database. It does, however, make for a pretty quick set up and explore, so I've written the steps here in case others would like to follow.

Assuming you've got a running `minikube` and you've got a checkout of the self-contained-authless branch (https://github.com/cdent/placedock/tree/self-contained-authless)

```
git clone -b self-contained-authless \
    git@github.com:cdent/placedock.git && \
    cd placedock
```

you can build, deploy and run the service with the following steps (in `bootstrap.sh` in the repo):

```
# use the right docker, the one in the minikube
eval `minikube docker-env`
# build the image via Dockerfile
docker build -t placedock:1.0 .
# create the deployment
kubectl apply -f deployment.yaml
# expose it
kubectl expose deployment placement-deployment --type=LoadBalancer
# get the URL
PLACEMENT=`minikube service placement-deployment --url`

# Get the version doc
curl -H 'x-auth-token: admin' $PLACEMENT
echo

type gabbi-run && gabbi-run -v all $PLACEMENT -- gabbi.yaml
```

It even runs some gabbi tests (https://gabbi.readthedocs.org) if you've got gabbi installed.

The most important change in the container is the way in which the placement WSGI application is run. In previous iterations a uwsgi protocol listener was run and needed to be fronted by an HTTP server. Based on some reading (https://github.com/nginxinc/kubernetes-ingress/issues/143#issuecomment-347814243), this isn't ideal in a kubernetes scenario, as a LoadBalancer service can take care of things for you. In that kind of setup, the uwsgi server in the container should listen on HTTP.

If everything goes well (you get a version document and maybe some passing gabbi tests) you now have a running placement service on which you create resource providers, set inventory, etc.

When you're done, the `cleanup.sh` script can tidy up the mess:

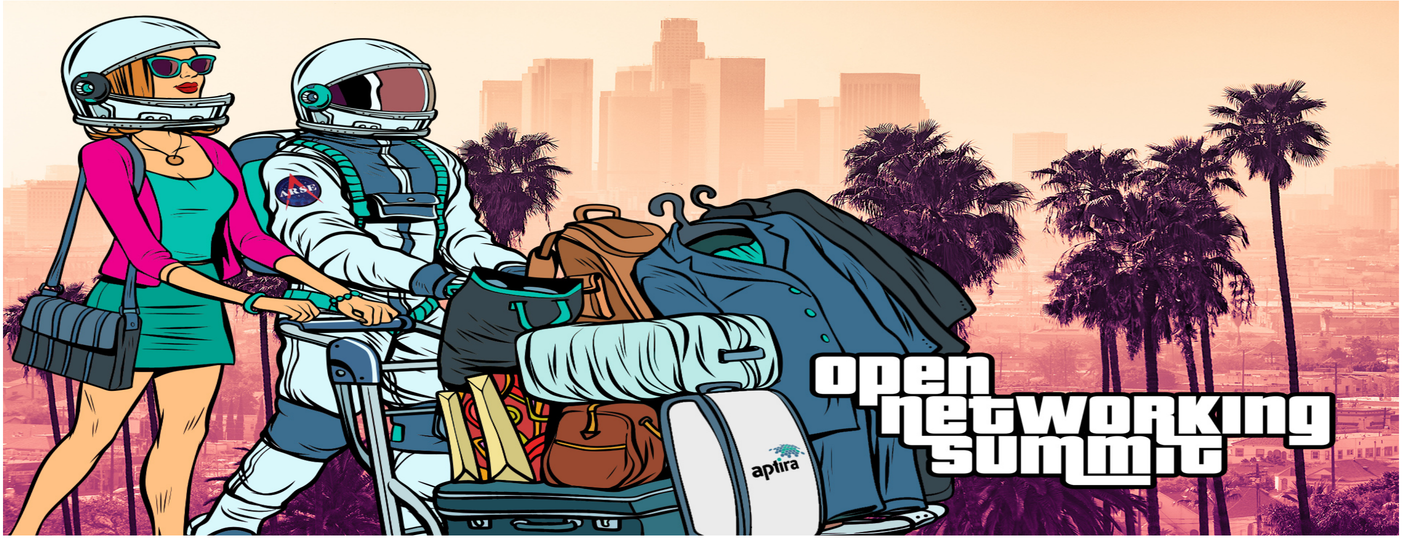
```
eval `minikube docker-env`
kubectl delete service placement-deployment
kubectl delete deployment placement-deployment
docker rmi placedock:1.0 -f
```

Please leave some comments if you have questions or there's a better way to do some of the things I've done. Next time we'll do some autoscaling with a shared database.

by Chris Dent at March 17, 2018 11:50 PM (https://anticdent.org/placement-container-playground-4.html)

Aptira (https://aptira.com)

Open Networking Summit: Aptira Solutionauts head to LA (https://aptira.com/open-networking-summit/)



Open Networking Summit North America

The Open Networking Summit (https://events.linuxfoundation.org/events/open-networking-summit-north-america-2018/) (ONS) is taking place at the Intercontinental Los Angeles Downtown from the 26th to 29th of March, and some of our Solutionauts will be there. This is the industry's premier Open Networking event, gathering enterprises, service providers and cloud providers across the ecosystem to share learnings, highlight innovation and discuss the future of Open Source Networking, including SDN, NFV, orchestration and the automation of Cloud, network, & IoT services.

We've been quite busy in the open networking space lately, recently joining the Linux Foundation Networking Project as a founding gold sponsor (https://aptira.com/aptira-joins-linux-foundation-networking-project/), as well as working on some very exciting ONAP (https://aptira.com/network-automation-onap/) projects. We've also launched Australia's first Open Networking User Group (AONUG!) (https://www.meetup.com/Australian-Open-Networking-User-Group/) with two meetups taking place in Sydney and Melbourne last month. This is where local open network enthusiasts can meet and learn more about Open Networking, covering topics such as ONAP, OPNFV, OpenDaylight and lots more. This meetup is currently looking for speakers, so if you're working on something open in networking we'd love to hear from you. We'll also be adding an Open Networking theme to the upcoming OpenStack Day (http://australiaday.openstack.org.au/) which will be held in Sydney in July. Stay tuned for more details regarding speaking opportunities, sponsorship and to reserve your place.

If you're attending ONS, please let us know. (https://aptira.com/contact/) Our Solutionauts would love to catch up with you to learn more about any Open Networking projects you are currently working on, or to discuss anything and everything open in networking. You can also also catch them at some of the event activities (https://events.linuxfoundation.org/events/open-networking-summit-north-america-2018/features-and-add-ons/event-experiences/), such as the 5k fun run or the Women in networking lunch (https://docs.google.com/forms/d/e/1FAIpQLSdei-hf0nk62VHKbtnl0MBzg_B8-0rbdCcVLjIXXlUzVl0biw/viewform). Please note our Solutionauts love to run (https://twitter.com/Aptira/status/790880349442215936), so if you'd prefer you can meet us at the finish for a much needed beverage.

You can register for the Open Networking Summit (https://events.linuxfoundation.org/events/open-networking-summit-north-america-2018/attend/register/) here. Hope to see you there!

The post Open Networking Summit: Aptira Solutionauts head to LA (https://aptira.com/open-networking-summit/) appeared first on Aptira (https://aptira.com).

by Jessica Field at March 17, 2018 07:39 AM (https://aptira.com/open-networking-summit/)

March 16, 2018

Doug Hellmann (https://doughellmann.com/blog)

openstack-summit-counter 0.2.0 (https://doughellmann.com/blog/2018/03/16/openstack-summit-counter-0-2-0/)

openstack-summit-counter is a plugin for python-openstackclient, the command line tool for interacting with OpenStack. This plugin helps you answer the summit registration question about how many summits you have attended in the past. What's new in 0.2.0? Add support for counting PTGs (contributed by Colleen Murphy)

by doug at March 16, 2018 10:05 PM (https://doughellmann.com/blog/2018/03/16/openstack-summit-counter-0-2-0/)

openstack-summit-counter 0.1.0 (https://doughellmann.com/blog/2018/03/16/openstack-summit-counter-0-1-0/)

openstack-summit-counter is a plugin for python-openstackclient, the command line tool for interacting with OpenStack. This plugin helps you answer the summit registration question about how many summits you have attended in the past. This is the first public release.

by doug at March 16, 2018 09:13 PM (https://doughellmann.com/blog/2018/03/16/openstack-summit-counter-0-1-0/)

Adam Young (http://adam.younglogic.com)

Generating a list of URL patterns for OpenStack services. (http://adam.younglogic.com/2018/03/generating-url-patterns/)

Last year at the Boston OpenStack summit, I presented on an Idea of using URL patterns to enforce RBAC (https://www.openstack.org/videos/boston-2017/per-api-role-based-access-control). While this idea is on hold for the time being, a related approach is moving forward building on top of application credentials (https://review.openstack.org/#/c/396331/). In this approach, the set of acceptable URLs is added to the role, so it is an additional check. This is a lower barrier to entry approach.

One thing I requested on the specification was to use the same mechanism as I had put forth on the RBAC in Middleware spec (https://specs.openstack.org/openstack/keystone-specs/specs/keystone/ongoing/role-check-from-middleware.html): the URL pattern. The set of acceptable URL patterns will be specified by an operator.

The user selects the URL pattern they want to add as a "white-list" to their application credential. A user could further specify a dictionary to fill in the segments of that URL pattern, to get a delegation down to an individual resource.

I wanted to see how easy it would be to generate a list of URL patterns. It turns out that, for the projects that are using the oslo-policy-in-code approach, it is pretty easy;

```
cd /opt/stack/nova
 . .tox/py35/bin/activate
(py35) [ayoung@ayoung541 nova]$ oslopolicy-sample-generator  --namespace nova | egrep "POST|GET|DELETE|PUT" | sed 's!#!!'
 POST  /servers/{server_id}/action (os-resetState)
 POST  /servers/{server_id}/action (injectNetworkInfo)
 POST  /servers/{server_id}/action (resetNetwork)
 POST  /servers/{server_id}/action (changePassword)
 GET  /os-agents
 POST  /os-agents
 PUT  /os-agents/{agent_build_id}
 DELETE  /os-agents/{agent_build_id}
...
```

Similar for Keystone

```
$ oslopolicy-sample-generator  --namespace keystone  | egrep "POST|GET|DELETE|PUT" | sed 's!# !!' | head -10
GET  /v3/users/{user_id}/application_credentials/{application_credential_id}
GET  /v3/users/{user_id}/application_credentials
POST  /v3/users/{user_id}/application_credentials
DELETE  /v3/users/{user_id}/application_credentials/{application_credential_id}
PUT  /v3/OS-OAUTH1/authorize/{request_token_id}
GET  /v3/users/{user_id}/OS-OAUTH1/access_tokens/{access_token_id}
GET  /v3/users/{user_id}/OS-OAUTH1/access_tokens/{access_token_id}/roles/{role_id}
GET  /v3/users/{user_id}/OS-OAUTH1/access_tokens
GET  /v3/users/{user_id}/OS-OAUTH1/access_tokens/{access_token_id}/roles
DELETE  /v3/users/{user_id}/OS-OAUTH1/access_tokens/{access_token_id}
```

The output of the tool is a little sub-optimal, as the oslo policy enforcement used to be done using only JSON, and JSON does not allow comments, so I had to scrape the comments out of the YAML format. Ideally, we could tweak the tool to output the URL patterns and the policy rules that enforce them in a clean format.

What roles are used? Turns out, we can figure that out, too:

```
$ oslopolicy-sample-generator  --namespace keystone  |  grep \"role:
#"admin_required": "role:admin or is_admin:1"
#"service_role": "role:service"
```

So only admin or service are actually used. On Nova:

```
$ oslopolicy-sample-generator  --namespace nova  |  grep \"role:
#"context_is_admin": "role:admin"
```

Only admin.

How about matching the URL pattern to the policy rule?
If I run

```
oslopolicy-sample-generator  --namespace nova  |  less
```

In the middle I can see an example like this (# marsk removed for syntax):

```
# Create, list, update, and delete guest agent builds

# This is XenAPI driver specific.
# It is used to force the upgrade of the XenAPI guest agent on
# instance boot.
 GET  /os-agents
 POST  /os-agents
 PUT  /os-agents/{agent_build_id}
 DELETE  /os-agents/{agent_build_id}
"os_compute_api:os-agents": "rule:admin_api"
```

This is not 100% deterministic, though, as some services, Nova in particular, enforce policy based on the payload.

For example, these operations can be done by the resource owner:

```
# Restore a soft deleted server or force delete a server before
# deferred cleanup
 POST  /servers/{server_id}/action (restore)
 POST  /servers/{server_id}/action (forceDelete)
"os_compute_api:os-deferred-delete": "rule:admin_or_owner"
```

Where as these operations must be done by an admin operator:

```
# Evacuate a server from a failed host to a new host
 POST  /servers/{server_id}/action (evacuate)
"os_compute_api:os-evacuate": "rule:admin_api"
```

Both map to the same URL pattern. We tripped over this when working on RBAC in Middleware, and it is going to be an issue with the Whitelist as well.

Looking at the API docs (https://developer.openstack.org/api-ref/compute/), we can see that difference in the bodies of the operations. The Evacuate call has a body like this (https://developer.openstack.org/api-ref/compute/#evacuate-server-evacuate-action):

```
{
    "evacuate": {
        "host": "b419863b7d814906a68fb31703c0dbd6",
        "adminPass": "MySecretPass",
        "onSharedStorage": "False"
    }
}
```

Whereas the forceDelete call has a body like this:

```
{
    "forceDelete": null
}
```

From these, it is pretty straight forward to figure out what policy to apply, but as of yet, there is no programmatic way to access that.

It would take a little more scripting to try and identity the set of rules that mean a user should be able to perform those actions with a project scoped token versus the set of APIs that are reserved for cloud operations. However, just looking at the admin_or_owner rule for most is sufficient to indicate that it should be performed using a scoped token. Thus, an end user should be able to determine the set of operations that she can include in a white-list.

by Adam Young at March 16, 2018 05:35 PM (http://adam.younglogic.com/2018/03/generating-url-patterns/)

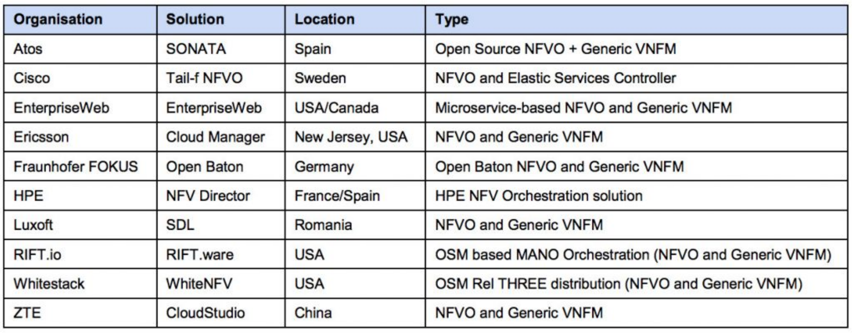
OpenStack Superuser (http://superuser.openstack.org)

Network Function Virtualization test results: Management and Orchestration solutions (http://superuser.openstack.org/articles/network-function-virtualization-test-results-management-and-orchestration-solutions/)

In the first post (http://superuser.openstack.org/articles/nfv-plugtests-part-1/) of this series, I shared a short introduction to NFV and its importance, as well as a summary about ETSI's second NFV Plugtests. In the second post (http://superuser.openstack.org/articles/etsi-nfv-plugtests-vim/) I shared the overall results of the tests and some details from the VIM perspective.

In this third and final post, I'll share a summary of the results related to the Management and Orchestration (MANO) solutions, which as you may already know are the software pieces (VNF Manager and NFV Orchestrator) that define how virtualized network services are built from the resources provided by the infrastructure through the VIM (VMs, containers, networks, etc), their life cycle, and of course, their behavior.

As you can see from the official report (https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/2nd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf) and the table below, 10 different MANO commercial solutions were tested, including six proprietary and four open-source based.



Organisation	Solution	Location	Type
Atos	SONATA	Spain	Open Source NFVO + Generic VNFM
Cisco	Tail-f NFVO	Sweden	NFVO and Elastic Services Controller
EnterpriseWeb	EnterpriseWeb	USA/Canada	Microservice-based NFVO and Generic VNFM
Ericsson	Cloud Manager	New Jersey, USA	NFVO and Generic VNFM
Fraunhofer FOKUS	Open Baton	Germany	Open Baton NFVO and Generic VNFM
HPE	NFV Director	France/Spain	HPE NFV Orchestration solution
Luxoft	SDL	Romania	NFVO and Generic VNFM
RIFT.io	RIFT.ware	USA	OSM based MANO Orchestration (NFVO and Generic VNFM)
Whitestack	WhiteNFV	USA	OSM Rel THREE distribution (NFVO and Generic VNFM)
ZTE	CloudStudio	China	NFVO and Generic VNFM

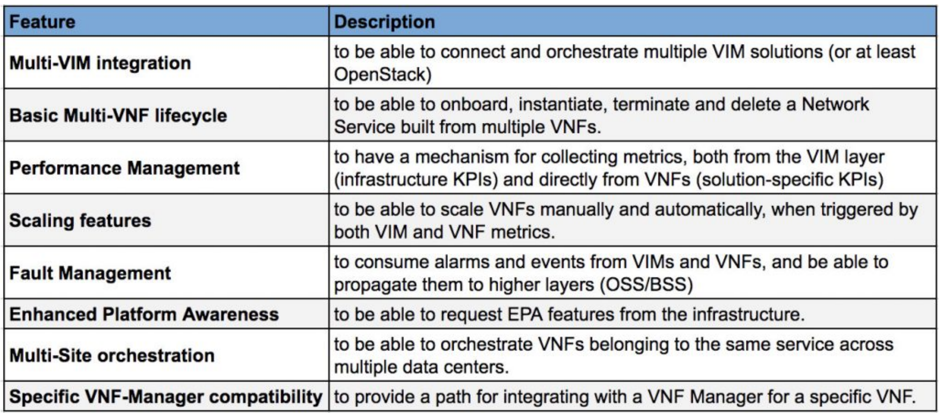
It's important to note that all MANO platforms included a Generic VNF Manager. This means that even though there is room for Specific VNF Managers at the standards, where a VNF brings its own manager (and there were tests for this kind of set-up), in most cases, a generic VNF Manager can cover all life cycle functions on most VNFs.

Another interesting observation is that at the VIM layer, 90 percent + of the vendors presented a single open-source based solution: OpenStack (https://www.openstack.org/), while at the MANO layer, only 40 percent of the platforms were open-source based.

Still, as opposed to the first Plugtests edition, where open-source projects participated directly, this time there were two commercial (branded) MANO solutions based on open-source software and both of those are based on Open Source MANO (https://osm.etsi.org/) (OSM)

OSM is an ETSI-hosted project to develop an open source NFV management and orchestration (MANO) software stack aligned with ETSI NFV. From my point of view, from the open-source MANO solutions available today (where OSM (http://osm.etsi.org), ONAP (https://www.onap.org/) and Open Baton (https://openbaton.github.io/) stand out), OSM has reached the highest level of maturity and production-readiness across the industry. It also has a lean approach and a large, growing and diverse community that keeps enhancing the project, which is about to launch its fourth release. That's why Whitestack selected OSM as the foundation for our WhiteNFV product. We believe that open source brings a brighter future for our industry and we are committed to launching ETSI-standard NFV deployments in the near future.

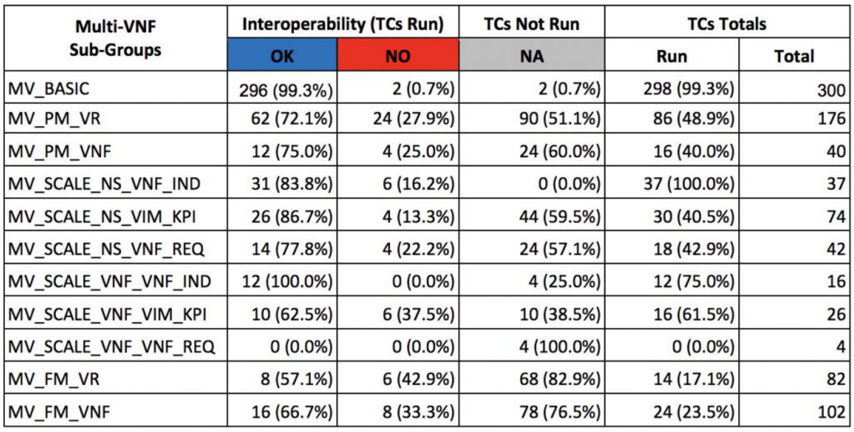
Back to the tests: as we did in the previous post (http://superuser.openstack.org/articles/etsi-nfv-plugtests-vim/) with the VIM, let's explore the main features expected from the MANO layer during the Plugtests:



Feature	Description
Multi-VIM integration	to be able to connect and orchestrate multiple VIM solutions (or at least OpenStack)
Basic Multi-VNF lifecycle	to be able to onboard, instantiate, terminate and delete a Network Service built from multiple VNFs.
Performance Management	to have a mechanism for collecting metrics, both from the VIM layer (infrastructure KPIs) and directly from VNFs (solution-specific KPIs)
Scaling features	to be able to scale VNFs manually and automatically, when triggered by both VIM and VNF metrics.
Fault Management	to consume alarms and events from VIMs and VNFs, and be able to propagate them to higher layers (OSS/BSS)
Enhanced Platform Awareness	to be able to request EPA features from the infrastructure.
Multi-Site orchestration	to be able to orchestrate VNFs belonging to the same service across multiple data centers.
Specific VNF-Manager compatibility	to provide a path for integrating with a VNF Manager for a specific VNF.

From those features, the ones that seem to have reached maturity are related to basic life cycle management and Multi-VIM integration (especially with OpenStack, including orchestration of advanced features like EPA, multi-site, etc.), however, performance management, fault management and scaling features seem to be the next set of features that need to reach that level of maturity.

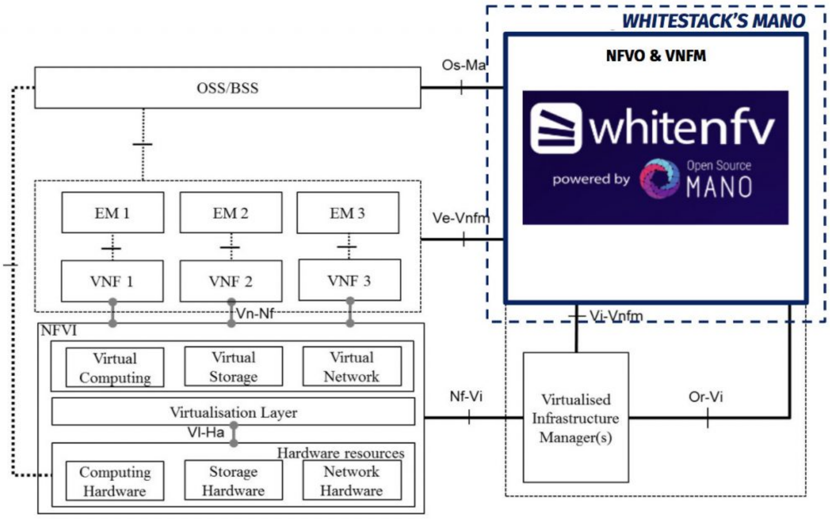
Let's take a look a the following table from the report:



Multi-VNF Sub-Groups	Interoperability (TCs Run)		TCs Not Run	TCs Totals	
	OK	NO	NA	Run	Total
MV_BASIC	296 (99.3%)	2 (0.7%)	2 (0.7%)	298 (99.3%)	300
MV_PM_VR	62 (72.1%)	24 (27.9%)	90 (51.1%)	86 (48.9%)	176
MV_PM_VNF	12 (75.0%)	4 (25.0%)	24 (60.0%)	16 (40.0%)	40
MV_SCALE_NS_VNF_IND	31 (83.8%)	6 (16.2%)	0 (0.0%)	37 (100.0%)	37
MV_SCALE_NS_VIM_KPI	26 (86.7%)	4 (13.3%)	44 (59.5%)	30 (40.5%)	74
MV_SCALE_NS_VNF_REQ	14 (77.8%)	4 (22.2%)	24 (57.1%)	18 (42.9%)	42
MV_SCALE_VNF_VNF_IND	12 (100.0%)	0 (0.0%)	4 (25.0%)	12 (75.0%)	16
MV_SCALE_VNF_VIM_KPI	10 (62.5%)	6 (37.5%)	10 (38.5%)	16 (61.5%)	26
MV_SCALE_VNF_VNF_REQ	0 (0.0%)	0 (0.0%)	4 (100.0%)	0 (0.0%)	4
MV_FM_VR	8 (57.1%)	6 (42.9%)	68 (82.9%)	14 (17.1%)	82
MV_FM_VNF	16 (66.7%)	8 (33.3%)	78 (76.5%)	24 (23.5%)	102

As you can see, basic VNF lifecycle testing had 300 tests with 99 percent success, followed by basic performance management (where metrics were grabbed from the VIM), with 176 tests executed and 72 percent success. We also see tests with less amount of participation, support and/or interoperability, with features related to performance management (where metrics were obtained directly from the VNFs), KPI-triggered scaling, and fault management.

In our particular case, with WhiteNFV/OSM, we took the chance to enhance our product by increasing our contributions to the open source project at those features required for this Plugtests event. In particular, we are strongly contributing to the existing OSM's Monitoring module and with new modules for achieving VNF metrics collection and auto-scaling, as well as other important features that should be starting to appear in Release 4 (April 2018). Plugtests gave us the unique opportunity to be side-by-side with engineers from the VNF vendors, so we started developing some of the features onsite during the sessions, taking advantage of the VNF vendor's feedback. Thanks to that opportunity we were able to successfully test VIM/VNF-based performance management and auto-scaling.



In summary, as basic life cycle features are working consistently across solutions, we can say that at the MANO level, NFV is ready for prime time, however, we should carefully select the right solution depending on the project's specific demands beyond basic VNF life cycle, especially when related to performance/fault management and scaling, or other more advanced ones like network slicing and service function chaining.

To close this series, I would like to thank the ETSI Plugtests team (http://www.etsi.org/about/what-we-do/plugtests) for organizing such an important event, and of course, the OSM Community (https://osm.etsi.org/) and the OpenStack Foundation (https://www.openstack.org/), for working so hard on these strong open source projects that enable us to prepare a world-class, end-to-end NFV solution.

If you're interested in any of the topics in the series, please leave a comment. Thanks for reading!

About the author

Gianpietro Lavado is a network solutions architect interested in the latest software technologies to achieve efficient and innovative network operations. He currently works at Whitestack, a company whose mission is to promote SDN, NFV, cloud and related deployments all around the world, including a new OpenStack distribution (https://www.openstack.org/marketplace/distros/distribution/whitestack/whitecloud).

This post first appeared on LinkedIn (https://www.linkedin.com/pulse/nfv-reaching-its-prime-time-part-3-2nd-etsi-plugtests-lavado/?lipi=urn%3Ali%3Apage%3Ad_flagship3_profile_view_base%3BTapawlukR6WmqEjgl%2FuDhg%3D%3D). Superuser is always interested in community content, get in touch at: editorATopenstack.org

Cover Photo (https://www.flickr.com/photos/rmlowe/3281353786/) // CC BY NC (https://creativecommons.org/licenses/by-nc/2.0/)

The post Network Function Virtualization test results: Management and Orchestration solutions (http://superuser.openstack.org/articles/network-function-virtualization-test-results-management-and-orchestration-solutions/) appeared first on Superuser (http://superuser.openstack.org).

by Gianpietro Lavado at March 16, 2018 02:11 PM (http://superuser.openstack.org/articles/network-function-virtualization-test-results-management-and-orchestration-solutions/)

James Page (https://javacruft.wordpress.com)

Winning with OpenStack Upgrades? (https://javacruft.wordpress.com/2018/03/16/winning-with-openstack-upgrades/)

On the Monday of the project teams gathering in Dublin a now somewhat familiar gathering of developers and operators got together to discuss upgrades – specifically fast forward upgrades but discussion over the day drifted into rolling upgrades and how to minimize downtime in supporting components as well. This discussion has been a regular feature over the last 18 months at PTG's, Forums and Ops Meetups.

Fast Forward Upgrades?

So what is a fast forward upgrade? A fast forward upgrade takes an OpenStack deployment through multiple OpenStack releases without the requirement to run agents/daemons at each upgrade step; it **does not** allow you to skip an OpenStack release – the process allows you to just **not run** a release as you pass through it. This enables operators using older OpenStack releases to catch up with the latest OpenStack release in as short an amount of time as possible, accepting the compromise that the cloud control plane is down during the upgrade process.

This is somewhat adjunct to a rolling upgrade, where access to the control plane of the cloud is maintained during the upgrade process by upgrading units of a specific service individually, and leveraging database migration approaches such as expand/migrate/contract (EMC) to provide as seamless an upgrade process as possible for an OpenStack cloud. In common with fast forward upgrades, releases cannot be skipped.

Both upgrade approaches specifically aim to not disrupt the data plane of the cloud – instances, networking and storage – however this may be unavoidable if components such as Open vSwitch and the Linux kernel need to be upgraded as part of the upgrade process.

Deployment Project Updates

The TripleO team have been working towards fast forward upgrades during the Queens cycle and have a 'pretty well defined model' for what they're aiming for with their upgrade process. They still have some challenges around ordering to minimize downtime specifically around Linux and OVS upgrades.

The OpenStack Ansible team gave an update – they have a concept of 'leap upgrades' which is similar to fast-forward upgrades – this work appears to lag behind the main upgrade path for OSA, which is a rolling upgrade approach which aims to be 100% online.

The OpenStack Charms team still continue to have a primary upgrade focus on rolling upgrades, minimizing downtime as much as possible for both the control and data plane of the Cloud. The primary focus for this team right now is supporting upgrades of the underlying Ubuntu OS between LTS releases with the imminent release of 18.04 on the horizon in April 2018, so no immediate work is planned on adopting fast-forward upgrades.

The Kolla team also have a primary focus on rolling upgrades, for which support starts at OpenStack Queens or later. There was some general discussion around automated configuration generation (https://review.openstack.org/#/c/520043/) using Oslo to ease migration between OpenStack releases.

No one was present to represent the OpenStack Helm team.

Keeping Networking Alive

Challenges around keeping the Neutron data-plane alive during an upgrade where discussed – this included:

- Minimising Open vSwitch downtime (http://docs.openvswitch.org/en/latest/intro/install/general/#hot-upgrading) by saving and restoring flows.
- Use of the 'neutron-ha-tool (https://github.com/SUSE-Cloud/cookbook-openstack-network/blob/neutron-ha-tool-maintenance/files/default/neutron-ha-tool.py)' from AT&T to manage routers across network nodes during an OpenStack cloud upgrade – there was also a bit of bike shedding on approaches to Neutron router HA in larger clouds. Plan are afoot to endeavor to make this part of the neutron code base.

Ceph Upgrades

We had a specific slot to discuss upgrade Ceph as part of an OpenStack Cloud upgrade; some deployment projects upgrade Ceph first (Charms), some last (TripleO) but there was general agreement that Ceph upgrades are pretty much always a rolling upgrade – i.e. no disruption to the storage services being provided. Generally there seems to be less pain in this area so it was not a long session.

Operator Feedback

A number of operators shared experiences of walking their OpenStack deployments through fast forward upgrades including some of the gotchas and trip hazards encountered.

Oath provided a lot of feedback on their experience of fast-forward upgrading their cloud from Juno to Ocata which included some increased complexity due to the move to using cells internally for Ocata. Ensuring compatibility between OpenStack and supporting projects was one challenge encountered – for example, snapshots worked fine with Juno and Libvirt 1.5.3, however on upgrade live snapshots where broken until Libvirt was upgraded to 2.9.0. Not all test combinations are covered in the gate!

Some of these have been shared on the OpenStack Wiki (https://wiki.openstack.org/wiki/Fast_forward_upgrades).

Upgrade SIG

Upgrade discussion has become a regular fixture at PTG's, Forums, Summits and Meetups over the last few years; getting it right is tricky and the general feeling in the session was that this is something that we should talk about more between events.

The formation of an Upgrade SIG (https://wiki.openstack.org/wiki/Upgrade_SIG) was proposed and supported by key participants in the session. The objective of the SIG is to improve the overall upgrade process for OpenStack Clouds, covering both offline 'fast-forward' and online 'rolling' upgrades by providing a forum for cross-project collaboration between operators and developers to document and codify best practice for upgrading OpenStack.

The SIG will initially be led by Lujin Luo (Fujitsu), Lee Yarwood (Redhat) and myself (Canonical) – we'll be sorting out the schedule for bi-weekly IRC meetings in the next week or so – OpenStack operators and developers from across all projects are invited to participate in the SIG and help move OpenStack life cycle management forward!

by JavaCruft at March 16, 2018 10:00 AM (https://javacruft.wordpress.com/2018/03/16/winning-with-openstack-upgrades/)

Aptira (https://aptira.com)

New Solutionaut Troops: Welcome Prashant, Shane and Adam to the Aptira Army. (https://aptira.com/new-solutionaut-troops-welcome-shane-adam-prashant-aptira-army/)

> The Aptira Army of Solutionauts continues to expand with 3 new Solutionaut troops joining the Aptira Army, and more on the way!

Prashant Abkari

Architecturenaut

Prashant joins Aptira after working mainly on building telco products starting from 2G roaming solutions to the 4G VoLTE. He has also been involved in virtualization of some of the LTE core products in Nokia. Outside of work, Prashant enjoys to eat and drink, go to the movies, and play cricket. Without them life would be bland.

Shane Howearth

Appstronaut

Shane has a broad background that stems from a love of computer science and networking. He also has interests in Machine Learning, AI (graph search), writing simple device drivers, mechanical keyboards, and LOVES Linux! Outside of work, Shane walks – a LOT! Last year it was ~100 Km/week, but now it's ~55Km/week. Back in New Zealand he was a volunteer for Forest and Bird, helping with pest management (trapping and poisoning pests).

Adam Russell

Mission Controllernaut

Adam has been working in the IT business for 40 years, after turning his back on a career as a zoologist and started learning to code in the mid 70's. Starting as a developer in the nascent days of the PC industry, Adam has spent the last 35 years working on complex systems integration projects both in pre and post Sales roles. Since first taking the role of a Project Manager in the early 80's, Adam has worked on many large-scale enterprise integration projects, mostly involving custom software, vendor product integration and a global supply chain. For much of the last 25 years these projects have been in the Telco domain and Adam has worked with some of the largest Telco's in the world to help them integrate solutions in both BSS and OSS domains. Check out Adam's views on project management at http://www.adamonprojects.com (http://www.adamonprojects.com)

Our troops are scattered all over the globe, but be sure to catch some of them at one of our upcoming Open Networking meetups (https://www.meetup.com/Australian-Open-Networking-User-Group/), an OpenStack meetup (https://www.meetup.com/Australian-OpenStack-User-Group/), or the Open Infrastructure Day (https://australiaday.openstack.org.au/).

The post New Solutionaut Troops: Welcome Prashant, Shane and Adam to the Aptira Army. (https://aptira.com/new-solutionaut-troops-welcome-shane-adam-prashant-aptira-army/) appeared first on Aptira (https://aptira.com).

by Jessica Field at March 16, 2018 06:49 AM (https://aptira.com/new-solutionaut-troops-welcome-shane-adam-prashant-aptira-army/)

March 15, 2018

Graham Hayes (http://graham.hayes.ie/)

Dublin PTG Summary (http://graham.hayes.ie/posts/dublin-ptg-summary/)

Not getting on a plane was a nice change for an OpenStack event :) - especially as it looks like I would not have made it home for a few days.

Cross Project Days (Monday / Tuesday)

These days are where I think the major value of the PTG is. The cross project days feel like the Summit of old, with more time for topics, and less running around a conference centre trying to cram 2 weeks worth of talks / developer sessions, and other meetings into a few days.

Unified Limits / olso.limits / keystone stored limit data

First up for me was the keystone based limits API for unifying quota data in keystone. It was decided to create oslo.limts (olso.limits repo (https://review.openstack.org/#/c/550491/), olso.limits spec (http://lists.openstack.org/pipermail/openstack-dev/2018-March/128175.html) & oslo.limits etherpad (https://etherpad.openstack.org/p/unified-limits-rocky-ptg)). The keystone team already created a keystone-limits-api (https://developer.openstack.org/api-ref/identity/v3/index.html#unified-limits) that is currently experimental, and the feeling in the room was that we should try and implement it using a new oslo library to find where changes need to be made.

The migration procedure was discussed, and how we (the services) would need to run multiple quota systems for quite a few cycles, due to partial upgrades that happen in OpenStack. [1] (http://graham.hayes.ie/posts/dublin-ptg-summary/#id2) Possible implementations were discussed, and the oslo.limits etherpad (https://etherpad.openstack.org/p/unified-limits-rocky-ptg) has a good overview of them.

olso.healthcheck

This is an idea that I have been very interested in since it was discussed in Sydney. We actually had 3 sessions on this in Dublin, across 3 different cross project rooms - API-SIG, Oslo and Self Healing SIG.

Overall, most people were receptive - the commentary is that the spec is too wordy, and contains my favorite description:

> It feels like OpenStack, but not really in a good way.

After listening to feedback, and talking offline to a few people I think I have a handle on where the issues are, and I think I have a rough spec I can work flesh out over the next few days. I think I will just start writing code at that point as well - I think with a more concrete example it could help clear up issues for people.

Edge Computing

I stopped by the Edge room on the Tuesday to listen in on the discussion. Personally, I really think this group needs to stop bikesheding on, well, everything, and actually go and implement a POC and see what breaks.

The push still seems to be "make OpenStack work on the edge" instead of (what I think is the quickest / most productive way forward) "write extra tooling to orchestrate OpenStack on the edge."

There was some interesting items brought up, like Glance, and image / data residency. I think that actually engaging with the Glance team might have been helpful, as they were completely unaware that the discussion was being held, but the concepts raised sounded interesting.

I lasted about an hour or so, before I gave up. From my limited exposure, it sounded exactly like the discussions I have heard on the Edge Calls, which were the same as the ones I heard in Sydney.

Designate Sessions

The Designate room was a quiet enough affair, but it marks the first time since the PTG's started that getting a dedicated room was justified. We did some onboarding with new folks, and laid out a plan for the cycle.

The plan so far looks like this:

- DNSSEC

> - A starting point, using signing keys in the Designate database, which we can use as a jumping point to storing keys in a HSM / Barbican
> - People are currently looking at PowerDNS's inline signing as a short term solution.

- Docs (it will never **not** be a priority :))
- Shared Zones
- Improve the UI

> - This really relies on us either `rm -rf openstack/designate-dashboard` or finding people who understand Angular.

TC / QA / Tempest / Trademark Programs

If you follow the mailing list, or openstack/governance (https://github.com/openstack/governance) reviews, you may have seen a long running discussion over where tempest tests used for Trademark Programs should go. From memory I seem to remember this being raised in Boston, but it could have been Barcelona. There was tension between QA, me, the InterOp Work Group, and others about the location. Chris Dent covered this pretty well in his TC updates (https://anticdent.org/tag/tc.html) over the last while, so I am not going to rehash it, but it does look like we finally have some sort of agreement on the location (https://review.openstack.org/#/c/550571/), after what was 2 other times I thought we had agreement :).

Board of Directors Meeting

The OpenStack Board of Directors met on the first day of the PTG. This was an issue in its own right, which was highlighted in a thread on the foundation list (http://lists.openstack.org/pipermail/foundation/2018-January/002558.html). Thankfully, I have been told that this situation will not happen again (I can't seem to find any record of the decision, so it may have been an informal board discussion, but if anyone from the board is reading, replying to the foundation list (http://lists.openstack.org/pipermail/foundation/2018-January/002558.html) would be great.).

As it met on day one, I didn't get to see much - I arrived to wait for the Add On Trademark program approvals, and happened to catch a very interesting presentation (https://docs.google.com/presentation/d/1_N7xhzwk6HzCl0fMm_cWfQ2UFu8bOaZYD2OOAV4y5yQ/edit#slide=id.g33768e8068_2_244) by Dims, Chris and Melvin. I then got to see the board approve DNS as a Trademark add on, which is great for the project, and people who want a constant DNSaaS experience across multiple clouds.

Johnathon Price's Board Recap (http://lists.openstack.org/pipermail/foundation/2018-March/002570.html) is also a good overview, with links to things that were presented at the meeting.

The Board, Community, and how we interact

One topic that was highlighted by the TC / QA / Tempest / Trademark Programs (http://graham.hayes.ie/posts/dublin-ptg-summary/#tc-qa-tempest-trademark-programs) discussion was that the QA team is very under resourced. This, combined with the board discussing future of the PTGs (https://anticdent.org/tc-report-18-10.html#talking-about-the-ptg-at-the-ptg) due to cost makes me very worried.

The foundation has (in my mind) two main focuses.

1. Promote the use of the software or IP produced by people working on projects under the foundation, and protect its reputation.
2. Empower the developers working on said software or IP to do so.

In my eyes, Trademark programs are very much part of #1, and the board should either:

1. Fund / find resources for the QA team, to ensure they have enough bandwidth to maintain all trademark programs, the associated tests, and tooling.
2. Fund / find a team that does it separately, but removes the entire burden from the QA team.

The PTG falls firmly under #2. I was initially a PTG skeptic, but I really think it works as an event, and adds **much** more value than the old mid-cycles did. I understand it has problems, but without it, teams will go back to the mid cycles, which may have looked cheaper at first glance, but for some people either meant multiple trips, or missing discussions.

One very disappointing thing to see was the list of Travel Support Program donors - there was some very generous individuals in the community that stood up and donated, but none of the corporate foundation members contributed. This, with members being added to the foundation that seem to stop at paying the membership fee (see tencent (http://stackalytics.com/?release=all&company=tencent) who were added at the Sydney board meeting), makes me wonder about the value placed on the community by the board.

I know the OpenStack Foundation is diversifying its portfolio of projects beyond just the OpenStack Project (this is going to get confusing :/), but we should still be supporting the community that currently exists.

Other great write ups

This turned into a bit of a PTG + 2 weeks after update, so here are some other write ups I have read over the last week or so, and prompted me to remember things that I would have otherwise forgotten.

- Chris Dent (https://anticdent.org/tc-report-18-10.html)
- Colleen Murphy (http://www.gazlene.net/dublin-ptg.html)

- Adam Spiers (https://blog.adamspiers.org/2018/03/09/openstack-ptg-dublin/)
- Mark Voelker (http://markvoelker.github.io/blog/dublin-ptg-edge-sessions/)

The Hotel

And, I saved the best until last. The Croke Park Hotel (https://www.doylecollection.com/hotels/the-croke-park-hotel) was absolutely amazing during the conference. When we needed to leave the main venue on Thursday, they managed the transition of a few hundred developers into all the public spaces we could find extremely well. They kept us fed, watered and happy the entire time we were in the hotel. The fact they managed to do this, while not leaving the hotel to go home and sleep themselves! I cannot say enough good things about them, and encourage anyone who is looking for a hotel in Dublin to stay there, or is running an event to use Croke Park and the hotel.

[1] (http://graham.hayes.ie/posts/dublin-ptg-summary/#id1) I have heard of companies running Newton / Ocata Designate (and other projects) on clouds as old as Liberty.

by Graham Hayes at March 15, 2018 07:23 PM (http://graham.hayes.ie/posts/dublin-ptg-summary/)

OpenStack Superuser (http://superuser.openstack.org)

What's new at the Vancouver Summit (http://superuser.openstack.org/articles/whats-new-vancouver-summit/)

With 85,000 community members and over 6 million compute cores in production, OpenStack is already one of the largest and most active open source communities in history. The OpenStack Foundation (OSF) and events are the home for people who are build and operate infrastructure.

Superuser has been talking to these extraordinary users for four years and recognizing them with the coveted Superuser Award (http://superuser.openstack.org/awards/) for the past three. Whether they're unraveling the mysteries of space (http://superuser.openstack.org/articles/openstack-operator-spotlight-cern/) or running high-speed train networks (http://superuser.openstack.org/articles/sydney-superuser-award-nominee-china-railway-corporation/), what can be done with open infrastructure is impressive.

We've also been hearing how infrastructure use cases are evolving — users are combining virtual machines, containers and bare metal in new architectures — and the walls of the data centers themselves are breaking down as computing power moves to the edge.

As use cases for infrastructure have evolved, so has the OSF. With a focus on integration and cross-project collaboration, the OpenStack community started to add new top level projects under the OSF including Kata Containers (http://superuser.openstack.org/articles/inside-kata-containers/) and the Summit also reflects these changes.

What's new at the Vancouver Summit

- The Summit schedule (https://www.openstack.org/summit/vancouver-2018/summit-schedule), published today, features over 100 sessions organized by use cases: including AI and machine learning; HPC; edge computing; NFV; container infrastructure; and public, private and multi-cloud strategies. The Vancouver Summit will gather the open infrastructure operators and developers to drive these use cases forward by focusing on the integration and operations of these technologies.
- In addition to OpenStack, there will be featured sessions on the newest project at the Foundation — Kata Containers — and a strong focus on other open source technologies relevant to infrastructure operators, including Kubernetes, Docker, Ansible, Ceph, Istio, Envoy, Spinnaker, OpenContrail, ONAP, OPNFV and many more.
- The second edition of OpenDev (http://www.opendevconf.com) will be co-located at the Vancouver Summit and focus on CI/CD, featuring collaborative working sessions, architecture show-and-tells and demos. Sessions are backed by a programming committee including Google, Netflix, Spinnaker, Mesosphere, TravisCI, Zuul, Ericsson, GoDaddy and the University of Cambridge.
- More than 30 new and featured speakers presenting their use cases including JPMorgan Chase & Co., Wells Fargo, Progressive Insurance, Target, Gap Inc., Ohio State University, Italy's University of Messina, Oath Inc. and AT&T. Stay tuned on Superuser as we preview these user stories.

Want to see it in person? Ticket prices for the Summit increase April 4 so register now (https://www.openstack.org/summit/vancouver-2018/) to join the developers and operators driving the future of open infrastructure. If you need help covering costs, apply for the Travel Support Program (http://bit.ly/2Fv01cN) by March 22.

Get involved

Community is central to OSF. In addition to the above events, check out the contributor portal (https://www.openstack.org/community) for how to get involved with code and documentation, user groups, users and sponsorship.

During the Summit, you can also attend the Forum (https://wiki.openstack.org/wiki/Forum). With three parallel tracks over four days, OpenStack users and developers gather to brainstorm requirements for the next release, give feedback on the past version and have strategic discussions that go beyond just one release cycle. Development teams, working groups and interested individuals talk about release requirements, connect with community members about cross-project collaboration and exchange feedback between design and operations members.

The post What's new at the Vancouver Summit (http://superuser.openstack.org/articles/whats-new-vancouver-summit/) appeared first on Superuser (http://superuser.openstack.org).

by Superuser at March 15, 2018 03:50 PM (http://superuser.openstack.org/articles/whats-new-vancouver-summit/)

RDO (https://blogs.rdoproject.org)

March 14 Community Blog Round-up (https://blogs.rdoproject.org/2018/03/march-14-community-blog-round-up/)

Hardware burn-in in the CERN datacenter by Tim Bell

> During the Ironic sessions at the recent OpenStack Dublin PTG in Spring 2018, there were some discussions on adding a further burn in step to the OpenStack Bare Metal project (Ironic) state machine. The notes summarising the sessions were reported to the openstack-dev list. This blog covers the CERN burn in process for the systems delivered to the data centers as one example of how OpenStack Ironic users could benefit from a set of open source tools to burn in newly delivered servers as a stage within the Ironic workflow.

Read more at http://openstack-in-production.blogspot.com/2018/03/hardware-burn-in-in-cern-datacenter.html (http://openstack-in-production.blogspot.com/2018/03/hardware-burn-in-in-cern-datacenter.html)

Using Docker macvlan networks by Lars Kellogg-Stedman

> A question that crops up regularly on #docker is “How do I attach a container directly to my local network?” One possible answer to that question is the macvlan network type, which lets you create “clones” of a physical interface on your host and use that to attach containers directly to your local network. For the most part it works great, but it does come with some minor caveats and limitations. I would like to explore those here.

Read more at http://blog.oddbit.com/2018/03/12/using-docker-macvlan-networks/ (http://blog.oddbit.com/2018/03/12/using-docker-macvlan-networks/)

A New Fencing Mechanism (TBD) by Andrew Beekhof

> Protecting Database Centric Applications. In the same way that some application require the ability to persist records to disk, for some applications the loss of access to the database means game over – more so than disconnection from the storage.

Read more at http://blog.clusterlabs.org/blog/2018/tbd-fencing (http://blog.clusterlabs.org/blog/2018/tbd-fencing)

Generating a Callgraph for Keystone by Adam Young

> Once I know a starting point for a call, I want to track the other functions that it calls. pycallgraph will generate an image that shows me that.

Read more at http://adam.younglogic.com/2018/03/callgraph-keystone/ (http://adam.younglogic.com/2018/03/callgraph-keystone/)

Inspecting Keystone Routes by Adam Young

> What Policy is enforced when you call a Keystone API? Right now, there is no definitive way to say. However, with some programmatic help, we might be able to figure it out from the source code. Lets start by getting a complete list of the Keystone routes.

Read more at http://adam.younglogic.com/2018/03/inspecting-keystone-routes/ (http://adam.younglogic.com/2018/03/inspecting-keystone-routes/)

SnowpenStack by rbowen

> I’m heading home from SnowpenStack and it was quite a ride. As Theirry said in our interview at the end of Friday (coming soon to a YouTube channel near you), rather than spoiling things, the freak storm and subsequent closure of the event venue served to create a shared experience and camaraderie that made it even better.

Read more at http://drbacchus.com/snowpenstack/ (http://drbacchus.com/snowpenstack/)

Expiry of VMs in the CERN cloud by Jose Castro Leon

> The CERN cloud resources are used for a variety of purposes from running compute intensive workloads to long running services. The cloud also provides personal projects for each user who is registered to the service. This allows a small quota (5 VMs, 10 cores) where the user can have resources dedicated for their use such as boxes for testing. A typical case would be for the CERN IT Tools training where personal projects are used as sandboxes for trying out tools such as Puppet.

Read more at http://openstack-in-production.blogspot.com/2018/03/expiry-of-vms-in-cern-cloud.html (http://openstack-in-production.blogspot.com/2018/03/expiry-of-vms-in-cern-cloud.html)

My 2nd birthday as a Red Hatter by Carlos Camacho

> This post will be about to speak about my experience working in TripleO as a Red Hatter for the last 2 years. In my 2nd birthday as a Red Hatter, I have learned about many technologies, really a lot… But the most intriguing thing is that here you never stop learning. Not just because you just don’t want to learn new things, instead, is because of the project’s nature, this project… TripleO…

Read more at https://www.anstack.com/blog/2018/03/01/2nd-birthday-as-a-red-hatter.html (https://www.anstack.com/blog/2018/03/01/2nd-birthday-as-a-red-hatter.html)

by Mary Thengvall at March 15, 2018 02:27 AM (https://blogs.rdoproject.org/2018/03/march-14-community-blog-round-up/)

March 14, 2018

OpenStack Superuser (http://superuser.openstack.org)

Kata Containers and OpenStack support open container standards (http://superuser.openstack.org/articles/openstack-foundation-joins-open-container-initiative-kata/)

To continue the strategy of cross-community collaboration and supporting standard technologies, today the OpenStack Foundation joined (https://www.openstack.org/news/view/373/open-container-initiative-continues-momentum-with-new-members) the Open Container Initiative (OCI). OCI was established in 2015 by Docker, CoreOS and other industry leaders to create open industry standards around container formats and runtimes.

The move for the OpenStack Foundation to join OCI (https://www.opencontainers.org/) was driven by the new Kata Containers project, launched in December 2017, with a mission to combine the speed and performance of containers with the isolation and security of virtual machines.

The Kata Containers project delivers extremely lightweight virtual machine hosted containers that confirm to the OCI runtime specification. The team plans to get involved with OCI-related tooling, compatibility and interoperability testing. The Kata Containers community is currently working on their 1.0 release scheduled for late Q2 and you can learn more about their mission or contribute to their efforts at katacontainers.io (https://katacontainers.io/) .

“OCI is an important place for the container ecosystem to come together and drive common formats across tools and deployments. We’re excited to support OCI specs in the Kata Containers project and look forward to working more closely with the OCI community to support the future of container image standardization” says James Kulina (https://www.linkedin.com/in/james-kulina-4299b6100/), COO of Hyper (https://hyper.sh/) and a member of the Kata Containers Working Committee.

Under the OpenStack project, there’s also a team packaging the cloud services as OCI compliant images and industry standard tooling, which makes it easier to deploy and manage the lifecycle of OpenStack clouds using services like OpenStack-Helm (https://wiki.openstack.org/wiki/Openstack-helm) and Kolla (https://wiki.openstack.org/wiki/Kolla).

In the future, the OCI community plans to launch a certification program for projects, which will be another opportunity for both Kata Containers and OpenStack to participate.

Get involved

To get involved with OCI efforts, you can join the community mailing list. (https://groups.google.com/a/opencontainers.org/forum/#!forum/dev)

To follow the OpenStack community efforts, check out the contributor portal (https://www.openstack.org/community) for how to get involved with code and documentation, user groups, users and sponsorship.

If you’re new to Kata, here are some ways to connect with the community.

Code and documentation:
GitHub / How to contribute
(https://github.com/kata-containers)Overview deck
(https://katacontainers.io/docs/Kata-Containers-On-boarding-Deck-for-Website01022018.pdf)Overview One Pager (https://katacontainers.io/docs/kata-containers-1pager.pdf)

Communication:
Mailing list
(http://lists.katacontainers.io)Freenode IRC: #kata-dev (http://webchat.freenode.net/?channels=kata-dev) and #kata-general
(http://webchat.freenode.net/?channels=kata-general)Slack channel (https://katacontainers.slack.com/) or invite
(http://bit.ly/KataSlack)Social:
Twitter
(https://twitter.com/KataContainers)Facebook (https://www.facebook.com/KataContainers)

Cover Photo (https://www.flickr.com/photos/16967950@N00/37289676701/in/photostream/) // CC BY NC (https://creativecommons.org/licenses/by-nc/2.0/)

The post Kata Containers and OpenStack support open container standards (http://superuser.openstack.org/articles/openstack-foundation-joins-open-container-initiative-kata/) appeared first on Superuser (http://superuser.openstack.org).

by Lauren Sell at March 14, 2018 05:14 PM (http://superuser.openstack.org/articles/openstack-foundation-joins-open-container-initiative-kata/)

Ed Leafe (https://blog.leafe.com)

Dublin PTG Recap (https://blog.leafe.com/dublin-ptg-recap/)

We recently held the OpenStack PTG for the Rocky cycle. The PTG ran from Monday to Friday, February 26 – March 2, in Dublin, Ireland. So of course the big stuff to write about would be the interesting meetings between the teams, and the discussions about future development, right? Wrong! The big news from the … Continue reading "Dublin PTG Recap" (https://blog.leafe.com/dublin-ptg-recap/)

by ed at March 14, 2018 04:48 PM (https://blog.leafe.com/dublin-ptg-recap/)

March 13, 2018

OpenStack Superuser (http://superuser.openstack.org)

Check out these OpenStack project updates (http://superuser.openstack.org/articles/check-out-these-openstack-project-updates/)

If you’re interested in getting up to speed on what’s next for OpenStack software (https://www.openstack.org/zh_CN/software/project-navigator) in the current development cycle, project updates from the recent Project Teams Gathering (PTG) are a good place to start.

RDO’s Rich Bowen (https://twitter.com/rbowen) sat down with attendees in Dublin to talk about what features have just landed with the Queens release and what’s coming for Rocky. In his video interviews (https://www.youtube.com/playlist?list=PLOuHvpVx7kYnVF3qjvIw-Isq2sQkHhy7q) you’ll hear from the project team leaders (PTLs) and core contributors about what they’re accomplished, where they’re heading for future releases (https://releases.openstack.org/) plus how you can get involved and influence the roadmap. (http://www.openstack.org/software/roadmap/)

Here are some excerpts:

Compute (Nova)

Matt Riedemann and Melanie Witt talk (https://www.youtube.com/watch?v=2usdGvQIYs0) about what's new to this release (https://docs.openstack.org/releasenotes/nova/queens.html) including multi-cells support for Cells V2, support for vGPUs and QEMU native LUKS decryption (https://github.com/openstack/nova/commit/e9ae961d92faaaed1f405c6b67815ea789d10317) and volume multi-attach (https://superuser.openstack.org/articles/meet-volume-multi-attach-great-new-feature-openstack-queens/). At the PTG, the developers were starting conversations with users including Go Daddy and CERN about implementing the new features. "It's really cool that people are actually upgrading and migrating to this stuff that we've been working on for a couple of years now," Riedemann says. "They're starting to use it and they're finding problems but that's good — they're coming back to us and we're working together."

Block storage (Cinder)

In addition to the much-talked about volume multi-attach (https://superuser.openstack.org/articles/meet-volume-multi-attach-great-new-feature-openstack-queens/), PTL Jay Bryant says (https://www.youtube.com/watch?v=Y979xuRXh1o) there are a couple of other updates that the end user will appreciate in this release. These include default policies in code, the ability to create a volume from a backup and better quota handling. This last item, which Byrant says is still in a work in progress, aims to improve the way quotas are handled for each of the different settings within Cinder. "It was possible to actually use more storage on your backend than you intended, so now we've added checks 0to make sure that creating a snapshot isn't going to go over the available storage quota you set." Bryant marks this as an area for further improvement but at least it will avoid the head-scratching situation "where Cinder thinks that 200 gig is getting used but the back end thinks I've got 350." The future focus for Cinder is on usability and improvements and that users can expect "fewer major enhancements with more focus on improving functionality."

DNS-as-a-service (Designate)

"The last cycle was fairly quiet, we were recovering from a period of low activity so we built up some debt… we spent this cycle paying it off," says (https://www.youtube.com/watch?v=EuvWTul_V5k) PTL Graham Hayes. "We started the groundwork for improving our upgrades, working towards rolling upgrades – meaning you can upgrade some parts of Designate and not others, so you have zero down time." The foundation for that was laid for that during Queens and merged right after the release, he adds. The team is also working on improving the dashboard and continuous live integration – keeping an eye on ensuring that drivers are working properly in conjunction with the interoperability team. At the PTG, the board approved the DNS trademark program, "so if you have an OpenStack product or a public cloud you can have a certified of the stock DNS program as well too offer a standard it's an interoperable DNS experience for your users."

TripleO/Ansible

Because the OpenStack TripleO (https://wiki.openstack.org/wiki/TripleO) (that's OpenStack on OpenStack) project focuses on deploying, upgrading and operating OpenStack clouds, it has to interact with pretty much all of the other OpenStack projects.

"During the Queens cycle, I focused pretty extensively on Ansible (https://docs.openstack.org/openstack-ansible/latest/) integration and moving more towards an Ansible-driven architecture," says (https://www.youtube.com/watch?v=QHZkPlkeDYc&list=PLOuHvpVx7kYnVF3qjvIw-Isq2sQkHhy7q&t=0s&index=8) developer James Slagle. Responding to the rapid growth of the project recently, Slagle says "We've integrated with a lot of the OpenStack services, we have a lot of contributions from all the different projects for configuring deployment of their services and so we've been converting a lot of that to Ansible as well as using

Ansible for a lot of upgrade tasks. As for what's next, Slagle says "During rocky we want to switch to this Ansible deployment mode by default as well as move all of our CI over to use that new model going forward, we're going to deprecate the old model. We'll also be looking at more Kubernetes integration to also be deployed with real ansible and also integrating with Ceph Ansible more, to use that as a deployment mechanism for Ceph as well."

For more on the other OpenStack project updates from the PTG, here's a roundup (https://www.openstack.org/blog/?p=8361) from the developer's mailing list (http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack-dev):

- Documentation and i18n (http://lists.openstack.org/pipermail/openstack-dev/2018-March/127936.html)
- Mistral (http://lists.openstack.org/pipermail/openstack-dev/2018-March/127988.html), workflow service
- Kolla, (http://lists.openstack.org/pipermail/openstack-dev/2018-March/128044.html)production-ready containers and deployment tools
- Oslo (http://lists.openstack.org/pipermail/openstack-dev/2018-March/128055.html), shared Python libraries across OpenStack
- Interoperability (http://lists.openstack.org/pipermail/openstack-dev/2018-March/127994.html)
- QA (http://lists.openstack.org/pipermail/openstack-dev/2018-March/128002.html)
- Ironic, (http://lists.openstack.org/pipermail/openstack-dev/2018-March/128085.html) bare metal
- Cyborg (http://lists.openstack.org/pipermail/openstack-dev/2018-March/128094.html), acceleration-as-a-service
- First Contact SIG (http://lists.openstack.org/pipermail/openstack-dev/2018-March/127937.html)

Get involved!

If you're new to OpenStack, check out the contributor portal (https://www.openstack.org/community) for how to get involved with code and documentation, user groups, users and sponsorship. You can also get up and running with the project navigator (https://www.openstack.org/software/project-navigator) or join the mailing lists (https://wiki.openstack.org/wiki/Mailing_Lists) or IRC-channels (https://wiki.openstack.org/wiki/IRC).

The post Check out these OpenStack project updates (http://superuser.openstack.org/articles/check-out-these-openstack-project-updates/) appeared first on Superuser (http://superuser.openstack.org).

by Superuser at March 13, 2018 04:13 PM (http://superuser.openstack.org/articles/check-out-these-openstack-project-updates/)

Chris Dent (https://anticdent.org/)

TC Report 18-11 (https://anticdent.org/tc-report-18-11.html)

Much of the activity of the TC in the past week has been devoted to discussing and sometimes arguing about two pending resolutions: Location of Interop Tests and Extended Maintenance (links below). While there has been a lot of IRC chat, and back and forth on the gerrit reviews, it has resulted in things moving forward.

Since I like to do this: a theme I would identify from this week's discussions is continued exploration of what the TC feels it can and should assert when making resolutions. This is especially apparent in the discussions surrounding Interop Tests. The various options run the gamut from describing and enforcing many details, through providing a limited (but relatively clear) set of options, to letting someone else decide.

I've always wanted the TC to provide enabling but not overly limiting guidance that actively acknowledges concerns.

Location of Interop Tests

There are three reviews related to the location of the Interop Tests (aka Trademark Tests):

- A detailed one, based on PTG discussion (https://review.openstack.org/#/c/521602/)
- A middle of the road one, simplifying the first (https://review.openstack.org/#/c/550571/)
- A (too) simple one (https://review.openstack.org/#/c/550863/)

It's looking like the middle one has the most support now, but that is after a lot of discussion. On Wednesday (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-07.log.html#t2018-03-07T19:07:37) I introduced the middle of the road version to make sure the previous discussion was represented in a relatively clear way. Then on Thursday (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-08.log.html#t2018-03-08T15:20:31), a version which effectively moves responsibility to the InteropWG.

Throughout this process there have been hidden goals (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-08.log.html#t2018-03-08T08:41:13) whereby this minor(?) crisis in policy is being used to attempt to address shortcomings in the bigger picture. It's great to be working on the bigger picture, but hidden doesn't sound like the right approach.

Tracking TC Goals

One of the outcomes from the PTG was an awareness that some granular and/or middle-distance TC goals tend to get lost. The TC is going to try to use StoryBoard (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-08.log.html#t2018-03-08T15:06:21) to track these sort of things. The hope is that this will result in more active and visible progress.

Extended Maintenance

A proposal to leave branches open for patches (https://review.openstack.org/#/c/548916/) for longer has received at least as much attention as the Interop discussions.

Some talk starts on Thursday afternoon (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-08.log.html#t2018-03-08T16:17:03) and then carries on intermittently for the rest of time^w^w^w^wthrough today. The review has a great deal of interaction as well.

There's general agreement on the principal ("let's not limit people from being able to patch branches for longer") but reaching consensus on the details has been more challenging. Different people have different goals.

What's a SIG for?

Discussion (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-13.log.html#t2018-03-13T09:16:57) about renaming (https://review.openstack.org/#/c/551413/) the recently named PowerStackers group (https://review.openstack.org/#/c/540165/) eventually migrated into talking about what SIGs are for or mean (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-13.log.html#t2018-03-13T09:31:56). There seems to be a few different interpretations, with some overlap:

- Upstream and downstream concern (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-13.log.html#t2018-03-13T09:41:17) or "breadth of potential paricipants".
- Not focused on producing code (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-13.log.html#t2018-03-13T09:56:04).
- Different people in the same "room" (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-13.log.html#t2018-03-13T09:43:40).
- In problem space rather than solution space (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-13.log.html#t2018-03-13T13:34:10).

None of these are really complete. I think of SIGs as a way to break down boundaries, provide forums for discussion and make progress without worrying too much about bureaucracy. We probably don't need to define them to death.

by Chris Dent at March 13, 2018 03:10 PM (https://anticdent.org/tc-report-18-11.html)

Cloudwatt (https://dev.cloudwatt.com/en/blog/index.html)

Upgrade Newton on Heat : what’s new for orchestration service ? (https://dev.cloudwatt.com/en/blog/heat-newton-upgrade.html)

Dear Users,

Orchestration service is now in Newton version: let’s deep dive into new features and Heat resources improvements.

New Features

Convergence feature allows to improve performance of your create, update, delete actions. Most effective when stacks are large. Enabled by default when service upgraded, you will take benefit from it without any action from your side.

Observe and Update feature is also enabled, it checks resources state before doing any update. For example, taking a stack with a volume and an instance, you aim to update, adding a volume attachment to this instance. If instance or/and volume do not exist anymore, the update without *Observe and Update* feature will fail. Reversely, with it, the update ends with success.

Heat resources improvements

Volume resources

If you plan to update one of your stack, we recommend you strongly to explicit the metadata property filling “{}” in its template.

New resources available

OS::Neutron::SecurityGroupRule

A resource for managing Neutron security group rules. Rules to use in security group resource.

Example :

```
type: OS::Neutron::SecurityGroupRule
properties:
  description: String
  direction: String
  ethertype: String
  port_range_max: Integer
  port_range_min: Integer
  protocol: String
  remote_group: String
  remote_ip_prefix: String
  security_group: String
```

OS::Heat::Value

A resource which exposes its value property as an attribute.

This is useful for exposing a value that is a simple manipulation of other template parameters and/or other resources.

Example :

```
type: OS::Heat::Value
properties:
  type: String
  value: Any
```

OS::Heat::ResourceChain

This resource allows to create multiple resources with same properties.

Example :

```
resource:
  type: OS::Heat::ResourceChain
  properties:
      concurrent: True
      resource_properties: {size: 5, volume_type: "performant"}
      resources: ["OS::Cinder::Volume","OS::Cinder::Volume", "OS::Cinder::Volume"]
```

Here above, 3 performant volumes of 5G have been created.

Property “concurrent” defines if resource creation can be done in parallel.

Property “resource_properties” defines properties to apply to resources

Property “resources” defines resources to create.

This resource allows mainly to factorize yaml templates.

OS::Heat::None

It allows to disable easily some resources through “resource_registry” on environment file.

This resource works with all properties and return any attribute.

Example :

```
resource_registry:
  OS::Cinder::Volume: OS::Heat::None
```

OS::Heat::TestResource

This a resource which store the value from the chain.

Some control actions are available, such as ‘action_wait_secs’, ‘wait_secs’, ‘attr_wait_secs’. This allow to adjust time for stack creation or for waiting period to obtain resource output.

Example ::

```
alarm_cache_wait:
    type: OS::Heat::TestResource
    properties:
        action_wait_secs:
            create: 60
            update: 60
        value: "test"
outputs:
    res_value:
        value: {get_attr: [alarm_cache_wait, output]}
```

OS::Heat::SoftwareDeploymentGroup

This resource associates a group of servers with some configuration.

Replaces the resource OS::Heat::SoftwareDeployments

Example :

```
type: OS::Heat::SoftwareDeploymentGroup
properties:
  actions: [Value, Value, ...]
  config: String
  input_values: {...}
  name: String
  servers: {...}
  signal_transport: String
```

OS::Heat::StructuredDeploymentGroup

This resource associates a group of servers with some configuration.

This resource works similar as OS::Heat::SoftwareDeploymentGroup, but for structured resources.

Replaces the resource OS::Heat::StructuredDeployments

Example :

```
type: OS::Heat::StructuredDeploymentGroup
properties:
  actions: [Value, Value, ...]
  config: String
  input_key: String
  input_values: {...}
  input_values_validate: String
  name: String
  servers: {...}
  signal_transport: String
```

New attributes

For resource OS::Heat::AutoScalingGroup

- refs : A list of resource IDs for the resources in the group

- refs_map : A map of resource names to IDs for the resources in the group

For resource OS::Heat::ResourceGroup

- refs_map : A map of resource names to IDs for the resources in the group
- removed_rsrc_list : A list of removed resource names

New properties

For resource OS::Nova::Server

For block_device_mapping_v2:

- image : The ID or name of the image to create a volume from. Updated by image property.

For OS::Glance::Image

- architecture
- os_distro (In order to make images more easily searchable in different openstack installations)
- kernel_id (The ID of image stored in Glance that should be used as the kernel when booting an AMI-style image,)
- ramdisk_id (The ID of image stored in Glance that should be used as the ramdisk when booting an AMI-style image)
- extra_properties
- tags
- owner

New update policies

For resource OS::Heat::SoftwareDeploymentGroup and OS::Heat::StructuredDeploymentGroup (resources added in mitaka)

- Batch_create
  - Map value expected :
  - max_batch_size : The maximum number of resources to create at once
  - pause_time
- Rolling_update
  - Map value expected :
  - max_batch_size : The maximum number of deployments to replace at once
  - Pause_time

Example :

```
type: OS::Heat::SoftwareDeploymentGroup
  update_policy:
    rolling_update:
      max_batch_size: 2
      pause_time: 1
  properties:
    config: {get_resource: config}
    input_values:
      foo: {get_param: input}
    servers:
     '0': dummy0
```

Deprecated resources

For resource OS::Nova::Server and For block_device_mapping_v2

The `image_id` has to be replaced by `image`

End of support for the following resources

OS::Neutron::PortPair

You can contact our support service *support@cloudwatt.com*

by Horia Merchi at March 13, 2018 12:00 AM (https://dev.cloudwatt.com/en/blog/heat-newton-upgrade.html)

March 12, 2018

OpenStack Superuser (http://superuser.openstack.org)

Getting started with multi-cloud (http://superuser.openstack.org/articles/multi-cloud-101/)

There are many reasons companies consider going multi-cloud: to lift and shift their data center into the cloud, extend their data center into the cloud or to move applications to a different environment. Or maybe it's because a single cloud service provider can't meet all of their complex compute and storage needs.

Start with the right questions

Before you start multiplying your clouds though, it makes sense to take a step back.

"Far too many times IT managers and directors assume they can cram their entire infrastructure into one platform and that might not be the best case scenario for every project," says Morgan Martensen, business development at IT hosting company SingleHop (https://www.singlehop.com/)on the company's tech chat, SingleHoppy Hou (https://www.youtube.com/channel/UCTFdB_C9fYj_OL2ZDPIJ4TQ)r. "Our number one rule is: application first, infrastructure second." That means asking questions about your applications before starting to plan what will power them.

Too many people don't consider what's driving the move. "It's like walking into a car dealership and asking: 'Do you sell cars?' You have to have to understand what that IT director is trying to accomplish because there are a lot of delivery models," says TJ Waldorf, VP of inside sales and marketing at SingleHop.

Asking questions is key, some to start with include:

- What workloads do they run

- What apps are they running and supporting
- What are the compliance requirements
- What are the geographical requirements

Keep an eye on the bills

Cost is often a key driver for going multi-cloud, but it may not be the simplest route. Instead of hashing out cost optimization on one platform, you've got five. "It's definitely challenging," says Waldorf. "If you're using multiple vendors you've got multiple bills. AWS in particular…if you've ever seen an invoice from AWS it's probably 47 pages with 10,000 line items that you're not entirely sure what they are and when costs go up go or down it's really hard to pinpoint why." There are plenty of tools to help with cost optimization, but you still have to have the expertise to know how to run those tools and understand how to do it right, he adds. Waldorf cites the case of one customer who moved from their own dedicated servers to the cloud looking for greater scalability, but realized that the unexpected bandwidth costs were much higher than anticipated. In a lot of cases, it makes sense to leave a few applications and workloads on dedicated servers or a private cloud. "It boils down to figuring out where those things need to go and optimizing for performance but also cost. There's a lot of trade-offs with public cloud in any cloud platform," he says.

Catch the whole 19-minute conversation on YouTube (https://www.youtube.com/watch?v=L6G7DwK_O2E)

The post Getting started with multi-cloud (http://superuser.openstack.org/articles/multi-cloud-101/) appeared first on Superuser (http://superuser.openstack.org).

by Superuser at March 12, 2018 03:36 PM (http://superuser.openstack.org/articles/multi-cloud-101/)

OpenStack in Production (http://openstack-in-production.blogspot.com/)

Hardware burn-in in the CERN datacenter (http://openstack-in-production.blogspot.com/2018/03/hardware-burn-in-in-cern-datacenter.html)

During the Ironic sessions at the recent OpenStack Dublin PTG in Spring 2018, there were some discussions on adding a further burn in step to the OpenStack Bare Metal project (Ironic) state machine (https://docs.openstack.org/ironic/pike/contributor/states.html). The notes summarising the sessions were reported to the openstack-dev (https://www.mail-archive.com/openstack-dev@lists.openstack.org/msg116584.html) list. This blog covers the CERN burn in process for the systems delivered to the data centers as one example of how OpenStack Ironic users could benefit from a set of open source tools to burn in newly delivered servers as a stage within the Ironic workflow.

CERN hardware procurement follows a formal process compliant (http://procurement.web.cern.ch/en/procurement-strategy-and-policy) with public procurements. Following a market survey to identify potential companies in CERN's member states, a tender specification is sent to the companies asking for offers based on technical requirements.

Server burn in goals

Following the public procurement processes at CERN, large hardware deliveries occur once or twice a year and smaller deliveries multiple times per year. The overall resource management at CERN was covered in a previous blog (http://openstack-in-production.blogspot.fr/2016/04/resource-management-at-cern.html). Part of the steps before production involves burn in of new servers. The goals are

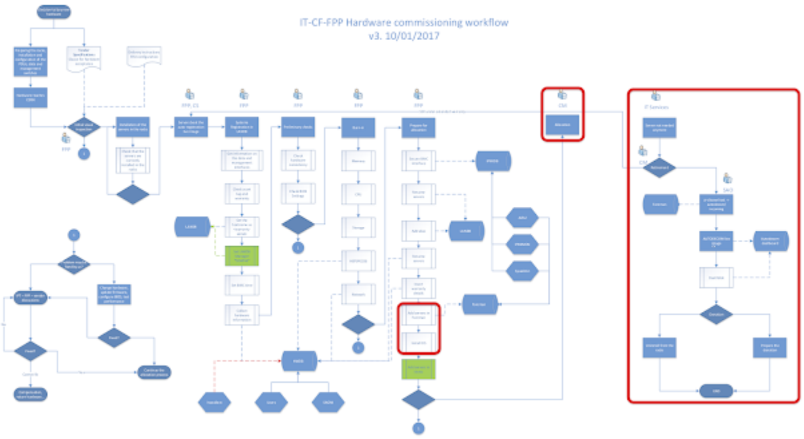
- Ensure that the hardware delivered complies with CERN Technical Specifications
- Find systematic issues with all machines in a delivery such as bad firmware
- Identify failed components in single machines
- Provoke early failure in failing components due to high load during stress testing

Depending on the hardware configuration, the burn-in tests take on average around two weeks but do vary significantly (e.g. for systems with large memory amounts, the memory tests alone can take up to two weeks). This has been found to be a reasonable balance between achieving the goals above compared to delaying the production use of the machines with further testing which may not find more errors.

Successful execution of the CERN burn in processes is required in the tender documents prior to completion of the invoicing.

Workflow

The CERN hardware follows a lifecycle from procurement to retirement as outlined below. The parts marked in red are the ones currently being implemented as part of the CERN Bare Metal deployment.



(https://1.bp.blogspot.com/-GgOIP-L4p0c/WqZhrDAQa4I/AAAAAAAAUuY/2wXS9uSK42k36lGgpFRVNddWAvn96jwYwCLcBGAs/s1600/FPP_Workflow_for_IRONIC_v3crop.png)

As part of the evaluation, test systems are requested from the vendor and these are used to validate compliance with the specifications. The results are also retained to ensure that the bulk equipment deliveries correspond to the initial test system configurations and performance.

Preliminary Checks

CERN requires that the Purchase Order ID and an unique System Serial Number are set in the NVRAM of the Baseboard Management Controller (BMC), in the Field Replaceable Unit (FRU) fields Product Asset Tag (PAT) and Product Serial (PS) respectively:

```
# ipmitool fru print 0 | tail -2
 Product Serial        : 245410-1
 Product Asset Tag     : CD5792984
```

The Product Asset Tag is set to the CERN delivery number and the Product Serial is set to the unique serial number for the system unit.

Likewise, certain BIOS fields have to be set correctly such as booting from network before disk to ensure the systems can be easily commissioned.

Once these basic checks have been done, the burn in process can start. A configuration file, containing the burn-in tests to be run, is created according on the information stored in the PAT and PS FRU fields. Based on the content of the configuration file, the enabled tests will automatically start.

Burn in

The burn in process itself is highlighted in red in the workflow above, consisting of the following steps

- Memory
- CPU
- Storage
- Benchmarking
- Network

Memory

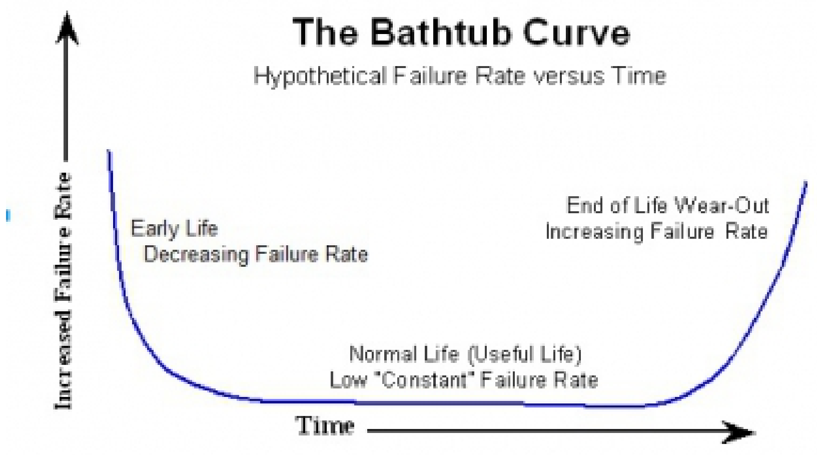
The memtest stress tester is used for validation of the RAM in the system. Details of the tool are available at http://www.memtest.org/ (http://www.memtest.org/).

CPU

Testing the CPU is performed using a set of burn tools, burnK7 or burnP6, and burn MMX. These tools not only test the CPU itself but are also useful to find cooling issues such as broken fans since the power load is significant with the processors running these tests.

Disk

Disk burn ins are intended to create the conditions for early drive failure. The bathtub curve (https://www.datarecoup.com/blog/data-recovery/the-bathtub-curve) aims to cause the early failure drives to fail prior to production.



(https://2.bp.blogspot.com/-1PoikJv0mVQ/WqVRcyYkGnI/AAAAAAAAUt0/Q_1VallsZFUyrK_uLAndrJ45xzZMOpJ9wCLcBGAs/s1600/b2ap3_thumbnail_bathtub-curve.jpg)

With this aim, we use the badblocks (https://en.wikipedia.org/wiki/Badblocks) code to repeatedly read/write the disks. SMART counters are then checked to see if there are significant numbers of relocated bad blocks and the CERN tenders require disk replacement if the error rate is high.

We still use this process although the primary disk storage for the operating system has now changed to SSD. There may be a case for minimising the writing on an SSD to maximise the life cycle of the units.

Benchmarking

Many of the CERN hardware procurements are based on price for total compute capacity needed. With the nature of most of the physics processing, the total throughput of the compute farm is more important than the individual processor performance. Thus, it may be that the most total performance can be achieved by choosing processors which are slightly slower but less expensive.

CERN currently measures the CPU performance using a set of benchmarks based on a subset of the SPEC 2006 suite. The subset, called HEPSpec06, is run in parallel on each of the cores in the server to determine the total throughput from the system. Details are available at the HEPiX Benchmarking Working Group (http://w3.hepix.org/benchmarking.html) web site.

Since the offers include the expected benchmark performance, the results of the benchmarking process are used to validate the technical questionnaire submitted by the vendors. All machines in the same delivery would be expected to produce similar results so variations between different machines in the same batch are investigated.

CPU benchmarking can also be used to find problems where there is significant difference across a batch, such as incorrect BIOS settings on a particular system.

Disk performance is checked using a reference fio (http://fio.readthedocs.io/en/latest/fio_doc.html) access suite. A minimum performance level in I/O is also required in the tender documents.

Networking

Networking interfaces are difficult to burn in compared to disks or CPU. To do a reasonable validation, at lest two machines are needed. With batches of 100s of servers, a simple test against a single end point will produce unpredictable results.

Using a network broadcast, the test finds other machines running the stress test, they pair up and run a number of tests.

- iperf3 (https://iperf.fr/) is used for bandwidth, reversed bandwidth, udp and reversed udp
- iperf for full duplex testing (currently missing from iperf3)
- ping is used for congestion testing

Looking forward

CERN is currently deploying Ironic into production for bare metal management of machines. Integrating the burn in and retirement stages into the bare metal management states would bring easy visibility of the current state as the deliveries are processed.

The retirement stage is also of interest to ensure that there is no CERN configuration in the servers (such as Ironic BMC credentials or IP addresses). CERN has often donated retired servers to other high energy physics sites such as SESAME in Jordan (https://home.cern/cern-people/updates/2017/10/servers-sesame) and Morocco (https://www.sharing-knowledge.org/the-cern-computer-centre-has-donated-161-retired-servers-to-universities-in-morocco/) which requires a full server factory reset before dismounting. This retirement step would be a more extreme cleaning followed by complete removal from the cloud.

Discussing with other scientific laboratories such as SKA (https://www.skatelescope.org/) through the OpenStack Scientific special interest group (https://wiki.openstack.org/wiki/Scientific_SIG) has shown interest in extending Ironic to automate the server on-boarding and retirement processes as described in the session at the OpenStack Sydney summit (https://www.openstack.org/videos/sydney-2017/future-science-on-future-openstack-developing-next-generation-infrastructure-at-cern-and-ska). We'll be following up on these discussions at Vancouver.

Acknowledgements

- CERN IT department - http://cern.ch/it (http://cern.ch/it)
- CERN Ironic and Rework Contributors
  - Alexandru Grigore
  - Daniel Abad
  - Mateusz Kowalski

References

- Computing in High Energy Physics procurement paper (CHEP 2013) - http://iopscience.iop.org/article/10.1088/1742-6596/513/6/062003/pdf (http://iopscience.iop.org/article/10.1088/1742-6596/513/6/062003/pdf)
- Automatic server registration and burn-in framework (HEPiX Fall 2013) - https://indico.cern.ch/event/247864/contributions/1570349/attachments/426708/592287/Automatic_Registration_Burn_in.ppt (https://indico.cern.ch/event/247864/contributions/1570349/attachments/426708/592287/Automatic_Registration_Burn_in.ppt)
- Backblaze failure report - https://www.backblaze.com/blog/hard-drive-stats-for-2017/ (https://www.backblaze.com/blog/hard-drive-stats-for-2017/)
- Resource Management at CERN - http://openstack-in-production.blogspot.fr/2016/04/resource-management-at-cern.html (http://openstack-in-production.blogspot.fr/2016/04/resource-management-at-cern.html)

by Tim Bell (noreply@blogger.com) at March 12, 2018 11:29 AM (http://openstack-in-production.blogspot.com/2018/03/hardware-burn-in-in-cern-datacenter.html)

Mirantis (https://www.mirantis.com)

OpenStack embraces the future with GPU, edge computing support (https://www.mirantis.com/blog/openstack-embraces-the-future-with-gpu-edge-computing-support/)

It wasn't that long ago that OpenStack was the hot new kid on the infrastructure block. That just might change with the latest release of OpenStack, code-named Queens.

by Nick Chase at March 12, 2018 07:38 AM (https://www.mirantis.com/blog/openstack-embraces-the-future-with-gpu-edge-computing-support/)

March 09, 2018

Adam Spiers (https://blog.adamspiers.org)

Report from the OpenStack PTG in Dublin (https://blog.adamspiers.org/2018/03/09/openstack-ptg-dublin/)

Last week I attended OpenStack's PTG (Project Teams Gathering) (https://www.openstack.org/ptg) in Dublin. This event happens every 6 months in a different city, and is a fantastic opportunity for OpenStack developers and upstream contributors to get together and turbo-charge the next phase of collaboration.

I wrote a private report for my SUSE colleagues summarising my experience, but then Colleen posted her report publicly (http://www.gazlene.net/dublin-ptg.html), which made me realise that it would be far more in keeping with OpenStack's Four Opens (https://governance.openstack.org/tc/reference/opens.html) to publish mine online. So here it is!

General observations

The PTG was well-attended with just over 400 people from all over the world. As with previous PTGs this felt like just the right size: enough to have most projects / organizations well represented, but not too big to prevent serendipitous meetings happening many times throughout the week, where I would just be wandering around and bump into exactly the right person I needed to talk to ☺

All the etherpads are publicly available (https://wiki.openstack.org/wiki/PTG/Rocky/Etherpads) in case you want to dive into the gory details.

IMHO the ptgbot (https://git.openstack.org/cgit/openstack/ptgbot/tree/README.rst) continued to work really well for adapting to the dynamic and fast-changing agenda. I submitted a few patches to polish it further.

The "extreme" (for Ireland, anyway) weather, which was reportedly the heaviest snow in 40 years, caused a lot of disruption, not just for travel, but also that our conference venue in Croke Park Stadium was forced to shut on Thursday afternoon, rather than Friday evening as originally planned. Luckily our amazing hotel, The Croke Park (https://www.doylecollection.com/hotels/the-croke-park-hotel), stepped up to the plate and allowed us to continue the conference there. It was quite a squash fitting everyone in, but somehow they managed it, and the ensuing organized mayhem had the unexpected side effect of building really strong comradery between everyone.

There was a piano-bar style sing-along night (https://twitter.com/e_monty/status/968658891507732481) (with OpenStack's Executive Director Jonathan Bryce (https://twitter.com/jbryce) doing a great job on the piano), a board games night (https://twitter.com/knelson92/status/969515964176969734) … I think it was my colleague Keith Berger who shrewdly observed that it was like being on a cruise ship ☺ Having the best Guinness in the world on tap probably helped too. Check out #SnowpenStack on Twitter (https://twitter.com/hashtag/SnowpenStack) for some of the hilarious antics.

We managed a couple of SUSE team dinners and it was great to spend more time with teammates, especially the ones who are normally on a different continent ☺

Self-healing SIG

- https://etherpad.openstack.org/p/self-healing-ptg-rocky (https://etherpad.openstack.org/p/self-healing-ptg-rocky)

My main reason for attending the PTG was bootstrapping this new SIG which I founded a few months ago (https://blog.adamspiers.org/announcing-openstacks-self-healing-sig/). We had a really good session filling Tuesday afternoon, with about 30 people attending.

(https://blog.adamspiers.org/wp-content/uploads/2018/03/self-healing-PTG-Dublin.jpeg)

There were discussions about various use cases, and some people are already keen to start work on use cases immediately. We also had representatives from the API SIG (https://wiki.openstack.org/wiki/API_SIG), including Graham Hayes a.k.a. `mugsie` (http://graham.hayes.ie/) who is leading the new health-check API initiative (https://review.openstack.org/#/c/531456/), and from OPNFV, since we are trying to promote better sharing of work on automated testing of self-healing / HA functionality (http://lists.openstack.org/pipermail/openstack-dev/2018-March/127956.html). (If you're interested, please sign up here (https://etherpad.openstack.org/p/extreme-testing-contacts)!)

We agreed some concrete next steps, some of which I already started working on:

- Populate our self-healing-sig repo with the standard Sphinx stuff and templates for use cases and specs (https://storyboard.openstack.org/#!/story/2001628)
- Compile a list of contact info for self-healing stakeholders across the various projects / organizations (https://storyboard.openstack.org/#!/story/2001639)
- Start to visualize the architectural integration points between involved projects.
- Start to document use cases and write specs for proposed work.

You can follow all activity via the StoryBoard project (https://storyboard.openstack.org/#!/project/917).

The only disappointment with this session was that no operators attended. However I managed to coordinate with some people involved with the Tokyo Ops meetup this week (https://etherpad.openstack.org/p/TYO-ops-meetup-2018), in order to gather feedback from ops, and we got a session slot booked in that agenda which happened yesterday. Additionally I am speaking on self-healing and leading a discussion session on it at the London OpenStack Meetup next Monday (https://www.meetup.com/Openstack-London/events/247958746/), so hopefully we'll get some good feedback there too.

First Contact SIG

- https://etherpad.openstack.org/p/FC_SIG_Rocky_PTG (https://etherpad.openstack.org/p/FC_SIG_Rocky_PTG)

I joined the contributor guide part of this session briefly to give some feedback and also organise collaboration on writing a quickstart guide for people who are used to GitHub and don't know Gerrit, since I promised to write that after creating a comparison matrix (https://ethercalc.openstack.org/github-gerrit) in Denver, which provides a base for the guide.

StoryBoard

The Self-healing SIG previously agreed to use StoryBoard for tracking work (http://lists.openstack.org/pipermail/openstack-sigs/2017-November/000180.html), not just to aid collaboration within the SIG, but also to help dogfood StoryBoard and provide feedback. So I spent quite a bit of time with the StoryBoard folks giving feedback and brainstorming ideas for improving usability. I also started work on a "Rosetta stone" matrix which aims to show the key differences between StoryBoard and other issue trackers people are used to, such as JIRA, Trello, GitHub, Bugzilla, Redmine etc. Similarly to the approach with the GitHub / Gerrit matrix above, it could serve as the basis for a document which helps "port" people to StoryBoard from other issue trackers.

Fast-forward upgrades (FFU)

- https://etherpad.openstack.org/p/ffu-ptg-rocky (https://etherpad.openstack.org/p/ffu-ptg-rocky)

This was a really full session, with some good discussions. Main points:

- It was agreed to form an Upgrade SIG, and this has now happened (https://governance.openstack.org/sigs/).
- AFAICS everyone is not yet on the same page about what FFU means exactly, especially regarding online migrations. There is perhaps room for improvement in the governance tags (https://governance.openstack.org/tc/reference/tags/index.html) provided for tracking upgradability (assert:supports-upgrade (https://governance.openstack.org/tc/reference/tags/assert_supports-upgrade.html), assert:supports-accessible-upgrade (https://governance.openstack.org/tc/reference/tags/assert_supports-accessible-upgrade.html)), which would help clarify this.
- IIUC Kolla (https://wiki.openstack.org/wiki/Kolla) seem to be going in a slightly different direction based on online rolling upgrades, although I don't think they necessarily have any strong objections to supporting offline upgrades so maybe things will converge in the long term.
- Noone from `openstack-helm` (https://wiki.openstack.org/wiki/Openstack-helm) present ☹
- Lots of gotchas, and consequent debates on how best to capture and share these.

Oslo

- https://etherpad.openstack.org/p/oslo-ptg-rocky (https://etherpad.openstack.org/p/oslo-ptg-rocky)

I only had time to pop into the `oslo` room briefly, to discuss an interesting new initiative in `oslo.config` to automate migration of config between releases (a bit like DB migrations, but for `oslo.config` instead of DB schema). This was also mentioned in the FFU session.

- gerrit#I851628c0 (Propose spec for handle config changes in code) (https://review.openstack.org/#/c/520043/)
- gerrit#I0a14a4a4 (Handle config mapping changes) (https://review.openstack.org/#/c/526314/)
- gerrit#I0b95d3ea ({WIP}{POC} Handle config mapping changes) (https://review.openstack.org/#/c/526261/)

I posed some questions about how this would handle config which requires extra per-project logic to migrate, and also how it would handle config which has been **explicitly** set to the default value. We had some interesting discussions on that but I think ultimately the answer was "let's cross that bridge when we come to it".

I also had a quick chat with Ken Giusti about a long-standing `oslo.service` bug we've worked on (https://bugs.launchpad.net/barbican/+bug/1705543), but we ran out of time and agreed to take it online.

OpenStack Ansible

I demo'd ansible-viz (https://github.com/aspiers/ansible-viz), the Ansible visualization tool of which I recently became the maintainer, to the OpenStack Ansible (https://docs.openstack.org/openstack-ansible/latest/) folks. They seemed really enthusiastic, and are looking into using it for embedding diagrams in their docs. Hopefully this means they will start contributing to make it better!

My colleague Dirk Müller and I also talked to them in general about other areas for potential collaboration. We were both impressed with how open they are to help and collaborate!

Misc

As usual with the PTGs, I got huge value from random unplanned conversations in hallways, at lunch tables, in the bar etc.

I spent a lot of time talking to Emmet Hickory (a.k.a. `persia`) and he gave me some great ideas about new use cases for the `git` workflow tools I have been building in hackweeks (https://hackweek.suse.com/) and spare time. I already knew that my `git-deps` (https://github.com/aspiers/git-deps) tool allows fairly accurate predictions of how difficult it will be to backport a given commit to a stable branch, but a new idea emerged that this could be added as a non-voting job to Zuul (https://docs.openstack.org/infra/zuul/)'s checks pipeline, to facilitate a more proactive / preemptive approach to backporting.

I talked with Tony Breeds (PTL of stable branch maintenance) about this who told me that Neutron is already extremely proactive when it comes to backporting, so they might be interested in this. So I met Miguel Lavalle (Neutron PTL) and he was really interested and requested that I put it on the agenda for a future Neutron IRC meeting. This has now happened (https://wiki.openstack.org/wiki/Network/Meetings#On_Demand_Agenda) and we will discuss during next Tuesday's meeting.

Emmet helped me realise that my `git-explode` tool could also help with backporting work, so I need to finish packaging that and publish it. There is a London Git meetup (https://www.meetup.com/londongit/) coming up soon, and I might present the tools at that.

I'm getting off-topic now so I'll finish here ☺ If you found this report useful please leave a comment below, so that I know it's worth bothering to do next time ☺

Share / Save (https://www.addtoany.com/share#url=https%3A%2F%2Fblog.adamspiers.org%2F2018%2F03%2F09%2Fopenstack-ptg-dublin%2F&title=Report%20from%20the%20OpenStack%20PTG%20in%20Dublin)

The post Report from the OpenStack PTG in Dublin (https://blog.adamspiers.org/2018/03/09/openstack-ptg-dublin/) appeared first on Structured Procrastination (https://blog.adamspiers.org).

by Adam at March 09, 2018 08:27 PM (https://blog.adamspiers.org/2018/03/09/openstack-ptg-dublin/)

OpenStack Superuser (http://superuser.openstack.org)

How to develop a new Horizon dashboard without DevStack (http://superuser.openstack.org/articles/horizon-dashboard-devstack-openstack/)

Everybody knows that the easiest way to build a new Horizon dashboard is to use DevStack. But, as you also already know, DevStack (https://docs.openstack.org/devstack/latest/) requires a fair amount of resources (16GB of RAM for a workable DevStack, in my experience).

So what can you do if you only have a laptop with 4-8GB RAM? Easy peasy, just follow these steps:

1. Install Keystone (the minimum requirement for Horizon)

Follow these official instructions: https://docs.openstack.org/keystone/latest/install/keystone-install-ubuntu.html (https://docs.openstack.org/keystone/latest/install/keystone-install-ubuntu.html)

Here are some common issues you may run into and fixes for them:

http://www.dangtrinh.com/2018/03/fix-error-attributeerror-module-object.html (http://www.dangtrinh.com/2018/03/fix-error-attributeerror-module-object.html)
http://www.dangtrinh.com/2018/03/fix-importerror-cannot-import-name.html (http://www.dangtrinh.com/2018/03/fix-importerror-cannot-import-name.html)
http://www.dangtrinh.com/2018/03/how-to-fix-index-column-size-too-large.html (http://www.dangtrinh.com/2018/03/how-to-fix-index-column-size-too-large.html)

***2.* Install Horizon**

Follow these official instructions on how to install Horizon on your computer: https://docs.openstack.org/horizon/latest/contributor/quickstart.html#quickstart (https://docs.openstack.org/horizon/latest/contributor/quickstart.html#quickstart)

3. Build your own dashboard

Using this tutorial: https://docs.openstack.org/horizon/latest/contributor/tutorials/dashboard.html (https://docs.openstack.org/horizon/latest/contributor/tutorials/dashboard.html)

Note: you may have to install additional services if your dashboard needs to communicate with those services (e.g: Tacker, Nova…)

4. Change the world

Trinh Nguyen (https://twitter.com/dangtrinhnt) is a hacker and ed-tech enthusiast with a startup called ***EdLab.XYZ*** *(http://www.edlab.xyz). This post (http://www.dangtrinh.com/) first appeared on his blog (http://www.dangtrinh.com/2018/03/how-to-dev-new-horizon-dashboard.html?spref=tw).*

Superuser is always interested in community content, especially tutorials. Get in touch at editoATopenstack.org

Cover Photo (http://superuser.openstack.org/feed/"https://www.flickr.com/photos/andertoons-cartoons/4448997876//") // CC BY NC (https://creativecommons.org/licenses/by-nc/2.0/)

The post How to develop a new Horizon dashboard without DevStack (http://superuser.openstack.org/articles/horizon-dashboard-devstack-openstack/) appeared first on Superuser (http://superuser.openstack.org).

by Trinh Nguyen at March 09, 2018 05:11 PM (http://superuser.openstack.org/articles/horizon-dashboard-devstack-openstack/)

OpenStack in Production (http://openstack-in-production.blogspot.com/)

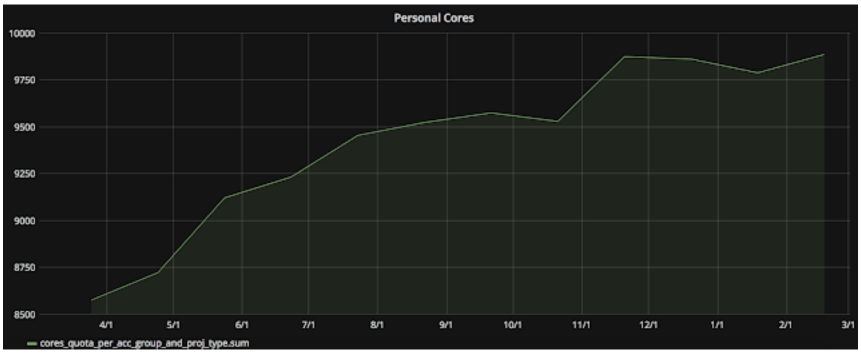
Expiry of VMs in the CERN cloud (http://openstack-in-production.blogspot.com/2018/03/expiry-of-vms-in-cern-cloud.html)

The CERN cloud resources are used for a variety of purposes from running compute intensive workloads to long running services. The cloud also provides personal projects for each user who is registered to the service. This allows a small quota (5 VMs, 10 cores) where the user can have resources dedicated for their use such as boxes for testing. A typical case would be for the CERN IT Tools training where personal projects are used as sandboxes for trying out tools such as Puppet.

Personal projects have a number of differences compares to other projects in the cloud

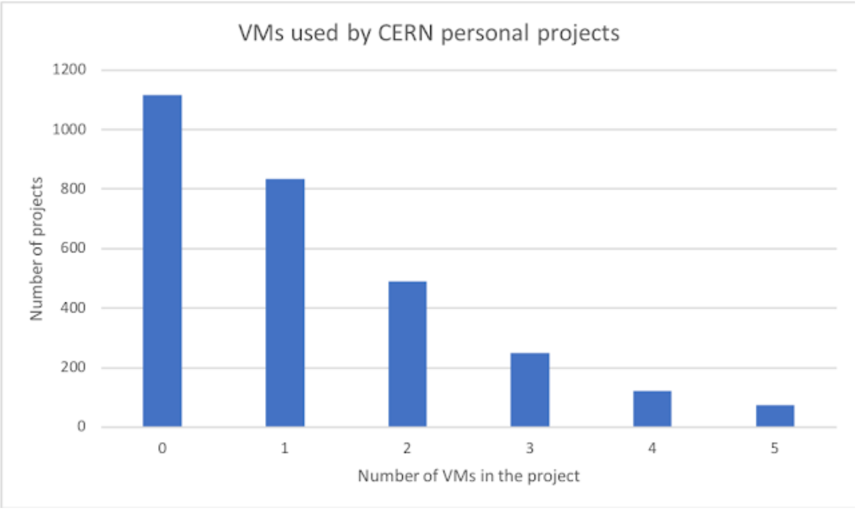
- No non-standard flavors
- No additional quota can be requested
- Should not be used for production services
- VMs are deleted automatically when the person stops being a CERN user

With the number of cloud users increasing to over 3,000, there is a corresponding growth in the number of cores used by personal projects, growing by 1,200 cores in the past year. For cases like training users, there is often the case that the VMs are created and the user then does not remember to delete the resources so they consume cores which could be used for compute capacity to analyse the data from the LHC.



(https://2.bp.blogspot.com/-UNngAR17x2U/WpgR5oaR9xI/AAAAAAAAUpU/CxiBN9fqdq4FoNrse2viyA0lRJ7T_nS2QCLcBGAs/s1600/personalcores.png)

One possible approach would be to reduce the quota further. However, tests such as setting up a Kubernetes cluster with OpenStack Magnum often need several VMs to perform the different roles so this would limit the usefulness of personal projects. The usage of the full quota is also rare.



(https://3.bp.blogspot.com/-AqfFcpBGqSc/WpkCxszwnBI/AAAAAAAAUpk/Foc7StUo-9cSaX1DyXMPuEXFsTnXSBnpQCLcBGAs/s1600/vmsused.png)

VM Expiration

Based on a previous service which offered resources on demand (called CVI based on Microsoft SCVMM), the approach was taken to expire personal virtual machines.

- Users can create virtual machines up to the limit of their quota
- Personal VMs are marked with an expiry date
- Prior to their expiry, the user is sent several mails to inform them their VM will expire soon and how to extend it if it is still useful.
- On expiry, the virtual machine is locked and shutdown. This helps to catch cases where people have forgotten to prolong their VMs.
- One week later, the virtual machine is deleted, freeing up the resources.

Implementation

We use Mistral (https://wiki.openstack.org/wiki/Mistral)to automate several OpenStack tasks in the cloud (such as regular snapshots (http://openstack-in-production.blogspot.fr/2017/08/scheduled-snapshots.html) and project creation/deletion). This has the benefit of a clean audit log to show what steps worked/failed along with clear input/output states supporting retries and an authenticated cloud cron for scheduling.

Our OpenStack projects have some properties set when they are created. This is used to indicate additional information like the accounting codes to be charged for the usage. There are properties for indicating if the type of project such as personal and if the expiration workflow should apply. Mistral YAQL code (https://gitlab.cern.ch/cloud-infrastructure/mistral-workflows/blob/master/workbooks/instance_expiration_global.yaml) can then select resources where expiration applies.

```
task(retrieve_all_projects).result.select(dict(id => $.id, name => $.name, enabled => $.enabled, type => $.get('type','none'),expire =>
$.get('expire','off'))).where($.type='personal').where($.enabled).where($.expire='on')
```

The expire_at parameter is stored as a VM property. This makes it visible for automation such as CLIs through the openstack client show server CLI.

There are several parts to the process

- A cron trigger'd workflow which
  - Machines in error state or currently building are ignored
  - A newly created machine which does not have an expiry date set has the expiration date set according to the grace period
  - Sees if any machines are entering close to their expiry time and sends a mail to the owner
  - Checks for invalid settings of the expire_at property (such as people setting it a long way in the future or deleting the property) and restores a reasonable value if this is detected
  - If a machine has reached it's expiry date, it's locked and shutdown
  - If a machine has past it's date by the grace period, it's deleted
- A workflow, launched by Horizon or from the CLI
  - Retrieves the expire_at value and extends it by the prolongation period

The user notification is done using a set of mail templates and a dedicated workflow (https://gitlab.cern.ch/cloud-infrastructure/mistral-workflows/blob/master/workflows/send_mail_template.yaml (https://gitlab.cern.ch/cloud-infrastructure/mistral-workflows/blob/master/workflows/send_mail_template.yaml)). This allows templates such as instance reminders (https://gitlab.cern.ch/cloud-infrastructure/mistral-workflows/blob/master/templates/instance_expiration_reminder_mail.yaml) to have details about the resources included, such as the example from the mail template.

The Virtual Machine {instance} from the project {project_name} in the Cloud Infrastructure Service will expire on {expire_date}.

A couple of changes to Mistral will be submitted upstream

- Support for HTML mail bodies which allows us to have a nicer looking e-mail for notification with links included
- Support for BCC/CC on the mail so that the OpenStack cloud administrator e-mail can also be kept on copy when there are notifications

A few minor changes to Horizon were also done (currently local patches)

- Display expire_at value on the instance details page
- Add a 'prolong' action so that instances can be prolonged via the web by using the properties editor to set the date of the expiry (defaulting to the current date with the expiry time). This launches the workflow for prolonging the instance.

Author

Jose Castro Leon from the CERN cloud team

References

- CERN IT department at http://information-technology.web.cern.ch/ (http://information-technology.web.cern.ch/)
- Workflow code is in the repository at https://gitlab.cern.ch/cloud-infrastructure/mistral-workflows (https://gitlab.cern.ch/cloud-infrastructure/mistral-workflows), specifically at https://gitlab.cern.ch/cloud-infrastructure/mistral-workflows/tree/master/workbooks (https://gitlab.cern.ch/cloud-infrastructure/mistral-workflows/tree/master/workbooks) for the YAQL.
- CERN end user documentation is at http://clouddocs.web.cern.ch/clouddocs/projects/vm_expiration.html (http://clouddocs.web.cern.ch/clouddocs/projects/vm_expiration.html)
- OpenStack Mistral Workflow as a Service documentation is at https://docs.openstack.org/mistral/latest/overview.html (https://docs.openstack.org/mistral/latest/overview.html)

by Jose Castro Leon (noreply@blogger.com) at March 09, 2018 03:11 PM (http://openstack-in-production.blogspot.com/2018/03/expiry-of-vms-in-cern-cloud.html)

Julie Pichon (https://www.jpichon.net/)

OpenStack PTG Dublin - Rocky (https://www.jpichon.net/blog/2018/03/openstack-ptg-dublin-rocky/)

I was so excited when it was first hinted in Denver that the next OpenStack PTG would be in Dublin. In my town! Zero jet lag! Commuting from home! Showing people around! Alas, it was not to be. Thanks, Beast from the East (https://en.wikipedia.org/wiki/2018_Great_Britain_and_Ireland_cold_wave). Now everybody hates Ireland forever.

The weather definitely had some impact on sessions and productivity. People were jokingly then worryingly checking on the news, dropping in and out of rooms as they tried to rebook their cancelled flights. Still we did what we could and had snow-related activities too - good for building team spirit, if nothing else!

I mostly dropped in and out of rooms, here are some of my notes and sometimes highlights.

OpenStack Client

Like before, the first two days of the PTG were focused on cross-projects concerns. The OpenStack Client didn't have a room this time, which seems fair as it was sparsely attended the last couple of times - I would have thought there'd have been one helproom session at least but if there was I missed it.

I regret missing the API Working Group (https://etherpad.openstack.org/p/api-sig-ptg-rocky) morning sessions on API discovery and micro-versions, which I think were relevant. The afternoon API sessions were more focused on services and less applicable for me. I need to be smarter about it next time.

First Contact SIG

Instead, that morning I attended the First Contact Special Interest Group (https://etherpad.openstack.org/p/FC_SIG_Rocky_PTG) sessions, which aim to make OpenStack more accessible to newcomers. It was well attended, with even a few new and want-to-be contributors who were first-time PTG attendes - I think having the PTG in Europe really helped with that. The session focused on making sure everyone in the room/SIG is aware of the resources that are out there, to be able to help people looking to get started.

The SIG is also looking for points of contact for every project (https://wiki.openstack.org/wiki/First_Contact_SIG#Project_Liaisons), so that newcomers have someone to ask questions to directly (even better if there's a backup person too, but difficult enough to find one as it is!).

Some of the questions that came up from people in the room related to being able to map projects to IRC channel (e.g. devstack questions go to #openstack-qa).

Also, the OpenStack community has a ton of mentoring programs, both formal and informal and just going through the list to explain them took a while. Outreachy, Google Summer of Code, Upstream Institute, Women of OpenStack, First Contact Liaisons (see above). Didn't realise there were so many!

I remember when a lot of the initiatives discussed were started. It was interesting to hear the perspectives from people who arrived later, especially the discussions around the ones that have become irrelevant.

Packaging RPMs

On Tuesday I dropped by the packaging RPMs Working Group (https://etherpad.openstack.org/p/rpm-packaging-rocky-ptg) session. A small group made up of very focused RDO/Red Hat/SUSE people. The discussions were intense, with Python 2 going End Of Life in under 2 years now.

The current consensus seems to be to create a RPM-based Python 3 gate based on 3.6. There's no supported distro that offers this at the moment, so we will create our own Fedora-based distro with only what we need at the versions we need it. Once RDO is ready with this, it could be moved upstream.

There were some concerns about 3.5 vs 3.6 as the current gating is done on 3.5. Debian also appears to prefer 3.6. In general it was agreed there should not be major differences and generally ok.

The clients must still support Python 2.

There was a little bit of discussion about the stable policy and how it doesn't apply to the specs or the rpm-packaging project (I think the example was with Monasca and the default backend not working (?), so a spec change to modify the backend was backported - which could be considered a feature backport, but since the project isn't under the stable policy remit it could be done).

There was a brief chat at the end about whether there is still interest in packaging services, as opposed to just shipping them as containers. There certainly still seems to be at this point.

Release management

A much more complete summary (http://lists.openstack.org/pipermail/openstack-dev/2018-March/128005.html) has already been posted on the list, and I had to leave the session halfway to attend something else.

There seems to be an agreement that it *is* getting easier to upgrade (although some people still don't want to do it, perhaps an education effort is needed to help with this). People do use the stable point release tags.

The "pressure to upgrade": would Long-Term Support release actually help? Probably it would make it worse. The pressure to upgrade will still be there except there won't be a need to work on it for another year, and it'll make life worse for operators/etc submitting back fixes because it'll take over a year for a patch to make it into their system.

Fast-Forward Upgrade (which is **not** skip-level upgrades) may help with that pressure... Or not, maybe different problems will come up because of things like not restarting services in between upgrades. It batches things and helps to upgrade faster, but changes nothing.

The conversation moved to one year release cycles just before I left. It seemed to be all concerns and I don't recall active support for the idea. Some of the concerns:

- Concerns about backports - so many changes
- Concerns about marketing - it's already hard to keep up with all that's going on, and it's good to show the community is active and stuff is happening more than once a year. It's not that closely tied to releases though, announces could still go out more regularly.
- Planning when something will land may become even harder as so much can happen in a year
- It's painful for both people who keep up *and* people who don't, because there is so much new stuff happening at once.

TripleO

The sessions began with a retrospective (https://etherpad.openstack.org/p/tripleo-ptg-rocky-retro) on Wednesday. I was really excited to hear that tripleo-common was going to get unit tests for workflows. I still love the idea of workflows but I found them becoming more and more difficult to work with as they get larger, and difficult to review. Boilerplate gets copy-pasted, can't work without a few changes that are easy to miss unless manually tested and these get missed in reviews all the time.

The next session was about CI (https://etherpad.openstack.org/p/tripleo-ptg-ci). The focus during Queens was on reliability, which worked well although promotions suffered as a result. There were some questions as to whether we should try to prevent people from merging anything when the promotion pipeline is broken but no consensus was really reached.

The Workflows session (https://etherpad.openstack.org/p/tripleo-ptg-rocky-workflows-api) was really interesting, there's been a lot of Lessons Learnt from our initial attempt with Mistral this last couple of years and it looks like we're setting up for a v2 overhaul that'll get rid of many of the issues we found. Exciting! There was a brief moment of talk about ripping Mistral out and reimplementing everything in Ansible, conclusions unclear.

I didn't take good notes during the other sessions and once the venue closed down (snow!) it became a bit difficult to find people in the hotel and then actually hear them. Most etherpads with the notes are linked from the main TripleO etherpad (https://etherpad.openstack.org/p/tripleo-ptg-rocky).

by jpichon at March 09, 2018 11:57 AM (https://www.jpichon.net/blog/2018/03/openstack-ptg-dublin-rocky/)

March 08, 2018

OpenStack Superuser (http://superuser.openstack.org)

When CI/CD becomes a battleground (http://superuser.openstack.org/articles/ci-cd-ops-battle/)

You can almost hear the roar vibrating from the cubicles. It's the epic tussle between tiger teams — those elite IT professionals — as they decide the future of operations. The decisions about how applications are deployed and who gets to decide that process are part of an ongoing fight spurred by the move to continuous integration and development.

Kong Yang spoke about the human side that often gets overlooked when it comes to adopting CI/CD and dev-ops on a recent edition (https://soundcloud.com/user-410091210) of L8ist Sh9y podcast (https://soundcloud.com/user-410091210) (that's "Latest Shiny," for those who appreciate vowels.)

Yang, a Dell alum whose current title is actually "head geek" at SolarWinds (https://www.solarwinds.com/), spoke of a scenario where two teams of engineers battle over the CI/CD pipeline, with each team given six months to see whose is better. This kind of showdown rarely works long term: typically, he says, both will walk away and leave it to somebody else to do the maintenance of a system that's designed to fail fast. And, in the meantime, one of those battle-scarred distinguished engineers probably has to go looking for greener pastures. Or the company loses both — because they can always walk away. Everyone loses.

"That's the piece that leaders on the business side don't get… What they want to see is more revenue, they want more profits. They want to see more stuff that they can offer to customers," Yang says. "That human aspect is one thing that a lot of organizations don't factor in because guess what: You have the option to go out and hire new talent. Well, the talent that you have in-house has the option to leave as well."

Host of the podcast Rob Hirschfeld (http://twitter.com/zehicle), CEO of RackN (http://www.rackn.com/company), agreed, adding that he's dubbed it the SRE half life, where people are brought into an organization to usher in change. When the company resists those changes, the site reliability engineer or architect leaves for a more conducive habitat. "Companies have to figure out how to accept and embrace these changes, it's a very big deal."

The pair covered a lot of other interesting ground, including:

- Why we're in the golden age of cloud 2:34 (https://soundcloud.com/user-410091210/kong-yang-on-golden-age-of-cloud-cicd-and-devops-and-operator-opportunity#t=02:34)
- The disrupting force of mobile apps and the CI/CD pipeline 15:38 (https://soundcloud.com/user-410091210/kong-yang-on-golden-age-of-cloud-cicd-and-devops-and-operator-opportunity#t=15:38)
- "Living data points" vs. monolith apps 17:15 (https://soundcloud.com/user-410091210/kong-yang-on-golden-age-of-cloud-cicd-and-devops-and-operator-opportunity#t=17:15)
- The challenge (and the opportunities) for IT ops professionals now 23:20 (https://soundcloud.com/user-410091210/kong-yang-on-golden-age-of-cloud-cicd-and-devops-and-operator-opportunity#t=23:20)

- More on the political debate around building CI/CD pipelines 27:11 (https://soundcloud.com/user-410091210/kong-yang-on-golden-age-of-cloud-cicd-and-devops-and-operator-opportunity#t=27:11)
- From AWS to bare metal 30:18 (https://soundcloud.com/user-410091210/kong-yang-on-golden-age-of-cloud-cicd-and-devops-and-operator-opportunity#t=30:18)

You can catch the whole 37-minute podcast (https://soundcloud.com/user-410091210) on SoundCloud.

Cover Photo (http://superuser.openstack.org/feed/"https://www.flickr.com/photos/theaustinot/9258040426//") // CC BY NC (https://creativecommons.org/licenses/by-nc/2.0/)

The post When CI/CD becomes a battleground (http://superuser.openstack.org/articles/ci-cd-ops-battle/) appeared first on Superuser (http://superuser.openstack.org).

by Superuser at March 08, 2018 05:29 PM (http://superuser.openstack.org/articles/ci-cd-ops-battle/)

Mark T. Voelker (http://markvoelker.github.io/blog/)

Dublin PTG 2018: The Edge Sessions (http://markvoelker.github.io/blog/dublin-ptg-edge-sessions/)

I recently got snowed in at a hotel in Dublin with a couple hundred people I consider friends, a well-stocked bar/restaurant, a fantastic event staff, and a lot of energy. I know…poor me, right? =) Before Storm Emma and the Beast from the East settled in on the OpenStack Project Team Gathering and forced the Croke Park Stadium to close (thereby giving birth to the hashtag “snowpenstack”), we got a solid start to the week.

March 08, 2018 12:00 AM (http://markvoelker.github.io/blog/dublin-ptg-edge-sessions/)

March 07, 2018

Major Hayden (https://major.io)

Reaching the fork in the road (https://major.io/2018/03/07/reaching-the-fork-in-the-road/)

(https://major.io/wp-content/uploads/2018/03/1024px-Mountain_trail_in_Panachaiko_mountains_Greece.jpg)

Walt Disney said it best:

> We keep moving forward, opening new doors, and doing new things, because we’re curious and curiosity keeps leading us down new paths.

The world of technology is all about change. We tear down the old things that get in our way and we build new technology that takes us to new heights. Tearing down these old things can often be difficult and that forces us to make difficult choices.

Rackspace has been a great home for me for over 11 years. I’ve made the incredibly difficult choice to leave Rackspace on March 9th to pursue new challenges.

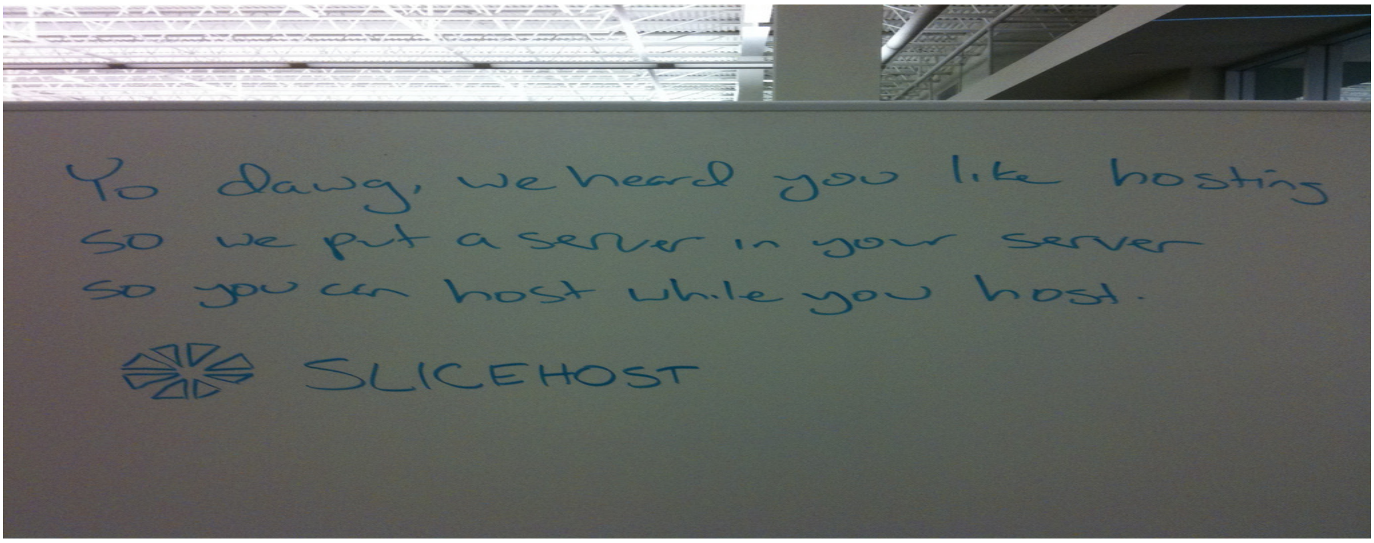
Humble beginnings

I came to Rackspace as an entry-level Linux administrator and was amazed by the culture generated by Rackers. The dedication to customers, technology, and quality was palpable from the first few minutes I spent with my team. Although I didn’t know it at the time, I had landed at the epicenter of a sink-or-swim technology learning experience. My team had some very demanding customers with complex infrastructures and it forced me to take plenty of notes (and hard knocks). My manager and teammates supported me through it all.

From there, I served in several different roles. I was a manager of technicians on a support team and had the opportunity to learn how to mentor. One of my favorite leaders said that “good managers know when to put their arm around to people and when to put a boot in their rear.” I reluctantly learned how to do both and I watched my people grow into senior engineers and great leaders.

(https://major.io/wp-content/uploads/2018/03/6519121761_ab65bab3c1_b.jpg)
Datapoint office closing in 2011

I was pulled to Mosso, Rackspace's first cloud offering, shortly after that and discovered an entirely new world. Rackers force-fed me "Why's (Poignant) Guide to Ruby" and I started building scripts and web front-ends for various services. Rackspace acquired Slicehost after that and I jumped at the chance to work as an operations engineer on the new infrastructure. That led to a lot of late nights diagnosing problems with Xen hypervisors and rails applications. I met some amazing people and began to realize that St. Louis has some pretty good barbecue (but Texas still has them beat).


Yo dawg, we heard you like hosting
so we put a server in your server
so you can host while you host.
SLICEHOST

(https://major.io/wp-content/uploads/2018/03/4171091103_7150ded95f_b.jpg)
Slicehost humor in 2009

Not long after that, I found myself managing an operations team that cared for Slicehost's infrastructure and Rackspace's growing Cloud Servers infrastructure. OpenStack appeared later and I jumped at the chance to do operations there. It was an extremely rough experience in the Diablo release, but it taught me a lot. My start with OpenStack involved fixing lots of broken Keystone tests that didn't run on Python 2.6.

(https://major.io/wp-content/uploads/2018/03/7730840100_01257c5fa4_b.jpg)
Working on OpenStack in 2012

If you've attended some of my talks on impostor syndrome, you may know what came next. We had a security issue and I sent some direct feedback to our CSO about how it was handled. I expected to be told to "pack a box" after that, but I was actually asked to lead a security architecture team in the corporate security group. It was definitely a surprise. I accepted and joined the team as Chief Security Architect. My coworkers called it "joining the dark side", but I did my best to build bridges between security teams and the rest of the company.

(https://major.io/wp-content/uploads/2018/03/24142777780_5196ca622b_h.jpg)
Talking at Rackspace::Solve in 2015

This role really challenged me. I had never operated at the Director level before and our team had a ton of work to do. I found myself stumbling (and floundering) fairly often and I leaned on other leaders in the business for advice. This led me to take some courses on critical thinking, accounting, finance, and tough conversations. I've never had a role as difficult as this one.

Our cloud team came calling and asked me to come back and help with some critical projects in the public cloud. We worked on some awesome skunkworks projects that could really change the business. Although they didn't get deployed in one piece, we found ways to take chunks of the work and optimize different areas of our work. An opportunity came up to bring public cloud experience to the private cloud team and I jumped on that one. I discovered the awesome OpenStack-Ansible project and a strong set of Rackers who were dedicated to bringing high-touch service to customers who wanted OpenStack in their own datacenter.

(https://major.io/wp-content/uploads/2018/03/imposter-syndrome_hayden.jpg)
Impostor syndrome talk at the Boston OpenStack Summit in 2017

During this time, I had the opportunity to deliver several conference talks about OpenStack, Fedora, security, and Ansible. My favorite topic was impostor syndrome and I set out on a mission to help people understand it. My first big talk was at the Fedora Flock conference in Rochester in 2015. This led to deep conversations with technical people in conference hallways, evening events, and even airport terminals about how impostor syndrome affects them. I took those conversations and refined my message several times over.

(https://major.io/wp-content/uploads/2018/03/DSCF0425.jpg)
Talking about impostor syndrome at Fedora Flock 2015 (Photo credit: Kushal Das)

Gratitude

I couldn't even begin to name a list of Rackers who have helped me along the way. I wouldn't be where I am now without the help of hundreds of Rackers. They've taught me how to build technology, how to navigate a business, and how to be a better human. They have made me who I am today and I'm eternally grateful. I've had an incredible amount of hugs this week at the office and I've tried my best not to get a face full of tears in the process.

I'd also like to thank all of the people who have allowed me to mentor them and teach them something along the way. One of the best ways to understand something is to teach it to someone else. I relish any opportunity to help someone avoid a mistake I made, or at least be able to throw something soft under them to catch their fall. These people put up with my thick Texas accent, my erratic whiteboard diagrams, and worse of all, my dad jokes.

Another big "thank you" goes out to all of the members of the open source communities who have mentored me and dealt with my patches.

The first big community I joined was the Fedora Linux community. I've been fortunate to serve on the board and participate in different working groups. Everyone has been helpful and accommodating, even when I pushed broken package builds. I plan to keep working in the community as long as they will have me!

The OpenStack community has been like family. Everyone — from developers to foundation leaders — has truly been a treat to work with over several years. My work on Rackspace's public and private clouds has pushed me into various projects within the OpenStack ecosystem and I've found everyone to be responsive. OpenStack events are truly inspiring and it is incredible to see so many people from so many places who dedicate themselves to the software and the people that make cloud infrastructure work.

The next adventure

I plan to talk more on this later, but I will be working from home on some projects that are entirely different from what I'm working on now. That adventure starts on March 19th after a week of "funemployment." I'm incredibly excited about the new opportunity and I'll share more details when I can.

Top photo credit: Wikipedia (https://commons.wikimedia.org/wiki/File:Mountain_trail_in_Panachaiko_mountains,_Greece.jpg)

The post Reaching the fork in the road (https://major.io/2018/03/07/reaching-the-fork-in-the-road/) appeared first on major.io (https://major.io).

by Major Hayden at March 07, 2018 04:18 PM (https://major.io/2018/03/07/reaching-the-fork-in-the-road/)

OpenStack Superuser (http://superuser.openstack.org)

Takeaways from the first Helm Summit (http://superuser.openstack.org/articles/takeaways-helm-summit-2018/)

Snow and freezing temperatures in Portland didn't stop the inaugural Helm Summit from a well-attended two-day conference devoted to the Kubernetes packaging system.

The first day was devoted to user stories, the use cases and techniques who have adopted Helm (https://helm.sh/) to manage packages for their Kubernetes infrastructure. Following on experiences from the first day, the second day got down to unconference work sessions devoted to shaping the upcoming design and development of Helm 3. About 200 people braved the weather to attend.

Adam Reese, one of the original Helm developers, kicked off the summit with a keynote (https://www.youtube.com/watch?v=S4NYhymOQNQ) that covered the history and philosophy of Helm. "Let's solve problems instead of create more software" was the main takeaway that also doubles as kind of a mission statement for the project.

The rest of the day included user stories from Ubisoft on how they use Helm to deliver core services (https://www.youtube.com/watch?v=M2MmqN_nznI) to their games, Josh Dolitsky of Codefresh on launching the open-source project ChartMuseum to be a Helm chart repository (https://www.youtube.com/watch?v=f1GpCrxmbB0) and Greg Taylor about how Helm backs (https://www.youtube.com/watch?v=7Qxuo9W5SlY) the development-staging-production application life cycle at Reddit. Security and scalability were common themes of the talks.

Day two was more focused on getting developers and users together to scope out requirements for Helm 3.
Some highlights from the second day keynote (https://www.youtube.com/watch?v=MWkrU2dtpqs):

- Maintaining backwards compatibility with the wealth of existing Helm 2 charts
- Cleaning up the code base by removing deprecated code
- Refactoring code to make it cleaner, more readable, and less redundant
- Introducing more role based access controls (RBAC) and trying to design with "sensible security" in mind
- Making a commitment to excellent documentation to benefit the new and seasoned user alike

From there, attendees gave a series of lightning talks, many focused on proposing new features for Helm 3 based on current needs and use cases. Building a strong community was also a topic, with a panel of Kubernetes Community leaders emphasizing that kindness was one of the most powerful tools (https://youtu.be/7u90isT92Og?t=505) a project can use to grow both talented individual developers and a strong community.

As the day moved into afternoon work sessions, one theme that became clear was that within Helm there will always be a tension between the slower pace of Helm packaging and the faster pace of Kubernetes development. Backwards compatibility between not only Helm releases but for Kubernetes releases is important to K8s users whose organizations may be slower to adopt more recent releases of Kubernetes. The strength of open source projects that deliver applications depends on the strength of package managers, and Helm has already demonstrated itself as the essential Kubernetes package manager. It was exciting to participate in this event and I'm looking forward to seeing what changes are in store for Helm 3.

For a deeper write-up of the event from one of the Helm core maintainers, head over to Taylor Thomas's post (https://open.microsoft.com/2018/03/06/first-helm-summit-notes-recordings/) at the Microsoft + Open Source blog.

You can also catch all the talks on the Helm YouTube Channel. (https://www.youtube.com/channel/UC_kvCKc5EHNomq64f8C4sfA)

Get involved

In May, I'll be attending the first Kubernetes-SIG-Docs Summit in Portland, representing SIG-OpenStack (https://github.com/kubernetes/community/tree/master/sig-openstack)and helping out with provider documentation. If you're at the Write the Docs conference (http://www.writethedocs.org/conf/) and want to get involved with Kubernetes documentation, be sure to stop by! Details on the Kubernetes #sig-docs channel or here (https://kubernetes.io/docs/home/contribute/participating/).

Chris Hoge (https://twitter.com/hogepodge) is the OpenStack Foundation's senior strategic program manager.

Cover Photo (http://superuser.openstack.org/feed/"https://https://www.flickr.com/photos/henriblock/5878495038///") // CC BY NC (https://creativecommons.org/licenses/by-nc/2.0/)

The post Takeaways from the first Helm Summit (http://superuser.openstack.org/articles/takeaways-helm-summit-2018/) appeared first on Superuser (http://superuser.openstack.org).

by Chris Hoge at March 07, 2018 03:54 PM (http://superuser.openstack.org/articles/takeaways-helm-summit-2018/)

Chris Dent (https://anticdent.org/)

TC Report 18-10 (https://anticdent.org/tc-report-18-10.html)

This is a TC Report, but since everything that happened in its window of observation is preparing for the PTG (https://www.openstack.org/ptg), being at the PTG, trying to get home from the PTG, and recovering from the PTG, perhaps think of this as "What the TC talked about [at] the PTG". As it is impossible to be everywhere at once (especially when the board meeting overlaps with other responsibilities) this will miss a lot of important stuff. I hope there are other summaries.

As you may be aware, it snowed in Dublin (https://twitter.com/search?q=%23snowpenstack) causing plenty of disruption to the PTG (https://twitter.com/search?q=%23openstackptg) but everyone (foundation staff, venue staff, hotel staff, attendees, uisce beatha) worked together to make a good week.

Talking about the PTG at the PTG

At the board meeting (http://lists.openstack.org/pipermail/foundation/2018-March/002570.html), the future of the PTG was a big topic. As currently constituted it presents some challenges:

- It is difficult for some people to attend because of visa and other travel related issues.
- It is expensive to run and not everyone is convinced of the return on investment.
- Some people don't like it (they either miss the old way of doing the design summit, or midcycles, or $OTHER).
- Plenty of other reasons that I'm probably not aware of.

This same topic was reviewed at yesterday's office hours (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-06.log.html#t2018-03-06T09:19:32).

For now, the next 2018 PTG is going to happen (destination unknown) but plans for 2019 are still being discussed.

If you have opinions about the PTG, there will be an opportunity to express them in a forthcoming survey. Beyond that, however, it is important that management at contributing companies (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-06.log.html#t2018-03-06T22:29:24) hear from more people (notably their employees) than the foundation about the value of the PTG.

My own position is that of the three different styles of in-person events for technical contributors to OpenStack that I've experienced (design summit, mid-cycles, PTG), the PTG is the best yet. It minimizes distractions from other obligations (customer meetings, presentations, marketing requirements) while maximizing cross-project interaction.

One idea, discussed yesterday (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-06.log.html#t2018-03-06T22:02:24) and earlier today (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-07.log.html#t2018-03-07T05:07:20) was to have the PTG be open to technical participants of any sort, not just so-called "OpenStack developers". Make it more of a place for people who hack on and with OpenStack to hack and talk. Leave the summit (without a forum) for presentations, marketing, pre-sales, etc.

An issue raised with conflating the PTG and the Forum is that it would remove the inward/outward focus (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-07.log.html#t2018-03-07T08:20:17) concept that is supposed to distinguish the two events.

I guess it depends on how we define "we" but I've always assumed that both events were for outward focus and that for any inward focussing effort we ought to be able use asynchronous tools more.

Foundation and OCI

Thierry mentioned yesterday (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-06.log.html#t2018-03-06T09:08:04) that it is likely that the OpenStack Foundation will join the Open Container Initiative (https://www.opencontainers.org/) because of Kata (https://katacontainers.io/) and LOCI (https://governance.openstack.org/tc/reference/projects/loci.html).

This segued into some brief concerns about the attentions and intentions of the Foundation (http://eavesdrop.openstack.org/irclogs/%23openstack-tc/%23openstack-tc.2018-03-06.log.html#t2018-03-06T09:13:34), aggravated by the board meeting schedule conflict (there's agreement that will never ever happen again), and the rumor milling about the PTG.

Friday at the PTG with the TC

The TC had scheduled a half day of discussion for Friday at the PTG. A big agenda (https://etherpad.openstack.org/p/PTG-Dublin-TC-topics), a fun filled week, and the snow meant we went nearly all day (and since there's no place to go, let's talk, let's talk, let's talk) with some reasonable progress. Some highlights:

- There was some discussion on trying to move forward with constellations concept, but I don't recall specific outcomes from that discussion.
- The team diversity tags need to be updated to reflect adjustments in the very high bars we set earlier in the history of OpenStack. We agreed to not remove projects from the tc-approved tag, as that could be taken the wrong way. Instead we'll create a new tag for projects that are in the trademark program.
- Rather than having Long Term Support, which implies too much, a better thing to do is enable extended maintenance (https://review.openstack.org/#/c/548916/) for those parties who want to do it.
- Heat was approved to be a part of the trademark program, but then there were issues with where to put their tests and the tooling used to manage them. By the power of getting the right people in the room at the same time, we reached some consensus which is being finalized on a proposed resolution (https://review.openstack.org/#/c/521602/).
- We need to make an official timeline for the deprecation (and eventual removal) of support for Python 2, meaning we also need to accelerate the adoption of Python 3 as the primary environment.
- In a discussion about the availability of etcd (https://coreos.com/etcd/) it was decided that tooz needs to be finished (https://docs.openstack.org/tooz/latest/user/compatibility.html).

See the etherpad (https://etherpad.openstack.org/p/PTG-Dublin-TC-topics) for additional details.

by Chris Dent at March 07, 2018 12:10 PM (https://anticdent.org/tc-report-18-10.html)

March 06, 2018

Adam Young (http://adam.younglogic.com)

Generating a Callgraph for Keystone (http://adam.younglogic.com/2018/03/callgraph-keystone/)

Once I know a starting point for a call, I want to track the other functions that it calls. pycallgraph will generate an image that shows me that.

All this is done inside the virtual env set up by tox at keystone/.tox/py35

I need a stub of a script file in order to run it. I'll put this in tmp:

```
from keystone.identity import controllers
from keystone.server import wsgi
from keystone.common import request

def main():

    d = dict()
    r  = request.Request(d)

    wsgi.initialize_admin_application()
    c = controllers.UserV3()
    c.create_user(r)

if __name__ == '__main__':
    main()
```

To install pycallgraph:

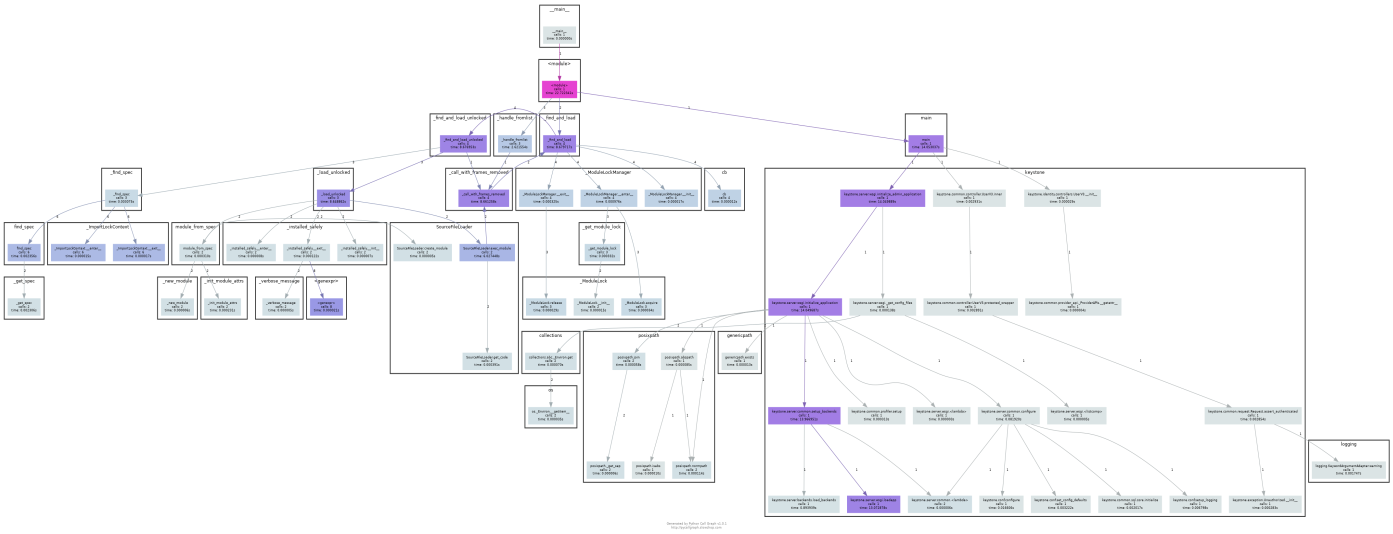
```
pip install pycallgraph
```

And to run it:

```
 pycallgraph  --max-depth 6  graphviz /tmp/call_create_user.py
```

It errors out do to auth issues (it is actually rtunning the code, so don't do this on a production server)

Here is what it generated.


__main__
__main__ calls: 1 time: 0.000000s
<module>
<module> calls: 1 time: 22.722341s
_find_and_load_unlocked
_find_and_load_unlocked calls: 4 time: 8.676953s
_handle_fromlist
_handle_fromlist calls: 3 time: 2.621554s
_find_and_load
_find_and_load calls: 4 time: 8.679717s
main
main calls: 1 time: 14.053037s
_find_spec
_find_spec calls: 3 time: 0.003075s
_load_unlocked
_load_unlocked calls: 3 time: 8.648862s
_call_with_frames_removed
_call_with_frames_removed calls: 4 time: 8.661258s
_ModuleLockManager
_ModuleLockManager.__exit__ calls: 4 time: 0.000325s
_ModuleLockManager.__enter__ calls: 4 time: 0.000976s
_ModuleLockManager.__init__ calls: 4 time: 0.000017s
cb
cb calls: 4 time: 0.000012s
find_spec
find_spec calls: 6 time: 0.002356s
_ImportLockContext
_ImportLockContext.__enter__ calls: 6 time: 0.000015s
_ImportLockContext.__exit__ calls: 6 time: 0.000017s
module_from_spec
module_from_spec calls: 2 time: 0.000310s
_installed_safely
_installed_safely.__enter__ calls: 2 time: 0.000008s
_installed_safely.__exit__ calls: 2 time: 0.000122s
_installed_safely.__init__ calls: 2 time: 0.000007s
SourceFileLoader
SourceFileLoader.create_module calls: 2 time: 0.000005s
SourceFileLoader.exec_module calls: 2 time: 6.027448s
SourceFileLoader.get_code calls: 2 time: 0.000391s
_get_module_lock
_get_module_lock calls: 3 time: 0.000332s
_get_spec
_get_spec calls: 2 time: 0.002306s
_new_module
_new_module calls: 2 time: 0.000006s
_init_module_attrs
_init_module_attrs calls: 2 time: 0.000231s
_verbose_message
_verbose_message calls: 2 time: 0.000005s
<genexpr>
<genexpr> calls: 8 time: 0.000021s
_ModuleLock
_ModuleLock.release calls: 3 time: 0.000029s
_ModuleLock.__init__ calls: 2 time: 0.000015s
_ModuleLock.acquire calls: 3 time: 0.000034s
collections
collections.abc._Environ.get calls: 2 time: 0.000070s
os
os._Environ.__getitem__ calls: 2 time: 0.000035s
posixpath
posixpath.join calls: 2 time: 0.000058s
posixpath.abspath calls: 1 time: 0.000085s
posixpath._get_sep
posixpath.isabs
posixpath.normpath
genericpath
genericpath.exists calls: 1 time: 0.000013s
keystone
keystone.server.wsgi.initialize_admin_application calls: 1 time: 14.049889s
keystone.common.controller.UserV3.inner calls: 1 time: 0.002931s
keystone.identity.controllers.UserV3.__init__ calls: 1 time: 0.000029s
keystone.server.wsgi.initialize_application calls: 1 time: 14.049687s
keystone.server.wsgi._get_config_files calls: 1 time: 0.000138s
keystone.common.controller.UserV3.protected_wrapper calls: 1 time: 0.002891s
keystone.common.provider_api._ProviderAPIs.__getattr__ calls: 1 time: 0.000004s
keystone.server.common.setup_backends calls: 1 time: 13.966951s
keystone.common.profiler.setup calls: 1 time: 0.000313s
keystone.server.wsgi.<lambda> calls: 1 time: 0.000003s
keystone.server.common.configure calls: 1 time: 0.081920s
keystone.server.wsgi.<listcomp> calls: 1 time: 0.000005s
keystone.common.request.Request.assert_authenticated calls: 1 time: 0.002854s
keystone.server.backends.load_backends calls: 1 time: 0.893939s
keystone.server.wsgi.loadapp calls: 1 time: 13.072878s
keystone.server.common.<lambda> calls: 2 time: 0.000006s
keystone.conf.configure calls: 1 time: 0.016606s
keystone.conf.set_config_defaults calls: 1 time: 0.003222s
keystone.common.sql.core.initialize calls: 1 time: 0.002017s
keystone.conf.setup_logging calls: 1 time: 0.006798s
keystone.exception.Unauthorized.__init__ calls: 1 time: 0.000283s
logging
logging.KeywordArgumentAdapter.warning calls: 1 time: 0.001747s

(https://admiyo.fedorapeople.org/pycallgraph.png)

Click to enlarge. Not great, but it is a start.

by Adam Young at March 06, 2018 09:53 PM (http://adam.younglogic.com/2018/03/callgraph-keystone/)

Inspecting Keystone Routes (http://adam.younglogic.com/2018/03/inspecting-keystone-routes/)

What Policy is enforced when you call a Keystone API? Right now, there is no definitive way to say. However, with some programmatic help, we might be able to figure it out from the source code. Lets start by getting a complete list of the Keystone routes.

In the WSGI framework that Keystone uses, a Route is the object that used to match the URL. For example, when I try to look at the user with UserId abcd1234, I submit a GET request to the URL https://hostname:port/v3/users/abcd1234. The route path is the pattern /users/{user_id}. The WSGI framework handles the parts of the URL prior to that, and eventually needs to pull out a Python function to execute for the route. Here is how we can generate a list of the route paths in Keystone

```
from keystone.server import wsgi
app = wsgi.initialize_admin_application()
composing = app['/v3'].application.application.application.application.application.application._app.application.application.application.applicati
for route in composing._router.mapper.matchlist:
    print(route.routepath)
```

I'll put the output at the end of this post.

That long chain of **.application** properties is due to the way that the pipeline is built using the paste file. In **keystone/etc/keystone-paste.ini** we see:

```
[pipeline:api_v3]
# The last item in this pipeline must be service_v3 or an equivalent
# application. It cannot be a filter.
pipeline = healthcheck cors sizelimit http_proxy_to_wsgi osprofiler url_normalize request_id build_auth_context token_auth json_body ec2_extensio
```

Each of those pipeline elements are python classes specified earlier in the file, that honor the middleware contract. Most of these can be traced back to the keystone.common.wsgi.Middleware base class, which implements this as __call__ method.

```
    @webob.dec.wsgify(RequestClass=request_mod.Request)
    @middleware_exceptions
    def __call__(self, request):
        response = self.process_request(request)
        if response:
            return response
        response = request.get_response(self.application)
        return self.process_response(request, response)
```

The odd middleware out is AuthContextMiddleware which extends from from keystonemiddleware.auth_token.BaseAuthProtocol. See if you can spot the difference:

```
    @webob.dec.wsgify(RequestClass=_request._AuthTokenRequest)
    def __call__(self, req):
        """Handle incoming request."""
        response = self.process_request(req)
        if response:c
            return response
        response = req.get_response(self._app)
        return self.process_response(response
```

Yep: **self._app**.

Here is the output from the above code, executed in the python interpreter. This does not have the Verbs in it yet, but a little more poking should show where they are stored:

```
>>> for route in composing._router.mapper.matchlist:
...     print(route.routepath)
...
/auth/tokens
/auth/tokens
/auth/tokens
/auth/tokens
/auth/tokens/OS-PKI/revoked
/auth/catalog
/auth/projects
/auth/domains
/auth/system
/users/{user_id}/projects
/roles/{prior_role_id}/implies
/roles/{prior_role_id}/implies/{implied_role_id}
/roles/{prior_role_id}/implies/{implied_role_id}
/roles/{prior_role_id}/implies/{implied_role_id}
/roles/{prior_role_id}/implies/{implied_role_id}
/role_inferences
/projects/{project_id}/users/{user_id}/roles/{role_id}
/projects/{project_id}/users/{user_id}/roles/{role_id}
/projects/{project_id}/users/{user_id}/roles/{role_id}
/projects/{project_id}/groups/{group_id}/roles/{role_id}
/projects/{project_id}/groups/{group_id}/roles/{role_id}
/projects/{project_id}/groups/{group_id}/roles/{role_id}
/projects/{project_id}/users/{user_id}/roles
/projects/{project_id}/groups/{group_id}/roles
/domains/{domain_id}/users/{user_id}/roles/{role_id}
/domains/{domain_id}/users/{user_id}/roles/{role_id}
/domains/{domain_id}/users/{user_id}/roles/{role_id}
/domains/{domain_id}/groups/{group_id}/roles/{role_id}
/domains/{domain_id}/groups/{group_id}/roles/{role_id}
/domains/{domain_id}/groups/{group_id}/roles/{role_id}
/domains/{domain_id}/users/{user_id}/roles
/domains/{domain_id}/groups/{group_id}/roles
/system/users/{user_id}/roles
/system/users/{user_id}/roles/{role_id}
/system/users/{user_id}/roles/{role_id}
/system/users/{user_id}/roles/{role_id}
/system/groups/{group_id}/roles
/system/groups/{group_id}/roles/{role_id}
/system/groups/{group_id}/roles/{role_id}
/system/groups/{group_id}/roles/{role_id}
/role_assignments
/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/inherited_to_projects
/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/inherited_to_projects
/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects
/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects
/regions/{region_id}
/OS-EP-FILTER/endpoints/{endpoint_id}/projects
/OS-EP-FILTER/projects/{project_id}/endpoints/{endpoint_id}
/OS-EP-FILTER/projects/{project_id}/endpoints/{endpoint_id}
/OS-EP-FILTER/projects/{project_id}/endpoints/{endpoint_id}
/OS-EP-FILTER/projects/{project_id}/endpoints
/OS-EP-FILTER/projects/{project_id}/endpoint_groups
/OS-EP-FILTER/endpoint_groups
/OS-EP-FILTER/endpoint_groups
/OS-EP-FILTER/endpoint_groups/{endpoint_group_id}
/OS-EP-FILTER/endpoint_groups/{endpoint_group_id}
/OS-EP-FILTER/endpoint_groups/{endpoint_group_id}
/OS-EP-FILTER/endpoint_groups/{endpoint_group_id}/projects/{project_id}
/OS-EP-FILTER/endpoint_groups/{endpoint_group_id}/projects/{project_id}
/OS-EP-FILTER/endpoint_groups/{endpoint_group_id}/projects/{project_id}
/OS-EP-FILTER/endpoint_groups/{endpoint_group_id}/projects
/OS-EP-FILTER/endpoint_groups/{endpoint_group_id}/endpoints
/users/{user_id}/password
/groups/{group_id}/users
/groups/{group_id}/users/{user_id}
/groups/{group_id}/users/{user_id}
/groups/{group_id}/users/{user_id}
/users/{user_id}/groups
/users/{user_id}/application_credentials
/users/{user_id}/application_credentials
/users/{user_id}/application_credentials/{application_credential_id}
/users/{user_id}/application_credentials/{application_credential_id}
/registered_limits
```

```
/registered_limits
/registered_limits
/registered_limits/{registered_limit_id}
/registered_limits/{registered_limit_id}
/limits
/limits
/limits
/limits/{limit_id}
/limits/{limit_id}
/domains/{domain_id}/config
/domains/{domain_id}/config
/domains/{domain_id}/config
/domains/{domain_id}/config
/domains/{domain_id}/config/{group}
/domains/{domain_id}/config/{group}
/domains/{domain_id}/config/{group}
/domains/{domain_id}/config/{group}/{option}
/domains/{domain_id}/config/{group}/{option}
/domains/{domain_id}/config/{group}/{option}
/domains/config/default
/domains/config/{group}/default
/domains/config/{group}/{option}/default
/projects/{project_id}/tags
/projects/{project_id}/tags
/projects/{project_id}/tags
/projects/{project_id}/tags/{value}
/projects/{project_id}/tags/{value}
/projects/{project_id}/tags/{value}
/OS-REVOKE/events
/OS-FEDERATION/identity_providers/{idp_id}
/OS-FEDERATION/identity_providers/{idp_id}
/OS-FEDERATION/identity_providers/{idp_id}
/OS-FEDERATION/identity_providers/{idp_id}
/OS-FEDERATION/identity_providers
/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}
/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}
/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}
/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}
/OS-FEDERATION/identity_providers/{idp_id}/protocols
/OS-FEDERATION/mappings/{mapping_id}
/OS-FEDERATION/mappings/{mapping_id}
/OS-FEDERATION/mappings/{mapping_id}
/OS-FEDERATION/mappings/{mapping_id}
/OS-FEDERATION/mappings
/OS-FEDERATION/service_providers/{sp_id}
/OS-FEDERATION/service_providers/{sp_id}
/OS-FEDERATION/service_providers/{sp_id}
/OS-FEDERATION/service_providers/{sp_id}
/OS-FEDERATION/service_providers
/OS-FEDERATION/domains
/OS-FEDERATION/projects
/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}/auth
/auth/OS-FEDERATION/saml2
/auth/OS-FEDERATION/saml2/ecp
/auth/OS-FEDERATION/websso/{protocol_id}
/auth/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}/websso
/OS-FEDERATION/saml2/metadata
/OS-OAUTH1/consumers
/OS-OAUTH1/consumers
/OS-OAUTH1/consumers/{consumer_id}
/OS-OAUTH1/consumers/{consumer_id}
/OS-OAUTH1/consumers/{consumer_id}
/users/{user_id}/OS-OAUTH1/access_tokens
/users/{user_id}/OS-OAUTH1/access_tokens/{access_token_id}
/users/{user_id}/OS-OAUTH1/access_tokens/{access_token_id}
/users/{user_id}/OS-OAUTH1/access_tokens/{access_token_id}/roles
/users/{user_id}/OS-OAUTH1/access_tokens/{access_token_id}/roles/{role_id}
/OS-OAUTH1/request_token
/OS-OAUTH1/access_token
/OS-OAUTH1/authorize/{request_token_id}
/endpoints/{endpoint_id}/OS-ENDPOINT-POLICY/policy
/policies/{policy_id}/OS-ENDPOINT-POLICY/endpoints
/policies/{policy_id}/OS-ENDPOINT-POLICY/endpoints/{endpoint_id}
/policies/{policy_id}/OS-ENDPOINT-POLICY/endpoints/{endpoint_id}
/policies/{policy_id}/OS-ENDPOINT-POLICY/endpoints/{endpoint_id}
/policies/{policy_id}/OS-ENDPOINT-POLICY/services/{service_id}
/policies/{policy_id}/OS-ENDPOINT-POLICY/services/{service_id}
/policies/{policy_id}/OS-ENDPOINT-POLICY/services/{service_id}
/policies/{policy_id}/OS-ENDPOINT-POLICY/services/{service_id}/regions/{region_id}
/policies/{policy_id}/OS-ENDPOINT-POLICY/services/{service_id}/regions/{region_id}
/policies/{policy_id}/OS-ENDPOINT-POLICY/services/{service_id}/regions/{region_id}
/OS-SIMPLE-CERT/ca
/OS-SIMPLE-CERT/certificates
/OS-TRUST/trusts
/OS-TRUST/trusts
/OS-TRUST/trusts/{trust_id}
/OS-TRUST/trusts/{trust_id}
```

```
/OS-TRUST/trusts/{trust_id}/roles
/OS-TRUST/trusts/{trust_id}/roles/{role_id}
/roles
/roles
/roles/{role_id}
/roles/{role_id}
/roles/{role_id}
/regions
/regions
/regions/{region_id}
/regions/{region_id}
/regions/{region_id}
/services
/services
/services/{service_id}
/services/{service_id}
/services/{service_id}
/endpoints
/endpoints
/endpoints/{endpoint_id}
/endpoints/{endpoint_id}
/endpoints/{endpoint_id}
/credentials
/credentials
/credentials/{credential_id}
/credentials/{credential_id}
/credentials/{credential_id}
/users
/users
/users/{user_id}
/users/{user_id}
/users/{user_id}
/groups
/groups
/groups/{group_id}
/groups/{group_id}
/groups/{group_id}
/policies
/policies
/policies/{policy_id}
/policies/{policy_id}
/policies/{policy_id}
/domains
/domains
/domains/{domain_id}
/domains/{domain_id}
/domains/{domain_id}
/projects
/projects
/projects/{project_id}
/projects/{project_id}
/projects/{project_id}
/
```

by Adam Young at March 06, 2018 05:17 PM (http://adam.younglogic.com/2018/03/inspecting-keystone-routes/)

OpenStack Superuser (http://superuser.openstack.org)

Pushing embedded devices closer to the edge (http://superuser.openstack.org/articles/pushing-embedded-devices-closer-edge/)

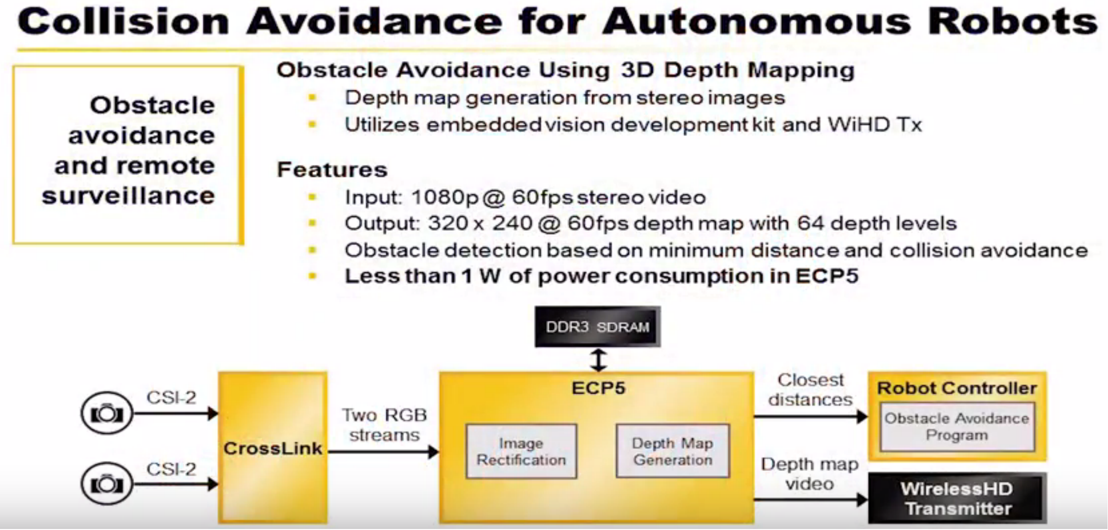
Lattice Semiconductor is a Silicon Valley maker of programmable logic devices founded in 1983 (yep, that's the "Halt and Catch Fire" era (https://en.wikipedia.org/wiki/Halt_and_Catch_Fire_(TV_series))), that sees the future in edge computing.

"AI and the cloud is old news, it's been implemented over the last few years," says Deepak Boppana, senior director, product and segment marketing at Lattice, said at the recent edition of Embedded World (https://www.embedded-world.eu) held in Nuremberg, Germany. "The move of AI is towards the edge: that's the future."

To show how they're pushing closer to that future, Boppana has been on the conference circuit with the company's low-power, wireless field-programmable gate array (FPGA) devices. And when they say low power, they mean low power: some of the solutions are in the range of milliwatts — as in your average laser pointer.

At the 2018 edition of the Consumer Electronics Show (CES), he showcased (https://www.youtube.com/watch?v=34xqb--2zTQ) this demo of vision applications for collision avoidance — with the toy robot gracefully moving around cones without hitting any.

The autonomous robot glides without incident across the factory floor using under one watt of power, Boppana says, adding "that's really key for battery powered applications at the edge." A wireless HD link uploads the video of the robot to its human overlords. Some key specs:



Collision Avoidance for Autonomous Robots

Obstacle avoidance and remote surveillance

Obstacle Avoidance Using 3D Depth Mapping

- Depth map generation from stereo images
- Utilizes embedded vision development kit and WiHD Tx

Features

- Input: 1080p @ 60fps stereo video
- Output: 320 x 240 @ 60fps depth map with 64 depth levels
- Obstacle detection based on minimum distance and collision avoidance
- Less than 1 W of power consumption in ECP5

Other solutions Lattice is working on include AI face-based tracking (https://youtu.be/34xqb--2zTQ?t=86) (it works for objects, too) that could be implemented with drones or surveillance cameras and facial recognition (https://youtu.be/xkFeVFcHPQU?t=82) solution. That low power iCE40 UltraPlus FPGA performs edge computing applications locally on a device, sending metadata to the cloud but does that despite consuming than a milliwatt of power. Lattice offerings also feature a host of city applications (https://www.youtube.com/watch?v=xZhGiHrnghY) — cameras that could be used for traffic flow monitoring, driving violations, smart parking and toll collection.

Check out the interview from Embedded World 2018 (and take a look the robots!) below. And you can catch up on these IRL at the upcoming Embedded Vision Conference (http://superuser.openstack.org/summit/sponsors-exhibitors) in May 2018.

Via Electronics Weekly (https://www.electronicsweekly.com/tag/embedded-world/)

The post Pushing embedded devices closer to the edge (http://superuser.openstack.org/articles/pushing-embedded-devices-closer-edge/) appeared first on Superuser (http://superuser.openstack.org).

by Superuser at March 06, 2018 04:01 PM (http://superuser.openstack.org/articles/pushing-embedded-devices-closer-edge/)

March 05, 2018

OpenStack Superuser (http://superuser.openstack.org)

OpenStack Community Contributor Awards now open for nominations (http://superuser.openstack.org/articles/openstack-contributor-awards-nominations/)

So many folks work tirelessly behind the scenes to make OpenStack great, whether they are fixing bugs, contributing code, helping newbies on IRC or just making everyone laugh at the right moment.

You can help them get recognized (with a very shiny medal! (http://superuser.openstack.org/articles/openstack-community-contributor-awards-recognize-unsung-heroes-2/)) by nominating them (https://openstackfoundation.formstack.com/forms/cca_nominations_vancouver) for the next Contributor Awards given out at the upcoming OpenStack Summit Vancouver (https://www.openstack.org/summit/). These are informal, quirky awards — winners in previous editions included the "Duct Tape" award and the "Don't Stop Believin' Cup" — that shine a light on the extremely valuable work that makes OpenStack excel.

There's so many different areas worthy of celebration, but there are a few kinds of community members who deserve a little extra love:

- They are undervalued
- They don't know they are appreciated
- They bind the community together
- They keep it fun
- They challenge the norm
  Other: (write-in)

As before (http://superuser.openstack.org/articles/openstack-community-contributor-awards-recognize-unsung-heroes/), rather than starting with a defined set of awards, the community is asked to submit names in those broad categories. The OSF community team then has a little bit of fun on the back end, massaging the award titles to devise something worthy of their underappreciated efforts.

> Thanks @OpenStack (https://twitter.com/OpenStack) community for the mentor of mentors award 🙂 I feel honored! #WeAreOpenStack (https://twitter.com/hashtag/WeAreOpenStack?src=hash) pic.twitter.com/LzY1pRKGZE (https://t.co/LzY1pRKGZE)
>
> — vkmc (@vkmc) April 29, 2016 (https://twitter.com/vkmc/status/726109994567917568)

"There are A LOT of people out there who could use a pat on the back and the affirmation that they do good work for the community, " says OSF's Kendall Nelson. Please nominate anyone you think is deserving of an award! Deadline is May 14.

https://openstackfoundation.formstack.com/forms/cca_nominations Vancouver (https://openstackfoundation.formstack.com/forms/cca_nominations_vancouver)

Awards will be presented during the feedback session at the Summit.

Cover Photo (https://www.flickr.com/photos/morenoberti/8606537726//) // CC BY NC (https://creativecommons.org/licenses/by-nc/2.0/)

The post OpenStack Community Contributor Awards now open for nominations (http://superuser.openstack.org/articles/openstack-contributor-awards-nominations/) appeared first on Superuser (http://superuser.openstack.org).

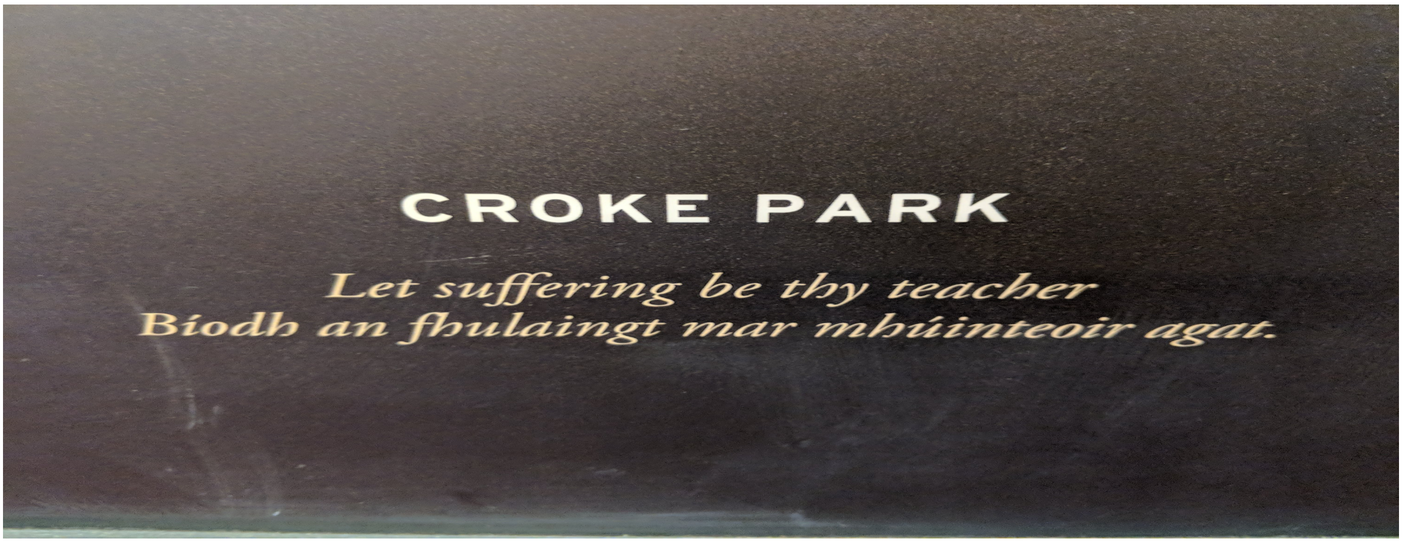
by Superuser at March 05, 2018 12:15 PM (http://superuser.openstack.org/articles/openstack-contributor-awards-nominations/)

March 04, 2018

Rich Bowen (http://drbacchus.com)

SnowpenStack (http://drbacchus.com/snowpenstack/)

I'm heading home from SnowpenStack (http://openstack.org/ptg) and it was quite a ride. As Theirry said in our interview at the end of Friday (coming soon to a YouTube channel (http://youtube.com/RDOCommunity) near you), rather than spoiling things, the freak storm and subsequent closure of the event venue served to create a shared experience and camaraderie that made it even better.


CROKE PARK
Let suffering be thy teacher
Bíodh an fhulaingt mar mhúinteoir agat.

In the end I believe I got 29 interviews, and I'll hopefully be supplementing this with a dozen online interviews in the coming weeks.

If you missed your interview, or weren't at the PTG, please contact me and we'll set something up. And I'll be in touch with all of the PTLs who were not already represented in one of my interviews.

A huge thank you to everyone that made time to do an interview, and to Erin and Kendall for making everything onsite go so smoothly.

by rbowen at March 04, 2018 07:52 AM (http://drbacchus.com/snowpenstack/)

March 02, 2018

OpenStack Superuser (http://superuser.openstack.org)

How a large multinational seeds cloud native (http://superuser.openstack.org/articles/huawei-k8s-user-story/)

By any measure, the scale of Huawei is impressive. To keep things humming for its 180,000 employees worldwide, the internal IT department of the Shenzen-based multinational telco giant runs eight data centers, powering over 800 applications and 100,000 virtual machines. Cost and agility were key factors in deciding the future of that system.

Huawei's Kubernetes journey began two years ago with one developer, who started contributing code to the open-source project and quickly became an internal evangelist. It seemed like a good solution for the internal IT department, which needs to launch tens of thousands of containers. Because those containerspower tasks across thousands of global nodes, managing them in a consistent way is always a challenge, says Peixin Hou, Huawei's chief software architect and community director for open source, in a recent case study (https://kubernetes.io/case-studies/huawei/). About 30 percent of them are now cloud native.

It's hardly the first time Huawei embraced (https://databricks.com/blog/2015/06/09/huawei-embraces-open-source-apache-spark.html) open source. A gold member of the OpenStack Foundation, Huawei offers customers a carrier-grade OpenStack distribution supporting NFV deployments and Huawei FusionSphere (http://e.huawei.com/en/products/cloud-computing-dc/cloud-computing/fusionsphere/fusionsphere) to deploy private, public and hybrid clouds. They're also (http://www.huawei.com/en/about-huawei/publications/communicate/81/open-source-powers-cloud-ecosystem) a top contributor to Hadoop, OPNFV, a founding member of the Cloud Native Computing Foundation (CCNF) and employ Spark. In addition to an online lab (http://developer.huawei.com/ict/en/remotelab) and courses, the company's website also offers an open source release center (https://consumer.huawei.com/en/opensource/) where anyone can download or request open source code found in its phones, computers, wearables and smart home devices.

Although took a year to adopt that engineer's suggestion – the process "is not overnight," says Hou – but once in use, he says: "Kubernetes basically solved most of our problems. Before, the time of deployment took about a week, now it only takes minutes. The developers are happy. That department is also quite happy."

The results were impressive enough that the company also decided to become a vendor in 2016 with FusionStage, (http://developer.huawei.com/ict/en/site-paas) a platform-as-a-service offering. Huawei's customers, from very big telecommunications operators to banks, love the idea of cloud native, Hou says.

After overseeing the initial move to Kubernetes at Huawei, Hou has some advice for other companies considering the technology: "When you start to design the architecture of your application, think about cloud native, think about microservice architecture from the beginning," he says.

For companies dealing with legacy apps, Hou advises to dip a toe in with the microservice-friendly part of those applications first, parts that are relatively easy to be decomposed into simpler pieces and are relatively lightweight.

More from the full case study here. (https://kubernetes.io/case-studies/huawei/)

Cover image: Huawei Shenzen R&D center, courtesy Huawei.

The post How a large multinational seeds cloud native (http://superuser.openstack.org/articles/huawei-k8s-user-story/) appeared first on Superuser (http://superuser.openstack.org).

by Superuser at March 02, 2018 04:52 PM (http://superuser.openstack.org/articles/huawei-k8s-user-story/)

RDO (https://blogs.rdoproject.org)

March 1 Blogpost Roundup (https://blogs.rdoproject.org/2018/03/march-1-blogpost-roundup/)

It's been a busy few weeks of blogging! Thanks as always to those of you who continue to write great content.

OpenStack Role Assignment Inheritance for CloudForms by Adam Young

> Operators expect to use CloudForms to perform administrative tasks. For this reason, the documentation for OpenStack states that the Keystone user must have an 'admin' role. We found at least one case, however, where this was not sufficient. Fortunately, we have a better approach, and one that can lead to success in a wider array of deployments.

Read more at http://adam.younglogic.com/2018/02/openstack-hmt-cloudforms/ (http://adam.younglogic.com/2018/02/openstack-hmt-cloudforms/)

Listening for connections on all ports/any port by Lars Kellogg-Stedman

> On IRC — and other online communities — it is common to use a "pastebin" service to share snippets of code, logs, and other material, rather than pasting them directly into a conversation. These services will typically return a URL that you can share with others so that they can see the content in their browser.

Read more at http://blog.oddbit.com/2018/02/27/listening-for-connections-on-a/ (http://blog.oddbit.com/2018/02/27/listening-for-connections-on-a/)

Grouping aggregation queries in Gnocchi 4.0.x by Lars Kellogg-Stedman

> In this article, we're going to ask Gnocchi (the OpenStack telemetry storage service) how much memory was used, on average, over the course of each day by each project in an OpenStack environment.

Read more at http://blog.oddbit.com/2018/02/26/grouping-aggregation-queries-i/ (http://blog.oddbit.com/2018/02/26/grouping-aggregation-queries-i/)

TripleO deep dive session #12 (config-download) by Carlos Camacho

> This is the 12th release of the TripleO "Deep Dive" sessions. In this session we will have an update for the TripleO ansible integration called config-download. It's about applying all the software configuration with Ansible instead of doing it with the Heat agents.

Read more at https://www.anstack.com/blog/2018/02/23/tripleo-deep-dive-session-12.html (https://www.anstack.com/blog/2018/02/23/tripleo-deep-dive-session-12.html)

Maximizing resource utilization with Preemptible Instances by Theodoros Tsioutsias

> The CERN cloud consists of around 8,500 hypervisors providing over 36,000 virtual machines. These provide the compute resources for both the laboratory's physics program but also for the organisation's administrative operations such as paying bills and reserving rooms at the hostel.

Read more at http://openstack-in-production.blogspot.com/2018/02/maximizing-resource-utilization-with.html (http://openstack-in-production.blogspot.com/2018/02/maximizing-resource-utilization-with.html)

Testing TripleO on own OpenStack deployment by mrunge

> For some use cases, it's quite useful to test TripleO deployments on a OpenStack powered cloud, rather than using a baremetal system. The following article will show you how to do it.

Read more at http://www.matthias-runge.de/2018/02/16/tripleo-ovb/ (http://www.matthias-runge.de/2018/02/16/tripleo-ovb/)

A New Thing by Andrew Beekhof

> If you're interested in Kubernetes and/or managing replicated applications, such as Galera, then you might also be interested in an operator that allows this class of applications to be managed natively by Kubernetes.

Read more at http://blog.clusterlabs.org/blog/2018/replication-operator (http://blog.clusterlabs.org/blog/2018/replication-operator)

Two Nodes – The Devil is in the Details by Andrew Beekhof

> tl;dr – Many people love 2-node clusters because they seem conceptually simpler and 33% cheaper, but while it's possible to construct good ones, most will have subtle failure modes.

Read more at http://blog.clusterlabs.org/blog/2018/two-node-problems (http://blog.clusterlabs.org/blog/2018/two-node-problems)

by Mary Thengvall at March 02, 2018 01:08 AM (https://blogs.rdoproject.org/2018/03/march-1-blogpost-roundup/)

March 01, 2018

Corey Bryant (https://wrestlingpenguins.wordpress.com)

OpenStack Queens for Ubuntu 16.04 LTS (https://wrestlingpenguins.wordpress.com/2018/03/01/openstack-queens-for-ubuntu-16-04-lts/)

Hi All,

The Ubuntu OpenStack team at Canonical is pleased to announce the general availability of OpenStack Queens on Ubuntu 16.04 LTS via the Ubuntu Cloud Archive. Details of the Queens release can be found at: https://www.openstack.org/software/queens (https://www.openstack.org/software/queens)

To get access to the Ubuntu Queens packages:

Ubuntu 16.04 LTS

You can enable the Ubuntu Cloud Archive pocket for OpenStack Queens on Ubuntu 16.04 installations by running the following commands:

```
sudo add-apt-repository cloud-archive:queens
sudo apt update
```

The Ubuntu Cloud Archive for Queens includes updates for:

aodh, barbican, ceilometer, ceph (12.2.2), cinder, congress, designate, designate-dashboard, dpdk (17.11), glance, glusterfs (3.13.2), gnocchi, heat, heat-dashboard, horizon, ironic, keystone, libvirt (4.0.0), magnum, manila, manila-ui, mistral, murano, murano-dashboard, networking-bagpipe, networking-bgpvpn, networking-hyperv, networking-l2gw, networking-odl, networking-ovn, networking-sfc, neutron, neutron-dynamic-routing, neutron-fwaas, neutron-lbaas, neutron-lbaas-dashboard, neutron-taas, neutron-vpnaas, nova, nova-lxd, openstack-trove, openvswitch (2.9.0), panko, qemu (2.11), rabbitmq-server (3.6.10), sahara, sahara-dashboard, senlin, swift, trove-dashboard, vmware-nsx, watcher, and zaqar.

For a full list of packages and versions, please refer to [0].

Branch Package Builds

If you would like to try out the latest updates to branches, we deliver continuously integrated packages on each upstream commit via the following PPA's:

```
sudo add-apt-repository ppa:openstack-ubuntu-testing/mitaka
sudo add-apt-repository ppa:openstack-ubuntu-testing/newton
sudo add-apt-repository ppa:openstack-ubuntu-testing/ocata
sudo add-apt-repository ppa:openstack-ubuntu-testing/pike
sudo add-apt-repository ppa:openstack-ubuntu-testing/queens
```

Reporting bugs

If you have any issues please report bugs using the 'ubuntu-bug' tool to ensure that bugs get logged in the right place in Launchpad:

```
sudo ubuntu-bug nova-conductor
```

Thanks to everyone who has contributed to OpenStack Queens, both upstream and downstream!

Have fun and see you in Rocky!

Regards,
Corey
(on behalf of the Ubuntu OpenStack team)

[0] http://reqorts.qa.ubuntu.com/reports/ubuntu-server/cloud-archive/queens_versions.html (http://reqorts.qa.ubuntu.com/reports/ubuntu-server/cloud-archive/queens_versions.html)

by coreycb at March 01, 2018 09:31 PM (https://wrestlingpenguins.wordpress.com/2018/03/01/openstack-queens-for-ubuntu-16-04-lts/)

OpenStack Superuser (http://superuser.openstack.org)

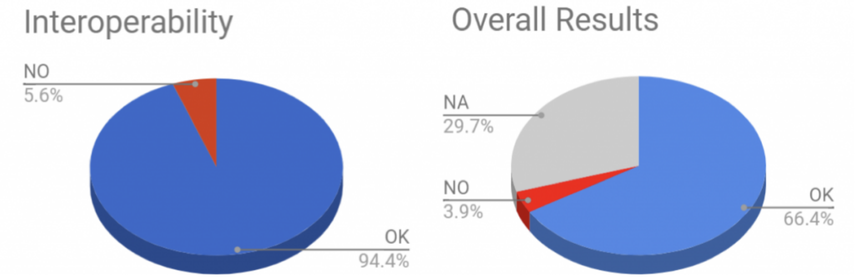
Standards tests show that network functions virtualization has VIM (http://superuser.openstack.org/articles/etsi-nfv-plugtests-vim/)

In the first post (http://superuser.openstack.org/articles/nfv-plugtests-part-1/) of this series, I shared a summary about the week I spent in France as part of the European Telecommunications Standards Institute's (ETSI) second edition of NFV Plugtests.

Now that the official results (https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/2nd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf) are out, let's dive in!

A quick summary: 95 percent of the 189 interoperability test sessions succeeded. (A 'session' means a combination of different vendor's VNF, VIM+NFVI and MANO solutions.) Overall, only about 30 percent of the features were not yet supported by one of the involved participants, compared to 61 percent at the first Plugtest session. This speaks very well to NFV readiness, considering that more than 50 percent of additional features were considered for this second edition, including multi-vendor network services, fault and performance management, enhanced platform awareness and multi-site deployments.

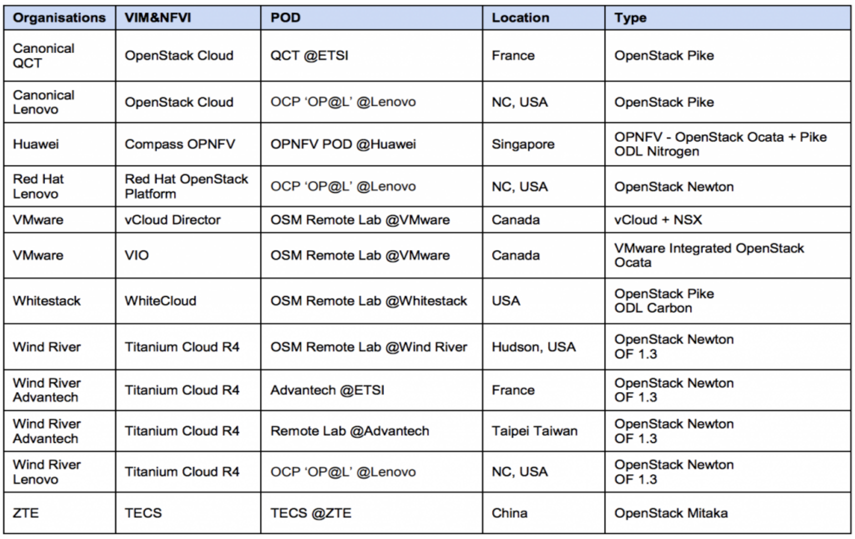
Note: all graphics and tables are as published in the second NFV Plugtestsreport (https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/2nd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf)



(http://portal.etsi.org/Portals/0/TBpages/CTI/Docs/2nd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf)

Here, I'll focus on the results related to the NFVI and virtualized infrastructure manager (VIM, a.k.a. telco cloud), which as you may already know, are the physical compute, storage and network resources (NFVI), as well as their manager, that provide a cloud environment for virtualized network functions.

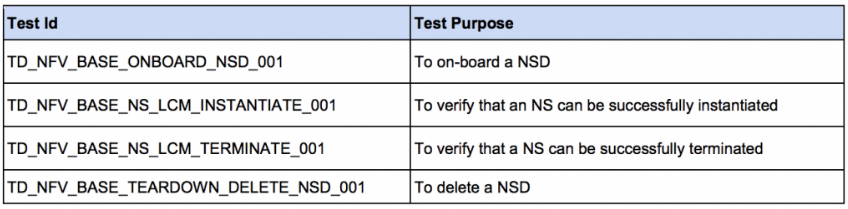
Twelve different NFVI+VIM combinations were tested, includin 11 with different OpenStack distributions (from seven VIM vendors) and one with VMWare's vCloud Director.



Organisations	VIM&NFVI	POD	Location	Type
Canonical QCT	OpenStack Cloud	QCT @ETSI	France	OpenStack Pike
Canonical Lenovo	OpenStack Cloud	OCP 'OP@L' @Lenovo	NC, USA	OpenStack Pike
Huawei	Compass OPNFV	OPNFV POD @Huawei	Singapore	OPNFV - OpenStack Ocata + Pike ODL Nitrogen
Red Hat Lenovo	Red Hat OpenStack Platform	OCP 'OP@L' @Lenovo	NC, USA	OpenStack Newton
VMware	vCloud Director	OSM Remote Lab @VMware	Canada	vCloud + NSX
VMware	VIO	OSM Remote Lab @VMware	Canada	VMware Integrated OpenStack Ocata
Whitestack	WhiteCloud	OSM Remote Lab @Whitestack	USA	OpenStack Pike ODL Carbon
Wind River	Titanium Cloud R4	OSM Remote Lab @Wind River	Hudson, USA	OpenStack Newton OF 1.3
Wind River Advantech	Titanium Cloud R4	Advantech @ETSI	France	OpenStack Newton OF 1.3
Wind River Advantech	Titanium Cloud R4	Remote Lab @Advantech	Taipei Taiwan	OpenStack Newton OF 1.3
Wind River Lenovo	Titanium Cloud R4	OCP 'OP@L' @Lenovo	NC, USA	OpenStack Newton OF 1.3
ZTE	TECS	TECS @ZTE	China	OpenStack Mitaka

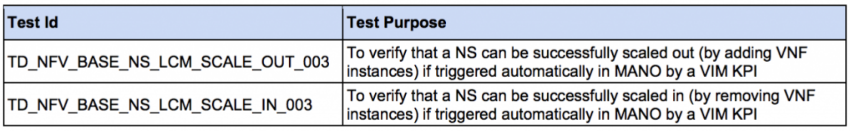
(http://portal.etsi.org/Portals/0/TBpages/CTI/Docs/2nd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf)

Since NFVI is expected to be based on off-the-shelf generic servers with homogeneous features, most challenges were presented to the VIM software. The following tables show some examples of the kind of tests that were included.
There were basic tests like these, requesting basic lifecycle management:



Test Id	Test Purpose
TD_NFV_BASE_ONBOARD_NSD_001	To on-board a NSD
TD_NFV_BASE_NS_LCM_INSTANTIATE_001	To verify that an NS can be successfully instantiated
TD_NFV_BASE_NS_LCM_TERMINATE_001	To verify that a NS can be successfully terminated
TD_NFV_BASE_TEARDOWN_DELETE_NSD_001	To delete a NSD

(http://portal.etsi.org/Portals/0/TBpages/CTI/Docs/2nd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf)
… and more advanced ones, like these, requesting autoscaling based on VIM metrics:



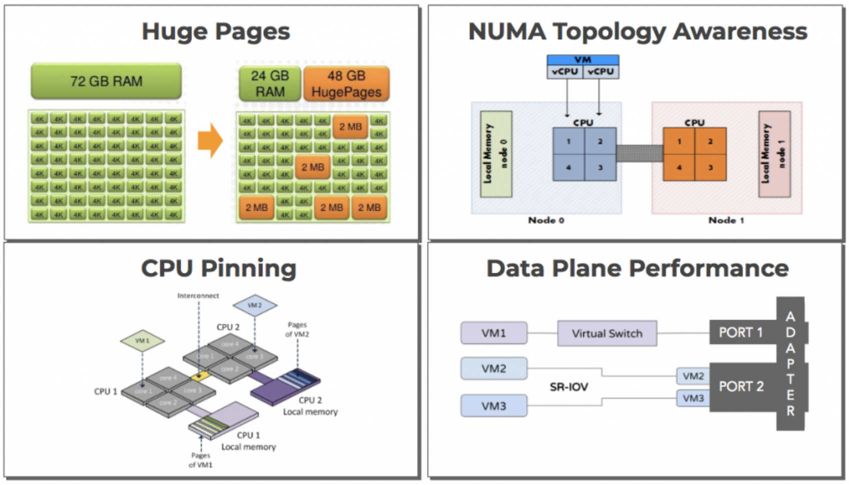
Test Id	Test Purpose
TD_NFV_BASE_NS_LCM_SCALE_OUT_003	To verify that a NS can be successfully scaled out (by adding VNF instances) if triggered automatically in MANO by a VIM KPI
TD_NFV_BASE_NS_LCM_SCALE_IN_003	To verify that a NS can be successfully scaled in (by removing VNF instances) if triggered automatically in MANO by a VIM KPI

(http://portal.etsi.org/Portals/0/TBpages/CTI/Docs/2nd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf)
As you can see, test descriptions cover the NFV system as whole, but, how do these tests relate to specific features that VIM platforms had to support?

VIMs were expected to support:

-
  - **Standard APIs**: to provide a way for MANO platforms to connect and orchestrate VIM functions.
  - **Standard hypervisor features**: to be ready to support different VM image formats that VNFs may require.
  - **Complete VM lifecycle**: to be able to start, pause, stop and destroy VMs, along with their compute, networking and storage resources.
  - **Networking**: to provide standard networking features for VMs to interconnect in a number of ways: private and public networks, NAT, security features, etc.
  - **Performance management**: to collect and store performance metrics for all VMs, while being able to set thresholds and alarms for any of them, so that MANO platforms can monitor KPIs and trigger actions.
  - **Fault Management**: to generate events and alarms and be able to propagate them to higher layers.
  - **Enhanced Platform Awareness (EPA)**: to be able to set EPA-related configurations on NFVI resources (SR-IOV, CPU pinning, NUMA topology awareness, Huge pages, etc.)

Cloud platforms are currently mature enough so most of the features are completely supported by both OpenStack distributions and VMWare vCD. Maybe the only one that today still presents some challenges at the VIM layer is EPA management, since this is a recent concept created during the NFV evolution to optimize the NFVI performance using techniques that should work with generic servers.

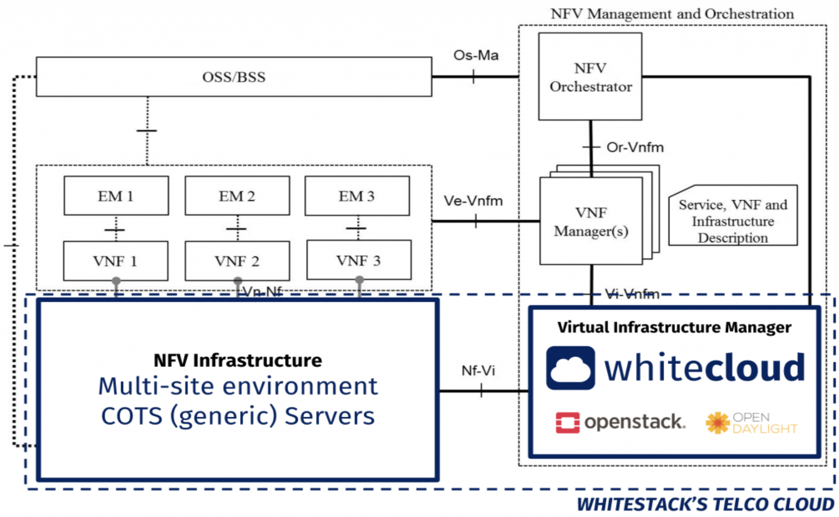


During these Plugtests, EPA testing was included in the optional set of sessions and the actual feature to test was not specified, so participants could choose to test simple control-plane related features like Huge Pages or CPU Pinning, or choose more complex, data-plane related features like SR-IOV, OVS-DPDK.

As evidenced in the following table, for Multi-VNF EPA there were only eight sessions, which included 26 tests. Only 12 could actually be run, while 14 asked for features that were not supported by participants. Out of these 12 tests, nine succeeded and three exposed interoperability issues. In short, only about a third of the EPA tests were successful.

Results per Group (optional sessions)	Number of Test Sessions	Interoperability (TCs Run)		TCs Not Run	TCs Totals	
		OK	NO	NA	Run	Total
Multi-VNF-EPA	8	9 (75%)	3 (25%)	14 (53.8%)	12 (46.2%)	26
Multi-Site	9	32 (100%)	0 (0%)	2 (5.9%)	32 (94.1%)	34
S-VNFM-D	0	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0
S-VNFM-I	0	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0
Auto-LCM-validation	8	31 (96.9%)	1 (3.1%)	16 (33.3%)	32 (66.7%)	48

(http://portal.etsi.org/Portals/0/TBpages/CTI/Docs/2nd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf)
We participated with Whitestack's VIM solution, WhiteCloud (https://www.openstack.org/marketplace/distros/distribution/whitestack/whitecloud), a multi-site telco cloud environment based on OpenStack Pike (https://www.openstack.org/software/pike/)over generic servers and switches. I'm happy to report that it passed most tests successfully when integrating with different MANO solutions and VNFs. We didn't focus our testing efforts for data-plane related EPA features this time, but we expect to disclose full (and successful) results in the third edition of these tests, expected to take place in May 2018.



In conclusion, because most features and interoperability worked fine for participants, we can say that at the NFVI/VIM level, NFV is ready for prime time! Of course, new features will keep being developed at both standards and by vendors at different speeds, which increases the importance for continuing this event.

For production environments, additional attention must be paid when evaluating telco cloud solutions that require high throughput from servers. Data-plane acceleration features of the NFVI and VIM must be carefully assessed and tested, including the most popular alternatives: OVS-DPDK and SR-IOV.

Stay tuned for the final post in this series, where I'll share the results from the MANO perspective. Thanks for reading!

About the author

Gianpietro Lavado is a network solutions architect interested in the latest software technologies to achieve efficient and innovative network operations. He currently works at Whitestack, a company whose mission is to promote SDN, NFV, cloud and related deployments all around the world, including a new OpenStack distribution (https://www.openstack.org/marketplace/distros/distribution/whitestack/whitecloud).

This post first appeared on LinkedIn (https://www.linkedin.com/pulse/nfv-reaching-its-prime-time-part-2-2nd-etsi-plugtests-lavado/). Superuser is always interested in community content, get in touch at: editorATopenstack.org

Cover Photo (http://superuser.openstack.org/feed/"https://www.flickr.com/photos/clement127/21940272315/") // CC BY NC (https://creativecommons.org/licenses/by-nc/2.0/)

The post Standards tests show that network functions virtualization has VIM (http://superuser.openstack.org/articles/etsi-nfv-plugtests-vim/) appeared first on Superuser (http://superuser.openstack.org).

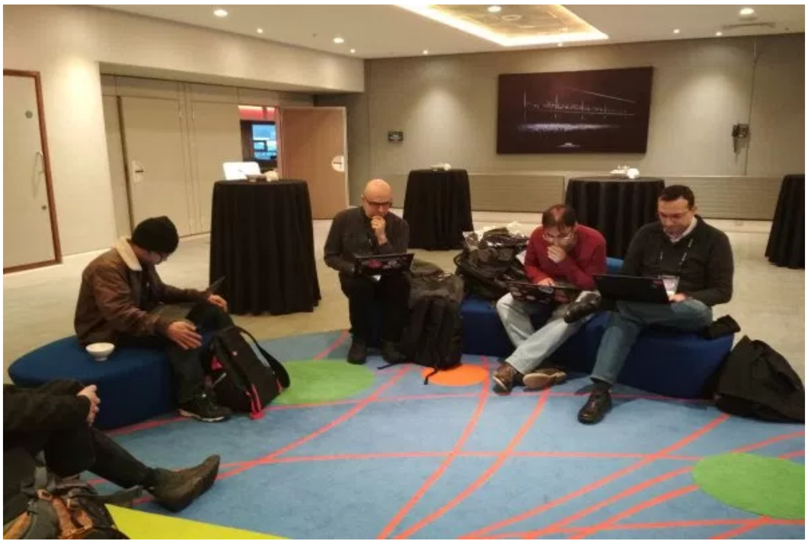
by Gianpietro Lavado at March 01, 2018 04:19 PM (http://superuser.openstack.org/articles/etsi-nfv-plugtests-vim/)

RDO (https://blogs.rdoproject.org)

RDO An OpenStack Distribution

RDO Queens Released (https://blogs.rdoproject.org/2018/03/rdo-queens-released/)

The RDO community is pleased to announce the general availability of the RDO build for OpenStack Queens for RPM-based distributions, CentOS Linux 7 and Red Hat Enterprise Linux. RDO is suitable for building private, public, and hybrid clouds. Queens is the 17th release from the OpenStack project, which is the work of more than 1600 contributors from around the world (source – http://stackalytics.com/ (http://stackalytics.com/)).



]2 (https://blogs.rdoproject.org/wp-content/uploads/2018/03/rdo_release_hackathon.jpg) RDO team doing the release at the PTG

The release is making its way out to the CentOS mirror network, and should be on your favorite mirror site momentarily.

The RDO community project curates, packages, builds, tests and maintains a complete OpenStack component set for RHEL and CentOS Linux and is a member of the CentOS Cloud Infrastructure SIG. The Cloud Infrastructure SIG focuses on delivering a great user experience for CentOS Linux users looking to build and maintain their own on-premise, public or hybrid clouds.

All work on RDO, and on the downstream release, Red Hat OpenStack Platform, is 100% open source, with all code changes going upstream first.

New and Improved

Interesting things in the Queens release include:

- Ironic now supports Neutron routed networks with flat networking and introduces support for Nova traits when scheduling
- RDO now includes rsdclient, an OpenStack client plugin for Rack Scale Design architecture
- Support for octaviaclient and Octavia Horizon plugin has been added to improve Octavia service deployments.
- Tap-as-a-Service (TaaS) network extension to the OpenStack network service (Neutron) has been included.
- Multi-vendor Modular Layer 2 (ML2) driver networking-generic-switch si now available of operators deploying RDO Queens.

Other improvements include:

- Most of the bundled intree tempest plugins have been moved to their own repository during Queens cycle. RDO has adapted plugin packages for these new model.
- In an effort to improve the quality and reduce the delivery time for our users, RDO keeps refining and automating all required processes needed to build, test and publish the packages included in RDO distribution.

Note that packages for OpenStack projects with cycle-trailing release models[*] will be created after a release is delivered according to the OpenStack Queens schedule. [*] https://releases.openstack.org/reference/release_models.html#cycle-trailing

Contributors

During the Queens cycle, we saw the following new contributors:

- Aditya Ramteke
- Jatan Malde
- Ade Lee
- James Slagle
- Alex Schultz
- Artom Lifshitz
- Mathieu Bultel
- Petr Viktorin
- Radomir Dopieralski
- Mark Hamzy
- Sagar Ippalpalli
- Martin Kopec
- Victoria Martinez de la Cruz
- Harald Jensas
- Kashyap Chamarthy
- dparalen

- Thiago da Silva
- chenxing
- Johan Guldmyr
- David J Peacock
- Sagi Shnaidman
- Jose Luis Franco Arza

Welcome to all of you, and thank you so much for participating!

But, we wouldn’t want to overlook anyone. Thank you to all 76 contributors who participated in producing this release. This list includes commits to rdo-packages and rdo-infra repositories, and is provided in no particular order:

- Yatin Karel
- Aditya Ramteke
- Javier Pena
- Alfredo Moralejo
- Christopher Brown
- Jon Schlueter
- Chandan Kumar
- Haikel Guemar
- Emilien Macchi
- Jatan Malde
- Pradeep Kilambi
- Luigi Toscano
- Alan Pevec
- Eric Harney
- Ben Nemec
- Matthias Runge
- Ade Lee
- Jakub Libosvar
- Thierry Vignaud
- Alex Schultz
- Juan Antonio Osorio Robles
- Mohammed Naser
- James Slagle
- Jason Joyce
- Artom Lifshitz
- Lon Hohberger
- rabi
- Dmitry Tantsur
- Oliver Walsh
- Mathieu Bultel
- Steve Baker
- Daniel Mellado
- Terry Wilson
- Tom Barron
- Jiri Stransky
- Ricardo Noriega
- Petr Viktorin
- Juan Antonio Osorio Robles
- Eduardo Gonzalez
- Radomir Dopieralski
- Mark Hamzy
- Sagar Ippalpalli
- Martin Kopec
- Ihar Hrachyshka
- Tristan Cacqueray
- Victoria Martinez de la Cruz
- Bernard Cafarelli
- Harald Jensas
- Assaf Muller
- Kashyap Chamarthy
- Jeremy Liu
- Daniel Alvarez
- Mehdi Abaakouk
- dparalen
- Thiago da Silva
- Brad P. Crochet
- chenxing
- Johan Guldmyr
- Antoni Segura Puimedon
- David J Peacock
- Sagi Shnaidman
- Jose Luis Franco Arza
- Julie Pichon
- David Moreau-Simard
- Wes Hayutin
- Attila Darazs
- Gabriele Cerami

- John Trowbridge
- Gonéri Le Bouder
- Ronelle Landy
- Matt Young
- Arx Cruz
- Joe H. Rahme
- marios
- Sofer Athlan-Guyot
- Paul Belanger

Getting Started

There are two ways to get started with RDO.

To spin up a proof of concept cloud, quickly, and on limited hardware, try an All-In-One Packstack installation. You can run RDO on a single node to get a feel for how it works. For a production deployment of RDO, use the TripleO Quickstart and you'll be running a production cloud in short order.

Getting Help

The RDO Project participates in a Q&A service at ask.openstack.org. We also have our users@lists.rdoproject.org for RDO-specific users and operrators. For more developer-oriented content we recommend joining the dev@lists.rdoproject.org mailing list. Remember to post a brief introduction about yourself and your RDO story. The mailng lists archives are all available at https://mail.rdoproject.org (https://mail.rdoproject.org) You can also find extensive documentation on the RDO docs site (http://rdoproject.org/use).

The #rdo channel on Freenode IRC is also an excellent place to find help and give help.

We also welcome comments and requests on the CentOS mailing lists and the CentOS and TripleO IRC channels (#centos, #centos-devel, and #tripleo on irc.freenode.net), however we have a more focused audience in the RDO venues.

Getting Involved

To get involved in the OpenStack RPM packaging effort, see the RDO community pages and the CentOS Cloud SIG page. See also the RDO packaging documentation.

Join us in #rdo on the Freenode IRC network, and follow us at @RDOCommunity on Twitter. If you prefer Facebook, we're there too, and also Google+.

by Rich Bowen at March 01, 2018 10:05 AM (https://blogs.rdoproject.org/2018/03/rdo-queens-released/)

SWITCH Cloud Blog (https://cloudblog.switch.ch)

Deploy Kubernetes on the SWITCHengines Openstack cloud (https://cloudblog.switch.ch/2017/06/26/deploy-kubernetes-on-the-switchengines-openstack-cloud/)

Increasing demand for container orchestration tools is coming from our users. Kubernetes has currently a lot of hype, and often it comes the question if we are providing a Kubernetes cluster at SWITCH.

At the moment we suggest that our users deploy their own Kubernetes cluster on top of SWITCHengines. To make sure our Openstack deployment works with this solution we tried ourself.

After deploying manually with kubeadm (https://kubernetes.io/docs/setup/independent/create-cluster-kubeadm/) to learn the tool, I found a well written ansible playbook from Francois Deppierraz. (https://github.com/infraly/k8s-on-openstack)I extended the playbook to make Kubernetes aware that SWITCHengines implements the LBaaSv2, and the patch (https://github.com/infraly/k8s-on-openstack/pull/1) is now merged in the original version.

The first problem I discovered deploying Kubernetes is the total lack of support for IPv6. Because instances in SWITCHengines get IPv6 addresses by default, I run into problems running the playbook and nothing was working. The first thing you should do is to create your own tenant network with a router, with IPv4 only connectivity. This is already explained in detail in our standard documentation. (https://help.switch.ch/engines/documentation/configure-your-network/)

Now we are ready to clone the ansible playbook:

```
git clone https://github.com/infraly/k8s-on-openstack
```

Because the ansible playbook creates instances through the Openstack API, you will have to source your Openstack configuration file. We extend a little bit the usual configuration file with more variables that are specific to this ansible playbook. Lets see a template:

```
export OS_USERNAME=username
export OS_PASSWORD=mypassword
export OS_PROJECT_NAME=myproject
export OS_PROJECT_ID=myproject_uuid
export OS_AUTH_URL=https://keystone.cloud.switch.ch:5000/v2.0
export OS_REGION_NAME=ZH
export KEY=keyname
export IMAGE="Ubuntu Xenial 16.04 (SWITCHengines)"
export NETWORK=k8s
export SUBNET_UUID=subnet_uuid
export FLOATING_IP_NETWORK_UUID=network_uuid
```

Lets review what changes. It is important to add also the variable *OS_PROJECT_ID* because the Kubernetes code that creates Load Balancers requires this value, and it is not able to extract it from the project name. To find the uuid just use the Openstack cli:

```
openstack project show myprojectname -f value -c id
```

The *KEY* is the name of an existing keypair that will be used to start the instances. The *IMAGE* is also self explicative, at the moment only Xenial is tested by me. The variable *NETWORK* is the name of the tenant network you created earlier. When you created a network you created also a subnet, and you need to set the uuid into *SUBNET_UUID*. The last variable is *FLOATING_IP_NETWORK_UUID* that tells kubernetes the network where to get floating IPs. In SWITCHengines this network is always called *public*, so you can extract the uuid like this:

```
openstack network show public -f value -c id
```

You can customize your configuration even more, reading the README file in the git repository you will find more options like the flavors to use or the cluster size. When your configuration file is ready you can run the playbook:

```
source /path/to/config_file
cd k8s-on-openstack
ansible-playbook site.yaml
```

It will take a few minutes to go through all the tasks. When everything is done you can ssh into the kubernetes master instance and check that everything is running as expected:

```
ubuntu@k8s-master:~$ kubectl get nodes
NAME         STATUS    AGE       VERSION
k8s-1        Ready     2d        v1.6.2
k8s-2        Ready     2d        v1.6.2
k8s-3        Ready     2d        v1.6.2
k8s-master   Ready     2d        v1.6.2
```

I found very useful adding bash completion for kubectl:

```
source <(kubectl completion bash)
```

Lets deploy an instance of nginx to test if everything works:

```
kubectl run my-nginx --image=nginx --replicas=2 --port=80
```

This will create two containers with nginx. You can monitor the progress with the commands:

```
kubectl get pods
kubectl get events
```

At this stage you have your containers running but the service is still not accessible from the outside. One option is to use the Openstack LBaaS to expose it, you can do it with this command:

```
kubectl expose deployment my-nginx --port=80 --type=LoadBalancer
```

The expose command will create the Openstack Load Balancer and will configure it. To know the public floating ip address you can use this command to describe the service:

```
ubuntu@k8s-master:~$ kubectl describe service my-nginx
Name:                   my-nginx
Namespace:              default
Labels:                 run=my-nginx
Annotations:
Selector:               run=my-nginx
Type:                   LoadBalancer
IP:                     10.109.12.171
LoadBalancer Ingress:   10.8.10.15, 86.119.34.151
Port:                           80/TCP
NodePort:                       30620/TCP
Endpoints:              10.40.0.1:80,10.43.0.1:80
Session Affinity:       None
Events:
  FirstSeen     LastSeen        Count   From                    SubObjectPath   Type            Reason                  Message
  ---------     --------        -----   ----                    -------------   --------        ------                  -------
  1m            1m              1       service-controller                      Normal          CreatingLoadBalancer    Creating load balancer
  10s           10s             1       service-controller                      Normal          CreatedLoadBalancer     Created load balancer
```

Conclusion

Following this blog post you should be able to deploy Kubernetes on Openstack to understand how things work. For a real deployment you might want to make some customisations, we encourage you to share any patch to the ansible playbook with github pull requests.
Please note that Kubernetes is not bug free. When you will delete your deployment you might find this bug where Kubernetes is not able to delete correctly the load balancer (https://github.com/kubernetes/kubernetes/issues/48094). Hopefully this is fixed by the time you read this blog post.

by Saverio Proto at March 01, 2018 12:23 AM (https://cloudblog.switch.ch/2017/06/26/deploy-kubernetes-on-the-switchengines-openstack-cloud/)

Carlos Camacho (https://www.anstack.com/)

My 2nd birthday as a Red Hatter (https://www.anstack.com/blog/2018/03/01/2nd-birthday-as-a-red-hatter.html)

This post will be about to speak about my experience working in TripleO as a Red Hatter for the last 2 years. In my 2nd birthday as a Red Hatter, I have learned about many technologies, really a lot… But the most intriguing thing is that here you never stop learning. Not just because you just don't want to learn new things, instead, is because of the project's nature, this project… TripleO…

TripleO (Openstack On Openstack) is a software aimed to deploy OpenStack services using the same OpenStack ecosystem, this means that we will deploy a minimal OpenStack instance (Undercloud) and from there, deploy our production environment (Overcloud)… Yikes! What a mouthful, huh? Put simply, TripleO is an installer which should make integrators/operators/developers lives easier, but the reality sometimes is far away from the expectation.

TripleO is capable of doing wonderful things, with a little of patience, love, and dedication, your hands can be the right hands to deploy complex environments at easy.

One of the cool things being one of the programmers who write TripleO, from now on TripleOers, is that many of us also use the software regularly. We are writing code not just because we are told to do it, but because we want to improve it for our own purposes.

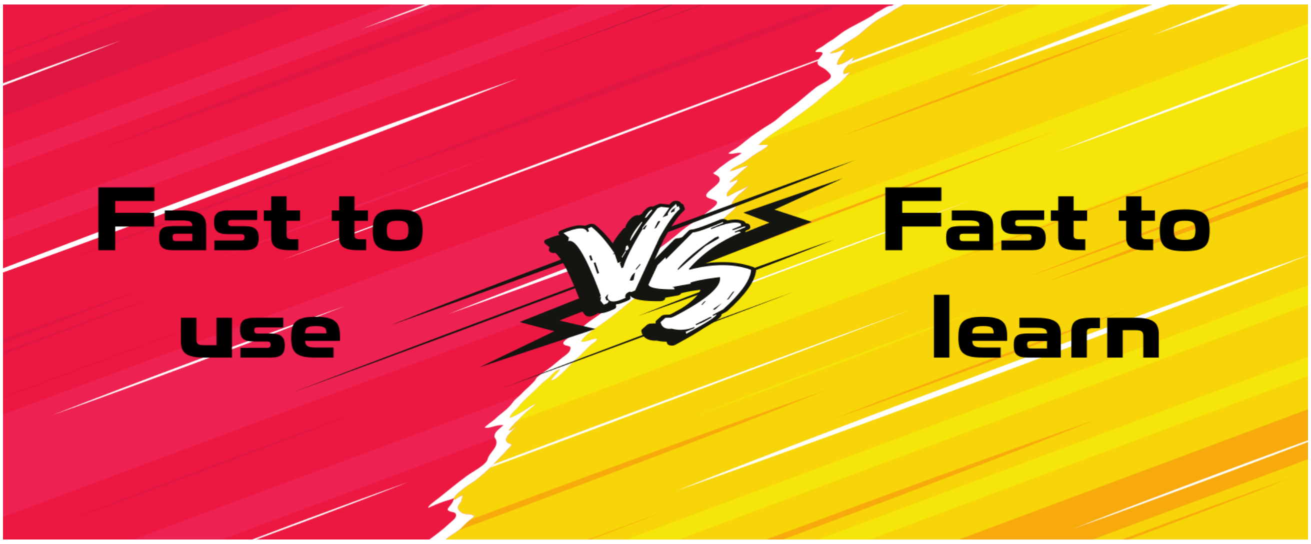
Part of the programmers' motivation momentum have to do with TripleO's open-source nature, so if you code in TripleO you are part of a community.

Congratulations! As a TripleO user or a TripleOer, you are a part of our community and it means that you're joining a diverse group that spans all age ranges, ethnicities, professional backgrounds, and parts of the globe. We are a passionate bunch of crazy people, proud of this "little" monster and more than willing to help others enjoy using it as much as we do.

Getting to know the interface (the templates, Mistral, Heat, Ansible, Docker, Puppet, Jinja, …) and how all components are tight together, probably is one of the most daunting aspects of TripleO for newcomers (and not newcomers). This for sure will raise the blood pressure of some of you who tried using TripleO in the past, but failed miserably and gave up in frustration when it did not behave as expected. Yeah.. sometimes that "$h1t" happens…

Although learning TripleO isn't that easy, the architecture updates, the decoupling of the role services "compostable roles", the backup and restore strategies, the integration of Ansible among many others have made great strides toward alleviating that frustration, and the improvements continue through to today.

So this is the question…



Is TripleO meant to be "fast to use" or "fast to learn"?

There is a significant way of describing software products, but we need to know what our software will be used for… TripleO is designed to work at scale, it might be easier to deploy manually a few controllers and computes, but what about deploying 100 computes, 3 controllers and 50 cinder nodes, all of them configured to be integrated and work as one single "cloud"? Buum!. So there we find the TripleO benefits if we want to make it scale we need to make it fast to use…

This means that we will find several customizations, hacks, workarounds, to make it work as we need it.

The upside to this approach is that TripleO evolved to be super-ultra-giga customizable so operators are enabled to produce great environments blazingly fast..

The downside, Jaja, yes.. there is a downside "or several". As with most things that are customized, TripleO became somewhat difficult for new people to understand. Also, it's incredibly hard to test all the possible deployments, and when a user does non-standard or not supported customizations, the upgrades are not as intuitive as they need…

This trade-off is what I mean when I say "fast to use versus fast to learn." You can be extremely productive with TripleO after you understand how it thinks "yes, it thinks".

However, your first few deployments and patches might be arduous. Of course, alleviating that potential pain is what our work is about. IMHO the pros are more than the cons and once you find a niche to improve it will be a really nice experience.

Also, we have the TripleO YouTube channel a place to push video tutorials and deep dive sessions driven by the community for the community.

For the Spanish community we have a 100% translated TripleO UI, go to https://translate.openstack.org and help us to reach as many languages as possible!!!

www.anstack.com was born on July 5th of 2016 (first GitHub commit), yeah is my way of expressing my gratitude to the community doing some CtrlC + CtrlV recipes to avoid the frustration of working with TripleO and not having something deployed and easy to be used ASAP.

Anstack does not have much traffic but it reached superuser.openstack.org, the TripleO cheatsheets were on devconf.cz and FOSDEM, so in general, is really nice. When people reference your writings anywhere. Maybe in the future can evolve to be more related to ANsible and openSTACK ;) as TripleO is adding more and more support for Ansible.

What about Red Hat? Yeahp, I have a long time speaking about the project but haven't spoken about the company making it all real. Red Hat is the world's leading provider of open source solutions, using a community-powered approach to provide reliable and high-performing cloud, virtualization, storage, Linux, and middleware technologies.

There is a strong feeling of belonging in Red Hat, you are part of a team, a culture and you are able to find a perfect balance between your work and life. Also, having all people from all over the globe makes a perfect place for sharing ideas and collaborate. Not all of it is good, i.e. Working mostly remotely in upstream communities can be really hard to manage if you are not 100% sure about the tasks that need to be done.

Keep rocking and become part of the TripleO community!

by Carlos Camacho at March 01, 2018 12:00 AM (https://www.anstack.com/blog/2018/03/01/2nd-birthday-as-a-red-hatter.html)

February 28, 2018

Adam Young (http://adam.younglogic.com)

OpenStack Role Assignment Inheritance for CloudForms (http://adam.younglogic.com/2018/02/openstack-hmt-cloudforms/)

Operators expect to use CloudForms to perform administrative tasks. For this reason, the documentation for OpenStack states that the Keystone user must have an 'admin' role. We found at least one case, however, where this was not sufficient. Fortunately, we have a better approach, and one that can lead to success in a wider array of deployments.

Table of contents

Background

CloudForms uses the role assignments for the give user account to enumerate the set of projects. Internally it creates a representation of these projects to be used to track resources. However, The way that 'admin' is defined on OpenStack is tied to a single project. This means that CloudForms really has no way to ask "what projects can this user manage?" Now, while admin anywhere is admin everywhere (https://bugs.launchpad.net/keystone/+bug/968696) so you would not think that you need to enumeration projects, but it turns out that some of the more complex operations, such as mounting a volume, has to cross service boundaries, and need the project abstraction to link the sets of operations. CloudForms design did not see this disconnect, and so some of those operations fail.

Lets assume, for the moment, that a user had to have a role on project in order to perform operations on that project. The current admin-everywhere approach would break. What CloudForms would require is an automated way to give a user a role on a project as soon as that project was created. It turns out that CloudForms is not the only thing that has this requirement.

Role Assignment Inheritance

Keystone projects do not have to be organized as a flat collection. They can be nested into a tree form. This is called "Hierarchical Multitenancy." Added to that, a role can be assigned to a user or group on parent project and that role assignment is inherited down the tree. This is called "Role Assignment Inheritance."

This presentation, while old, does a great job of putting the details together. (https://www.openstack.org/assets/presentation-media/Flat-no-more-Hierarchical-multitenancy-and-projects-acting-as-domains-in-OpenStack.pdf)

You don't need to do anything different in your project setup to take advantage of this mechanism. Here's something that is subtle: a Domain IS A project. Every project is already in a domain, and thus has a parent project. Thus, you can assign a user a role on the domain-as-a-project, and they will have that role on every project inside that domain.

Sample Code

Here is in command line form.

```
openstack role add --user CloudAdmin --user-domain Default --project Default --project-domain Default --inherited admin
```

Lets take those arguments step by step.

```
--user CloudAdmin  --user-domain Default
```

This is the user that CloudForms is using to connect to Keystone and OpenStack. Every user is owned by a domain, and this user is owned by the Default" domain.

```
--project Default --project-domain Default
```

This is blackest of magic. The Default domain IS-A project. So it owns itself.

```
--inherited
```

A role assignment is either on a project OR on all its subprojects. So, the user does not actually have a role that is usable against the Default DOMAIN-AS-A-PROJECT, but only on all odf the subordinate projects. This might seem strange, but it was built this way for exactly this reason: being able to distinguish between levels of a hierarchy.

```
admin
```

This is the role name.

Conclusion

With this role assignment, the CloudForms Management Engine instance can perform all operations on all projects within the default domain. If you add another domain to manage a separate set of projects, you would need to perform this same role assignment on the new domain as well.

I assume this is going to leave people with a lot of questions. Please leave comments, and I will try to update this with any major concepts that people want made lucid.

by Adam Young at February 28, 2018 07:04 PM (http://adam.younglogic.com/2018/02/openstack-hmt-cloudforms/)

OpenStack Superuser (http://superuser.openstack.org)

A close-up on the Queens release (http://superuser.openstack.org/articles/queens-release-faq/)

As OpenStack's 17th release comes to a cloud near you, Superuser answers some common questions about Queens and the ecosystem. Bonus answer, in case you were wondering: the name comes from Queens Park, a suburb of Sydney, Australia. (More on the release naming process here. (https://governance.openstack.org/tc/reference/release-naming.html))

What are the main trends in user adoption?

- People want their clouds to do more than they used to, expanding to new workloads including machine learning and edge.
- Integration is key. There's a growing demand for large enterprises to run mission critical and backup as apps in a cloud environment, so the Queens release has a lot of high availability features, there's a new project called Masakari (https://wiki.openstack.org/wiki/Masakari) and multi-attach in Cinder block storage (http://superuser.openstack.org/articles/meet-volume-multi-attach-great-new-feature-openstack-queens/). Multi-attach allows you to have more than one virtual machine attaching to the same volume, so if one goes down you can rely on another VM accessing the same storage.

What's the current situation with GPU capabilities in OpenStack and where's it headed?

People running GPUs today generally do it two ways:

- PCI passthrough (https://wiki.openstack.org/wiki/Pci_passthrough) (the downside — it requires post-provisioning manual setup so it can be time consuming)
- Using Ironic (https://www.openstack.org/software/releases/ocata/components/ironic) to manage the entire physical server

Early users of the underlying hardware include eBay and Commonwealth Bank. eBay is using Ironic on top of hardware layering TensorFlow (https://www.tensorflow.org/) and Kubernetes on top of the GPUs. Once these components are getting delivered as fully supported technologies inside OpenStack, it's a pretty good indication the use case will go mainstream across a lot of different environments and organizations — and now it's much easier to get started.

New for the Queens release is support for virtual GPUs. Nova now lets cloud administrators define flavors to request vGPU resources and set vGPU resolutions. The feature currently supports Intel GVT-g and NVIDIA GRID vGPU. See the Nova documentation (https://docs.openstack.org/nova/latest/admin/virtual-gpu.html) for additional details.

What enhancements in Queens are the most important for edge computing?

The three most relevant things for edge are OpenStack-Helm (https://wiki.openstack.org/wiki/Openstack-helm), LOCI (https://wiki.openstack.org/wiki/LOCI) and acceleration support, which will be a requirement in a lot of edge environments.

Helm containerizes OpenStack services and puts them in a Kubernetes pod. (You can see how this works with a live upgrade demoed (https://www.google.com/url?q=https://www.openstack.org/videos/sydney-2017/at-and-ts-strategy-for-implementing-a-next-generation-openstack-cloud&sa=D&ust=1519741338270000&usg=AFQjCNGWFIwcJy7KOK6ZC1xtKlGFx3Q-Yg) by AT&T at the Sydney Summit).

LOCI is a set of lightweight container images, the project takes a thinner approach to the container – offering a smaller, much more portable, lower footprint. Then operators can add configuration to the outside and use Kubernetes to manage the overall service functionality.

Together they make it easier to deploy a lot of OpenStack environments and to upgrade in an automated way as well as operate in a zero-touch way over the long term and with a smaller footprint than a traditional datacenter OpenStack deployment. Those attributes will all be very important in edge deployments.

There are a lot of existing projects for OpenStack and containers; where does OpenStack-Helm fit in?

OpenStack-Helm (https://wiki.openstack.org/wiki/Openstack-helm) provides a collection of Helm charts and tools for managing the lifecycle of OpenStack on top of Kubernetes and running OpenStack projects as independent services. It's a promising project for users who want to put OpenStack services at the edge, or who want to containerize OpenStack services for easier upgrade paths.

Which end users also contributed code to Queens?

A total of 158 companies and organizations contributed (http://stackalytics.com/) to this release; Some end users who also contributed code include:

- AT&T
- Verizon
- Walmart
- Universidade Federal de Campina Grande
- Boston University
- China Mobile
- Deutsche Telekom
- MIT
- Johns Hopkins University Applied Physics Laboratory
- HFBank
- Workday
- University of Melbourne

What's the main direction for the next release, Rocky?

This week, the development teams are in Dublin at the PTG (https://www.openstack.org/ptg/) to start working on the Rocky release. Although the 18th release of OpenStack is not due until the end of August, there are already a few planned features to get excited about:

- Fast forward upgrade work will continue, which is a method for letting users jump more than two releases ahead with alternative install paths that help users speed their way through the intermediary releases and get to where they want to be.
- Minimum bandwidth and bandwidth-based scheduling is a networking feature that may be coming, which will be of particular interest to NFV and cloud service providers for things like ensuring a minimum level of performance for streaming services.
- One of the community wide goals for Rocky is to enable mutable configuration across services, which will let operators change configuration settings without restarting a service.

Get involved!

Learn more about how organizations are using OpenStack at the upcoming Summit (https://www.openstack.org/summit/vancouver-2018/), meet the community (https://www.openstack.org/community/) and start contributing, or visit the Marketplace (https://www.openstack.org/marketplace/) to find an OpenStack service provider.

Cover Photo (http://superuser.openstack.org/feed/"https://www.flickr.com/photos/wwarby/36216717556/") // CC BY NC (https://creativecommons.org/licenses/by-nc/2.0/)

The post A close-up on the Queens release (http://superuser.openstack.org/articles/queens-release-faq/) appeared first on Superuser (http://superuser.openstack.org).

by Nicole Martinelli at February 28, 2018 05:16 PM (http://superuser.openstack.org/articles/queens-release-faq/)

Opensource.com (https://opensource.com/taxonomy/term/5126/feed/feed)

Celebrate the Queens release with 5 new OpenStack tips and guides (https://opensource.com/article/18/2/openstack-tutorials)

Today marks the release of OpenStack's seventeenth release, Queens (https://releases.openstack.org/queens/). After a 26-week release schedule, Queens brings into the fold new projects and new features, including strong container integration, support for vGPUs, and many advancements around NFV, edge computing, and machine learning applications.

by Jason Baker at February 28, 2018 09:00 AM (https://opensource.com/article/18/2/openstack-tutorials)

February 27, 2018

OpenStack Superuser (http://superuser.openstack.org)

Meet volume multi-attach, a great new feature in OpenStack Queens (http://superuser.openstack.org/articles/meet-volume-multi-attach-great-new-feature-openstack-queens/)

Volume multi-attach is a feature that enables users to attach and access a single block storage volume to/from multiple servers.
The use cases for this functionality include active-active and hot-standby scenarios, when the data on a volume needs to be accessible by multiple servers in order to recover fast in case of failures or being able to handle increased load on the system. Volume multi-attach requires support from two OpenStack projects, Cinder (https://www.openstack.org/software/releases/ocata/components/cinder) (block storage) and Nova (https://www.openstack.org/software/releases/ocata/components/nova)(compute.)

The support for it added in the Queens release (https://releases.openstack.org/queens/index.html) (February 28, 2018) represents years of collaboration, so it's with great pride that we introduce you to it here.

It's not cloudy anyway

The question often arises whether or not we should add features to the cloud platform that we don't consider a good fit. It's even more common for functionalities that are more challenging to add due to previous architectural decisions. Multi-attach was no different.

While the desire for cloud native applications is industry-wide, it's quite a luxury to rewrite everything to fit. In order to support this transformation period, we needed to make the platform suitable for workloads that are not fully adapted yet.

In the past couple of release cycles, while working to overcome the challenges that prevented us from implementing the volume multi-attach functionality in Cinder and Nova, we had weekly meetings to decide on the design of new APIs and further improvements to the platform.

So, what do you get?

As a starting point, we added (https://developer.openstack.org/api-ref/block-storage/v3/index.html#attachments) a new attach/detach API to Cinder to make the flows more flexible. The new API introduces the concept of 'attachment' which you can create, update, complete or delete in order to attach or detach a volume.

You can use the new attach/detach API to attach a volume multiple times in case the Cinder back end you chose reports support for the functionality. In the Queens release, three drivers support multi-attach: LVM, NetApp/SolidFire and Oracle ZFSSA. You can check the Driver Support Matrix (https://wiki.openstack.org/wiki/CinderSupportMatrix) for updates on when additional drivers add support.

You need to create a multi-attach volume type to create a multi-attach-capable volume to be able to use this functionality. You can also re-type a volume to be non-multi-attach-capable or vice-versa, but only when the volume is in 'available' state. Using the 'multiattach' flag during the volume create operation is **deprecated** starting from the Queens release.

On the compute side, in order to attach a single volume to multiple VM instances you'll need to run Nova microversion 2.60 or higher. Similarly, to backends in Cinder you'll find limitations regarding hypervisors in Nova as well. In the Queens release only Libvirt is supporting volume multi-attach and you'll need either Libvirt >= 3.10 or QEMU < 2.10. For further information please refer to the Feature Support Matrix. (https://docs.openstack.org/nova/latest/user/support-matrix.html)

Currently all the subsequent volume attachments will be attached in read-write (RW) mode including the boot from volume case. You can turn the functionality on or off with two new Cinder policies:

```
volume:multiattach
volume:multiattach_bootable_volume
```

You can read more about the functionality and its limitations in the admin guide chapters of Cinder (https://docs.openstack.org/cinder/latest/admin/blockstorage-volume-multiattach.html) and Nova (https://docs.openstack.org/nova/latest/admin/manage-volumes.html#volume-multi-attach).

Beyond new APIs and functionality, we've also started to cover the testing side not just within the projects, but also Tempest. We've added test cases like the basic functionality, booting a server from a multi-attach volume, or resizing servers with a multi-attach volume attached to both.

Check out this overview video with a demo from Matt Riedemann:

What's next ?

The two teams are looking for feedback and discussing improvements to the functionality. Current improvement ideas to discuss include providing support to specify the attach-mode for the secondary attachments. For the latest information, please take a look into Cinder's specs list for the current release. (https://specs.openstack.org/openstack/cinder-specs/index.html)

For further improvements in the area of testing, we are considering adding more robust integration testing such as writing and reading to/from a shared volume between two VMs.

If you'd like to get involved or have feedback, you can reach the teams on the OpenStack Developer Mailing List (http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack-dev), please use the tags '[cinder]' and '[nova]' in the subject line to ensure it gets the right people's attention. You can also reach out on IRC (https://wiki.openstack.org/wiki/IRC), each teams has its own channel: #openstack-cinder and #openstack-nova.

Special thanks to

This long-awaited feature wouldn't have landed without the hard work of the two teams and especially the following contributors: John Griffith, Matt Riedemann, Steve Noyes, John Garbutt, Walter Boring, Balazs Gibizer, Sylvain Bauza, Melanie Witt, Sean McGinnis and Jay Bryant.

The post Meet volume multi-attach, a great new feature in OpenStack Queens (http://superuser.openstack.org/articles/meet-volume-multi-attach-great-new-feature-openstack-queens/) appeared first on Superuser (http://superuser.openstack.org).

by Ildiko Vancsa at February 27, 2018 04:58 PM (http://superuser.openstack.org/articles/meet-volume-multi-attach-great-new-feature-openstack-queens/)

SWITCH Cloud Blog (https://cloudblog.switch.ch)

Openstack Keystone authentication for your Kubernetes cluster (https://cloudblog.switch.ch/2018/02/02/openstack-keystone-authentication-for-your-kubernetes-cluster/)

At SWITCH we are looking to provide a container platform as a Service solution. We are working on Kubernetes and Openshift to gauge what is possible and how a service could be structured. It would be really nice to use the existing Openstack username and password to authenticate to Kubernetes. We tested this solution and it works great.

How does it work ? Lets start from the client side.

Kubernetes users use the kubectl client to access the cluster. The good news is that since version v1.8.0 of the client, kubectl is able to read the usual openstack env variables, contact keystone to request a token, and forward the request to the kubernetes cluster using the token. This was merged the 7th of August 2017 (https://github.com/kubernetes/kubernetes/pull/39587). I could not find anywhere how to correctly configure the client to use this functionality. Finally I wrote some documentation notes HERE (https://github.com/dims/k8s-keystone-auth#new-kubectl-clients-v180-and-later).

How does it work on the Kubernetes master side ?

The Kubernetes API receives a request with a keystone token. In the Kubernetes language this is a Bearer Token (https://kubernetes.io/docs/admin/authentication/#static-token-file). To verify the keystone token the Kubernetes API server will use a WebHook (https://kubernetes.io/docs/admin/authentication/#webhook-token-authentication). What does it means ? That the Kubernetes API will contact yet another Kubernetes component that is able to authenticate the keystone token.

The k8s-keystone-auth (https://github.com/dims/k8s-keystone-auth) component developed by Dims (https://www.linkedin.com/in/davanum/) makes exactly this. I tested his code and I created a Docker container (https://github.com/dims/k8s-keystone-auth/blob/master/kube-system/Dockerfile) to integrate the k8s-keystone-auth in my kube-system namespace (https://github.com/dims/k8s-keystone-auth/blob/master/kube-system/k8s-keystone-auth.yaml). When you run the k8s-keystone-auth container your pass as an argument the URL of your keystone server.

If you are deploying your cluster with k8s-on-openstack you find this integration summarized in a single commit. (https://github.com/zioproto/k8s-on-openstack/commit/b966b1e0452a6857334c6881cba9bd7f6bc562f3)

Now that everything is setup I can try:

```
source ~/openstackcredentials
kubectl get pods
```

I will be correctly authenticated by keystone that will verify my identity, but I will have no authorization to do anything:

```
Error from server (Forbidden): pods is forbidden: User "saverio.proto@switch.ch" cannot list pods in the namespace "default"
```

This is because we need to set up some authorization for this keystone user. You can find detailed documentation about RBAC (https://kubernetes.io/docs/admin/authorization/rbac/) but I make here a simple example:

```
kubectl create rolebinding saverio-view --clusterrole view --user saverio.proto@switch.ch --namespace default
```

Now the my user is able to view anything at the default namespace, and I will be able to do kubectl get pods

Of course setting up RBAC specific rules for every user is not optimal. You can at least use the keystone projects, that are mapped to `kind: Group` in Kubernetes. Here an example:

```
---
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
 name: read-pods
 namespace: default
subjects:
- kind: Group
 name: <openstack_project_uuid>
 apiGroup: rbac.authorization.k8s.io
roleRef:
 kind: Role
 name: pod-reader
 apiGroup: rbac.authorization.k8s.io
```

You can then achieve a “soft multitenancy” where every user belonging to a specific keystone project is limited with permissions to a specific namespace. I talk about soft multitenancy because all the pods from all the namespaces, depending on your networking solution, could end up on the same network with a completely open policy.

I would like to thank Dims and the other people on the Slack channel #sig-openstack for the great help while developing this Kubernetes deployment.

by Saverio Proto at February 27, 2018 09:59 AM (https://cloudblog.switch.ch/2018/02/02/openstack-keystone-authentication-for-your-kubernetes-cluster/)

Deploy Kubernetes v1.8.3 on Openstack with native Neutron networking (https://cloudblog.switch.ch/2017/11/15/deploy-kubernetes-v1-8-3-on-openstack-with-native-neutron-networking/)

Hello,
I wrote in the past how to deploy Kubernetes on SWITCHengines (Openstack) using this ansible playbook. (https://github.com/infraly/k8s-on-openstack) When I wrote that article, I did not care about the networking setup, and I used the proposed weavenet plugin. I went to Sydney at the Openstack Summit and I saw the great presentation from Angus Lees (https://www.youtube.com/watch?v=cbAISC6uHcw&t=4s). It was the right time to see the presentation because I recently watched this video (https://www.youtube.com/watch?v=y2bhV81MfKQ) where they explain the networking of Kubernetes when running on GCE. Going back to Openstack, Angus mentioned that the Kubernetes master can talk to neutron, to inject routes in the tenant router to provide connectivity without NAT among the pods that live in different instances. This would make easier the troubleshooting, and would leave MTU 1500 between the pods.

It looked very easy, just use:

```
--network-plugin=kubenet
```

and specify in the cloud config the router uuid.

Our first tests with version 1.7.0 did not work. First of all I had to fix the Kubernetes documentation (https://github.com/kubernetes/website/pull/6257/files), because the syntax to specify the router UUID was wrong. Then I had a problem with Security groups disappearing from the instances. After troubleshooting and asking for help on the Kubernetes slack channel, I found out that I was hitting a gophercloud known bug (https://github.com/gophercloud/gophercloud/commit/96ac26acb8d1e39587402b27c66834c41d63f3cc).

The bug was already fixed in gophercloud at the time of my finding, but I learned that Kubernetes freezes an older version of this library in the folder “vendor/github.com/gophercloud/gophercloud”. So the only way to get the updated library version was to upgrade to Kubernetes v1.8.0, or any newer version including this commit. (https://github.com/kubernetes/kubernetes/commit/74a3d89ad4ff1508a16bacb0c333a1f795c31470)

After a bit of testing every works now. The changes are summarised in this PR (https://github.com/infraly/k8s-on-openstack/pull/8), or you can just use the master branch from my git repository (https://github.com/zioproto/k8s-on-openstack).

After you deploy, the K8s master will assign from network ClusterCIDR (usually a /16 address space) a smaller /24 subnet per each Openstack instance. The Pods will get addresses from the subnet assigned to the instance. The kubernetes master will inject static routes to the neutron router, to be able to route packets to the Pods. It will also configure the neutron ports of the instances with the correct allowed_address_pairs value, so that the traffic is not dropped by the Openstack antispoofing rules.

This is what a show of the Openstack router looks like:

```
$ openstack router show b11216cb-a725-4006-9a55-7853d66e5894 -c routes
+--------+--------------------------------------------------+
| Field  | Value                                            |
+--------+--------------------------------------------------+
| routes | destination='10.96.0.0/24', gateway='10.8.10.3'  |
|        | destination='10.96.1.0/24', gateway='10.8.10.8'  |
|        | destination='10.96.2.0/24', gateway='10.8.10.11' |
|        | destination='10.96.3.0/24', gateway='10.8.10.10' |
+--------+--------------------------------------------------+
```

And this is what the allowed_address_pairs on the port of one instance looks like:

```
$ openstack port show 42f2a063-a316-4fe2-808c-cd2d4ed6592f -c allowed_address_pairs
+-----------------------+--------------------------------------------------------------+
| Field                 | Value                                                        |
+-----------------------+--------------------------------------------------------------+
| allowed_address_pairs | ip_address='10.96.1.0/24', mac_address='fa:16:3e:3e:34:2c'  |
+-----------------------+--------------------------------------------------------------+
```

There is of course more work to be done.

I will improve the ansible playbook to create automatically the Openstack router and network, at the moment these steps are done manually before starting the playbook.

Working with network-plugin=kubenet is actually deprecated, so I have to understand what is the long term plan for this way of deployment.

The Kubernetes master is still running on a single VM, the playbook can be extended to have an HA setup.

I really would like to have feedback from users of Kubernetes on Openstack. If you use this playbook please let me know, and if you improve it, the Pull Requests on github are very welcome! 🙂

by Saverio Proto at February 27, 2018 09:58 AM (https://cloudblog.switch.ch/2017/11/15/deploy-kubernetes-v1-8-3-on-openstack-with-native-neutron-networking/)

February 26, 2018

OpenStack Superuser (http://superuser.openstack.org)

Why cyborgs need free software (http://superuser.openstack.org/articles/medical-devices-free-software/)

There probably aren't many questions from patients that leave electrophysiologists speechless. Karen Sandler stumped hers with just one. Sandler, executive director at the Software Freedom Conservancy (https://sfconservancy.org), wanted to know about the software powering the pacemaker that she needed.

Sandler has an outsize heart (three times the size of the average ticker), a condition called hypertrophic cardiomyopathy (https://en.wikipedia.org/wiki/Hypertrophic_cardiomyopathy). For her, the condition is without symptoms but she runs an increased chance of sudden death with each year that passes.

"I was told that everything would be fine, because I could get this defibrillator implanted in my body. If I get into trouble it'll be like a fairy godmother shocking me," Sandler recounted in a recent Index Conference keynote (https://www.youtube.com/watch?v=e3vKE7B2ahY). The specialist was showing Sandler and her mother the small, lightweight device that some patients refer to as "internal bling."

"I had just one question for him: what does it run on?"

Sandler, a lawyer by profession, said the doctor didn't understand the question. Then he called in a representative for the device, who also didn't know anything about the software inside. After the doctor's visit, Sandler dogged the company's help line, a scenario she describes as "phone tree hell" but found that none of the manufacturers had answers for her.

"I did what any normal person would do, I put it off," she says. She changed her mind, however, after both her mother and a close friend admitted to fearing Sandler was dead every time they got her voicemail.

So Sandler got the implant – embracing her "inner cyborg (https://twitter.com/hashtag/CyborgLawyer?src=hash)" she says while flashing a slide of "Star Trek: Voyager" character Seven of Nine (https://en.wikipedia.org/wiki/Seven_of_Nine) – and embarked on a research project about the proprietary software inside medical devices.

> haha "I believe that every Free Software advocate that still uses the story of Stallman's Printer instead of that the "Large-Hearted Queen of Open Source code" Karen Sandler… should be sentenced to only use unpatched Windows. 95, not the current version" https://t.co/5AMpDHwFaz (https://t.co/5AMpDHwFaz)
>
> — karen sandler (@o0karen0o) January 22, 2018 (https://twitter.com/o0karen0o/status/955418222571929605?ref_src=twsrc%5Etfw)

The results? "Software has bugs. Yeah, surprise!" she notes, demonstrating exactly zero surprise. "These devices are totally vulnerable," Sandler concludes. Unsurprisingly, she found that free and open software was less vulnerable than proprietary systems for medical devices. The Four Freedoms (https://en.wikipedia.org/wiki/The_Free_Software_Definition) — to use, study, share and improve the software – not only lead to fewer bugs but help wipe out the remaining ones. (More on this in a paper from the Software Freedom Law Center titled "Killed with Code.") (https://www.softwarefreedom.org/resources/2010/transparent-medical-devices.html)

"That means that if you face a problem, probably other people do too and you can share those changes with others," Sandler says. Patients don't have to rely on any particular company, which is really important when it comes to things like your heart.

"Right now what we have in in the medical device space is the worst of both worlds," she says. "These devices are broadcasting remote wirelessly (https://www.theatlantic.com/technology/archive/2018/01/my-pacemaker-is-tracking-me-from-inside-my-body/551681/)all the time, they have radio telemetry on them — but no real security and there's no basic password protection."

Image
(http://superuser.openstack.org/feed/"https://www.flickr.com/photos/cnast/5619800224")
// CC BY NC

(https://creativecommons.org/licenses/by-nc/2.0/)

Sandler had a terrifying experience with the standard issue programming of the device going wrong when she was pregnant – her slightly elevated heart rate caused it to shock her. Because as a pregnant woman she was not the typical for the device, the doctor's solution was medication to slow down her heart rate. It could've been worse: there were 16,000 deaths associated with medical devices in the FDA database (https://nypost.com/2017/12/16/your-medical-implant-could-be-killing-you/). But with the devices as black boxes, it's hard to diagnose where the devices went wrong.

"I would have liked to have seen the source code to my device — even if I got all the pregnant women with defibrillators together, there's not enough money to get those to get the device manufacturer to consider our case," she says. It's a perfect example of a company's good intentions leading to dangerous situations for some.

Sandler, a one-time "Mac fan girl" who previously worked at the Gnome Foundation (https://www.gnome.org/foundation/) and has been a proponent of FOSS since the 90s, says that now her interest in it has taken on a new dimension.

"Now I'm someone who thinks that software freedom is essential, not this esoteric issue," she says. "It's in fact something that is going to undermine the safety and the very nature of our society."

Still, she considers it a privilege to be what she calls a voluntary cyborg.

"The truth is we will all become cyborgs over time," she says. "Because I want to incorporate enough technology in my body to take advantage of advances as they go forward."

"So I say cyborgs unite!...We're at the beginning of a road with free and open source software and with it, we have options."

Catch the entire 43-minute session below.

Hat tip: Monty Taylor (https://twitter.com/e_monty/status/967121732510343168)

Cover Photo (http://superuser.openstack.org/feed/"https://www.flickr.com/photos/s3a/10583291614/") // CC BY NC (https://creativecommons.org/licenses/by-nc/2.0/)

The post Why cyborgs need free software (http://superuser.openstack.org/articles/medical-devices-free-software/) appeared first on OpenStack Superuser (http://superuser.openstack.org).

by Nicole Martinelli at February 26, 2018 05:37 PM (http://superuser.openstack.org/articles/medical-devices-free-software/)

Lars Kellogg-Stedman (http://blog.oddbit.com/)

Grouping aggregation queries in Gnocchi 4.0.x (http://blog.oddbit.com/2018/02/26/grouping-aggregation-queries-i/)

In this article, we're going to ask Gnocchi (the OpenStack telemetry storage service) how much memory was used, on average, over the course of each day by each project in an OpenStack environment.

Environment

I'm working with an OpenStack "Pike" deployment, which means I have Gnocchi 4.0.x. More ...

by Lars Kellogg-Stedman at February 26, 2018 05:00 AM (http://blog.oddbit.com/2018/02/26/grouping-aggregation-queries-i/)

February 24, 2018

OpenStack in Production (http://openstack-in-production.blogspot.com/)

Maximizing resource utilization with Preemptible Instances (http://openstack-in-production.blogspot.com/2018/02/maximizing-resource-utilization-with.html)

Motivation

The CERN cloud consists of around 8,500 hypervisors providing over 36,000 virtual machines. These provide the compute resources for both the laboratory's physics program but also for the organisation's administrative operations such as paying bills and reserving rooms at the hostel.

The resources themselves are generally ordered once to twice a year with servers being kept for around 5 years. Within the CERN budget, the resource planning teams looks at:

- The needs of the physics program for the coming years under the review of the Computing Scrutiny Review Board (http://wlcg.web.cern.ch/collaboration/management/computing-resources-scrutiny-group) and the LHC Experiments Committee (http://committees.web.cern.ch/committees/lhcc/).
- The resources required to run the computing services requirements for the CERN laboratory. These are projected using capacity planning trend data and upcoming projects such as video conferencing.

With the installation and commissioning of thousands of servers concurrently (along with their associated decommissioning 5 years later), there are scenarios to exploit underutilised servers. Programs such as LHC@Home (http://lhcathome.web.cern.ch/) are used but we have also been interested to expand the cloud to provide virtual machine instances which can be rapidly terminated in the event of

- Resources being required for IT services as they scale out for events such as a large scale web cast on a popular topic or to provision instances for a new version of an application.
- Partially full hypervisors where the last remaining cores are not being requested (the Tetris problem (http://openstack-in-production.blogspot.ch/2014/07/openstack-plays-tetris-stacking-and.html)).
- Compute servers at the end of their lifetime which are used to the full before being removed from the computer centre to make room for new deliveries which are more efficient and in warranty.

The characteristics of this workload is that it should be possible to stop an instance within a short time (a few minutes) compared to a traditional physics job.

Resource Management In Openstack

Operators use project quotas for ensuring the fair sharing of their infrastructure. The problem with this, is that quotas pose as hard limits.This leads to actually dedicating resources for workloads even if they are not used all the time or to situations where resources are not available even though there is quota still to use.

At the same time, the demand for cloud resources is increasing rapidly. Since there is no cloud with infinite capabilities, operators need a way to optimize the resource utilization before proceeding to the expansion of their infrastructure.

Resources in idle state can occur, showing lower cloud utilization than the full potential of the acquired equipment while the users' requirements are growing.

The concept of Preemptible Instances can be the solution to this problem. These type of servers can be spawned on top of the project's quota, making use of the underutilised capabilities. When the resources are requested by tasks with higher priority (such as approved quota), the preemptible instances are terminated to make space for the new VM.

Preemptible Instances with Openstack

Supporting preemptible instances, would mirror the AWS Spot Market and the Google Preemptible Instances. There are multiple things to be addressed here as part of an implementation with OpenStack, but the most important can be reduced to these:

1. Tagging Servers as Preemptible

In order to be able to distinguish between preemptible and non-preemptible servers, there is the need to tag the instances at creation time. This property should be immutable for the lifetime of the servers.

2. Who gets to use preemptible instances

There is also the need to limit which user/project is allowed to use preemptible instances. An operator should be able to choose which users are allowed to spawn this type of VMs.

3. Selecting servers to be terminated

Considering that the preemptible instances can be scattered across the different cells/availability zones/aggregates, there has to be "someone" able to find the existing instances, decide the way to free up the requested resources according to the operator's needs and, finally, terminate the appropriate VMs.

4. Quota on top of project's quota

In order to avoid possible misuse, there could to be a way to control the amount of preemptible resources that each user/project can use. This means that apart from the quota for the standard resource classes, there could be a way to enforce quotas on the preemptible resources too.

OPIE : IFCA and Indigo Dataclouds

In 2014, there were the first investigations into approaches by Alvaro Lopez from IFCA (https://blueprints.launchpad.net/nova/+spec/preemptible-instances (https://blueprints.launchpad.net/nova/+spec/preemptible-instances)).
As part of the EU Indigo Datacloud (https://www.indigo-datacloud.eu/) project, this led to the development of the OpenStack Pre-Emptible Instances package (https://github.com/indigo-dc/opie (https://github.com/indigo-dc/opie)).
This was written up in a paper to Journal of Physics: Conference Series (http://iopscience.iop.org/article/10.1088/1742-6596/898/9/092010/pdf (http://iopscience.iop.org/article/10.1088/1742-6596/898/9/092010/pdf)) and presented at the OpenStack summit (https://www.youtube.com/watch?v=eo5tQ1s9ZxM (https://www.youtube.com/watch?v=eo5tQ1s9ZxM))

Prototype Reaper Service

At the OpenStack Forum during a recent OpenStack summit, a detailed discussion took place on how spot instances could be implemented without significant changes to Nova. The ideas were then followed up with the OpenStack Scientific Special Interest Group (https://wiki.openstack.org/wiki/Scientific_SIG).

Trying to address the different aspects of the problem, we are currently prototyping a "Reaper" service. This service acts as an orchestrator for preemptible instances. It's sole purpose is to decide the way to free up the preemptible resources when they are requested for another task.

The reason for implementing this prototype, is mainly to help us identify possible changes that are needed in Nova codebase to support Preemptible Instances.

More on this WIP can be found here:
https://gitlab.cern.ch/ttsiouts/ReaperServicePrototype (https://gitlab.cern.ch/ttsiouts/ReaperServicePrototype)

Summary

The concept of Preemptible Instances gives operators the ability to provide a more "elastic" capacity. At the same time, it enables the handling of increased demand for resources, with the same infrastructure, by maximizing the cloud utilization.

This type of servers is perfect for tasks/apps that can be terminated at any time, enabling the users to take advantage of extra cpu power on demand without the fixed limits that quotas enforce.

Finally, here in CERN, there is an ongoing effort to provide a prototype orchestrator for Preemptible Servers with Openstack, in order to pinpoint the changes needed in Nova to support this feature optimally. This could also be available in future for other OpenStack clouds in use by CERN such as the T-Systems Open Telekom Cloud through the Helix Nebula Open Science Cloud project.

Contributors

- Theodoros Tsioutsias (CERN openlab fellow working on Huawei collaboration)
- Spyridon Trigazis (CERN)
- Belmiro Moreira (CERN)

References

- CERN Huawei openlab collaboration at http://openlab.cern/about/industry_members/huawei (http://openlab.cern/about/industry_members/huawei)
- Helix Nebula Science Cloud project at http://www.helix-nebula.eu (http://www.helix-nebula.eu/)
- Indigo Datacloud at https://www.indigo-datacloud.eu/ (https://www.indigo-datacloud.eu/)
- CERN SKA collaboration plans at https://www.openstack.org/videos/sydney-2017/future-science-on-future-openstack-developing-next-generation-infrastructure-at-cern-and-ska (https://www.openstack.org/videos/sydney-2017/future-science-on-future-openstack-developing-next-generation-infrastructure-at-cern-and-ska)
- OpenStack Scientific Special Interest Group at https://wiki.openstack.org/wiki/Scientific_SIG (https://wiki.openstack.org/wiki/Scientific_SIG)
- Nova patch to support Reaper - https://review.openstack.org/547450 (https://review.openstack.org/547450)

by Theodoros Tsioutsias (noreply@blogger.com) at February 24, 2018 08:13 AM (http://openstack-in-production.blogspot.com/2018/02/maximizing-resource-utilization-with.html)

David Moreau Simard (https://dmsimard.com/categories/openstack/)

Awesome things in software engineering: open source (https://dmsimard.com/2018/02/24/awesome-things-in-software-engineering-open-source/)

This is part of a blog series highlighting awesome things in software engineering because not everything has to be depressing, about bugs, vulnerabilities, outages or deadlines. If you'd like to collaborate and write about awesome things in software engineering too, let's chat: reach out on Twitter or LinkedIn. What's this blog series about ? Between you and me, software engineering isn't always fun. You're not always working on what you like.

February 24, 2018 12:00 AM (https://dmsimard.com/2018/02/24/awesome-things-in-software-engineering-open-source/)

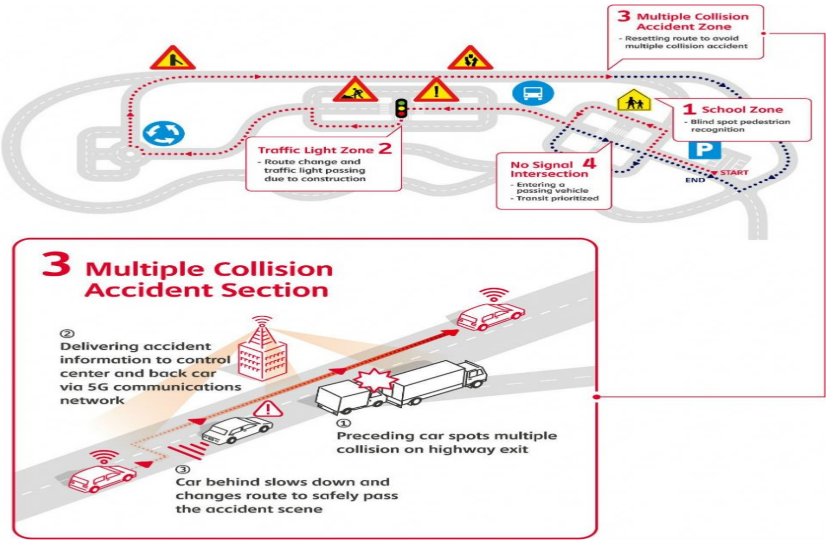
February 23, 2018

OpenStack Superuser (http://superuser.openstack.org)

5G and the really smart self-driving car (http://superuser.openstack.org/articles/5g-sktelecom-tests/)

On a regular weekday, some time in the future, you'll be heading to work like everyone else, comfortably chauffeured in a self-driving car. As you approach a school zone, your car and the cars next to you heed the speed limit but stop sharply as a child rushes out into the street.

That's what happened during tests at K-City (https://qz.com/1121372/south-korea-opens-k-city-the-latest-fake-town-built-for-self-driving-cars/), testing grounds for self-driving cars in Hwaseong, South Korea. The recent experiment (https://www.globalskt.com/home/info/2254) from SK telecom and the Korean Transportation Safety Authority (TS) ran two 5G self-driving cars successfully through what they're calling "cooperative driving," where smart-self driving cars constantly ping each other with updated traffic information.


3 Multiple Collision Accident Zone
- Resetting route to avoid multiple collision accident
1 School Zone
- Blind spot pedestrian recognition
Traffic Light Zone 2
- Route change and traffic light passing due to construction
No Signal 4 Intersection
- Entering a passing vehicle
- Transit prioritized
P
START
END
3 Multiple Collision Accident Section
② Delivering accident information to control center and back car via 5G communications network
① Preceding car spots multiple collision on highway exit
③ Car behind slows down and changes route to safely pass the accident scene

Cooperative driving in action on the K-City track. Courtesy SK telecom.

And while the cars in the tests were on a closed track and the "child" running out into the street was a life-size dummy, the experiment proves that the ultra-low-latency of less than 1ms of response time can handle everyday traffic situations. The incessant chatter — the car communicated more than 100 times a second using 5G network with the control center and with another vehicle — means that vehicles reroute for construction sites, gauge the right speed for freeway on-ramps and determine how best to merge traffic after an accident closes a lane.

An earlier test below shows the car whizzing at over 100 mph to demonstrate potential point-to-point data transmission speeds.

Putting software-defined networking in the driver's seat

SK telecom is the largest wireless provider in South Korea with a 50 percent market share. The company has been at the forefront of developing and commercializing advanced wireless technology — for example, it was the first to commercialize CDMA technology way back in 1990. And while 5G is said to have had its "coming out party" (https://www.fastcompany.com/40528802/the-winter-olympics-will-be-a-coming-out-party-for-5g-wireless) with a number of firms including SK telecom at the Winter Olympics, SKT is once again ahead of the curve.

Their team has been pioneering the network slicing and virtualization key to make 5G run smoothly. Back in 2017, they released TACO (T-All Container OpenStack) for SKT's 5G infrastructure. The team has also been developing software-defined networking (SDN) based network virtualization solution (SONA) for TACO. SONA uses VxLAN to provide fully independent virtual networks to 5G services and supports multiple gateways to provide multiple paths to gateways for its performance.

SONA also supports containers (Docker Swarm and Kubernetes) and allows communication between containers and VMs by controlling their flows using an SDN controller. (This video using Amazon's Alexa to deploy a container (https://superuser.openstack.org/articles/openstack-korea-days-2017/) in just a few minutes was a crowd pleaser.) The connections among containers and VMs are essential to 5G infrastructure where some of VNFs have micro-service architecture utilizing containers to improve its agility.

For more details on the carrier-grade SDN-based OpenStack networking solution, check out this 30-minute session by SK telecom's Daniel Park and Sangho Shin at the OpenStack Summit Sydney.

Cover Photo (http://superuser.openstack.org/feed/"https://www.flickr.com/photos/clement127/17069392397/") // CC BY NC (https://creativecommons.org/licenses/by-nc/2.0/)

The post 5G and the really smart self-driving car (http://superuser.openstack.org/articles/5g-sktelecom-tests/) appeared first on OpenStack Superuser (http://superuser.openstack.org).

by Superuser at February 23, 2018 05:10 PM (http://superuser.openstack.org/articles/5g-sktelecom-tests/)

David Moreau Simard (https://dmsimard.com/categories/openstack/)

Rebranding Ansible Run Analysis to ARA Records Ansible (https://dmsimard.com/blog/rebranding-ansible-run-analysis-to-ara-records-ansible/)

So I got an idea recently… Let's rebrand Ansible Run Analysis to ARA records Ansible. If you'd like to review and comment on the code change, you can do so here: https://review.openstack.org/#/c/547245/. Why ? I watched the last season of Sillicon Valley recently. The series, while exaggerated, provides a humorous look at the world of startups. I don't have any plans on creating a startup but I love that it makes you think about things like needing a clever name or how you would do a proper "elevator" pitch to get funding.

February 23, 2018 12:00 AM (https://dmsimard.com/blog/rebranding-ansible-run-analysis-to-ara-records-ansible/)

Rebranding Ansible Run Analysis to ARA Records Ansible (https://dmsimard.com/2018/02/23/rebranding-ansible-run-analysis-to-ara-records-ansible/)

So I got an idea recently… Let's rebrand Ansible Run Analysis to ARA records Ansible. If you'd like to review and comment on the code change, you can do so here: https://review.openstack.org/#/c/547245/. Why ? I watched the last season of Sillicon Valley recently. The series, while exaggerated, provides a humorous look at the world of startups. I don't have any plans on creating a startup but I love that it makes you think about things like needing a clever name or how you would do a proper "elevator" pitch to get funding.

February 23, 2018 12:00 AM (https://dmsimard.com/2018/02/23/rebranding-ansible-run-analysis-to-ara-records-ansible/)

Carlos Camacho (https://www.anstack.com/)

TripleO deep dive session #12 (config-download) (https://www.anstack.com/blog/2018/02/23/tripleo-deep-dive-session-12.html)

This is the 12th release of the TripleO (http://www.tripleo.org/) "Deep Dive" sessions

Thanks to James Slagle (http://blog-slagle.rhcloud.com/) for this new session, in which he will describe and speak about a feature called `config-download` .

In this session we will have an update for the TripleO ansible integration called `config-download` . It's about aplying all the software configuration with Ansible instead of doing it with the Heat agents.

So please, check the full session (https://www.youtube.com/watch?v=-6ojHT8P4RE) content on the TripleO YouTube channel (https://www.youtube.com/channel/UCNGDxZGwUELpgaBoLvABsTA/).

Please check the sessions index (http://www.anstack.com/blog/2017/06/15/tripleo-deep-dive-session-index.html) to have access to all available content.

by Carlos Camacho at February 23, 2018 12:00 AM (https://www.anstack.com/blog/2018/02/23/tripleo-deep-dive-session-12.html)

About

Planet OpenStack is a collection of thoughts from the developers and other key players of the OpenStack projects. If you are working on OpenStack technology you should add your OpenStack blog (http://wiki.openstack.org/AddingYourBlog).

Subscriptions

() Aaron Rosen

- (https://blog.adamspiers.org/tag/openstack/feed/atom/) Adam Spiers (https://blog.adamspiers.org)
- (http://adam.younglogic.com/category/software/openstack/feed/) Adam Young (http://adam.younglogic.com)
- (http://blog.adityapatawari.com/feeds/posts/default/-/OpenStack) Aditya Patawari (http://blog.adityapatawari.com/search/label/OpenStack)
- (http://www.17od.com/tag/openstack/feed/) Adrian Smith (http://www.17od.com)
- (https://cloudbase.it/category/openstack/feed/) Alessandro Pilotti (https://cloudbase.it)
- (https://aababilov.wordpress.com/category/openstack-2/feed/) Alessio Ababilov (https://aababilov.wordpress.com)
- (https://alvarolopez.github.io/feeds/openstack.atom.xml) Alvaro Lopez Garcia (http://alvarolopez.github.io/)
- (http://www.buildcloudstorage.com/feeds/posts/default) Amar Kapadia (http://www.buildcloudstorage.com/)
- (https://hypecycles.com/category/OpenStack/feed/) Amrith Kumar (https://hypecycles.com)
- (http://amalagon.github.io/blog/categories/openstack/atom.xml) Ana Malagon (http://amalagon.github.io/)
- (https://andreafrittoli.me/tag/openstack/feed/) Andrea Frittoli (https://andreafrittoli.me)
- (http://jaegerandi.blogspot.com/feeds/posts/default/-/OpenStack) Andreas Jaeger (http://jaegerandi.blogspot.com/search/label/OpenStack)
- (https://virtualandy.wordpress.com/tag/openstack/feed/atom/) Andy Hill (https://virtualandy.wordpress.com)
- (https://ahsalkeld.wordpress.com/category/openstack/feed/atom/) Angus Salkeld (https://ahsalkeld.wordpress.com)
- (http://anteaya.info/atom.xml) Anita Kuno (http://anteaya.info/)
- (http://justwriteclick.com/tag/openstack/feed/) Anne Gentle (https://justwriteclick.com)
- () Anthony Goddard
- (https://www.reversengineered.com/tag/openstack/feed/) Antony Messerli (https://www.reversengineered.com)
- (http://blog.appformix.com/rss.xml?tag=openstack) AppFormix (http://blog.appformix.com)
- (https://feeds.feedburner.com/Aptira?format=xml) Aptira (https://aptira.com)
- (http://abregman.com/category/openstack/feed/) Arie Bregman (http://abregman.com)
- (https://arrfab.net/feeds/rdo.atom.xml) Arrfab's Blog (https://arrfab.net/)
- (http://www.berezins.com/category/openstack/feed/) Arthur Berezin (http://www.berezins.com)
- (https://notartom.net/category/openstack/feed/) Artom Lifshitz (https://notartom.net)
- (https://arunsag.wordpress.com/tag/openstack/feed/) Arun S A G (https://arunsag.wordpress.com)
- (http://old.arxcruz.net/category/openstack/feed/) Arx Cruz (http://old.arxcruz.net)
- (https://assafmuller.com/category/openstack/feed/) Assaf Muller (https://assafmuller.com)
- (http://hackhackblog.blogspot.com/feeds/posts/default/-/openstack) Augustina Ragwitz (http://hackhackblog.blogspot.com/search/label/openstack)
- (http://blog.nemebean.com/taxonomy/term/3/feed) Ben Nemec (http://blog.nemebean.com/tags/openstack)
- (http://benjaminkerensa.com/category/openstack/feed) Benjamin Kerensa (http://benjaminkerensa.com)
- (http://blog.cafarelli.fr/category/openstack/feed/) Bernard Cafarelli (http://blog.cafarelli.fr)
- (https://blog.betacloud.io/tag/openstack-planet/feed/atom/) Betacloud (https://betacloud.io)
- (http://www.logicative.com/category/openstack/feed/) Beth Elwell (http://www.logicative.com)
- (http://www.bodenr.blogspot.com/feeds/posts/default/-/openstack) Boden Russell (http://bodenr.blogspot.com/search/label/openstack)
- (http://www.bretpiatt.com/blog/category/openstack/feed/) Bret Piatt
- (http://blog.briancurtin.com/categories/openstack.xml) Brian Curtin (http://briancurtin.com/blog/)
- (http://blog.seader.us/feeds/posts/default/-/OpenStack) Cameron Seader (http://blog.seader.us/search/label/OpenStack)
- (http://captainkvm.com/tag/openstack/feed/) Captain KVM (http://captainkvm.com)
- (http://blog.episodicgenius.com/tags/openstack/index.xml) Carl Baldwin (http://blog.episodicgenius.com/tags/openstack/)
- (https://www.anstack.com/openstack.xml) Carlos Camacho (https://www.anstack.com/)
- (https://ceph.com/community/blog/tag/openstack/feed/) Ceph
- () Chandan Kumar
- (https://blog.chmouel.com/category/openstack/feed/) Chmouel Boudjnah (https://blog.chmouel.com)
- (https://anticdent.org/feeds/openstack.atom.xml) Chris Dent (https://anticdent.org/)
- (http://chris.yeoh.info/category/openstack/feed/) Chris Yeoh (http://chris.yeoh.info)
- (https://blog.christophersmart.com/category/foss/openstack/feed/) Christopher Smart (https://blog.christophersmart.com)
- () Cindy Pallares
- (https://blogs.cisco.com/tag/openstack/feed) Cisco Cloud Blog (https://blogs.cisco.com)
- (https://communities.cisco.com/community/technology/cloudsolutions/cisco-metacloud/blog/feeds/posts) Cisco Metacloud (https://communities.cisco.com/community/technology/cloudsolutions/cisco-metacloud/blog)
- (http://citrix-openstack.siteleaf.net/feed.xml) Citrix OpenStack Blog (http://citrix-openstack.siteleaf.net/)
- (http://fewbar.com/feed.xml) Clint Byrum (http://fewbar.com/)
- (https://www.symantec.com/connect/item-feeds/all/all/feed/all/all/3249921) Cloud Platform @ Symantec (https://www.symantec.com/connect/item-feeds/all/all/feed/all/all/3249921)
- (https://cloudbau.github.io/feed.openstack.xml) Cloudbau Blog (https://cloudbau.github.io)
- (https://cloudify.co/tags/OpenStack/atom.xml) Cloudify Engineering (http://getcloudify.org)
- () Cloudscaling Corporate Blog
- (http://feeds.feedburner.com/SimplicityScales?format=xml) Cloudscaling Engineering (http://engineering.cloudscaling.com)
- (https://dev.cloudwatt.com/en/blog/rss.xml) Cloudwatt (https://dev.cloudwatt.com/en/blog/index.html)
- () Cody Bunch
- (http://www.gazlene.net/feeds/all.rss.xml) Colleen Murphy (http://gazlene.net/)
- (https://wrestlingpenguins.wordpress.com/category/openstack/feed/) Corey Bryant (https://wrestlingpenguins.wordpress.com)
- (https://mcwhirter.com.au/tags/OpenStack/index.atom) Craige McWhirter (http://mcwhirter.com.au//tags/OpenStack/)
- (https://dronopenstack.wordpress.com/category/openstack/feed/) Dafna Ron (https://dronopenstack.wordpress.com)
- () Dan Kingshott
- (https://dprince.github.io/feeds/tag_openstack.atom.xml) Dan Prince (https://dprince.github.io/)
- (http://www.jaddog.org/category/openstack/feed/) Dan Radez (http://www.jaddog.org)
- (http://www.danplanet.com/blog/category/openstack/feed/) Dan Smith (http://www.danplanet.com/blog)
- (https://www.berrange.com/topics/openstack/feed/) Daniel P. Berrangé (https://www.berrange.com)
- (https://www.stackevolution.com/taxonomy/term/2/feed) Darryl Weaver (https://www.stackevolution.com/openstack)
- (http://dowdberry.blogspot.com/feeds/posts/default/-/openstack) David Medberry (http://dowdberry.blogspot.com/search/label/openstack)
- (https://dmsimard.com/categories/openstack/index.xml) David Moreau Simard (https://dmsimard.com/categories/openstack/)
- (http://dstanek.com/tags/openstack/index.xml) David Stanek (https://dstanek.com/tags/openstack/index.xml)
- (http://hackstack.org/x/blog/category/openstack/feed/) Dean Troyer (http://hackstack.org/x/blog)
- (http://goodsquishy.com/tag/openstack/feed/atom/) Derek Higgins (http://goodsquishy.com)
- (http://feeds.doughellmann.com/doughellmann/openstack) Doug Hellmann (https://doughellmann.com/blog)
- (http://www.dougalmatthews.com/feeds/tag/openstack.atom.xml) Dougal Matthews (http://www.dougalmatthews.com/)
- (http://www.dragonflow.net/feeds/posts/default/-/OpenStack) Dragonflow Team (http://www.dragonflow.net/search/label/OpenStack)

- (https://www.dreamhost.com/blog/tag/openstack/feed/) DreamHost (https://www.dreamhost.com/blog)
- (http://oubiwann.blogspot.com/feeds/posts/default/-/openstack) Duncan McGreggor (http://oubiwann.blogspot.com/search/label/openstack)
- (https://blog.leafe.com/category/openstack/feed/) Ed Leafe (https://blog.leafe.com)
- (http://princessleia.com/journal/category/openstack/feed/) Elizabeth K. Joseph (http://princessleia.com/journal)
- (http://my1.fr/blog/category/virtualization/openstack/feed/atom/) Emilien Macchi (http://my1.fr/blog)
- (https://enriquetaso.wordpress.com/category/OpenStack/feed/) Enriquez Laura Sofia (https://enriquetaso.wordpress.com)
- (http://itsonlyme.name/openstack/storlets/rss.xml) Eran Rom (http://itsonlyme.name/blog)
- (http://blog.phymata.com/category/openstack.atom.xml) Everett Toews (http://blog.phymata.com/)
- (http://blog.fsquat.net/?tag=openstackplanet&feed=rss2) Fabien Boucher (http://blog.fsquat.net)
- (http://blog.flaper87.com/feeds/openstack.atom.xml) Flavio Percoco (https://blog.flaper87.com/)
- (https://fleio.com/blog/tag/openstack/feed/) Fleio Blog (https://fleio.com/blog)
- (http://www.florentflament.com/blog/feeds/openstack.atom.xml) Florent Flament (http://www.florentflament.com/blog/)
- (http://galsagie.github.io/feed.xml) Gal Sagie (http://galsagie.github.io)
- (http://galeracluster.com/category/planet-openstack/feed/) Galera Cluster by Codership (http://galeracluster.com/category/blog/)
- () Geetika Batra
- (http://giuliofidente.com/feeds/tag/openstack.atom.xml) Giulio Fidente (http://giuliofidente.com/)
- (https://gorka.eguileor.com/feed/?cat=10&feed=rss2) Gorka Eguileor (https://gorka.eguileor.com)
- (http://graham.hayes.ie/categories/cat_openstack.xml) Graham Hayes (http://graham.hayes.ie/)
- (https://openstackgd.wordpress.com/feed/) Grid Dynamics OpenStack Team (https://openstackgd.wordpress.com)
- () HP Public Cloud
- (http://ehaselwanter.com/en/openstack.xml) Haselwanter Edmund (http://ehaselwanter.com)
- () Hendrik Volkmer
- (https://www.host-telecom.com/tag/openstack/feed/) Host-Telecom.com (https://www.host-telecom.com)
- (https://hugh.blemings.id.au/category/openstack/feed/) Hugh Blemings (https://hugh.blemings.id.au)
- (https://developer.ibm.com/opentech/category/openstack/feed/) IBM OpenTech Team (https://developer.ibm.com/opentech)
- (https://blog.zhaw.ch/icclab/category/articles/openstack-2/feed/) ICCLab (https://blog.zhaw.ch/icclab)
- (http://drop.isi.edu/node/407/rss) ISI High Performance Cloud Computing Group (http://www.isi.edu/research_groups/HPCC/blog)
- (https://ihrachyshka.com/category/openstack/feed/) Ihar Hrachyshka (https://ihrachyshka.com)
- () Imaginea Technologies
- (https://blog.imaginea.com/tag/openstack/feed/) Imaginea Technologies (https://blog.imaginea.com)
- (http://www.infralovers.com/en/articles/categories/openstack/atom.xml) Infralovers (http://www.infralovers.com/)
- (http://www.inap.com/feed/?cat=1001410) Internap (http://www.inap.com)
- () JJ Asghar
- (http://amo-probos.org/feed/rss2_0?tag=openstack) James E. Blair (http://amo-probos.org/)
- (https://javacruft.wordpress.com/category/openstack/feed/) James Page (https://javacruft.wordpress.com)
- (http://blog-slagle.rhcloud.com/?feed=rss2&tag=OpenStack) James Slagle (http://blog-slagle.rhcloud.com)
- (http://www.jamielennox.net/blog/categories/openstack/atom.xml) Jamie Lennox (http://www.jamielennox.net/)
- (https://jpenatech.wordpress.com/feed/atom/?cat=7) Javier Peña (https://jpenatech.wordpress.com)
- () Jay Clark
- (http://www.joinfu.com/category/openstack/feed/) Jay Pipes (http://www.joinfu.com)
- (http://odyssey4me.github.io/openstack.xml) Jesse Pretorius (http://odyssey4me.github.io/)
- (http://www.isjian.com/openstack/feed.xml) Jian Li (http://www.isjian.com)
- (https://software.intel.com/en-us/user/335367/feed) Jiangang Duan (Intel) (https://software.intel.com/en-us/recent/335367)
- (https://jfehlig.wordpress.com/category/openstack/feed/) Jim Fehlig (https://jfehlig.wordpress.com)
- (https://www.jistr.com/blog/tags/openstack/feed.xml) Jiří Stránský (https://www.jistr.com/blog/tags/openstack/)
- (https://tropicaldevel.wordpress.com/category/openstack/feed/) John Bresnahan (https://tropicaldevel.wordpress.com)
- (http://programmerthoughts.com/openstack/openstackposts.xml) John Dickinson (http://programmerthoughts.com/)
- (http://john.eckersberg.com/feeds/tag/openstack.atom.xml) John Eckersberg (http://john.eckersberg.com/)
- (https://griffithscorner.wordpress.com/category/openstack/feed/) John Griffith (https://griffithscorner.wordpress.com)
- (http://blog.johnlikesopenstack.com/feeds/posts/default) John Likes OpenStack (http://blog.johnlikesopenstack.com/)
- (http://blog.iwokeupcoveredinblood.com/blog/categories/openstack/atom.xml) Jon Proulx (http://blog.iwokeupcoveredinblood.com/)
- (https://josh.people.rcbops.com/tag/openstack/feed/) Joshua Hesketh (https://josh.people.rcbops.com)
- (https://www.jpichon.net/feeds/tag-openstack.rss.xml) Julie Pichon (https://www.jpichon.net/)
- (https://julien.danjou.info/blog/tags/OpenStack.xml) Julien Danjou (https://julien.danjou.info/blog/)
- (http://www.juliosblog.com/tag/openstack/rss/) Julio Villarreal Pelegrino (http://www.juliosblog.com/)
- (https://kashyapc.wordpress.com/tag/openstack/feed/) Kashyap Chamarthy (https://kashyapc.wordpress.com)
- (https://cloud-guy.net/category/openstack/feed/atom/) Keith Tobin (https://cloud-guy.net)
- () Ken Pepple
- (https://pigdogweb.wordpress.com/tag/openstack/feed/atom/) Kenneth Giusti (https://pigdogweb.wordpress.com)
- (https://cloudarchitectmusings.com/tag/openstack/feed/) Kenneth Hui (https://cloudarchitectmusings.com)
- (https://kimizhang.wordpress.com/category/openstack/feed/) Kimi Zhang (https://kimizhang.wordpress.com)
- (https://www.siliconloons.com/categories/openstack/index.xml) Kyle Mestery (https://blog.siliconloons.com/categories/openstack/)
- (https://www.lbragstad.com/category/openstack/feed/) Lance Bragstad (http://lbragstad.com)
- (http://blog.oddbit.com/tag/openstack/rss/) Lars Kellogg-Stedman (http://blog.oddbit.com/)
- (http://www.thequietstack.com/?tag=openstack&feed=rss) Laura Alves
- (https://blog.yarwood.me.uk/tags/openstack/index.xml) Lee Yarwood (https://blog.yarwood.me.uk/tags/openstack/)
- (http://uxd-stackabledesign.rhcloud.com/category/openstack/feed/) Liz Blanchard (http://uxd-stackabledesign.rhcloud.com)
- (http://feeds.feedburner.com/logilaborg_openstack) Logilab
- (https://lorinhochstein.wordpress.com/category/openstack/feed/atom/) Lorin Hochstein (https://lorinhochstein.wordpress.com)
- (https://blog.dachary.org/category/libcrush/feed/) Loïc Dachary (https://blog.dachary.org)
- (http://technodrone.blogspot.com/feeds/posts/default/-/OpenStack) Maish Saidel-Keesing (http://technodrone.blogspot.com/search/label/OpenStack)
- (https://major.io/tag/openstack/feed/) Major Hayden (https://major.io)
- (http://manishankert.blogspot.in/feeds/posts/default/-/openstack) Manishanker Talusani (http://manishankert.blogspot.com/search/label/openstack)
- (http://blog.mbonell.com/category/openstack/feed/) Marcela Bonell (http://blog.mbonell.com)
- (https://crustyblaa.com/feeds/openstack.atom.xml) Mark McLoughlin (https://crustyblaa.com/)
- (http://www.markshuttleworth.com/archives/tag/openstack/feed) Mark Shuttleworth (http://www.markshuttleworth.com)
- (http://markvoelker.github.io/blog/index.xml) Mark T. Voelker (http://markvoelker.github.io/blog/)
- (http://hybridcloudburst.com/category/openstack/feed/atom/) Marten Hauville

- () Marton Kiss (ios-app://305343404/tumblr/x-callback-url/blog?blogName=mkissam&postID=)
- (https://domsch.com/blog/?feed=atom&tag=openstack) Matt Domsch (https://domsch.com/blog)
- (http://madorn.com/feeds/openstack.rss.xml) Matt Dorn (http://www.madorn.com/)
- (https://codeengineered.com/tag/OpenStack%20Planet/atom.xml) Matt Farina (https://codeengineered.com/tag/OpenStack%20Planet/)
- (http://www.mattfischer.com/blog/?feed=rss2&cat=50) Matt Fischer (http://www.mattfischer.com/blog)
- (http://www.mattgriffin.com/tag/openstack/feed/) Matt Griffin (http://www.mattgriffin.com)
- (https://leastresistance.wordpress.com/category/openstack/feed/) Matt Ray (https://leastresistance.wordpress.com)
- (https://spinningmatt.wordpress.com/category/openstack/feed/atom/) Matthew Farrellee (https://spinningmatt.wordpress.com)
- (https://blog.kortar.org/?cat=2&feed=rss2) Matthew Treinish (https://blog.kortar.org)
- () Matthias Runge
- (https://blog.sileht.net/feeds/openstack.atom.xml) Mehdi Abaakouk (https://blog.sileht.net/)
- (http://lifelog.michaeldavies.org/feeds/posts/default/-/openstack) Michael Davies (http://lifelog.michaeldavies.org/search/label/openstack)
- (https://krotscheck.net/tag/openstack/feed) Michael Krotscheck (https://krotscheck.net)
- (http://www.stillhq.com/openstack/index.rss) Michael Still (http://www.stillhq.com)
- (http://acksyn.org/categories/openstack.xml) Michele Baldessari (http://acksyn.org/)
- (http://blog.midonet.org/category/openstack/feed/) MidoNet (https://blog.midonet.org)
- (http://blog.midokura.com/category/OpenStack/feed/) Midokura (http://blog.midokura.com)
- (http://www.ajo.es/tagged/openstack/rss) Miguel Ángel Ajo (http://www.ajo.es/)
- () Mika Ayenson
- (http://www.dorm.org/blog/category/openstack/feed/) Mike Dorman (http://www.dorm.org/blog)
- (https://mirandazhangq.wordpress.com/category/openstack/feed/atom/) Miranda ZHANG (https://mirandazhangq.wordpress.com)
- (https://www.mirantis.com/feed/planet_openstack/) Mirantis (https://www.mirantis.com)
- () Monty Taylor
- (https://blog.nfvpe.site/feed/) NFVPE @ Red Hat (https://blog.nfvpe.site)
- (https://blog-nkinder.rhcloud.com/?cat=2&feed=atom) Nathan Kinder (https://blog-nkinder.rhcloud.com)
- (https://nikhilkathole.wordpress.com/category/OpenStack/feed/) Nikhil Kathole (https://nikhilkathole.wordpress.com)
- (http://www.nirmagnezi.com/openstack/feed/) Nir Magnezi
- (https://thenetworkway.wordpress.com/tag/openstack/feed/) Nir Yechiel (https://thenetworkway.wordpress.com)
- (https://osic.org/blog/feed) OSIC - The OpenStack Innovation Center (https://osic.org/blog)
- (https://www.olindata.com/en/taxonomy/term/309/feed) OlinData (https://www.olindata.com/en/blog/review)
- (http://netapp.io/category/openstack/feed/) OpenStack @ NetApp (https://netapp.io)
- (https://www.openstack.org/blog/feed/) OpenStack Blog (https://www.openstack.org/blog)
- (https://cloudnativetech.wordpress.com/category/openstack/feed/) OpenStack Blog - Swapnil Kulkarni (https://cloudnativetech.wordpress.com)
- (https://openstackindia.wordpress.com/feed/) OpenStack India (https://openstackindia.wordpress.com)
- (http://openStackindiaonline.github.io/feeds/all.atom.xml) OpenStack India Online (http://openStackindiaonline.github.io/)
- (https://novarollup.wordpress.com/feed/) OpenStack Nova Developer Rollup (https://novarollup.wordpress.com)
- () OpenStack Reactions
- (http://secstack.org/category/openstack/feed/) OpenStack Security Blog
- (http://superuser.openstack.org/feed) OpenStack Superuser (http://superuser.openstack.org)
- (http://openstack-in-production.blogspot.com/feeds/posts/default) OpenStack in Production (http://openstack-in-production.blogspot.com/)
- (https://opensource.com/taxonomy/term/5126/feed/feed) Opensource.com (https://opensource.com/taxonomy/term/5126/feed/feed)
- (https://openstack-security.github.io/feed.xml) Openstack Security Project (https://openstack-security.github.io/)
- (http://openstackbr.blogspot.com.br/feeds/posts/default/-/openstack) Openstack-br (http://openstackbr.blogspot.com/search/label/openstack)
- (http://mezzanine-lordkrandel.rhcloud.com/tag/openstack/feeds/rss/) Paolo Gatti (http://mezzanine-lordkrandel.rhcloud.com/)
- (https://www.percona.com/blog/category/openstack/feed/) Percona (https://www.percona.com/blog)
- (https://www.cisco.com/c/en/us/about/corporate-strategy-office/acquisitions/piston.html) Piston
- (http://www.pixelbeat.org/feed/openstack.rss2.xml) Pádraig Brady (http://www.pixelbeat.org/)
- (https://blogs.rdoproject.org/feed/) RDO (https://blogs.rdoproject.org)
- (https://developer.rackspace.com/blog/categories/openstack/atom.xml) Rackspace Developer Blog (https://developer.rackspace.com)
- (https://trickycloud.wordpress.com/category/openstack/feed/atom/) Ramon Acedo (https://trickycloud.wordpress.com)
- (https://www.ravellosystems.com/blog/tag/openstack/feed) Ravello Systems (https://www.ravellosystems.com/blog)
- (https://redhatstackblog.redhat.com/feed/) Red Hat Stack (https://redhatstackblog.redhat.com)
- (http://drbacchus.com/tag/openstack/feed/) Rich Bowen (http://drbacchus.com)
- (http://mechanicalcat.net/richard/log/OpenStack/rss) Richard Jones (http://mechanicalcat.net/richard/log/OpenStack)
- (https://robhirschfeld.com/category/openstack/feed/) Rob Hirschfeld (https://robhirschfeld.com)
- (https://rbtcollins.wordpress.com/tag/openstack/feed/atom/) Robert Collins (https://rbtcollins.wordpress.com)
- (http://ronaldbradford.com/blog/category/cloud/openstack/feed/) Ronald Bradford (http://ronaldbradford.com/blog)
- (http://roozbehshafiee.com/category/openstack/feed/) Roozbeh Shafiee (http://roozbehshafiee.com)
- (https://rossella-sblendido.net/category/openstack/feed/) Rossella Sblendido (https://rossella-sblendido.net)
- (https://blog.russellbryant.net/category/openstack/feed/) Russell Bryant (https://blog.russellbryant.net)
- (http://feeds.feedburner.com/RyanLanesBlog_openstack) Ryan Lane (https://blog.ryandlane.com)
- (https://www.suse.com/communities/blog/tag/openstack/feed/) SUSE Conversations (https://www.suse.com/communities/blog)
- (https://cloudblog.switch.ch/category/openstack/feed/atom/) SWITCH Cloud Blog (https://cloudblog.switch.ch)
- () Sandro Mathys
- (https://saschpe.wordpress.com/category/openstack/feed/) Sascha Peilicke (https://saschpe.wordpress.com)
- (https://dague.net/category/openstack/feed/) Sean Dague (https://dague.net)
- (http://sarob.com/category/openstack/feed/) Sean Roberts (https://sarob.com)
- (https://shaifaliagrawal.wordpress.com/category/openstack/feed/) Shaifali Agrawal (https://shaifaliagrawal.wordpress.com)
- (http://www.debug-all.com/?feed=rss2&tag=openstack) Shannon McFarland (http://www.debug-all.com)
- (https://solinea.com/tag/openstack/feed) Solinea (https://solinea.com)
- (http://blog.warma.dk/tag/planetopenstack/feed/) Soren Hansen (http://blog.linux2go.dk)
- (http://engineering.spilgames.com/category/planetopenstack/feed/) Spilgames Engineering (http://engineering.spilgames.com)
- (https://www.stackhpc.com/feeds/stackhpc.atom.xml) StackHPC Team Blog (https://www.stackhpc.com/)
- (https://www.stackmasters.eu/tag/openstack/feed/) Stackmasters team (https://www.stackmasters.eu)
- (http://maffulli.net/category/openstack/feed/) Stefano Maffulli (https://maffulli.net)
- (https://that.guru/categories/openstack/index.xml) Stephen Finucane (https://that.guru/categories/openstack/)
- (http://blog.stevebaker.org/feeds/posts/default/-/openstack) Steve Baker (http://blog.stevebaker.org/search/label/openstack)
- (https://sdake.io/tag/openstack/feed/) Steve Dake (https://sdake.io)

(http://hardysteven.blogspot.co.uk/feeds/posts/default/-/openstack) Steve Hardy (http://hardysteven.blogspot.com/search/label/openstack)
() SwiftStack Team
(https://sbauza.wordpress.com/tag/openstack/feed/) Sylvain Bauza (https://sbauza.wordpress.com)
() Sébastien Han
(http://telekomcloud.github.io/atom.xml) TelekomCloud DevOps team (http://telekomcloud.github.io)
() Terri Yu
(https://blog.otherwiseguy.com/tag/openstack/rss/) Terry Wilson (http://blog.otherwiseguy.com/)
(http://www.tesora.com/tag/openstack/feed/) Tesora Corp (http://www.tesora.com)
(https://blog.rackspace.com/tag/openstack/feed) The Official Rackspace Blog (https://blog.rackspace.com)
(https://storyboard-blog.sotk.co.uk/) The StoryBoard Team
(https://ttx.re/feeds/openstack.atom.xml) Thierry Carrez (https://ttx.re/)
(https://toabctl.wordpress.com/tag/openstack/feed/) Thomas Bechtold (https://toabctl.wordpress.com)
(https://ubuntuserver.wordpress.com/tag/openstack/feed/) Ubuntu Server & Cloud blog (https://ubuntuserver.wordpress.com)
(http://vmartinezdelacruz.com/category/OpenStack/feed/) Victoria Martínez de la Cruz (http://vmartinezdelacruz.com)
(http://www.vuntz.net/journal/feed/tag/openstack/atom) Vincent Untz (http://www.vuntz.net/journal/)
(https://blog.xenproject.org/tag/openstack/feed/) Xen Project Blog (https://blog.xenproject.org)
(http://cloudystuffhappens.blogspot.com/feeds/posts/default/-/openstack) Yun Mao (http://cloudystuffhappens.blogspot.com/search/label/openstack)
(https://www.zerobanana.com/feeds/tags/OpenStack) Zane Bitter (https://www.zerobanana.com/tags/OpenStack)
(http://www.zmanda.com/blogs/?feed=rss2&cat=22) Zmanda (http://www.zmanda.com/blogs)
(https://www.hastexo.com/feeds/openstack.atom.xml) hastexo (https://www.hastexo.com/)
(http://sites.rcbops.com/lca2014_openstack/feed/) linux.conf.au 2014 OpenStack miniconf (http://sites.rcbops.com/opencloud_symposium)
(https://lizards.opensuse.org/tag/openstack/feed/) openSUSE Lizards (https://lizards.opensuse.org)

Last updated:
March 20, 2018 01:52 PM
All times are UTC.

Powered by:
PLANET (http://www.planetplanet.org/)

OpenStack

About the Foundation (/foundation)
Projects (http://openstack.org/projects/)
OpenStack Security (http://openstack.org/projects/openstack-security/)
Common Questions (http://openstack.org/projects/openstack-faq/)
Blog (http://openstack.org/blog/)

Community

User Groups (http://openstack.org/community/)
Events (http://openstack.org/community/events/)
Jobs (http://openstack.org/community/jobs/)
Companies (http://openstack.org/foundation/companies/)
Contribute (https://wiki.openstack.org/wiki/How_To_Contribute)

Documentation

OpenStack Manuals (http://docs.openstack.org)
Getting Started (http://openstack.org/software/start/)
API Documentation (http://developer.openstack.org)
Wiki (https://wiki.openstack.org)

Branding & Legal

Logos & Guidelines (http://openstack.org/brand/)
Trademark Policy (http://openstack.org/brand/openstack-trademark-policy/)
Privacy Policy (http://openstack.org/privacy/)
OpenStack CLA (https://wiki.openstack.org/wiki/How_To_Contribute#Contributors_License_Agreement)

Stay In Touch

(https://tw[illegible]StackFoundation)