Suspending and Restarting the OpenStack Cookbook Lab

In our previous post about Installing the OpenStack Cookbook on OSX, we explored the simple first steps needed to get our OpenStack lab environment up and running on OSX. This is OSX specific, but quite similar to the lab being run on Windows and Linux. I will put together a quick post to show how to do the same on Windows and Linux also.

Because we want to manage our resources by bringing the systems down and up on occasion, it is important that we know how to quiesce the lab instances, and to be able to rebuild the lab from scratch when it is already deployed.

Suspending and Restarting your OpenStack Cookbook Lab

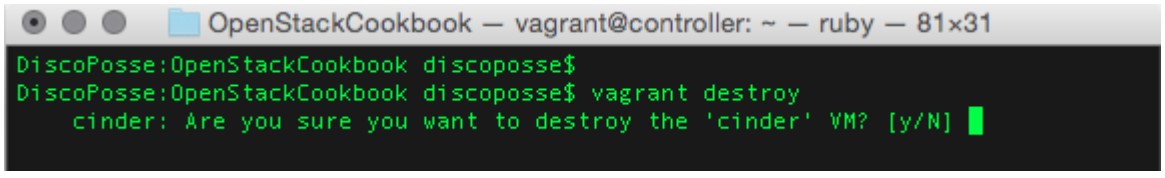
Luckily, there's a blog for that! I had written in the past about how to suspend and resume a number of virtual machines using vagrant and awk here: https://discoposse.com/2014/10/12/suspend-and-resume-vagrant-boxes-in-bulk-osx-linux/ which will give you the ability to quiesce and to spin up your OpenStack Cookbook lab easily.

Remember that your lab will use resources such as CPU, RAM, and most importantly power. I have been bitten by this in the past where I left a number of virtual instances running in the background while on battery and my laptop did see a rather rapid reduction in battery time as a result.

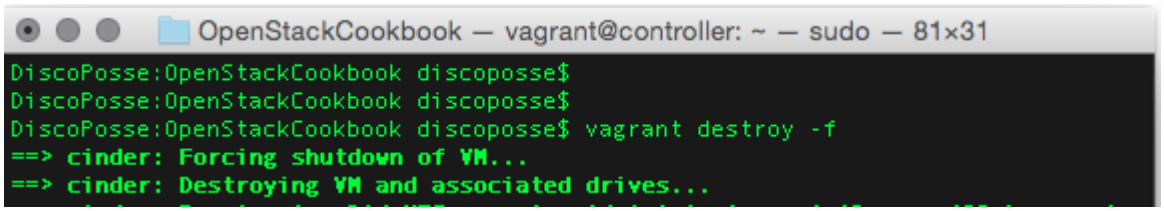
Deleting the OpenStack Cookbook Lab

There may be events that require you to remove the OpenStack Cookbook lab from your host. This could be to update the code, test some parts of the build, or it could be to correct an issue that may have occurred which is easier to recover from by rebuilding the lab in its entirety.

Removing the OpenStack Cookbook lab from your environment is done easily using the `vagrant destroy` command while in the home folder of the lab. You will be prompted for each machine as it begins the process of removing the VM:

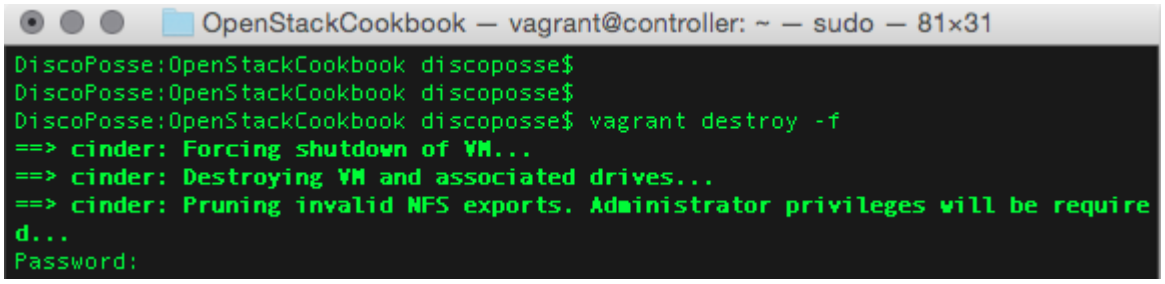


You can also use the `vagrant destroy -f` command instead but tread carefully as this command will run without questioning you at all. The `-f` parameter is an option to force the command without confirmation:

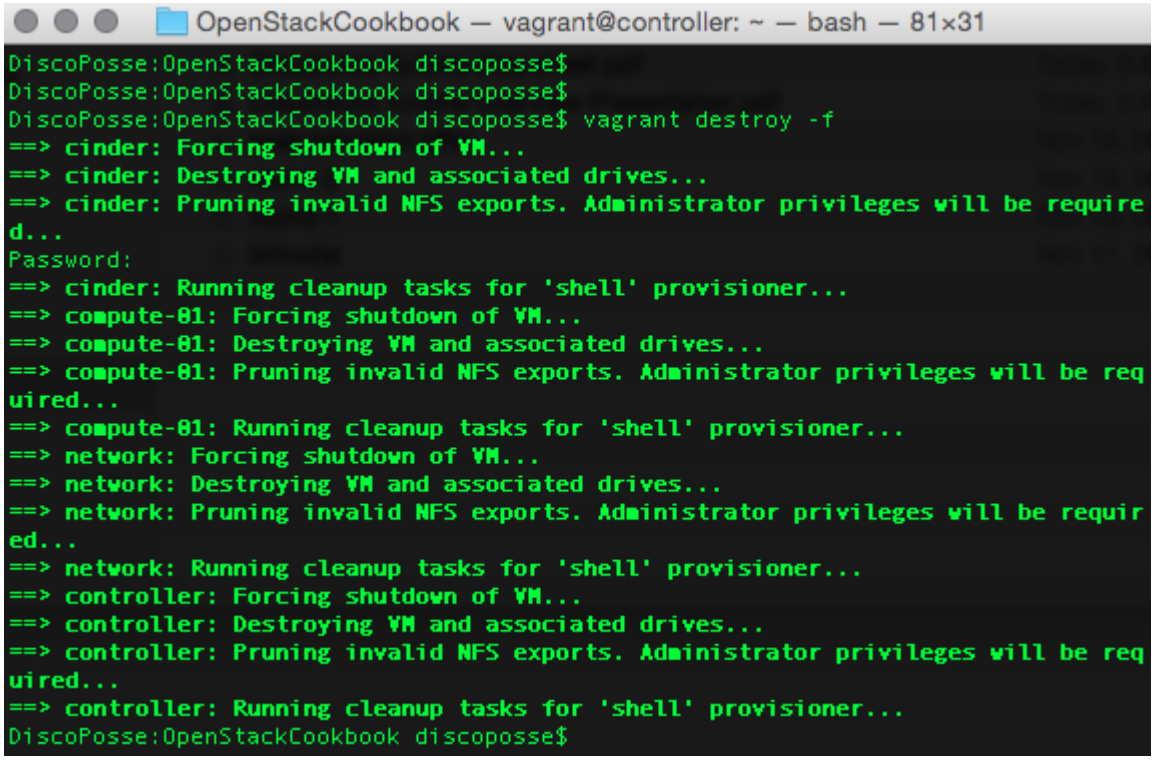


Because this affects the NFS exports that were created to share the /vagrant/ folder to your local

host machine, you will be prompted for credentials to remove the NFS attachments:



Once it is all done, you will see the resulting screen as follows:



Once done that, you can simply rebuild the lab by going back to the `vagrant up` part of the process and you will be back to stacking up the goodness again.

Installing the OpenStack Cookbook on OSX

As an advocate for OpenStack, I am always keen to help give newcomers to the open cloud ecosystem a chance to have a positive learning experience. A great tool for that is the OpenStack Cookbook lab. I've been lucky enough to have helped a little along the way as a reviewer for the 2nd edition of the book, and to have done some small code contributions as well.

Luckily, it doesn't take much to run the OpenStack Cookbook lab, and most importantly, it is free!

You will need to have the following on your Mac in order to run the OpenStack Cookbook lab:

- 8GB RAM minimum (16 is ideal)

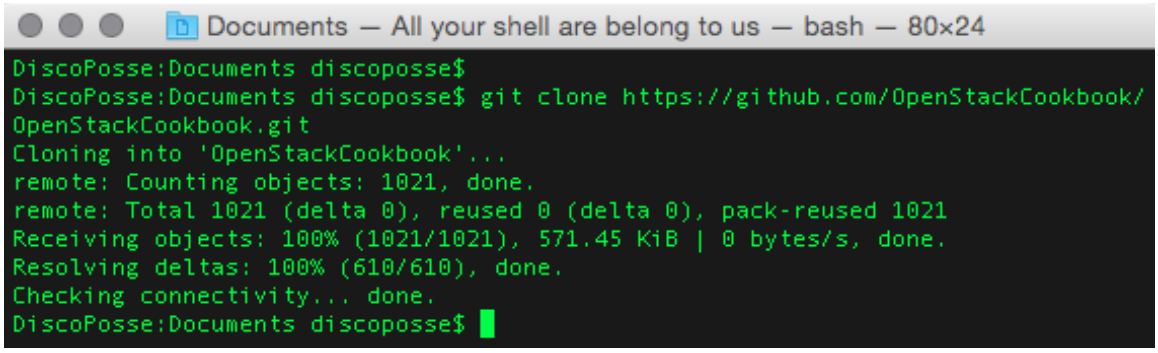
- Vagrant (http://www.vagrantup.com/)
- VirtualBox (http://www.virtualbox.org/)
- Internet access

I highlight internet access because you can run the cookbook lab without internet access, but only once it is fully downloaded and built. The lab works great for me during testing when on the road and while disconnected, but we do run a live download of the code and base machines during the setup process.

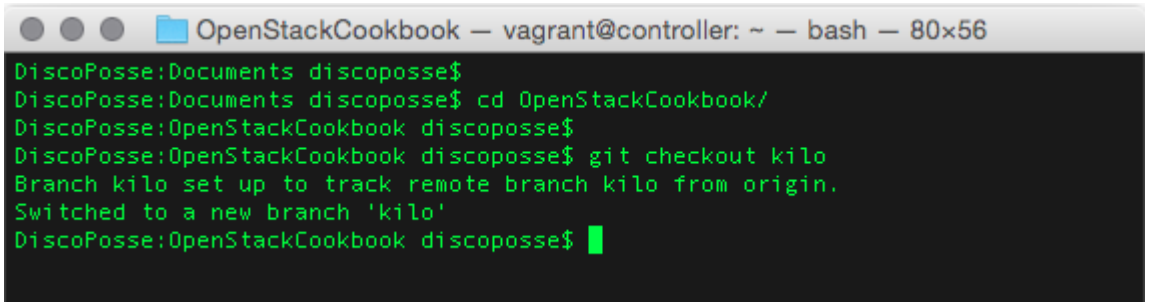
NOTE: Once you've installed Vagrant, it is highly recommended to also install the vagrant-cachier plugin. I've documented that process here: https://discoposse.com/2014/11/24/c-r-e-a-m-cache-rules-everything-around-me-with-vagrant-cachier/

Downloading the OpenStack Cookbook code

```
git clone https://github.com/OpenStackCookbook/OpenStackCookbook.git
```



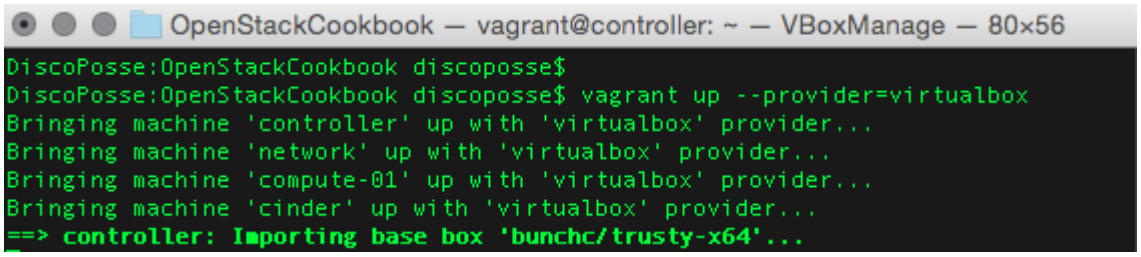
Let's change directory to the OpenStack Cookbook folder with the `cd OpenStackCookbook` and then we will want to ensure that we are on the stable Kilo branch of the cookbook using `git checkout kilo` to set the branch:



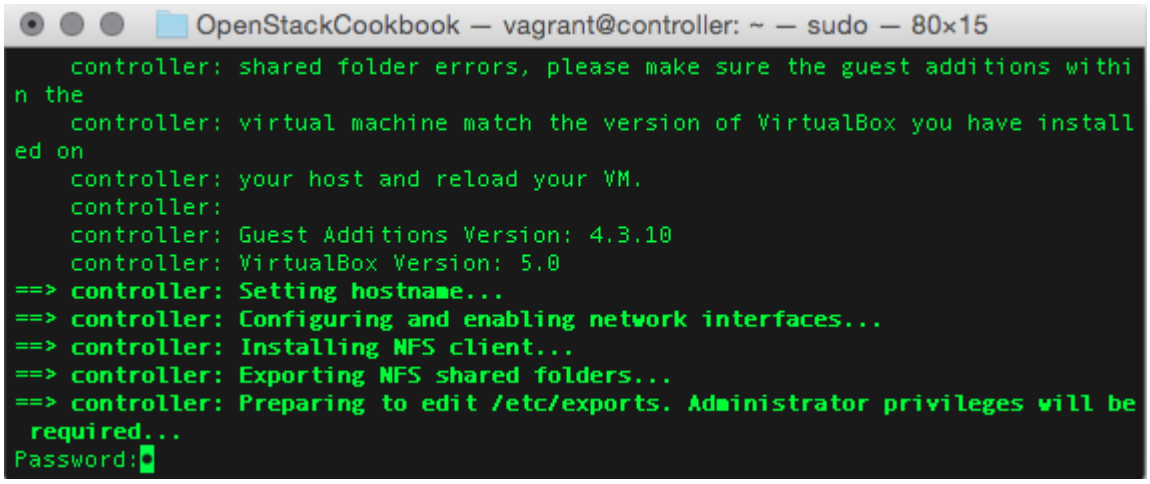
Now comes the fun part!

Vagrant up

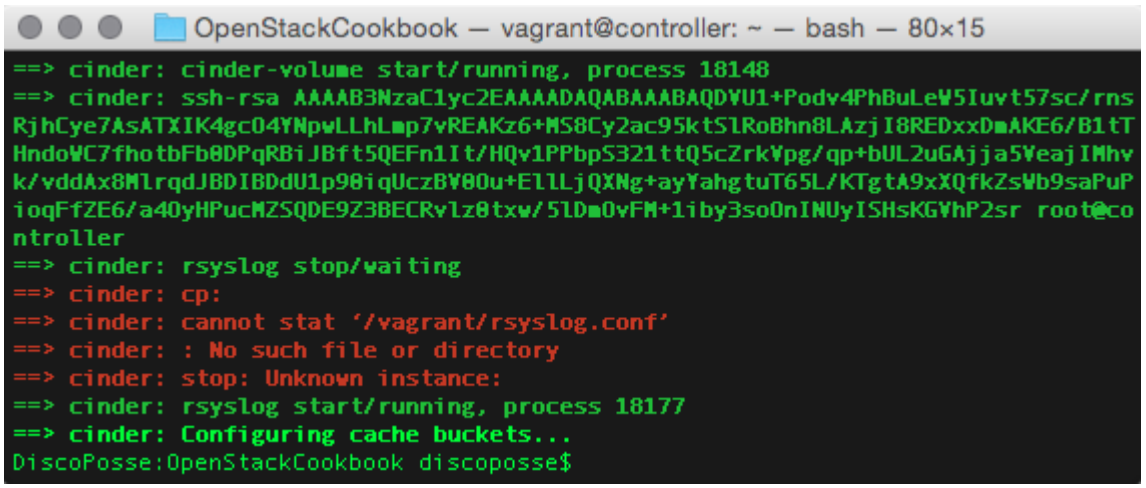
Since you have the Vagrant and VirtualBox all set up on your system and the code is downloaded, it's time to start up the build of the lab. This is easily done thanks to the magic of automation by using the `vagrant up --provider=virtualbox/`command:



You will be prompted for credentials during the process of the build at least once as it needs to set up NFS shares to map the /vagrant/ directory from the guest virtual machines to your local host for access the build scripts on the shared folder:

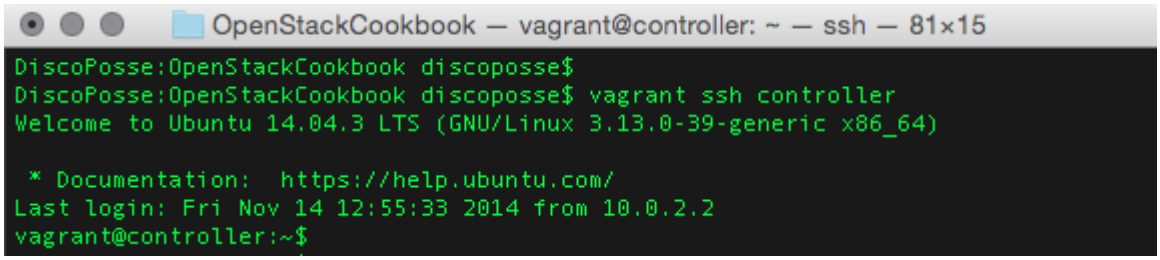


The build takes about 20-30 minutes depending on your internet connection speed and the CPU, memory, and disk performance in your lab host. Once it's all done building you will see a screen similar to this:

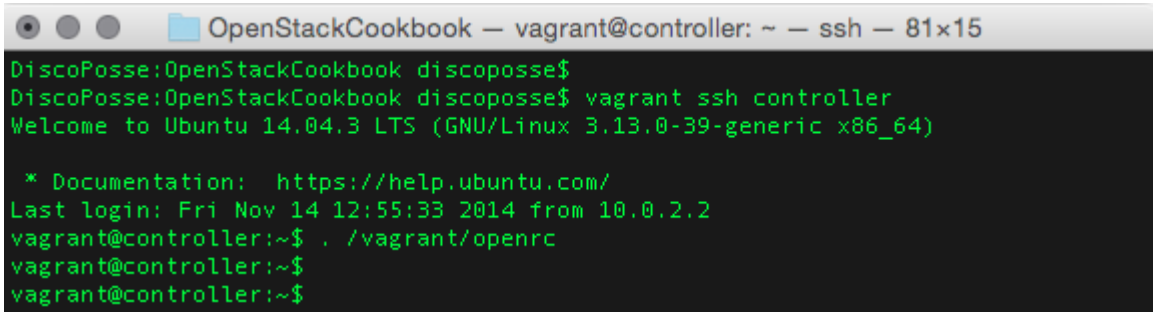


Now it's time to start up the demo script which will build out the first Neutron router, some floating IP addresses, download a couple of guest instance images to Glance, and spark up a Nova instance. This is done from the console of the controller virtual machine.

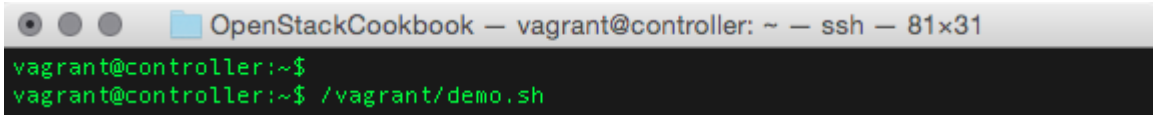
First, SSH into the machine from the command line using the `vagrant ssh controller` command:



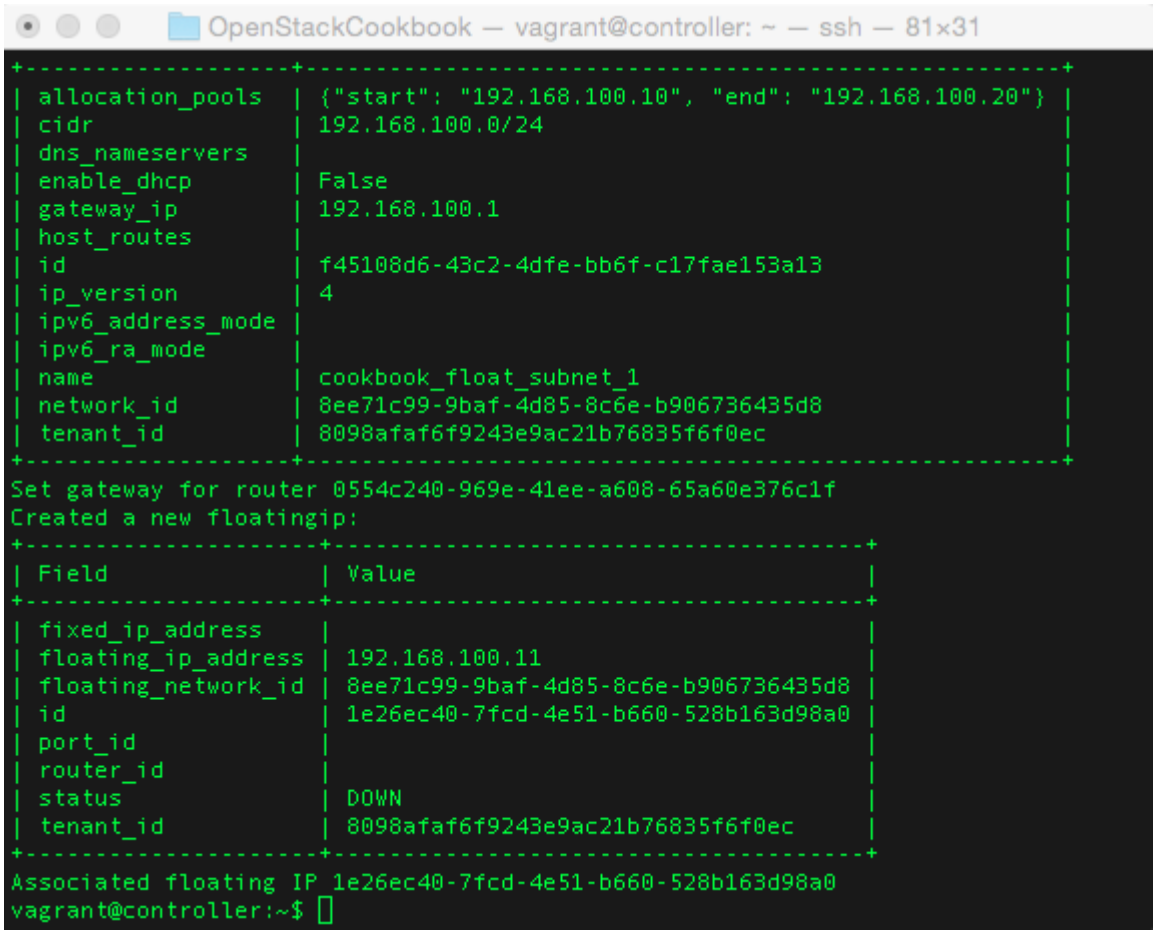
Next, we import the openrc file to put some environment variables together which will give us access to the endpoints to run the demo script. That's done by using the `. /vagrant/openrc` command:



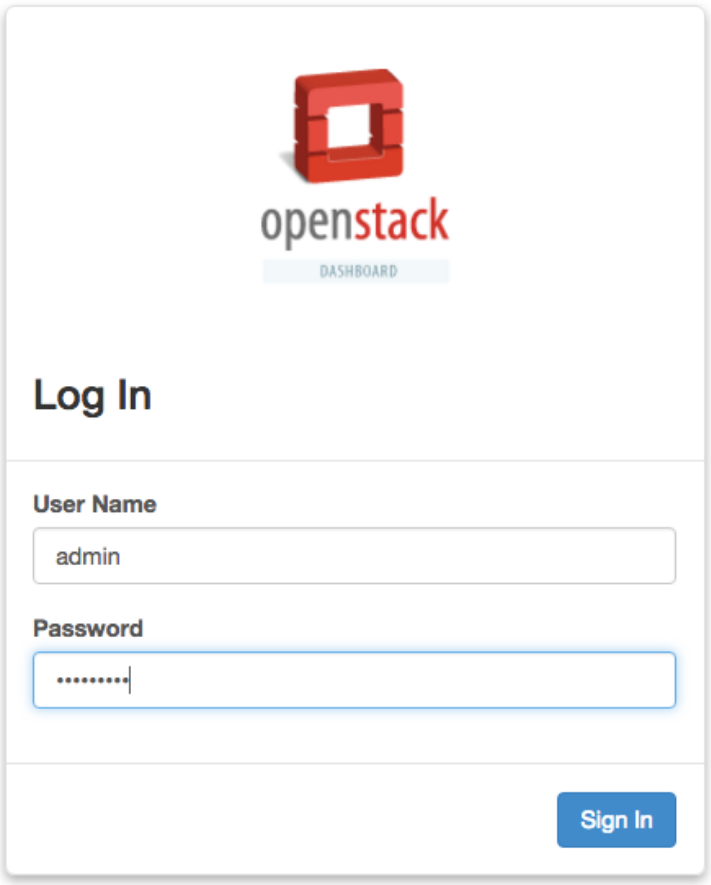
The last step is to run the demo script which is located in the /vagrant/ folder that's mapped to your local system with the `/vagrant/demo.sh` command. You will see a bunch of content roll by on the screen and you can scroll up to see all of the commands that were run and the results:



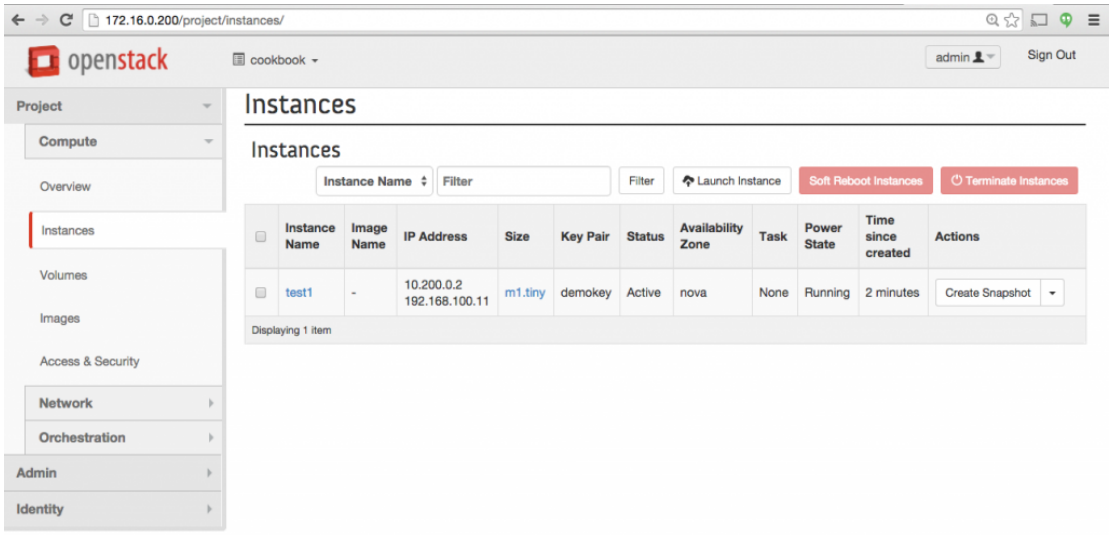
Results will complete with the final command showing an associated floating IP:



Now you open up your browser to access the Horizon dashboard at http://172.16.0.200 using the default credentials for the lab which is a username of **admin** and a password of **openstack** to log in:



Click on the **Projects** tab in the left hand pane and then the **Instances** menu option to see the active instance that was spun up with the demo script:



You now have a two-node OpenStack on KVM lab running and you can dig around to see all of the various features and functionality available.

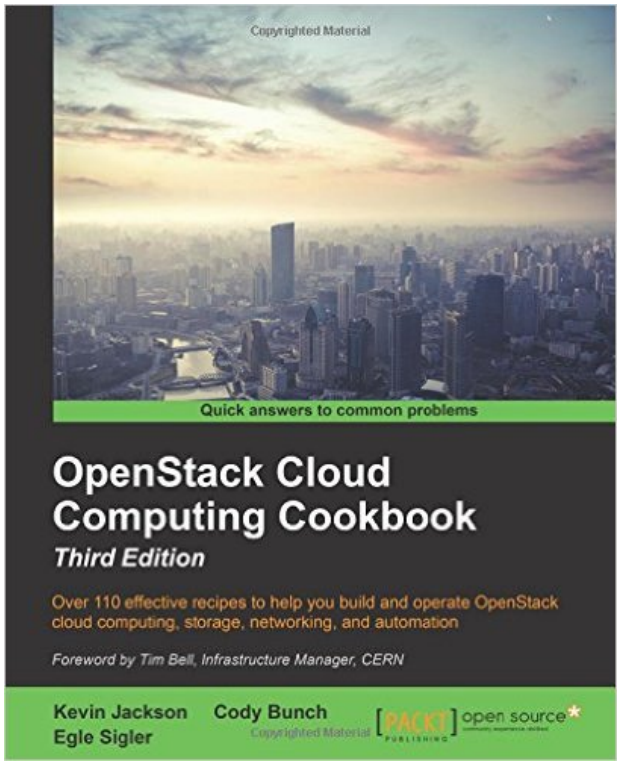
More OpenStack Cookbook Resources

I will be doing some other posts to show a few nifty ways to use the OpenStack Cookbook lab to learn about the various project features.

Read here on how to suspend and resume your OpenStack Cookbook lab using Vagrant suspend and

awk.

That said, there is nothing that makes a better companion to your OpenStack Cookbook lab than the actual guide written by Cody Bunch, Kevin Jackson, and Egle Sigler, which is available here: http://www.amazon.com/OpenStack-Cloud-Computing-Cookbook-Third/dp/1782174788/ref=sr_1_1?ie=UTF8&qid=1448141088&sr=8-1&keywords=openstack+cookbook



You can also read through some great posts at the home site for the book at http://www.openstackcookbook.com

Happy Stacking!

November 2015 London/UK VMUG Recap

Having just finished up a trans-Atlantic trip, I am reminded again about how important the community is. The theme of the VMUG may be changing as we watch the progression in technology, but the core importance has not changed, and if anything, has strengthened.

Meeting of the Mentors

A few folks in the community stand out personally for me from the experiences that I've had. It wasn't that long ago that we were lucky enough to have Mike Laverick come to speak at the Toronto VMUG on the concepts of cloud and how it is related to traditional virtualization. That was over 3 years ago when I was just a member of the Toronto VMUG. I had the pleasure of having dinner with Angelo Luciani and Mike Laverick while Mike was in town, and he was very kind to share stories and ideas on how to advance careers in IT.

Those words stuck, as did the impact of what Mike has done for the community. It made it very special to finally come and to now be a guest speaker at a VMUG in the home VMUG of Mike and the team.

Josh Atwell and John Mark Troyer also flew in to speak at the event, both of whom have been excellent personal mentors to me and many in the community. I had a chance to meet so many of the folks I have been reading content from for years, and despite being a few thousand kilometers from where I started the journey, everyone in the UK VMUG made it feel like I was right at home.

OpenStack at the VMUG?

Not only did I have the pleasure of delivering a session in the main room about OpenStack and how to embrace the open cloud as part of our IT portfolio, but I was able to also sit with a great group of folks for a round-table session on containers and OpenStack in the afternoon before delivering my session for Turbonomic.

The crowd was very keen to listen and to ask questions. There is no doubt that OpenStack, while seemingly off topic at a VMUG, was quite important for the future of our careers. I always say that before you dismiss OpenStack, it is as important to fully understand why it may not work for the organization as it is to understand why your organization may need it.

> Open Source IS fundamentally good – @discoposse #UKVMUG pic.twitter.com/2y3AmYCNg1
>
> — Ather Beg (@AtherBeg) November 19, 2015

During the containers round-table, I had a chance to hear various thoughts from folks on how they think containers will impact their roles, and companies who are embracing the technology. The group came with a variety of backgrounds and skill levels, and the chat was superb as we ran right up to the finish of the session having seen that we could clearly have spent the day learning and discussing much more on the subject. The key moment I enjoyed was when I asked everyone what they thought the skill that they need to set as a learning priority, and the answers from all around the table were OpenStack, and only one who said “OpenStack, but for my direct work I’m doing today, I need to ramp up on configuration management tools like Ansible, Puppet, Chef and Salt”. It’s a proud moment when you see that all the work that you do gets validated as the community really latches on to the work that we have been doing to help open up everyones opportunities to learn.

Changing of the Guard

It was rather touching to see the final moments of the event. Congratulations were being shared all around from a great day of sharing and learning, which led to a slide that VMUG co-leader Alaric Davies brought up titled “UKVMUG: The Red Wedding Edition”. In a speech that was mixed with humor and touching moments, we learned that three of the four co-leaders are stepping down from the committee.

> Your #UKVMUG committee welcomes you! pic.twitter.com/cUv5R3QPXg — Jane Rimmer (@Rimmergram) November 19, 2015

Having been able to get to know Alaric Davies and Jane Rimmer personally, and to have met Stuart Thompson at the event, it was clear that their impact on the community has been amazing, and I have no doubt that Simon Gallagher will be filling up the leadership committee with other great advocates from the group.

If you are interested in helping to lead the UK VMUG, I encourage you to take the step of becoming a leader. It is a very special experience, and one that will give both a personal and professional boost to you as you have an opportunity to engage with, and give guidance to this great technology community that we have grown.

Visit http://bit.do/lonvmug to sign up for a leadership role and find out more about what's next for the upcoming UK VMUG events.

Thank You Alaric, Jane, Stuart, and Simon, for leading the VMUG event. Thanks as well to the VMUG head office team, plus Jean from VMware, all of whom have continued to drive the community and give a special experience to me and all of the attendees.

See you all at the next one, hopefully!

p.s. Don't stay at the Travelodge Birmingham Airport if you attend the event. It's like a penitentiary except without the charm.

OpenStack by the Numbers: Welcome Kilo!

As April 30th arrived, so did the next named release of OpenStack. Kilo is the 11th official release of OpenStack since its inception in 2010. Born of the with great hopes and minds at Rackspace and NASA, the widely known open cloud ecosystem continues to gain steam, awareness, and sometimes a little bit of criticism. Whether you're using OpenStack today or investigating it for down the road, this is a good time to take a quick look at what went into the Kilo release.

By the People, For the People

Not even taking into account the corporate support and customer side of the equation, I always like to see how the developer and operations side of OpenStack fares as each release comes out. One thing that is undeniable is that the velocity continues to increase for OpenStack development, and the stability is also doing the same. The program list (aka project list…and probably both because that keeps changing) is widening with the official integration of Ironic, a bare metal provisioning project.

Features are growing rapidly also, with a lot of fixes and enhancements that have been wrapped into this release.

Juno in Review

How does 18,992 code commits sound? That's pretty impressive if you ask me, and what is just as impressive is if you look at how we assessed Juno in the past here.

So, with Juno in the bank, how about we look at how Kilo measured up by these numbers.

Kilo Kicking it up a Notch

As we took a look at the analytics from activity.openstack.org, the numbers were both impressive and telling. 21,125 code commits from 1,593 developers and a total of 1,839 submitters which covers code, documentation, and training.

Looking at Juno in comparison shows that we have a continued rise above a linear scale with the Kilo release. This is telling as we can see the velocity of development and the growth of the ecosystem which is increasing at each release.

If we look at the picture since the beginning, from 124,499 total code commits, 3,279 total developers, and 3,611 total submitters, it tells us that Kilo produces 16.96% of the total code in just the Kilo release alone!

Code counting doesn't tell you success stories. What does show a positive story in these numbers is that they keep moving up and to the right. Increasing both on participation, fixes, features, and total growth.

OpenStack Kilo Presentation

As the launch press makes its way around, we will see a lot of good detail on features, advancements, and also on challenges. One thing I do appreciate about a lot of the OpenStack community is that they acknowledge that the challenges are real, and are being taken on as much as possible.

Here is a quick view of the Kilo release as presented by the OpenStack Foundation:

OpenStack Kilo – April 2015 from **OpenStack Foundation**

Eric's Favorite Features

There are lots of very interesting features and improvements that were packed into the Kilo development cycle. According to the OpenStack Foundation there were 394 to be exact. I don't have an exhaustive list, but I can tell you a couple of things that I am particularly keen on that arrived with Kilo.

Keystone-to-Keystone Federation

Identity federation was available in previous releases, but was well known to be challenging to work with and very new. The Kilo release of Keystone has seen the federation features listed as stable with much more documentation and working examples of how to federate your OpenStack clouds to one another. As people explored OpenStack without federation, the issue rose that if more than one OpenStack cloud was implemented that the identity environments were separated, and thus difficult to manage.

By adding stable Keystone to Keystone federation we have the ability to create a true hybrid OpenStack cloud. This is also important as companies are embracing OpenStack and the potential for two organizations to have to merge during a purchase or partnership was a bit limiter up to this point.

Ironic for VirtualBox (experimental)

Although this is an experimental feature within the Ironic (bare-metal provisioning) program, the availability of a way to test out Ironic deployment for people is going to be very helpful as we get more folks exploring OpenStack. It's one thing to have a working OpenStack environment, and a whole other thing to have procedures and methods to build, deploy, and grow the infrastructure. Using VirtualBox as the test bed for Ironic will allow more people to test the waters and provide feedback to the development teams. This can only lead to good things.

Scale Preparation for Neutron

This is a work in progress, but it is top of mind as we head into the Liberty release. Networking within the OpenStack environment has often been shown to have scale challenges when growing in large environments. Many won't hit any of the scale limits that are currently a challenge, but it is important just the same that the teams are working to address this and have done lots of work to build in future improvements as upcoming releases come out.

People

Yes, people are a feature. This includes a growing community, an eager developer ecosystem, and the most important thing of all which is people using OpenStack. The focus on the operator and what is called the "superuser" has been big with the last couple of OpenStack Summits, and with good reason. The more people who are using OpenStack in some form, the better the overall ecosystem will do.

Looking forward to an equally exciting Liberty release, and I hope to see lots of you in Vancouver at the OpenStack Summit!