

# [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:

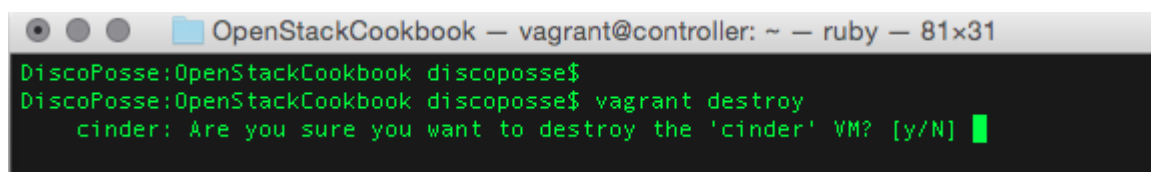
<https://discoposse.com/2014/10/12/suspend-and-resume-vagrant-boxes-in-bulk-osx-linux/> which will give you the ability to quiesce and to spin up your OpenStack Cookbook lab easily.

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**

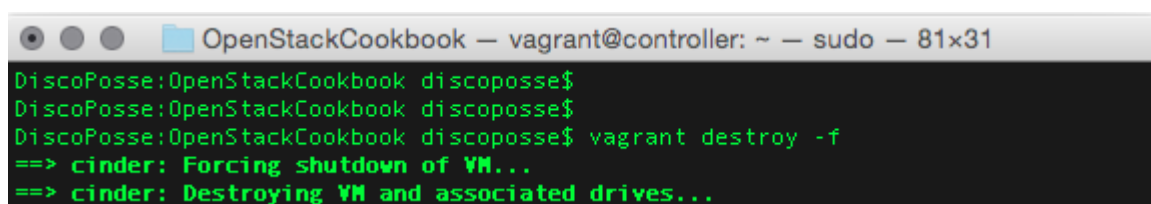
There may be events that require you to remove the OpenStack Cookbook lab from your host. This could be to update the code, test some parts of the build, or it could be to correct an issue that may have occurred which is easier to recover from by rebuilding the lab in its entirety.

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:



```
OpenStackCookbook — vagrant@controller: ~ — ruby — 81x31
DiscoPosse:OpenStackCookbook discoposse$
DiscoPosse:OpenStackCookbook discoposse$ vagrant destroy
cinder: Are you sure you want to destroy the 'cinder' VM? [y/N]
```

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:



```
OpenStackCookbook — vagrant@controller: ~ — sudo — 81x31
DiscoPosse:OpenStackCookbook discoposse$
DiscoPosse:OpenStackCookbook discoposse$
DiscoPosse:OpenStackCookbook discoposse$ vagrant destroy -f
==> cinder: Forcing shutdown of VM...
==> cinder: Destroying VM and associated drives...
```

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:

```
OpenStackCookbook — vagrant@controller: ~ — sudo — 81x31
DiscoPosse:OpenStackCookbook discoposse$
DiscoPosse:OpenStackCookbook discoposse$
DiscoPosse:OpenStackCookbook discoposse$ vagrant destroy -f
==> cinder: Forcing shutdown of VM...
==> cinder: Destroying VM and associated drives...
==> cinder: Pruning invalid NFS exports. Administrator privileges will be require
d...
Password:
```

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

```
OpenStackCookbook — vagrant@controller: ~ — bash — 81x31
DiscoPosse:OpenStackCookbook discoposse$
DiscoPosse:OpenStackCookbook discoposse$
DiscoPosse:OpenStackCookbook discoposse$ vagrant destroy -f
==> cinder: Forcing shutdown of VM...
==> cinder: Destroying VM and associated drives...
==> cinder: Pruning invalid NFS exports. Administrator privileges will be require
d...
Password:
==> cinder: Running cleanup tasks for 'shell' provisioner...
==> compute-01: Forcing shutdown of VM...
==> compute-01: Destroying VM and associated drives...
==> compute-01: Pruning invalid NFS exports. Administrator privileges will be req
uired...
==> compute-01: Running cleanup tasks for 'shell' provisioner...
==> network: Forcing shutdown of VM...
==> network: Destroying VM and associated drives...
==> network: Pruning invalid NFS exports. Administrator privileges will be requir
ed...
==> network: Running cleanup tasks for 'shell' provisioner...
==> controller: Forcing shutdown of VM...
==> controller: Destroying VM and associated drives...
==> controller: Pruning invalid NFS exports. Administrator privileges will be req
uired...
==> controller: Running cleanup tasks for 'shell' provisioner...
DiscoPosse:OpenStackCookbook discoposse$
```

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

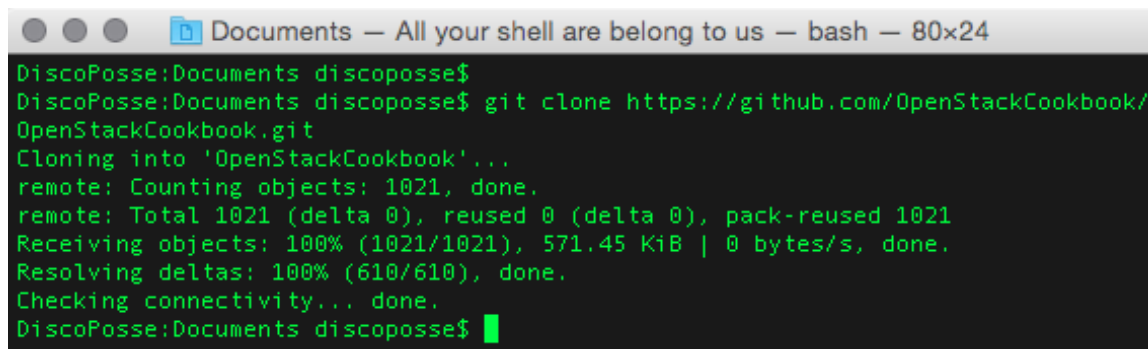
I highlight internet access because you can run the cookbook lab without internet access, but only once it is fully downloaded and built. The lab works great for me during testing when on the road and while disconnected, but we do run a live download of the code and base machines during the setup process.

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/>

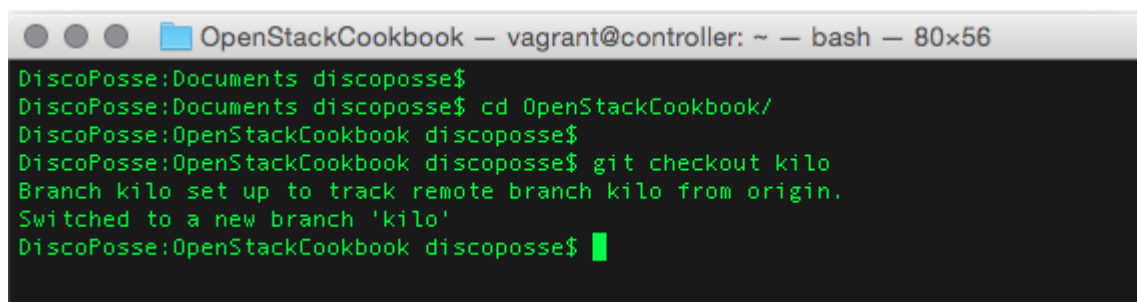
## Downloading the OpenStack Cookbook code

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



```
Documents — All your shell are belong to us — bash — 80x24
DiscoPosse:Documents discoposse$
DiscoPosse:Documents discoposse$ git clone https://github.com/OpenStackCookbook/
OpenStackCookbook.git
Cloning into 'OpenStackCookbook'...
remote: Counting objects: 1021, done.
remote: Total 1021 (delta 0), reused 0 (delta 0), pack-reused 1021
Receiving objects: 100% (1021/1021), 571.45 KiB | 0 bytes/s, done.
Resolving deltas: 100% (610/610), done.
Checking connectivity... done.
DiscoPosse:Documents discoposse$
```

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:



```
OpenStackCookbook — vagrant@controller: ~ — bash — 80x56
DiscoPosse:Documents discoposse$
DiscoPosse:Documents discoposse$ cd OpenStackCookbook/
DiscoPosse:OpenStackCookbook discoposse$
DiscoPosse:OpenStackCookbook discoposse$ git checkout kilo
Branch kilo set up to track remote branch kilo from origin.
Switched to a new branch 'kilo'
DiscoPosse:OpenStackCookbook discoposse$
```

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:

```
OpenStackCookbook — vagrant@controller: ~ — VBoxManage — 80x56
DiscoPosse:OpenStackCookbook discoposse$
DiscoPosse:OpenStackCookbook discoposse$ vagrant up --provider=virtualbox
Bringing machine 'controller' up with 'virtualbox' provider...
Bringing machine 'network' up with 'virtualbox' provider...
Bringing machine 'compute-01' up with 'virtualbox' provider...
Bringing machine 'cinder' up with 'virtualbox' provider...
==> controller: Importing base box 'bunchc/trusty-x64'...
```

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:

```
OpenStackCookbook — vagrant@controller: ~ — sudo — 80x15
controller: shared folder errors, please make sure the guest additions with
n the
controller: virtual machine match the version of VirtualBox you have install
ed on
controller: your host and reload your VM.
controller:
controller: Guest Additions Version: 4.3.10
controller: VirtualBox Version: 5.0
==> controller: Setting hostname...
==> controller: Configuring and enabling network interfaces...
==> controller: Installing NFS client...
==> controller: Exporting NFS shared folders...
==> controller: Preparing to edit /etc/exports. Administrator privileges will be
required...
Password:
```

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:

```
OpenStackCookbook — vagrant@controller: ~ — bash — 80x15
==> cinder: cinder-volume start/running, process 18148
==> cinder: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDYU1+Podv4PhBuLeW5Iuyt57sc/rns
RjhCye7AsATXIK4gc04YNpwLLhLmp7vREAKz6+MS8Cy2ac95kt51RoBhn8LAzjI8REDxxDmAKE6/B1tT
HndoWC7fhotbFb8DPqRbiJBft5QEFn1It/HQv1PPbpS321ttQ5cZrkVpg/qp+bUL2uGAjja5VeajIMhv
k/vddAx8MlrqDJBDIBDdU1p98iqUczBV80u+E1LLjQXNg+ayYahgtuT65L/KTgtA9xXQfkZsWb9saPuP
ioqFfZE6/a40yHPucMZSQDE9Z3BEcRvLz8txw/5LDm0vFM+1iby3so0nINUyISHsKGVhP2sr root@co
ntroller
==> cinder: rsyslog stop/waiting
==> cinder: cp:
==> cinder: cannot stat '/vagrant/rsyslog.conf'
==> cinder: : No such file or directory
==> cinder: stop: Unknown instance:
==> cinder: rsyslog start/running, process 18177
==> cinder: Configuring cache buckets...
DiscoPosse:OpenStackCookbook discoposse$
```

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:

```
OpenStackCookbook — vagrant@controller: ~ — ssh — 81x15
DiscoPosse:OpenStackCookbook disco posse$
DiscoPosse:OpenStackCookbook disco posse$ vagrant ssh controller
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-39-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Fri Nov 14 12:55:33 2014 from 10.0.2.2
vagrant@controller:~$
```

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:

```
OpenStackCookbook — vagrant@controller: ~ — ssh — 81x15
DiscoPosse:OpenStackCookbook disco posse$
DiscoPosse:OpenStackCookbook disco posse$ vagrant ssh controller
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-39-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Fri Nov 14 12:55:33 2014 from 10.0.2.2
vagrant@controller:~$ . /vagrant/openrc
vagrant@controller:~$
vagrant@controller:~$
```

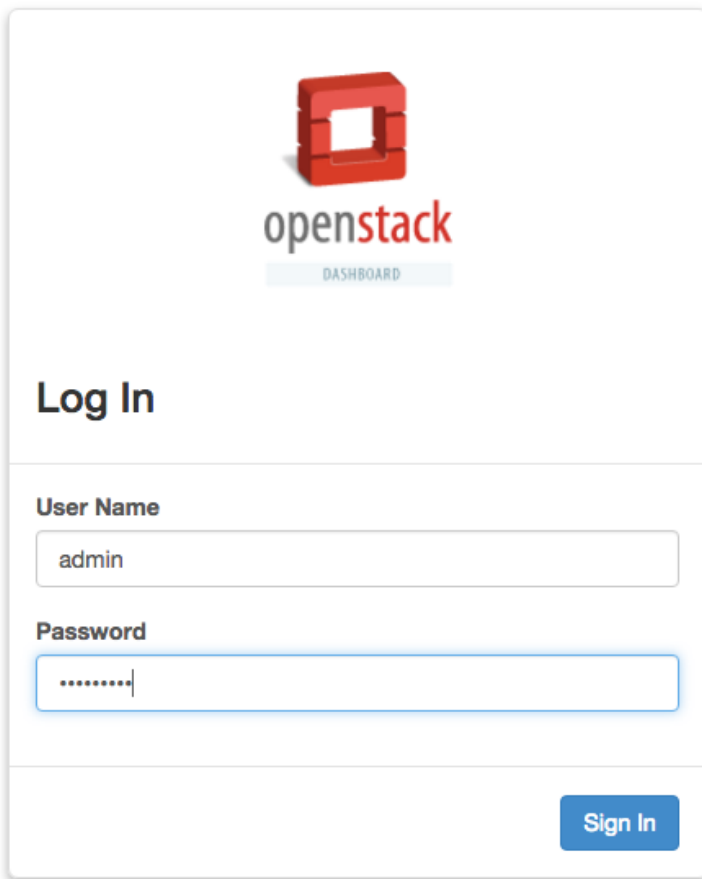
The last step is to run the demo script which is located in the `/vagrant/` folder that's mapped to your local system with the `/vagrant/demo.sh` command. You will see a bunch of content roll by on the screen and you can scroll up to see all of the commands that were run and the results:

```
OpenStackCookbook — vagrant@controller: ~ — ssh — 81x31
vagrant@controller:~$
vagrant@controller:~$ /vagrant/demo.sh
```

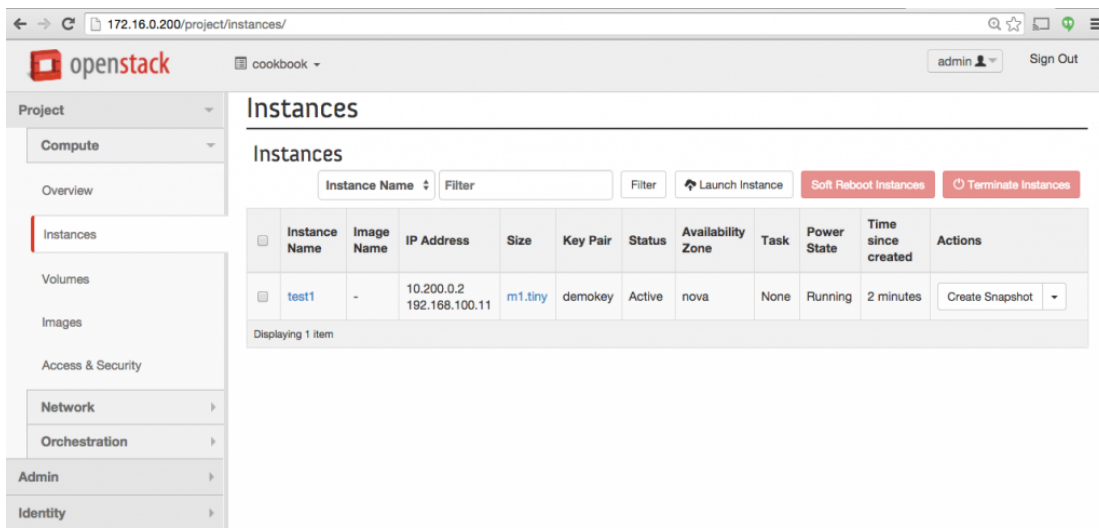
Results will complete with the final command showing an associated floating IP:

```
OpenStackCookbook — vagrant@controller: ~ — ssh — 81x31
+-----+
| allocation_pools | {"start": "192.168.100.10", "end": "192.168.100.20"} |
| cidr              | 192.168.100.0/24 |
| dns_nameservers  | |
| enable_dhcp      | False |
| gateway_ip       | 192.168.100.1 |
| host_routes      | |
| id               | f45108d6-43c2-4dfe-bb6f-c17fae153a13 |
| ip_version       | 4 |
| ipv6_address_mode | |
| ipv6_ra_mode     | |
| name             | cookbook_float_subnet_1 |
| network_id       | 8ee71c99-9baf-4d85-8c6e-b906736435d8 |
| tenant_id        | 8098afaf6f9243e9ac21b76835f6f0ec |
+-----+
Set gateway for router 0554c240-969e-41ee-a608-65a60e376c1f
Created a new floatingip:
+-----+
| Field            | Value |
+-----+
| fixed_ip_address | |
| floating_ip_address | 192.168.100.11 |
| floating_network_id | 8ee71c99-9baf-4d85-8c6e-b906736435d8 |
| id               | 1e26ec40-7fcd-4e51-b660-528b163d98a0 |
| port_id          | |
| router_id        | |
| status           | DOWN |
| tenant_id        | 8098afaf6f9243e9ac21b76835f6f0ec |
+-----+
Associated floating IP 1e26ec40-7fcd-4e51-b660-528b163d98a0
vagrant@controller:~$
```

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

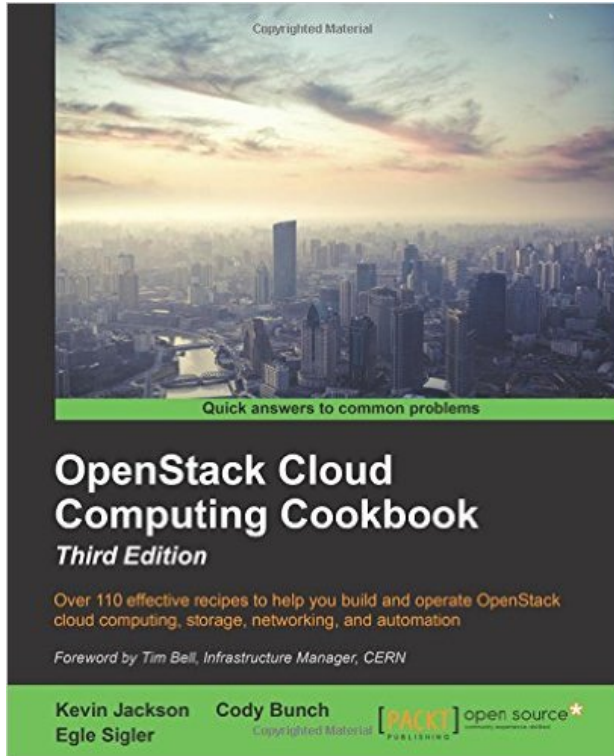
I will be doing some other posts to show a few nifty ways to use the OpenStack Cookbook lab to learn about the various project features.

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](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!