Life Without DevStack: 
OpenStack Development With OSA

Miguel Grinberg
@miguelgrinberg
About Me

● Software Developer with Rackspace
  ○ Rackspace Private Cloud
● OpenStack Contributor
  ○ openstack-ansible, heat, keystone, horizon
● O’Reilly Author
  ○ Flask Web Development
● Blogger
  ○ http://blog.miguelgrinberg.com
Agenda

- What’s Wrong with DevStack?
- What is openstack-ansible (OSA)?
- Installation
- Development Workflow
- Demonstration
- Tips, Tricks & Gotchas
- Pros & Cons
What’s Wrong with DevStack?

- Monolithic install
- Not possible to modify or reinstall individual services
- All packages are installed for development, not just the one you need.
- Services run in a shared environment
- Dependencies are shared
What is openstack-ansible (OSA)?

- Ansible playbooks to deploy OpenStack
- Deploys production-ready clouds from a handful of nodes to hundreds or even thousands
- Services are deployed from the git repositories
- No vendor-specific patches or add-ons
- Services are isolated from each other in LXC containers
OSA Architecture
OSA All-In-One

- Used by the openstack-ansible team for development, testing and gating
- Can be deployed on a cloud server or virtual machine
- Requirements
  - 16GB RAM
  - 80GB disk space
  - Ubuntu 14.04
- Redundant services are supported
  - Easy to do when services are isolated in containers!
  - HAProxy is installed on the host to load balance
OSA All-In-One Installation

- Install on a fresh Ubuntu 14.04 host

```
# git clone https://github.com/openstack/openstack-ansible \\
   /opt/openstack-ansible
# cd /opt/openstack-ansible
# scripts/bootstrap-aio.sh
# vi /etc/openstack_deploy/...
# scripts/bootstrap-ansible.sh
# scripts/run-playbooks.sh
```

← Set up host and config files
← Customize configuration (optional)
← Install Ansible
← Install or update OpenStack
LXC Cheat Sheet

# list all containers in a host
lxc-ls [--fancy|-f]

# access a container’s console
lxc-attach -n <container-name>

# start a container as a daemon
lxc-start -d -n <container-name>

# stop and destroy a container
lxc-stop -n <container-name>
lxc-destroy -n <container-name>
Development Workflow

1. Deploy OSA-AIO
2. Attach to the target container
3. Stop the target service
4. Clone development version
5. Update dependencies
6. Sync database
7. Make changes to the original config files, if necessary
8. Run manually, or install and run as a service
Demonstration

https://asciinema.org/a/28461
Where is Everything

- The container file systems are accessible in the host at 
  `/var/lib/lxc/<container-name>/rootfs`
- Logs are in the host at `/openstack/log/<container-name>`
- Each container has its virtual environments in `/openstack/venvs`
- HAProxy is running on the host
  - The service configuration files are in `/etc/haproxy/conf.d/*`
Tips, Tricks & Gotchas

● Python packages and the cheese shop
  ○ openstack-ansible creates its own private package repository
  ○ To enable access to external packages, disable option `no-index` in `~/.pip/pip.conf`

● Using debuggers
  ○ Terminal based debuggers work great inside containers (pudb, pdb)
  ○ Remote debugging from host to container works too (pycharm, pydev)
• Fix Apache configuration (keystone and horizon only)
  ○ To run the service from the home directory, add
    `home=/root/<project>` to the WSGI setup in the apache site
    configuration file
• Database migrations don’t have downgrade
  ○ Backup and restore the database if you need to downgrade
  ○ Or delete the database, and let the playbooks create a new one
• Galera restarts are tricky
  ○ See documentation for restart instructions
OSA-AIO: The Pros

- Less dependency headaches
- Easy to partially update a live system
- Easy to restore a dead or sick system
- Closer to the real thing
OSA-AIO: The Cons

- Only core services supported at this time
  - nova, neutron, keystone, glance, cinder, swift, heat, ceilometer, horizon
  - We welcome contributions with open arms!
- Strongly opinionated networking
  - Linux Bridge or bust!
Help Wanted

We welcome new contributors!
Talk to us on freenode: #openstack-ansible
Thank You

(Read my blog post: http://bit.ly/osa-devel)