



# From How-To to POC to Production: Learning by Building

*Presented By: Grant Kirkwood, CTO – Unitas Global*

*March 4, 2017 – SCALE 15x – Pasadena, Calif*



## WHAT IS OPENSTACK?

---

- “Science experiment”
- “A mess of competing projects and priorities”
- “De facto choice for new private clouds”

**ONE DOES NOT SIMPLY**

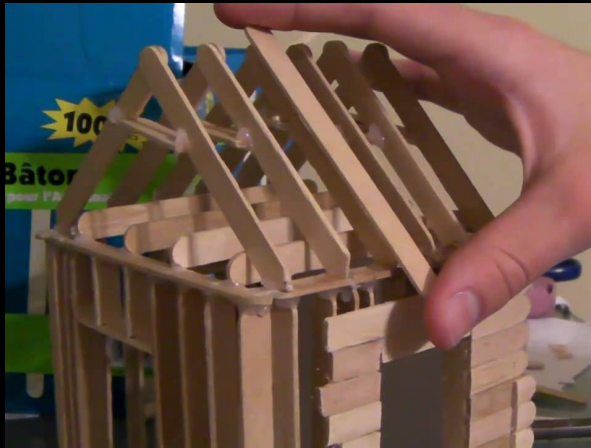
**LEARN OPENSTACK**



**FROM  
HOW-TO:**

**TO POC:**

**TO  
PRODUCTION!**



***LEARNING BY BUILDING...***

# STEP 1:





## WHAT IS IT?

---

- Scripted install
- All-in-one
- Up and running in minutes



# DEVSTACK

## WHAT YOU NEED

---

- Laptop!
- VirtualBox
- Ubuntu
- DevStack



VirtualBox





## DOWNLOAD

---

- **VirtualBox:**  
<https://www.virtualbox.org/wiki/Downloads>
- **Ubuntu:**  
<http://releases.ubuntu.com/16.04/ubuntu-16.04.10-server-amd64.iso>





## INSTRUCTIONS

---

- Install VirtualBox
- Create an Ubuntu VM
- Install DevStack
- Help found here:  
<http://ronaldbradford.com/blog/downloading-and-installing-devstack-2016-04-02/>



# DEVSTACK

## Downloading and installing devstack

April 2, 2016 by [ronald](#)

The following instructions assume you have a running Linux virtual machine that can support the installation of [devstack](#) to demonstrate a simple working [OpenStack](#) cloud.

For more information about the preparation needed for this step, see these pre-requisite instructions:

- [Installing VirtualBox](#)
- [Setting up an Ubuntu virtual machine using VirtualBox or Setting up Ubuntu using vagrant](#)

### Pre-requisites

You will need to login to your Linux virtual machine as a normal user (e.g. **stack** if you followed [these instructions](#)).

To verify the IP address of your machine you can run:

```
$ ifconfig eth1
```

NOTE: This assumes you configured a second network adapter as [detailed](#).

You need to determine the IP address assigned. If this is your first-time using VirtualBox and this was configured with default settings, the value will be **192.168.56.101**

```
eth1      Link encap:Ethernet  HWaddr 08:00:27:db:42:6e
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fedb:426e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:398500 errors:0 dropped:0 overruns:0 frame:0
          TX packets:282829 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

- Step by step instructions here
- Tested.... it works!
- Thanks Ronald Bradford

<http://ronaldbradford.com/blog/downloading-and-installing-devstack-2016-04-02/>



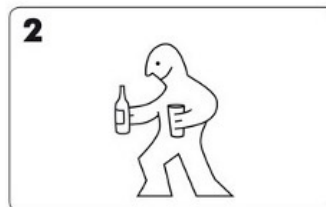
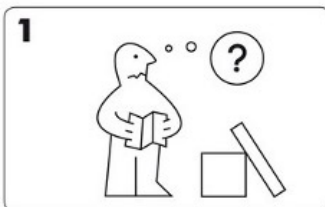
## WHAT WE LEARNED

---

- CLI basics
- How Horizon (etc) *should* work (and look)

# STEP 2:

## POÄNG



The IKEA assembly service. 0180/5 35 34 35.



# HOW-TO

## WHAT IS IT?

---

- Step-by-step walkthrough
- “Vanilla” / trunk-based
- Not scripted – each component built by hand



# HOW-TO

## WHAT YOU NEED

- Some hardware ideal
- Intel NUCs are good
- VLAN-capable switch
- Patience!





# HOW-TO

## DOWNLOAD

---

- **Ubuntu:**  
<http://releases.ubuntu.com/16.04/ubuntu-16.04.10-server-amd64.iso>
- **Documentation:**  
<http://docs.openstack.org/mitaka/install-guide-ubuntu/>
- **A good text editor:**  
<https://atom.io>
- **A note on versions...**



# HOW-TO

## DOWNLOAD

- Documentation:

1. Disable the keystone service from starting automatically after installation:

```
# echo "manual" > /etc/init/keystone.override
```

2. Run the following command to install the packages:

```
# apt-get install keystone apache2 libapache2-mod-wsgi
```

3. Edit the `/etc/keystone/keystone.conf` file and complete the following actions:

- In the `[DEFAULT]` section, define the value of the initial administration token:

```
[DEFAULT]
...
admin_token = ADMIN_TOKEN
```

Replace `ADMIN_TOKEN` with the random value that you generated in a previous step.

- In the `[database]` section, configure database access:

```
[database]
...
connection = mysql+pymysql://keystone:KEYSTONE_DBPASS@controller/keystone
```





# HOW-TO

## DOCUMENTING

---

- Why the good text editor?
- Documenting your work critical
- You WILL do things wrong
- You WILL want to do things differently next time around... yes next time



# HOW-TO

## DOCUMENTING

- Leave yourself some notes!
- Will save you grief later...
- And helps with learning
- ...AND will need for HA

```
296
297 ===== Keystone =====
298
299 # note that default location for keystone executables changes from kilo to liber
300
301 (controller1)
302 mysql -u root -p
303 CREATE DATABASE keystone;
304 GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY 'cef6
305 GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'cef61f1c4354
306 quit
307
308 (all controllers)
309 echo "manual" > /etc/init/keystone.override
310 apt-get install keystone python-openstackclient apache2 libapache2-mod-wsgi memc
311 cd /etc/keystone
312 cp keystone.conf keystone.conf-ORIG
313 touch keystone.conf
314 chown keystone:keystone keystone.conf
315 edit /etc/keystone/keystone.conf
316 [default]
317 admin_token = ADMIN_TOKEN
318 [database]
319 connection = mysql+pymysql://keystone:KEYSTONE_DBPASS@ha2/keystone
320 [memcache]
321 servers = localhost:11211
322 [token]
323 provider = uuid
324 driver = memcache
325 [revoke]
326 driver = sql
327 # see other items added CUSTOM CONFIG STUFF
328 # rabbitmq section, etc....
329 # rpc_backend = rabbit
330 service keystone restart
331
```



# HOW-TO

## INSTRUCTIONS

---

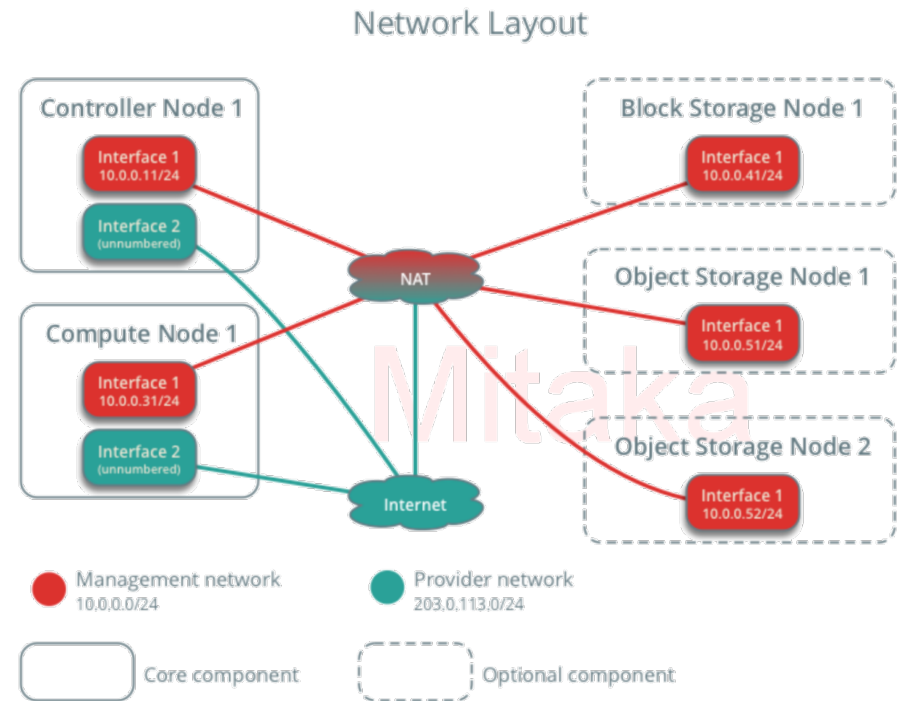
- Internet (2 networks)
- Switch
- 3+ Servers
- Architecture decisions in official how-to doc
- Use provider networks with self-service option



# HOW-TO

## INSTRUCTIONS

- Initial network setup
- Translate to home network...



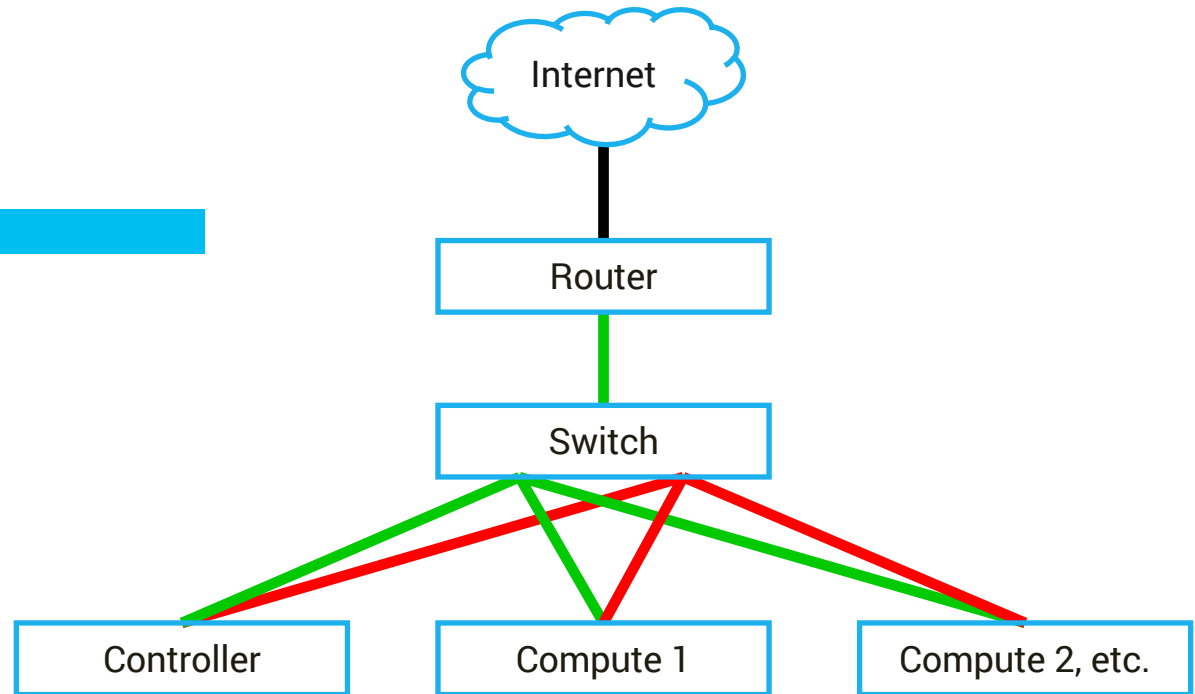
<http://docs.openstack.org/mitaka/install-guide-ubuntu/environment-networking.html>



# HOW-TO

## TOPOLOGY

- 192.168.10.0/24
  - Cut it in half...
- 10.0.10.0/24





# HOW-TO

## CORE COMPONENTS

---

- Contents from How-To Tutorial:
- Essential: Environment, Identity, Image, Compute, Networking, Dashboard, Block Storage

### Contents

- Conventions
- Overview
  - Example architecture
  - Networking
- Environment
- Identity service
- Image service
- Compute service
- Networking service
- Dashboard
- Block Storage service
- Shared File Systems service
- Object Storage service
- Orchestration service
- Telemetry service
- Database service
- Launch an instance
- Community support
- Glossary

# OPENSTACK TIME MACHINE





# HOW-TO

## LIFTOFF

---

- Time to launch your first VM...
- Good chance it won't work
- Where to look for clues
- Remember Devstack...

### Contents

- Conventions
- Overview
  - Example architecture
  - Networking
- Environment
- Identity service
- Image service
- Compute service
- Networking service
- Dashboard
- Block Storage service
- Shared File Systems service
- Object Storage service
- Orchestration service
- Telemetry service
- Database service
- Launch an instance
- Community support
- Glossary





# WHEN YOU (SUCCESSFULLY) LAUNCH YOUR FIRST VM...

## LOOKS LIKE THIS:

```
[ 0.545397] TCP cubic registered
[ 0.546185] NET: Registered protocol family 10
[ 0.547984] NET: Registered protocol family 17
[ 0.548888] Registering the dns_resolver key type
[ 0.549978] registered taskstats version 1
[ 0.553093]   Magic number: 0:485:1003
[ 0.553897] input input0: hash matches
[ 0.554849] vc vcs: hash matches
[ 0.555754] serial 00:05: hash matches
[ 0.556678] rtc_cmos 00:01: setting system clock to 2016-09-12 20:57:07 UTC (
1473713827)
[ 0.558484] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found
[ 0.559804] EDD information not available.
[ 0.677595] Freeing unused kernel memory: 928k freed
[ 0.679192] Write protecting the kernel read-only data: 12288k
[ 0.684365] Freeing unused kernel memory: 1596k freed
[ 0.688490] Freeing unused kernel memory: 1184k freed

further output written to /dev/ttyS0

login as 'cirros' user. default password: 'cubswin:)', use 'sudo' for root.
cirros login:
login as 'cirros' user. default password: 'cubswin:)', use 'sudo' for root.
cirros login: _
```

## FEELS LIKE THIS:





# HOW-TO

## WHAT WE LEARNED

---

- OpenStack under the hood
- How all the components work together
- Where to look for clues when something goes wrong
- CLI in-depth



# HOW-TO

## SHORTCOMINGS

---

- Not designed to scale
- Not optimized for performance
- Not built highly available
- Wouldn't use this in production



# HOW-TO

## NEXT STEPS

---

- Tear it down and start over! 😊
- Do it from your notes this time (how-to phone-a-friend)
- Fix the things you want to improve
- Make sure documentation complete – will need for POC



# HOW-TO

## NOT QUITE READY FOR POC...

---

- A proof of concept has to be built with business use in mind
- That means we have a bunch of new stuff to learn



# HOW-TO

## SCALABILITY & RELIABILITY

---

- Now the fun begins...
- First, time to upgrade storage: Ceph!
- Start by changing Glance to a Ceph backend
- Keep it simple – only change one thing at a time!
- Then change Cinder to use Ceph for volumes



# HOW-TO

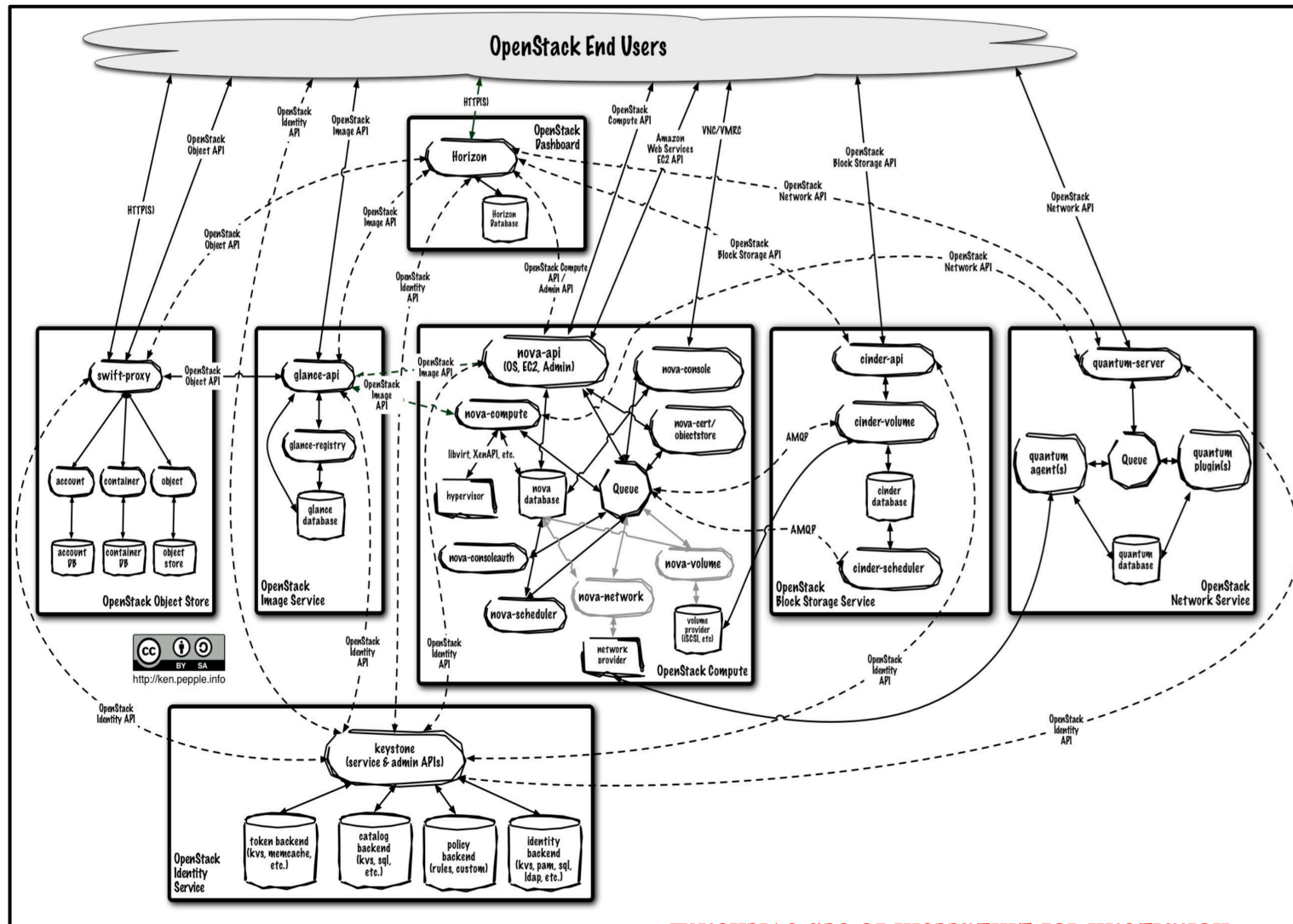
## SCALABILITY & RELIABILITY

---

- Building for high availability: wrapping services in more services... service burritos?
- What components do we use? Where do they fit?
- Can't really build on top of the existing config
- This is just one way to do it...



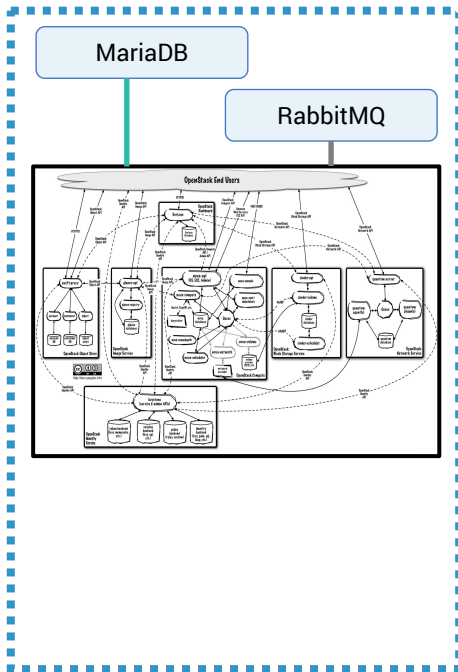
# HIGH AVAILABILITY





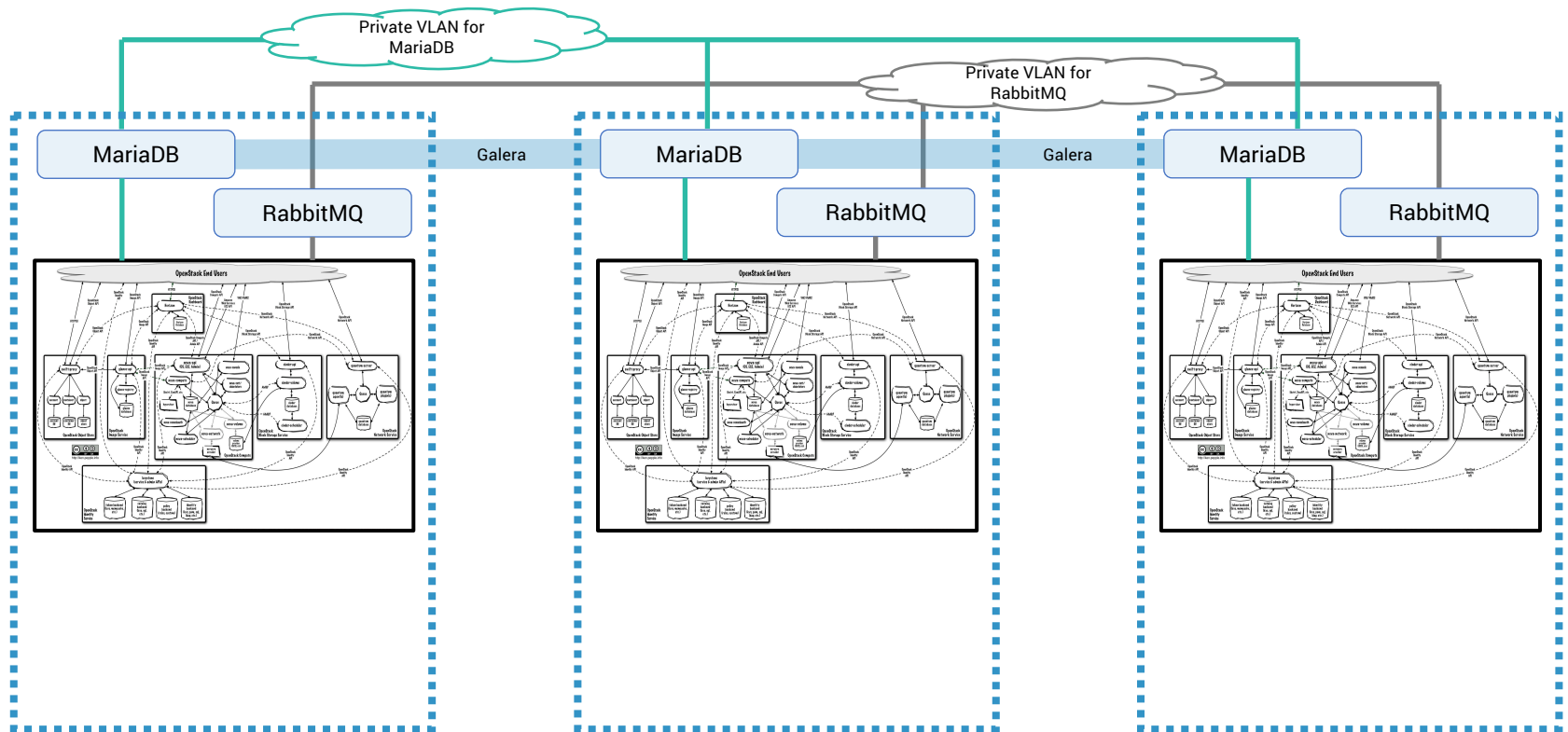


# HIGH AVAILABILITY



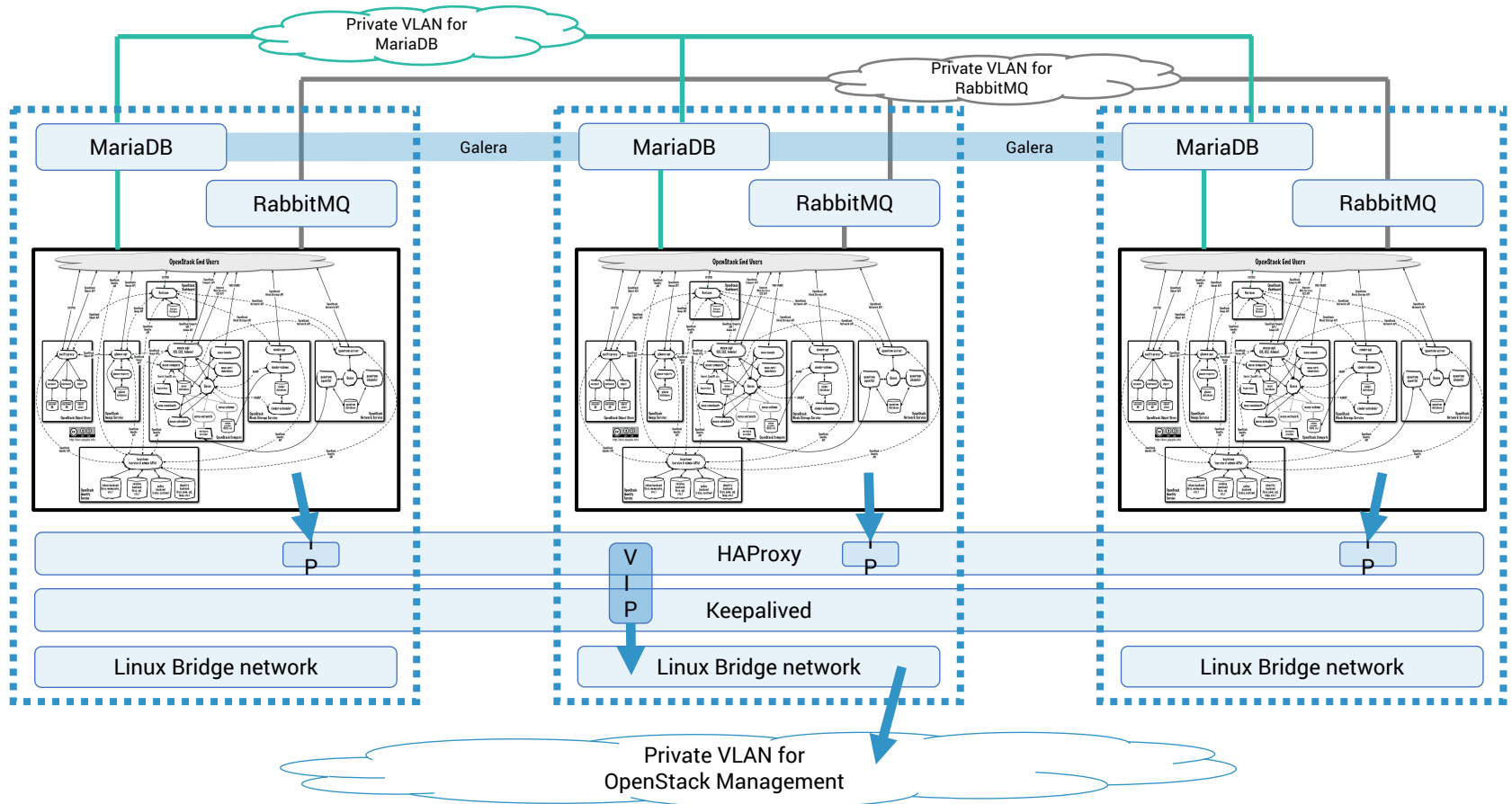


# HIGH AVAILABILITY





# HIGH AVAILABILITY





# HOW-TO

## SCALABILITY & RELIABILITY

---

- Ok, so that's the controllers!
- Now that we know how it fits together...
- Back to your build notes!



# HOW-TO

## SCALABILITY & RELIABILITY

---

- Notable changes – service endpoints
- Update hosts file with pointer to HA endpoint
- ...which means build the service endpoints to IPs that don't initially exist!
- Update haproxy as you go along




# HOW-TO

## SCALABILITY & RELIABILITY

- Example:

```
25
26 # NOTE: need to update things below with -ha version
27 openstack service create --name keystone --description "OpenStack Identity" id
28 openstack endpoint create --publicurl http://keystone-ha:5000/v2.0 --internalu
29 openstack project create --description "Admin Project" admin
30 openstack user create --password-prompt admin
31 openstack role create admin
32 openstack role add --project admin --user admin admin
33 openstack project create --description "Service Project" service
34 #openstack project create --description "Demo Project" demo
```





# HOW-TO

## TESTING

---

- Break it!
- Looping VM launching
- Power cycle controllers mid-cycle
- Haproxy stats helpful
- Tempest, Rally, etc.



# HOW-TO

## Statistics Report for pid 3740 on node1

### > General process information

pid = 3740 (process #4, nbproc = 4)  
 uptime = 184d 23h01m26s  
 system limits: memmax = unlimited; ulimit-n = 8075  
 maxsock = 8075; maxconn = 4000; maxpipes = 0  
 current conns = 295; current pipes = 0/0; conn rate = 0/sec  
 Running tasks: 1/334; idle = 100 %

■ active UP                      ■ backup UP  
■ active UP, going down       ■ backup UP, going down  
■ active DOWN, going up       ■ backup DOWN, going up  
■ active or backup DOWN       ■ not checked  
■ active or backup DOWN for maintenance (MAINT)  
■ active or backup SOFT STOPPED for maintenance  
 Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

stats																					
	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	R
Frontend				0	4	-	1	3	4 000	51			2 336 217	547 653 143	0	0	2				
Backend	0	0		0	2		0	1	400	8	0	0s	2 336 217	547 653 143	0	0		8	0	0	0

db																					
	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	R
Frontend				0	33	-	291	400	4 000	2 198 235			104 448 339 915	171 581 810 830	0	0	0	0	0	0	0
node1	0	0	-	0	33		291	400	-	2 198 151	2 198 151	5s	104 448 339 915	171 581 810 830	0	0	0	0	0	0	0
node2	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0
node3	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0
Backend	0	0		0	33		291	400	400	2 198 235	2 198 151	5s	104 448 339 915	171 581 810 830	0	0		84	0	0	0

keystone_admin																					
	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redi
Frontend				0	11	-	1	51	10 000	382 219			140 818 210	1 453 045 449	0	0	0				
node1	0	0	-	0	19		0	5	-	473 787	473 787	8s	140 818 210	1 453 045 449	0	0	0	0	0	0	0
node2	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0
node4	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0
Backend	0	0		0	19		0	5	1 000	473 787	473 787	8s	140 818 210	1 453 045 449	0	0		0	0	0	0

keystone_api																					
	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis
Frontend				0	11	-	0	8	10 000	415			158 591	668 805	0	0	53				
node1	0	0	-	0	11		0	7	-	551	551	1d17h	157 339	661 661	0	0	0	0	0	0	0
node2	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0
node4	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0
Backend	0	0		0	11		0	7	1 000	551	551	1d17h	158 591	668 805	0	0		0	0	0	0



# STEP 3:





# PROOF OF CONCEPT

## NOW THE FUN BEGINS

- You know how it all works now...  
but...
- Throw everything you just learned  
out the window!





# PROOF OF CONCEPT

## QUESTION:

---

- Are you in the business of building and operating technology infrastructure?



# PROOF OF CONCEPT

## DECISIONS

---

- Are we building for a business requirement?
- Must assume POC will be used  
(after all, if not, it wasn't a successful POC!)
- Therefore, build as if for production
- Don't want to be doing this by hand...



# PROOF OF CONCEPT

## DECISIONS

---

- Deployment method
- Distribution?  
(refer to question 1)
- Automation  
ie; OpenStack-Ansible
- Hardware



# PROOF OF CONCEPT

## BUSINESS REQUIREMENTS

---

- Define business requirements & success criteria
- Keep it simple to start!
- Each service adds complexity, whether or not used
- Conduct surveys: regimented/consistent process for adding new services



# PROOF OF CONCEPT

## GETTING STARTED

---

- Build small environment using distribution of choice now that you understand how it works inside (or non-distribution with OpenStack-Ansible)
- Test business applications in environment



# PROOF OF CONCEPT

## UPDATES & UPGRADES

---

- Brain surgery...



# STEP 4:





# PRODUCTION

## DEFINITION

---

- Supporting business-critical workload
- Downtime not an option
- Tied to revenue (and thus our jobs!)



# PRODUCTION

## OPERATIONAL REQUIREMENTS

---

- Monitoring
- Tracking
- Measuring performance
- Updates & upgrades



# PRODUCTION

## AUTOMATION

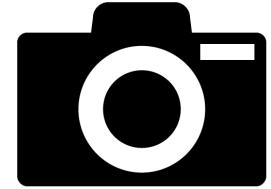
---

- Much of what you did is automated with major distributions
- Still critical to understand how it all works
- Serves as foundation for production grade operational excellence



# LEARNING BY BUILDING

## ROADMAP / SUMMARY



### Testing the waters with DevStack

- VirtualBox and a laptop
- Up and running in minutes
- Get comfortable with how it should work

### Build with How-To

- Build it by hand with [openstack.org](http://openstack.org) tutorial
- Learn how the pieces work together
- Document and take notes like crazy!

### High availability and scalability

- Upgrade storage to Ceph
- Make all services HA
- Try to break it

### Proof of Concept

- Define business objectives
- Desire to build/operate or consume?
- Switch to distribution
- Keep it simple, add services slowly/thoughtfully
- Test, break, break again

### Production

- Monitoring
- Tracking
- Updates & upgrades



# CONTACT



## Contact Us

### *Presenter*

Grant Kirkwood - CTO, Unitas Global  
grant.kirkwood@unitasglobal.com  
IRC: tnarg  
Twitter: @proximity

### *Visit Us!*

Booth #610 (on the left when you walk  
in – next to Disney)