redhat.

CHALLENGES IN CONTAINERIZING OPENSTACK

War stories of a Kolla veteran

Martin André
OpenStack Day France
November 22nd 2016

WHO AM I?

ABOUT ME

- Martin André
- Working on OpenStak deployment at Red Hat
- OpenStack contributor since Kilo
  - Long time Kolla contributor
  - Now focusing on TripleO
- Lived for a decade in 東京

THE THEORY

CREATING A DOCKER CONTAINER

It all starts from a Dockerfile:

```
FROM docker/whalesay:latest

RUN apt-get -y update && apt-get install -y fortunes

CMD /usr/games/fortune -a | cowsay
```

CREATING A DOCKER CONTAINER

Build the image:

```
$ docker build -t docker-whale .
Sending build context to Docker daemon 2.048 kB
...snip...
Removing intermediate container a8e6faa88df3
Successfully built 7d9495d03763
```

RUNNING A DOCKER CONTAINER

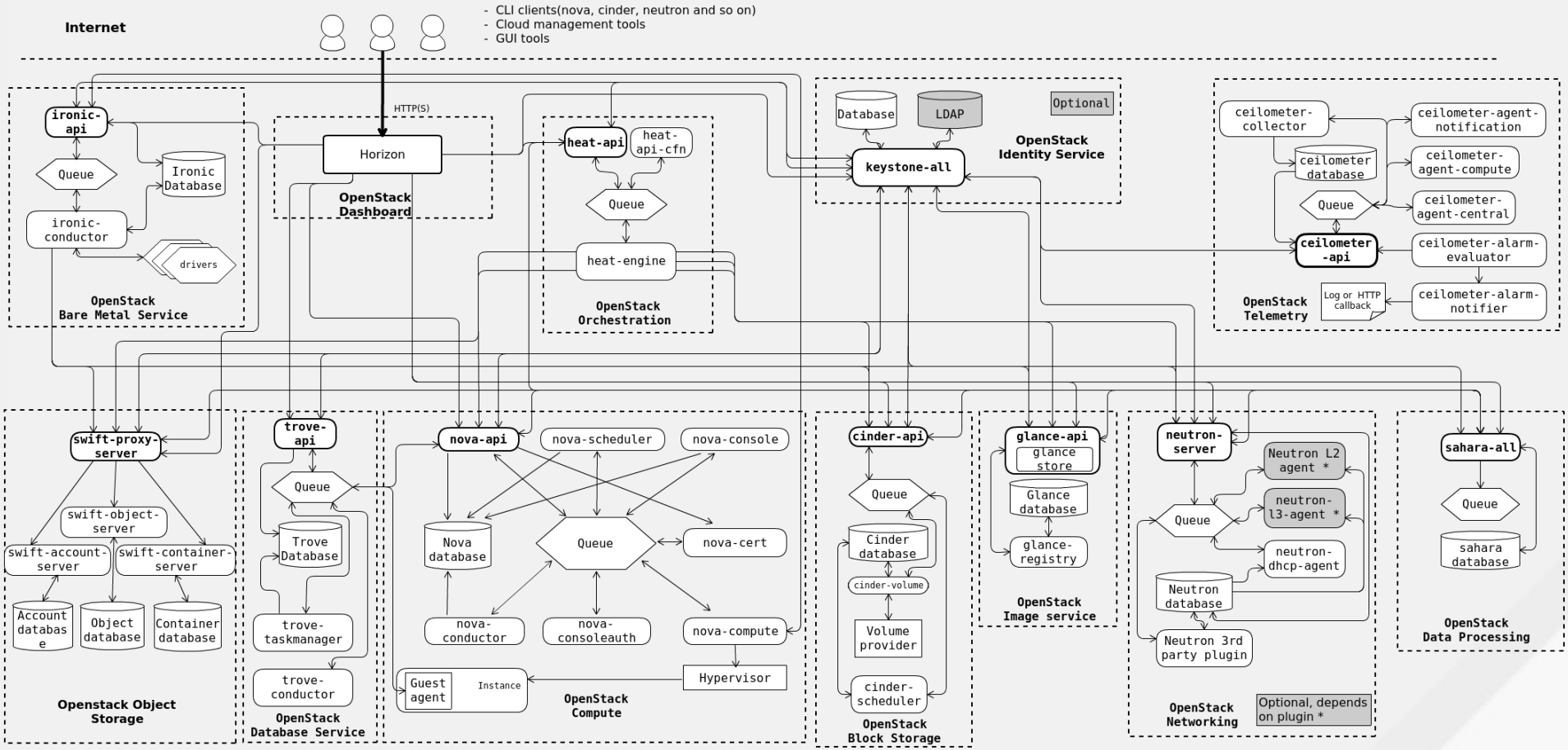
```
$ docker run docker-whale
 ----------------------------------------
/ "He was a modest, good-humored boy. It  \
\ was Oxford that made him insufferable." /
 ----------------------------------------
          \
           \
            \
                        ##         .
                  ## ## ##        ==
               ## ## ## ##       ===
           /""""""""""""""""___/ ===
      ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
           \______ o          __/
            \    \        __/
              \____\______/
```

TADAAAA!

YEAH, RIGHT...

NOW TELL ME HOW I CONTAINERIZE THIS?

MEET OPENSTACK



redhat.

1st CHALLENGE
OPENSTACK IS COMPLEX

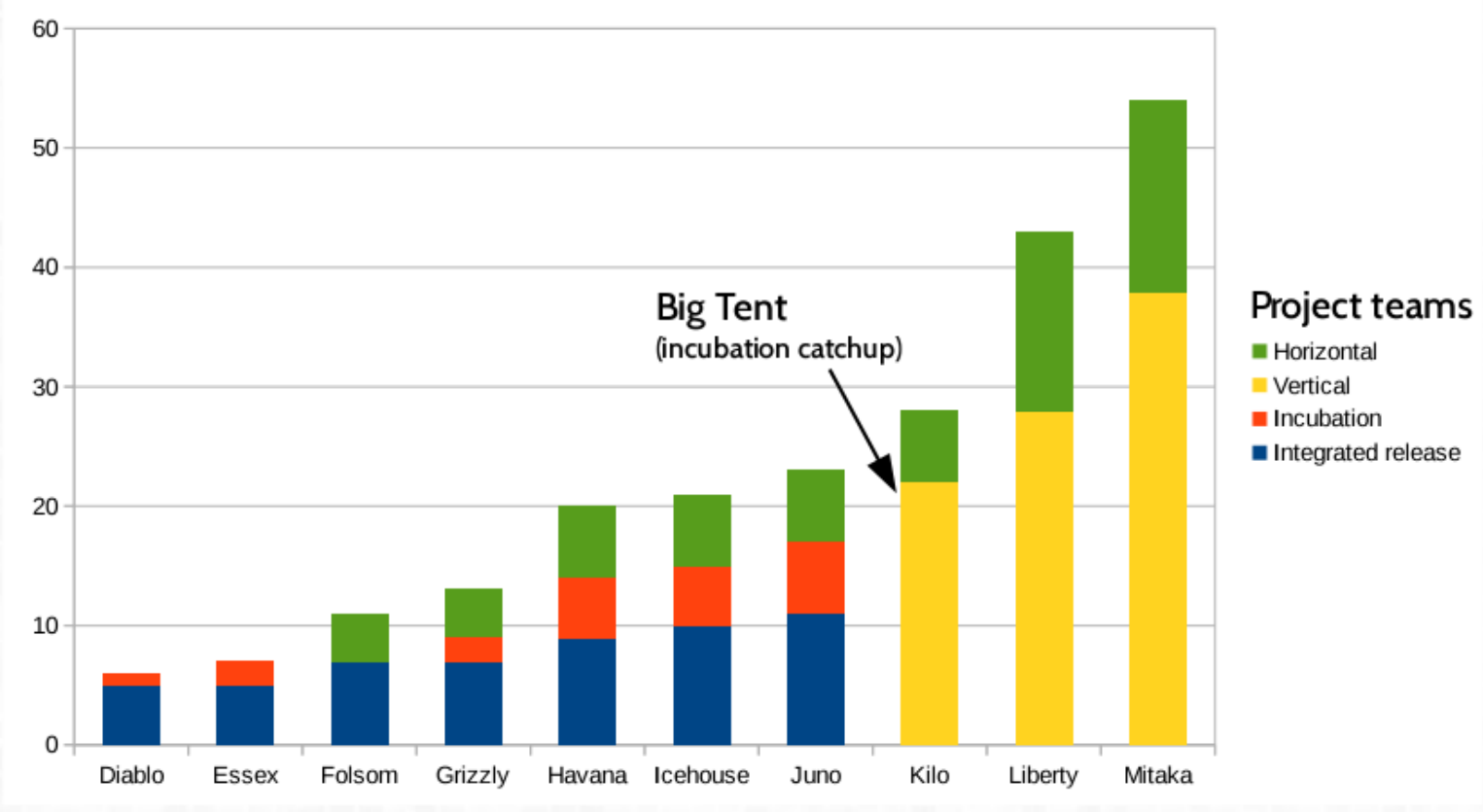
OPENSTACK IS COMPLEX

- That's how OpenStack is, deal with it
- Every day talented people are working hard to make OpenStack more complex
  - New configuration options
  - Support for new hardware
  - Adding new features
  - New You-Name-it-as-a-Service projects
- OpenStack projects are building bricks for your cloud
  - Infinity of ways to compose your cloud
- OpenStack favors configurability
- Nightmare for deployment tools

OPENSTACK BIG TENT

- OpenStack has been growing
- Big Tent is a change in governance to adapt the the ever expanding number of projects
- From 11 ‘integrated’ projects in Juno:
  - nova, swift, cinder, neutron, horizon, keystone, heat, ceilometer, trove, glance, sahara
- To 58 projects supervised by the Technical Committee in Newton
  - http://governance.openstack.org/reference/projects/

ALWAYS MORE PROJECTS



Source: https://www.openstack.org/assets/presentation-media/bigtent-retro.pdf

BUILD A COMMUNITY

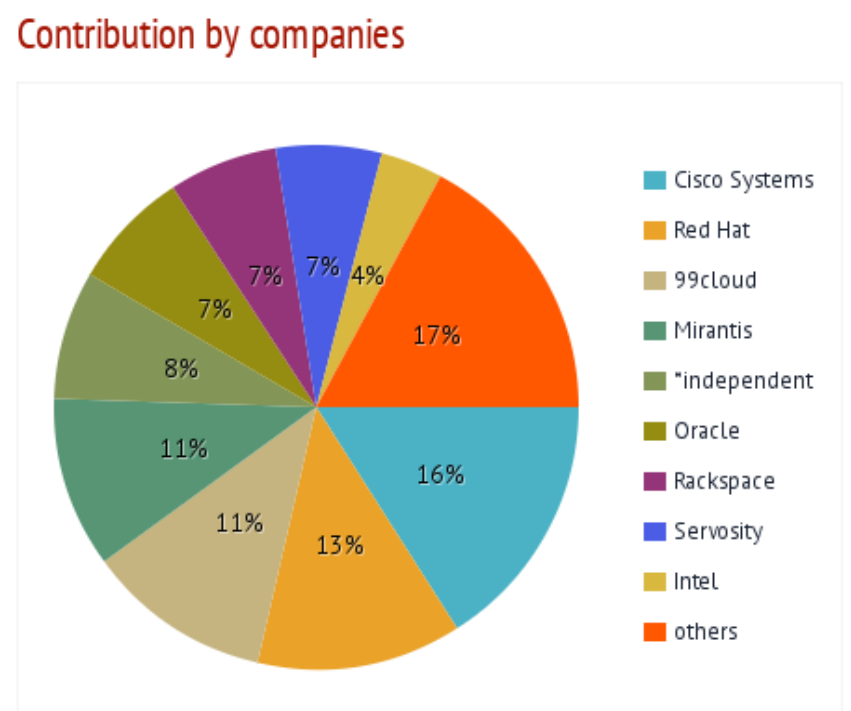
- Taking services separately, they are not that complex to containerize
- It's the profusion of services that makes it difficult to handle
  - Today Kolla contains 173 Dockerfiles
- Importance of universal container image
  - Provide (stable) API to configure containers
    - What files to bind mount, with ownership/permissions
    - What command to run
  - Provide way for image customization (ex vendor plugin), source or binary built, change of base image
  - Ensures no lock-in
- Cost of fork
  - Example of fuel-ccp

IMAGE CUSTOMIZATION

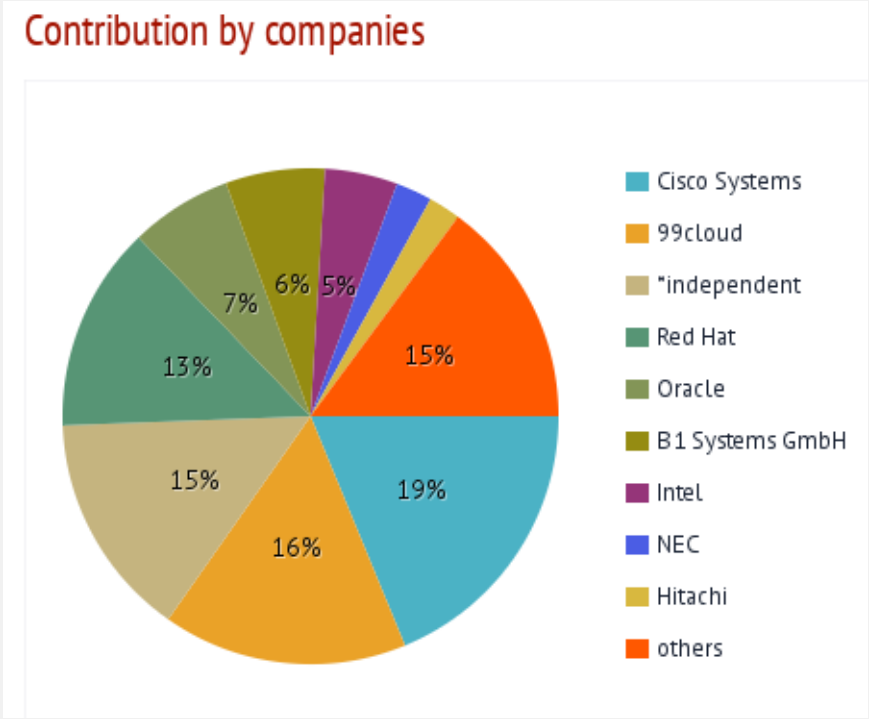
- Containers are immutable by nature
  - Content does not change over time

- Shipping all existing plugins with the container image is unrealistic
- Bootstrap the image to load custom code
  - Relies on network connectivity
  - Breaks the idem-potency property of containers
- Bind mount the customizations
- Provide a mechanism to build custom images
  - Kolla does it by using advanced jinja2 templating (template inheritance)
  - Provide tooling to build images

DIVERSITY MATTERS



Overall Kolla commits



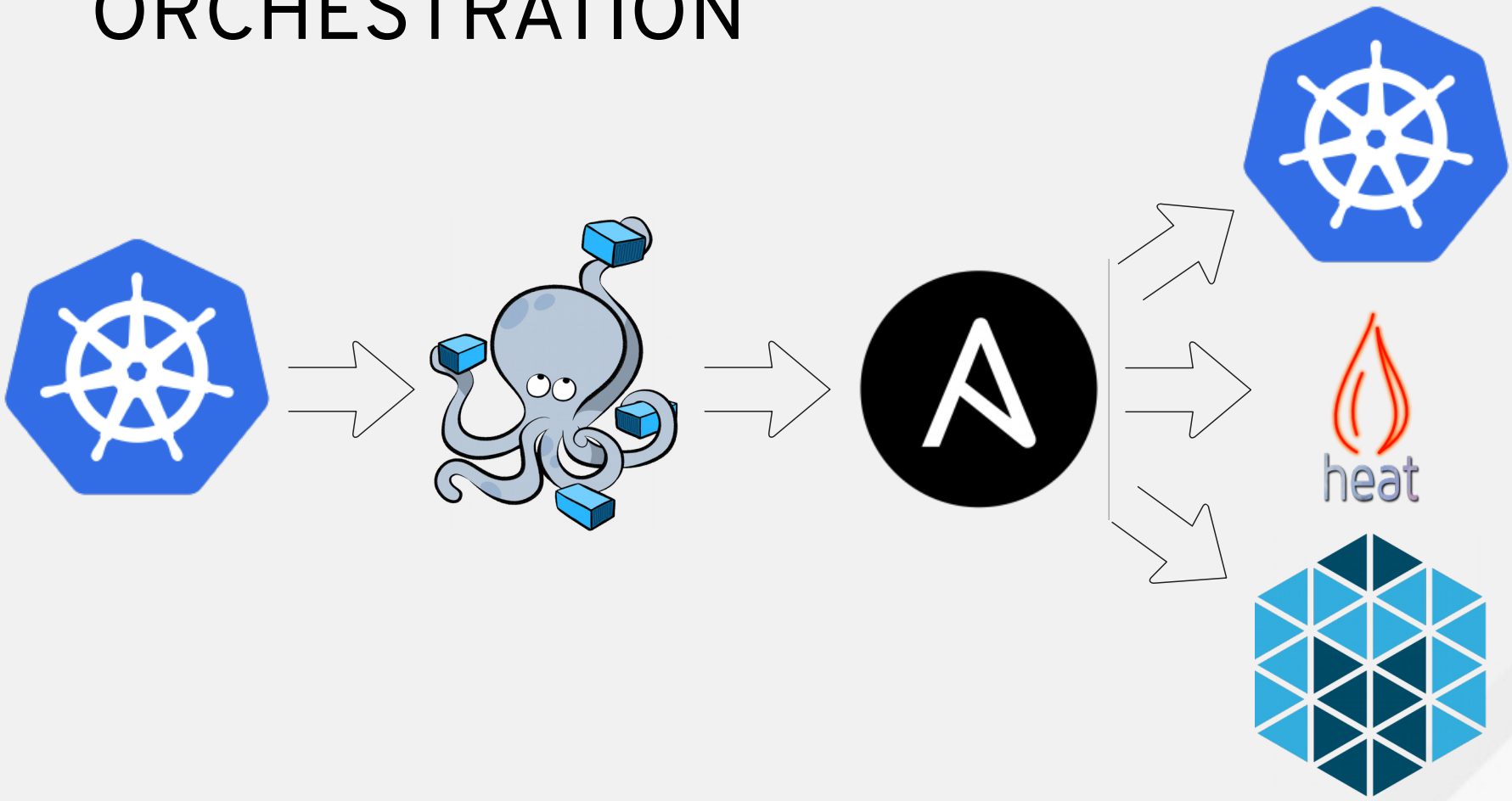
Newton Kolla commits

ORCHESTRATING CONTAINERS

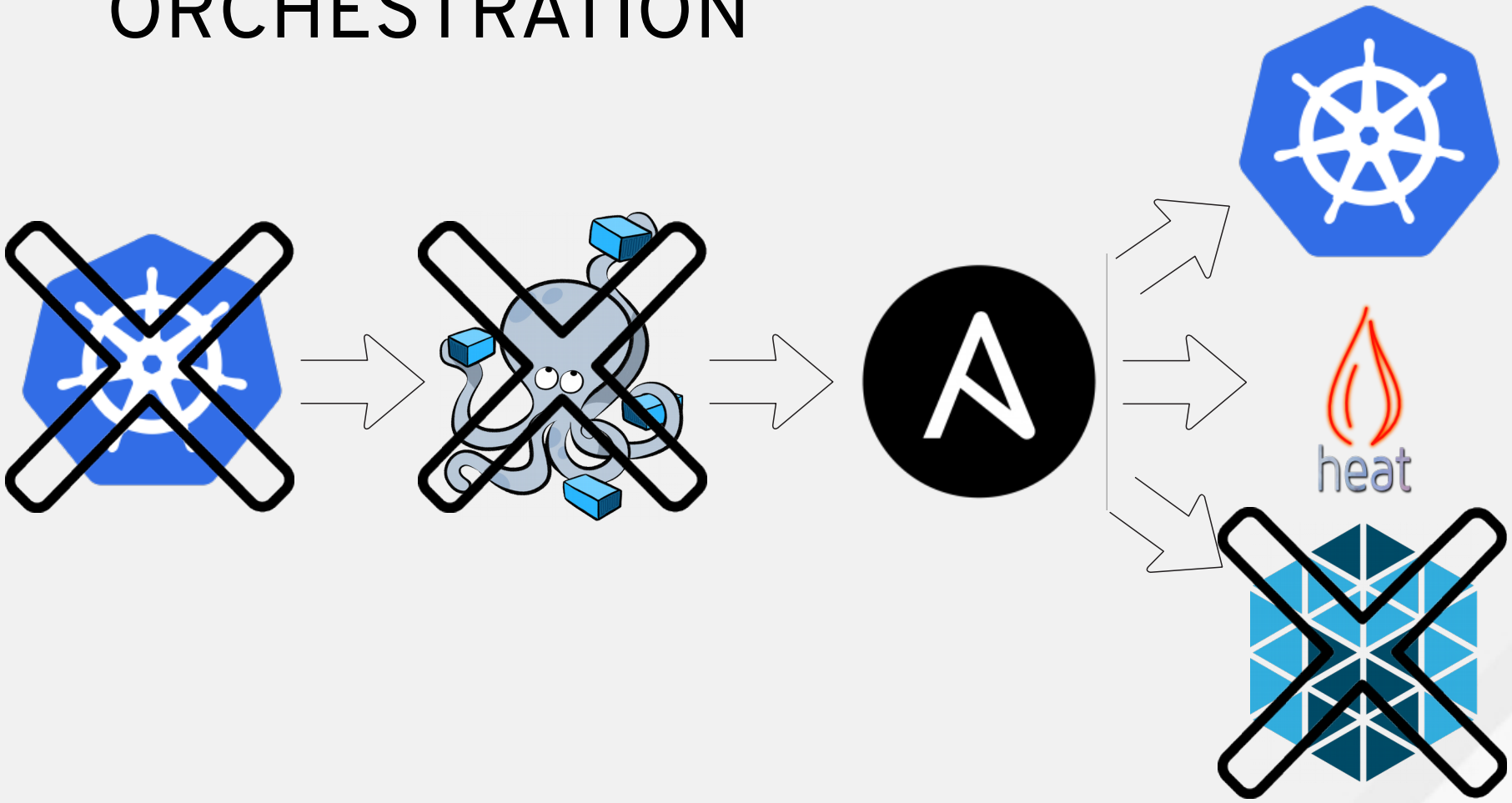
- Orchestration is the automated arrangement, coordination, and management of computer systems, middleware, and services (Wikipedia)
- **THE** real difficulty with this level of complexity
- Without orchestration it's just a pile of useless container images
- Need to deal with
  - Service dependencies
  - Stateful services
  - Bootstrap tasks
  - Updates/Upgrades
- A framework can help with these tasks

2nd CHALLENGE
TECHNOLOGY IS STILL MATURING

A BRIEF HISTORY OF KOLLA ORCHESTRATION



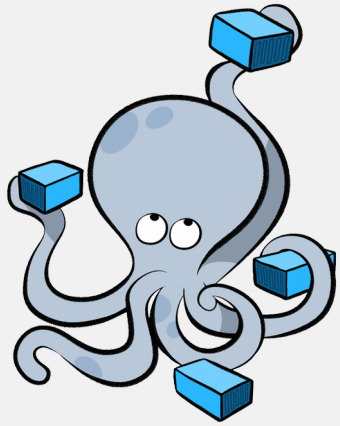
A BRIEF HISTORY OF KOLLA ORCHESTRATION



KUBERNETES 1st TRY

- Sept 2014 – Feb 2015
- Impossibility to update the compute kit without loosing VMs
- Lack of support for Super Privileged Container
  - Use of host network namespace
  - Use of host PID namespace
  - Use of host bind mounting

DOCKER COMPOSE



- Feb 2015 – Aug 2015
- Strategy is the same as with Kubernetes:
  - Start all containers at the same time
  - Set restart policy to “always”
  - Rely on checks inside the container to know if their dependencies are satisfied
  - Wait for containers to converge and reach a stable state
- Container dependency scripts not reliable
  - Prone to race conditions due to lack of single source of truth about state of system
  - Containers restarting (instead of sleep) cause all sort of errors
- Not idempotent

ANSIBLE

- Since Apr 2015
- First started as a wrapper around docker-compose
- Serialize deployment, brings idem-potency
- Eventually grew into its own deployment tool

- Still hit a couple of issues with Ansible
- Ansible versioning with docker-py incompatibility
  - Wrote own docker module instead of depending on Ansible one
- Licensing issues
  - GPLv3 not compatible with Apache licence, and forbidden by OpenStack policy

MESOS

- Nov 2015 – Apr 2016
- Discontinued by Mirantis
- Business decision, not a technical issue

HEAT

- Since Aug 2015
- Heat is the OpenStack orchestration project
- Used in TripleO
- Compute Node

KUBERNETES, 2nd TRY

- Since Feb 2016
- Kubernetes has evolved a lot in the meantime
- Added support for host networking, host pid
- Has InitContainers for bootstrap tasks
- Has StatefulSet for stateful services
- Helm as a dependency manager
- Kolla-kubernetes is still a Work in Progress
  - The compute kit is already functional

IS DOCKER PRODUCTION READY?

- Need for a recent kernel
- Choice of a graph (storage) driver
  - It's a mess
    https://docs.docker.com/engine/userguide/storagedriver/selectadriver/
  - Uses the distribution's default storage driver which can heavily hit performance
  - The Kolla community recommends the btrfs or aufs graph drivers for storing data as sometimes the LVM graph driver loses track of its reference counting and results in an unremovable container.
- Many rants about Docker not production ready on the Internet

DOCKER PROGRESSES

- Docker 1.7 allowed bindmounting of the /dev filesystem
  - Mandatory for the cinder container
- Docker 1.9 introduced named volumes
  - Prevents possible data loss with data containers
  - Happening in case the data container is rebuilt
- Docker 1.10, allowed mount propagation to the containers
  - Permits thin containers for neutron agents
  - Provided /run/netns is bind mounted in the container
  - Docker restart no longer kills neutron routers
  - Operators can access namespaces from the host
- Docker 1.10 also brings decent performances for registry v2

THE GREAT L RELEASE FIASCO

- Ansible 1.9.4 made a change that broke compatibility with docker 1.8.3
- Stable/liberty branch was stuck on Ansible 1.9 since Ansible 2 was not backward compatible
- Stable/liberty branch could not use named volumes that came with Docker 1.10
- This lead to "the great Liberty release fiasco"
  - Replaced the content of Liberty branch with Mitaka and have it deploy Liberty containers
  - Goes against all the stable branch management policies
- http://docs.openstack.org/developer/kolla/liberty/liberty-deployment-warning.html

OTHER RANDOM ISSUES

- Because of system technical limitations, upgrade of a libvirt container when using software emulation (virt_driver=qemu in nova.conf), does not work at all. This is acceptable because KVM is the recommended virtualization driver to use with Nova.
- Issue with host/guest combinations:
  - Docker storage driver issue (usually an AUFS issue with older kernels)
  - Impossibility to load host kernel modules inside container due to format incompatibility (modules compressed with xz compression format)
  - Usually best to use same OS for host and guest

CONCLUSION

EXPLORING POSSIBILITIES

- Containers represent a shift in technology
- A lot of what we do is experimental
  - Never know when you'll hit a blocker
  - Trial and error until success
- Need to learn from our experience
  - define best practices

WE CAN DO IT

- With the help of a community, nothing is impossible
- Technology is maturing quickly
  - Docker is stabilizing
  - Kubernetes is evolving very rapidly and closing feature gap
- Exploring a new world
  - Can be tough…
  - … but it's also exciting
  - And rewarding

IT'S WORTH IT

- Soon we will see production grade containerized deployments
  - Easier updates/upgrades
  - Simplified day 2 operations
- Containerizing OpenStack is only the beginning
- Containers can be an integral part of developers work flow
  - Keep developers' machines clean. Ever worked with devstack?
  - Containers are deterministic. Remember immutability?
  - Speed up feedback loop
- Containers can be used in CI
  - You test the exact same bits that will run in production

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos