# Deploying OpenStack at Scale

...with TripleO, Ansible and Containers

OpenStack Days UK - 26th September 2017

Steve Hardy (shardy@redhat.com)

# Who am I

- Software Engineer at Red Hat
- Working on OpenStack since 2012
- Heat & TripleO Core reviewer
- Former Heat & TripleO PTL

HEAT
an OpenStack Community Project

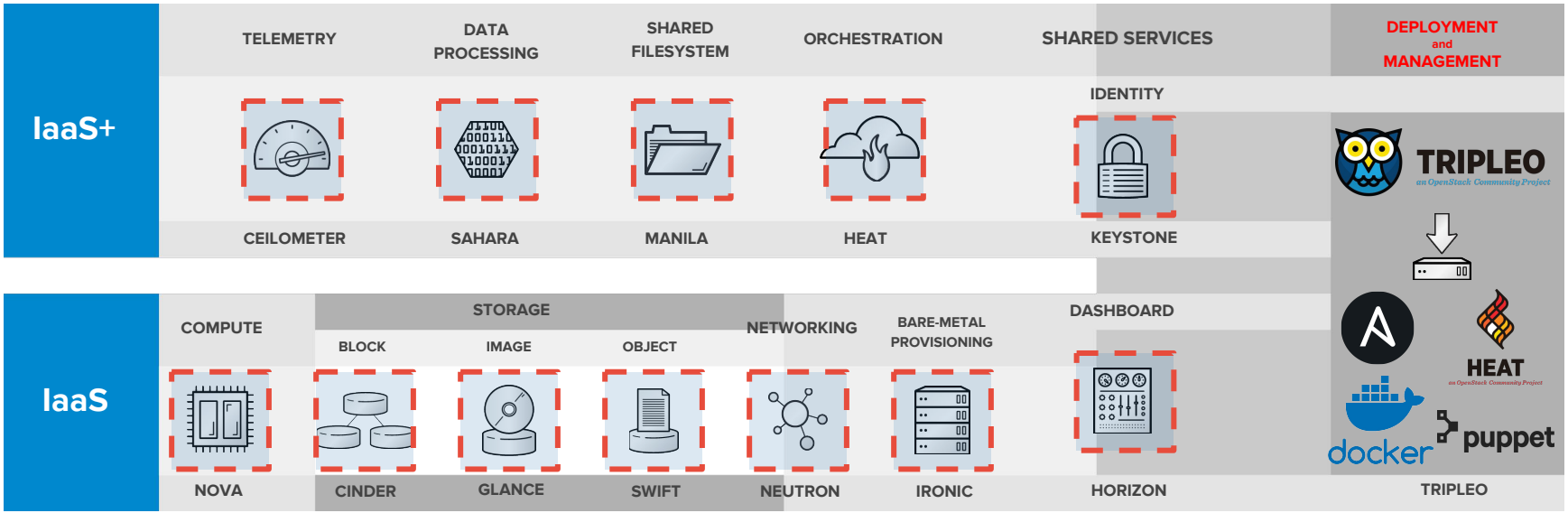
TRIPLEO
an OpenStack Community Project

openstack®

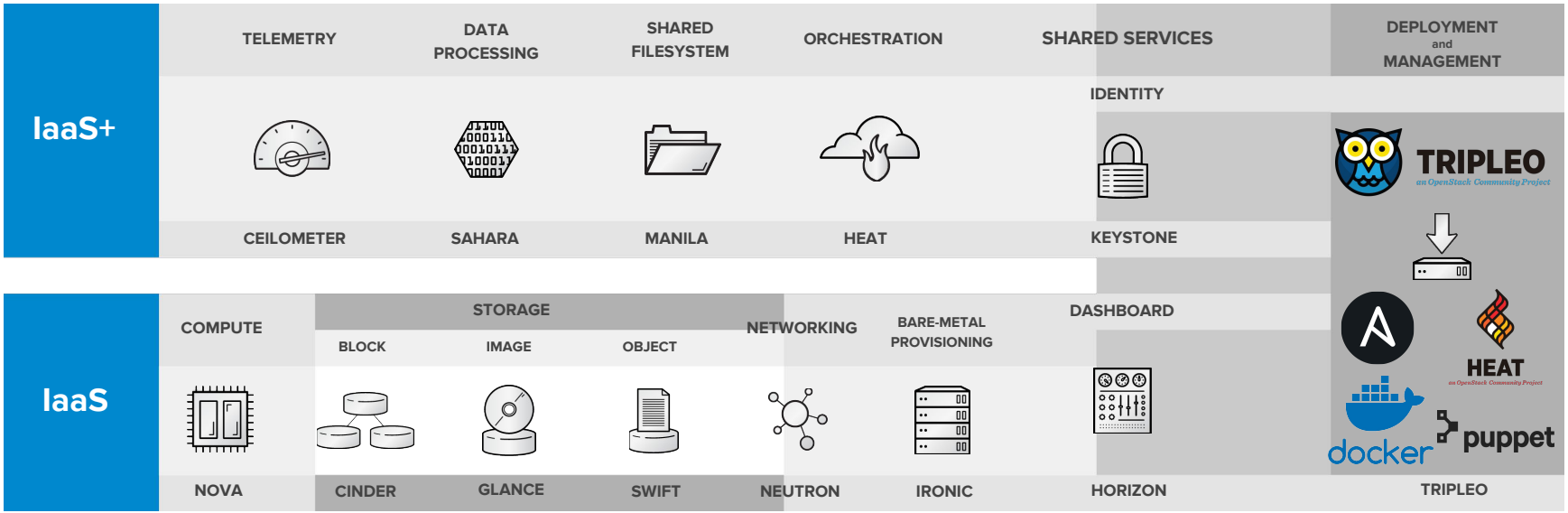
redhat.

# What is this talk about?


# What this talk IS about.
# Deploying/managing the OpenStack services




# What this talk IS NOT about.
# Containerizing workloads **RUNNING ON** OpenStack.

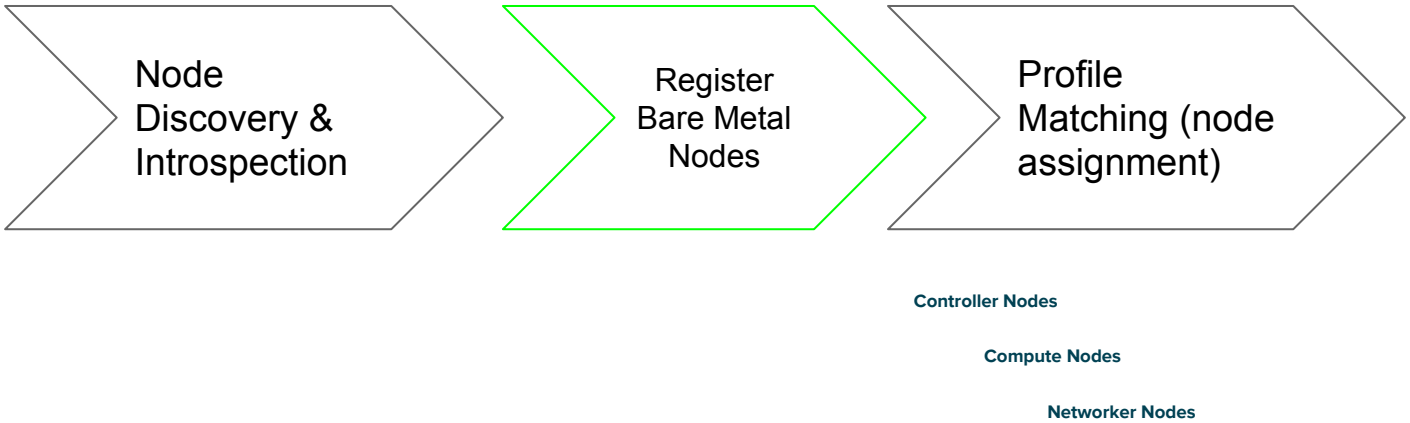


# Introduction to TripleO

| PLANNING | DEPLOYMENT | OPERATIONS |
| --- | --- | --- |
| Network topology | Pre-flight checks/validations | Major version upgrades |
| Service parameters | Deployment orchestration | Applying minor updates |
| Resource capacity | Service configuration | Scaling up and down |

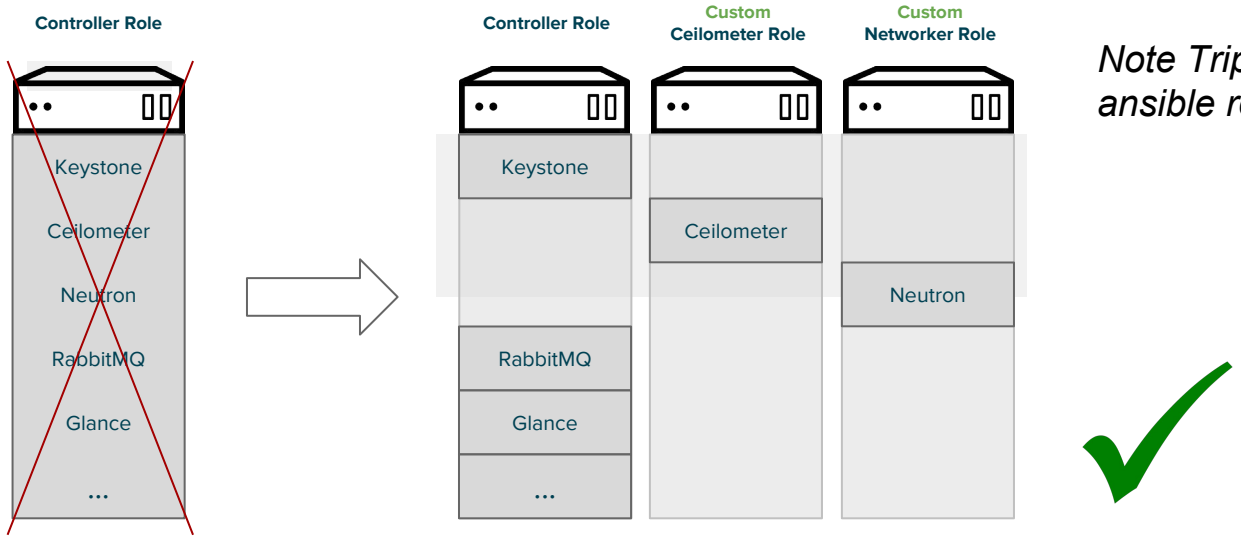
# Planning

## Manage hardware inventory, capacity planning



# Planning

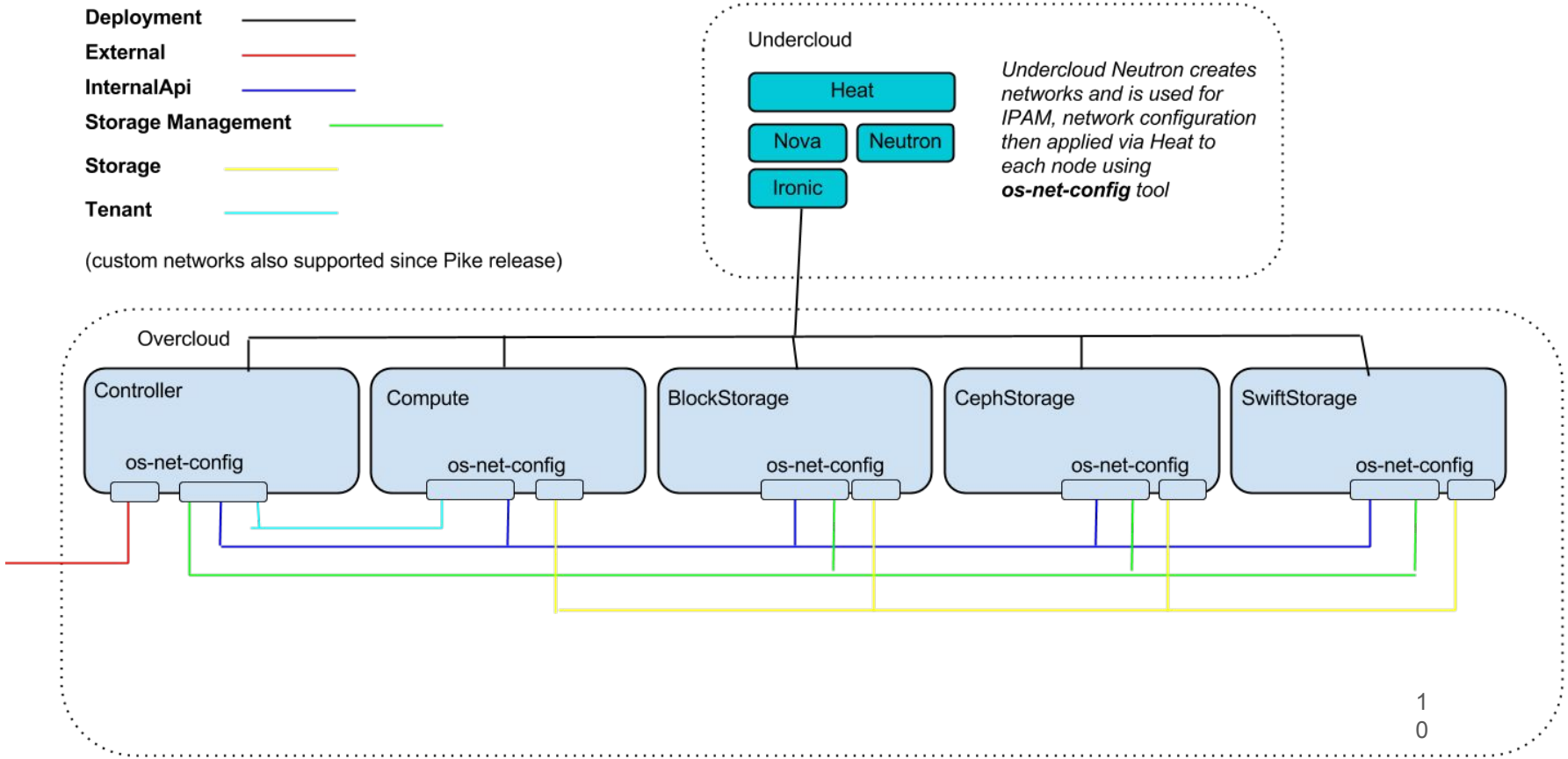
## Composable roles, customize enabled services and placement




*Note TripleO role != ansible role...*

# Planning

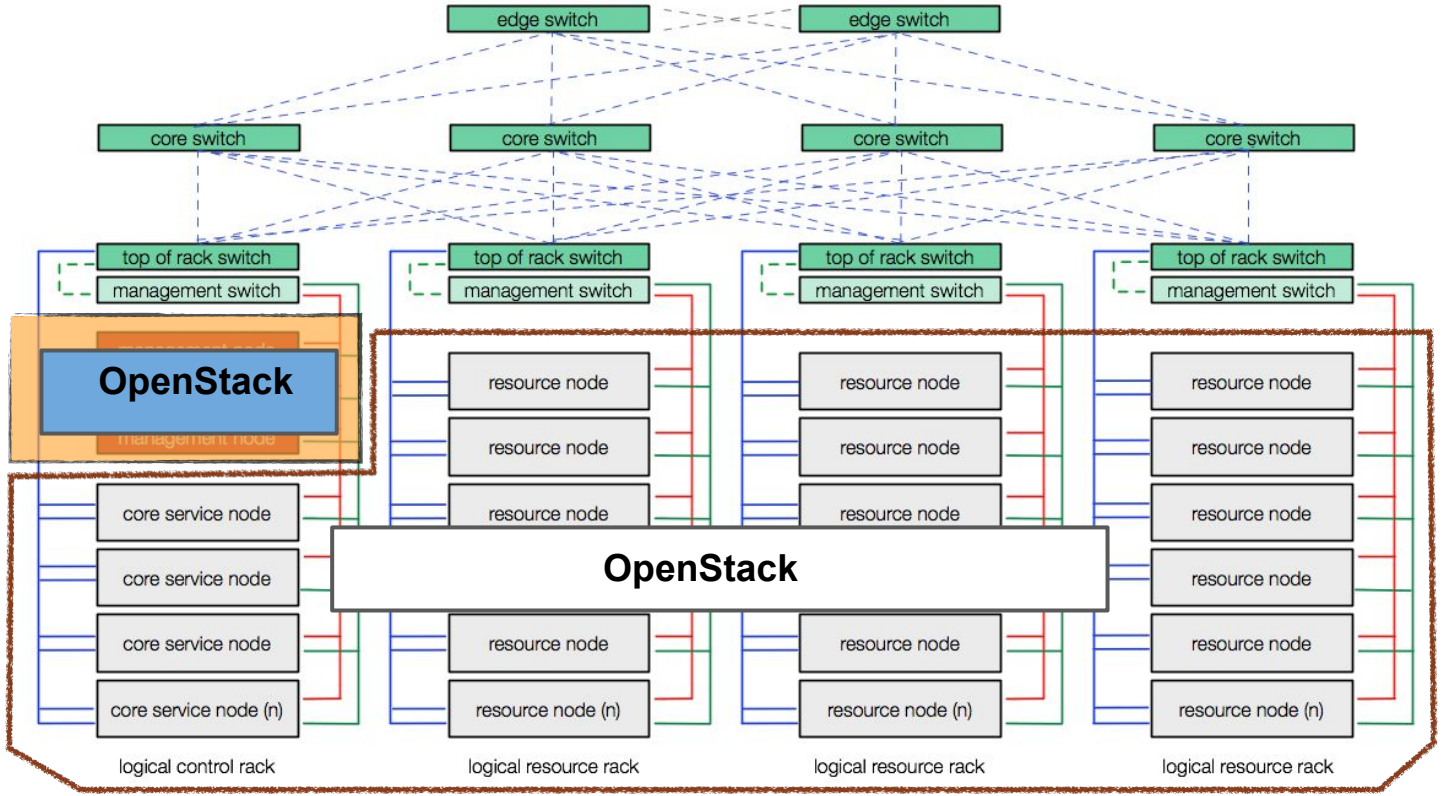
## Customizable network isolation, declarative nic configuration



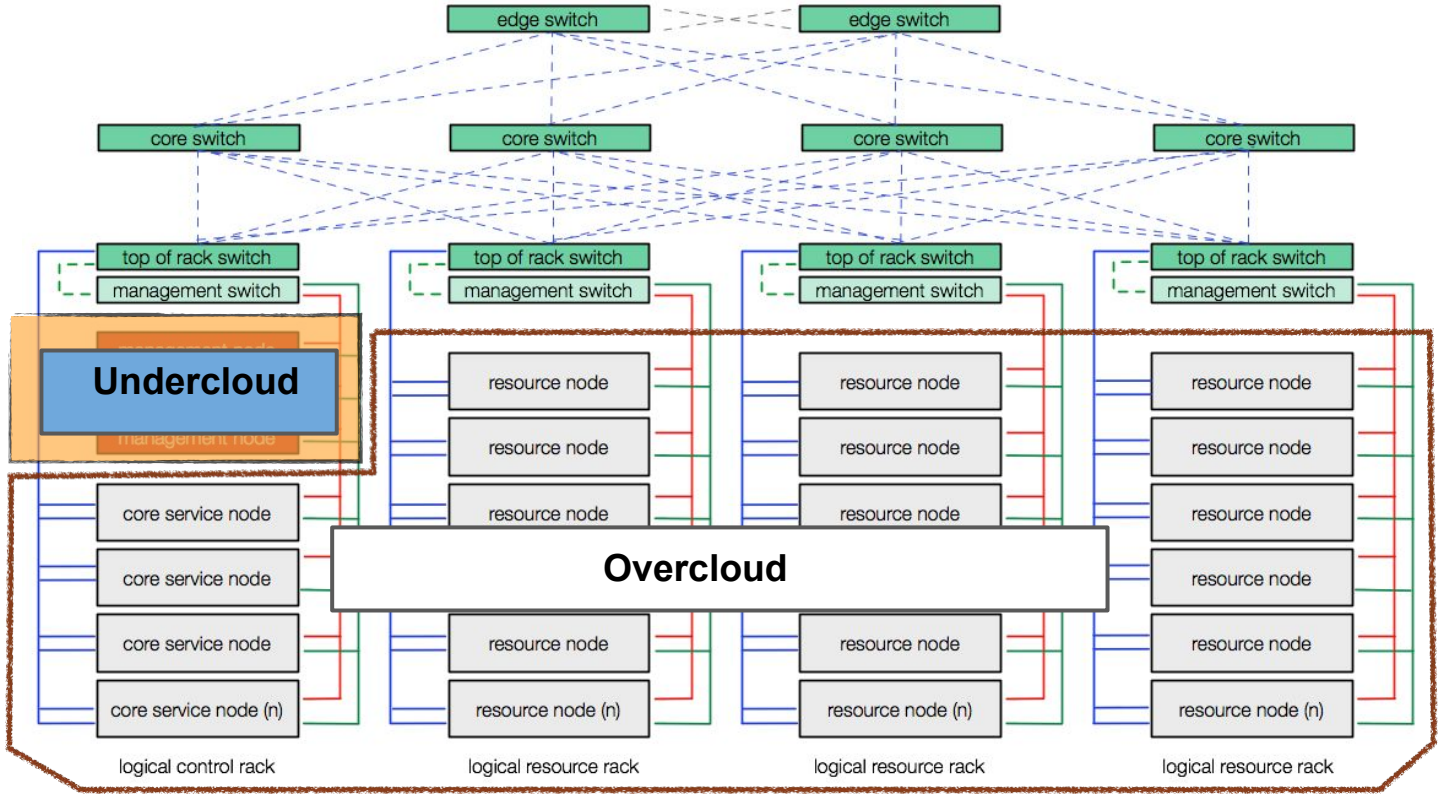
# TripleO - Planning

- Node discovery, introspection & grouping (Ironic)
- Customizable “roles” (group of nodes)
- Pluggable “composable services” (Heat templates)
- Per-service network isolation (Neutron/os-net-config)
- HA support (Pacemaker)
- Predictable placement & preassigned IPs (Ironic/Nova)
- Flexible service configuration (Heat parameters)
- Version control your infrastructure “as code” (yaml)

# TripleO - Deployment

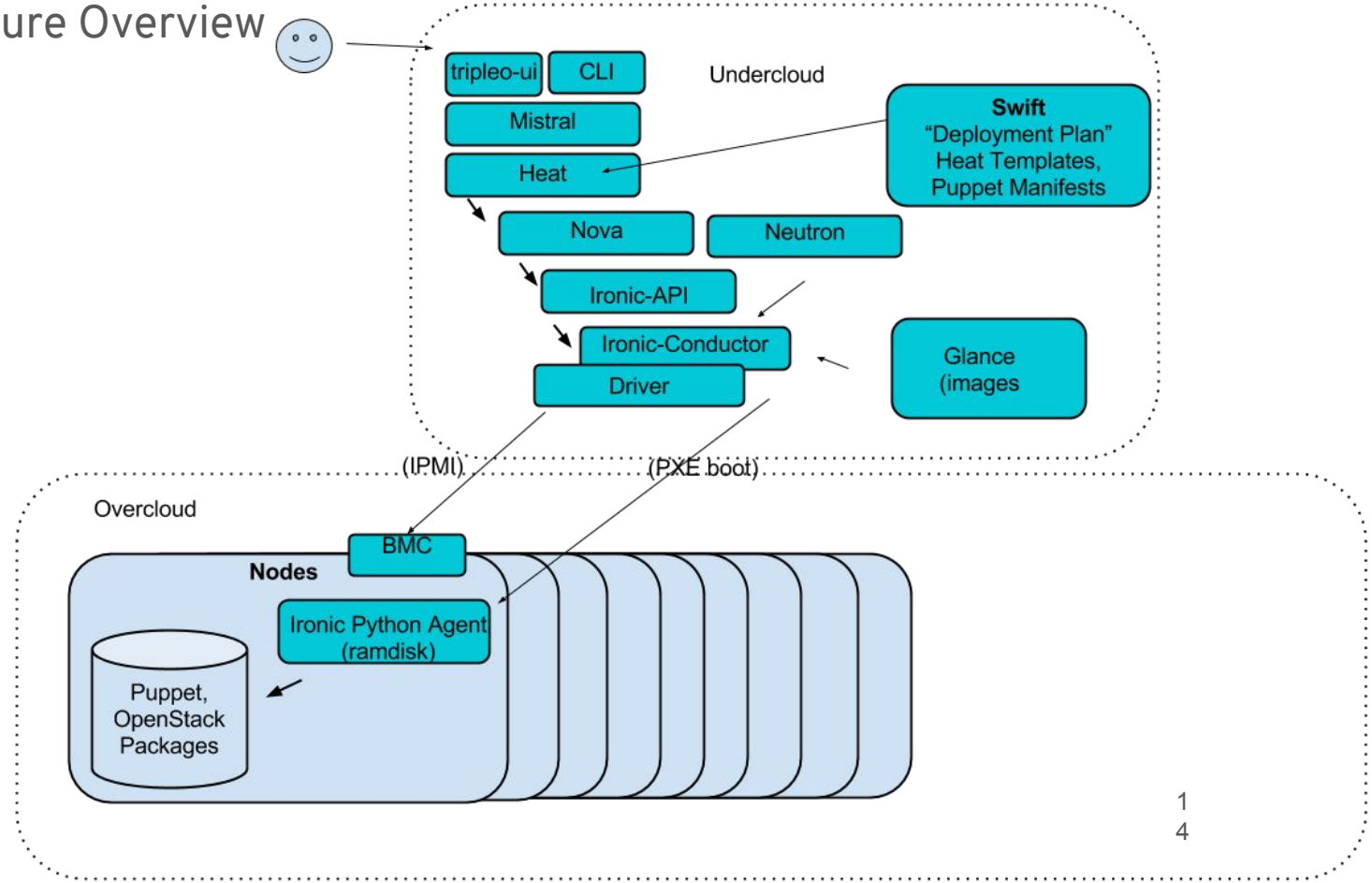


# TripleO - Deployment



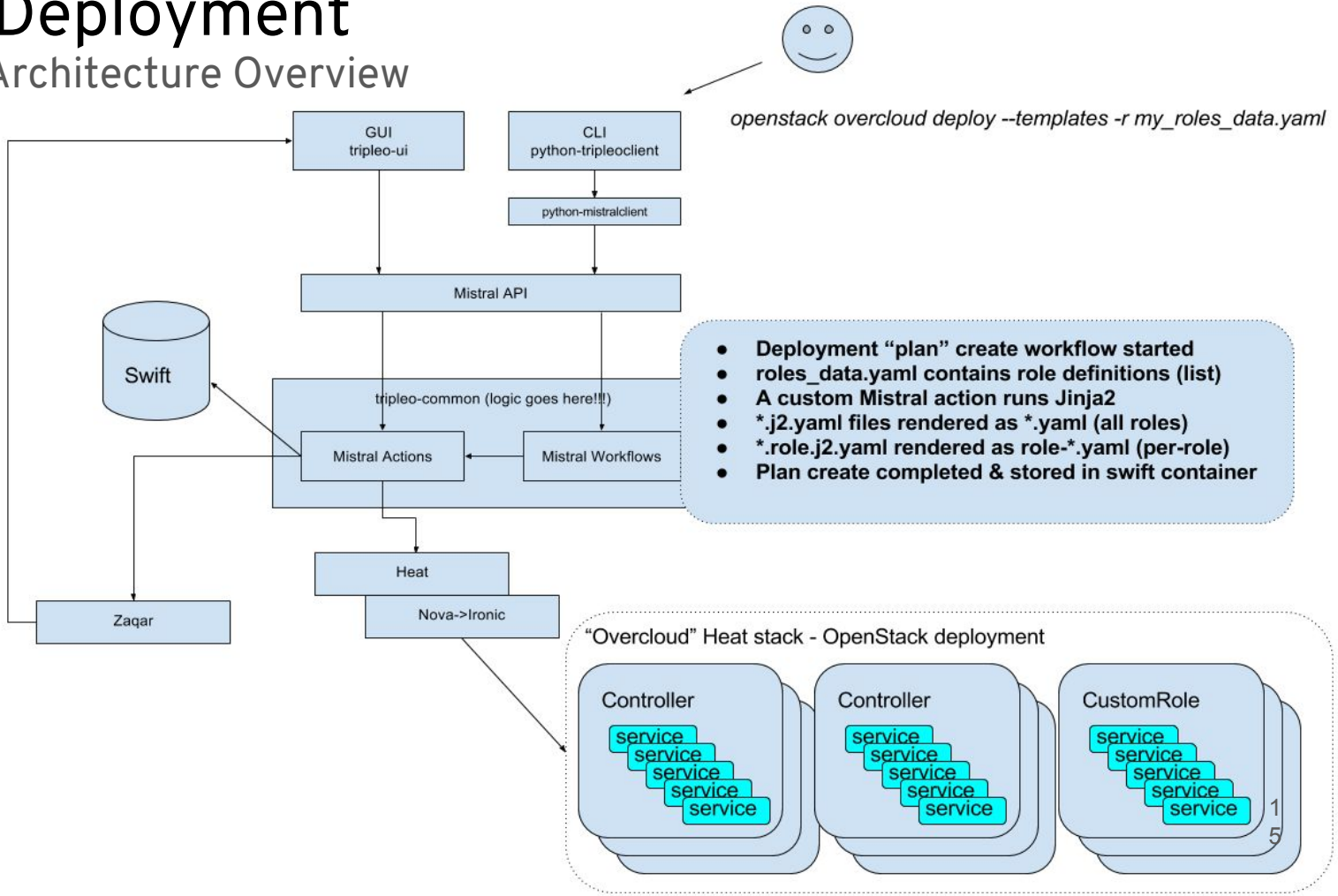
# Deployment

## Architecture Overview



# Deployment

## Architecture Overview



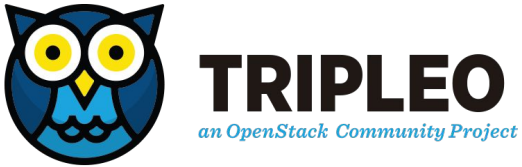
# TripleO - Deployment

- Pre-flight validations (Ansible based)
- Disk image based deployment (Glance)
- Baremetal provisioning (Ironic)
- Baremetal service configuration (Puppet)
- Heat templates define each service & configuration
- Mistral workflow API provides pluggable interface for UX
- Increasingly using Ansible “under the hood”

# TripleO - Operations

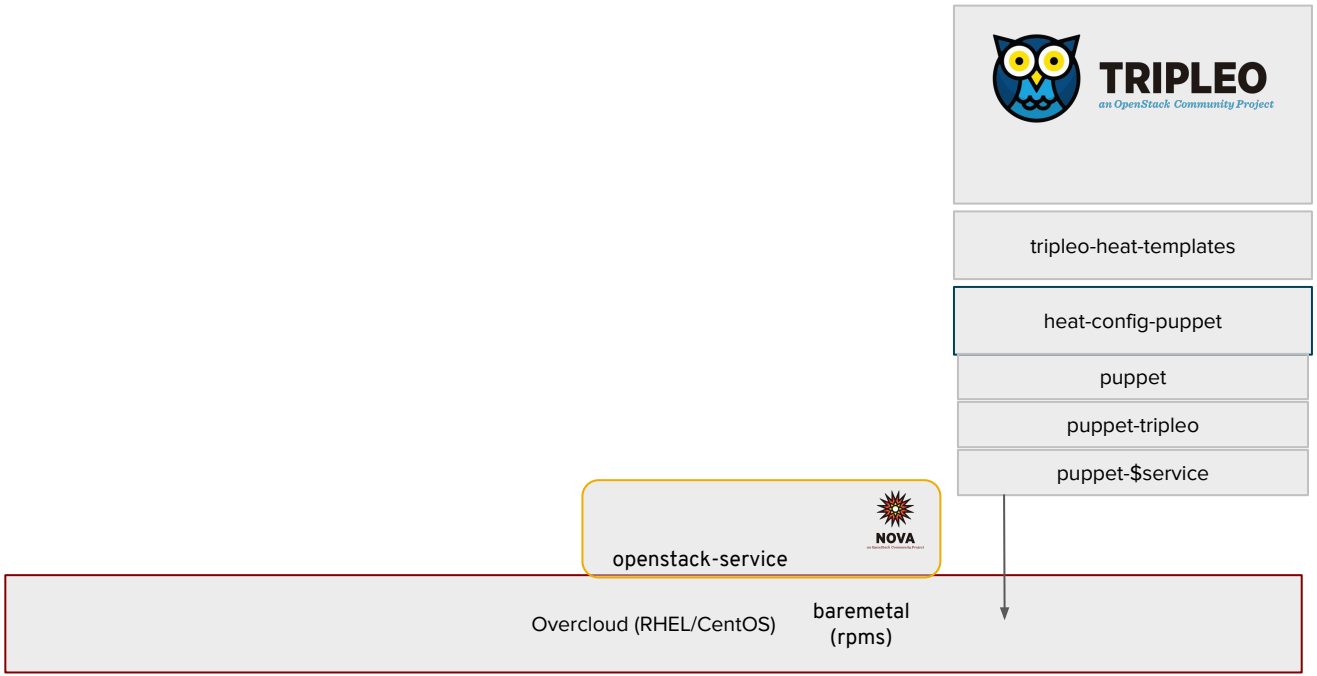
- Automated configuration of monitoring clients (Sensu)
- Support for centralized logging
- Major version upgrades, orchestrated by Heat+Ansible
- Minor version updates - rolling no downtime
- Automated scale up to add capacity
- Orchestrated scale down and node removal

# TripleO + Ansible



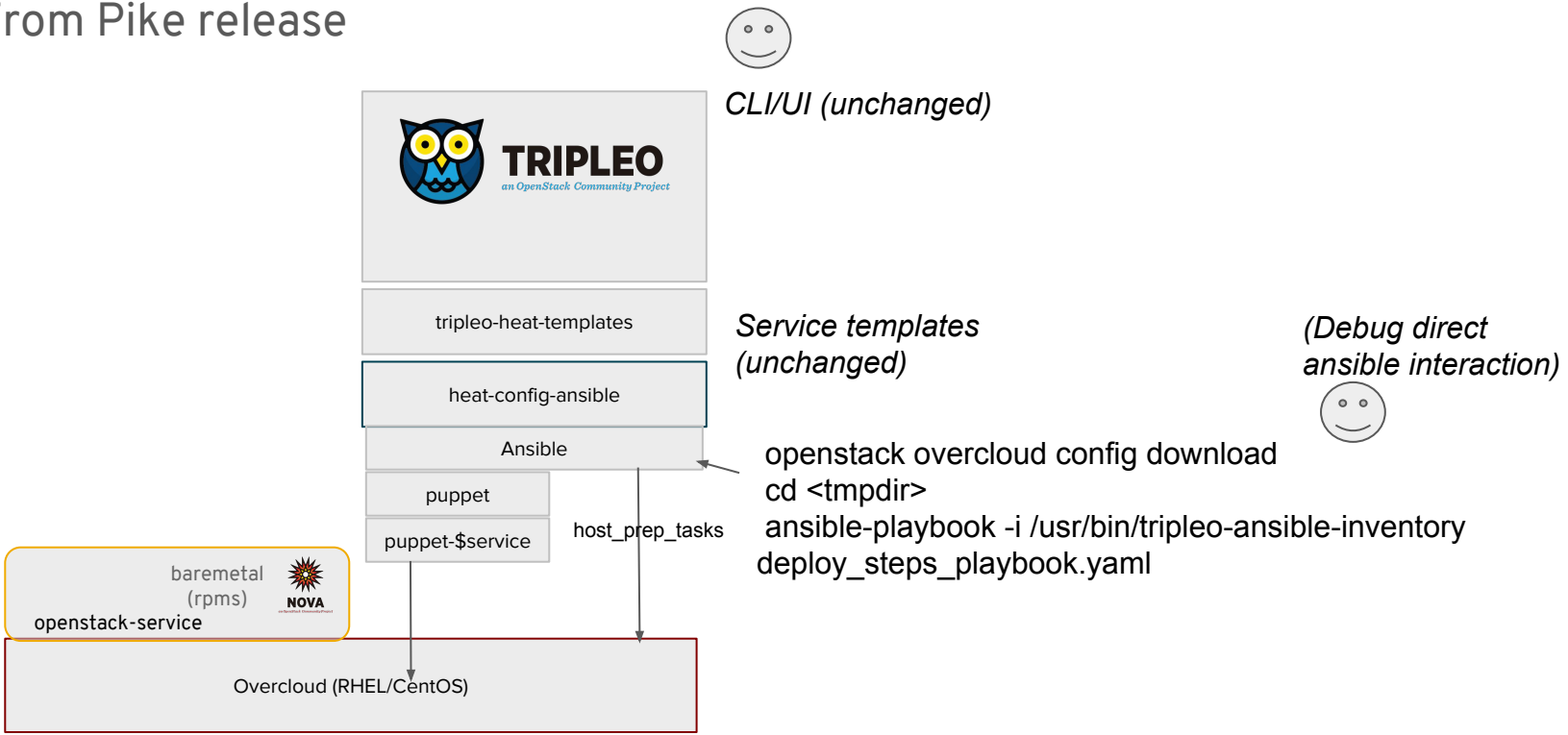

# Heat+Puppet Architecture

## How we used to do it...



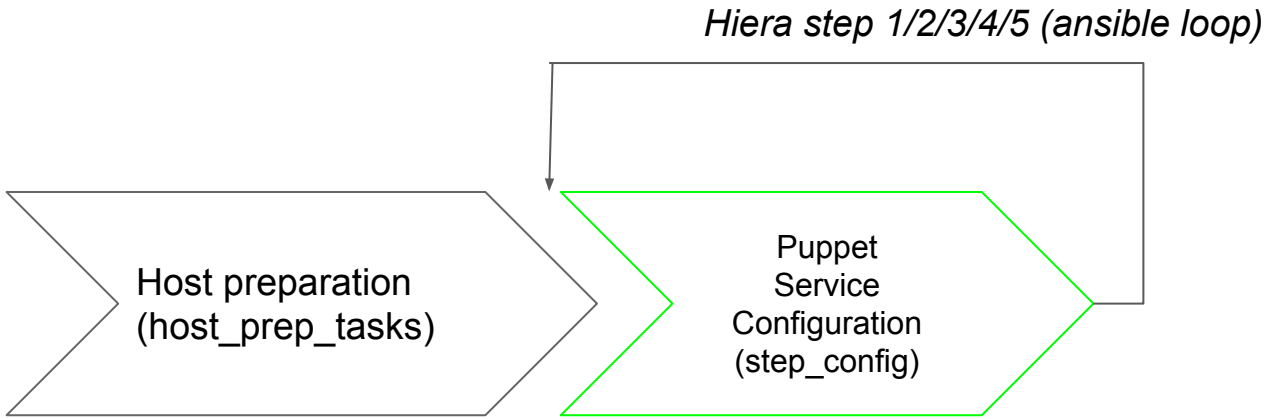
# Heat+Ansible Architecture

## From Pike release



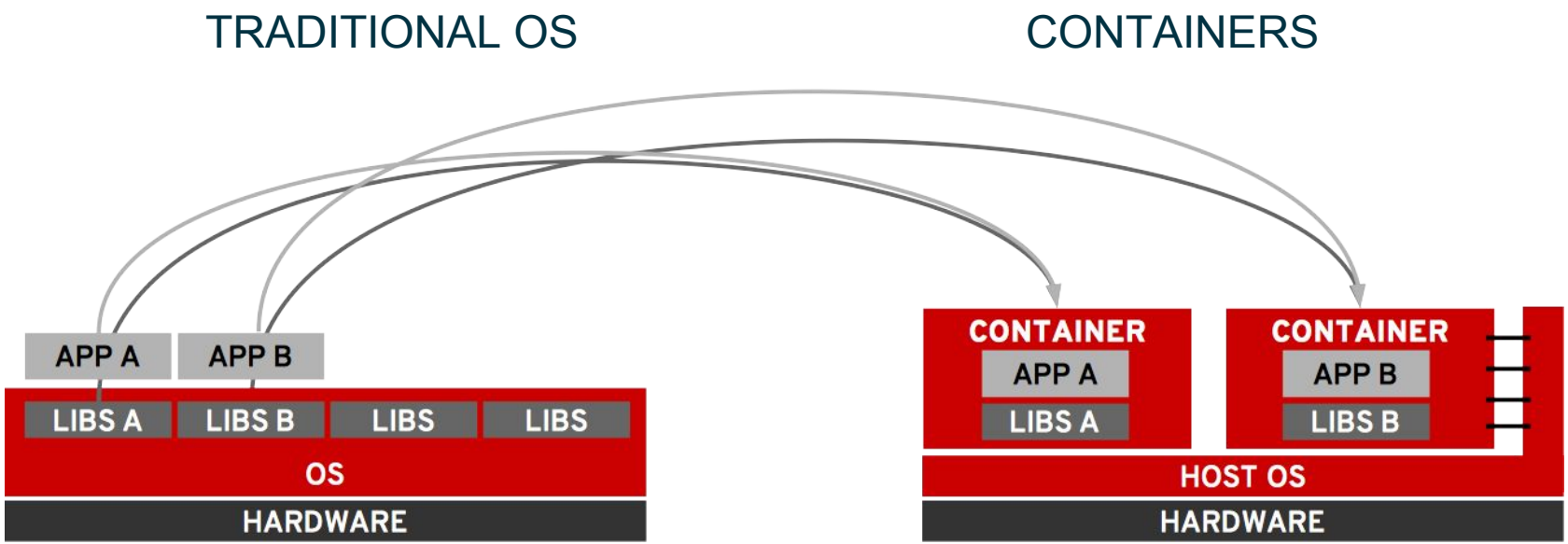
# Deployment

## Ansible + puppet deploy workflow

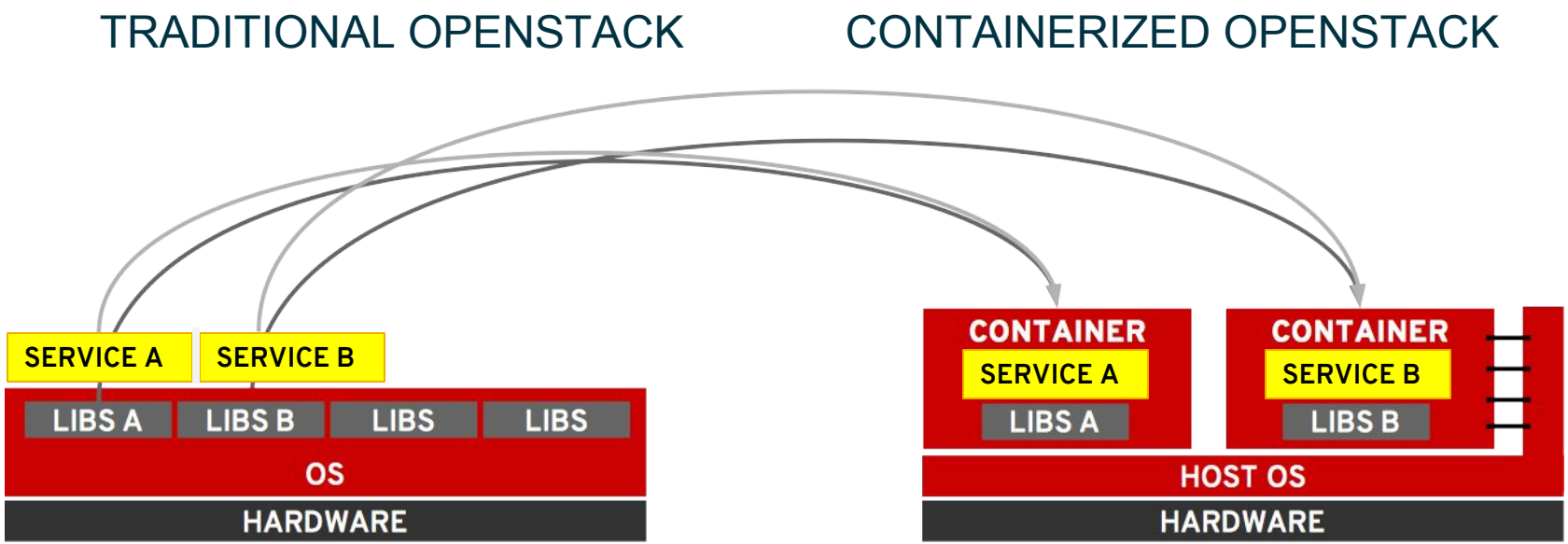


# Deployment with Containers

# Containers for apps ...



# Containers for OpenStack services

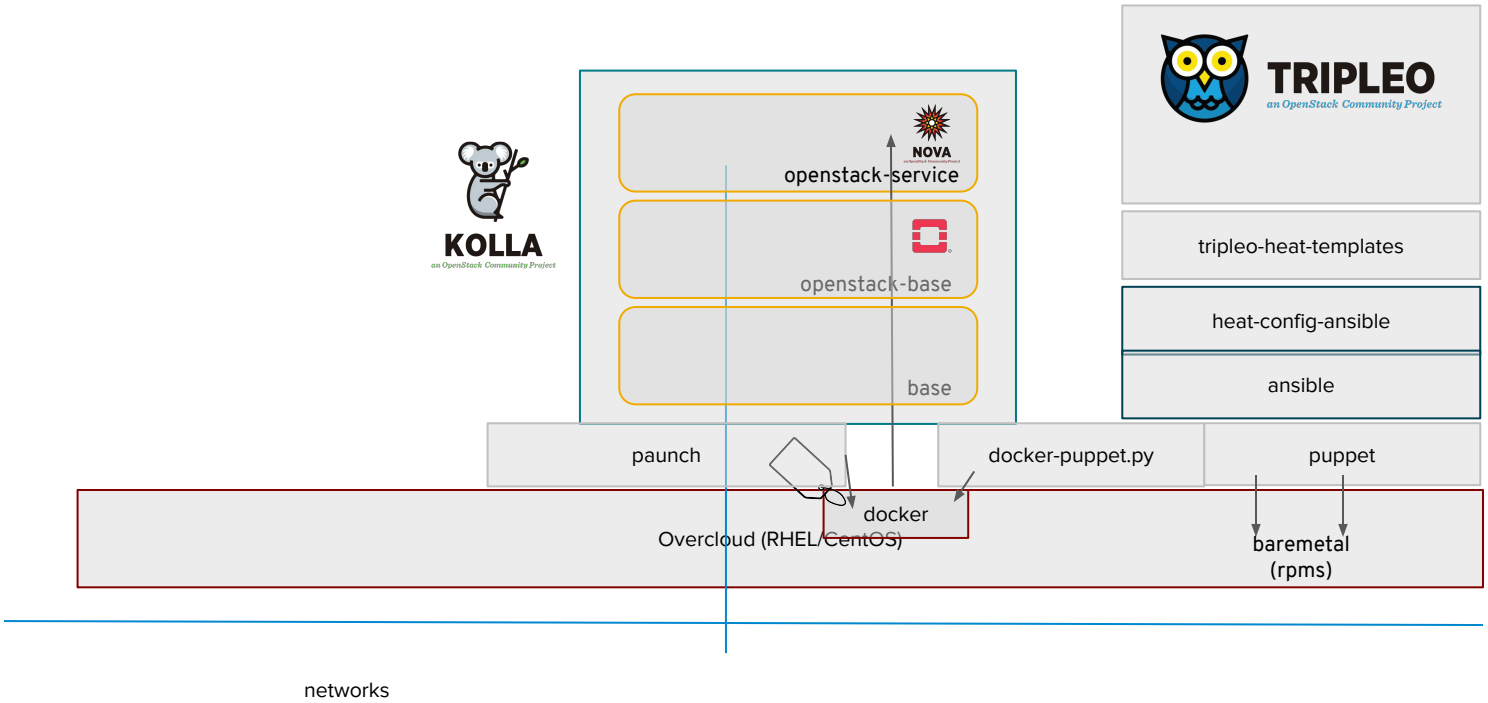


# TripleO - Containers

- Pike release supports most services in containers
- A hybrid baremetal/container model is possible
  - Assists with migration of vendor plugins e.g Neutron
- Includes upgrade support (from baremetal)
  - Most state e.g DB is left on the host filesystem
- Reuses the same “composable services” architecture
  - Some new interfaces specific to containers added
- HA support (Pacemaker managed container bundles)
- Host networking used to retain network isolation
- Backwards compatibility (Heat parameters & puppet)

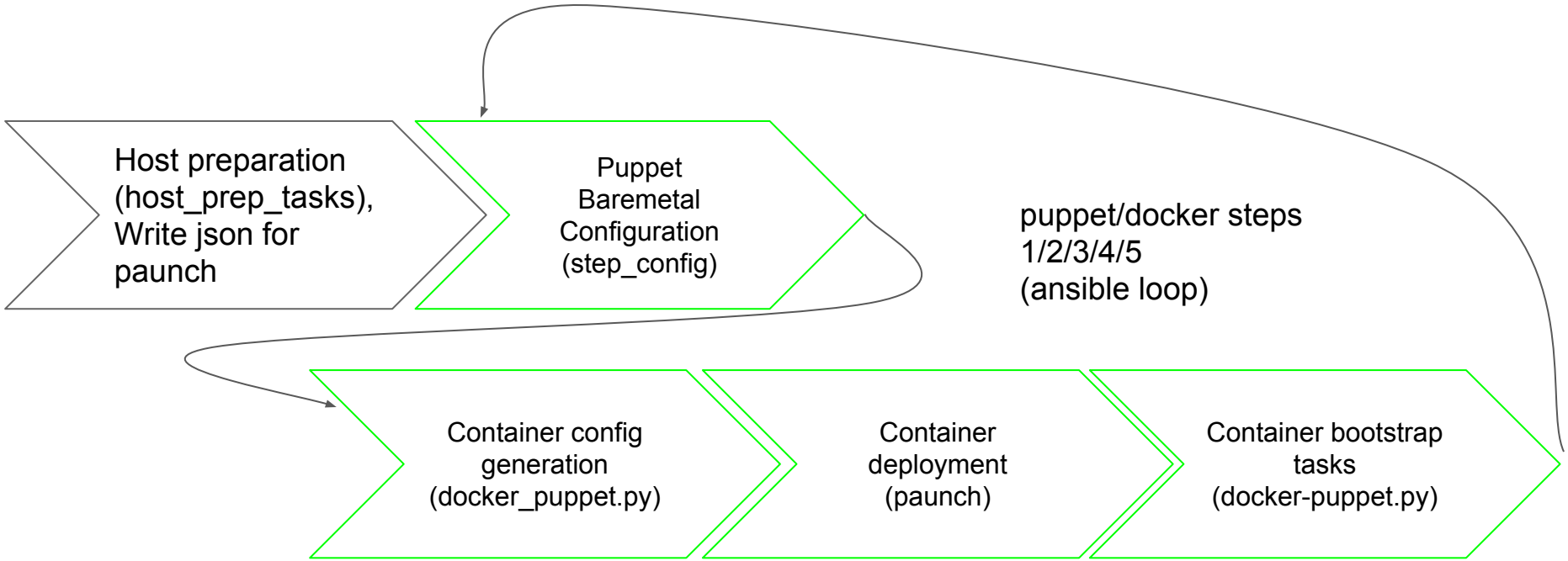
# How does it all come together?

## Sort of like this ...



# Deployment

## Ansible + puppet + containers deploy workflow



# How does it all come together?

## Multi step deployments

- Baremetal configuration (still via puppet)
- Tool to generate config via puppet (docker-puppet.py).
- Ansible playbook runs paunch via command module
- Paunch takes that config and starts up the containers
- Docker-puppet.py runs again to perform bootstrapping via puppet
- ansible deploy tasks applied with an incrementing step

```
  - name: Start containers for step {{step}}
      command: >-
        paunch --debug apply
        --file
/var/lib/tripleo-config/hashed-docker-container-startup-config-step_{{step}}.json
        --config-id tripleo_step{{step}} --managed-by tripleo-{{role_name}}
```

# Paunch 101

## Container orchestration

**Paunch** is a …

Utility to launch and manage containers using the configuration data found in the TripleO service template. (It's an orchestrator of sorts)

It's a **wrapper** for the docker-cli.

Supports standalone use outside of heat/TripleO (derived from the heat docker-cmd hook initially).

# Paunch 101

## http://git.openstack.org/cgit/openstack/paunch/

Single host only, operations are performed via the docker client on the currently configured docker service.

Zero external state, only labels on running containers are used when determining which containers an operation will perform on.

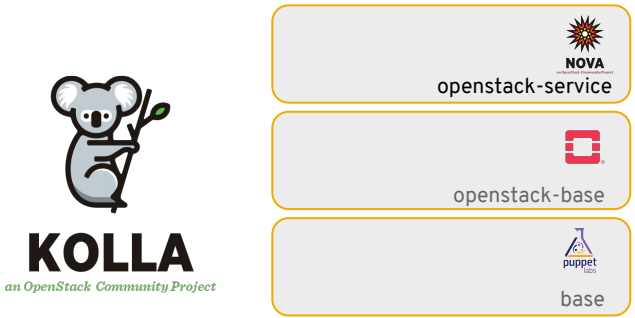
Single threaded and blocking, containers which are not configured to detach will halt further configuration until they exit (config generation & bootstrap containers)

Co-exists with other container configuration tools - only containers tagged as being managed by paunch are modified

Idempotent - leave containers running when their config has not changed, but replace containers which have modified config (minimise minor update downtime)

# How does it all come together?

## Container Images




TripleO uses Kolla containers recipes to build OCI-compliant container images - some additional packages are added to the default kolla build to support TripleO.

# How does it all come together?

## Container image management using openstack CLI plugins

Create a yaml file to set the image, location, namespace, etc:

```
$ openstack overcloud container image prepare
```

Consume the image file generated by prepare:

```
$ openstack overcloud container image upload
```

These steps generate an environment file that is then consumed by heat on deployment/update.

```
$ openstack overcloud deploy --templates $templates -e $templates/environments/docker.yaml
```

This command is used to deploy a (minimal) container enabled overcloud - only one additional environment file is needed to enable containerized deployment.


msc
XINES
tex.
Matson
LEASING COMPANY
GOLD
TAL
BEACON
FLORENS

# What is happening now?

## Pike Release

- Pike release imminent
- Ansible refactor & “config download” available
- Most services will be containerized for Pike
- Hybrid model support to enable Neutron/Cinder/Manila vendor support
- Upgrade from previous baremetal to containers
- Ansible based minor update workflow
- Deploy ceph via ceph-ansible
- Composable Networking (user defined networks)

# What is happening tomorrow?

## Queens and Beyond

- Queens has stabilization theme
- Further refine the Heat/Ansible “split stack” model & flexibility
- Investigations into k8s integration (http://blog.flaper87.com/)
- Further collaboration with other deployment teams (deployment WG)
- Additional composable networks features (generated nic templates)
- Spine/Leaf deployment
- Enable rolling major upgrade for services which support it
- Potentially reduce the number of required tools (“tripleo lite?”)

# Links & where to find out more

- IRC (Freenode) #tripleo
- openstack-dev mailing list [tripleo]
- https://docs.openstack.org/developer/tripleo-docs/
- https://docs.openstack.org/tripleo-quickstart/
- http://tripleo.org/planet.html (has links to blogs)
- https://hardysteven.blogspot.co.uk/
- https://etherpad.openstack.org/p/tripleo-deep-dive-topics
    - Has links to previous talks, we can do more if you request specific topics!