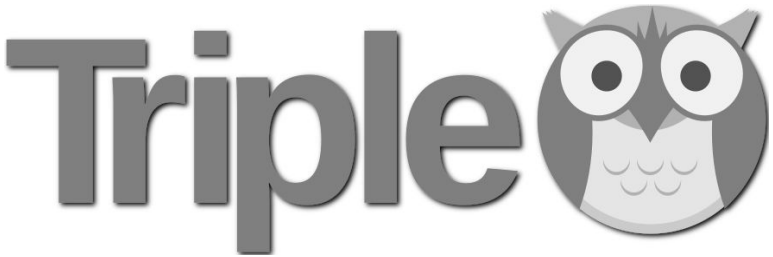
TripleO Project Onboarding

OpenStack Summit Boston

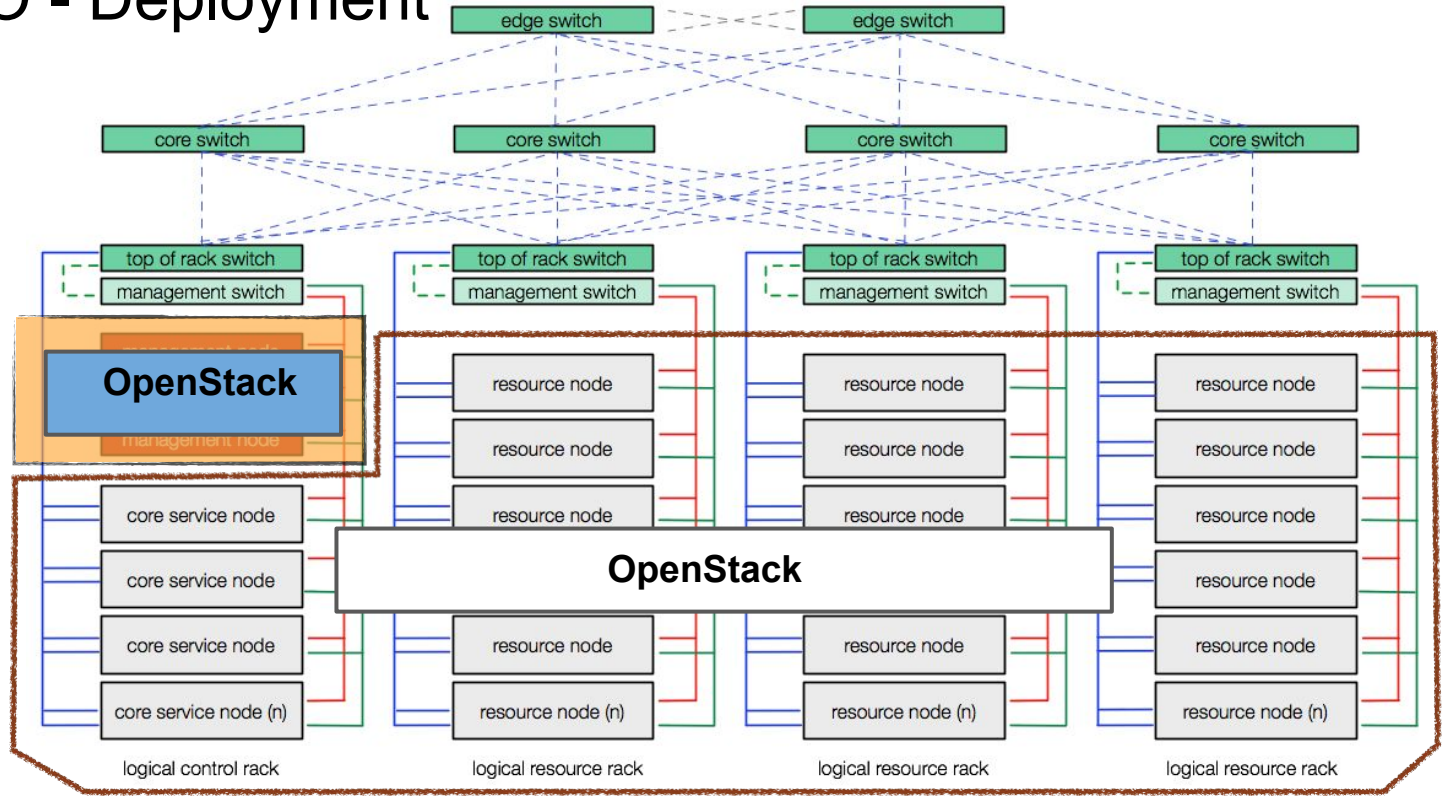
9th May 2017

Steve Hardy (shardy@redhat.com)
Dan Prince (dprince@redhat.com)
James Slagle (jslagle@redhat.com)

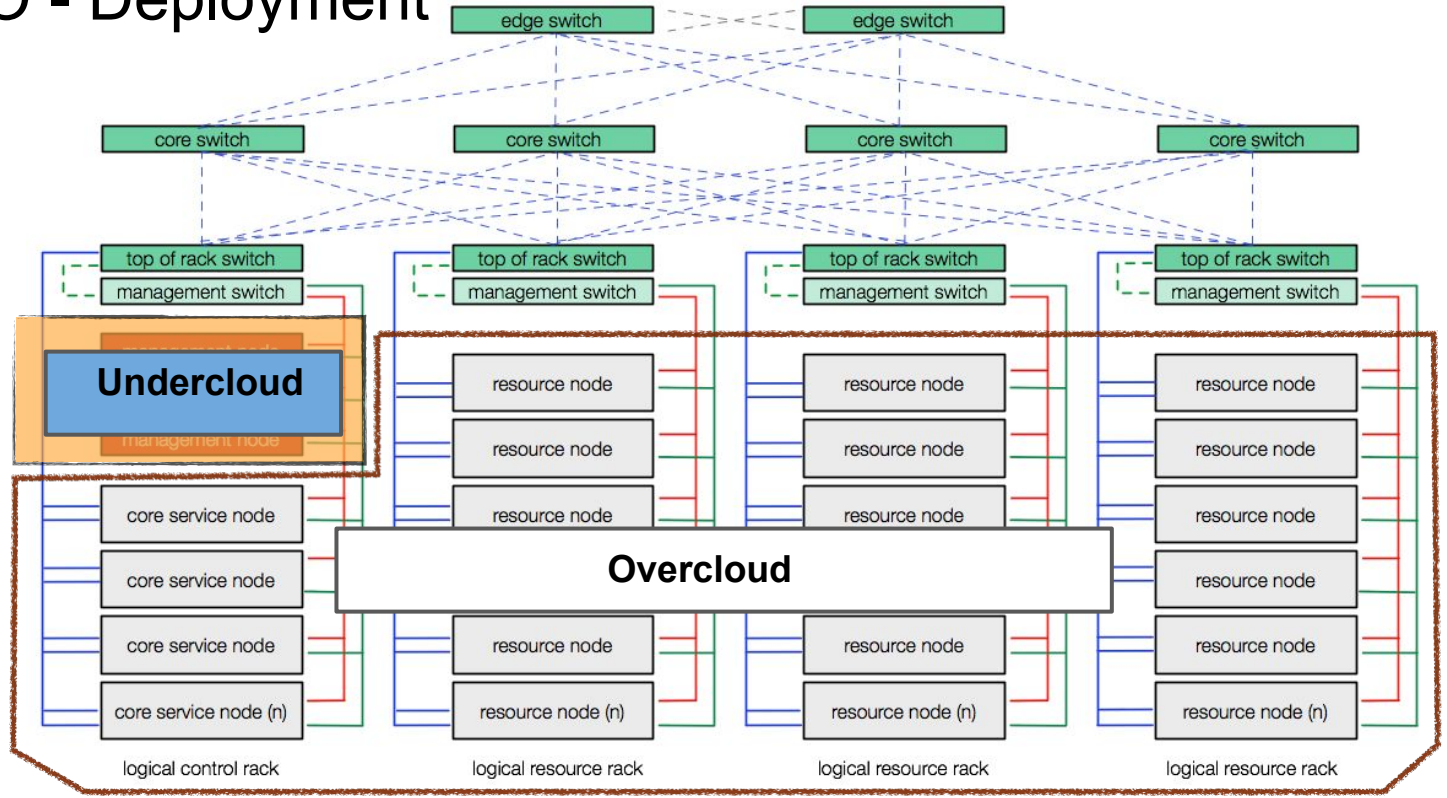


Introduction, Overview

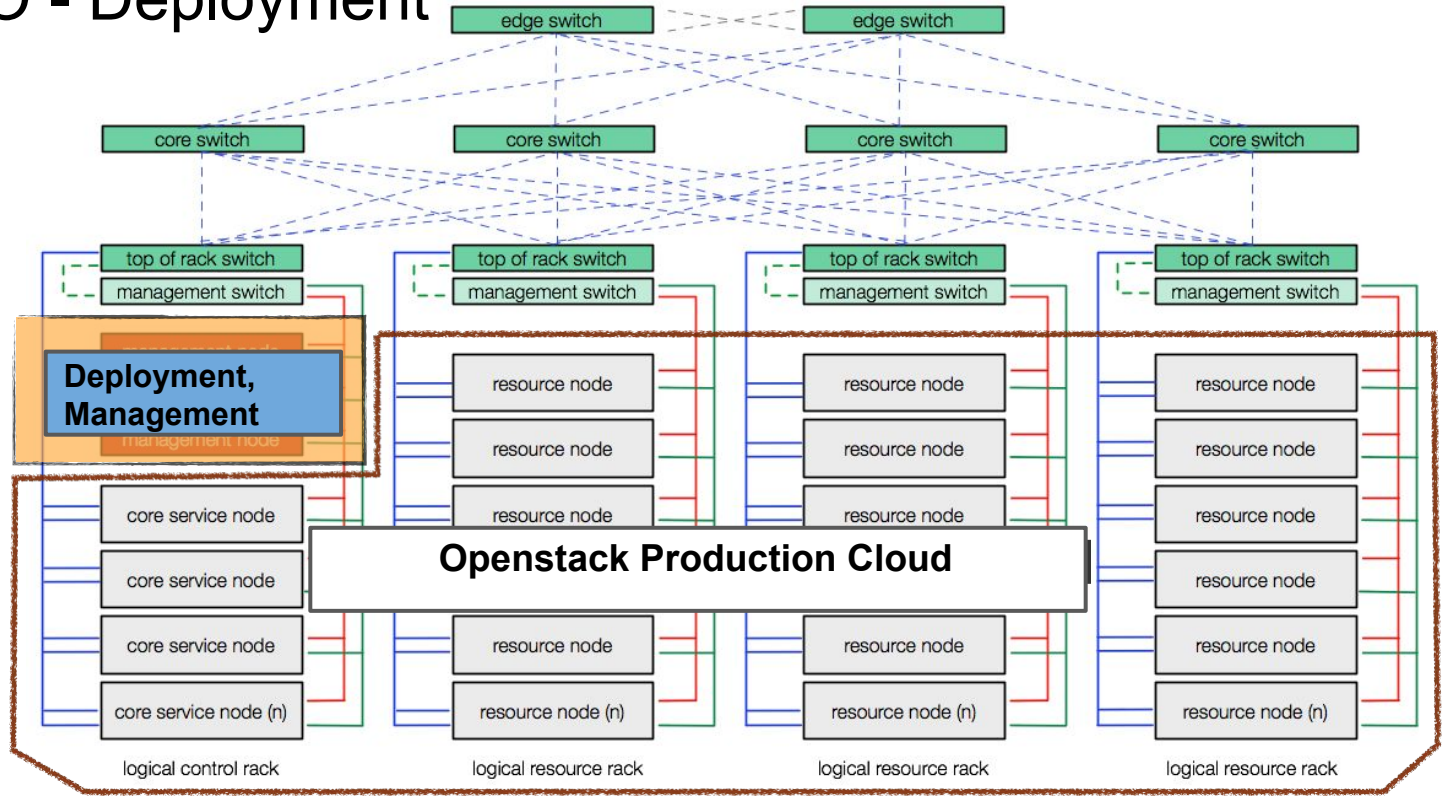
TripleO - Deployment



TripleO - Deployment



TripleO - Deployment



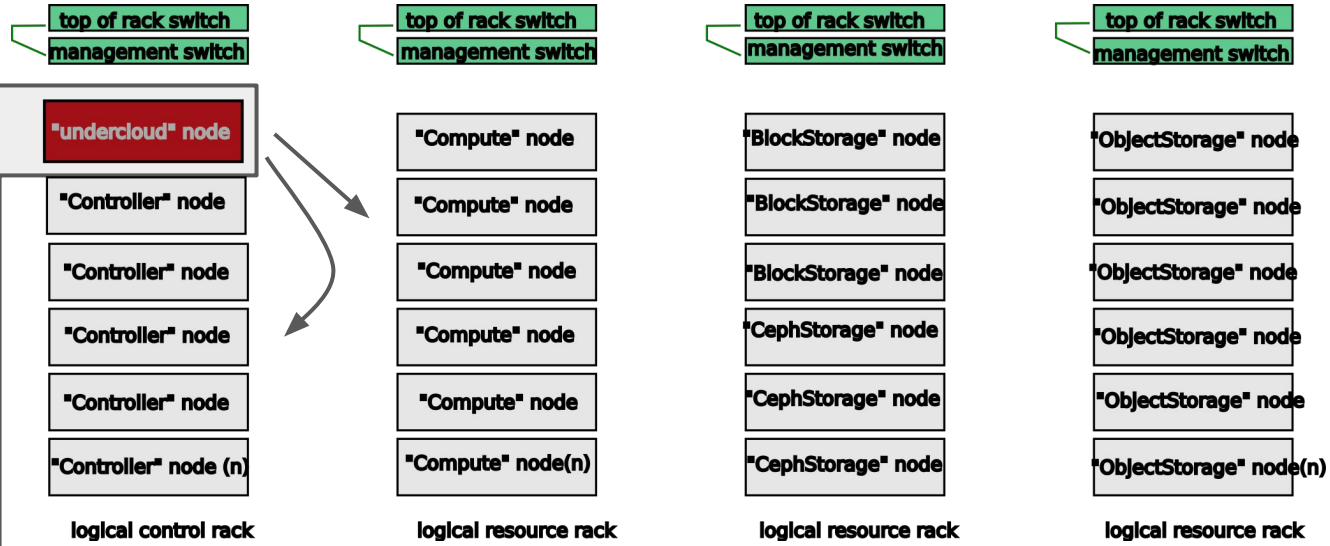
TripleO Logical View

Deployment and Management

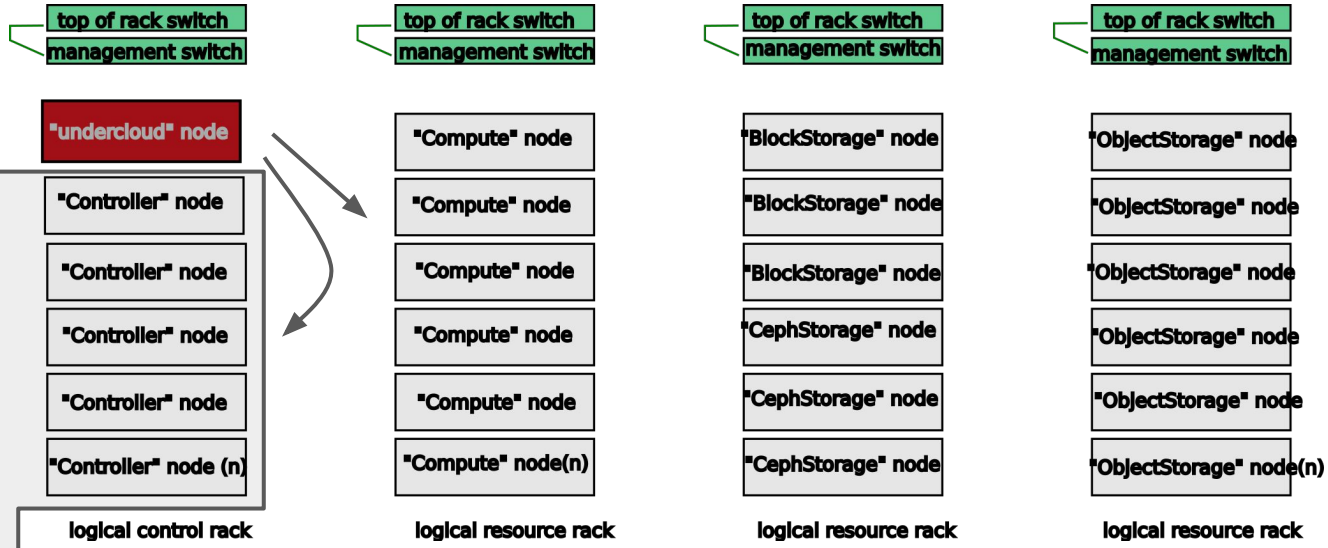
- OpenStack Services
  - Ironic (provisioning)
  - Heat (Orchestration)
  - Mistral (Workflow)
- UI/CLI
- API (based on Mistral)

Used to deploy and manage the nodes, defined by:

- Images
- Heat Templates
- Puppet Manifests
- Container images (WIP)



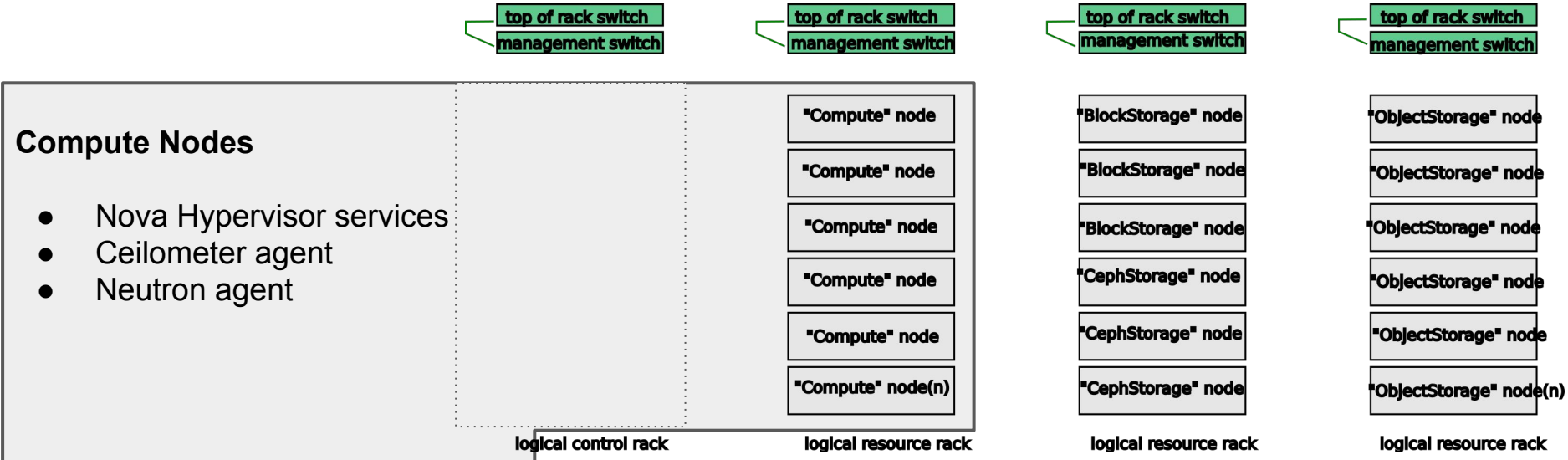
TripleO Logical View



Controller Nodes

- OpenStack Services
- RPC
- Database
- Pacemaker/Corosync
- Ceph Monitor

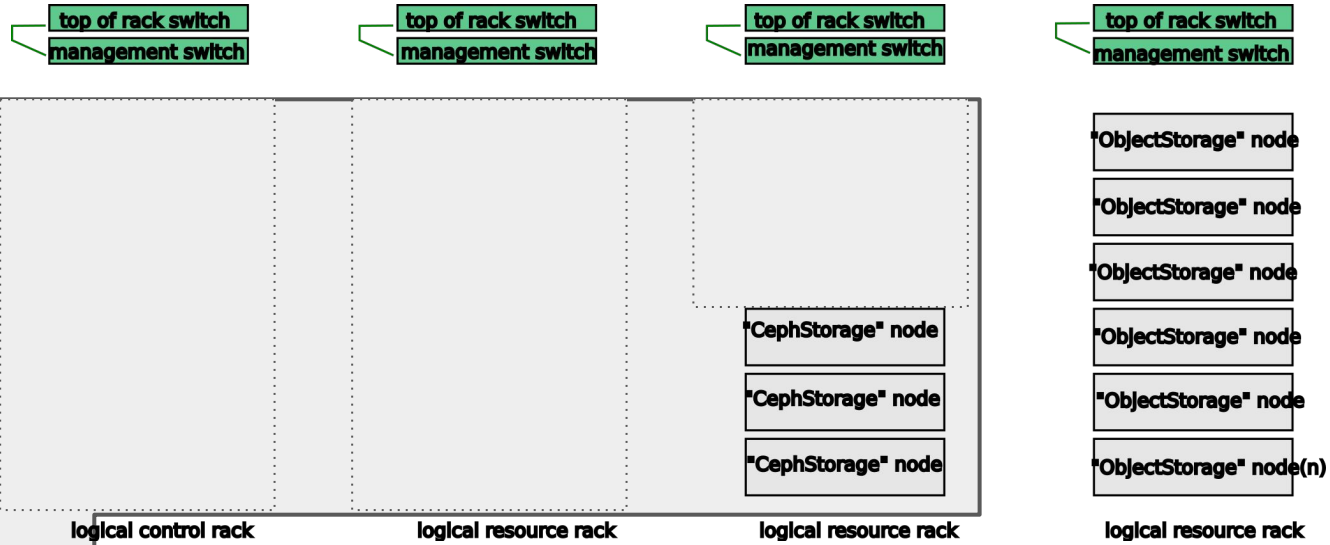
TripleO Logical View



Compute Nodes

- Nova Hypervisor services
- Ceilometer agent
- Neutron agent

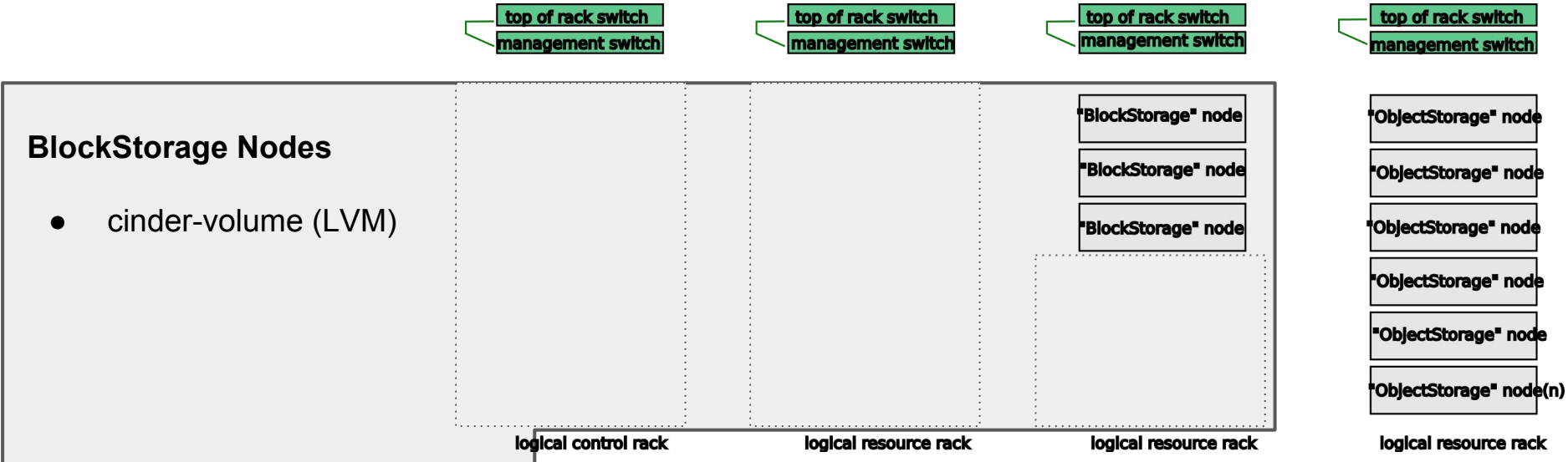
TripleO Logical View



CephStorage Nodes

- Ceph OSD
- Can be backend for:
  - Nova
  - Cinder
  - Glance
  - Gnocchi

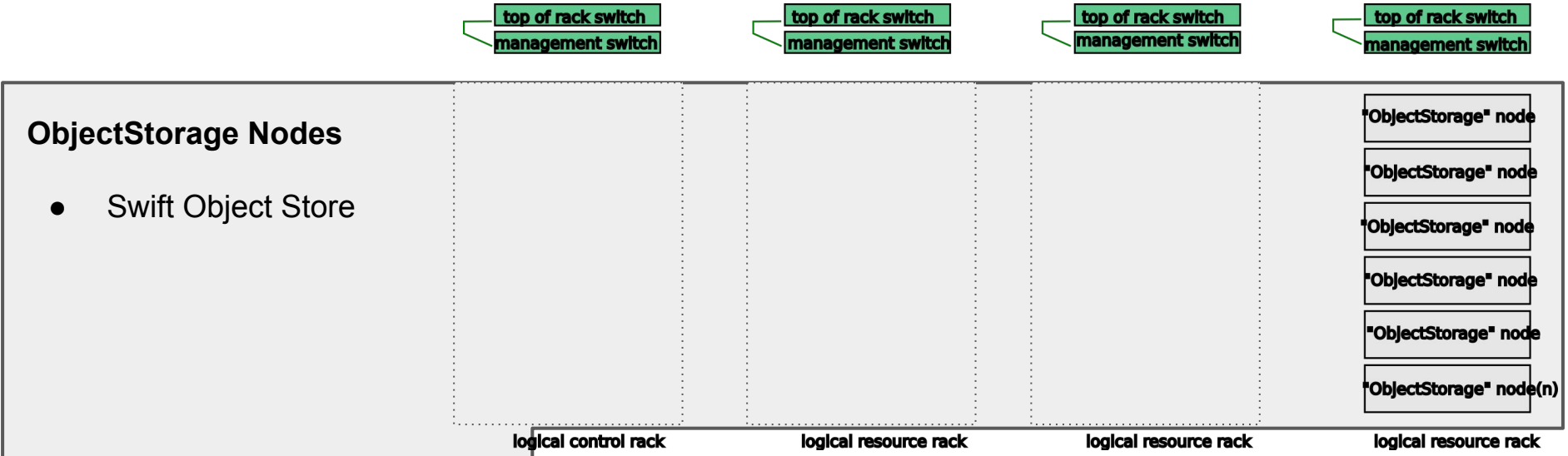
TripleO Logical View



BlockStorage Nodes

- cinder-volume (LVM)

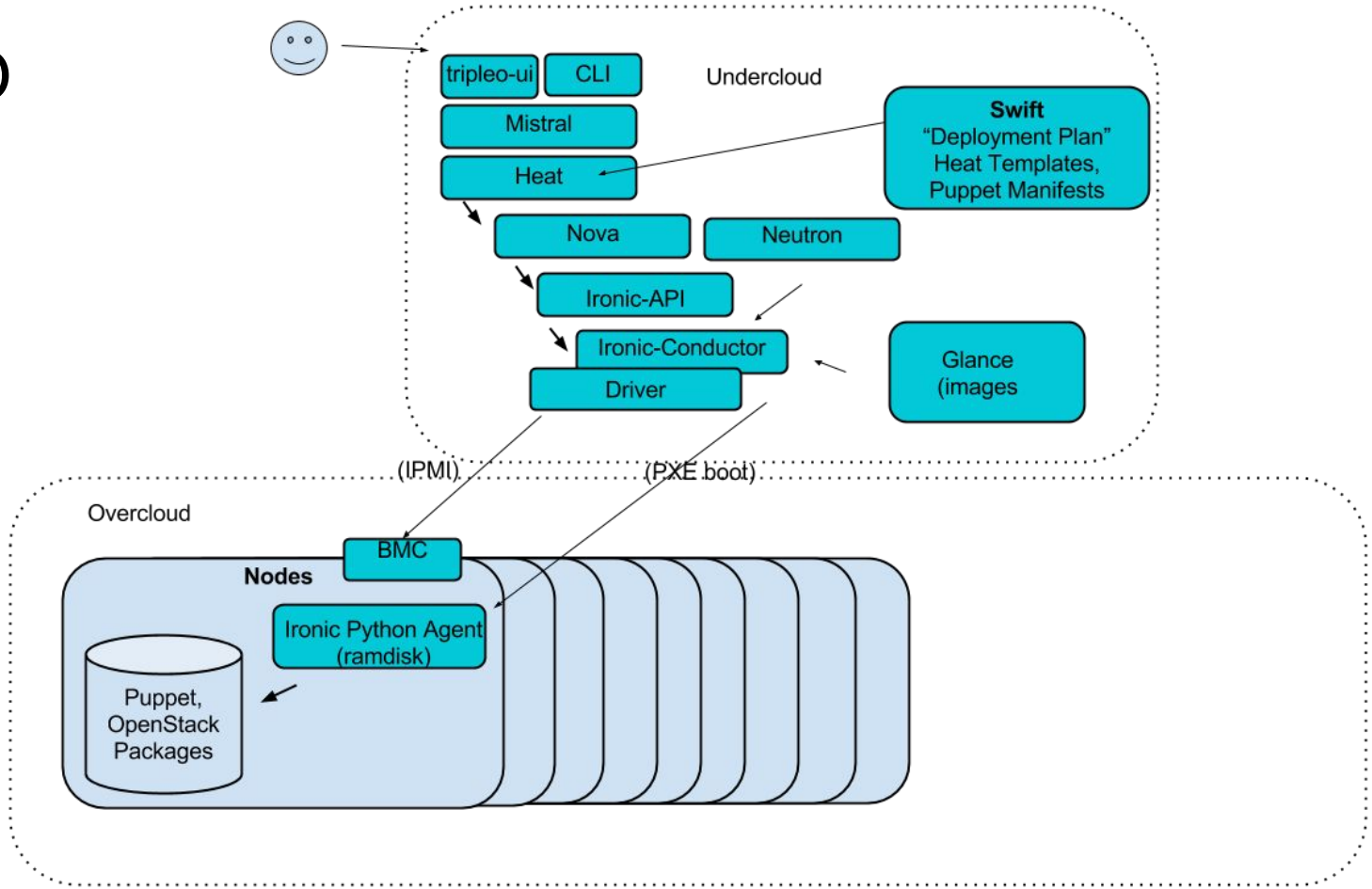
TripleO Logical View



ObjectStorage Nodes

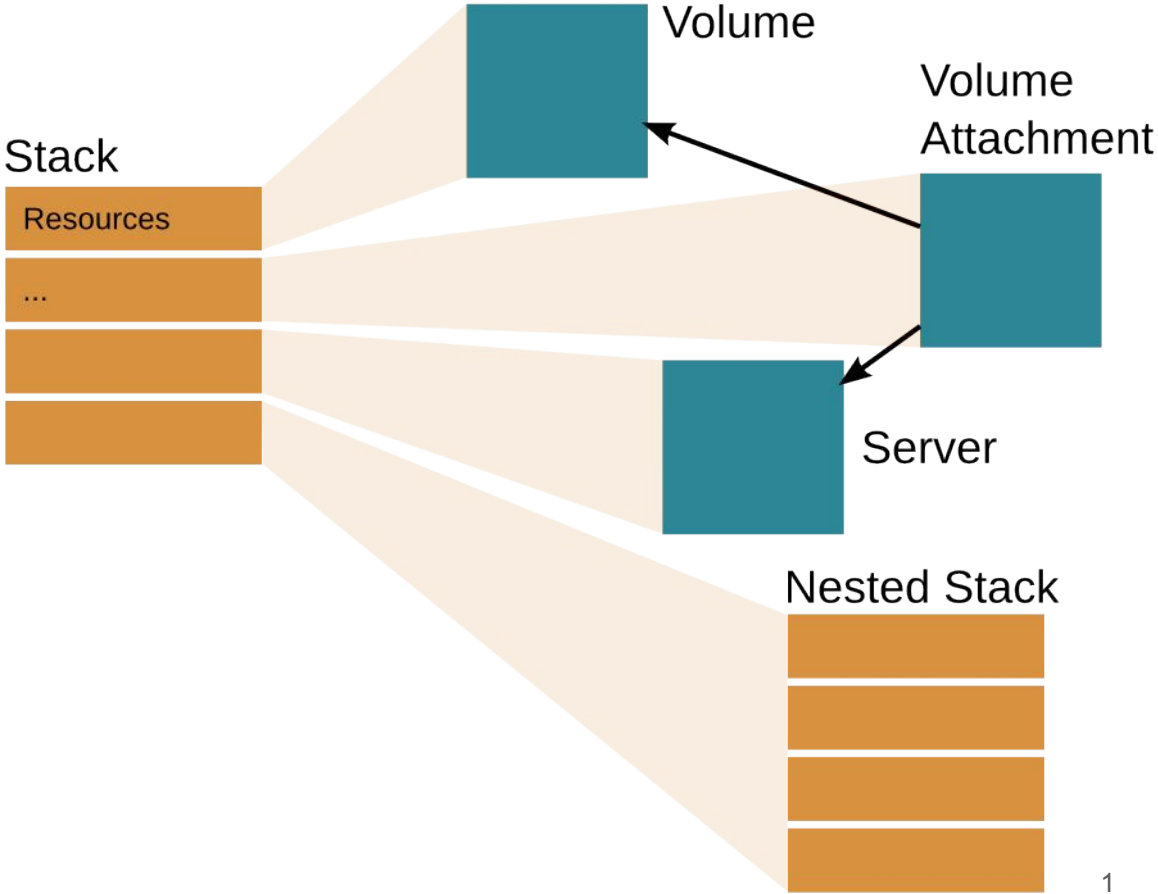
- Swift Object Store

TripleO

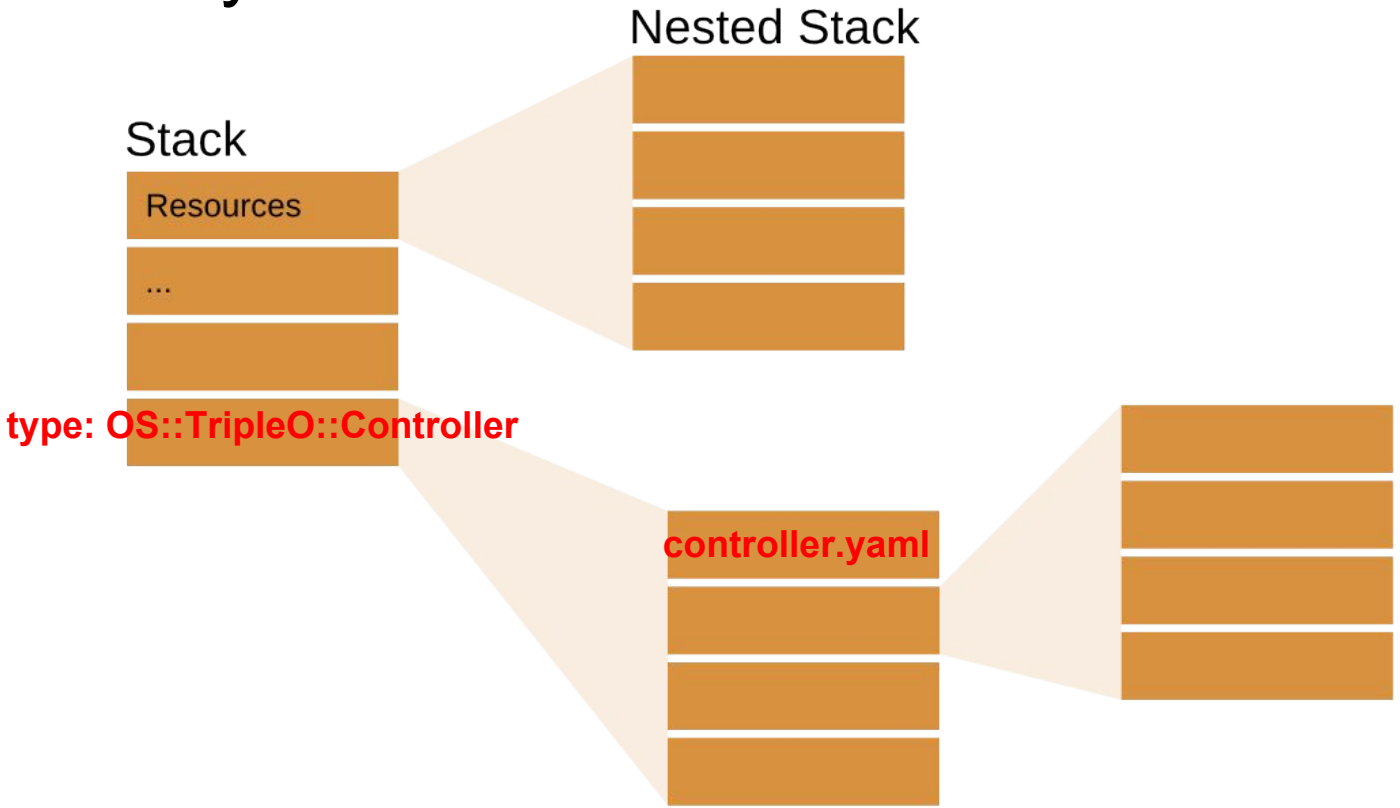

tripleo-ui
CLI
Undercloud
Mistral
Swift
"Deployment Plan"
Heat Templates,
Puppet Manifests
Heat
Nova
Neutron
Ironic-API
Ironic-Conductor
Driver
Glance
(images
(IPMI)
(PXE boot)
Overcloud
BMC
Nodes
Ironic Python Agent
(ramdisk)
Puppet,
OpenStack
Packages

Heat concepts, composability

Heat “Stack”


Volume
Volume
Attachment
Stack
Resources
...
Server
Nested Stack

Composability



Composability

- **Optional type aliases via “environment” file**

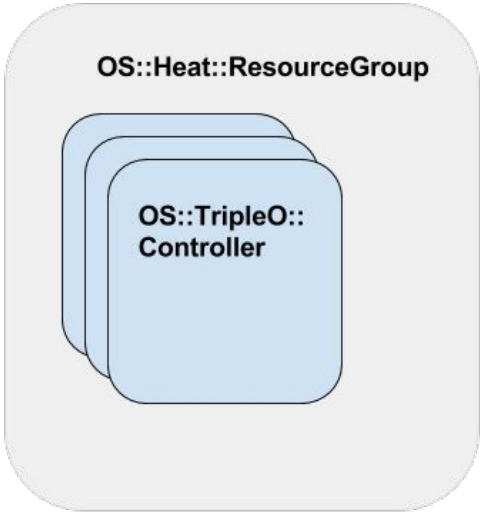
```
resources:
  Controller:
    type: OS::TripleO::Controller
    properties:
       image {get_param: image_id}
       ...

--

resource_registry:
 OS::TripleO::Controller:  puppet/controller.yaml
```

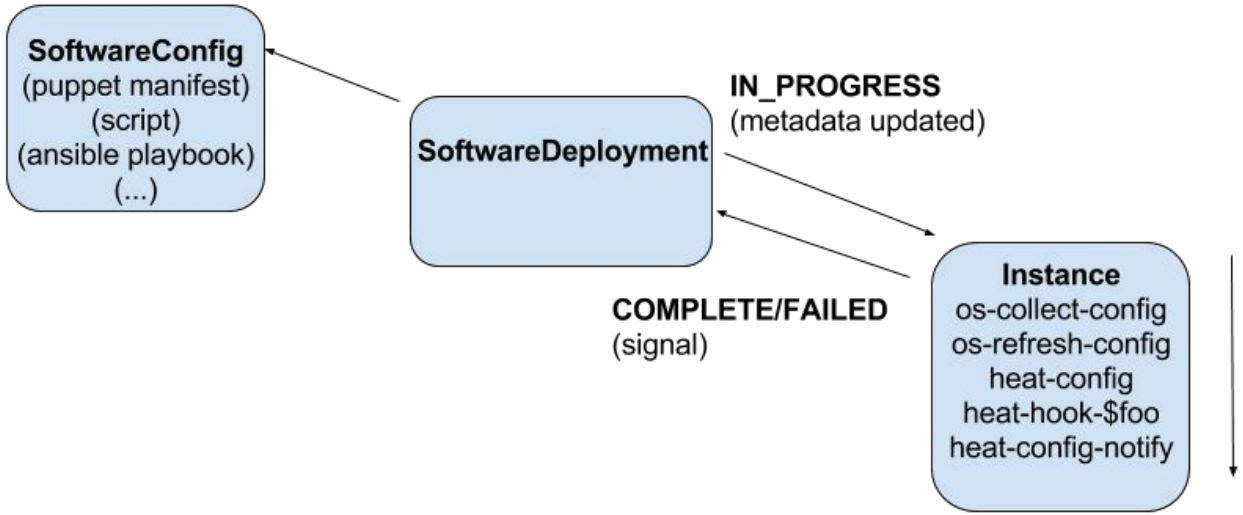
heat stack-create mystack -f the_template.yaml -e the_environment.yaml

Grouping

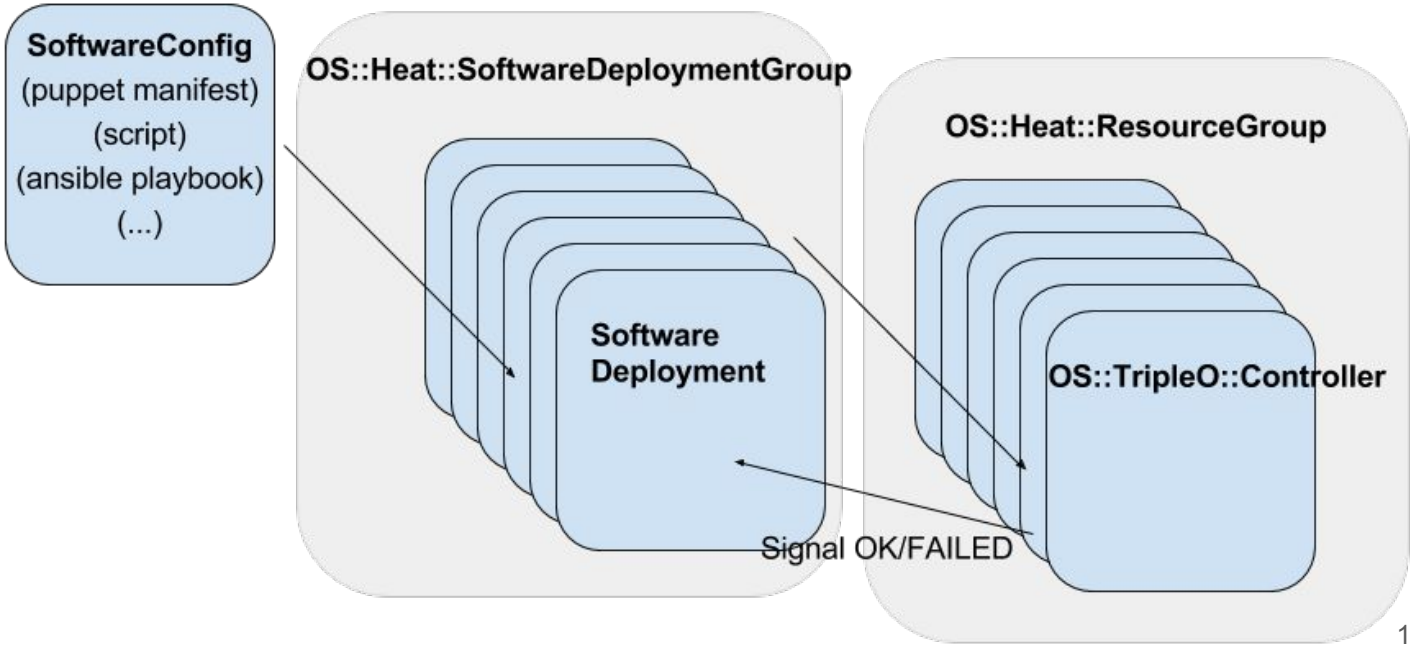



```
resources:
  Controller:
    type: OS::Heat::ResourceGroup
    properties:
      count: {get_param: ControllerCount}
      resource_def:
        type: OS::TripleO::Controller
```

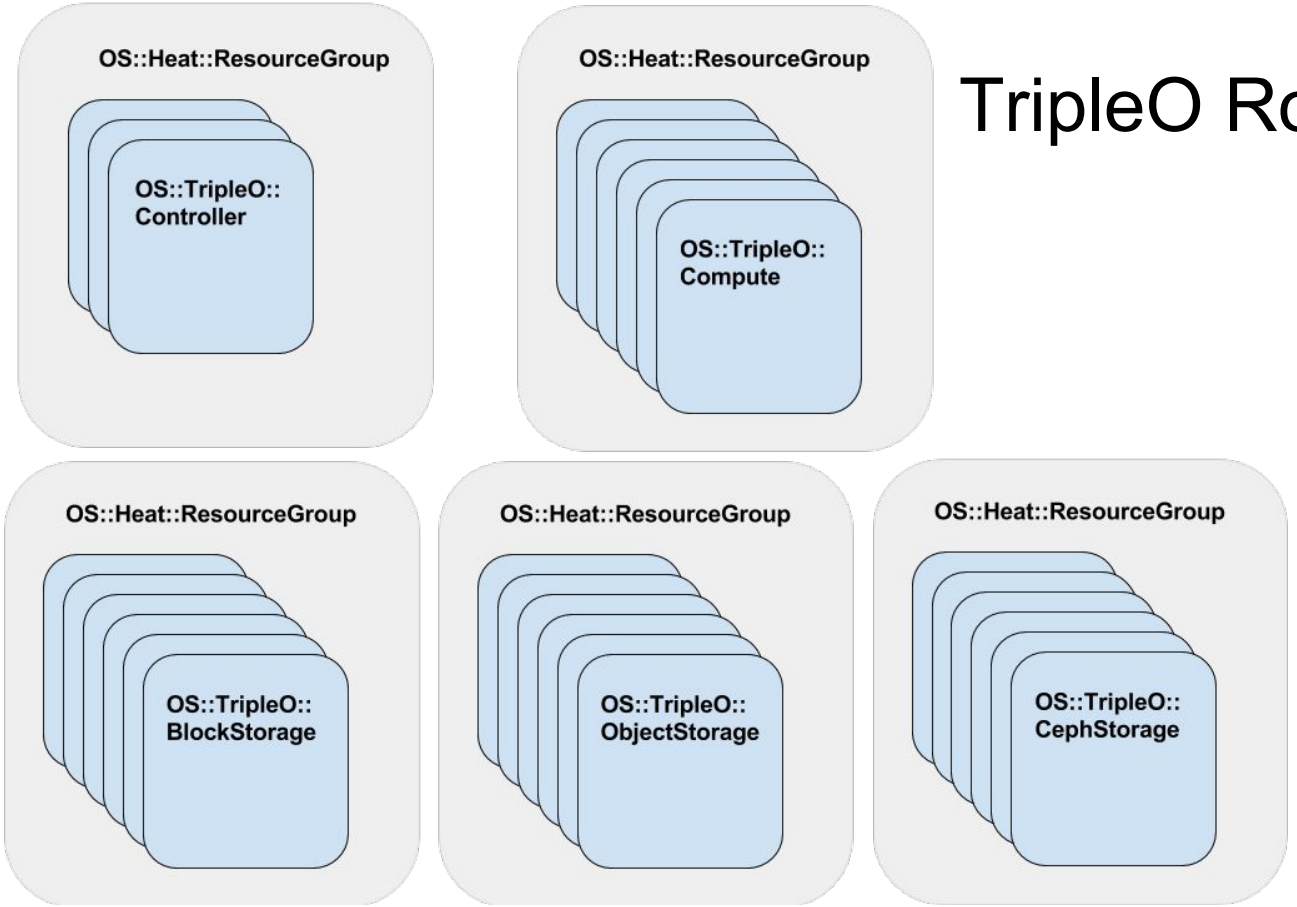
Software Configuration



Cluster Configuration

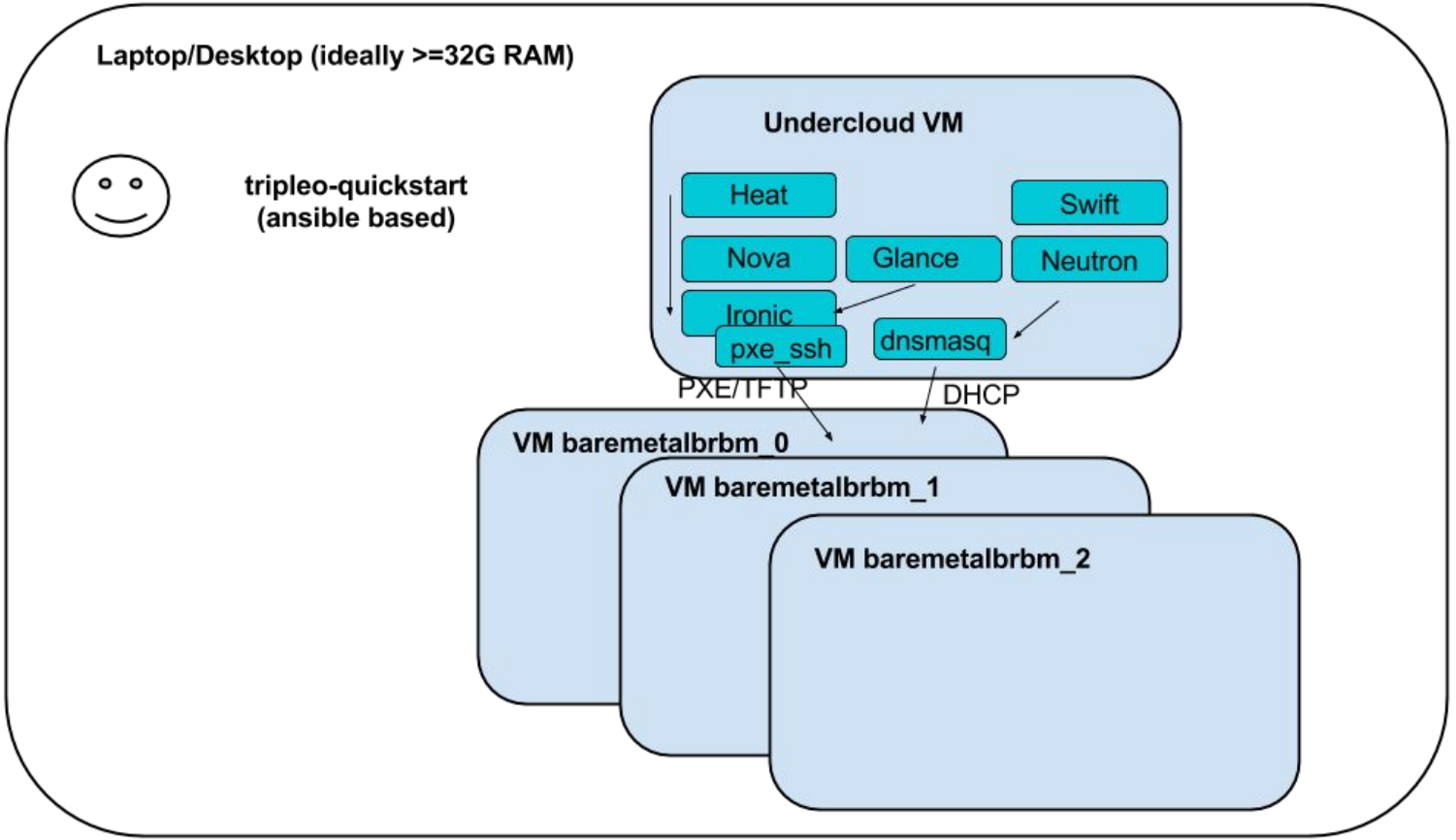


TripleO Roles

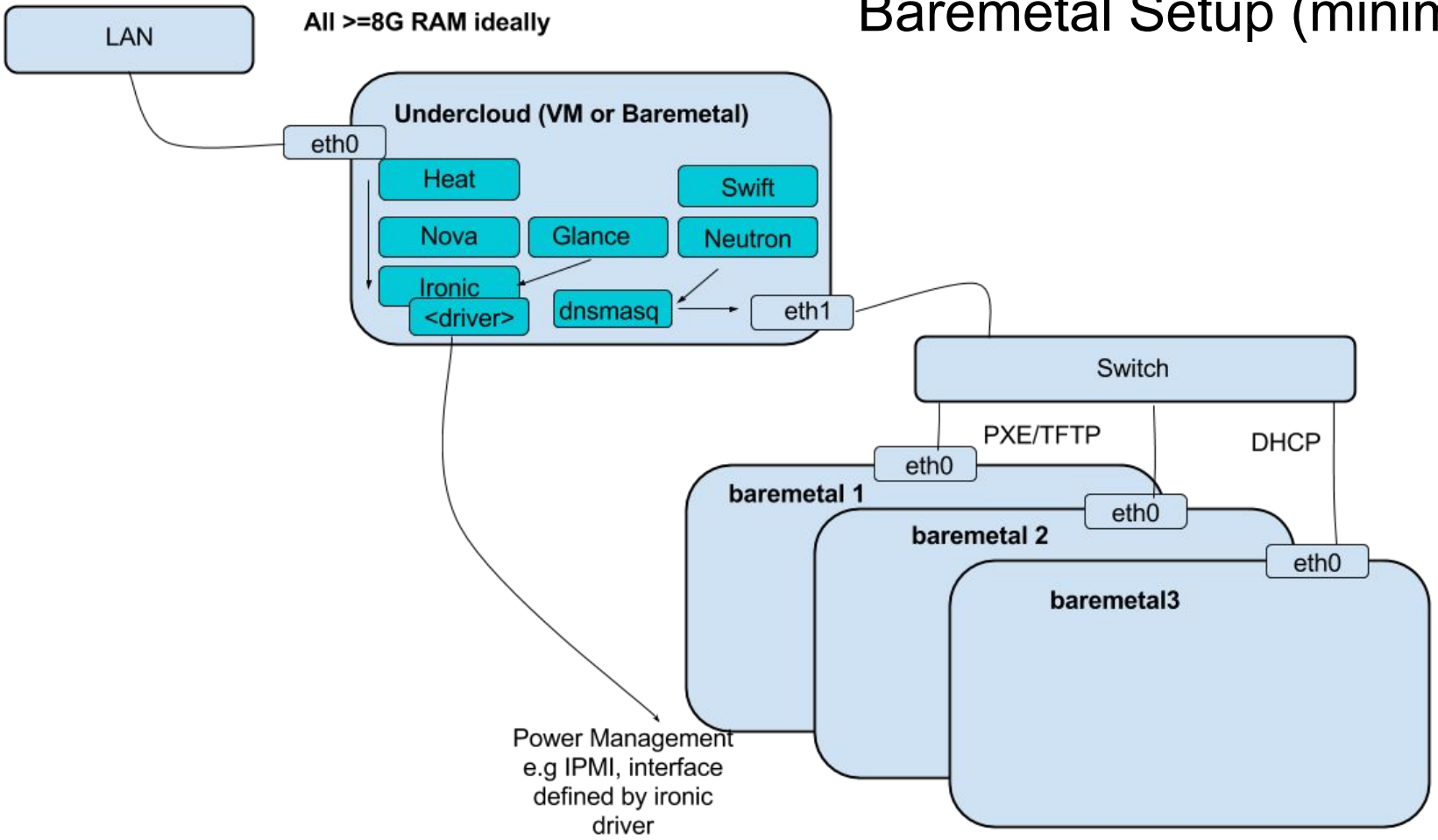


Developer Test Setups

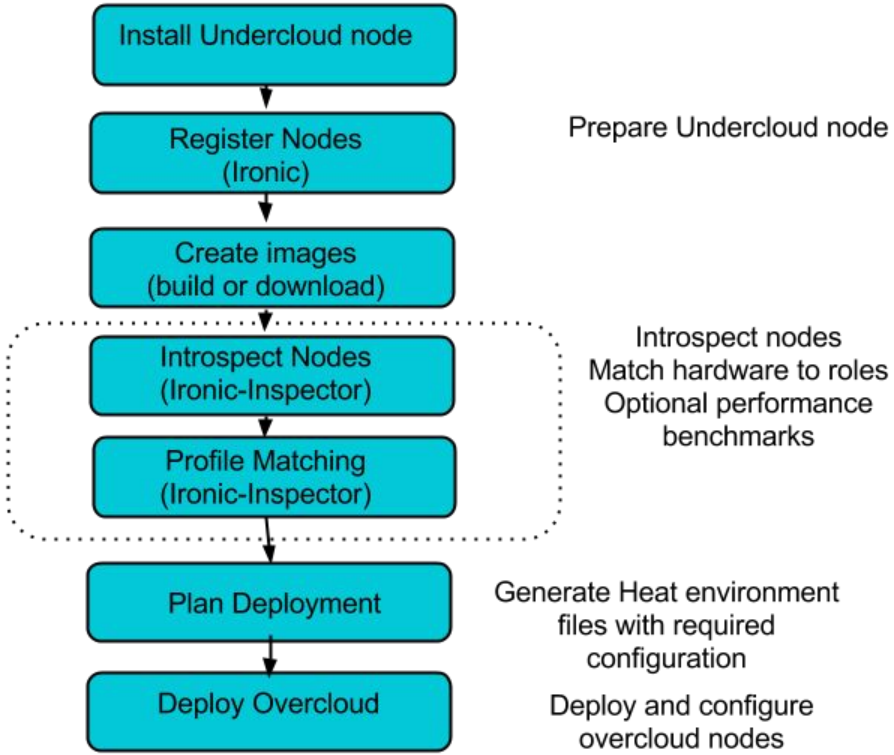
VM Setup


Laptop/Desktop (ideally >=32G RAM)
tripleo-quickstart
(ansible based)
Undercloud VM
Heat
Swift
Nova
Glance
Neutron
Ironic
pxe_ssh
dnsmasq
PXE/TFTP
DHCP
VM baremetalbrbm_0
VM baremetalbrbm_1
VM baremetalbrbm_2

Baremetal Setup (minimal)



Deployment Workflow


Install Undercloud node
Register Nodes (Ironic)
Create images (build or download)
Introspect Nodes (Ironic-Inspector)
Profile Matching (Ironic-Inspector)
Plan Deployment
Deploy Overcloud
Prepare Undercloud node
Introspect nodes
Match hardware to roles
Optional performance benchmarks
Generate Heat environment files with required configuration
Deploy and configure overcloud nodes

Development/test workflows

- openstack overcloud deploy --templates <git checkout>
  - Can re-run from a FAILED state!
  - openstack stack delete overcloud

- upload-puppet-modules -d <dir of local puppet modules>
  - tripleo-common/scripts

- **openstack stack failures list overcloud**

Development/test workflows

- You can inspect/run the config on the nodes
  - /var/lib/heat-config/heat-config-$hook
-

Developer Tools & Environment

TripleO Quickstart

http://git.openstack.org/cgit/openstack/tripleo-quickstart/

- Ansible based
- Automates dev VM setup
- Used by CI and users
- Creates VMs in libvirt unprivileged mode (su - stack!)

Basic environment, getting started

1. SSH onto your allocated node

2. Ensure the host is updated
   yum -y update && /sbin/reboot

3. Configure SSH keys and check passwordless ssh is working
   ssh-keygen
   cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys
   ssh root@testhost uname -a

Basic environment, getting started

3. Install needed tools & clone/run tripleo-quickstart

yum -y install git vim-enhanced screen <your favourite tools here>

git clone https://github.com/openstack/tripleo-quickstart

bash tripleo-quickstart/quickstart.sh --install-deps

bash tripleo-quickstart/quickstart.sh --release master-tripleo-ci testhost

Note this only deploys two nodes by default, you can add -c config.yaml

Basic environment, getting started

4. SSH to installed undercloud and configure things

ssh -F /root/.quickstart/ssh.config.ansible undercloud

Then follow steps from the docs:

http://tripleo.org/basic_deployment/basic_deployment_cli.html#upload-images

Basic environment, getting started

5. Deploy overcloud (from undercloud)

. stackrc

openstack overcloud image upload

openstack baremetal import instackenv.json

openstack baremetal introspection bulk start # Optional

neutron subnet-list

neutron subnet-update <ID> --dns-nameserver 192.168.122.1 (or dns from virthost resolv.conf)

The above steps are one-time, now we're ready to deploy

openstack overcloud deploy --templates

Check out ironic node-list, nova flavor-list and glance image-list!

Basic deployment, passing parameters

- Basic deployment
  - source stackrc
  - openstack overcloud deploy --templates **-e my_params.yaml**

my_params.yaml:

```
parameter_defaults:
  NovaComputeLibvirtType: qemu
  ControllerCount: 1
  ComputeCount: 1
  OvercloudControlFlavor: control
  OvercloudComputeFlavor: compute
```

Basic environment, customizing

Quickstart only deploys two nodes by default, you can add -c config.yaml

```
overcloud_nodes:
  - name: control_0
    flavor: control
  - name: compute_0
    flavor: compute
  - name: ceph_0
    flavor: ceph
  - name: swift_0
    flavor: objectstorage
```

http://docs.openstack.org/developer/tripleo-quickstart/configuration.html

Where to find out more

Links & where to find out more

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