LEARN MORE ABOUT

openstack™
CLOUD SOFTWARE

OpenStack Fundamentals Workshop
(Deploy OpenStack Newton w/ Kolla-Ansible)

@OpenStack-Cologne Meetup

Meetup
STARTPLATZ

2016 June, 29th
Arash Kaffamanesh
@kaffamanesh

THE CLOUD EXPERT

About OpenStack-Cologne Meetup

- Initiated: April 2014
- Meetups: 8
- Stackers: 271
- Location: Cologne
- Sponsors:
  - Clouds Sky GmbH
  - HPE
  - K3 Innovationen GmbH
  - teuto.net
  - HyperHQ Inc.
  - more sponsors are more than welcome ;-)

Agenda

- OpenStack History

- OpenStack Projects, Components & Services

- OpenStack Demo

- **HandsOn OpenStack deployment with Kolla-Ansible (dockerized)**

- OpenStack Deployment with TripleO (for HA'ed deplyoments) → next meetup

- **Introduction to OpenStack Administration Fundamentals and Certification by Linux Foundation**
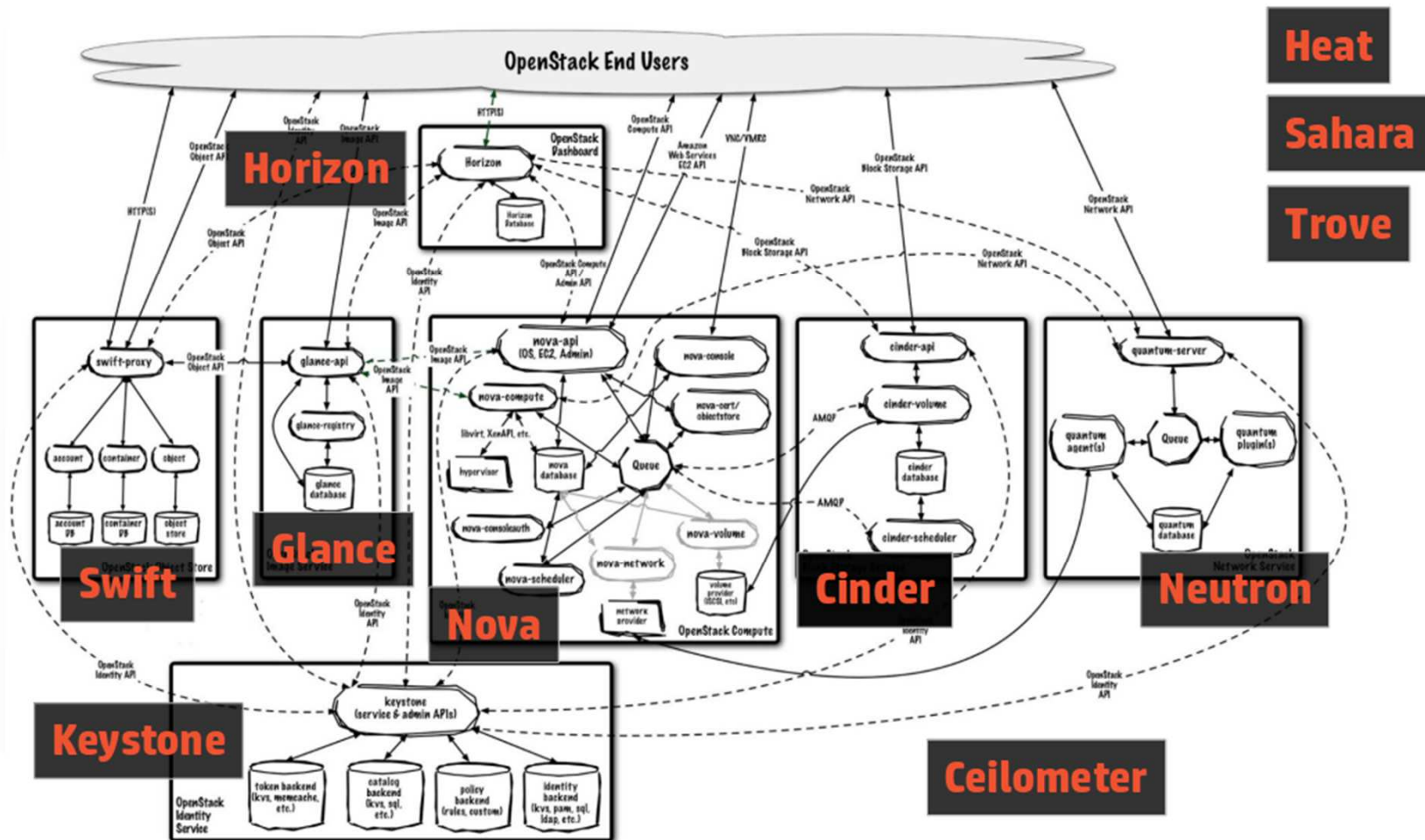
OpenStack History

- Was initiated by NASA and Rackspace in June 2010
- OpenStack Foundation founded in 2012
- OpenStack is one of the most popular Open Source Cloud Operating Systems
- Is the largest open source **project** helping to build your own private cloud environment or to build public or hybrid cloud offerings
- Over 1 million lines of code

OpenStack Projects, Services and Infrastructure Components

- OpenStack Core, Optional & Independent Projects
- Major Components of OpenStack
- Services provided by core projects
- OpenStack Infrastructure Components
- A project provides one or more services
- Currently 9 core projects
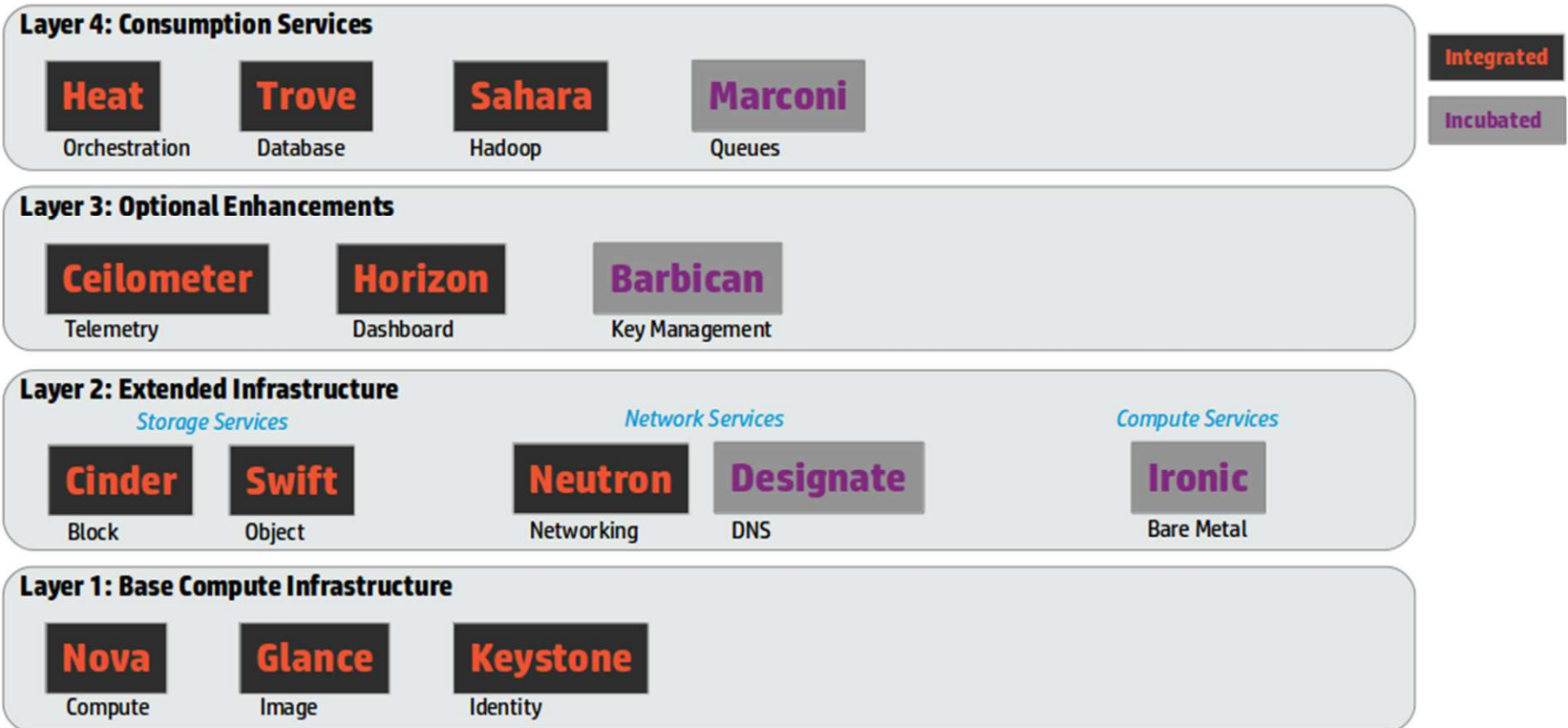- More than 20 Additional projects

OpenStack Components

OpenStack as Layers

OpenStack as Layers (Compute Centric View)

Layer 4: Consumption Services

Heat	**Trove**	**Sahara**	**Marconi**
Orchestration	Database	Hadoop	Queues

Integrated
Incubated

Layer 3: Optional Enhancements

Ceilometer	**Horizon**	**Barbican**
Telemetry	Dashboard	Key Management

Layer 2: Extended Infrastructure

Storage Services   *Network Services*   *Compute Services*

Cinder	**Swift**	**Neutron**	**Designate**	**Ironic**
Block	Object	Networking	DNS	Bare Metal

Layer 1: Base Compute Infrastructure

Nova	**Glance**	**Keystone**
Compute	Image	Identity

http://hackstack.org/x/blog/2013/09/05/openstack-seven-layer-dip-as-a-service/

Source: http://hackstack.org/x/blog/2013/09/05/openstack-seven-layer-dip-as-a-service/

OpenStack Component Naming

- Official "component name", e.g. "Compute Service", describes the components function
- Code name (nice name, e.g. "Nova")

OpenStack Core Services

- Nova (Compute)
- Glance (Image Services)
- Keystone (Auth. / Authz API Service)
- Cinder (Block Storage)
- Horizon (Frontend)
- Neutron (Software Defined Networking)
- Swift (Object Storage)
- Ceilometer (Telemetry)
- Heat (Orchestration)

OpenStack Core Services
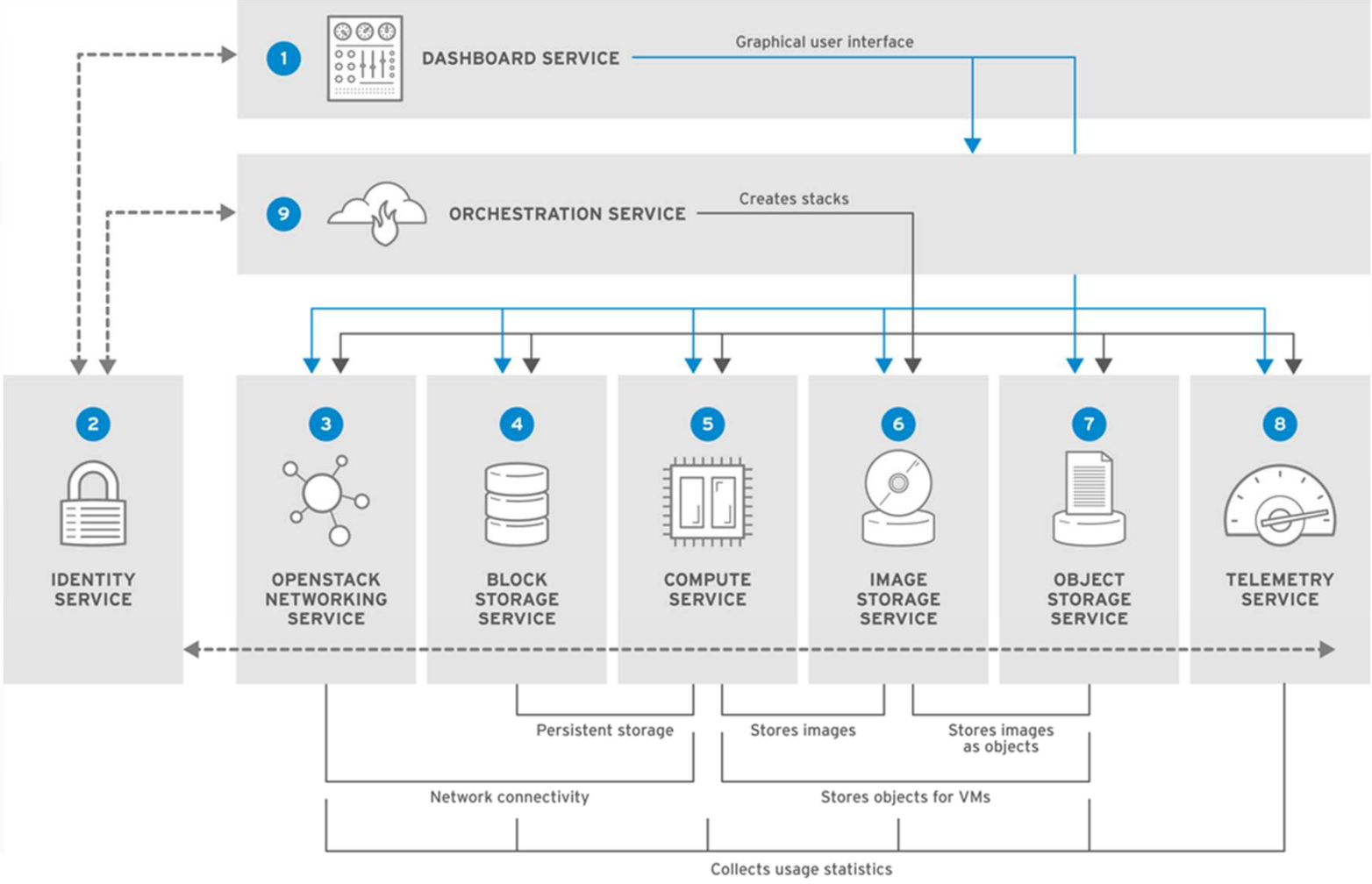
Source: Red Hat OpenStack Platform 8 Architecture Guide

OpenStack Demo

OpenStack Additional Services

- Ironic (bare-metal provisioning)
- Trove (DBaaS)
- Sahara (Data Processing, Hadoop aaS)
- Magnum (Containers as a Service w/ Docker Swarm, Kubernetes)
- Manila (Fileshare)
- Murano (Application Catalog)
- etc..

OpenStack Infrastructure Components

- [Ceph](#) implementation for Cinder, Glance and Nova

- [Openvswitch](#) and Linuxbridge backends for Neutron

- [MongoDB](#) as a database backend for Ceilometer and Gnocchi

- [RabbitMQ](#) as a messaging backend for communication between services.

- [HAProxy](#) and [Keepalived](#) for high availability of services and their endpoints.

- [MariaDB and Galera](#) for highly available MySQL databases

- [Heka](#) A distributed and scalable logging system for openstack services.

OpenStack Reference Architecture (Nodes and Roles)

- The Cloud Controller (CC) / head node

- The API node

- The Network Controller (NC) node

- The Compute nodes

- The Storage Controller node

The Cloud Controller (CC)

- The CC hosts:

    - MariaDB

    - Mongo NoSQL DB

    - RabbitMQ (non-OpenStack services: SPOF, will be replaced by ØMQ)

Note: CC is typically not be reachable via public

The API Node

- The API node hosts:

  - nova-api

  - keystone-api

  - cinder-api

  - neutron-server

Note: API node is typically reachable via public

The Network Controller Node

- Runs neutron networking services:

  ○ L3 agent

  ○ DHCP agent

  ○ Metadata agent

Note: NC is connected to all physical networks

The Compute Nodes

- Runs the hypervisor (e.g. KVM)

- Is connected to the management- and internal VM physical networks

Note: the compute nodes don't need to be reachable via public addresses

The Storage Controller Nodes

- Runs Swift Proxy

- Runs Ceph components with load balancing via CRUSH

Nova (Compute Service)

- Provides Virtualization to an OS Cloud
- Start, stop, control VMs
- Keeps track of all VMs for e.g. load balancing
- Reports VM states to the cloud

Components of Nova

- Nova compute
- Nova api
- Nova scheduler
- Nova cert
- Nova objectstore
- Nova conductor
- Nova consoleauth
- Nova novncproxy

Hypervisors Supported by Nova

- KVM
- LXC
- Qemu (non-KVM accelerated version)
- VMWare vSphere
- Xen
- IBM PowerVM
- Microsoft Hyper-V

Note: Hypervisor mixture is possible, but w/o live-migration capabilities

Glance (Image Service)

- Manages VM images and their administration and storage via **glance-api** and **glance-registry** and MariaDB
- **glance-api** is used to upload images
- **glance-registry** manages the Glance database and provides the information about the stored images and their location
- Images can be stored in Swift, S3, Rados or on the file system
- W/ glance-cli you can list, upload, delete or snapshot images, as well as w/ Horizon

Keystone (Identity Service)

- Provides Authentication & Authorization for members (tenants == projects) and admins
- Other OS Services need to identify with Keystone too
- Authentication is token based
- Every user or service needs to acquire a token, which is used to send commands to Keystone and other services
- Tokens are only valid for a limited time
- Keystone is aware of the ever changing location of endpoints of other services!

Cinder (Volume Service)

- By default VMs are assigned ephermeral storage on the hypervisor node
- If the VM is started somewhere else or gets deleted, the ephermeral data is lost!
- Cinder provides VMs with block storage
- One can boot even VMs off of volumes
- Cinder supports software based storage such as Ceph, NFS, etc., or
- Hardware-base storage such as SolidFire, Nexenta, etc..

Cinder (Volume Service)

- By default VMs are assigned ephermeral storage on the hypervisor node
- If the VM is started somewhere else or gets deleted, the ephermeral data is lost!
- Cinder provides VMs with block storage
- One can boot even VMs off of volumes
- Cinder supports software-based storage such as Ceph, NFS, GlusterFS, etc., or hardware-based storage such as SolidFire, Nexenta, EMC, etc..

Cinder Services

- Cinder-API is the interface to talk to Cinder
- Cinder-Volume creates and deletes volumes and manages the storage backend
- Cinder-Scheduler is responsible to coordinate storage access in steups which have more than one storage backend

Horizon (OS Dashboard)

- Horizon is the web-user interface for doing things such as:
  Starting, stopping, deleting, live-migrating or creating snapshots of VMs
  Managing projects (tenants)
  Connecting to VM's console
  Creating Security Groups, Networks and much more …

Neutron (Software Defined Networking)

- Neutron uses Open vSwitch and OpenFlow plugins by default to manage the switch and network packet flows
- Based on OpenFlow rules, packets are forwarded between VMs on the same node or across physical nodes
- Other plugins are:
  Cisco UCS plugins
  Midonet
  ML2 (Modular Layer 2) SDN Plugins
  etc..

OpenStack Services Distribution and Scalability across nodes (simple)
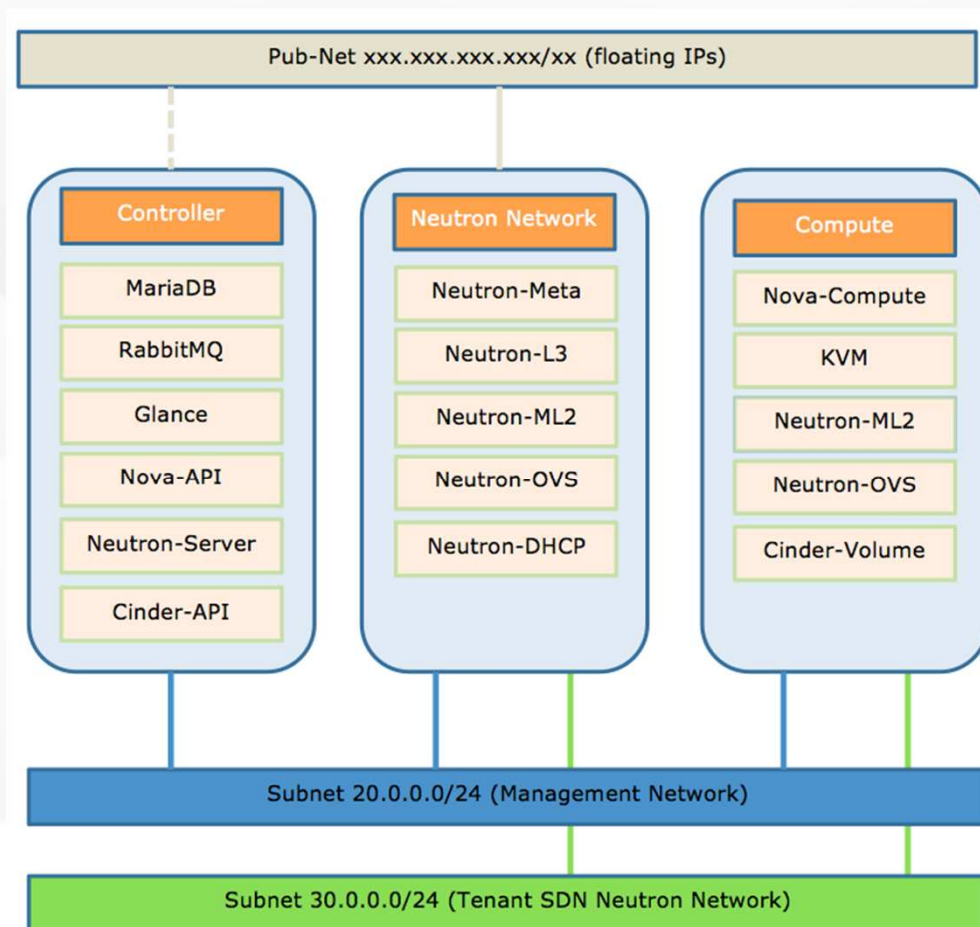
Some OpenStack Distros (products)

Distros	Support	OS	Hypervisor
Debian OS	community	Debian	KVM / Xen
HP Helion	Y	Hlinux (Debian based)	KVM / ESX / Hyper-V
IBM Cloud Manager with OpenStack	Y	RHEL	KVM / PowerKVM
Mirantis Fuel	Y	Ubuntu / RHEL	KVM / ESX / Xen
Oracle OS	Y	Oracle Linux / Solaris	KVM / **Solaris Zone**
Piston OS (Cisco)	Y	**Iocane Linux**	KVM / ESX / Xen
Rackspace OS (OpenCenter)	Y	Ubuntu	KVM / ESX / Xen
Red Hat OSP	Y	RHEL	**Red Hat's KVM**
Red Hat RDO	community	CentOS / Fedora / RHEL	KVM / LXC / **Nova-Docker**
SUSE Cloud	Y	SUSE	KVM / ESX / Xen
Ubuntu OpenStack	Y	Ubuntu	KVM / ESX / **LXD**
VMware VIO	Y	vSphere	ESX

OpenStack Deployment Options (for Developers)

- DevStack
- OpenStack Ansible (OSA)
- RDO Packstack (for POCs)
- RDO TripleO (for production)
- HPE Helion TripleO
- Kolla-Ansible (dockerized, uses DLRN RDO packages and more..)
- Kolla-Kubernetes (just borne)
- Kolla-Rancher (just borne, very exciting!!!)

Kolla (Ansible + Docker OS Components and Services

- Kolla provides **Docker** containers and **Ansible** playbooks to meet Kolla's mission
- Kolla's mission is to provide production ready containers and deployment tools for operating OpenStack clouds.

Links for AIO and Multi-Node deployments:
https://github.com/openstack/kolla
http://docs.openstack.org/developer/kolla/

Kolla w/ **Cisco!Devnet OpenStack Image**

- HandsOn Session:
- Get it, run it and enjoy OS in less than 30 minutes :-)
  https://cisco.app.box.com/v/KollaCLBerlin2016

Kolla AIO Deployment Demo

- Using CentOS 7.2.1511 (Core) on bare-metal
- With 2 Nics
  enp2s0 (public interface)
  eno1 (internal with no IP configured)
- 16 GB RAM
- 8 Core Intel(R) Xeon(R) CPU E3-1230 V2 @ 3.30GHz

Kolla AIO Deployment Steps I

- yum -y update
- reboot
- yum -y install epel-release
- yum -y install python-pip
- curl -sSL https://get.docker.io | bash
- mkdir -p /etc/systemd/system/docker.service.d
- tee /etc/systemd/system/docker.service.d/kolla.conf <<-'EOF'

[Service]

MountFlags=shared

EOF

- systemctl daemon-reload
- systemctl restart docker
- yum install -y python-docker-py
- yum -y install ntp
- systemctl enable ntpd.service
- systemctl start ntpd.service
- systemctl stop libvirtd.service
- systemctl disable libvirtd.service
- yum -y install ansible

Kolla AIO Deployment Steps II

- systemctl daemon-reload
- yum install git -y
- git clone https://git.openstack.org/openstack/kolla
- pip install kolla/
- cd kolla/
- cp -r etc/kolla /etc/
- yum -y install python-devel libffi-devel openssl-devel gcc
- pip install -U python-openstackclient python-neutronclient

- ## kolla-build → options

- docker images
- kolla-genpwd
- vi /etc/kolla/globals.yml

network_interface: "enp2s0"
neutron_external_interface: "eno1"
kolla_install_type: "binary"
kolla_base_distro: "centos"
openstack_release: "**3.0.0**"

- **kolla-ansible prechecks**
- **kolla-ansible deploy**

Kolla AIO Deployment Steps III

- cd /etc/kolla/
- kolla-ansible post-deploy
- cat admin-openrc.sh
- cp /etc/kolla/admin-openrc.sh openrc
- **source openrc**
- docker ps -a
- vi /usr/share/kolla/init-runonce
- adapt to your pub. network: **neutron subnet-create --name 1-subnet --disable-dhcp --allocation-pool start=x.x.x.x,end=x.x.x.x public1 x.x.x.x/26 --gateway x.x.x.x**
- . /usr/share/kolla/init-runonce
- docker exec -i -t neutron_server /bin/bash
- neutron net-list
- nova boot --flavor m1.tiny --image cirros --key-name oskey --nic net-id=36bbbe4b-64e0-4d87-9fda-a9f254acbc3c test
- glance image-list
- nova list

- OpenStack CLI Cheat Sheet: http://docs.openstack.org/user-guide/cli_cheat_sheet.html

Kolla AIO: Access Horizon

- iptables -F —> unless horizon is not accessible
  - Access horizon through the public IP: http://x.x.x.x
  - You'll find the admin credential in openrc file

Kolla AIO: Cleanup

- . /usr/share/kolla/tools/cleanup-containers
- kolla-ansible deploy
- kolla-ansible post-deploy
- . /usr/share/kolla/init-runonce
- **Boot Cirros (**if sometimes :-) not possible over horizon, current BUG):
- neutron net-list (to find the net-id)
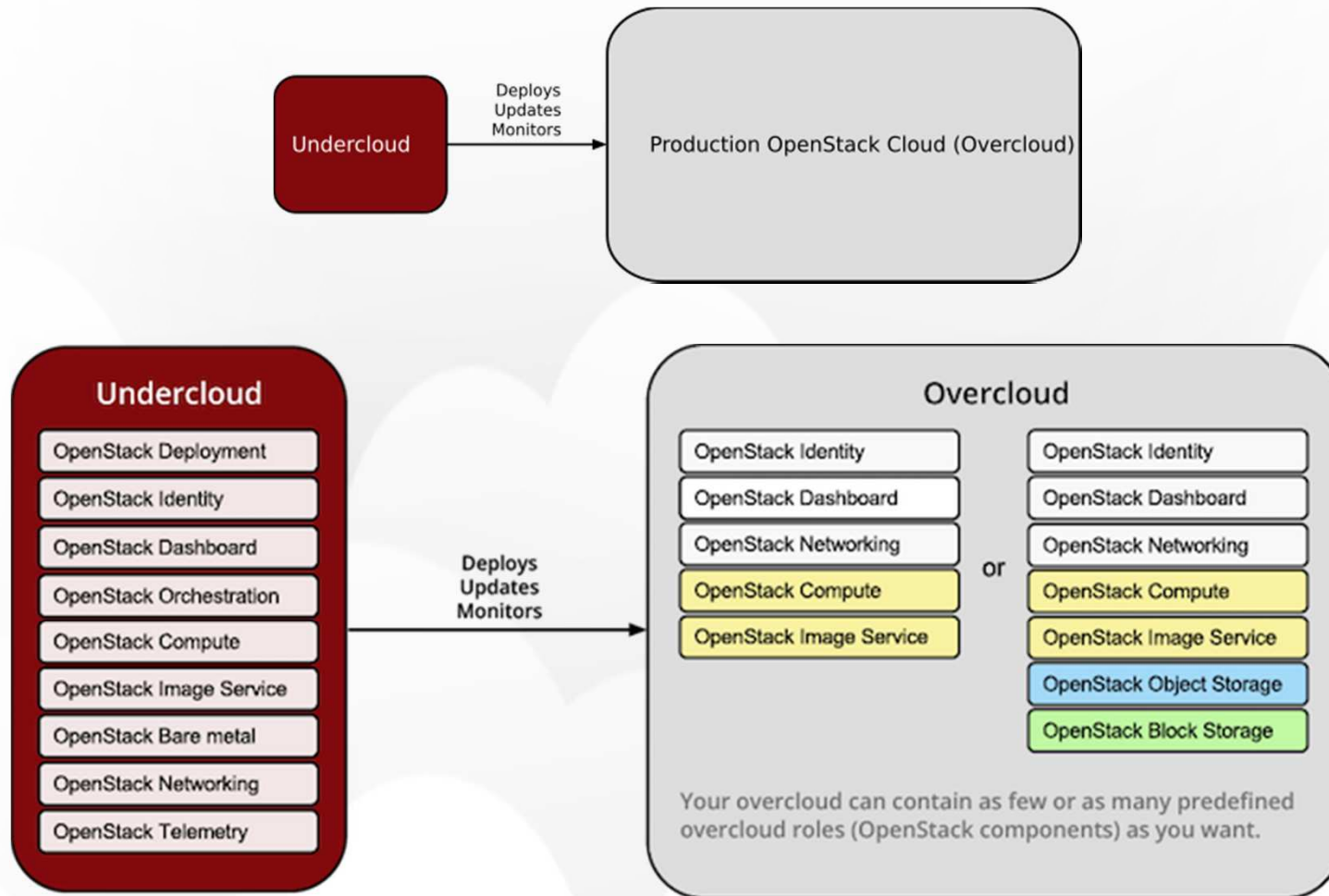- nova boot --flavor m1.tiny --image cirros --key-name oskey --nic net-id=xxxxxxxxx cirros1

Kolla Multi-Node Setup

- Presented by October workshop :-)
- Probably running with:
    - Kolla-Kubernetes
    - Kolla-Rancher, or
    - Kolla-Mesos

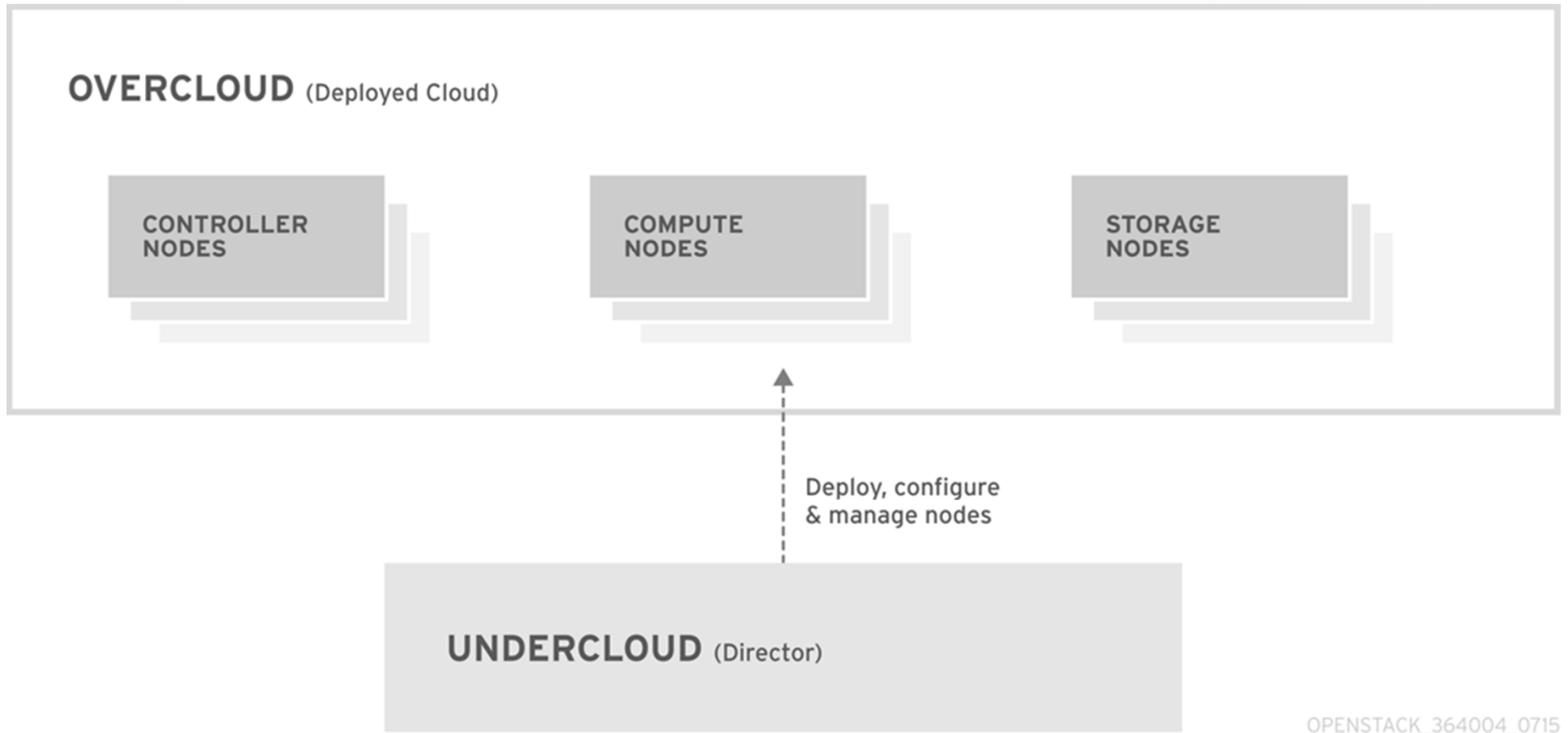# TripleO (**O**penStack **O**n **O**penStack)

- TripleO stands for OpenStack On OpenStack and is an OpenStack project for deploying production ready OpenStack Clouds with its own tools such as Heat, Ironic and Nova.
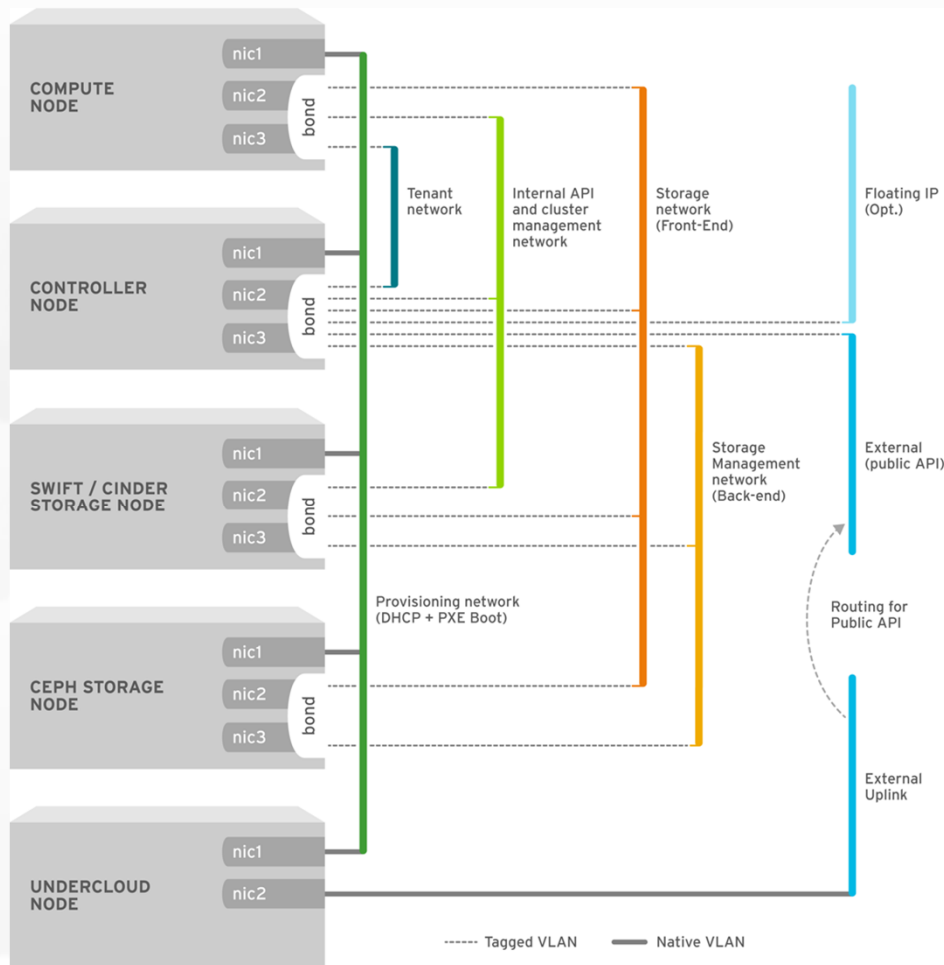
TripleO Architecture

Source: http://docs.openstack.org/developer/tripleo-docs/introduction/architecture.html

TripleO (OpenStack On OpenStack)

OVERCLOUD (Deployed Cloud)

CONTROLLER NODES

COMPUTE NODES

STORAGE NODES

Deploy, configure & manage nodes

UNDERCLOUD (Director)

OPENSTACK_364004_0715

Source: https://access.redhat.com/documentation/en/red-hat-openstack-platform/version-8/director-installation-and-usage/#chap-Introduction

TripleO (Network Planning)

Source: https://access.redhat.com/documentation/en/red-hat-openstack-platform/version-8/director-installation-and-usage/#sect-Planning_Networks

TripleO High Availability Example

Source: https://access.redhat.com/documentation/en/red-hat-openstack-platform/version-8/understanding-red-hat-openstack-platform-high-availability/
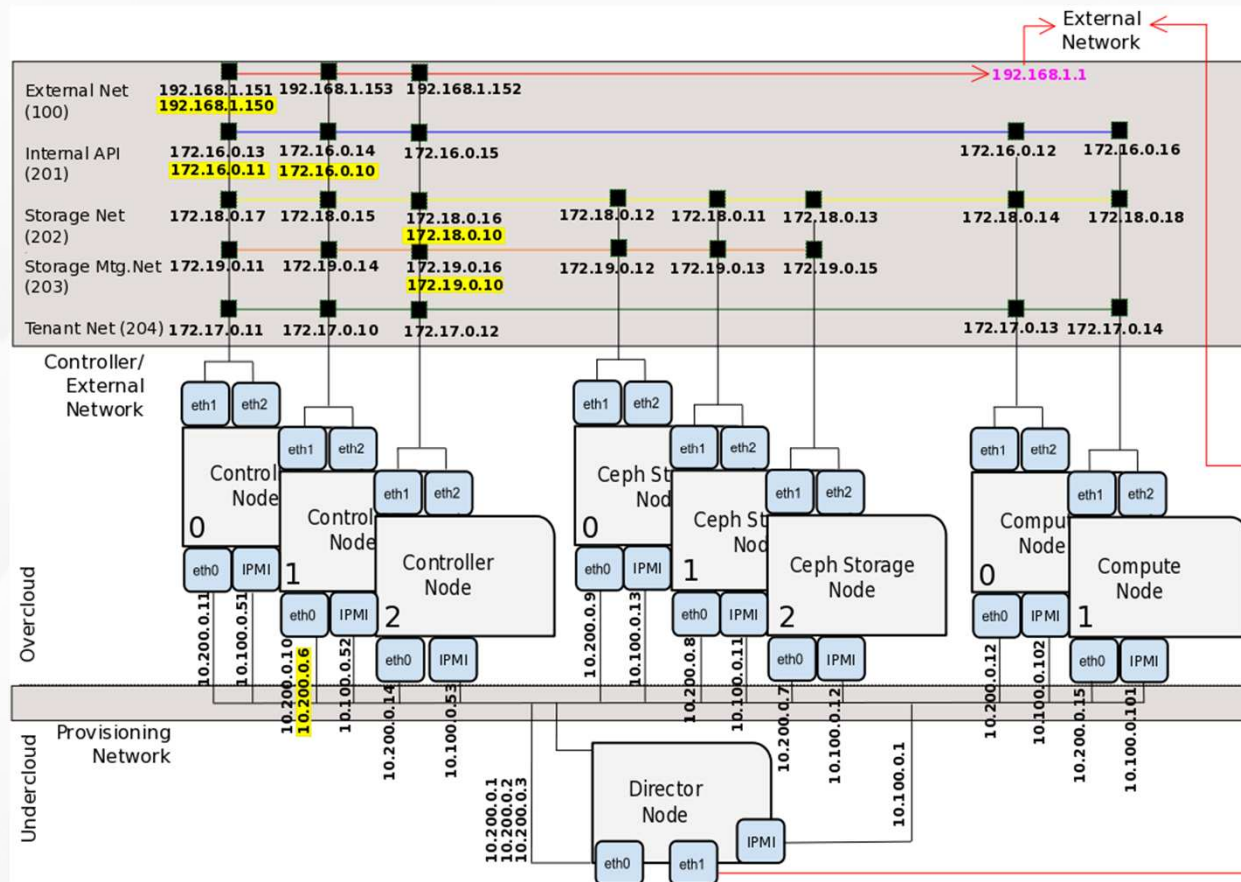
Live Introduction to Linux Foundation Certified OpenStack Administrator (OCA)

Q&A

Thanks You for Your attention!