



# Melting together OpenStack, RHV and OpenShift

Run Mode 1 and Mode 2 Applications side-by-side

---

Joachim von Thadden  
Principal Specialist Solution Architect



**Joachim von Thadden**

**EMEA Principal Specialist Solution Architect OpenStack and RHV**

- based in Germany, near Düsseldorf
- more than 20 years in IT
- more than 25 years working with Linux
- 6 years experience with OpenStack
- 3.5 years at Red Hat

# AGENDA

Motivation

The Stack Layout

Baseline: OpenStack

Mode 1: RHV

Mode 2: OpenShift

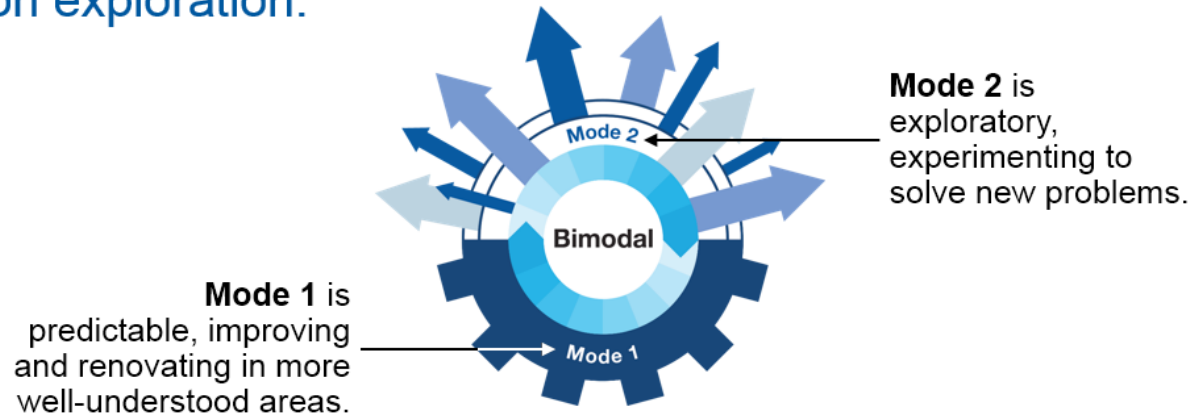
Q&A

# Motivation

# Bi-Modal IT

Gartner's definition

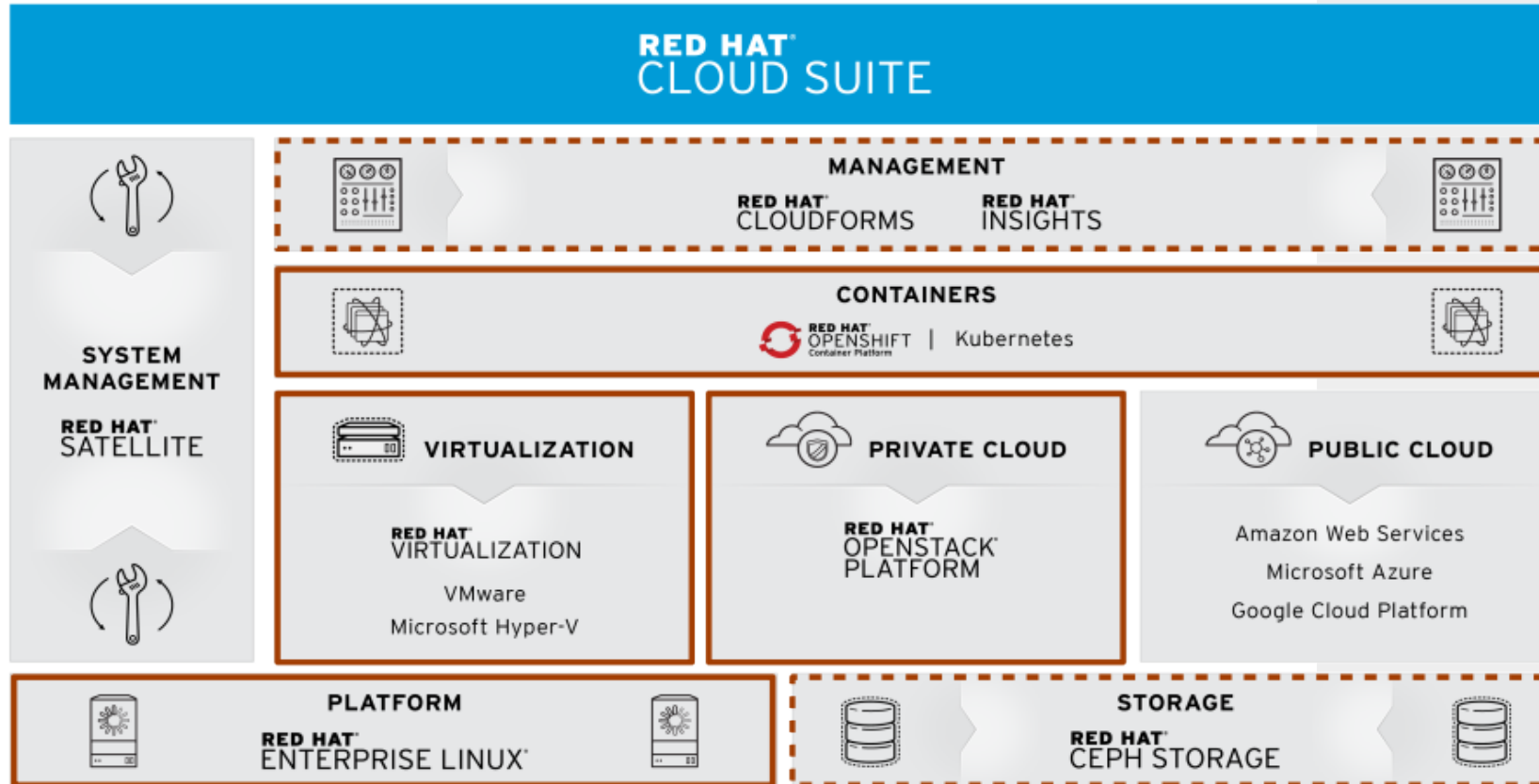
**Bimodal** is the practice of managing two separate but coherent styles of work — one focused on predictability and the other on exploration.



“Bimodal IT is the practice of managing two separate, coherent modes of IT delivery, one focused on stability and the other on agility. Mode 1 is traditional and sequential, emphasizing safety and accuracy. Mode 2 is exploratory and nonlinear, emphasizing agility and speed.”

# The Stack Layout

# The Stack Layout



CI-00/4-01

# Baseline: OpenStack



# Baseline: OpenStack

## *IaaS Mode 2 Virtualizer*



# Baseline: OpenStack

*What to do here?*

# Just install it!

Ask your precious SSA for help if you struggle :-)

# Mode 1: RHV

# Mode 1: RHV Virtualizer

## *Integrating Network (Neutron)*

- Integration is done on the RHV hosts with installing OVN, first.
- Easiest way here is to pre-install the hosts as “Networker Nodes” from RHOSP.
- Then install the hosts as you would do with pre-installed RHEL nodes for RHV:
  - Add required subs
  - Add to RHV-M or install the Hosted Engine
- Add the Neutron Network Provider to RHV like in the following slides.

# Mode 1: RHV Virtualizer

## *Creating the Networker Node*

```
$ # 1. Create the Role
$ openstack overcloud roles generate -o ~/templates/roles_data.yaml --roles-path ~/roles Controller Compute \
  Networker

$ # 2. Create the Flavor
$ openstack flavor create --id auto --ram 6144 --disk 40 --vcpus 4 networker
$ openstack flavor set --property "cpu_arch"="x86_64" --property "capabilities:boot_option"="local" --property
"capabilities:profile"="Networker" networker
$ openstack flavor set --property resources:VCPU=0 --property resources:MEMORY_MB=0 --property resources:DISK_GB=0
--property resources:CUSTOM_BAREMETAL=1 networker

$ # 3. Tag the Baremetal Node to the new Flavor
$ openstack baremetal node set --property capabilities='profile:networker,boot_option:local' \
  58c3d07e-24f2-48a7-bbb6-6843f0e8ee13
```

# Mode 1: RHV Virtualizer

## *Creating the Networker Node*

```
$ # 4. Create the Scaling environment
$ # snippet from ~/templates/node-count-flavor.yaml
parameter_defaults:
  OvercloudNetworkerFlavor: networker
  NetworkerCount: 3

$ # 5. Deploy
$ openstack overcloud deploy --templates -r ~/templates/roles_data.yaml -e ~/templates/node-count-flavor.yaml
```

# Mode 1: RHV Virtualizer

## *Integrating Network (Neutron)*

- Integration is done on the RHV hosts with installing OVN, first.
- Easiest way here is to pre-install the hosts as “Networker Nodes” from RHOSP.
- Then install the hosts as you would do with pre-installed RHEL nodes for RHV:
  - Add required subs
  - Add to RHV-M or install the Hosted Engine
- Add the Neutron Network Provider to RHV like in the following slides.

# Mode 1: RHV Virtualizer

## Integrating Network (Neutron)

https://iac-rhvm.iac.local/ovirt-engine/webadmin/?locale=en\_US#networks 110% Suchen

RED HAT VIRTUALIZATION

Dashboard

Compute >

Network >

Storage >

Administration >

Events

Network > Networks

Network: [x] [☆] [v] [Q]

New Import Edit Remove

1 - 11 < >

Name	Comment	Data Center	Description	Role	VLAN Tag	QoS Nam	Label	Provider	MTU
App_Tier_private_network		Default		✓	-	-	-	OSP-Neutron	1450
DB_Tier_private_network		Default		✓	-	-	-	OSP-Neutron	1450
gluster		Default			2057	-	-		Default (1500)
HA network tenant 0e27ce69b84d4e		Default		✓	-	-	-	OSP-Neutron	1450
HA network tenant 622e4ad7fe2849		Default		✓	-	-	-	OSP-Neutron	1450
internal		Default		✓	-	-	-	OSP-Neutron	1450
lb-mgmt-net		Default		✓	-	-	-	OSP-Neutron	1450
ovirtmgmt		Default	Management Network	✓	-	-	-		Default (1500)
public_flat		Default		✓	-	-	-	OSP-Neutron	1500
test_net		Default		✓	-	-	-	OSP-Neutron	1450
Web_Tier_private_network		Default		✓	-	-	-	OSP-Neutron	1450



# Mode 1: RHV Virtualizer

## Integrating Network (Neutron)

Administration » Providers

Provider:  [x] [☆] [v] [Q]

[↻] [v] 1-3 < >

Name	Type	Description	Provider URL
<a href="#">OSP-Glance</a>	OpenStack Image		https://192.168.70.25:13292
<a href="#">OSP-Neutron</a>	OpenStack Networking		https://192.168.70.25:13696
<a href="#">ovirt-provider-ovn</a>	External Network Provider	oVirt network provider for OVN	https://iac-rhvm.iac.local:9696

Buttons: Add, Edit, Remove

# Mode 1: RHV Virtualizer

## Integrating Network (Neutron)

```
(overcloud) [stack@iac-director ~]$ source overcloudrc
(overcloud) [stack@iac-director ~]$ openstack endpoint list |grep -i 192.168.70.25|grep -i neutron
| f20bd692ebf5473a9c1009e92353faa4 | regionOne | neutron | network | True | public | https://192.168.70.25:13696
```

Edit Provider
✕

**General** >

---

**Agent Configuration**

Name

Description

Type

Networking Plugin

Automatic Synchronization

Provider URL

Read-Only

Requires Authentication

Username

✕

Username

Password

Protocol

Host Name

API Port

API Version

User Domain Name

Project Name

Project Domain Name

✕

**⚠** Automatic deployment of the Neutron provider on host may not work. It is highly recommended to manually deploy the OVS agent on the host.

Interface Mappings

**Messaging Broker Configuration**

Broker Type

Host

Port

Username

Password

# Mode 1: RHV Virtualizer

## Integrating Network (Neutron)

Administration » Providers

Provider:  ✕ ☆ ▼ 🔍 Add Edit Remove

Name	Type	Description	Provider URL
<a href="#">OSP-Glance</a>	OpenStack Image		https://192.168.70.25:13292
<a href="#">OSP-Neutron</a>	OpenStack Networking		https://192.168.70.25:13696
<a href="#">ovirt-provider-ovn</a>	External Network Provider	oVirt network provider for OVN	https://iac-rhvm.iac.local:9696

# Mode 1: RHV Virtualizer

## Integrating Network (Neutron)

The screenshot shows the Red Hat Virtualization webadmin interface. The main content area displays a table of provider networks and a dialog box for importing them.

**Import Networks Dialog:**

- Network Provider:** OSP-Neutron
- Provider Networks:**

Name	Provider Network ID
<input type="checkbox"/> HA network tenant 0e27ce69b84d4e92953a7e5862dab78b	524c61c5-b2a2-48b8-bd81-635e9dbb756e
<input type="checkbox"/> HA network tenant 622e4ad7fe2849d2bda9fd6e64c87bbd	dcc2fec1-5d19-4f6d-9705-e95d8a8150eb
<input type="checkbox"/> internal	8d78926d-e4d9-43ea-bae3-b01317c97e4e
<input type="checkbox"/> lb-mgmt-net	477c1c2c-206b-4699-91e2-b305f40851e1
<input type="checkbox"/> public_flat	8d223107-fd92-417f-a05b-841f3223b3d7
<input type="checkbox"/> test_net	8baece4a-7148-48c3-b42a-9cebc4434634
- Networks to Import:**

Name	Provider Network ID	Data Center	Allow All
<input type="checkbox"/> App_Tier_private_network	81b96d25-2138-4c90-a76f-3bdb8b4...	▼	<input checked="" type="checkbox"/>
<input type="checkbox"/> DB_Tier_private_network	65c3d7d6-8100-404a-98c0-c9a89806...	▼	<input checked="" type="checkbox"/>
<input type="checkbox"/> Web_Tier_private_network	75155e05-e4d7-451b-8125-1130fa34...	▼	<input checked="" type="checkbox"/>

# Mode 1: RHV Virtualizer

## Integrating Network (Neutron)

The image shows two overlapping screenshots of the Red Hat Virtualization (RHV) web interface. The left screenshot displays the 'Networks' page with a table of network configurations. A red circle highlights the 'App\_Tier\_private\_network' entry. The right screenshot shows the 'Edit Virtual Machine' dialog for a VM named 'Fedora30WS'. A red circle highlights the 'nic1' dropdown menu, which is set to 'App\_Tier\_private\_network/App\_Tie'. A red arrow points from the circled network name in the left screenshot to the circled dropdown in the right screenshot.

Name	Comment	Data Center	Description	Role	VM
App_Tier_private_network		Default			-
DB_Tier_private_network		Default			-
gluster		Default			20
HA network tenant 0e27ce69b84d4e		Default			-
HA network tenant 622e4ad7fe2849		Default			-
internal		Default			-
lb-mgmt-net		Default			-
ovirtmgmt		Default	Management Network		-
public_flat		Default			-
test_net		Default			-
Web_Tier_private_network		Default			-

# Mode 1: RHV Virtualizer

## *Integrating Storage (Cinder + Glance)*

- Cinder is Tech Preview as of 4.3
- Glance is Production Ready (in theory)
- **Still only keystone v2**

# Mode 1: RHV Virtualizer

## *Pros and Cons*

### PRO

- **Enhanced Network (and other SDx) Capabilities:**  
Support software defined networking (SDN) and other software defined modules (if keystone v3 is working in RHV) in virtualization along with overlay, encapsulation, and security groups.
- **Integrate Mode 1 and 2:**  
Support applications that require both scale up and scale out technologies simultaneously for different segments of the application.
- **Unified Network Management:**  
Manage the network topology from a single management platform for both the virtualization environment and the OpenStack environment.
- **Efficiency and reduced Complexity:**  
Using a unified Storage access method means better usage of resources (man and machine).

### ● CONTRA

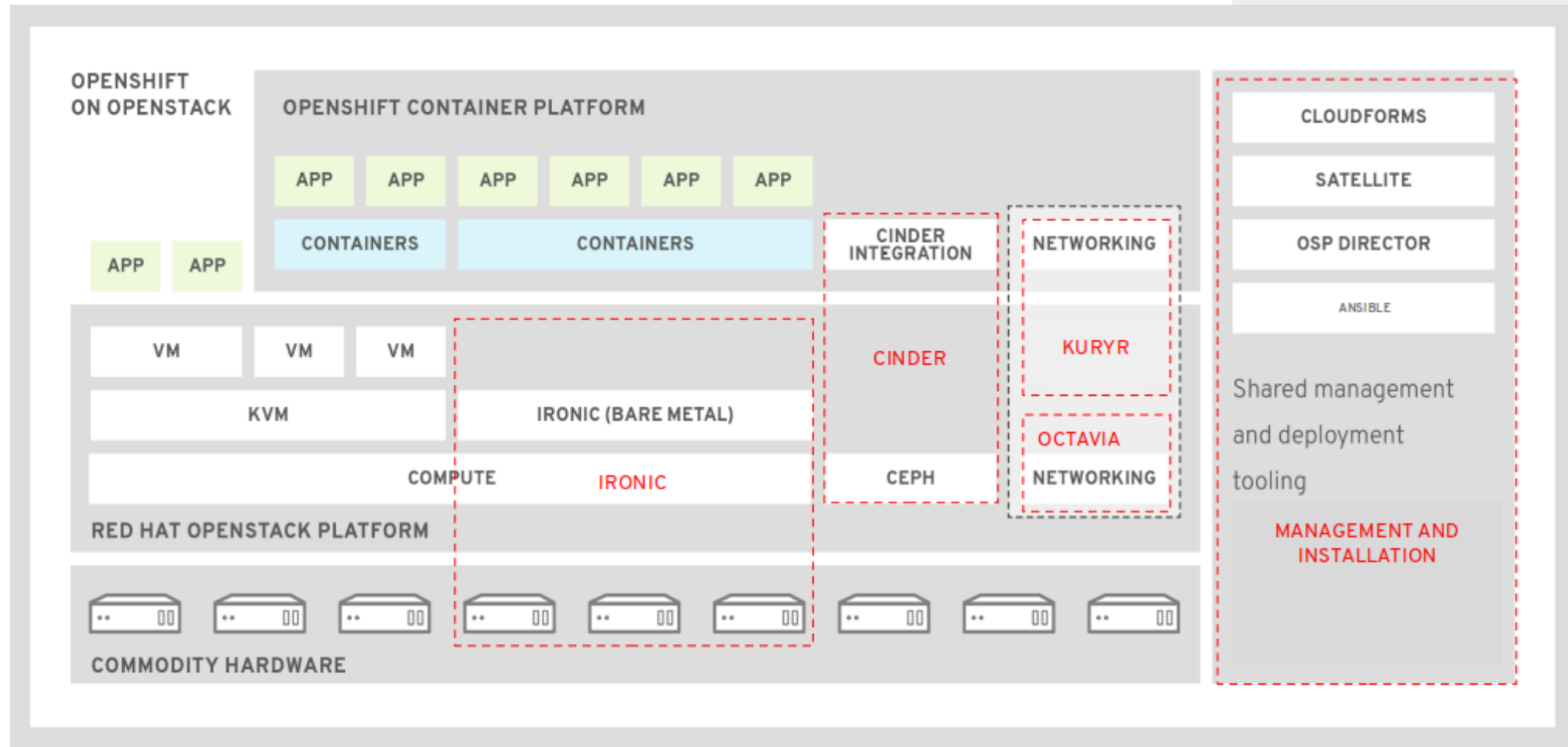
- **api stabilities**  
Apis must remain stable or at least downward compatible, to be able to always upgrade.
- **authentication method stability**  
This is as 1), but mentioned here separately because of latest keystone V2 -> V3 movement and deprecation of the former.
- **SPOF**  
OpenStack is the pivotal point. A major incident here might bring down other, non-osp parts, as well.  
To mitigate, at least the following is needed:
  - clean SOPs
  - fully automated and audited installation procedures
  - fully automated and audited operation procedures
  - Monitoring
  - DR procedures
  - **more than one Network Provider!**

# Mode 2: OpenShift



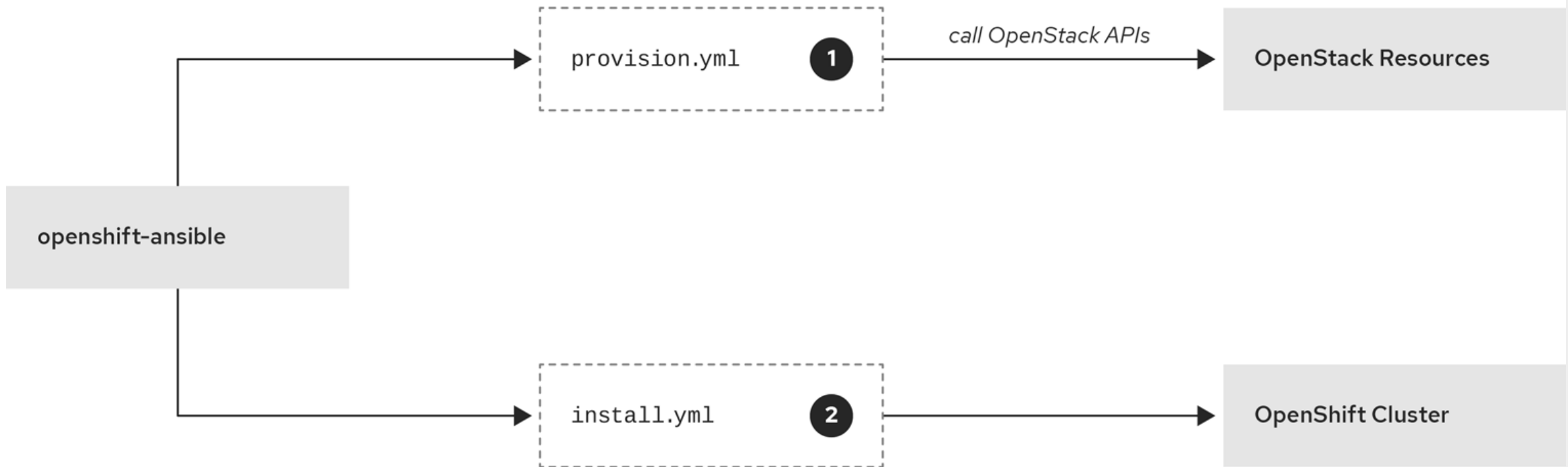
# Mode 2: OpenShift

## Integration Parts



# Mode 2: OpenShift

*General Installation Procedure*



OpenShift\_31\_0619

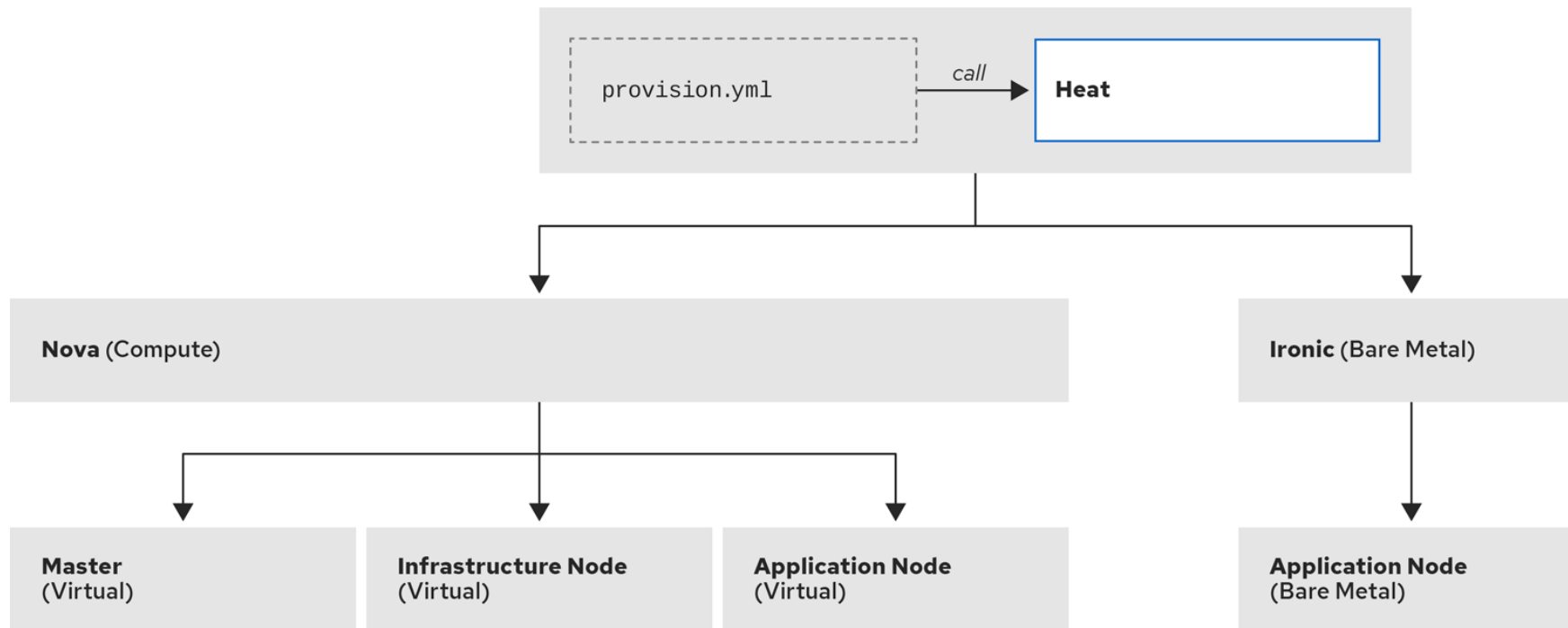
# Mode 2: OpenShift

## *Sequence of Integration*

- openshift-ansible is the driver
  - creating the infrastructure on OpenStack
    - using Nova to create VMS for running various OpenShift roles based on predefined flavors and images)
    - Ironic can also be used to push operating system images to bare metal servers
  - configuring the container runtime environment on virtual machines
  - provisioning storage for an internal registry
  - configuring the OpenShift SDN
  - connecting to authentication systems

# Mode 2: OpenShift

## *Integrating Compute (Nova)*



OpenShift\_31\_0619

# Mode 2: OpenShift

## *Advantages*

- easily scaleable (as VM **and** BM) using the same facilities as OpenStack

# Mode 2: OpenShift

## *Advantages*

- easily scaleable (as VM **and** BM) using the same facilities as OpenStack
- Administrators do not have to learn Heat or any other OpenStack tools to deploy OpenShift on OpenStack

## Mode 2: OpenShift

### *Advantages*

- easily scaleable (as VM **and** BM) using the same facilities as OpenStack
- Administrators do not have to learn Heat or any other OpenStack tools to deploy OpenShift on OpenStack
- OpenStack administrators familiar with Heat can use the tools they are already familiar with to examine and manage the deployed stack

# Mode 2: OpenShift

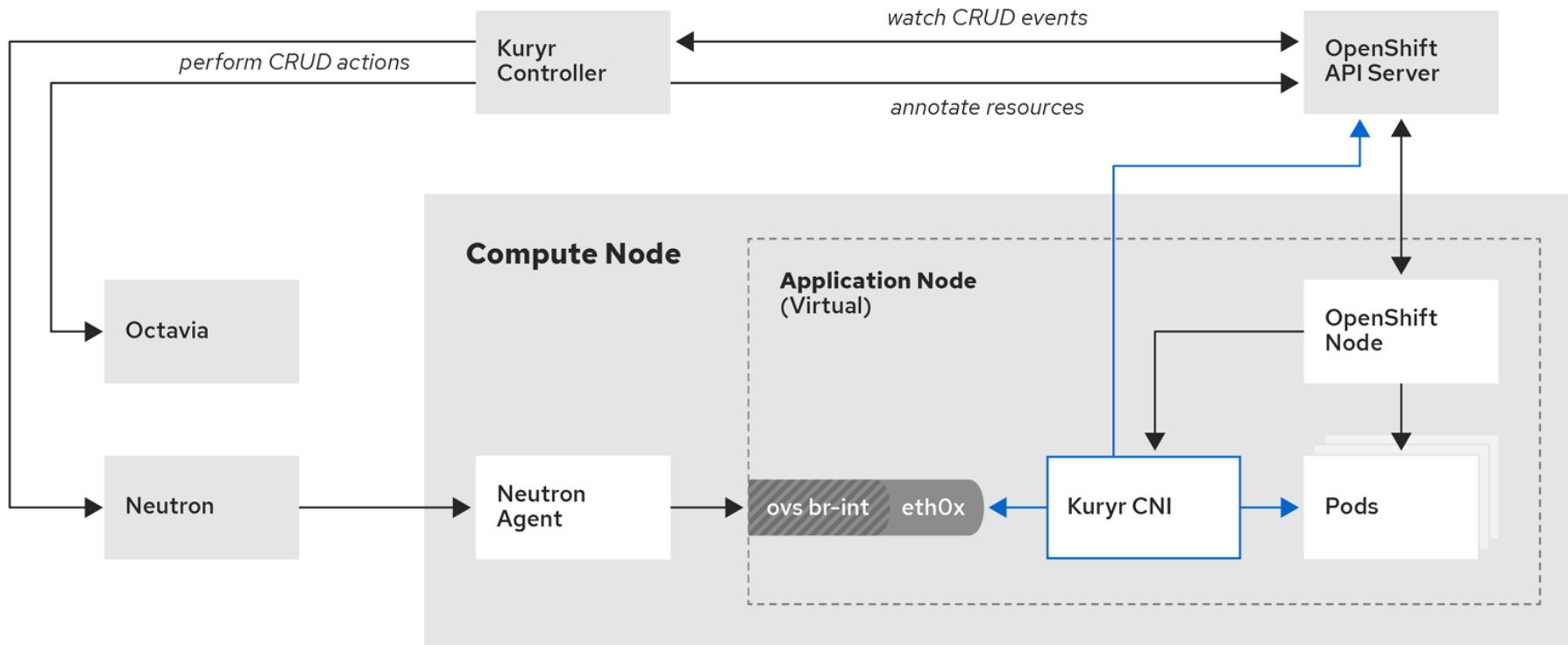
## *Advantages*

- easily scaleable (as VM **and** BM) using the same facilities as OpenStack
- Administrators do not have to learn Heat or any other OpenStack tools to deploy OpenShift on OpenStack
- OpenStack administrators familiar with Heat can use the tools they are already familiar with to examine and manage the deployed stack
- reliable interface for automating OpenShift installations



# Mode 2: OpenShift

*Integrating the OpenStack network via KURYR*



OpenShift\_31\_0619

# Mode 2: OpenShift

## *Integrating Network via Kuryr (Neutron)*

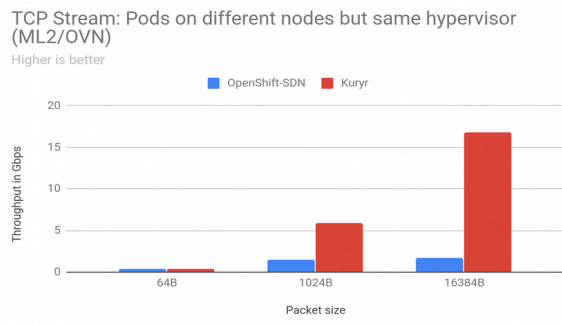
- Installed by openshift-ansible
- Kuryr is a CNI plugin that uses OpenStack Neutron and Octavia to provide networking for pods and services
  - *deployed as pods in the kuryr namespace*
  - *kuryr-cni container installs kuryr CNI driver on OpenShift masters, infrastructure and compute nodes as a daemonset*
  - *kuryr controller maps OCP API CRUD events to Neutron and Octavia objects.*
  - *kuryr avoids double encapsulation of OCP on OSP*
  - *only use when not using provider networks or third party SDN integrations*

# Mode 2: OpenShift

## *Pros and Cons*

### PRO

- **Kuryr with Neutron is faster than OCP SDN:**



- **Integrate VMs and Container workloads**

- Seamless integration
- no double network encapsulation
- no usage of network breakouts.

### CONTRA

- **Octavia Loadbalancer is created for each pod in use**
  - Duplication of resources
  - we are working on this :-)

Questions? Remarks?

# Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



[twitter.com/RedHat](https://twitter.com/RedHat)