

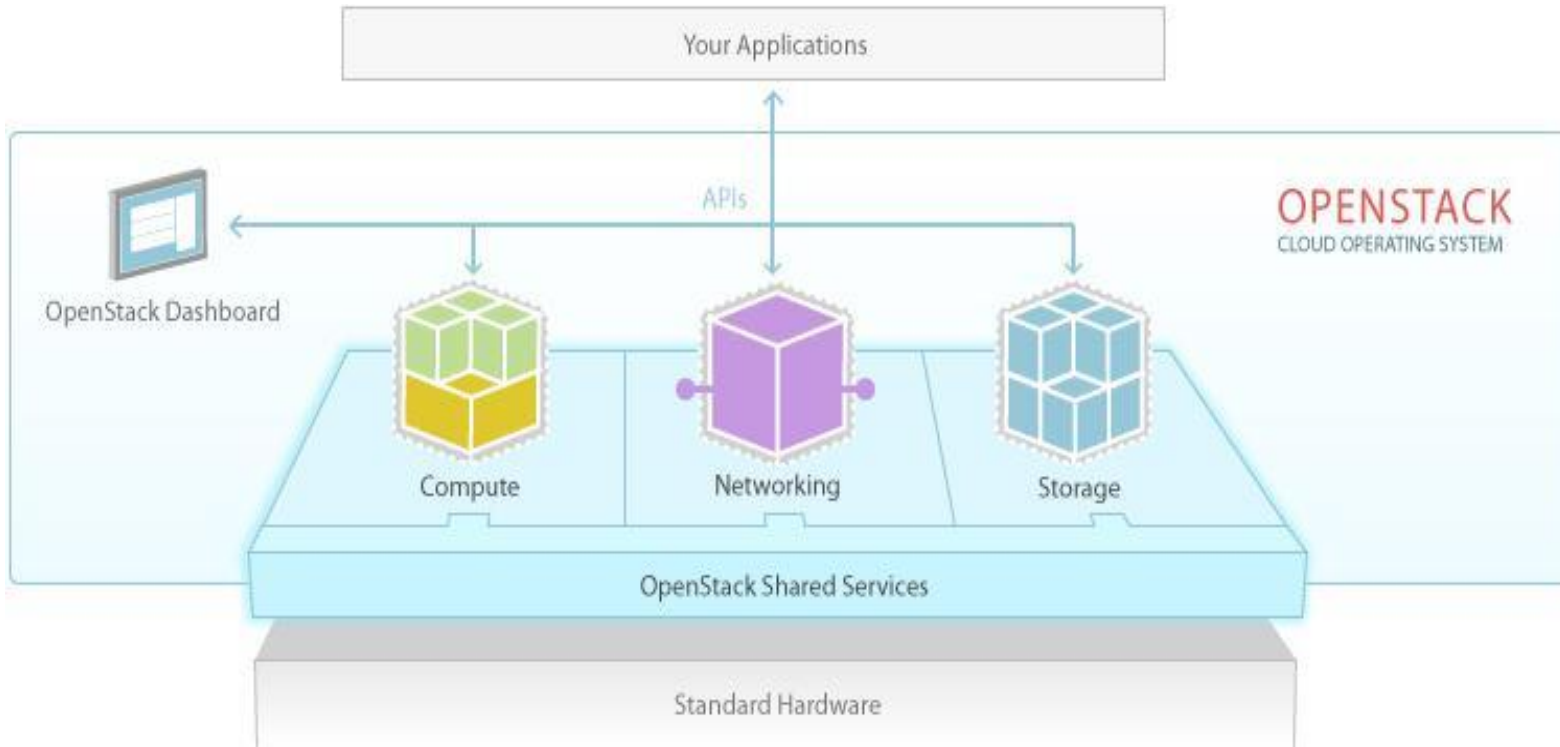
# **OPEN STACK OVERVIEW**



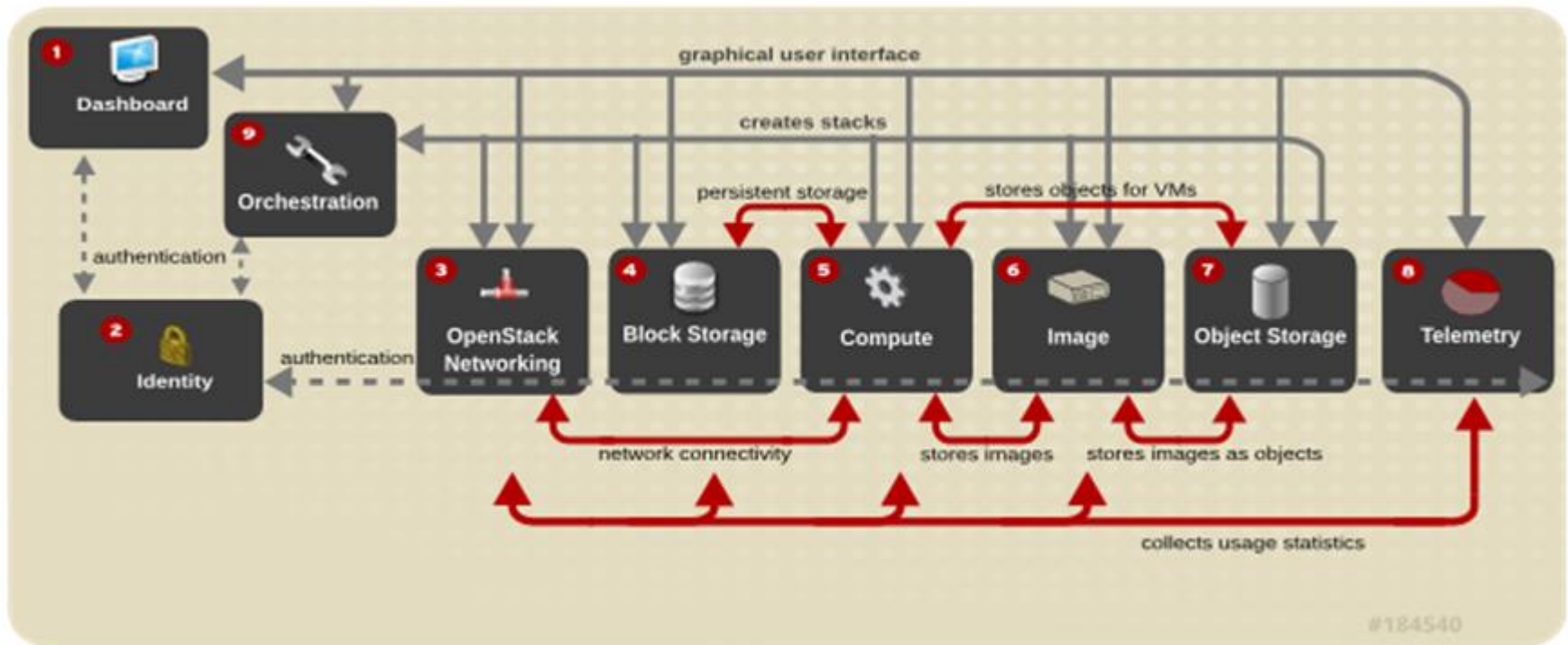
## AGENDA

- Cloud Operating System providing IaaS (aka “Infrastructure as a Service”)
- Allows users to build public or private clouds that scale massively on commodity hardware
- Founded by Rackspace and NASA in 2010
- Managed by OpenStack Software Foundation
- Platinum Sponsors: Canonical, IBM, Nebula, Red Hat, AT&T, Hewlett Packard, Rackspace and Suse
- Open Design, Open Source, Open Community
- Written in Python


## CONTEXT



## SERVICES (CODENAMES)



# NEUTRON NETWORKING TOPOLOGY



**openstack**  
DASHBOARD

Project Admin

CURRENT PROJECT  
**demo**

**Manage Compute**

- Overview
- Instances
- Volumes
- Images & Snapshots
- Access & Security

**Manage Network**

- Network Topology
- Networks
- Routers

**Object Store**

- Containers

**Orchestration**

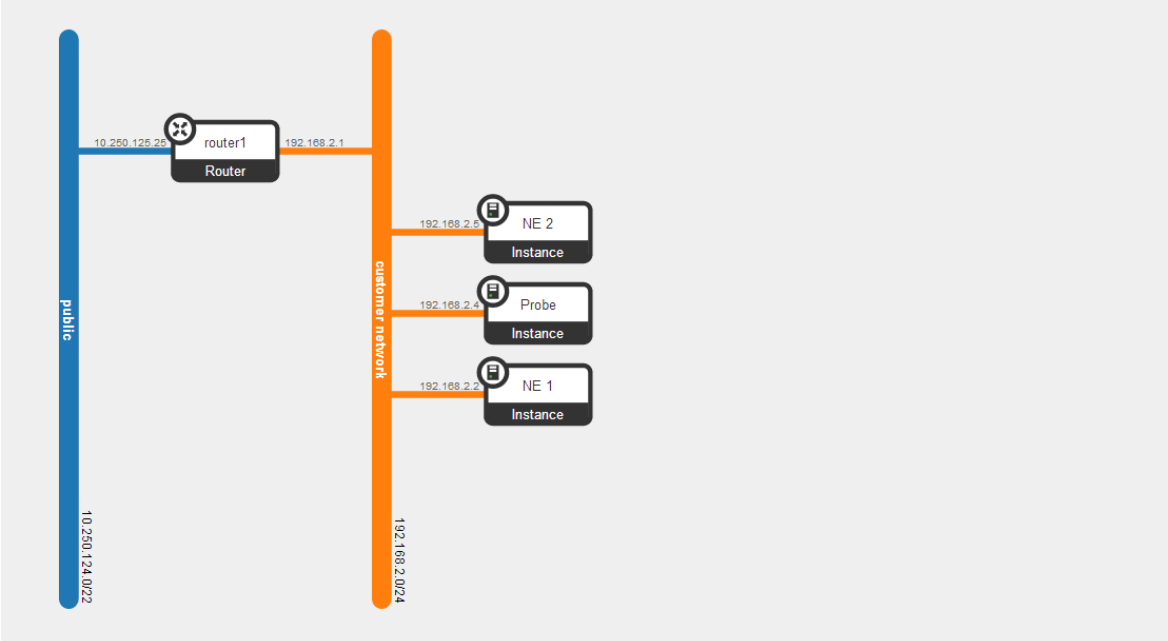
- Stacks

## Network Topology

Logged in as: demo
[Settings](#)
[Help](#)
[Sign Out](#)

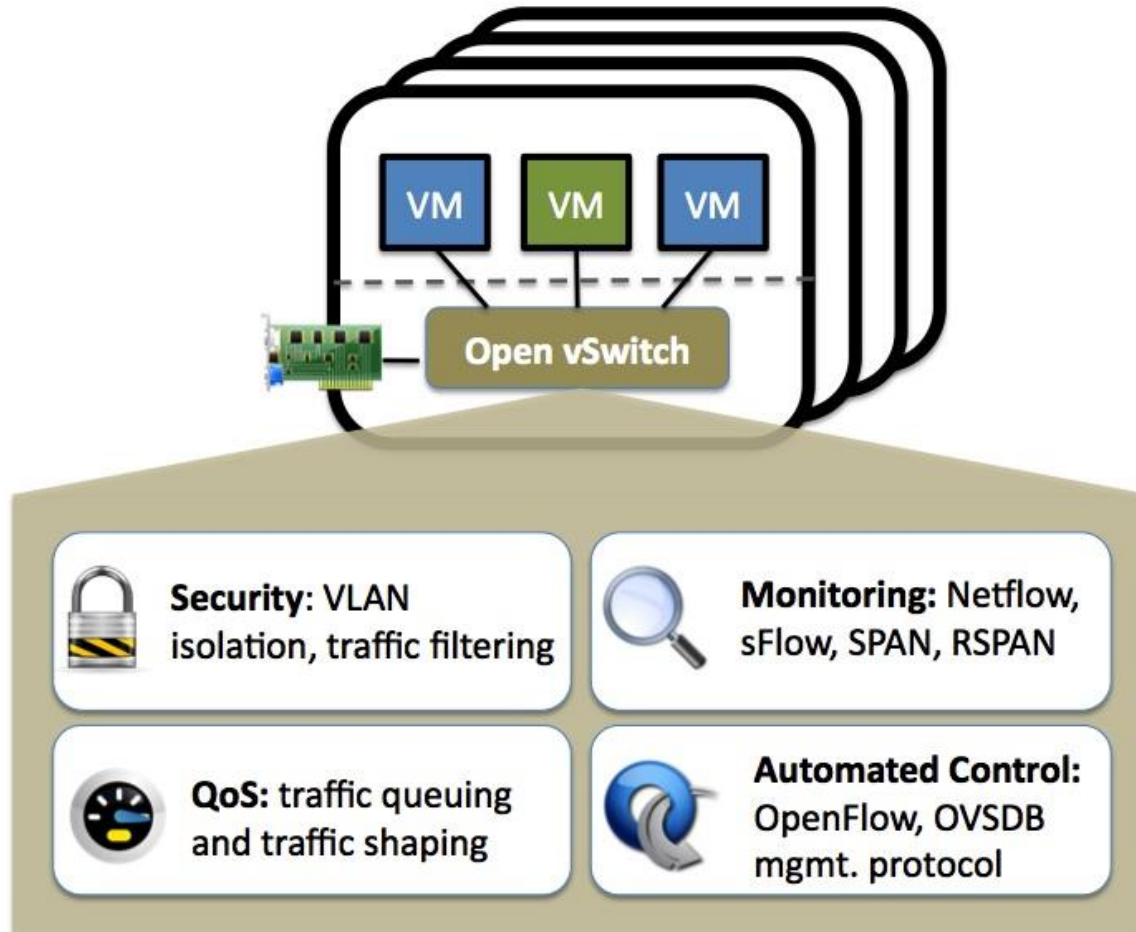
Small Normal

Launch Instance
Create Network
Create Router



The diagram illustrates a network topology with two main vertical bars representing networks. On the left is a blue bar labeled 'public' with IP address '10.250.124.0/22'. On the right is an orange bar labeled 'customer network' with IP address '192.168.2.0/24'. A central router labeled 'router1' connects these two networks. The router has an IP of '192.168.2.1' on the customer network side and '10.250.125.25' on the public network side. Three instances are connected to the customer network: 'NE 2 Instance' (IP 192.168.2.6), 'Probe Instance' (IP 192.168.2.4), and 'NE 1 Instance' (IP 192.168.2.2).

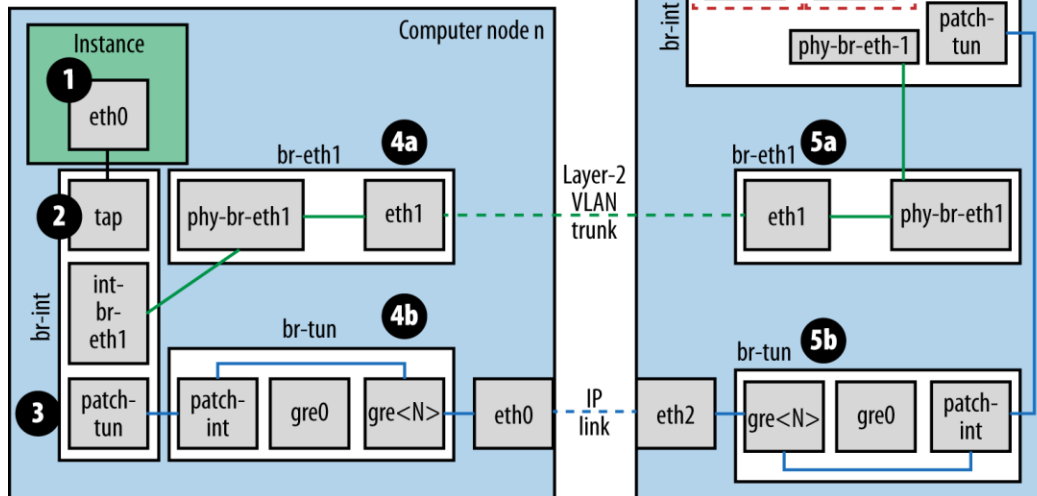
# OPEN VSWITCH



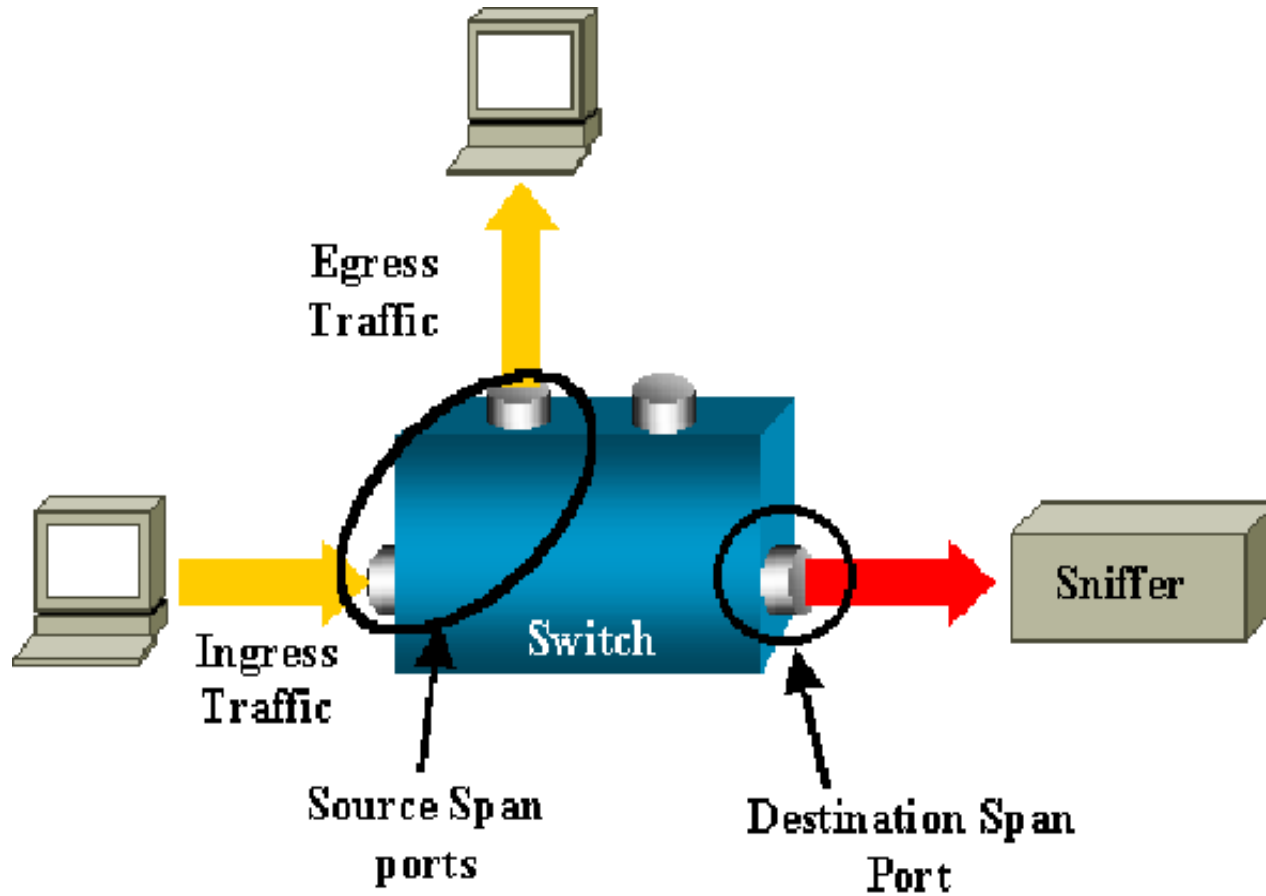
# NEUTRON NETWORKING PATHS

Neutron network paths

- VLAN networks
- GRE networks
- VLAN and GRE networks



# PORT MIRRORING





# **OPENSTACK HEAT TEMPLATES**

## **THE OPENSTACK HEAT MODULE**

- Heat Engine
- Optional
- Deployment
- Orchestration

## DEPLOY CONTROL ELEMENTS

- Scaling
- Counters
- Limits
- High Availability
- Heat can help you with these

## A MANUAL DEPLOYMENT ON OPENSTACK

- Login to the web UI
- Create your things: servers, networks, subnets, routers, gateways, NICs, drives, ssh keys, firewall rules and ips.
- Attach all that together
- Lots of clicks
- Don't make errors
- Diagnostic may be hard
- Repeat

## **LOTS OF STACKS**

- Many people
- Many versions
- Many configurations
- You want automation

## AUTOMATING DEPLOYMENTS

### Programmatic APIs (CRUD)

- Shell Script
- Python
- REST
- Java

### Declarative

- Heat Templates

## WHAT ARE HEAT TEMPLATES ?

- Text Files
- YAML markup
- Describe virtual elements
- See basic template example here:

[http://docs.openstack.org/developer/heat/template\\_guide/hot\\_guide.html](http://docs.openstack.org/developer/heat/template_guide/hot_guide.html)

## THE WORDING

- A template is a stack description
- One template can launch many stacks



## WHAT'S A STACK?

- Deployment unit
- Groups virtual elements
- Creatable
- Deletable
- Entity

## **STACK EXAMPLES**

- A virtual data-center
- A virtualized website and its database

## **MY FIRST ASSIGNMENT WITH HEAT**

- Hardware affinity
- Instance affinity
- Startup ordering
- Multi-NIC servers

## **ALSO COVERED BY HEAT**

- Firewall rules
- Automatic instance creation
- Usage counters
- Block stores as a service
- Relational databases as a service
- Image Configuration
- Does not implement, it exposes and delegates

## ADVANTAGES OF HEAT

- Dependencies (During Creation and Deletion)
- Parallelization
- Validation
- Provides a CRUD UI
- Virtual elements retrieval
- Incremental Updates
- Already Coded

## **ADVANTAGES OF PROGRAMMING**

- Lots of logic
- Lots of dynamic aspects

## BE HYBRID

Use custom programming **and** Heat templates

## **WHAT INSPIRED HEAT?**

- Cloud Formation
- Amazon Web Services
- Same Purpose
- Shared Terminology
- Partial Compatibility