
TOMORROW starts here.
cisco
Cisco live!

Introduction to OpenStack on Cisco

BRKDCT-1367

Robert Starmer
Principal Cloud Architect
Office of the Cloud CTO

Cisco live!

Agenda

- Market Direction – Applications Shifting Data Centre Operations
- DevOps – Dynamic Development and Operations
- OpenStack – Middleware for IaaS/CaaS
- Network Services – Scalable Apps
- Case Study – Enterprise Consumer




Market Direction

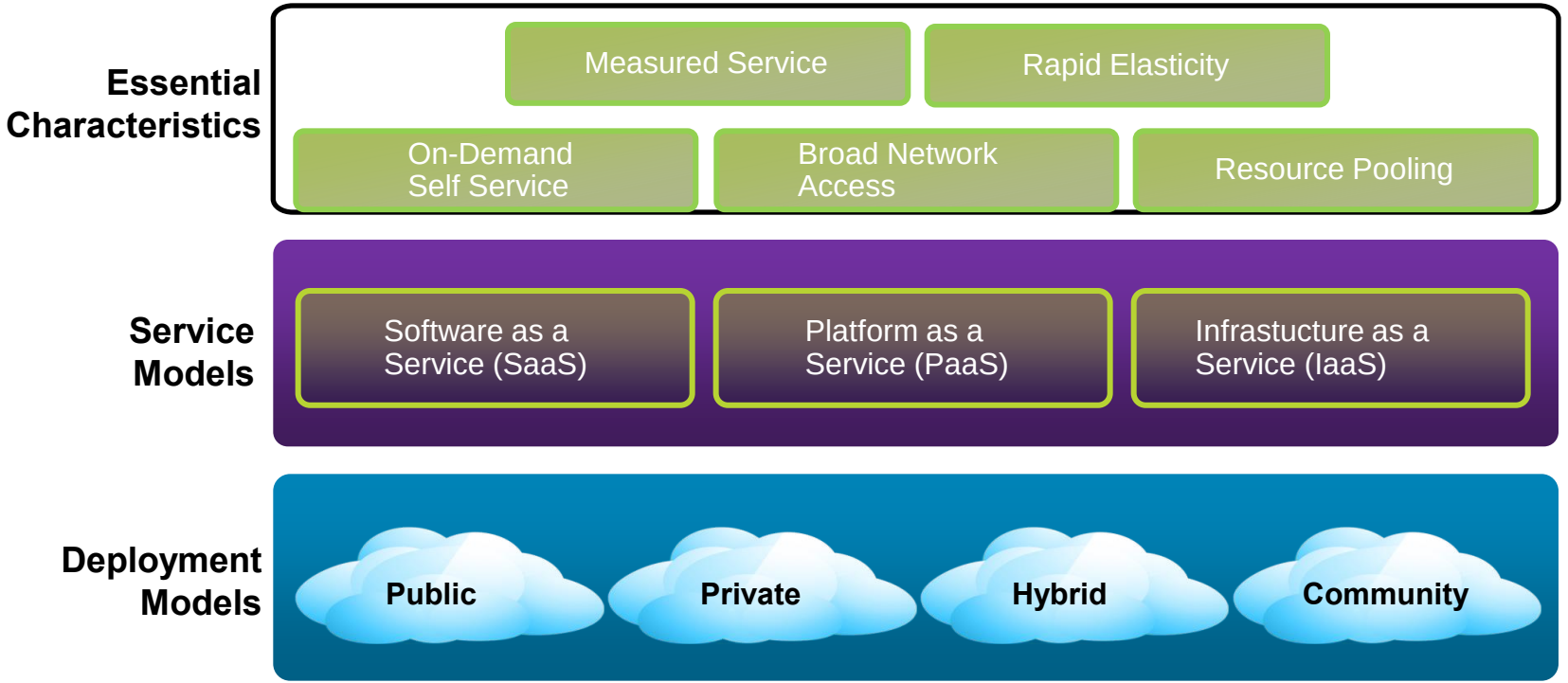

Nebulous


Ethereal

Foggy

Cloud Defined

IT resources and services that are abstracted from the underlying infrastructure provided “on-demand” and “at scale” in a multi-tenant environment



Application History

Monolithic (1940-1980)
- Mega scale “single” compute needed to deal with app scale
- Development as waterfall process, major projects for development, upgrades, etc.
- Security and scale through dedicated external appliances (firewall, SAN storage)

* Classic ERP “System” needed a single monolithic server

Distributed (1980-2010)
- Application broken into scalable units
- Development still waterfall, with complex management and staging required for component upgrades
- Security and scale rely more heavily on network, QoS, embedded services
- Internet bubble web sites, still on monolithic servers with separate databases

WebScale (2010-Present)
- Applications broken into smaller easy to replace units with managed APIs for interaction
- “Agile” development with Continuous Integration (embedded test) and Continuous Deployment (embedded operations)
- Distributed security, either in the app, or at the network edge
- Current super-scale web services and rapidly developed super-custom apps

Data Centre History

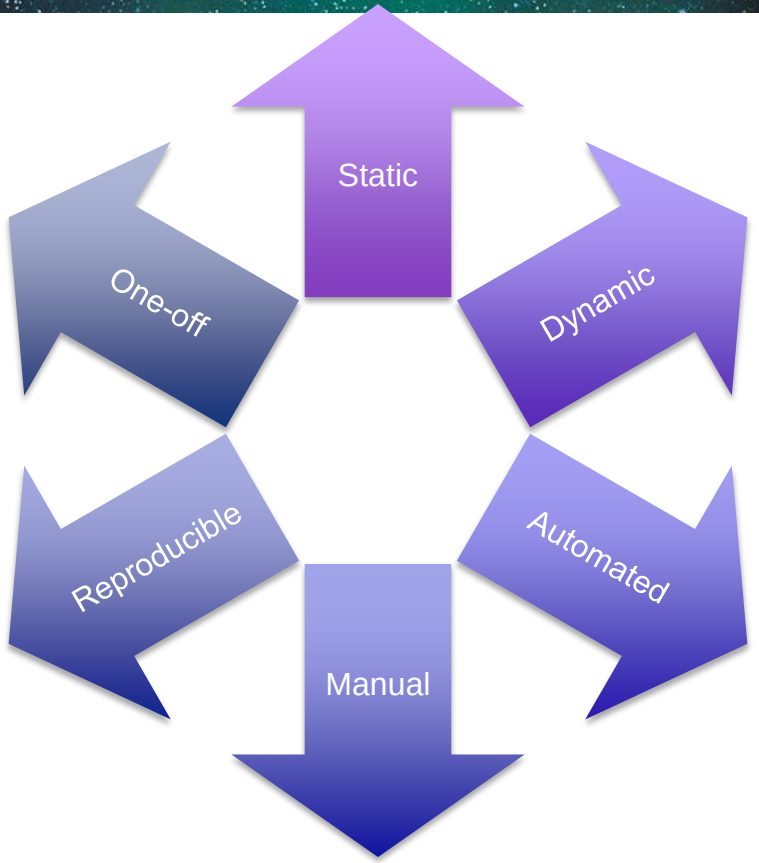
Monolithic (1940-1980)
- Era of the “grey beard” glass walled data centre
- Change is dangerous, change control is the norm
- No “networks” just access services (terminal, dialup terminal, etc.)
- Security == Password control

Distributed (1980-2010)
- Networking enables application change, both in the data centre and in the access paradigm
- Distributed apps, but manual operations, change control still critical to all operations
- Network access drives a culture of security via change control
- Domain expertise leads to silos of effort and understanding

WebScale (2010-Present)
- Network not just for inter-data centre and access, but now applications may be split between sites, or even enterprises
- Agile development leads to a revolt in development organisations against rigid and slow change control
- Security still important (perhaps more so!), but more clearly segmented into well know patterns, allowing for automation of many previously manual processes

Future

- Nothing is changing overnight
- Apps Lead a shift from static, slow-paced development, to dynamic rapidly evolving applications
- Public API driven infrastructure, platforms, and remote or local services enable application scale
- Continuous Integration
- Continuous Deployment



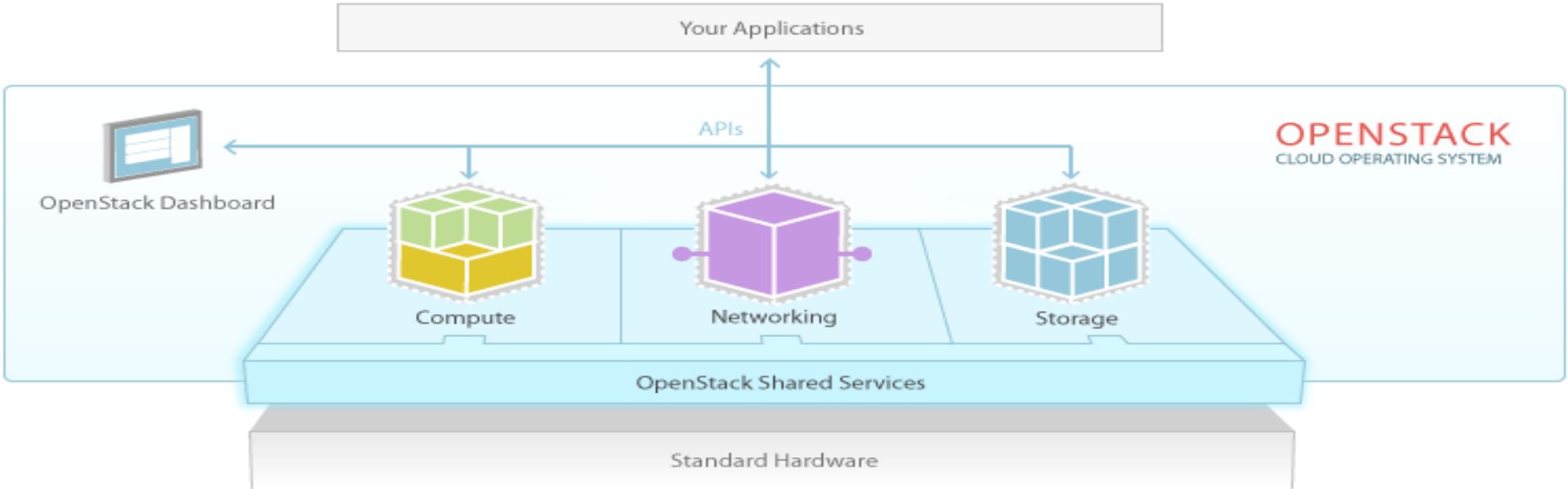



OpenStack

Welcome to OpenStack

The Cloud needs an Open Source platform to achieve Internet Scale:

OpenStack

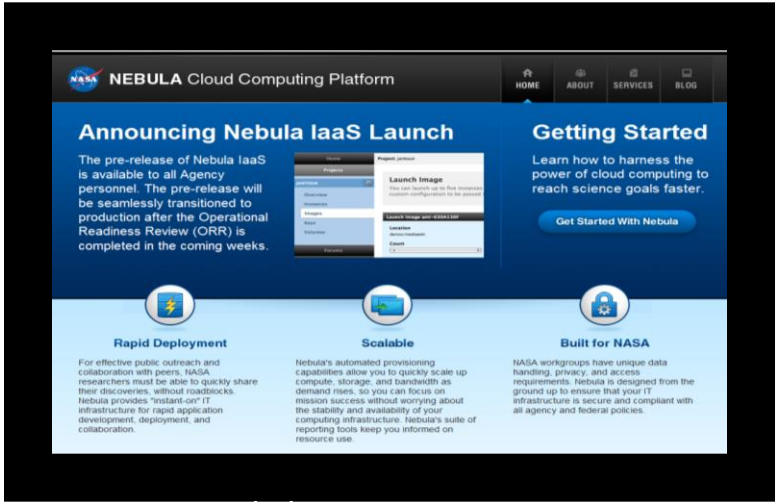


OpenStack Community

160 and counting

OpenStack: A Brief History

- **NASA Launches Nebula**
  - One of the first cloud computing platforms built by the Federal Government for the Federal Government
- **March 2010:** Rackspace Open Sources Cloud Files software, aka Swift
- **May 2010:** NASA open sources compute software, aka “Nova”
- **June 2010:** OpenStack is formed
- **July 2010:** The inaugural Design Summit



OpenStack Introduction

- A Cloud Platform
  - A collection of interrelated software components delivering capabilities to build and manage cloud infrastructure.
- A global community of developers devoted to innovation and openness
- Flexibility in deployment and features
- Standards for broad deployment
- No fear of vendor “lock-in”

OpenStack Terminology

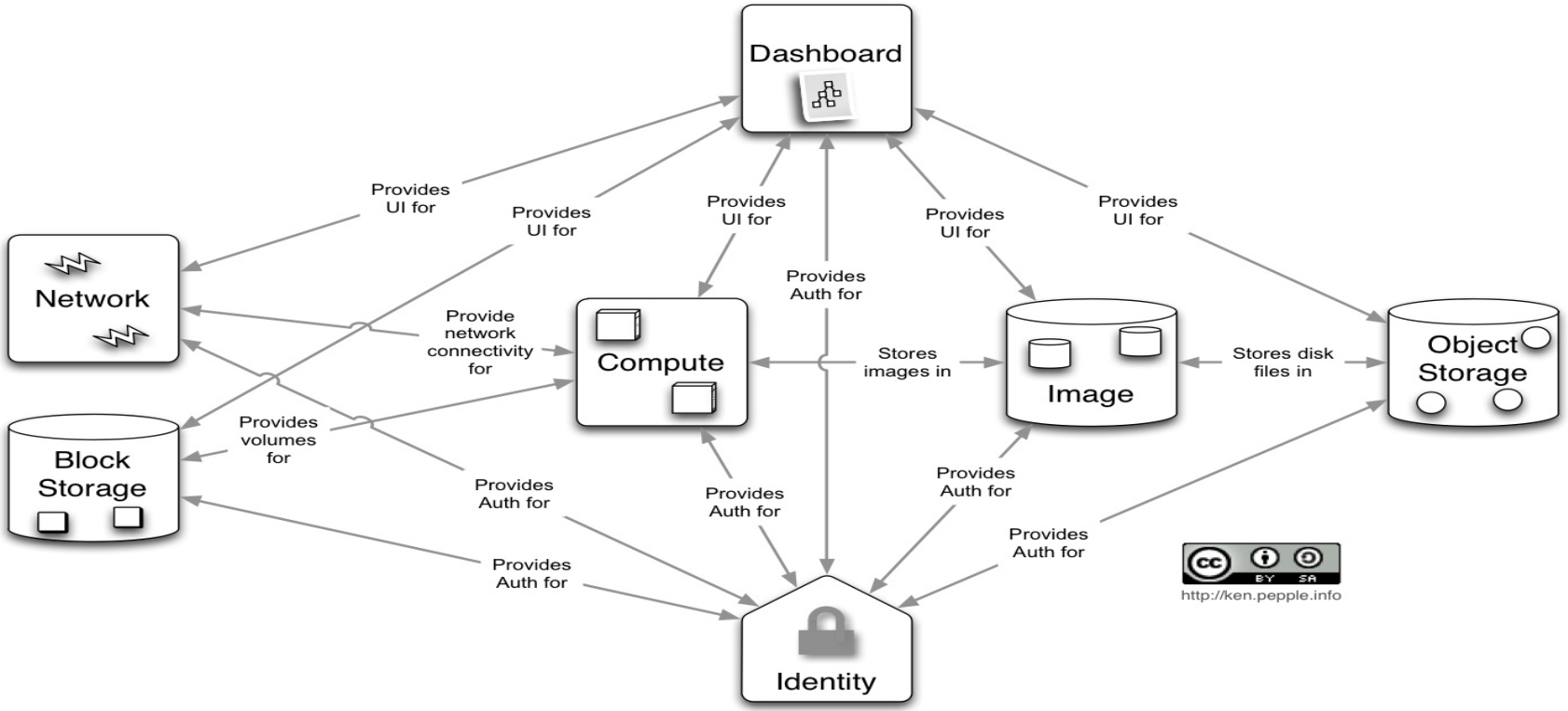
- Instance- Running virtual machine
- Image- Non-running virtual machine, multiple formats (AMI, OVF, etc.)
- Application Programming Interface (API)- Interface for computer programs
- Message Queue- Acts as a hub for passing messages between daemons
- Volume- Provides persistent block storage to instances
- Project- aka Tenants, provides logical separation among cloud users
- Flavors- Pre-created bundles of compute resources
- Fixed IP- Associated to an instance on start-up, internal only
- Floating IP- Public facing IP address

OpenStack Services

- Core Services
  - Compute: Nova
  - Network: Neutron
  - Storage: Cinder and Swift
- Additional Components
  - Authentication: Keystone
  - Image Management: Glance
  - Monitoring: Telemetry
  - Orchestration: HEAT

- Additional Incubated Projects
  - Network Services
    - LBaaS: Neutron
    - FWaaS: Neutron
    - VPNaaS: Neutron
    - DNSaaS: Designate
  - Application Services
    - DBaaS: Trove
    - PaaS: Solum
    - BDaaS: Savanna
    - MQaaS: Marconi

OpenStack Pieces, Interaction



http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/




Case Study

User Models

- Who can best use OpenStack?

- Need “rapid”, “automated”, “dynamic” workload provisioning, preferably API driven
- Applications most likely to leverage a “Web2.0” deployment model
- Understanding of the need for development resources as a part of the Cloud Infrastructure team
- Development teams already using Amazon or equivalent, and expect rapid automated response to change requests

Mid-Sized Enterprise Cloud Deployment

Start small, and grow

Deploy with Puppet (or similar tooling)

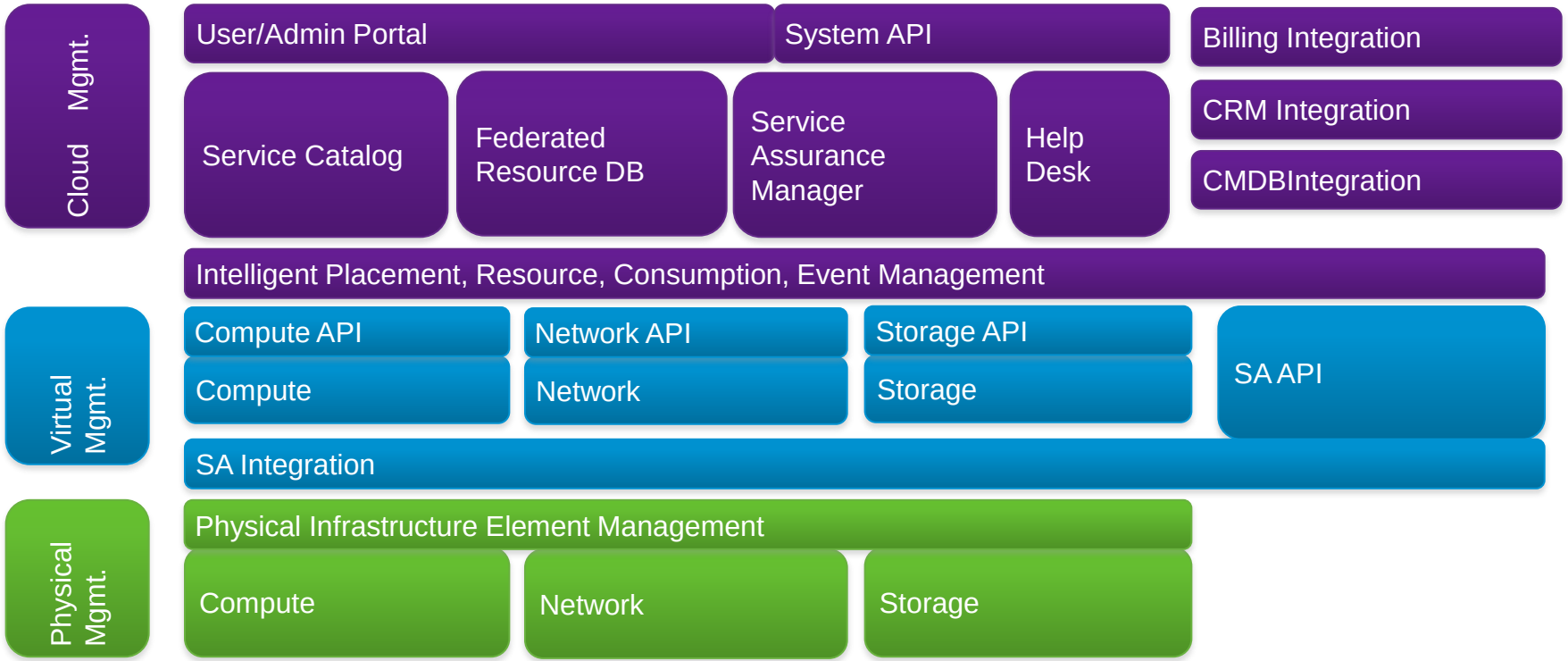
Capture Service knowledge with Nagios/Telemetry

Deploy applications with HEAT

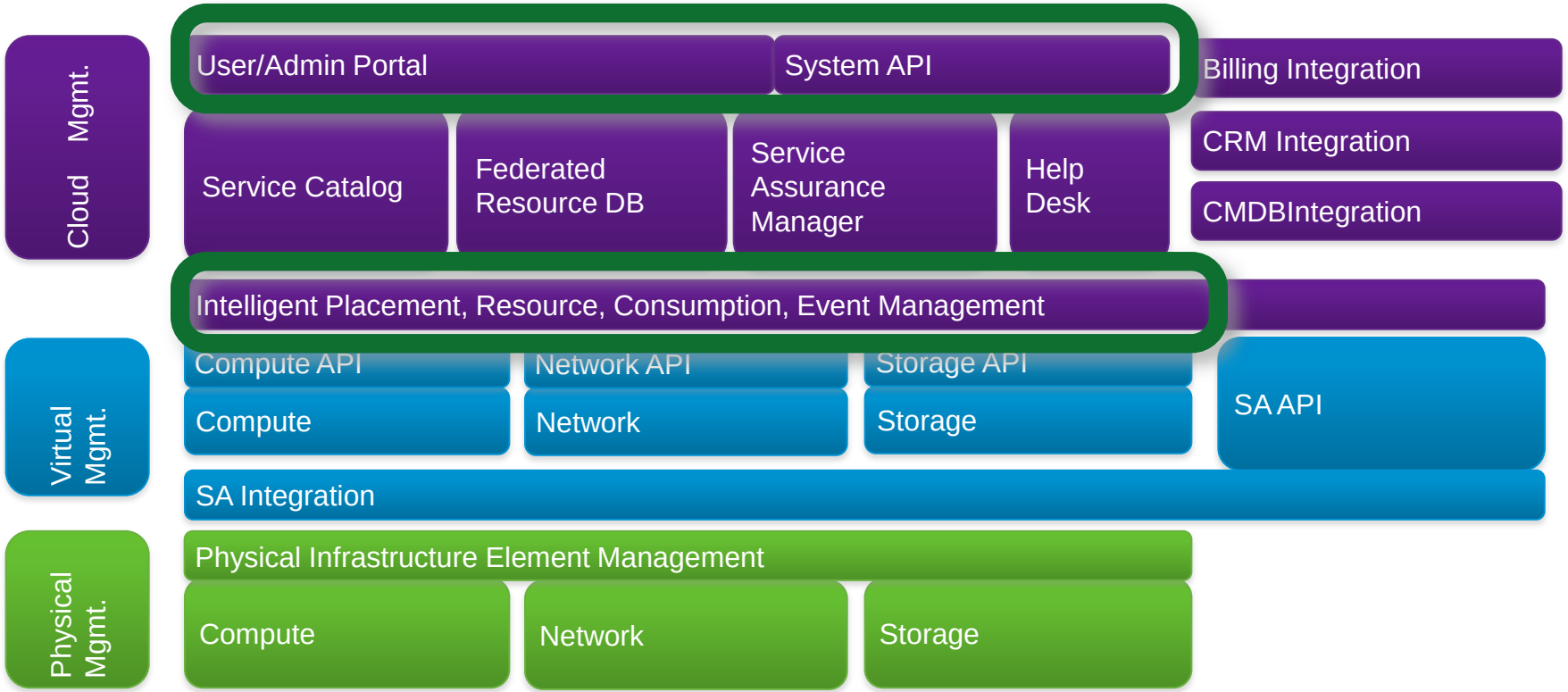
Application may use CI/CD workflow, Web based, Linux platform

Application is targeting internal users/customers, not looking to provide cloud services

Cloud Orchestration Stack Overview



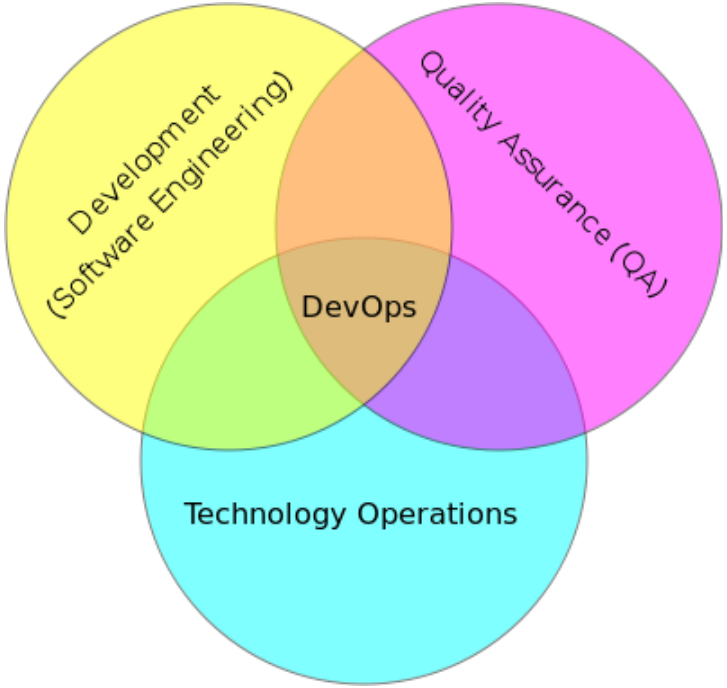
OpenStack Stack Touchpoints



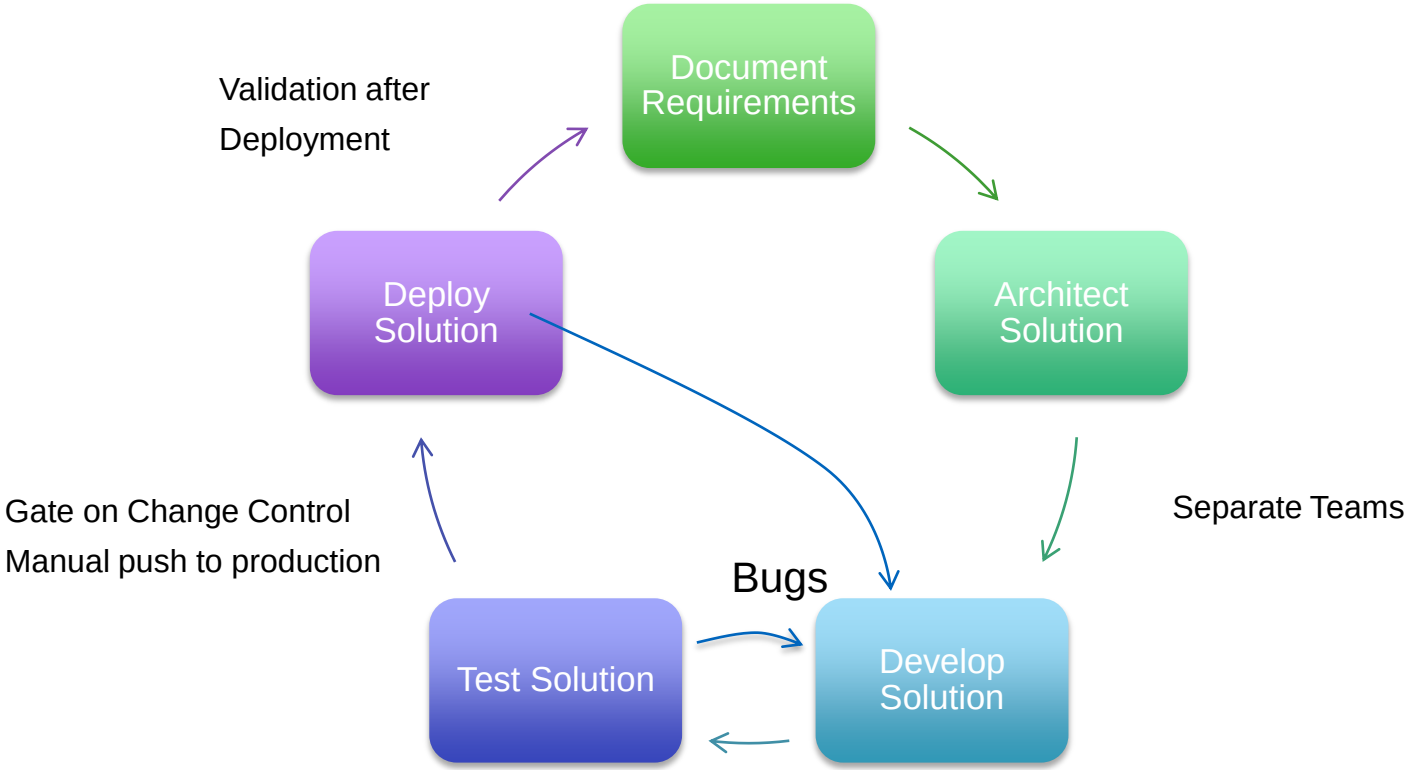
DevOps for Openstack Deployment

DevOps – Development, Deployment, and Operations

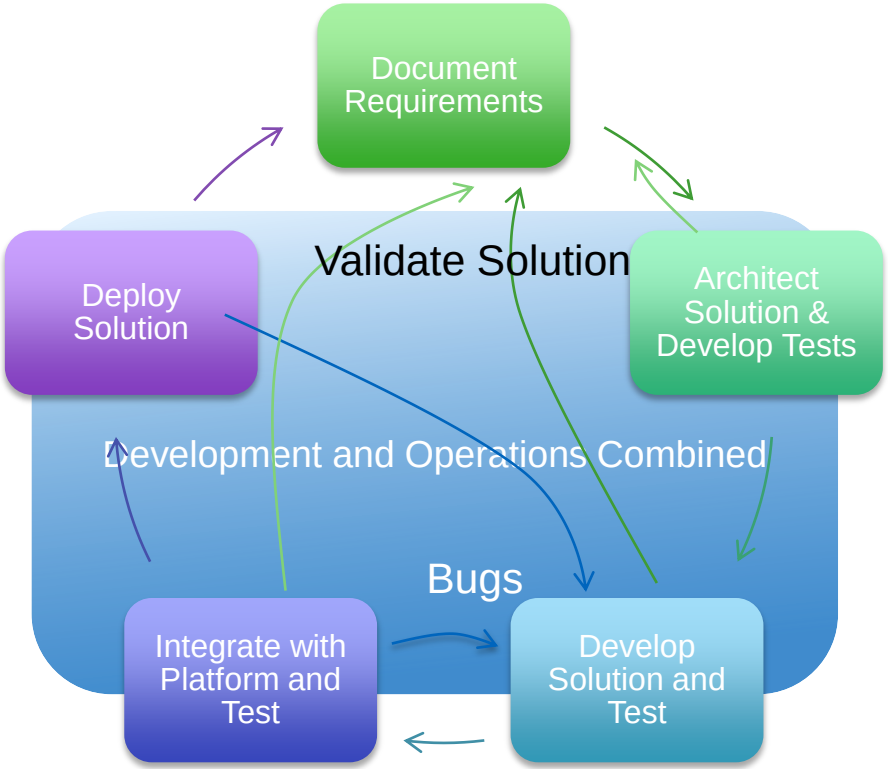
- Agile/Extreme/Lean/Etc. application development expect rapid turn from development->test->production
- Model for Deployment built into the development/test lifecycle
  - Unit test
  - Continuous Integration
- Move from semi-annual release to daily or weekly releases
- Some iterate ~40x/day dev->production!



A Change to Application Development



A Change to Application Development



Deployment:

Push based on successful test

GIT

- A “modern” Source Code Management (SCM) system
- Uses a pointer paradigm rather than patch model
- Similar but different to RCS/SCCS/CVS/SVN
- Biggest end user difference is that branches and merges become easier
- Driven by the need for concurrent development of very large projects (Linux Kernel dev community)

Why Git?

- OpenStack community selected Git as the repository of record for source, and specifically github.com (not the owner of Git, just a _VERY_ heavy user)
- Git integrates well with many of the development workflows and processes used in Agile/Extreme/Lean class development processes
- Git works well with large distributed teams working on the same codebase
- Try it you'll like it
- More info: http://git-scm.com , http://git-scm.com/book

Simple Puppet

- One of the new breed of “DevOps” tools – Data drive Operations
- Others include Chef, Ansible, JuJu, etc.

- Development driven operationalisation of systems
- Developers write the app
- Developers write the test
- Developers write the deployment
- Developers write the upgrade

- Developers wrote the operational model

Configuration Interactions

- Neutron auth config:

  [filter:authtoken]
  paste.filter_factory = keystone.middleware.auth_token:filter_factory
  auth_host=192.168.25.10
  auth_port = 35357
  auth_protocol = http
  admin_tenant_name=services
  admin_user=neutron
  admin_password=neutron
  neutron_config { 'auth_strategy': value => $auth_strategy

- Nova auth config:

  [filter:authtoken]
  paste.filter_factory = keystone.middleware.auth_token:filter_factory
  auth_host = 192.168.25.10
  auth_port = 35357
  auth_protocol = http
  auth_uri = http://192.168.25.10:35357/v2.0
  admin_tenant_name = services
  admin_user = nova
  admin_password = nova_pass
  nova_config { 'auth_strategy': value => $auth_strategy

UCS and OpenStack – Smart Bundles Reference

Cisco OpenStack Configurations

"Smartplay bundles" - building blocks

Compute Intensive

(2) UCS 96-Port 6296 Fabric Interconnect
(2) Nexus 2232 PP

(6) UCS C220 M3 Servers each with:
2 Intel Xeon E5-2665 Processors, 128GB of Memory, Cisco UCS VIC 1225, LSI MegaRAID, 2 x 900GB 10K SAS HDDs

Compute and Storage

(2) UCS 96-Port 6296 Fabric Interconnect
(2) Nexus 2232 PP

(6) UCS C220 M3 Servers each with:
2 Intel Xeon E5-2665 Proc, 128GB of Memory, LSI MegaRAID, Cisco UCS VIC 1225, 2 x 600GB 10K SAS HDDs
(2) UCS C240 M3 Servers each with:
2 Intel Xeon E5-2665 Proc, 256GB of Memory, LSI Mega RAID, Cisco UCS VIC, 12 x 900GB 10K SAS HDDs

Large-scale Storage

(2) UCS 96-Port 6296 Fabric Interconnect
(2) Nexus 2232 PP

(8) UCS C240 M3 Servers each with:
2 Intel Xeon E5-2665 Proc, 256GB of Memory, LSI MegaRAID, Cisco UCS VIC 1225, 12 x 900GB 10K SAS HDDs

UCS C220 M3

UCS C240 M3

RedHat OpenStack with Cisco UCS

Openstack Architecture on Cisco UCS Platform

Cisco Validated Design

November 2013

The following are the components used for the design and deployment:

- Cisco Unified Compute System (UCS) 2.1(2)
- Cisco C-series Unified Computing System servers for compute and storage needs
- Cisco UCS VIC adapters
- RedHat OpenStack 3.0 architecture
  - CEPH storage module supported by Ink Tank

Step 1. Packstack

Packstack is a Red Hat Enterprise Linux OpenStack Platform 3 installer. Packstack uses Puppet modules to install parts of OpenStack via SSH. Puppet modules ensure OpenStack can be installed and expanded in a consistent and repeatable manner. This reference architecture uses Packstack for a multi-server deployment. Through the course of this reference architecture, the initial Packstack installation is modified with OpenStack Network and Storage service enhancements.

Build Following Reference Architecture

- Multi-tenant (multi-project), compute, network, storage
- Use Quantum for network, L3Agent (virtual router) for L3 segregation
- Per tenant network, L3/NAT for segregation
- shared “public” network
- Non-HA control plane

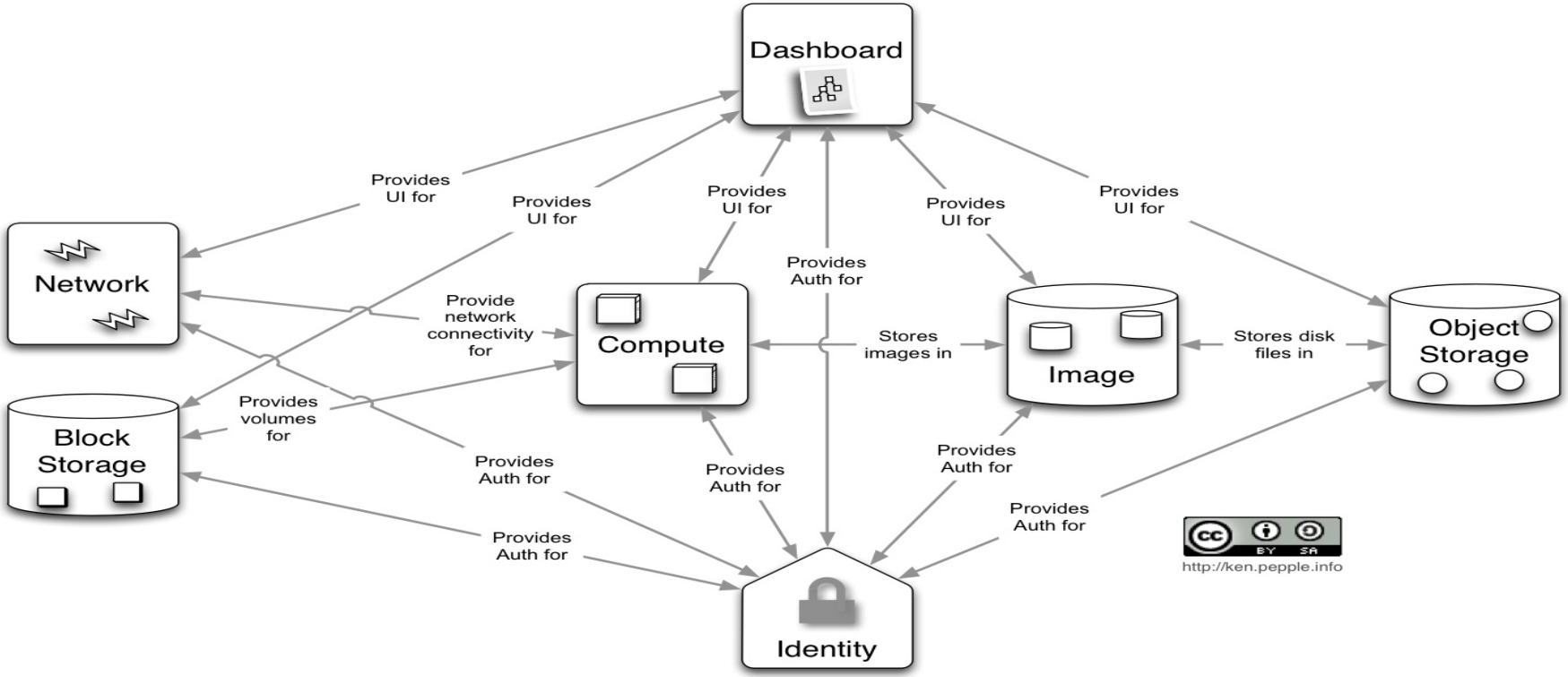
Build Process

- Build a build only node (may even be a VM), or build out a control node with “build” capabilities
  - Git clone the upstream repositories (or start with the puppet_openstack_builder codebase)
  - Update any configuration parameters based on documented guidelines
  - Run the baseline install script that configures the build system, and may deploy the OpenStack control components
- Power-on the rest of the system, let cobbler and puppet work their magic

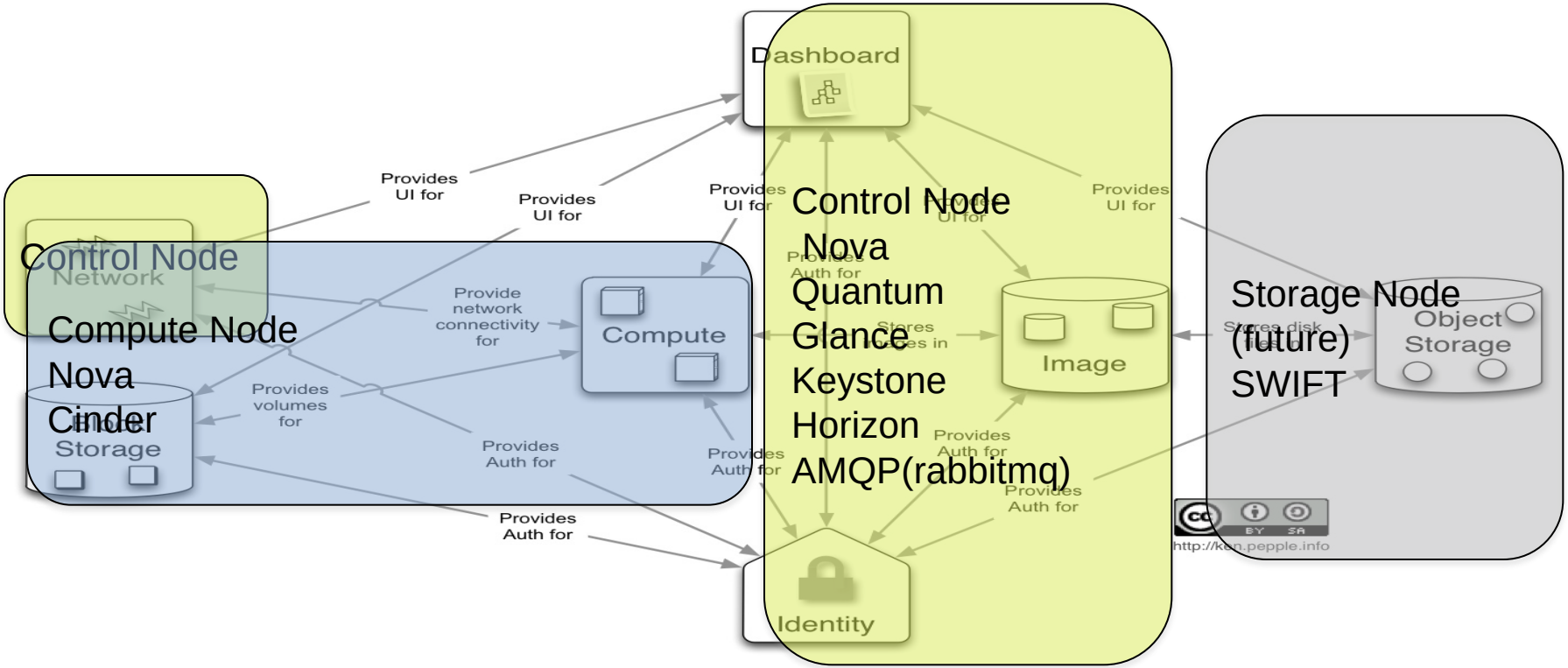



Let’s Build One!

Why all of this Complexity?



Node Type Distribution



Collect The Pieces You Need

- Reference Architecture Hardware
  - http://cisco.com/go/openstack
- Ubuntu Linux OS (best current support)
  - http://releases.ubuntu.com/precise, 12.04.4 x86_64 server version
- Access to the internet from the cloud system(s)
  - https://github.com/CiscoSystems
  - ftp://ftpeng.cisco.com/openstack/cisco
  - http://us.archive.ubuntu.com/
- Havate project: Build an all-in-one ISO
  - https://github.com/Havate/havate-openstack

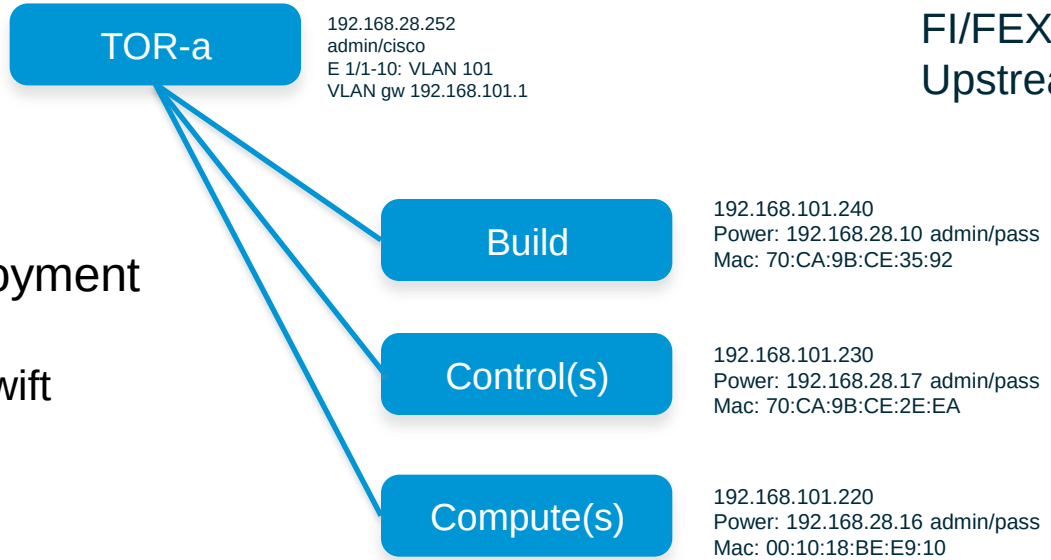
Puppet Modules - OpenStack

- Puppetlabs Github is source of record
  - puppetlabs-openstack
  - puppetlabs-nova
  - puppetlabs-quantum
  - puppetlabs-keystone
  - puppetlabs-rabbitmq
  - puppetlabs-glance
- https://github.com/puppetlabs
- Cisco validated variants (and quantum work)
- https://github.com/CiscoSystems

Collect The Address Information Needed

- IP addresses for management interfaces:
  - Build node, control node, compute node(s)
- MAC addresses from control/compute nodes
- DNS information (or at least upstream dns server)
- Determine Neutron network model
- Determine physical host interaction requirements

- OpenStack Puppet Deployment Demo
  - All-in-one Control/Build/Swift
  - Compute
  - Compute



Not Shown: OOB Network
FI/FEX
Upstream net

Install The First Node (manual)

- Use UCSM or CIMC interface to provide remote KVM and virtual CD
- Mount the ISO, and build the node. Default options, LVM against the local RAID configuration, OpenSSH installed. Havate ISO can automate most of this.
- When build is complete
  - Verify cobbler configuration for additional nodes
  - Power on (manually, or via script if power information is known)additional nodes

Deploy OpenStack Platform

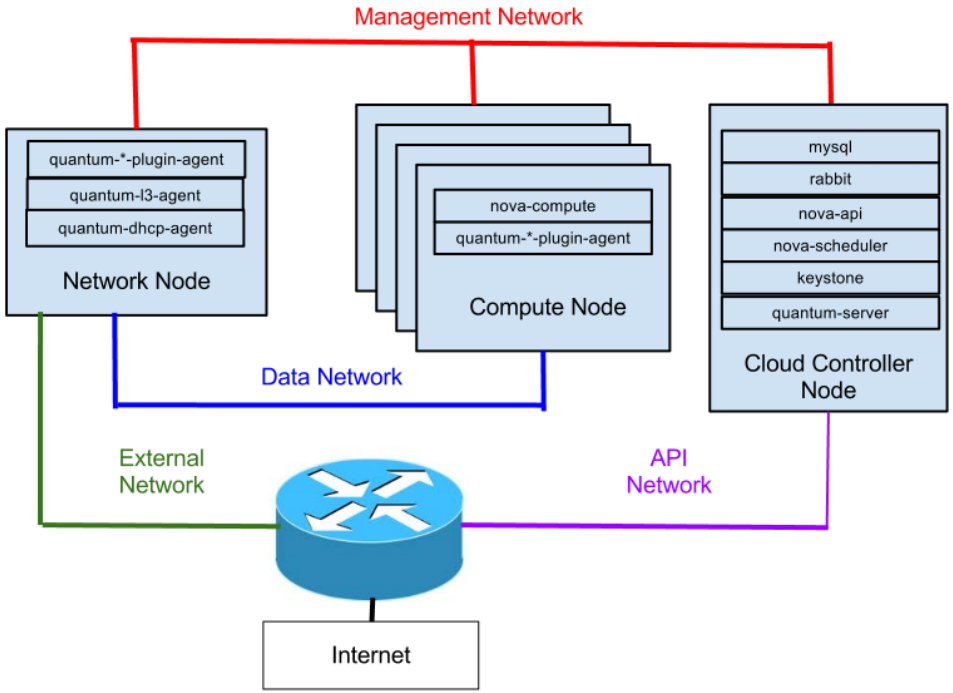
That's The Basics – Now For The Network

- Leveraging OpenVirtualSwitch for Linux based switching
- Leveraging Linux Iptables for firewall/router
- Using DNSMasq for DHCP and DNS proxy services
- For network, there are two deployment choices
  - Nova-network, proven, in production
  - Quantum, lots of testing, still not common in production use

OpenStack Network History

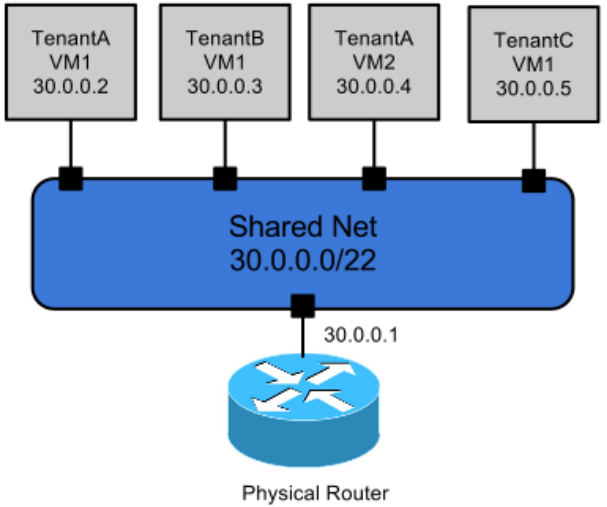
- NASA Nebula cloud
  - –Principally VPN session to “VLAN” segregated network
  - –FlatDHCP model per tenant with VPN outside->in access
- NOVA
  - –Flat model
  - –Flat + DHCP/iptables/meta-data
  - –Flat+DHCP on each compute node “multi-host”
  - –VLAN, like Flat, but with more than “one” target
- Neutron
  - –Break network out of Nova
- NOVA includes more than “network”
  - – DHCP
  - – L3
  - – IPAM
  - – Metadata (Cloud-Init)
- Neutron adding them
  - – Melange merged - Adds IPAM
  - – L3 agent extracted, but missing capabilities
  - – Service Insertion coming or here
  - – VPN, L3/HA, LB, FW, etc.

Quantum – Network Models



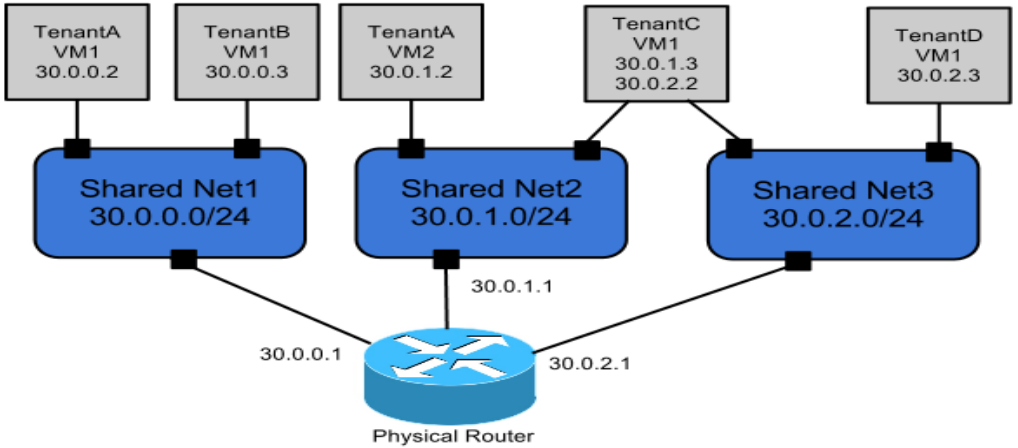
Single Flat or Provider Network

- Simple model
- Equivalent to nova-network VLAN
- No dhcp, metadata, NAT, etc.
- All tenants/projects see each other
- Router managed by something other than OS



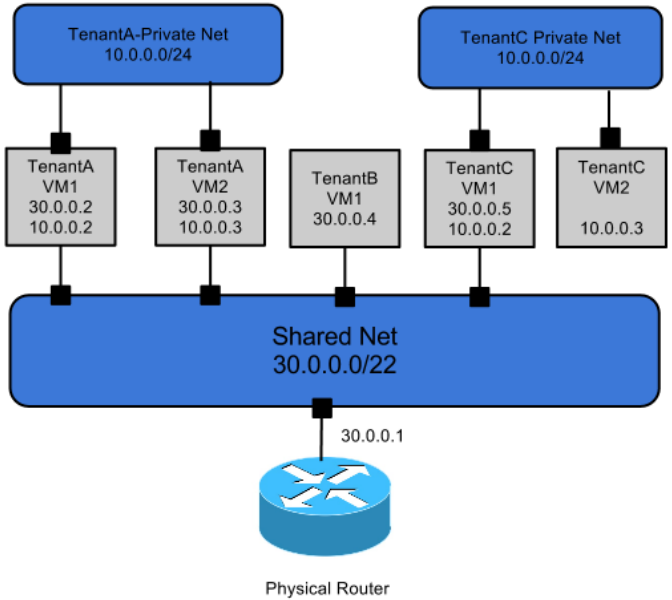
Shared Flat Networks

- Provides a model to expose multiple L2 domains to end users
- Can provide some tenant segregation
- Still no dhcp, metadata, NAT



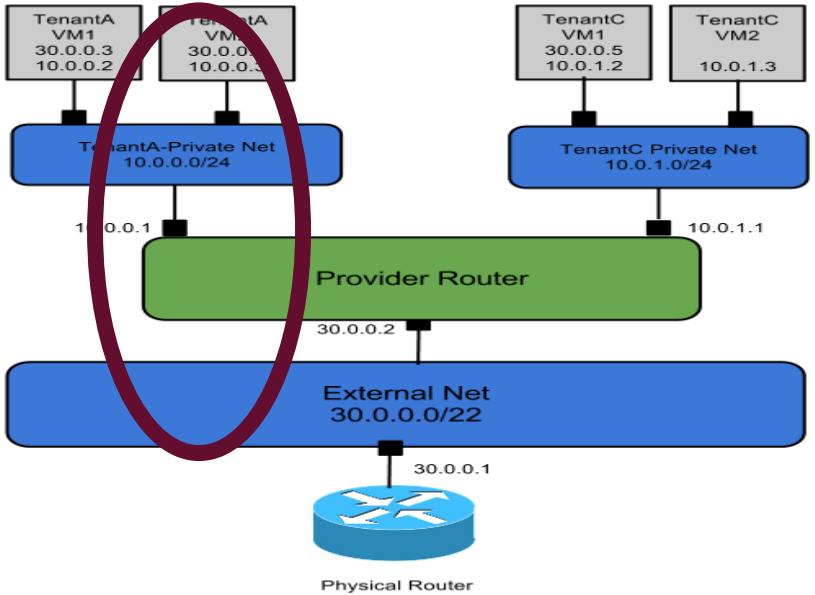
Mixed Flat and Private Networks

- Private networks (big difference in this example), provide:
  - DHCP
  - metadata
- Shared network in this model is as before

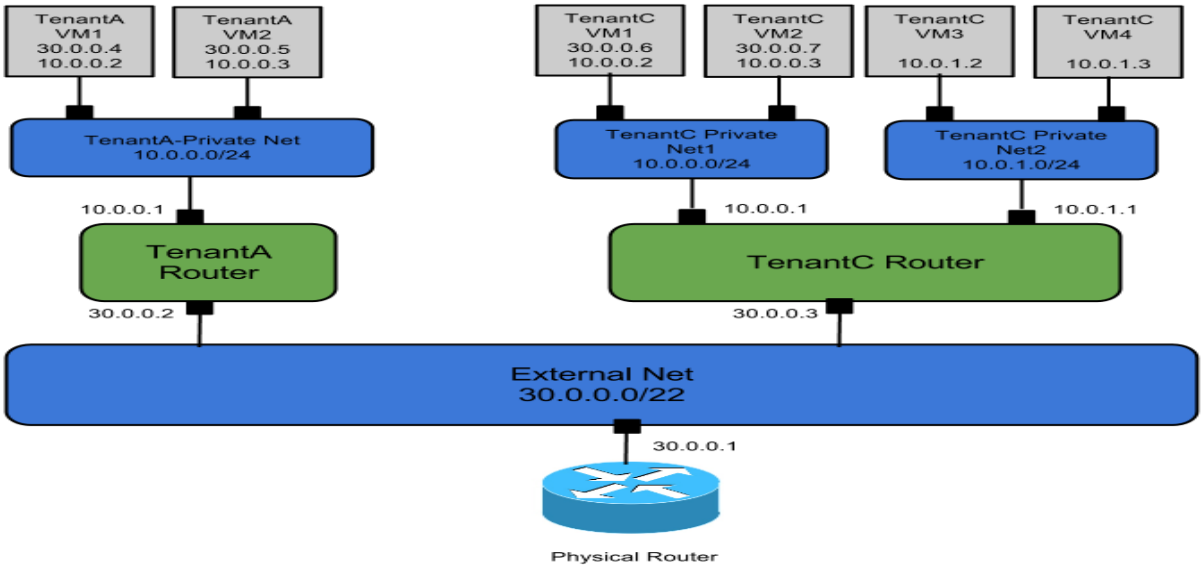


Provider Router and Private Networks

- Provider router:
  - Adds NAT
- Other networks as before



Per Tenant Routers with Private Networks

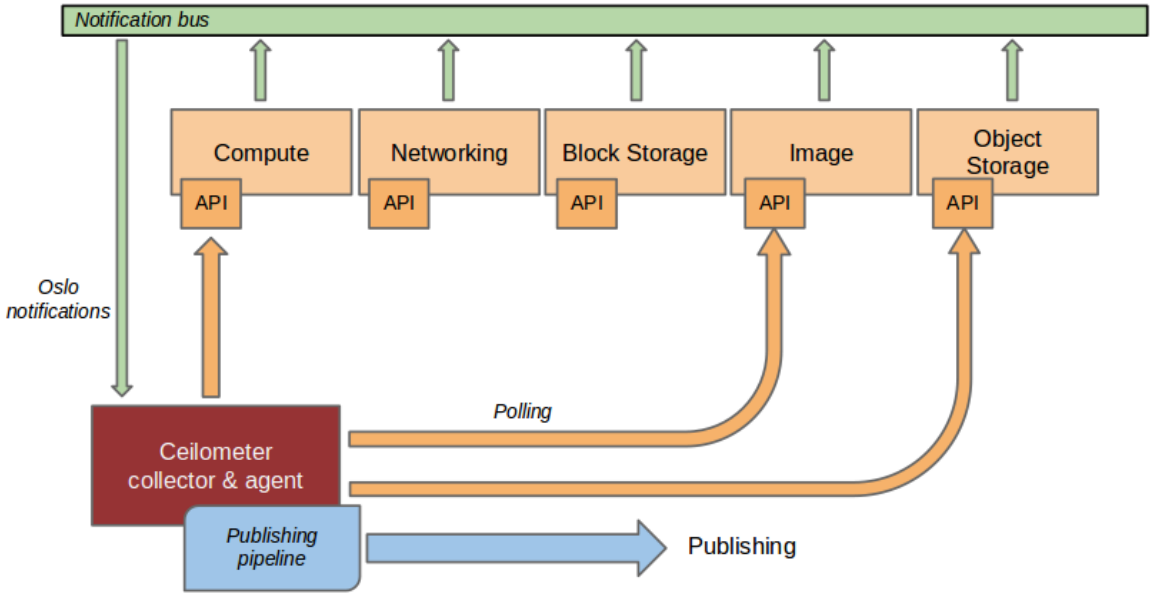


Great. Now What?

- Load Images
- Onboard Users
- Create additional tenants
- Deploy VMs
- Assign and manage quotas
- Connect in to billing/chargeback mechanisms

Assurance

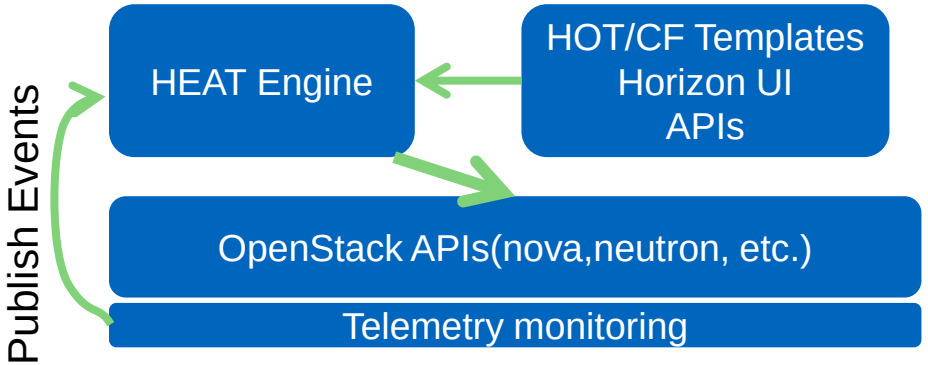
- Nagios/Zenoss/Xylan
  - IT Dashboard
- Collectd, Monitd, Statsd
  - RRD data collection
  - Process monitoring
- Graphite
  - Graph generation
- OpenStack Telemetry
  - Was Ceilometer project
  - Metering/Monitoring via AMQP/Rabbit



Application Development Workflow

- Template of App: HOT/Cloud Formation
- Integrate data from Telemetry
- Load against HEAT engine

- Template includes deployment service options
  - “masterless” puppet
  - Little-chef
  - Shell script(s)



HEAT/Cielometer/Nagios




Q & A

Complete Your Online Session Evaluation

Give us your feedback and receive a Cisco Live 2014 Polo Shirt!

Complete your Overall Event Survey and 5 Session Evaluations.

- Directly from your mobile device on the Cisco Live Mobile App
- By visiting the Cisco Live Mobile Site www.ciscoliveaustralia.com/mobile
- Visit any Cisco Live Internet Station located throughout the venue

Polo Shirts can be collected in the World of Solutions on Friday 21 March 12:00pm - 2:00pm

Learn online with Cisco Live!

Visit us online after the conference for full access to session videos and presentations.
www.CiscoLiveAPAC.com


CISCO
TM