



@randybias 

OpenStack Architected Like AWS (and GCP)

OpenStack's future is hybrid to public clouds

Randy Bias, Founder & CEO
OpenStack Juno Design Summit
May 14th, 2014



CCA - NoDerivs 3.0 Unported License - Usage OK, no modifications, full attribution*
* All unlicensed or borrowed works retain their original licenses

cloudscaling

Who

OpenStack Foundation Board of Directors

Scale Builder:

100k+ sq ft datacenters @ 14MW

1000s of servers, 100s of switches

Prod. OpenStack pioneer, Cloudscaling:

Part of OpenStack community since July 2010 (launch)

Firsts:

OpenStack Public Compute Cloud in U.S., Public Storage Cloud in U.S. and KR

Top 10 Cloud Computing Pioneer

InformationWeek

A Leader in Production Deployments*

supported clouds	→	10+
total cores	→	8,000+
total storage	→	~8.5PB
total servers	→	~600
# of Fortune 15?	→	2

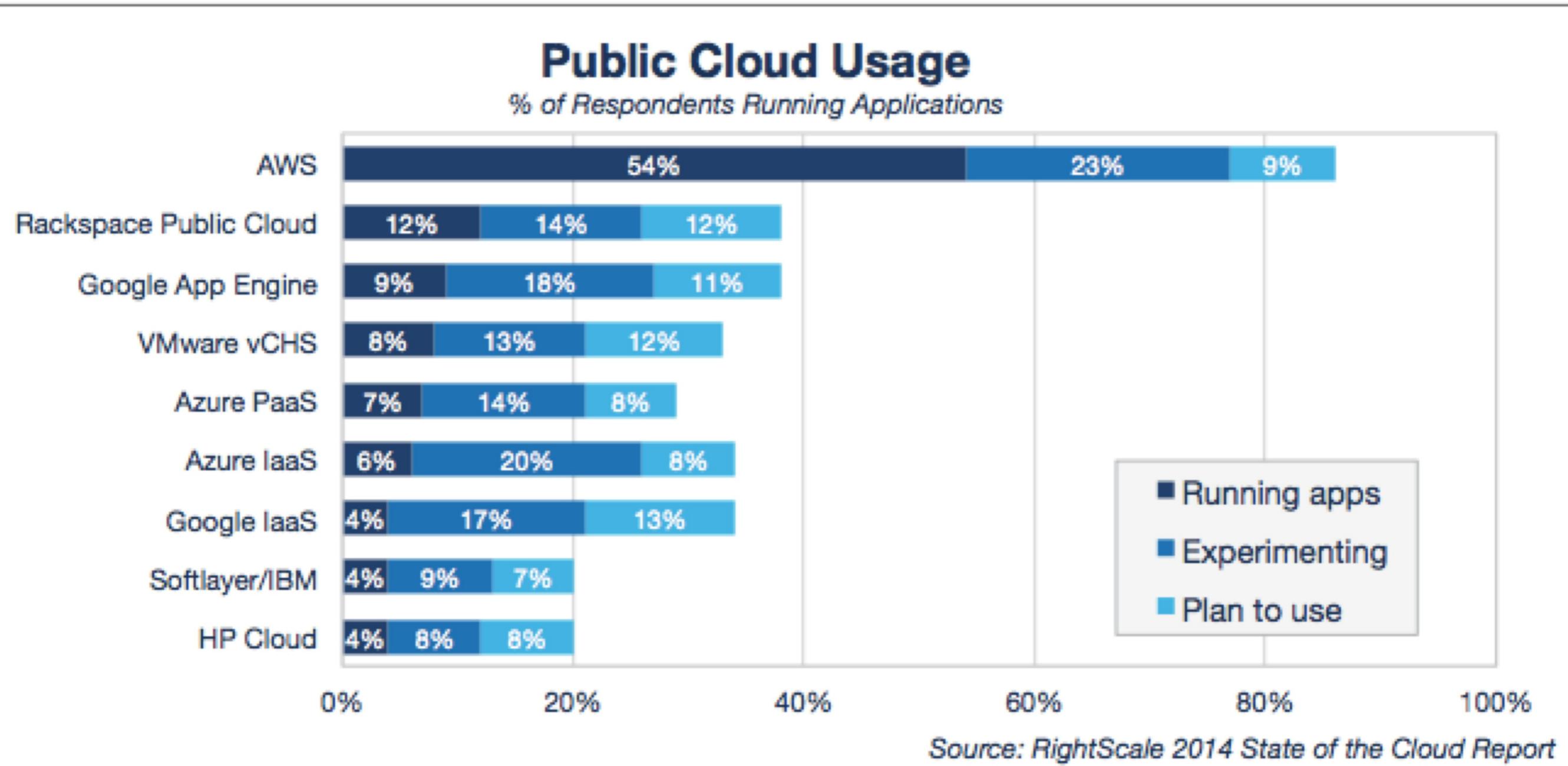
* e.g. Currently actively supported clouds; legacy clouds from PS days are not included (multiply by 2-3x if you care)

Today's Goals

- 1 Why an AWS-like “elastic” cloud?
- 2 Hybrid Cloud Interoperability
- 3 AWS Flavored OpenStack
- 4 Bring It Home

Why Elastic Cloud?

Top Clouds Are Elastic



What is an Elastic Cloud?



Google
Compute
Engine

Epitomized by AWS/GCE

- Scale-out model over scale-up model
- Small failure domains



Foundation for Cloud Native Apps

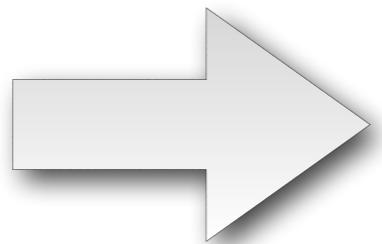
- Apps that “route around failure”
- Manage own data replication/redundancy
- Automated deployments

Economics are King

- Open source software
- Vanity-free commodity hardware
- Low-cost, scale-out networking

For more information see the Chicago DevOps presentation at <http://tinyurl.com/elasticcloud>

Elastic Cloud is a Mindset Change



bowzer.company.com
(*pets*)

vs.

web001.company.com
(*cattle*)

Cloud is the era of the disposable server

Attribution: Bill Baker, Distinguished Engineer, Microsoft

Elastic Clouds Host Cloud Native Apps

Legacy Enterprise Model
(inelastic)

99.9%
Applications

99.999%
Infrastructure
(\$\$\$\$)

Cloud Enterprise Model
(elastic)

99.999%
Applications

99%
Infrastructure
(\$\$)

Responsibility
Shift

Cloud Native Apps Drive Agility

Traditional Applications

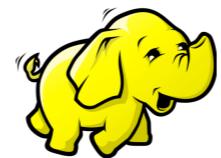


2012 → 89M

2015 → 139M

+56%

New Scale-Out Applications



2012 → 5M

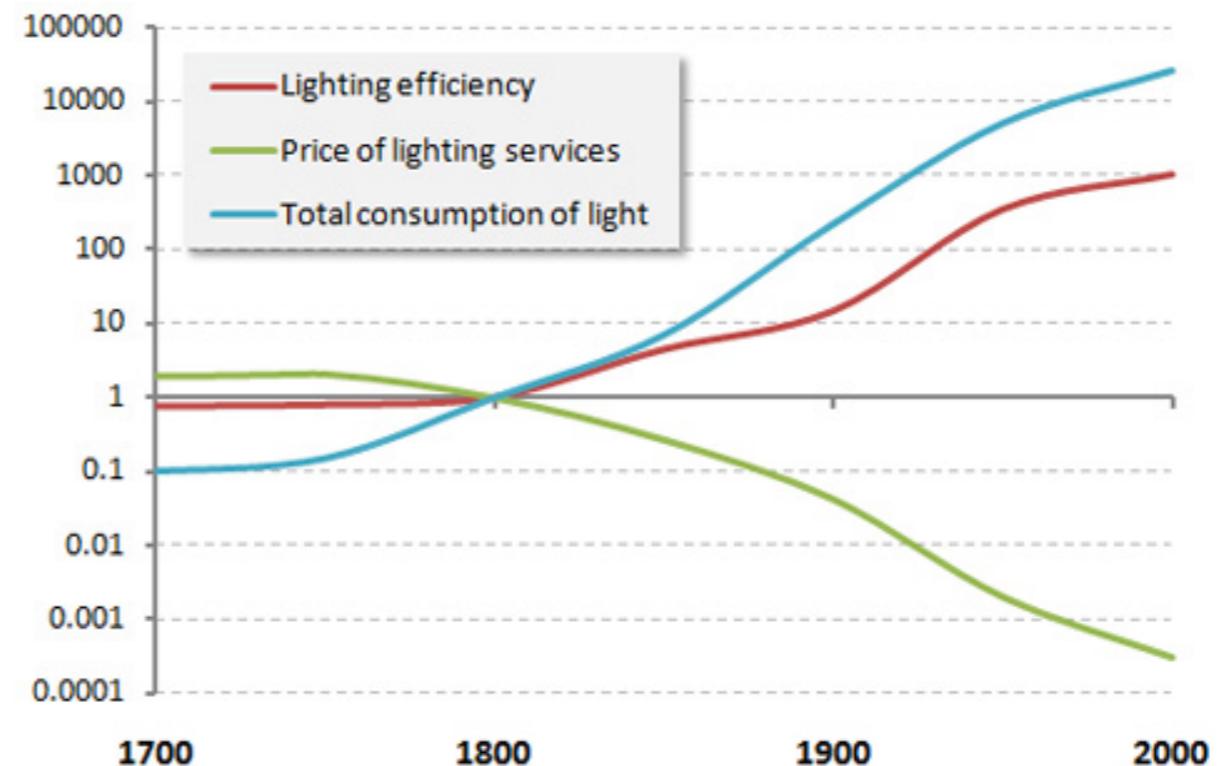
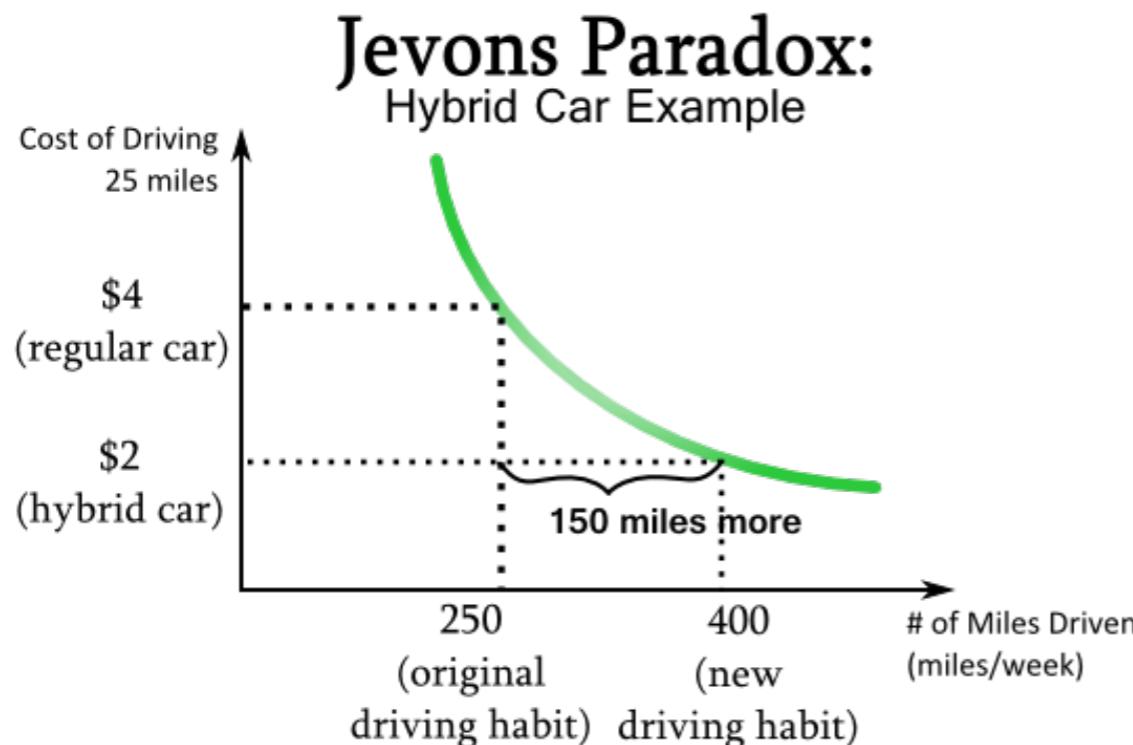
2015 → 32M

+540%

Source: IDC & Gartner via EMC @ http://stevetodd.typepad.com/my_weblog/2013/11/a-mobile-to-storage-analytics-architecture.html

Jevon's Paradox Accelerates Innovation

As cost ↓ ... consumption ↑



As the cost of compute approaches zero...
↓ cost of experimentation and ↑ agility drives innovation

* SOURCE: <http://greenimalist.com/2011/04/the-jevons-paradox/>

** SOURCE: <http://blogs.shell.com/climatechange/2010/09/the-other-side-of-energy-efficiency/>

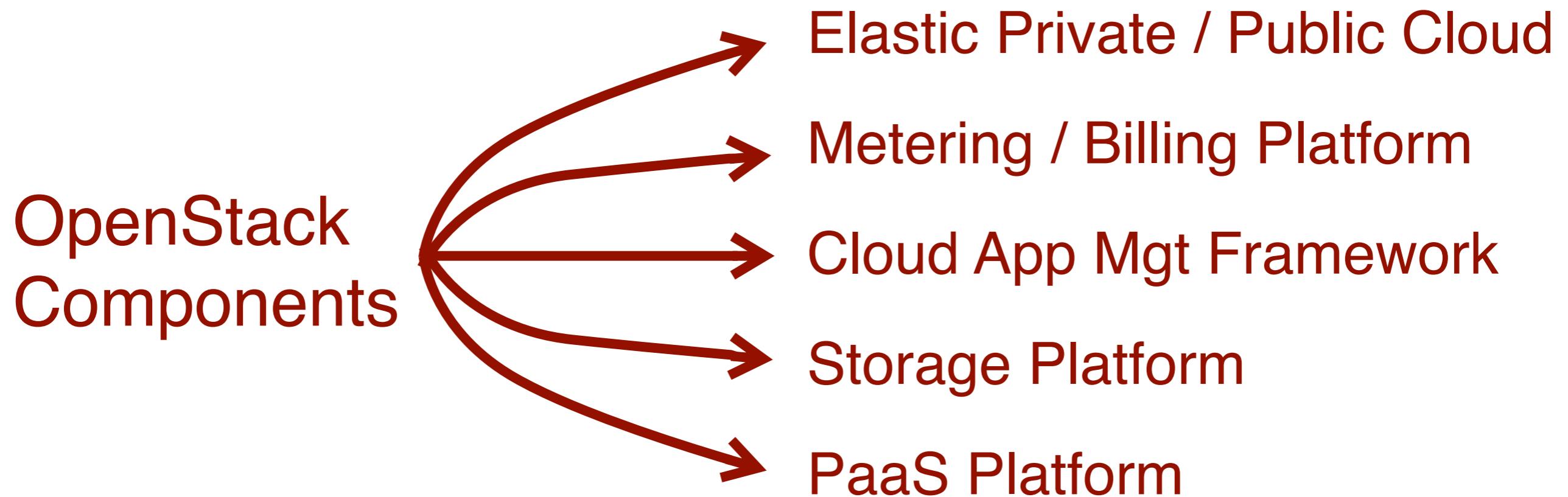
Why Should the Enterprise Care?

“Seventy percent of the companies that were on the Fortune 1000 list a mere 10 years ago have now vanished – unable to adapt to change . . . ”

— Forrester analyst Craig Le Clair*

* SOURCE: http://blogs.forrester.com/craig_le_clair/13-09-09-make_business_agility_a_key_corporate_attribute_itCouldBeWhatSavesYou

OpenStack: For ALL Clouds



Although ... the best fit is elastic cloud

Hybrid Cloud Enables Biz Agility

- 1 Top public clouds are elastic
- 2 Elastic clouds have similar attributes
essentially they share a “reference architecture”
- 3 Agility is primary value for enterprise
- 4 Cloud native apps + DevOps drive agility
they “route around failures”
- 5 Elastic cloud is ideal platform for these apps
doesn’t matter if it’s private or public
- 6 Businesses need choice -> hybrid cloud

Hybrid Cloud Interoperability

Hybrid Cloud Value Proposition

Application Portability ~ Independence

Move apps between clouds with impunity

Advanced capabilities such as Cloud-bursting/bridging

Future-proofing and reduction of vendor lock-in

Pick the right cloud for the job!

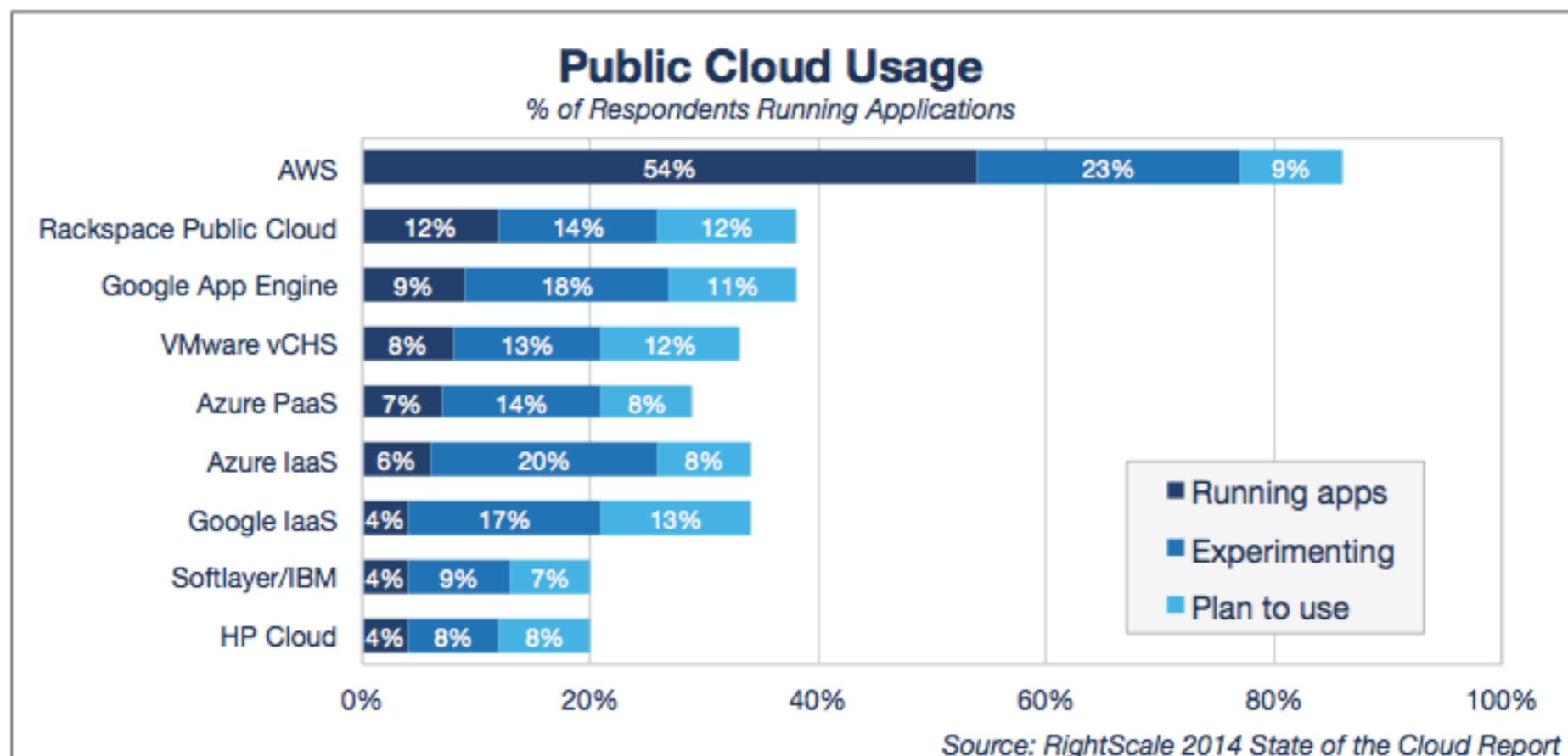
Portability Requires Interoperability

API compatibility is a subset of what is required!

Interoperability Is NOT Easy (details to follow)

Pub Cloud Interop Matters

Private Clouds Must be Interoperable with Public Cloud Leaders



OpenStack's Future is Hybrid

API Compatibility is Not Enough



Same API



Infiniti G37

Semantics
& Architecture

Engine

Braking

Lighting

Transmission

Suspension



Mac Truck

330 HP
270 lb ft torque
5 Second 0-60

Different Behavior

600 HP
2,100 lb ft torque
60 Second 0-60

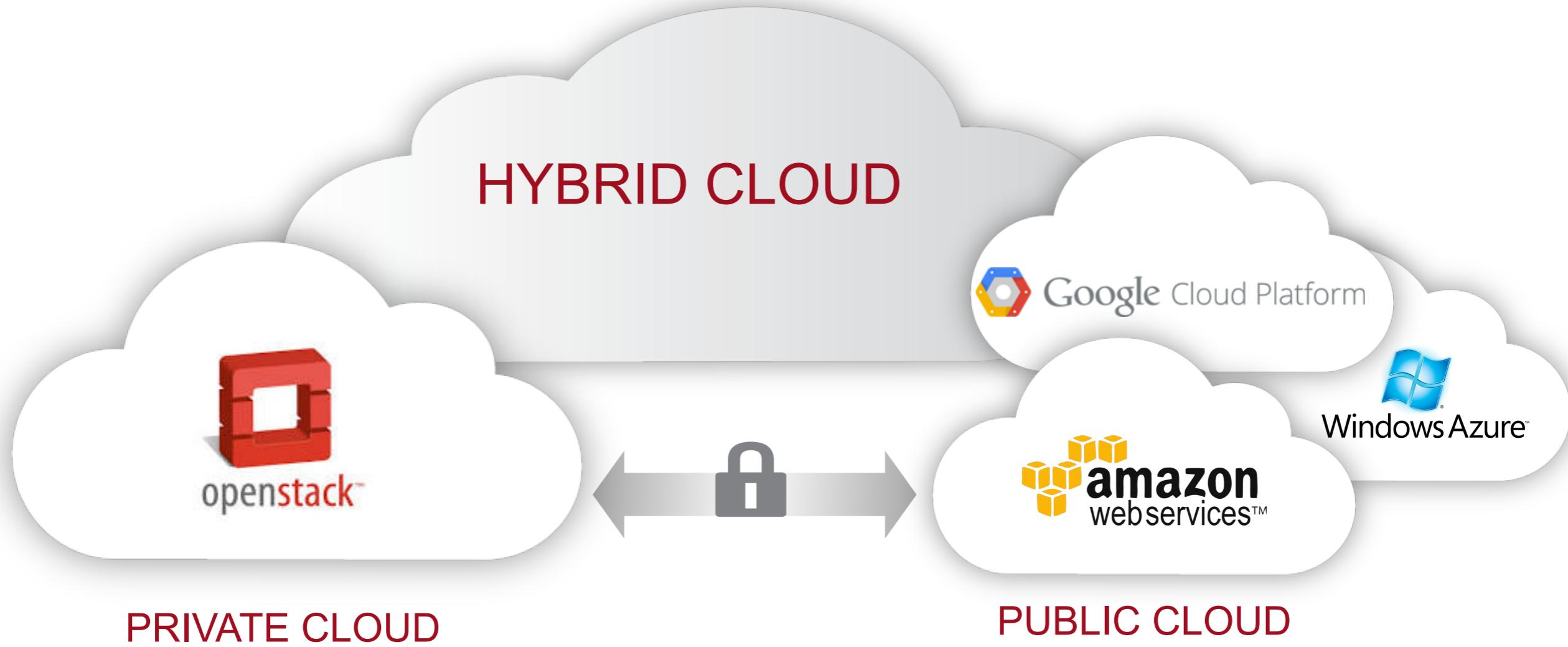
Hybrid Cloud Interop Requirements

- 1 Matching SLAs & Availability
- 2 Performance & QoS Guarantees
- 3 Infrastructure Feature Parity
- 4 Similar TCO / Economies of Scale
- 5 API Compatibility
- 6 Behavioral Compatibility

Hybrid Cloud Interop Requirements

Shared Requirement	Public / Private Cloud Interoperability Detail
1 Matching SLAs and Availability Guarantees	VM uptime (hypervisor availability) API endpoint availability Storage system availability
2 Performance & QoS Guarantees	VM vCPU, networking & local storage performance Block storage system performance matching Object storage aggregate throughput
3 Infrastructure Feature Parity	Must have a majority of same primary features AWS -> EC2, S3, EBS, VPC, ELB GCP -> VMs, persistent block/object, virt. networks, etc.
4 Similar Cost Economics & Economies of Scale	Within ½-2x of wholesale costs/pricing TCO must be similar, particularly for operational costs
5 API Compatibility	Support for public cloud APIs At the least: AWS, GCE, Azure, and OpenStack
6 Behavioral Compatibility	Implied, non-API behavior is same or similar example 1: VMs start in <30 mins example 2: floating/elastic IP auto-assignment

Enable a Hybrid-first Cloud Strategy



Interoperability means
APIs, QoS, SLAs, TCO, Feature Parity & Same Behavior

AWS Flavored OpenStack

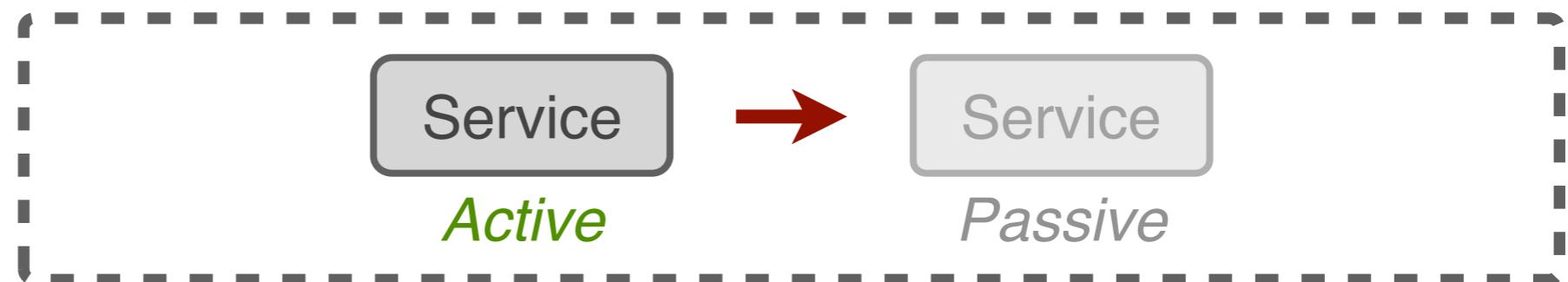
AWS Interoperable OpenStack

- 1 Matching SLAs & Availability
- 2 Performance & QoS Guarantees
- 3 Infrastructure Feature Parity
- 4 Similar TCO / Economies of Scale
- 5 API Compatibility
- 6 Behavioral Compatibility

1 Matching SLAs & Availability Guarantees

Delivers a 99.999% Uptime APIs & Control Plane on 99.5% Hardware

HA Pair Failover
/ Master Election



VS.

Load
Balancing

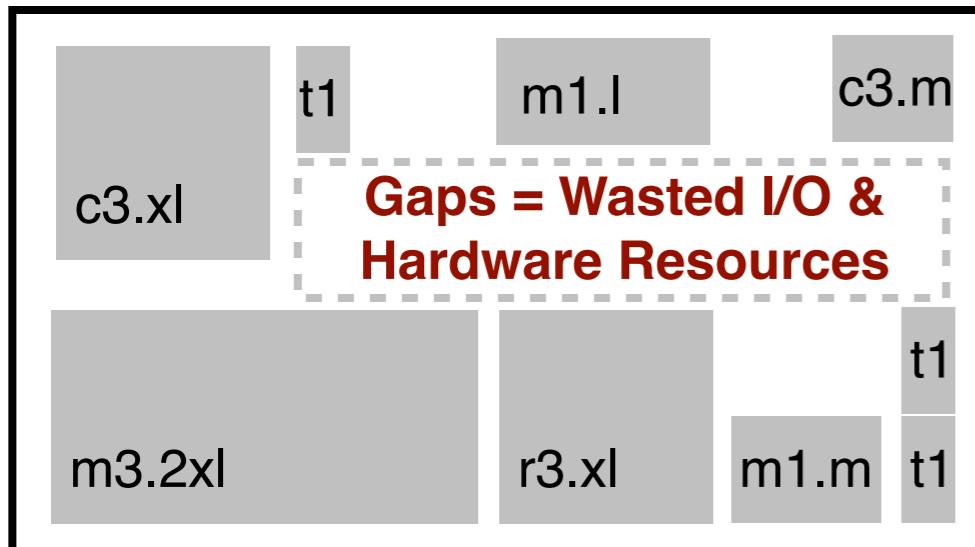


With Load Balancing, the control plane scales as the cloud grows

Performance & QoS Guarantees

AWS/GCP-Style Instance Sizing & Scheduling

Spread First Scheduling

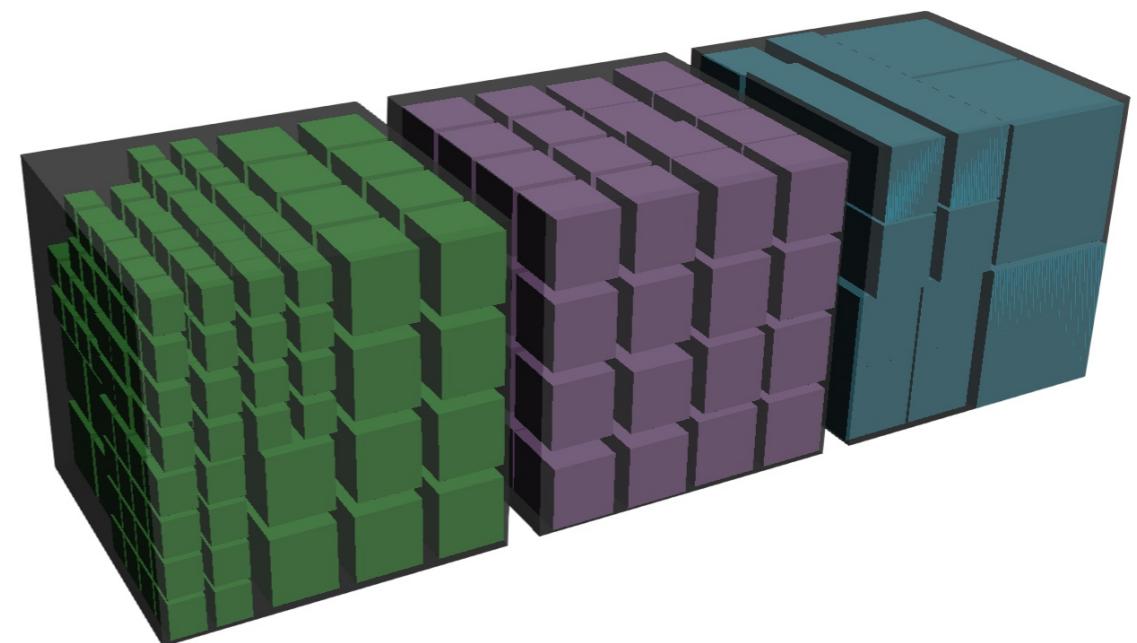


Bin Packing Scheduling (by Instance Family)



Bin Packing Scheduler

- 1) fixed proportions of resources
- 2) all resources allocated
- 3) manage oversubscription rates
- 4) ensures Quality of Service (QoS)
- 5) easy to do w/ filters & host-aggregates



3 Matching Infrastructure Feature Parity

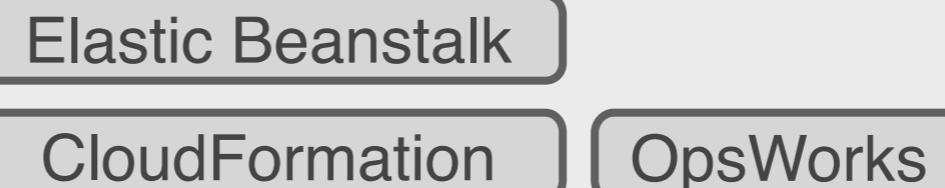
Core Infrastructure Services



Infrastructure Application Services



PaaS / Orchestration



Applications



Matching Infrastructure Feature Parity

Core Infrastructure Services



Mgt Console

CLI

IAM

Auto Scaling

CloudWatch

ELB

VPC Networking

Classic Networking

VM Import/Export

S3

EBS

EC2

Horizon

CLI

Keystone



Heat Auto Scaling

Heat Monitoring

LBaaS

Neutron SDN Plugin

L2/L3 Nova Networking

Glance

Swift

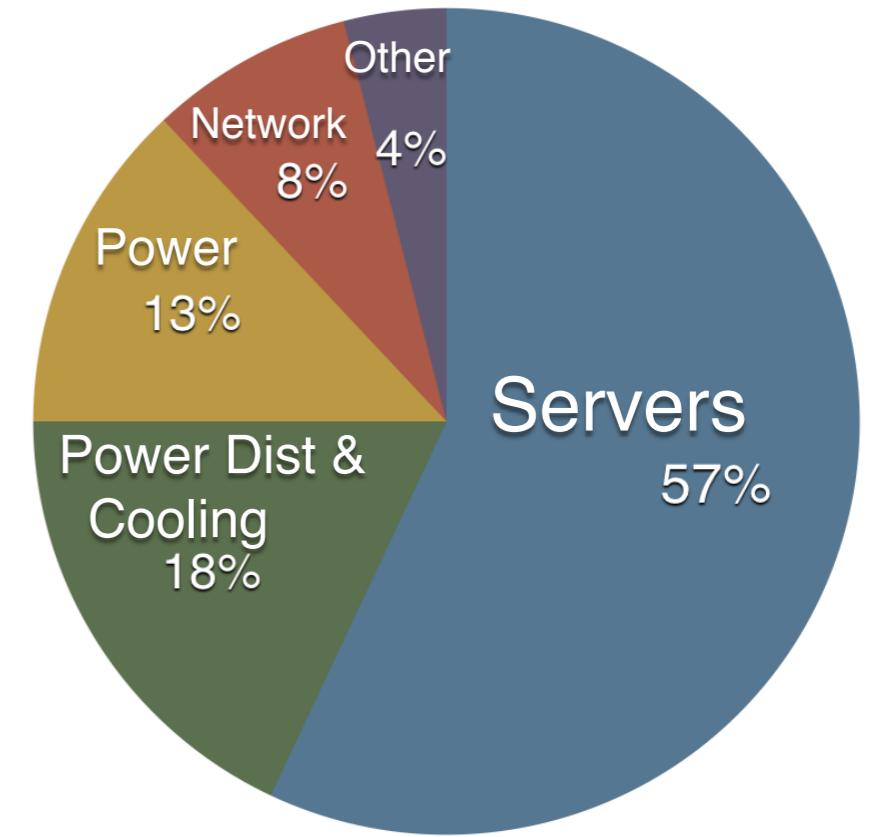
Cinder

Nova

4 Similar TCO & Economies of Scale

What is required to meet Amazon's cost structure?

- #1 Cost = Servers
- #2 Cost = Power
- #3 Cost = Network



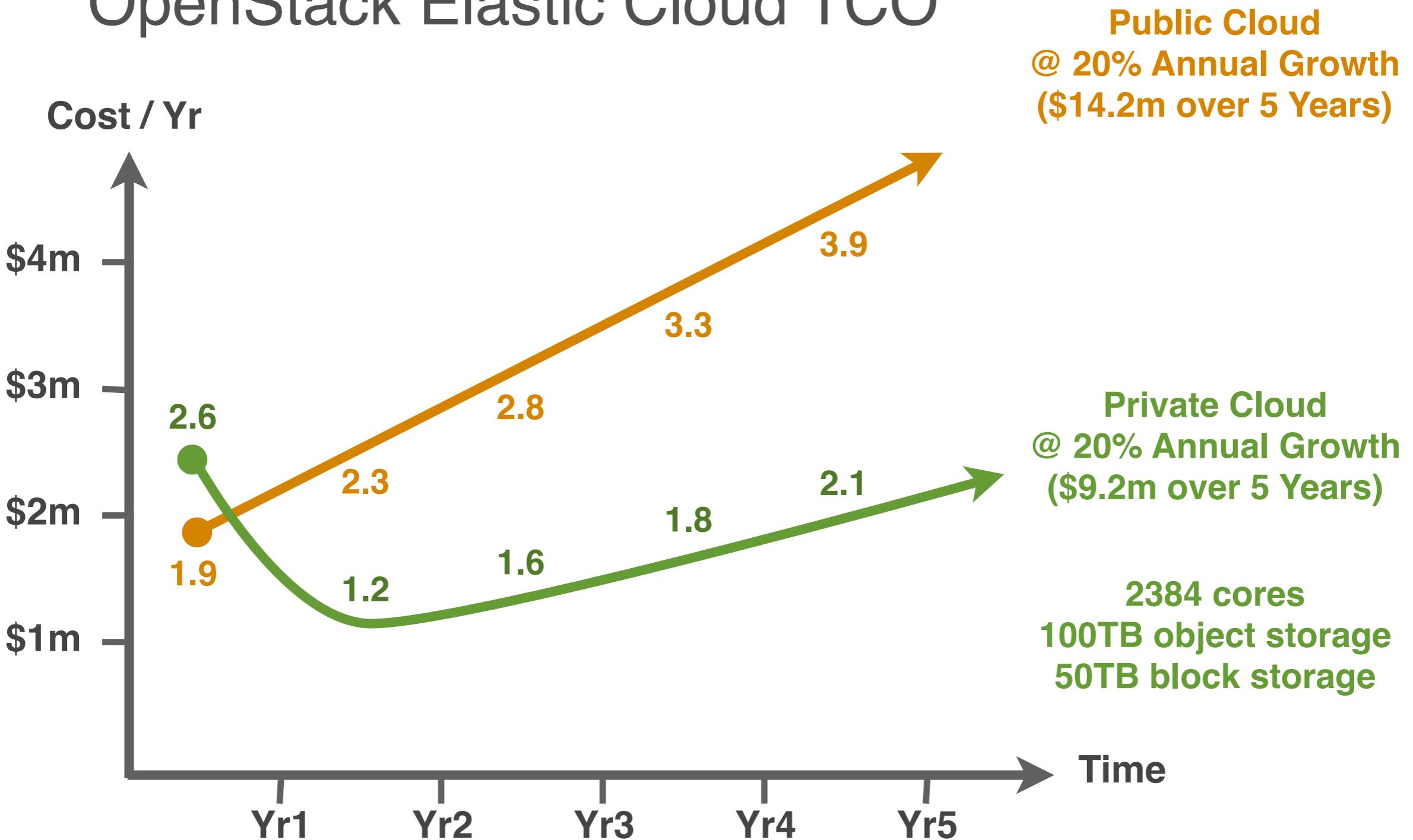
Use industry standard servers & build a cattle cloud
(one NIC, one switch, one PSU, etc)
reduces costs ***significantly!***

* SOURCE: http://mvdirona.com/jrh/TalksAndPapers/JamesHamilton_Reinvent20131115.pdf

4

Similar TCO & Economies of Scale

OpenStack Elastic Cloud TCO



35% savings over AWS over 3-5 years

API Compatibility

OpenStack already has multiple APIs

Native APIs

AWS APIs

GCE APIs

OpenStack “Core” Services

6 Behavioral Compatibility Testing

Implicit non-API behavior crucial to test

We have already been testing it w/ Tempest
Tempest has a pretty sweet set of AWS tests (see next)

Some of them are turned off by default
OpenStack default settings don't pass it seems

Cloudscaling has created more

Ultimately we need exhaustive Tempest
AWS behavioral test suite

Then GCP, then Azure

6 Tempest Tests for AWS Behavior

	AWS	OCS	OpenStack
<code>tempest.thirdparty.boto.test_s3_buckets.S3BucketsTest</code>			
<code>test_create_and_get_delete_bucket</code>			
<code>test_create_get_delete_object[gate,smoke]</code>			
<code>tempest.thirdparty.boto.test_ec2_network(EC2NetworkTest)</code>			
<code>test_disassociate_not_associated_floating_ip</code>			
<code>tempest.thirdparty.boto.test_ec2_volumes(EC2VolumesTest)</code>			
<code>test_create_get_delete[gate,smoke]</code>			
<code>test_create_volume_from_snapshot[gate,smoke]</code>			
<code>tempest.thirdparty.boto.test_ec2_security_groups(EC2SecurityGroupTest)</code>			
<code>test_create_authorize_security_group[gate,smoke]</code>			
<code>tempest.thirdparty.boto.test_ec2_instance_run(InstanceRunTest)</code>			
<code>test_integration_1[gate,smoke]</code>			
<code>test_run_idempotent_instances[gate,smoke]</code>			
<code>test_run_stop_terminate_instance[gate,smoke]</code>			
<code>test_run_stop_terminate_instance_with_tags[gate,smoke]</code>			
<code>test_run_terminate_instance[gate,smoke]</code>			
<code>tempest.thirdparty.boto.test_ec2_keys(EC2KeysTest)</code>			
<code>test_create_ec2_keypair[gate,smoke]</code>			
<code>test_delete_ec2_keypair[gate,smoke]</code>			
<code>test_duplicate_ec2_keypair[gate,smoke]</code>			
<code>test_get_ec2_keypair[gate,smoke]</code>			
<code>tempest.thirdparty.boto.test_s3_ec2_images(S3ImagesTest)</code>			
<code>test_register_get_deregister_aki_image</code>			
<code>test_register_get_deregister_ami_image[gate,smoke]</code>			
<code>test_register_get_deregister_ari_image</code>			

6 Cloudscaling Tempest Tests for AWS Behavior*

	AWS	OCS	OpenStack
<code>tempest.cloudscaling.thirdparty.aws_compat.test_ec2_instance_mysql.InstanceMySQLTest</code>			
<code>test_integration_mysql</code>		Green	Red
<code>tempest.cloudscaling.thirdparty.aws_compat.test_ec2_instance_restart.InstanceRestartTest</code>			
<code>test_reboot_instance</code>		Green	Red
<code>test_stop_start_instance</code>		Green	Red
<code>tempest.cloudscaling.thirdparty.scenario.aws_compat.test_ec2_scenario.EC2_ImageAPI</code>			
<code>test_check_ebs_image_type</code>		Green	Red
<code>test_check_ebs_image_volume_size</code>		Green	Red
<code>test_describe_image_with_filters</code>		Green	Red
<code>test_get_ebs_image</code>		Green	Red
<code>tempest.cloudscaling.thirdparty.scenario.aws_compat.test_ec2_scenario.EC2_VolumeAPI</code>			
<code>test_create_delete_volume</code>		Green	Green
<code>test_describe_volumes_with_filter</code>		Green	Red
<code>test_get_deleted_volume</code>		Green	Red
<code>tempest.cloudscaling.thirdparty.scenario.aws_compat.test_ec2_scenario.EC2_SnapshotAPI</code>			
<code>test_create_delete_snapshot</code>		Green	Green
<code>test_create_increased_volume_from_snapshot</code>		Green	Green
<code>test_delete_volume_leaving_snapshot</code>		Green	Red
<code>test_describe_snapshots_with_filter</code>		Green	Red
<code>test_get_deleted_snapshot</code>		Green	Red
<code>tempest.cloudscaling.thirdparty.scenario.aws_compat.test_ec2_scenario.EC2_EBSInstanceAPI</code>			
<code>test_create_root_volume_snapshot</code>		Green	Red
<code>test_describe_instances_with_filter</code>		Green	Green
<code>test_launch_terminate_ebs_instance</code>		Green	Green
<code>test_public_ip_is_assigned</code>		Green	Red
<code>tempest.cloudscaling.thirdparty.scenario.aws_compat.test_ec2_scenario.EC2_EBSInstanceAttaching</code>			
<code>test_attaching_by_full_name</code>		Green	Green
<code>test_attaching_by_short_name</code>		Green	Red
<code>test_check_attaching_state</code>		Green	Red
<code>tempest.cloudscaling.thirdparty.scenario.aws_compat.test_ec2_scenario.EC2_EBSInstanceTuneBDM</code>			
<code>test_launch_ebs_instance_with_creating_blank_volume</code>		Green	Red
<code>test_launch_ebs_instance_with_persistent_root_device</code>		Green	Red
<code>test_launch_ebs_instance_with_resized_root_device</code>		Green	Red
<code>tempest.cloudscaling.thirdparty.scenario.aws_compat.test_ec2_scenario.EC2_EBSInstanceResizeRootDevice</code>			
<code>test_resize_root_ebs_device</code>		Green	Red
<code>tempest.cloudscaling.thirdparty.scenario.aws_compat.test_ec2_scenario.EC2_EBSInstanceSnapshot</code>			
<code>test_create_ebs_instance_snapshot</code>		Green	Red

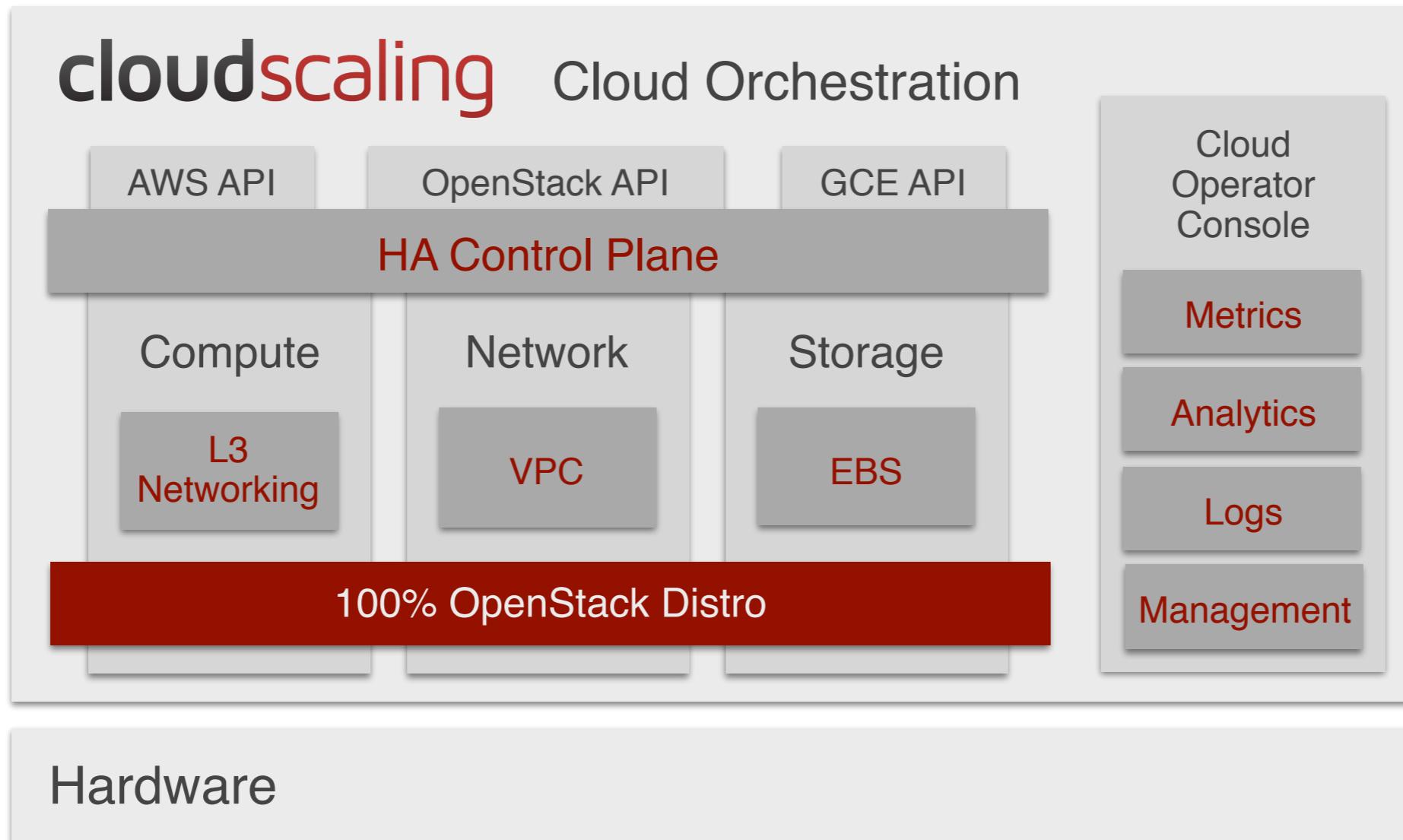
- * In process of upstreaming now
- * Plan to work with RefStack on “AWS” flavored reference stack for OpenStack

How Do You Know You're Done?

- 1 Same Availability
- 2 Same Performance
- 3 Similar Services
- 4 Equivalent TCO
- 5 Same APIs
- 6 Behaves Identically

Open Cloud System (OCS) Architecture

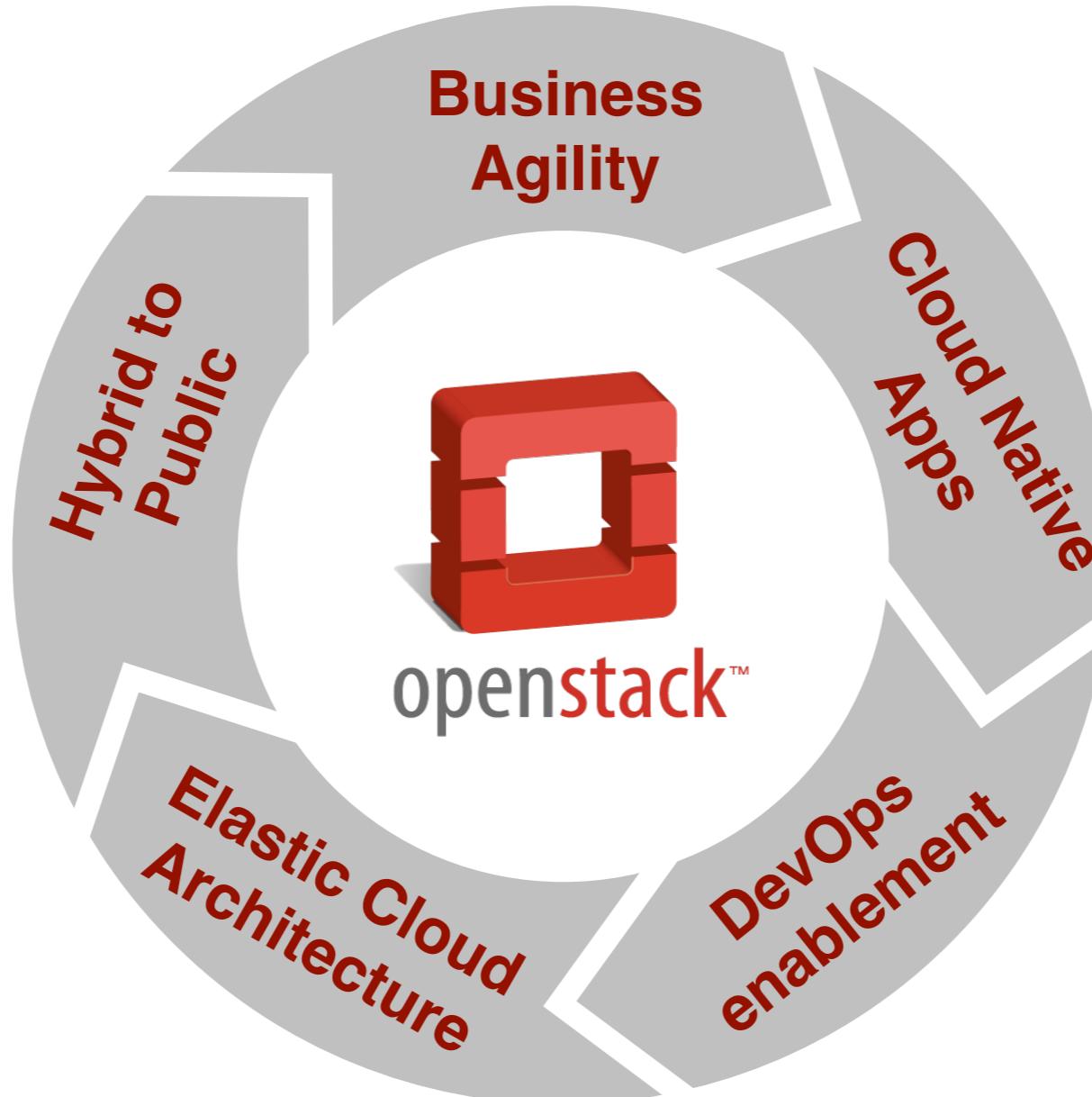
Available. Performant. Interoperable.



Hybrid Cloud Operating System

Agility, DevOps, & AWS + OpenStack

Bringing It Home



OpenStack provides ideal platform for a hybrid cloud operating system designed in the elastic cloud model



@randybias



Questions?