ESTR4300
Web-scale Information Analytics

Case Studies and Comparisons on Leading Cloud Service Providers

Prof. Wing C. Lau
Department of Information Engineering
wclau@ie.cuhk.edu.hk

Acknowledgements

- The slides used in this chapter are adapted from the following sources:
  - CS498 Cloud Computing, by Roy Campbell and Reza Farivar, UIUC.
  - Guest Lecture for CS498 of UIUC, “Distributed Services: AWS Overview,” Mirko Montanari, Jan 18, 2013.
  - NETS212 Scalable and Cloud Computing, by Andreas Haeberlen, Upenn
  - “Inside Windows Azure – A Cloud Operating System”, by Roger S. Barg, presented in LASER Summer School on Software Engineering, Sept 2013 http://laser.inf.ethz.ch/2013/lectures.php.
- All copyrights belong to the original authors of the materials.

History: Becoming a cloud provider

Technology	Cost in medium DC (~1,000 servers)	Cost in large DC (~50,000 servers)	Ratio
Network	$95 per Mbit/sec/month	$13 per Mbit/sec/month	7.1
Storage	$2.20 per GByte/month	$0.40 per GByte/month	5.7
Administration	~140 servers/admin	>1,000 servers/admin	7.1

Source: James Hamilton's Keynote, LADIS 2008

- Early 2000s: Phenomenal growth of web services
  - Many large Internet companies deploy huge data centers, develop scalable software infrastructure to run them
  - Due to economies of scale, these companies were now able to run computation very cheaply
  - What else can we do with this?

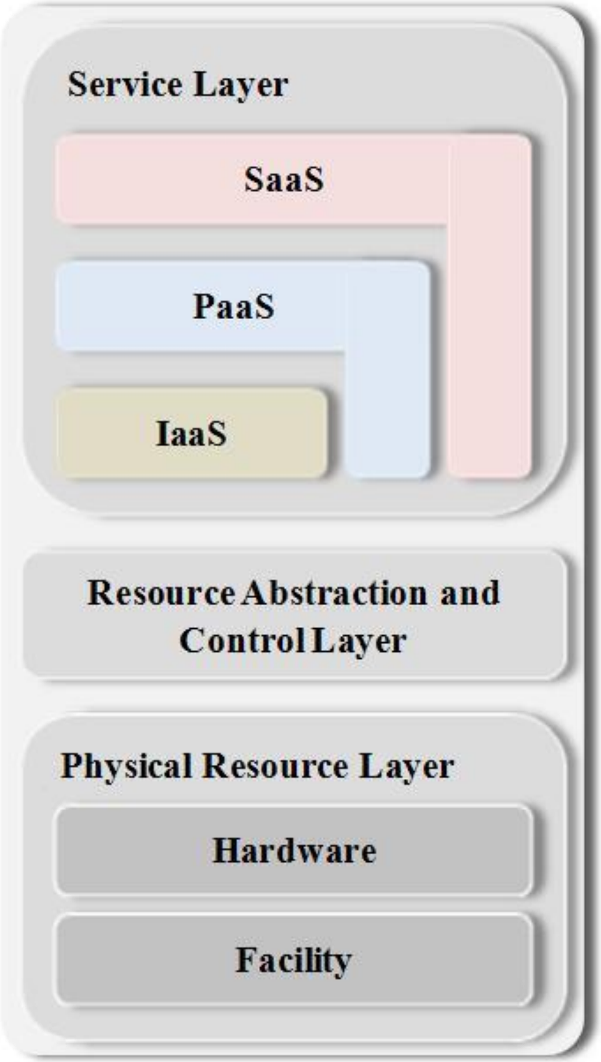
History: Incentives

- Idea: Use your existing data center to provide cloud services
- Why is this a good idea?
- Make a lot of money
  - Price advantage of 3x-7x → Can offer services much cheaper than medium-size company and still make profit
- Leverage existing investment
  - New revenue stream at low incremental cost (example: many Amazon AWS technologies were initially developed for Amazon's internal operations)
- Defend a franchise
  - Example: Microsoft enterprise apps → Microsoft Azure

History: Incentives (continued)

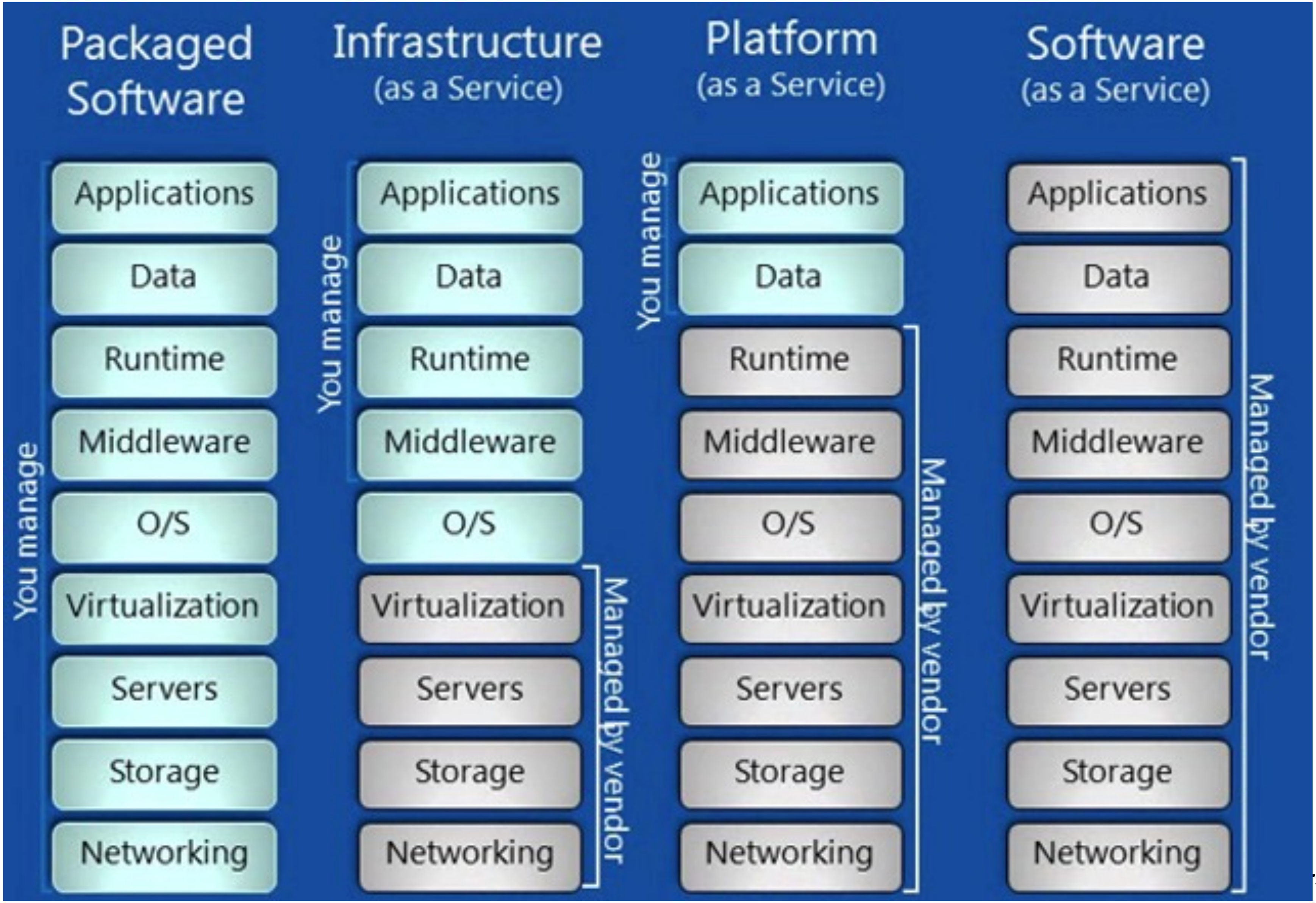
- Attack an incumbent
  - Company with requisite datacenter may want to establish a 'beach head' before a '800 pound gorilla' emerges
- Leverage existing customer relationships
  - IT service organizations like IBM Global Services have extensive customer relationships; provide anxiety-free migration path to existing customers
- Become a platform
  - Example: Facebook's initiative to enable plug-in applications is a great fit for cloud computing

Recap: Different Types of Cloud Services



- **IaaS**: OS layer, provides basic computational infrastructure
- **PaaS**: middleware layer. Provides a set of services to developers to build their applications
- **SaaS**: application layer. Provides applications to final users

IaaS PaaS SaaS Comparison



Packaged Software

You manage: Applications, Data, Runtime, Middleware, O/S, Virtualization, Servers, Storage, Networking

Infrastructure (as a Service)

You manage: Applications, Data, Runtime, Middleware, O/S

Managed by vendor: Virtualization, Servers, Storage, Networking

Platform (as a Service)

You manage: Applications, Data

Managed by vendor: Runtime, Middleware, O/S, Virtualization, Servers, Storage, Networking

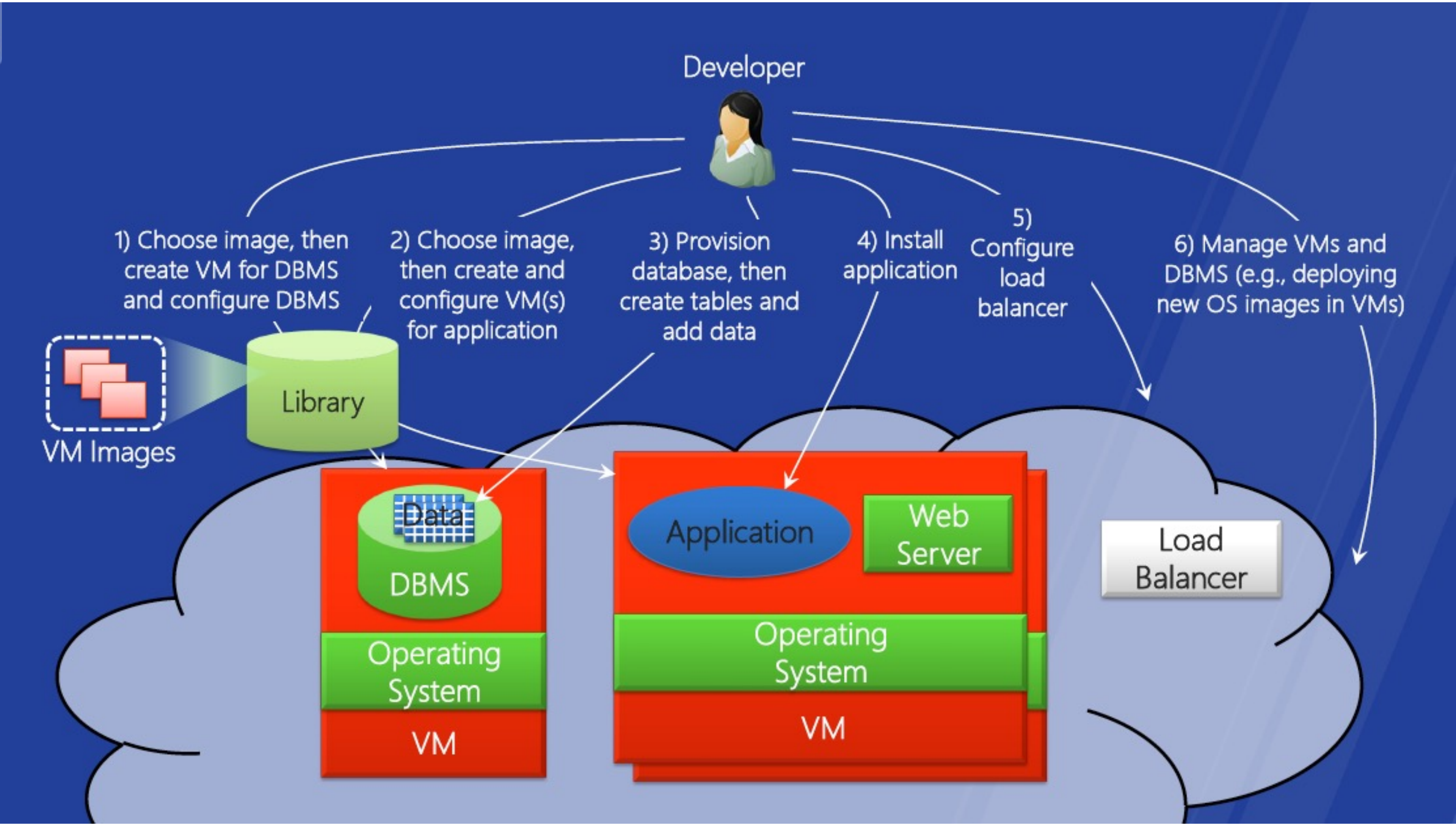
Software (as a Service)

Managed by vendor: Applications, Data, Runtime, Middleware, O/S, Virtualization, Servers, Storage, Networking

Recap: Examples of *aaS

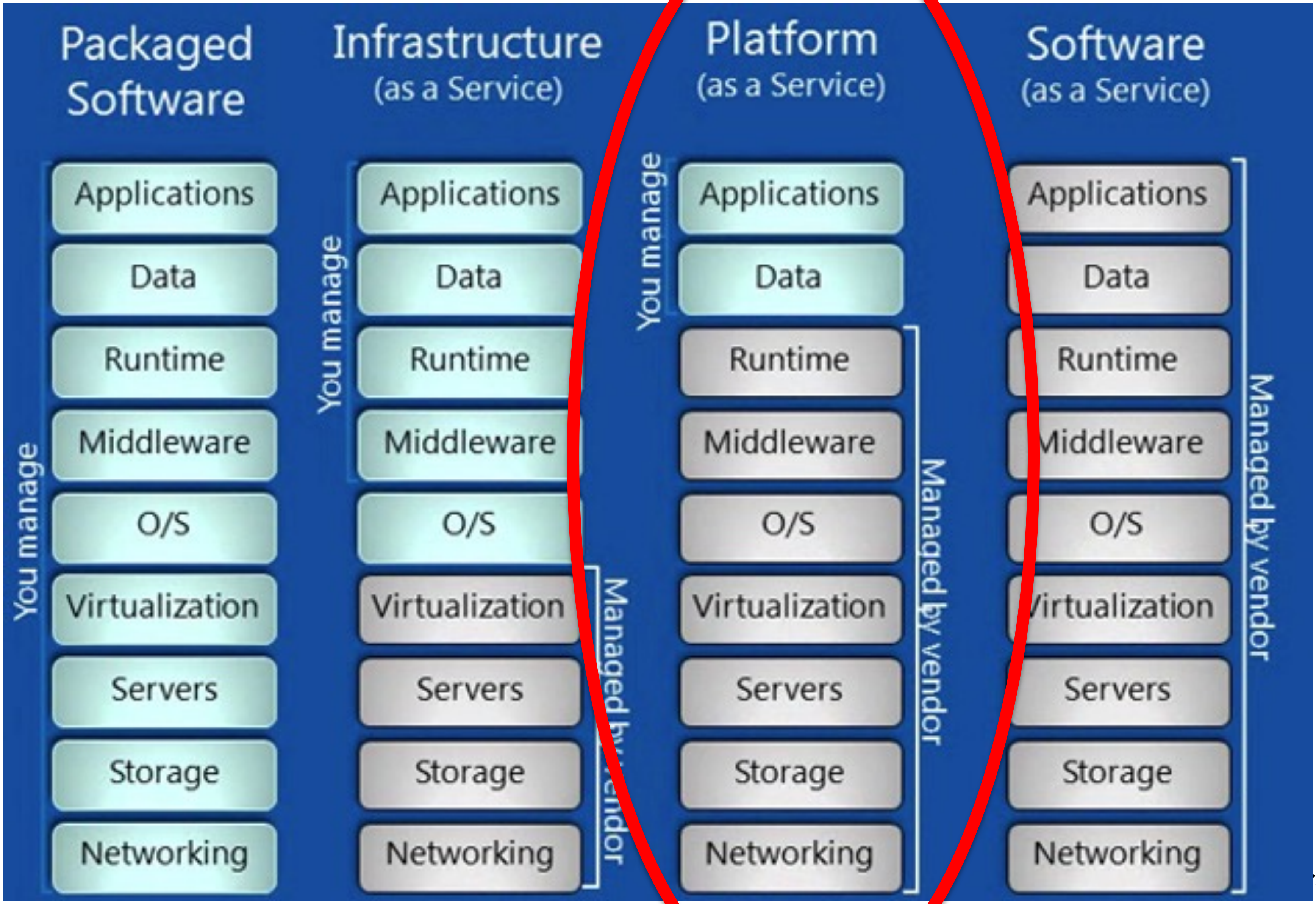
- Infrastructure as a Service (IaaS): basic compute and storage resources
  - On-demand servers
  - Amazon EC2, VMWare vCloud
- Platform as a Service (PaaS): cloud application infrastructure
  - On-demand application-hosting environment
  - E.g. Google AppEngine, Salesforce.com, Windows Azure, Amazon
- Software as a Service (SaaS): cloud applications
  - On-demand applications
  - E.g. GMail, Microsoft Office Web Companions

Example of IaaS: Using the IaaS from AWS



Developer
1) Choose image, then create VM for DBMS and configure DBMS
2) Choose image, then create and configure VM(s) for application
3) Provision database, then create tables and add data
4) Install application
5) Configure load balancer
6) Manage VMs and DBMS (e.g., deploying new OS images in VMs)
VM Images
Library
Data
DBMS
Operating System
VM
Application
Web Server
Operating System
VM
Load Balancer

IaaS PaaS SaaS Comparison



Packaged Software

You manage:
- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

Infrastructure (as a Service)

You manage:
- Applications
- Data
- Runtime
- Middleware
- O/S

Managed by vendor:
- Virtualization
- Servers
- Storage
- Networking

Platform (as a Service)

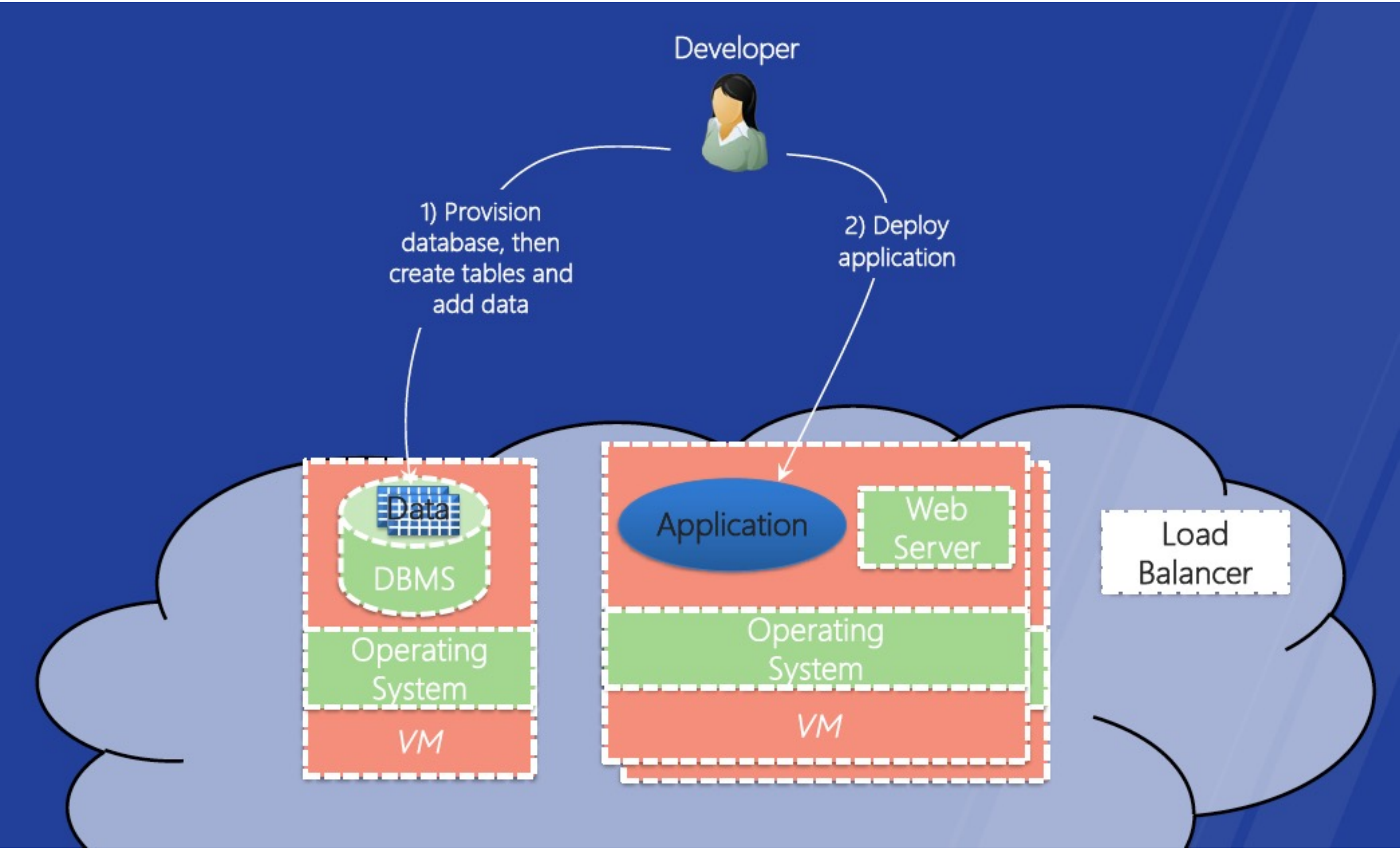
You manage:
- Applications
- Data

Managed by vendor:
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

Software (as a Service)

Managed by vendor:
- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

PaaS


Developer
1) Provision database, then create tables and add data
2) Deploy application
Data
DBMS
Operating System
VM
Application
Web Server
Operating System
VM
Load Balancer

Platform as a Service(PaaS)

- -**PaaS** is a cloud computing service that offers a platform for users to run applications onto the cloud

- -It is a level above Infrastructure as a service(**IaaS**) because unlike IaaS, Paas does not require users to develop their own operating system environment

Platform as a Service (PaaS)

- Middle ground between SaaS and IaaS
- Development platform
  - Customers use to develop applications that benefit from the scalability of the cloud without fully developing their own solution using an IaaS provider
- Offers an application development platform that will automatically scale with demand

Platform as a Service (PaaS)

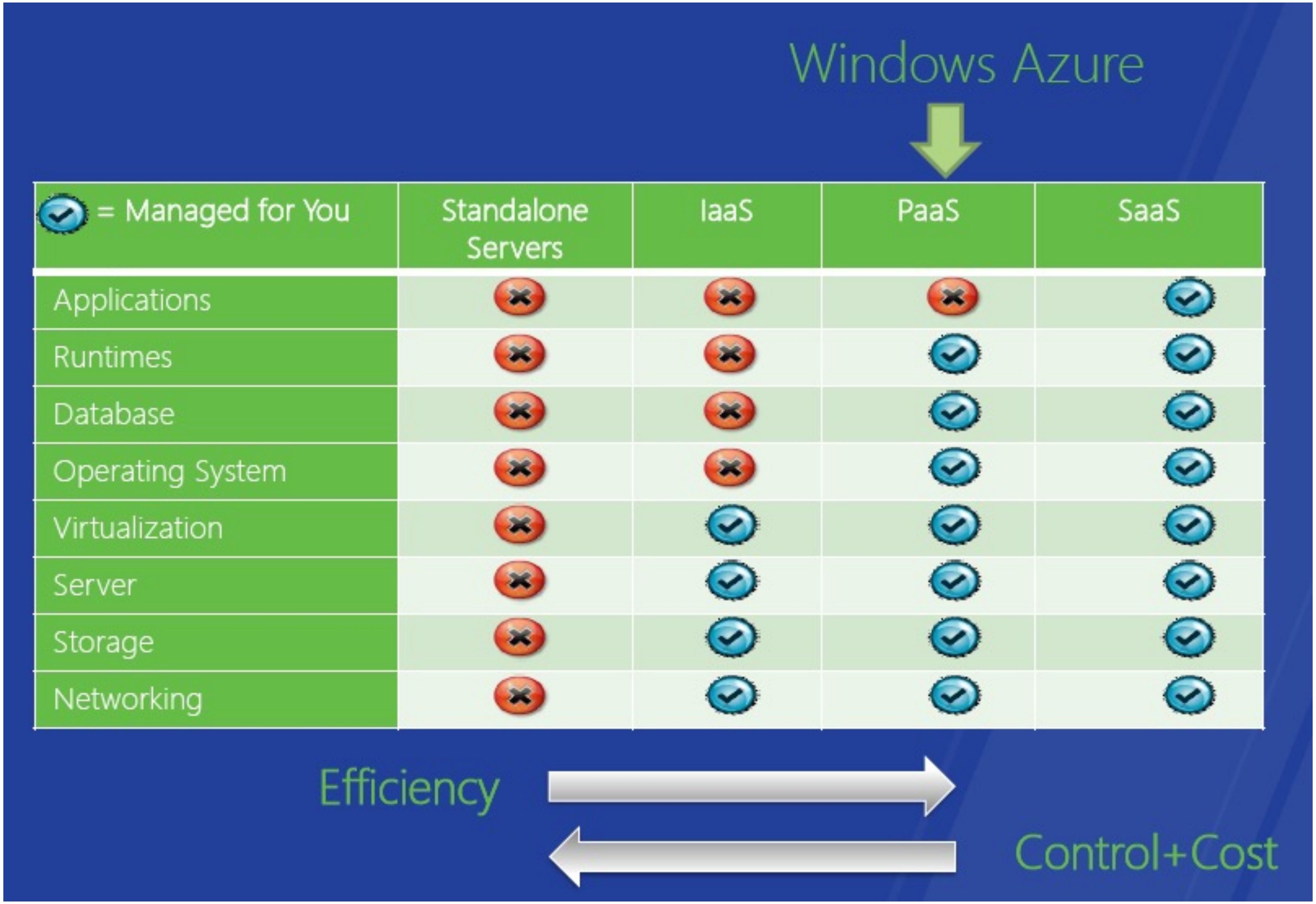
- **Official definition of PaaS from NIST standard**

“The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.”

PaaS Example: Windows Azure PaaS

- Platform as a Service
  - Application Platform in the Cloud
- Provides:
  - *Compute*
    - Web, Worker & VM Role
  - *Storage*
    - *Blob, Table, Queue & Azure SQL Server*
  - Application *Fabric*
    - *Service Bus, Access Control, (Future: Cache, Integration & Composite)*

Cloud Services Type: Efficiency Vs. Control



Windows Azure

✓ = Managed for You	Standalone Servers	IaaS	PaaS	SaaS
Applications	✗	✗	✗	✓
Runtimes	✗	✗	✓	✓
Database	✗	✗	✓	✓
Operating System	✗	✗	✓	✓
Virtualization	✗	✓	✓	✓
Server	✗	✓	✓	✓
Storage	✗	✓	✓	✓
Networking	✗	✓	✓	✓

Efficiency

Control+Cost

More Cost Effective

- PaaS can be better for costs than IaaS, as systems are optimized to run applications efficiently

- IaaS may only provide hardware and thus clients have to be in charge of load balancing and networking

Multi-Tenancy

- PaaS is better suited for **multi-tenancy** as the PaaS provider optimizes their infrastructure for use by many providers

- Multi-tenancy means that many users may share the same physical computer and database

Multi-Tenancy

- PaaS is better suited for multi-tenancy than an IaaS because an IaaS may provide each user with their own virtual machine and create a clear separation of resources
- However, in a PaaS, users may share the same machine, database, etc.

Vendor Lockin

- PaaS may lock in applications by requiring users to develop apps using proprietary interfaces and languages

- This means that it may be difficult for users to go to another vendor to host their app

- Businesses may risk their future on the dependability of the PaaS

Development Tools

- Many PaaS offer Browser-based development tools
- In this way, developers can create their own applications online
- Ease of deployment, the platform takes care of the scaling for you

Principles of Software Development

- As a developer, your objective is to create an application in the quickest, most effective way possible

- One should not create applications using convoluted methods that may take a long time to complete

- The user only sees the end product, not the development process.

PaaS vs. Iaas

- You need to make decisions with long-term consequences, when you use cloud

- If you choose to use a PaaS and get your application vendor locked in, then your business may fail if the PaaS greatly increases their prices

- You will not be able to move to another cloud since your app cannot be easily migrated somewhere else

PaaS vs. Iaas

- An app that is used to fulfill a temporary need, may be handled by a PaaS solution
- An app that may need to be deployed quickly, may be faster developed by a PaaS
- If your software team is small, it may be better to develop on a PaaS and let the PaaS provider handle the OS and networking for your team

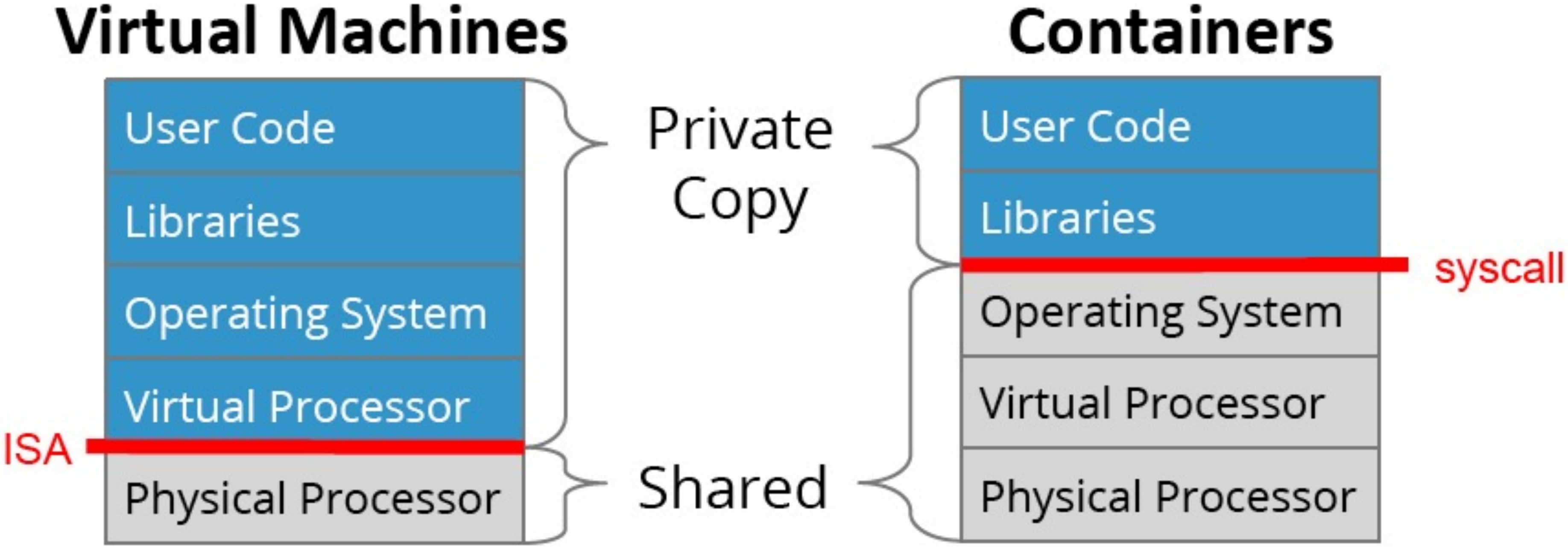
PaaS vs. Iaas

- An app that must serve a variety of purposes for the long term may be better developed on IaaS
- If you need flexibility to change dev tools, languages then an IaaS may be better
- A large software development team may have the resources to optimize and monitor an IaaS system

But,… IaaS, i.e.
Deploying Cloud-based Applications/ Services in form of (groups of) VMs may be too costly !

Container Technologies to our rescue !

VMs vs. Containers



Containers: less overhead, enable more “magic”

Virtual Machine vs. Container

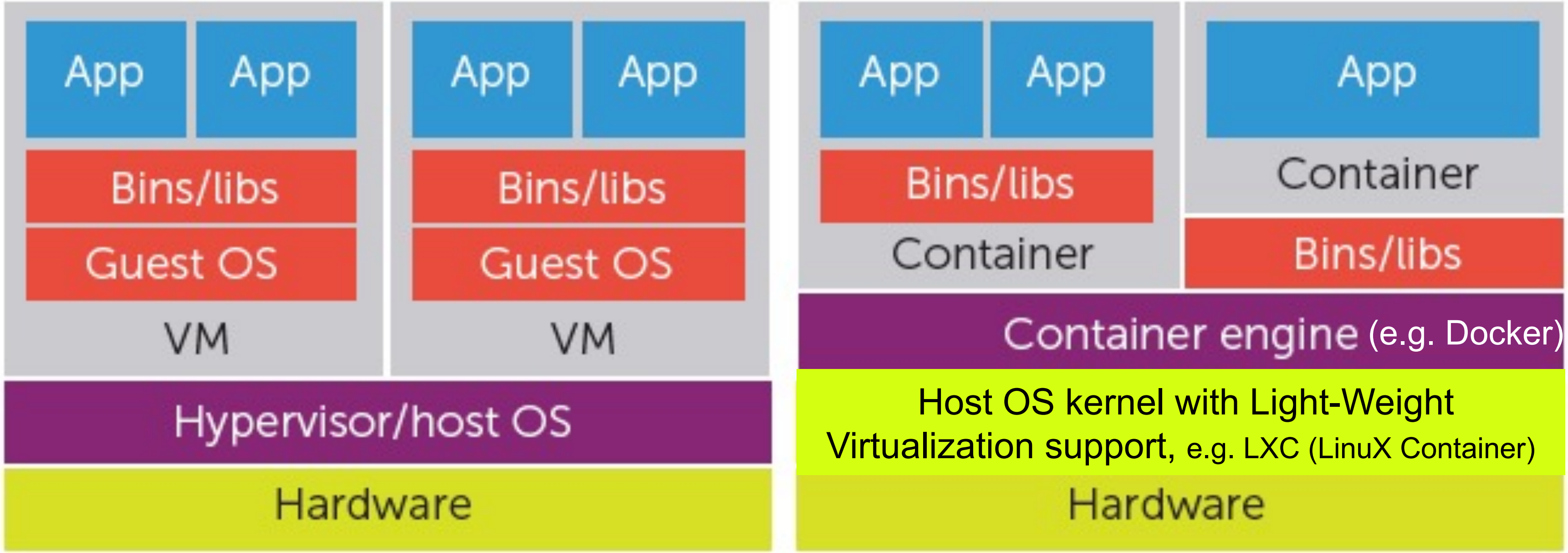
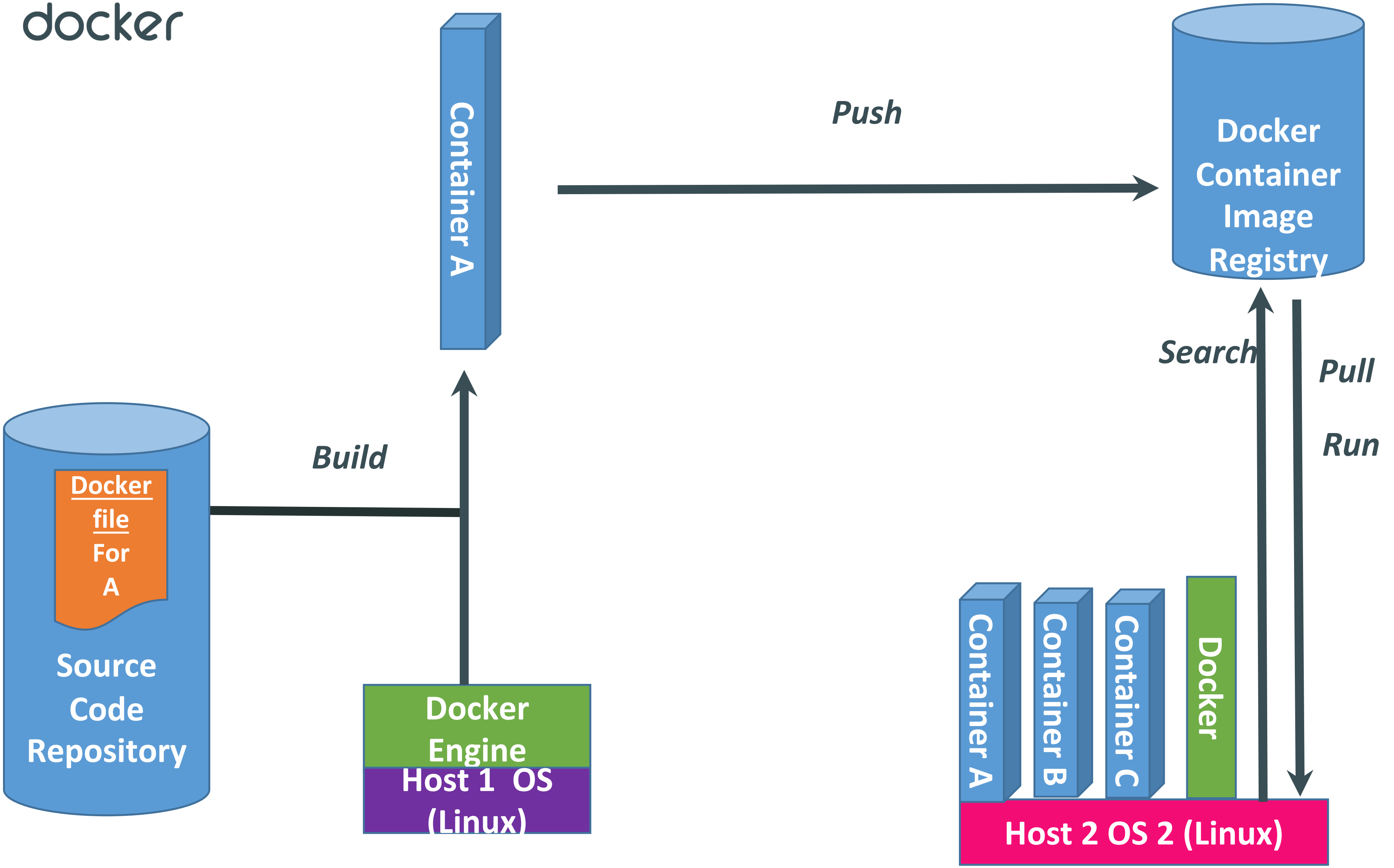


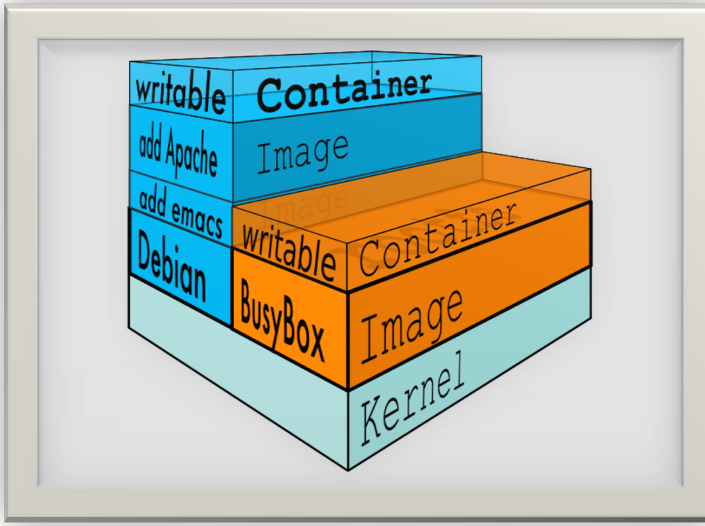
FIGURE 1. Virtualization architecture. The two possible scenarios, a traditional hypervisor architecture on the left and a container-based architecture on the right, differ in their management of guest operating system components.

Source: Claus Pahl, "Containerization and the PaaS Cloud," IEEE Cloud Computing Magazine, May/June 2015

Basic Operations of a Docker system



Docker Containers

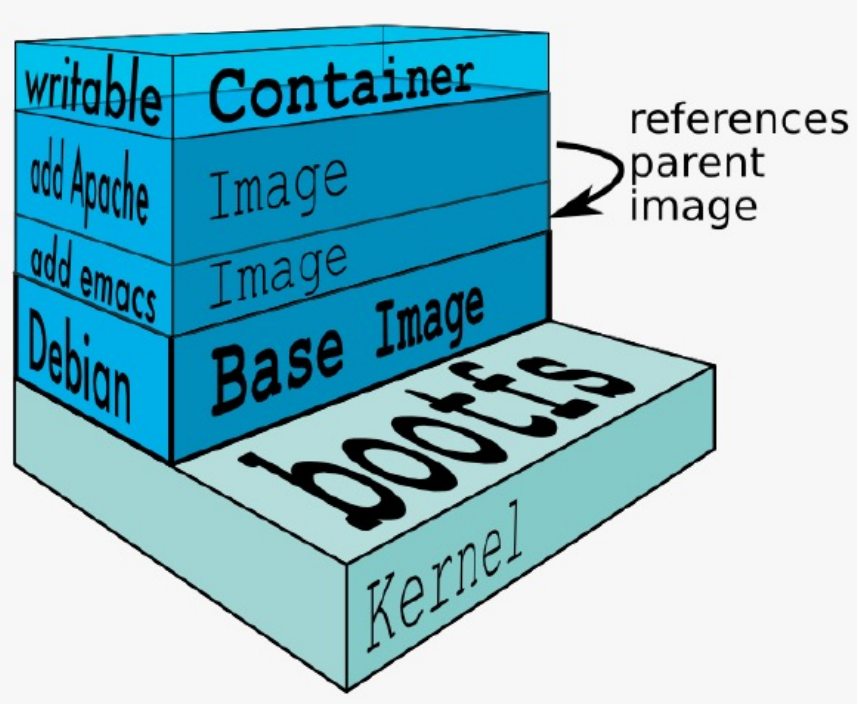


- Units of software delivery (ship it!)
  - run everywhere
    - regardless of kernel version
    - regardless of host distribution
    - (but container and host architecture must match*)
  - run anything
    - if it can run on the host, it can run in the container
    - i.e., if it can run on a Linux kernel, it can run

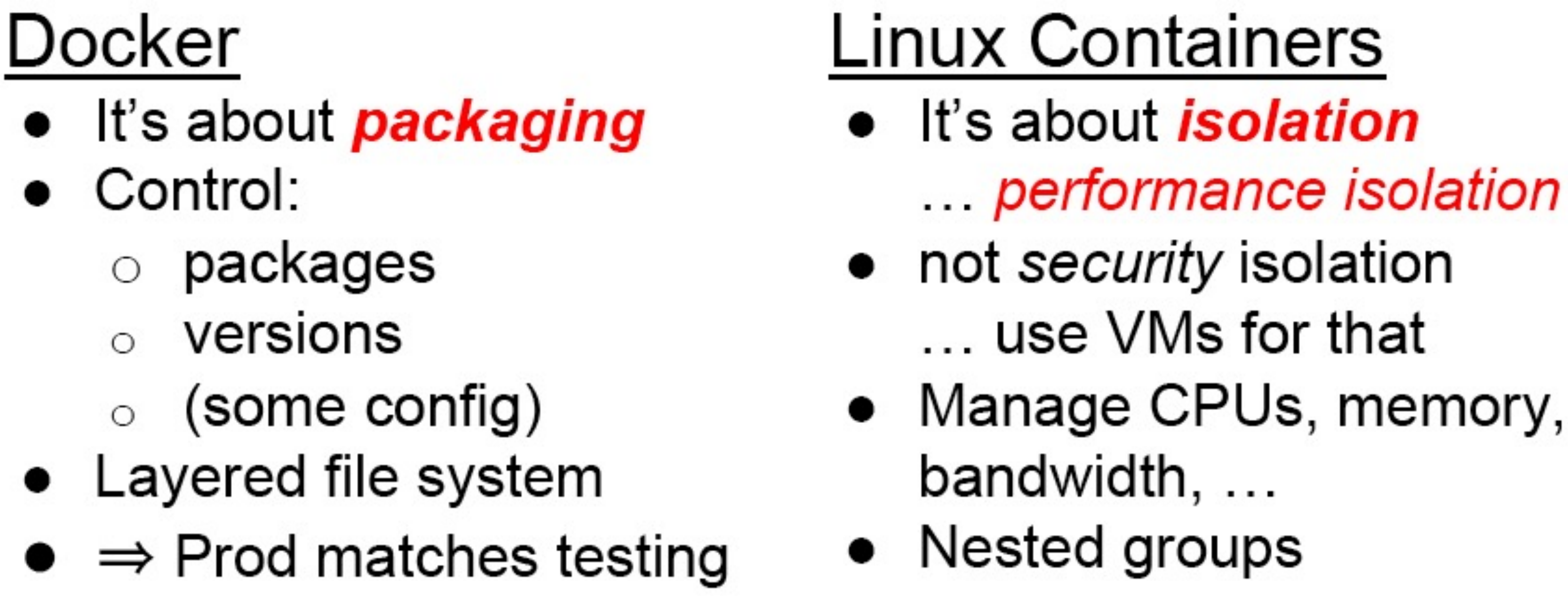
*Unless you emulate CPU with QEMU and binfmt

Docker Image structure

- NOT A Virtual Hark Disk (VHD) file
- NOT A FILESYSTEM
- uses a *Union File System*
- a read-only
- do not have state
- Basically a tar file
- Has a hierarchy
  - Arbitrary depth
- Fits into the Docker Registry



Google's Kubernetes: -
Merging 2 Different Types of Containers



Docker

- It's about ***packaging***
- Control:
  - packages
  - versions
  - (some config)
- Layered file system
- ⇒ Prod matches testing

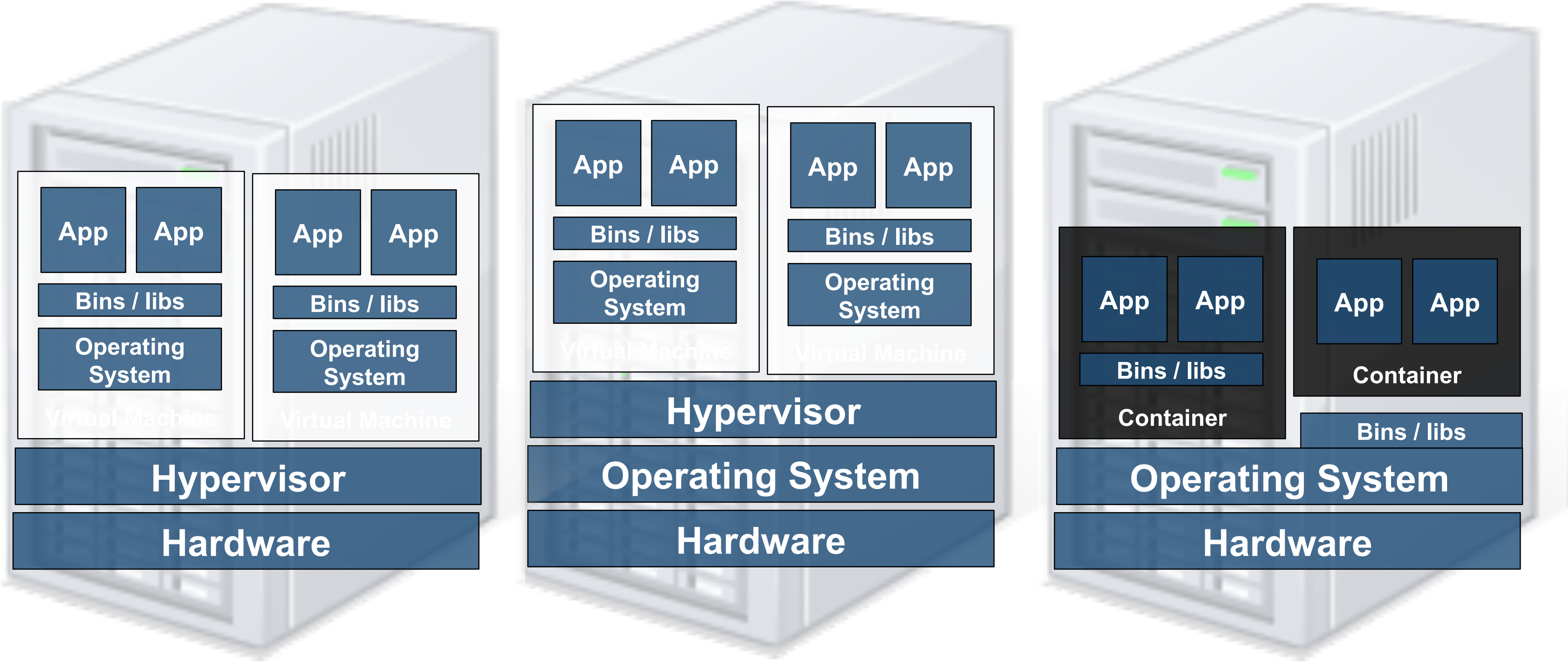
Linux Containers

- It's about ***isolation***
  … *performance isolation*
- not *security* isolation
  … use VMs for that
- Manage CPUs, memory, bandwidth, …
- Nested groups

Hypervisors vs. Linux Containers

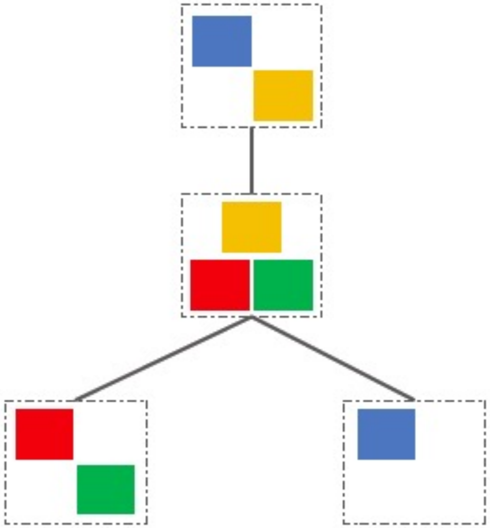
Conceptual Mapping

VM → Container
Hypervisor → LXC Engine



Kubernetes – Google's path towards "Cloud-native" Applications

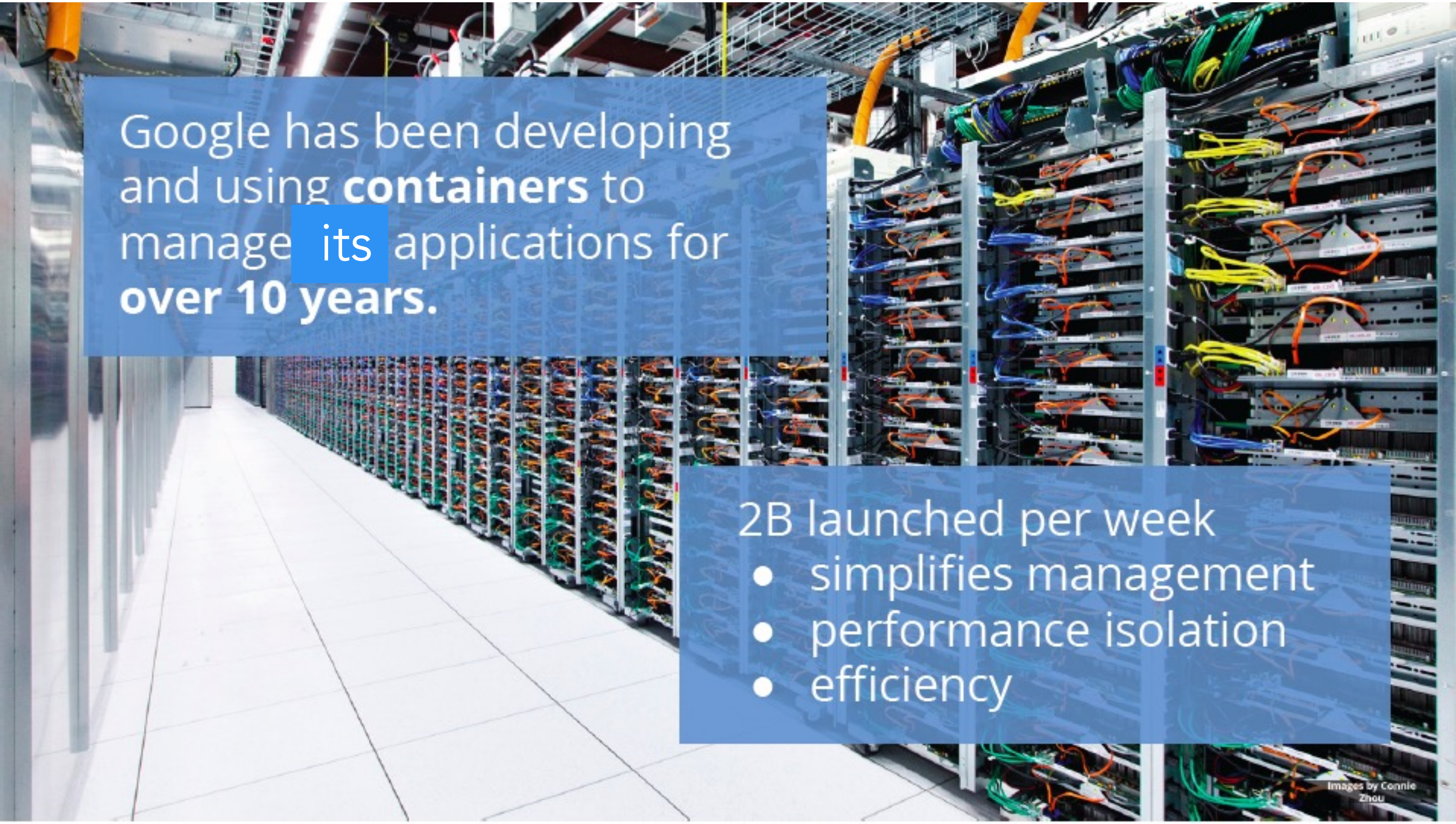
- Kubernetes serves as a distributed platform for hosting containers in a clustered environment
  - Provide orchestration of containers: container grouping, scheduling, load-balancing, auto-healing, scaling, service-discovery functions, etc .
- Apps structured as Independent (micro)services
  - Encapsulated states with APIs, like "Objects"
  - Mix of Programming Languages
  - Mix of Teams



Don't think of a container as the boundary of your application

"A container is more like a class in an object-oriented language."

--- Google's Brendan Burns



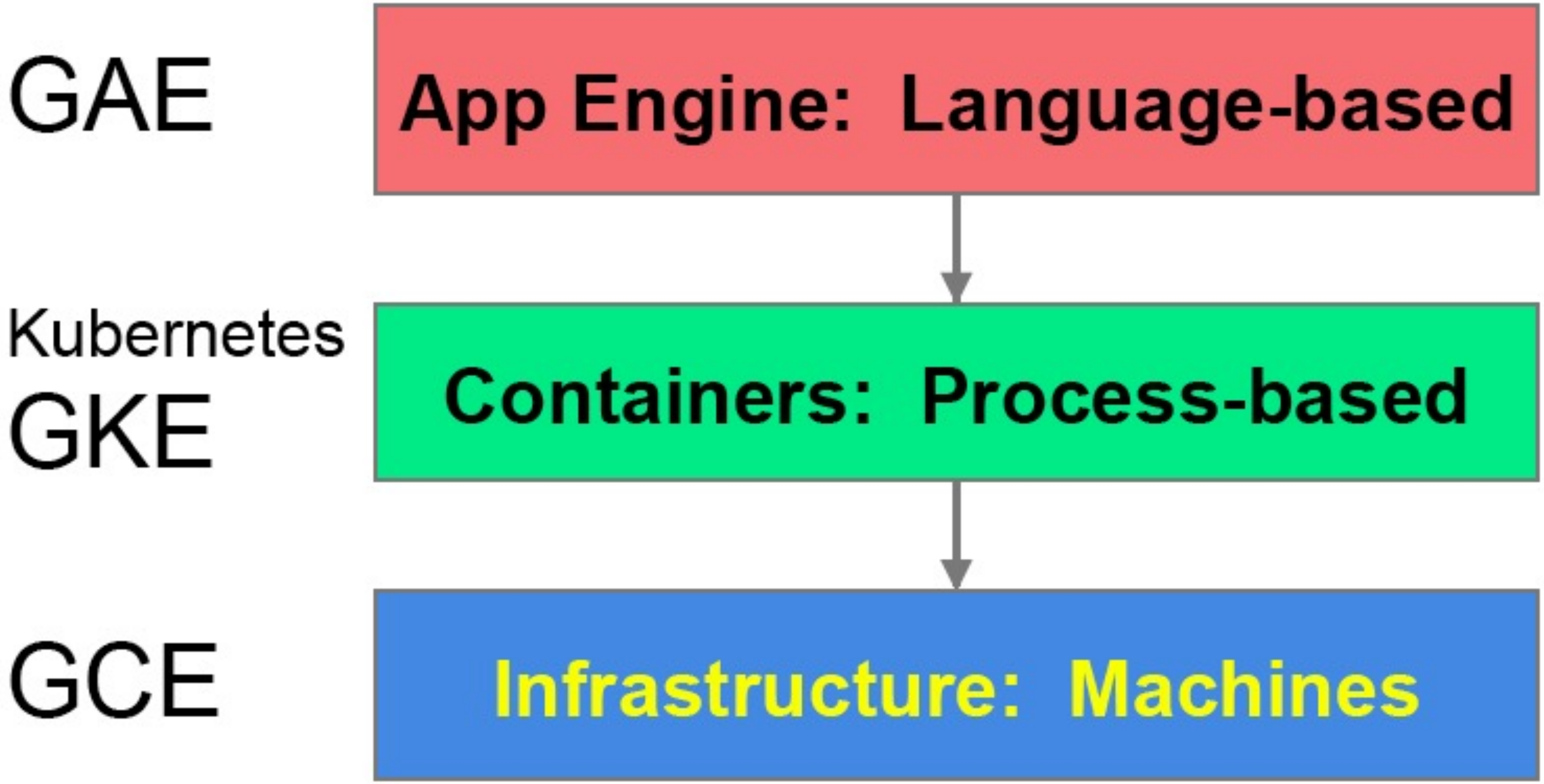
Google has been developing and using **containers** to manage its applications for **over 10 years.**

2B launched per week

- simplifies management
- performance isolation
- efficiency

Images by Connie Zhou

Different forms of Cloud-based Computing Services/Offerings from Google



GAE

App Engine: Language-based

Kubernetes

GKE

Containers: Process-based

GCE

Infrastructure: Machines

History of Public Cloud Services: The pioneers

- Jul 2002: Amazon Web Services launched
  - Third-party sites can search and display products from Amazon's web site, add items to Amazon shopping carts
  - Available through XML and SOAP
- Mar 2006: Amazon S3 launched
  - Innovative 'pay-per-use' pricing model, which is now the standard in cloud computing
  - Cheaper than many small/medium storage solutions: $0.15/GB/month of storage, $0.20/GB/month for traffic
  - Amazon no longer a pure retailer, entering technology space
- Aug 2006: EC2 launched
  - Core computing infrastructure becomes available

Other Cloud Service providers

- **Windows Azure**
  - Similar services now, was pushing the Platform-as-a service model (PaaS)
- **Rackspace**
  - Infrastructure-as-a-service, powered by OpenStack (opensource clone of EC2/S3)
- **Google App Engine, Google Compute Cloud, Google Apps**
  - Google App Engine is a Platform-as-a-Service
  - Google Compute Engine is an IaaS;
  - Google Apps is a Software-as-a-Service ;
  - Most recently (Aug 2015) Google Container Engine (GKC)
    - run Docker “Containers” over Google Cloud Platform, using Kubernetes for “orchestration”
    - CaaS – close to an Open PaaS on Linux Platform

DOCKER on Google Cloud

Kubernetes

κυβερνήτης: *Greek for "pilot" or "helmsman of a ship"*
the open source cluster manager from Google

History: Wide-spread adoption

- Apr 2008: Google App Engine launched
  - Same building blocks Google uses for its own applications: Bigtable and GFS for storage, automatic scaling and load balancing, ...
- Nov 2009: Windows Azure Beta launched
  - Becomes generally available in 21 countries in Feb 2010
- 2013: Windows Azure IaaS and Google Compute Engine (IaaS) available to the public !
- Aug 2015: Google Container Engine (GKC) – container-based computing, official product launch:
  - Based on Open-source Kubernetes container management system from Red-hat, Docker, IBM, OpenStack, VMWare, Mesosphere, Cisco, Intel
  - Forming the Cloud Native Computing Foundation: https://cncf.io

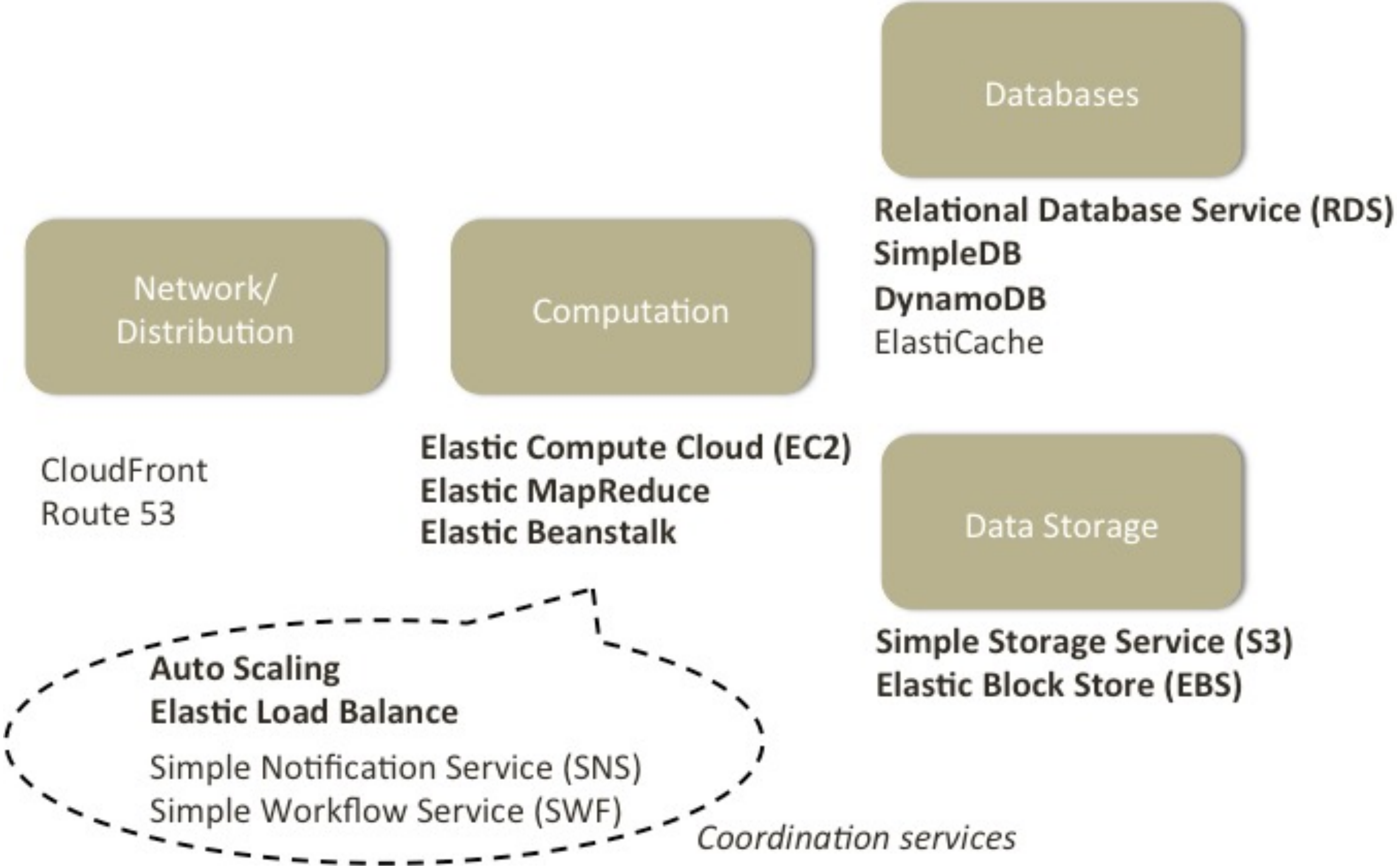
Amazon Web Services (AWS) Overview

What is Amazon Web Services (AWS) ?

- AWS provides a collection of services for building cloud applications
- Services for:
  - **Storage**: S3, EBS
  - **Computation**: Elastic Cloud Computing (EC2), scaling/load balancer, Elastic Map/Reduce, Elastic Beanstalk
  - **Databases**: RDS, DynamoDB, ElastiCache
  - **Coordination**: Simple Notification Service, Simple Workflow Framework
  - Content delivery network
  - Amazon CloudFront
  - Amazon Mechanical Turk (MTurk)
    A 'marketplace for work’
  - …
- All services are paid depending on use

http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=irol-corporateTimeline

Overview of AWS Services



Databases

Relational Database Service (RDS)
SimpleDB
DynamoDB
ElastiCache

Network/ Distribution

CloudFront
Route 53

Computation

Elastic Compute Cloud (EC2)
Elastic MapReduce
Elastic Beanstalk

Data Storage

Simple Storage Service (S3)
Elastic Block Store (EBS)

Auto Scaling
Elastic Load Balance
Simple Notification Service (SNS)
Simple Workflow Service (SWF)

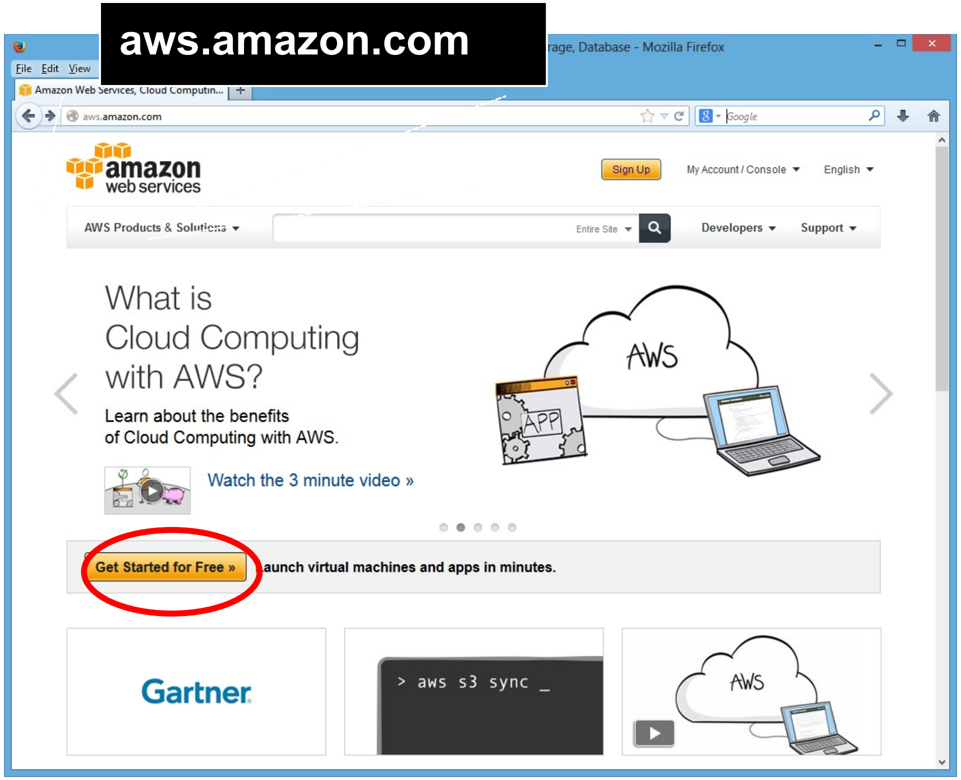
Coordination services

Using AWS Services

- AWS Management Console
  - Easy to use, great for manual configurations
  - Use **username** / **password** provided
- Command line tools
  - For writing scripts
    - e.g., create a set of machines to analyze data every day
  - Use **access key ID** and **secret access key,** or **certificates for EC2**
- AWS API
  - Integrating cloud services into your applications
    - e.g., storing data on the cloud, running computation in the background
  - Use **access key ID** and **secret access key, or certificates for EC2**
- SSH into EC2 instances is performed using a different keypair

Setting up an AWS account

aws.amazon.com

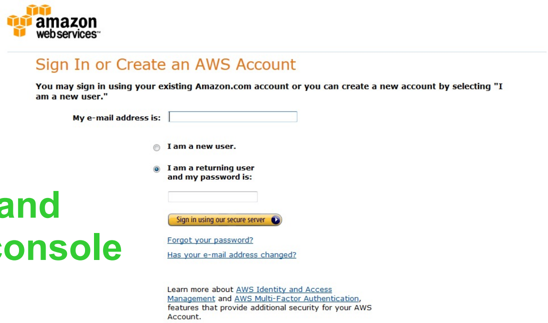


Amazon Web Services, Cloud Computin...
aws.amazon.com
rage, Database - Mozilla Firefox
File Edit View
amazon web services
Sign Up
My Account / Console
English
AWS Products & Solutions
Entire Site
Developers
Support
What is Cloud Computing with AWS?
Learn about the benefits of Cloud Computing with AWS.
Watch the 3 minute video »
Get Started for Free »
Launch virtual machines and apps in minutes.
Gartner
> aws s3 sync _
AWS

- Sign up for an account on aws.amazon.com
  - **You need to choose an username and a password**
  - These are for the management interface only
  - **Your programs will use other credentials (RSA keypairs, access keys, ...) to interact with AWS**

AWS credentials

AWS web site and
management console



Sign In or Create an AWS Account

You may sign in using your existing Amazon.com account or you can create a new account by selecting "I am a new user."

My e-mail address is:

○ I am a new user.

● I am a returning user
and my password is:

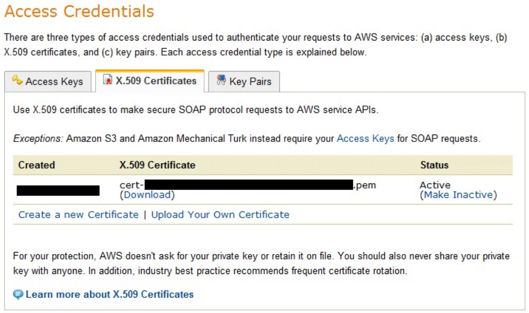
Forgot your password?

Has your e-mail address changed?

Learn more about AWS Identity and Access Management and AWS Multi-Factor Authentication, features that provide additional security for your AWS Account.

Sign-in credentials

Command-line tools
SOAP APIs



Access Credentials

There are three types of access credentials used to authenticate your requests to AWS services: (a) access keys, (b) X.509 certificates, and (c) key pairs. Each access credential type is explained below.

Use X.509 certificates to make secure SOAP protocol requests to AWS service APIs.

Exceptions: Amazon S3 and Amazon Mechanical Turk instead require your Access Keys for SOAP requests.

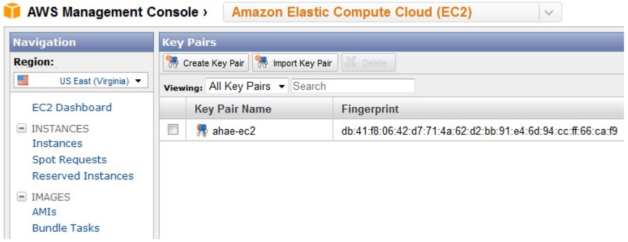
Created	X.509 Certificate	Status
[redacted]	cert-[redacted].pem (Download)	Active (Make Inactive)

Create a new Certificate | Upload Your Own Certificate

For your protection, AWS doesn't ask for your private key or retain it on file. You should also never share your private key with anyone. In addition, industry best practice recommends frequent certificate rotation.

Learn more about X.509 Certificates

X.509 certificates

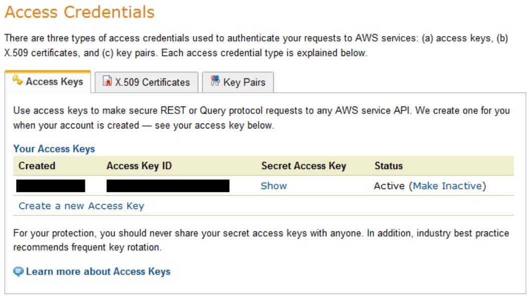


Viewing: All Key Pairs

□	Key Pair Name	Fingerprint
□	ahae-ec2	db:41:f8:06:42:d7:71:4a:62:d2:bb:91:e4:6d:94:cc:ff:66:ca:f9

Connecting to an
instance (e.g., via ssh)

EC2 key pairs



Access Credentials

There are three types of access credentials used to authenticate your requests to AWS services: (a) access keys, (b) X.509 certificates, and (c) key pairs. Each access credential type is explained below.

Use access keys to make secure REST or Query protocol requests to any AWS service API. We create one for you when your account is created — see your access key below.

Your Access Keys

Created	Access Key ID	Secret Access Key	Status
[redacted]	[redacted]	Show	Active (Make Inactive)

Create a new Access Key

For your protection, you should never share your secret access keys with anyone. In addition, industry best practice recommends frequent key rotation.

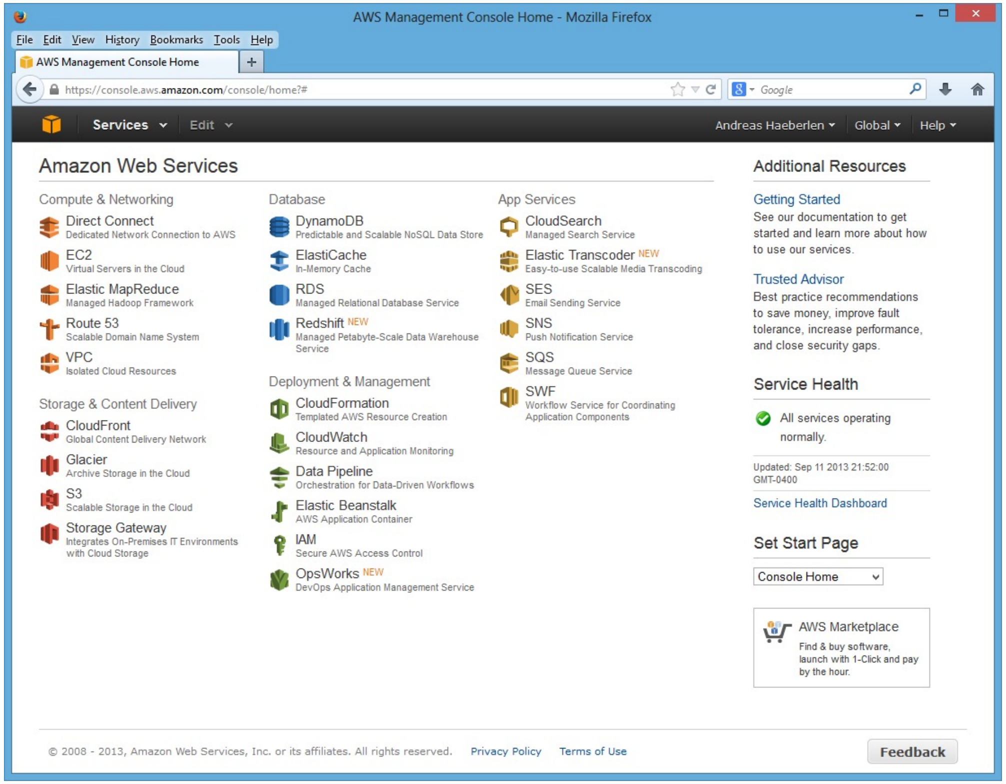
Learn more about Access Keys

Access keys

REST APIs

- Why so many different types of credentials?

The AWS management console


AWS Management Console Home - Mozilla Firefox
File Edit View History Bookmarks Tools Help
AWS Management Console Home
https://console.aws.amazon.com/console/home?#
Google
Services
Edit
Andreas Haeberlen
Global
Help

Amazon Web Services

Compute & Networking

- Direct Connect – Dedicated Network Connection to AWS
- EC2 – Virtual Servers in the Cloud
- Elastic MapReduce – Managed Hadoop Framework
- Route 53 – Scalable Domain Name System
- VPC – Isolated Cloud Resources

Storage & Content Delivery

- CloudFront – Global Content Delivery Network
- Glacier – Archive Storage in the Cloud
- S3 – Scalable Storage in the Cloud
- Storage Gateway – Integrates On-Premises IT Environments with Cloud Storage

Database

- DynamoDB – Predictable and Scalable NoSQL Data Store
- ElastiCache – In-Memory Cache
- RDS – Managed Relational Database Service
- Redshift NEW – Managed Petabyte-Scale Data Warehouse Service

Deployment & Management

- CloudFormation – Templated AWS Resource Creation
- CloudWatch – Resource and Application Monitoring
- Data Pipeline – Orchestration for Data-Driven Workflows
- Elastic Beanstalk – AWS Application Container
- IAM – Secure AWS Access Control
- OpsWorks NEW – DevOps Application Management Service

App Services

- CloudSearch – Managed Search Service
- Elastic Transcoder NEW – Easy-to-use Scalable Media Transcoding
- SES – Email Sending Service
- SNS – Push Notification Service
- SQS – Message Queue Service
- SWF – Workflow Service for Coordinating Application Components

Additional Resources

Getting Started
See our documentation to get started and learn more about how to use our services.

Trusted Advisor
Best practice recommendations to save money, improve fault tolerance, increase performance, and close security gaps.

Service Health

All services operating normally.

Updated: Sep 11 2013 21:52:00 GMT-0400

Service Health Dashboard

Set Start Page

Console Home

AWS Marketplace
Find & buy software, launch with 1-Click and pay by the hour.

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Used to control many AWS services:
  - For example, start/stop EC2 instances, create S3 buckets...

REST and SOAP

- How do your programs access AWS?
  - Via the REST or SOAP protocols
  - Example: Launch an EC2 instance, store a value in S3, ...
- Simple Object Access protocol (SOAP)
  - Not as simple as the name suggests
  - XML-based, extensible, general, standardized, but also somewhat heavyweight and verbose
  - Increasingly deprecated (e.g., for SimpleDB and EC2)
- Representational State Transfer (REST)
  - Much simpler to develop than SOAP
  - Web-specific; lack of standards

Example: REST

Invoked method

Response elements

Parameters

Credentials

```
https://sdb.amazonaws.com/?Action=PutAttributes
&DomainName=MyDomain
&ItemName=Item123
&Attribute.1.Name=Color&Attribute.1.Value=Blue
&Attribute.2.Name=Size&Attribute.2.Value=Med
&Attribute.3.Name=Price&Attribute.3.Value=0014.99
&AWSAccessKeyId=<valid_access_key>
&Version=2009-04-15
&Signature=[valid signature]
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2010-01-25T15   3A01%3A28-07%3A00
```

```
<PutAttributesResponse>
<ResponseMetadata>
<StatusCode>Success</StatusCode>
<RequestId>f6820318-9658-4a9d-89f8-
b067c90904fc</RequestId>
<BoxUsage>0.0000219907</BoxUsage
>
</ResponseMetadata>
</PutAttributesResponse>
```

Sample request

Sample response

Source: http://awsdocs.s3.amazonaws.com/SDB/latest/sdb-dg.pdf

Example: SOAP

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
xmlns:SOAP-
ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:SOAP-
ENC='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
<SOAP-ENV:Body>
<PutAttributesRequest
xmlns='http://sdb.amazonaws.com/doc/
2009-04-15'>
<Attribute><Name>a1</Name><Value>2</Value></Attribute>
<Attribute><Name>a2</Name><Value>4</Value></Attribute>
<DomainName>domain1</DomainName>
<ItemName>eID001</ItemName>
<Version>2009-04-15</Version>
</PutAttributesRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

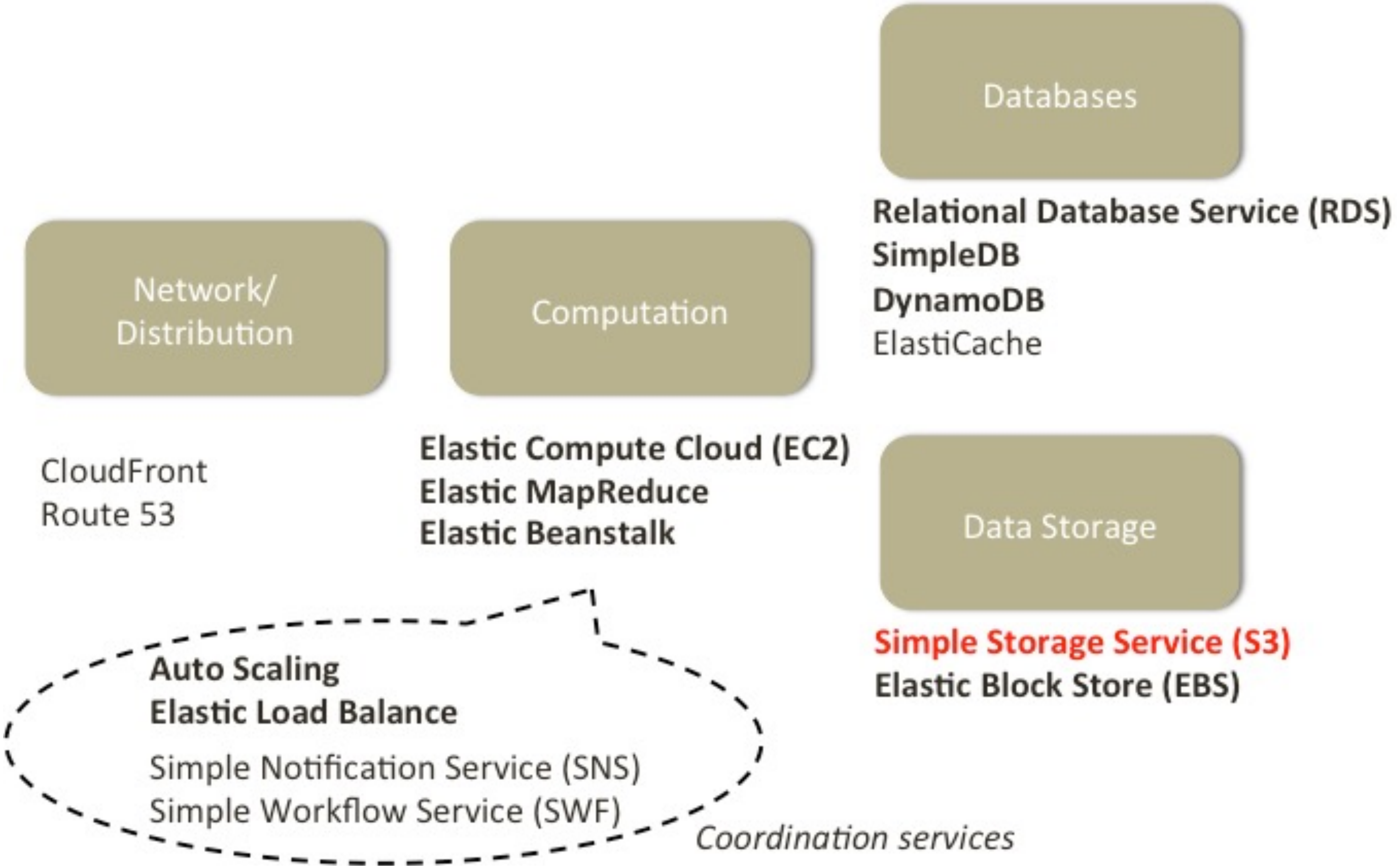
Sample request

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/env
elope/">
<SOAP-ENV:Body>
<PutAttributesResponse>
<ResponseMetadata>
<RequestId>4c68e051-fe45-43b2-992a-
a24017ffe7ab</RequestId>
<BoxUsage>0.0000219907</BoxUsage>
</ResponseMetadata>
</PutAttributesResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample response

Source: http://awsdocs.s3.amazonaws.com/SDB/latest/sdb-dg.pdf

Overview of Services



Network/ Distribution

CloudFront
Route 53

Computation

Elastic Compute Cloud (EC2)
Elastic MapReduce
Elastic Beanstalk

Databases

Relational Database Service (RDS)
SimpleDB
DynamoDB
ElastiCache

Data Storage

Simple Storage Service (S3)
Elastic Block Store (EBS)

Auto Scaling
Elastic Load Balance
Simple Notification Service (SNS)
Simple Workflow Service (SWF)

Coordination services

Simple Storage Service (S3)

- First publicly available web service from Amazon (2006)
- Unstructured storage of large amount of data with high reliability
  - Automatically replicated across multiple datacenters
  - write, read, delete data objects
- Storage pay-as-you-store model
  - $0.095 a gigabyte + per-request charges + network bandwidth
- Stores data objects into buckets, each data object is up to 5T
- Data can be read and written through a programming API
  - You can use S3 in your applications as a data storage layer
- Relaxed Eventual Consistency Model
  - If you PUT to an existing key, a subsequent read might return the old data or the updated data, but it will never write corrupted or partial data.

Demo: https://console.aws.amazon.com/s3/home?region=us-east-1

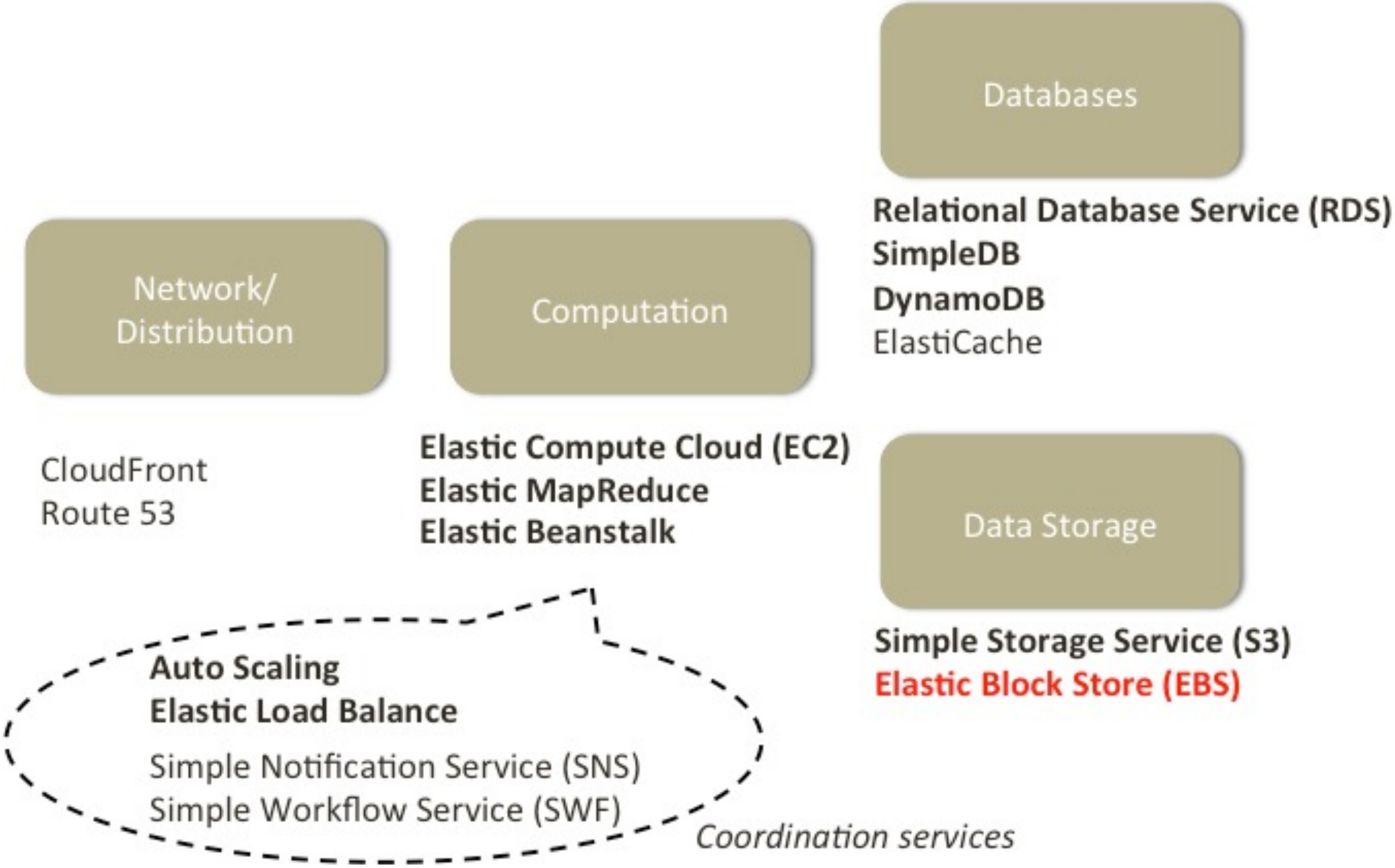
S3 Program Example

- Python + Boto
  - Boto = AWS SDK for Python, now at version 3

```
from boto.s3.connection import S3Connection
conn = S3Connection(AWS_KEY, AWS_SECRET)
bucket = conn.get_bucket(BUCKET)
destination = bucket.new_key()
destination.name = filename
destination.set_contents_from_file(myfile)
destination.make_public()
```

https://s3.amazonaws.com/cs498cc-mmontan2/greetings.html

Overview of AWS Services



Network/ Distribution

CloudFront
Route 53

Computation

Elastic Compute Cloud (EC2)
Elastic MapReduce
Elastic Beanstalk

Databases

Relational Database Service (RDS)
SimpleDB
DynamoDB
ElastiCache

Data Storage

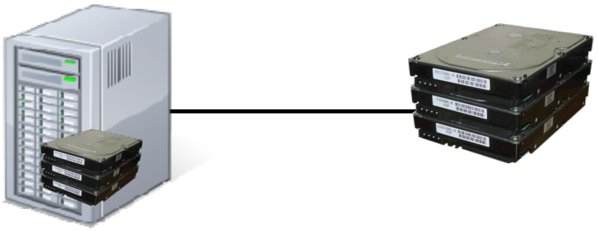
Simple Storage Service (S3)
Elastic Block Store (EBS)

Auto Scaling
Elastic Load Balance
Simple Notification Service (SNS)
Simple Workflow Service (SWF)

Coordination services

What is Amazon Elastic Block Store (EBS) ?

- **"Cloud-based virtual hard drives"**



AMZ EC2 Instance **EBS storage**

- Block level storage volumes for use with Amazon EC2 instances (EC2 instances = virtual machines, to be discussed next)
- Persistent, Off-instance Storage, persists independently from the life of an instance
  - Unlike the local instance store, data stored in EBS is not lost when an instance fails or is terminated

Should I use the instance store or EBS?

Typically, instance store is used for temporary data

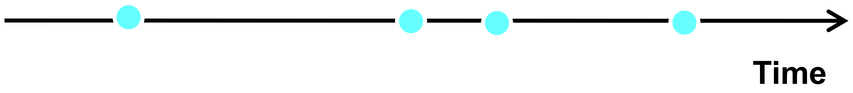
EBS Volumes

- EBS storage is allocated in volumes
  - A volume is a 'virtual disk' (size: 1GB - 1TB)
  - Basically, a raw block device
  - Can be attached to an instance (but only one at a time)
  - Can attach multiple volumes to a single instance and stripe across the volumes (RAID0) to achieve further increases in throughput.
- Placed in specific availability zones
  - Why is this useful?
  - Be sure to place it near instances (otherwise can't attach)
- Replicated across multiple servers
  - Data is not lost if a single server fails
  - Amazon: Annual failure rate is 0.1-0.5% for a 20GB volume

Elastic Block Store (EBS) cont’d

- Amazon CloudWatch exposes performance metrics for EBS volumes, giving insight into bandwidth, throughput, latency, …
- EBS can be (incrementally) backed up on S3
- Higher throughput than Amazon EC2 instance stores for applications performing a lot of random accesses

Snapshots



- You can create a snapshot of a volume
  - Copy of data in the volume at the time snapshot was made
  - Only the first snapshot makes a full copy; subsequent snapshots are incremental
- What are snapshots good for?
  - Sharing data with others
    - DBpedia snapshot ID is "snap-882a8ae3"
    - Access control list (specific account numbers) or public access
  - Instantiate new volumes
  - Point-in-time backups

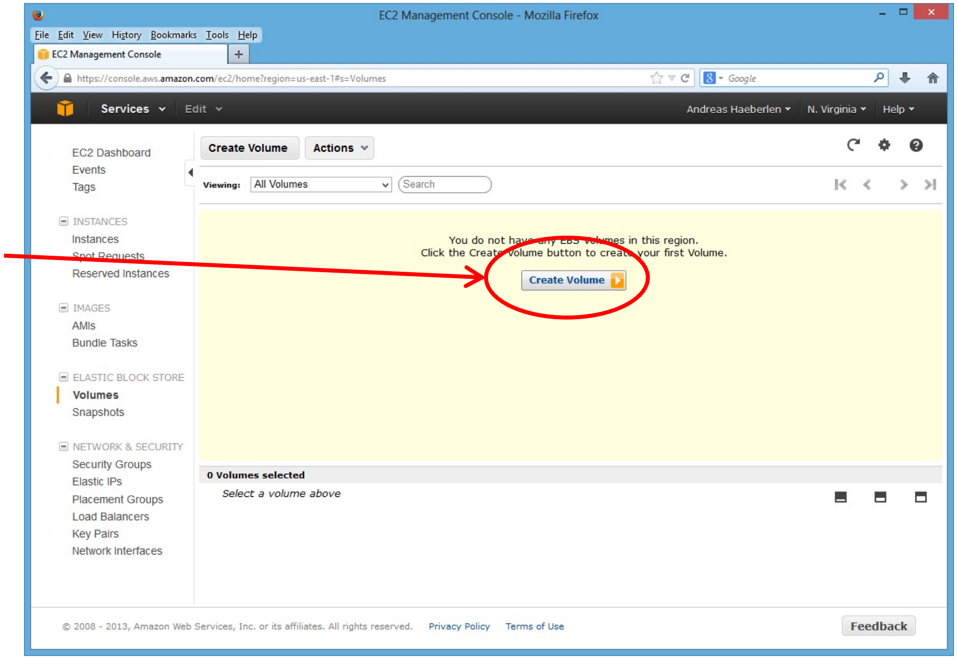
Pricing

- You pay for...
  - Storage space: e.g. $0.10 per allocated GB per month
  - I/O requests: e.g. $0.10 per million I/O requests
  - S3 operations (GET/PUT)

- Charge is only for actual storage used
  - Empty space does not count

Creating an EBS volume

Create volume



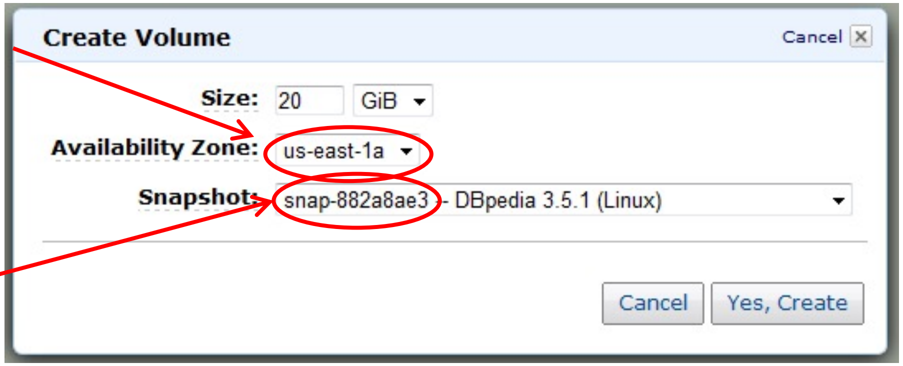
You do not have any EBS volumes in this region.
Click the Create Volume button to create your first Volume.

0 Volumes selected

Select a volume above

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Needs to be in same availability zone as your instance!



Size: 20 GiB

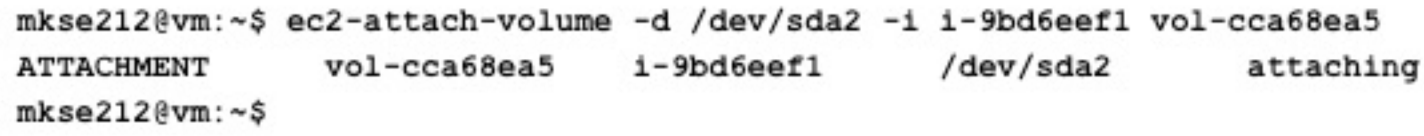
Availability Zone: us-east-1a

Snapshot: snap-882a8ae3 - DBpedia 3.5.1 (Linux)

DBpedia snapshot ID

Mounting an EBS volume

- Step 1: Attach the volume



```
mkse212@vm:~$ ec2-attach-volume -d /dev/sda2 -i i-9bd6eef1 vol-cca68ea5
ATTACHMENT      vol-cca68ea5    i-9bd6eef1      /dev/sda2       attaching
mkse212@vm:~$
```

- Step 2: Mount the volume in the instance

```
mkse212@vm:~$ ssh ec2-user@ec2-50-17-64-130.compute-1.amazonaws.com

       __|  __|_  )  Amazon Linux AMI
       _|  (     /     Beta
      ___|\___|___|

See /usr/share/doc/system-release-2011.02 for latest release notes. :-)
[ec2-user@ip-10-196-82-65 ~]$ sudo mount /dev/sda2 /mnt/
[ec2-user@ip-10-196-82-65 ~]$ ls /mnt/
dbpedia_3.5.1.owl  dbpedia_3.5.1.owl.bz2  en  other_languages
[ec2-user@ip-10-196-82-65 ~]$
```

Detaching an EBS volume

- Step 1: Unmount the volume in the instance

```
[ec2-user@ip-10-196-82-65 ~]$ sudo umount /mnt/
[ec2-user@ip-10-196-82-65 ~]$ exit
mkse212@vm:~$
```

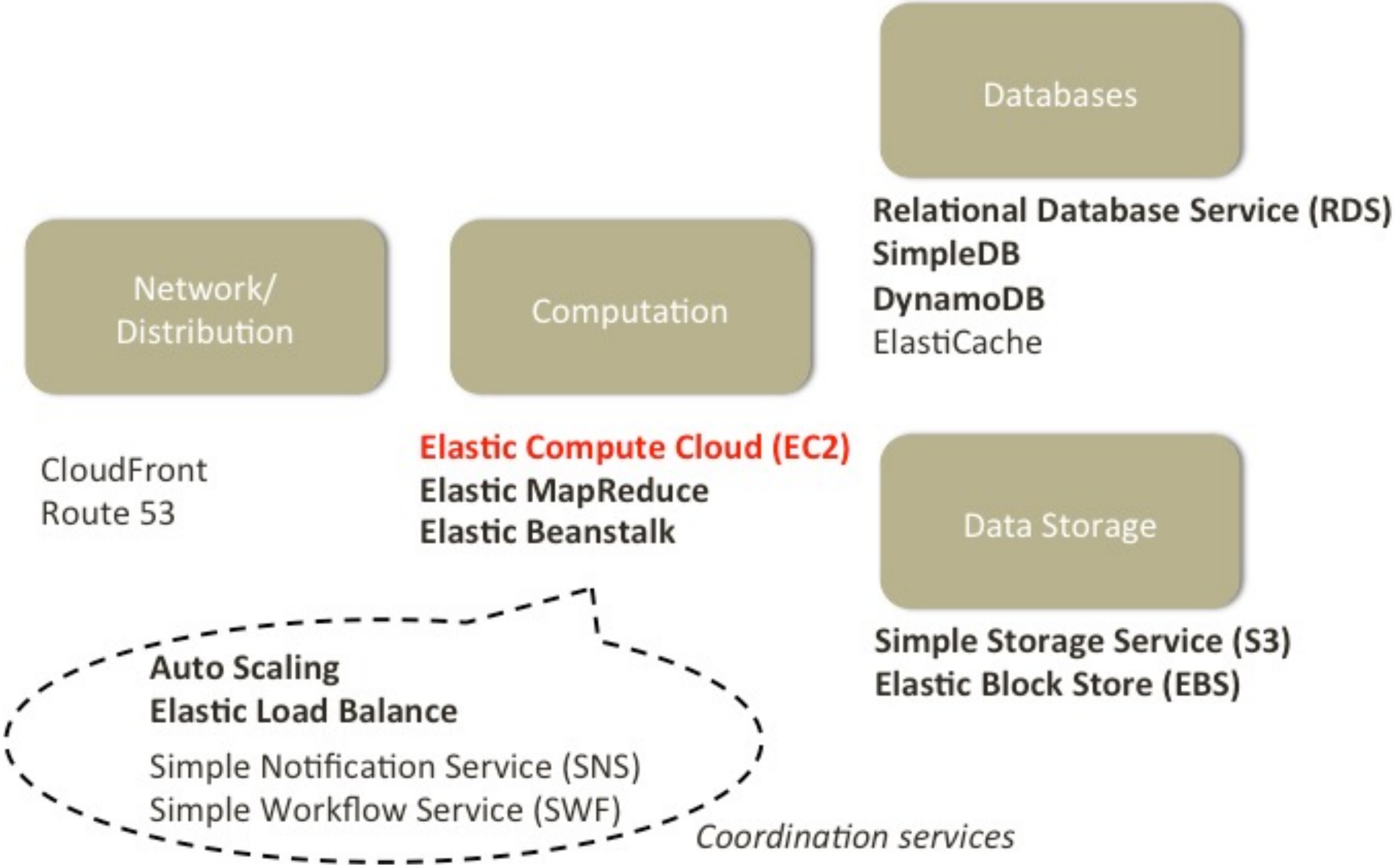
- Step 2: Detach the volume

```
mkse212@vm:~$ ec2-detach-volume vol-cca68ea5
ATTACHMENT      vol-cca68ea5    i-9bd6eef1      /dev/sda2       detaching
mkse212@vm:~$
```

Summary of Elastic Block Store (EBS)

- What EBS is:
  - Basically a virtual hard disk; can be attached to EC2 instances
  - Persistent - state survives termination of EC2 instance
- How to use EBS:
  - Allocate volume - empty or initialized with a snapshot
  - Attach it to EC2 instance and mount it there
  - Can create snapshots for data sharing, backup

Overview of AWS Services



Databases

Relational Database Service (RDS)
SimpleDB
DynamoDB
ElastiCache

Network/ Distribution

CloudFront
Route 53

Computation

Elastic Compute Cloud (EC2)
Elastic MapReduce
Elastic Beanstalk

Data Storage

Simple Storage Service (S3)
Elastic Block Store (EBS)

Auto Scaling
Elastic Load Balance

Simple Notification Service (SNS)
Simple Workflow Service (SWF)

Coordination services

What is Amazon Elastic Cloud Computing (EC2) ?

- Launched in 2006 for providing resizable compute capacity in the cloud
- Rent virtual computers on demand
    - Pay for what you use
        - Hourly rates, e.g., $0.065 an hour for a “small” instance
    - Create new instances within tens of seconds
    - Increase the computing capacity of an instance in minutes
- Most startups and several large organizations use EC2 for running their servers

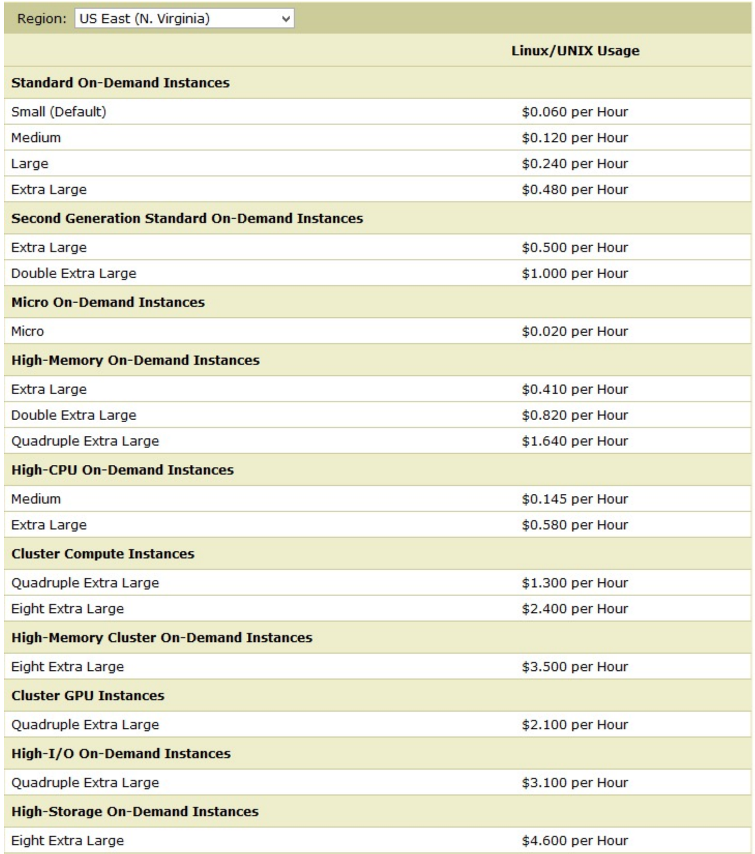
NETFLIX

foursquare

SHAZAM

Pinterest

What is Amazon EC2 ? (cont’d)



Region: US East (N. Virginia)

	Linux/UNIX Usage
Standard On-Demand Instances	
Small (Default)	$0.060 per Hour
Medium	$0.120 per Hour
Large	$0.240 per Hour
Extra Large	$0.480 per Hour
Second Generation Standard On-Demand Instances	
Extra Large	$0.500 per Hour
Double Extra Large	$1.000 per Hour
Micro On-Demand Instances	
Micro	$0.020 per Hour
High-Memory On-Demand Instances	
Extra Large	$0.410 per Hour
Double Extra Large	$0.820 per Hour
Quadruple Extra Large	$1.640 per Hour
High-CPU On-Demand Instances	
Medium	$0.145 per Hour
Extra Large	$0.580 per Hour
Cluster Compute Instances	
Quadruple Extra Large	$1.300 per Hour
Eight Extra Large	$2.400 per Hour
High-Memory Cluster On-Demand Instances	
Eight Extra Large	$3.500 per Hour
Cluster GPU Instances	
Quadruple Extra Large	$2.100 per Hour
High-I/O On-Demand Instances	
Quadruple Extra Large	$3.100 per Hour
High-Storage On-Demand Instances	
Eight Extra Large	$4.600 per Hour

1.7 GB memory
1 virtual core
(1 CU each)
160GB storage
'moderate' I/O

68.4 GB memory
8 virtual cores
(3.25 CU each)
1690 GB storage
'high' I/O

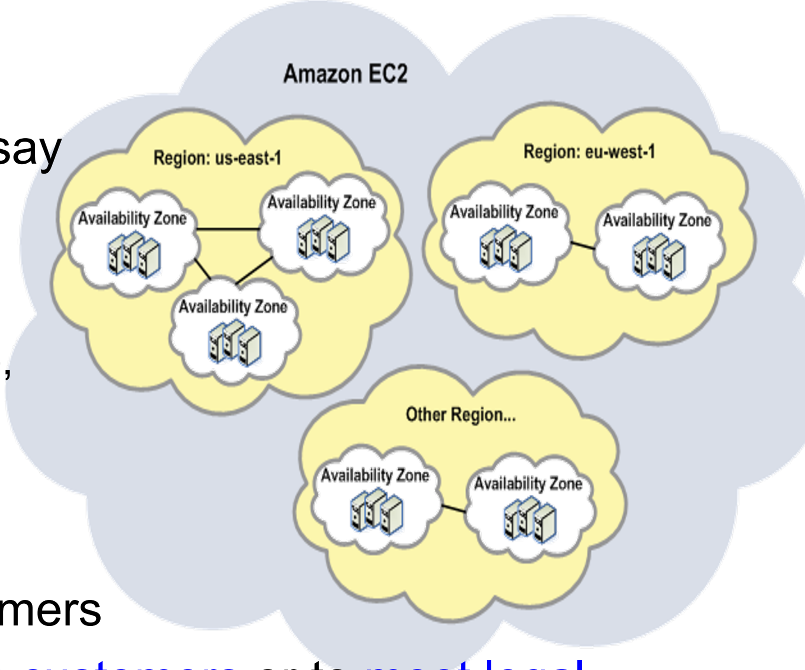
- Infrastructure-as-a-Service (IaaS)
  - You can rent various types of virtual machines by the hour
  - In your VMs, you can run your own (Linux/Windows) programs
    - Examples: Web server, search engine, movie renderer, ...

Amazon Machine Image (AMI)

- Virtual instances boot on a Amazon Machine Image (AMI)
- An image of an operating system ready to boot
  - Amazon Linux; Redhat; Ubuntu Server; Windows;…
  - They might be preconfigured with Apache, Mysql …
  - Anybody can create AMIs and send them to Amazon
    - It's created from a snapshot of the files in a computer
- For the class: use "free tier" enabled images.
  - e.g. Ubuntu Server 12.04.1 LTS

Where are my instances?

Amazon has data centers in different areas of the world (e.g., North America, Europe, Asia, etc.)



- Where exactly does my instance run?
  - No easy way to find out - Amazon does not say
- Instances can be assigned to **Regions**
  - Currently 9 available: US East (Northern Virginia), US West (Northern California), US West (Oregon), EU (Ireland), Asia/Pacific (Singapore), Asia/Pacific (Sydney), Asia/Pacific (Tokyo), South America (Sao Paulo), AWS GovCloud
  - Important, e.g., for reducing latency to customers
  - Design an application to be closer to specific customers or to meet legal or other requirements
- Each Region contains multiple distinct locations called **Availability Zones**
- Availability Zones are isolated from failures in others
- Inexpensive, low-latency network connectivity to other Zones in the same Region
- Launching instances in separate Availability Zones → protect applications from failure in a single location

Create a new instance

- *ec2-run-instances <ami> -k <keypair> --instance-type <type> -z <region-availability zone>*
- Instance Type: Micro (search for free tier)
- **EBS-backed**
  - In EBS-backed, your root disk is on a network storage.
  - Stop and restart maintain the data
    - Depending on settings, termination might delete the EBS volume
- **Instance-backed**
  - Everything is stored on the local disk of the machine
  - Data is lost when the machine is stopped / terminated.
  - Limited in what you can change after boot
  - Excellent for temporary jobs that require a local disk space

Using the Instance

- Once the instance starts, it is your computer
  - Users, configurations, servers, it's all up to the cloud user (you).
  - AMI provides initial configurations, but you can change anything you want
- Accessing the instance:
  - AMI come preconfigured with users
    - "ec2-user" for the Amazon Linux AMIs, "ubutu" for the Ubuntu image
  - At the first boot, Amazon loads a ssh public key that you provide in the user directory so you can log in.
  - After that, you can change anything you want
  - Use ssh to access the instance:
    - ssh -i <privatekey> user@instanceip

Creating the key-pair

- Create your own set of keys for the group
  - You can use the AWS Management Console
    - There are also command line tools and API
  - Put the name of your group in the key pair, so that you know which one you should use

- The private key can be downloaded only once, so put it in a safe place
  - If you lose the private key, you can create a new keypair.
  - However, you might not be able to access instances created with your previous key

EC2 API

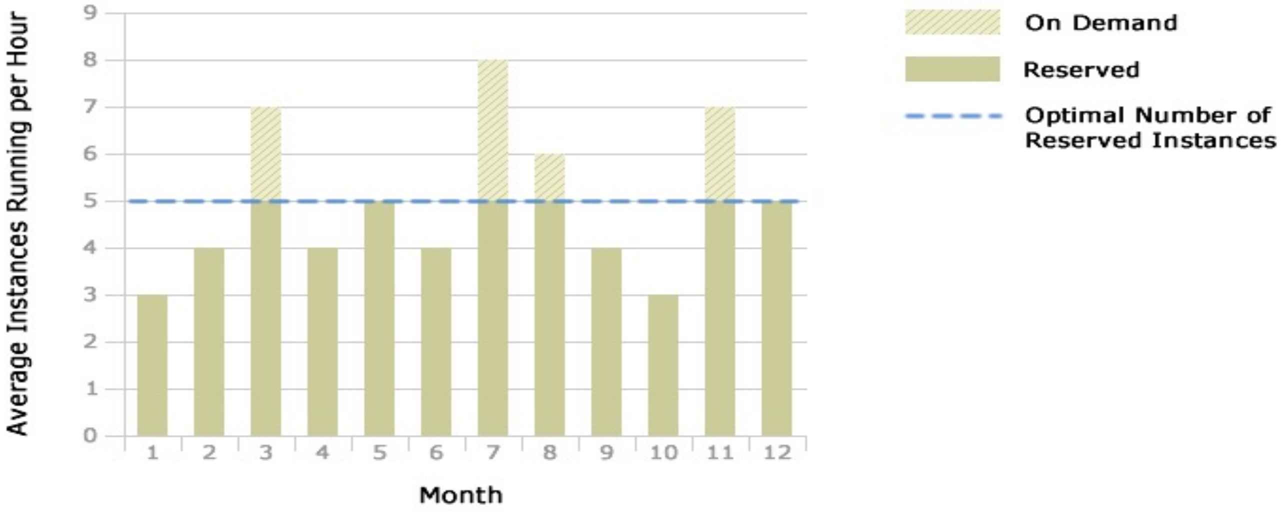
- Python + boto

```
from boto.ec2.connection import EC2Connection
conn = EC2Connection('<AWS_ACCESS_KEY_ID>',
'<AWS_SECRET_ACCESS_KEY>')

conn.run_instances(
      '<ami-image-id>',
      key_name='myKey',
      instance_type='c1.xlarge',
      security_groups=['your-security-group-here'])
```

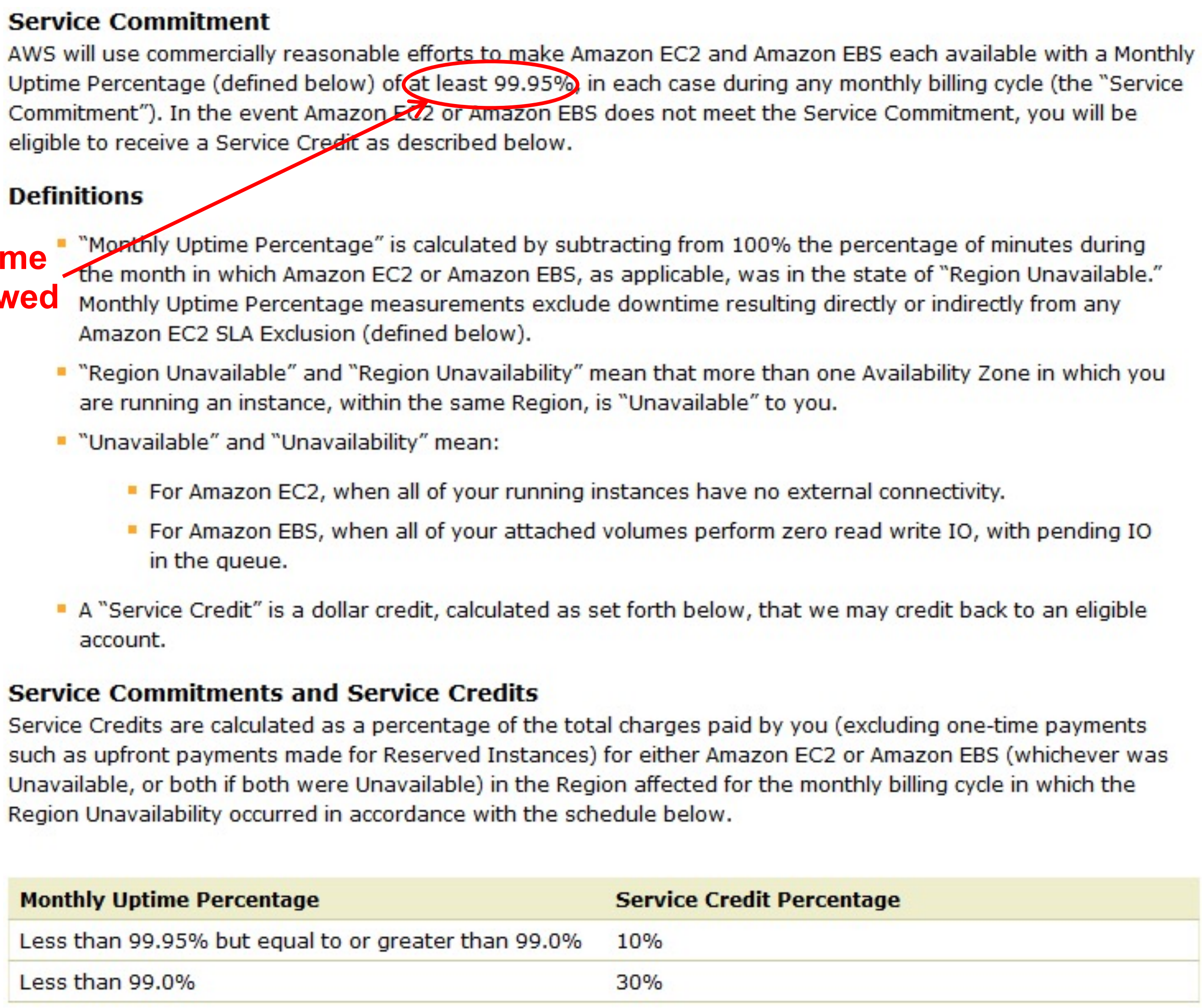
- **Java interface is a more syntactically complex, but semantically it's the same**

Instance types



- So far: On-demand instances
- Also available: Reserved instances
  - One-time reservation fee to purchase for 1 or 3 years
  - Usage still billed by the hour, but at a considerable discount
- Also available: Spot instances
  - Spot market: Can bid for available capacity
  - Instance continues until terminated or price rises above bid

Service Level Agreement



Service Commitment

AWS will use commercially reasonable efforts to make Amazon EC2 and Amazon EBS each available with a Monthly Uptime Percentage (defined below) of at least 99.95%, in each case during any monthly billing cycle (the "Service Commitment"). In the event Amazon EC2 or Amazon EBS does not meet the Service Commitment, you will be eligible to receive a Service Credit as described below.

4.38h downtime per year allowed

Definitions

- "Monthly Uptime Percentage" is calculated by subtracting from 100% the percentage of minutes during the month in which Amazon EC2 or Amazon EBS, as applicable, was in the state of "Region Unavailable." Monthly Uptime Percentage measurements exclude downtime resulting directly or indirectly from any Amazon EC2 SLA Exclusion (defined below).
- "Region Unavailable" and "Region Unavailability" mean that more than one Availability Zone in which you are running an instance, within the same Region, is "Unavailable" to you.
- "Unavailable" and "Unavailability" mean:
  - For Amazon EC2, when all of your running instances have no external connectivity.
  - For Amazon EBS, when all of your attached volumes perform zero read write IO, with pending IO in the queue.
- A "Service Credit" is a dollar credit, calculated as set forth below, that we may credit back to an eligible account.

Service Commitments and Service Credits

Service Credits are calculated as a percentage of the total charges paid by you (excluding one-time payments such as upfront payments made for Reserved Instances) for either Amazon EC2 or Amazon EBS (whichever was Unavailable, or both if both were Unavailable) in the Region affected for the monthly billing cycle in which the Region Unavailability occurred in accordance with the schedule below.

Monthly Uptime Percentage	Service Credit Percentage
Less than 99.95% but equal to or greater than 99.0%	10%
Less than 99.0%	30%

http://aws.amazon.com/ec2-sla/ (9/11/2013; excerpt

Reminder: Oh no - where has my data gone?

- EC2 instances do not have persistent storage
  - Data survives stops & reboots, but not termination

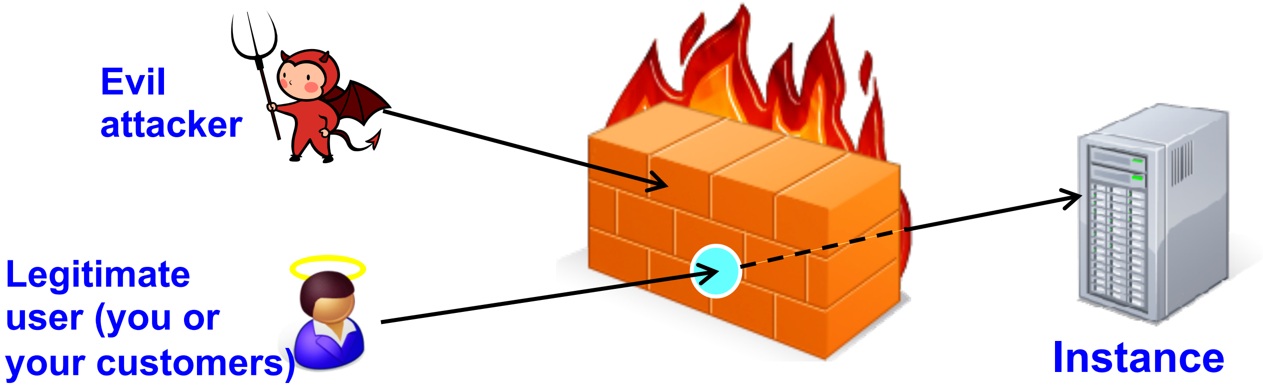
If you store data on the virtual hard disk of your instance
and the instance fails or you terminate it,
your data WILL be lost!

- So where should I put persistent data?
  - Elastic Block Store (EBS)
  - Ideally, use an AMI with an EBS root (Amazon's default AMI has this property)

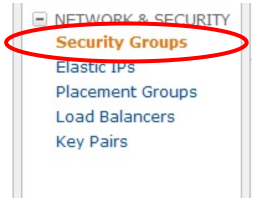
EC2 instances with EBS roots

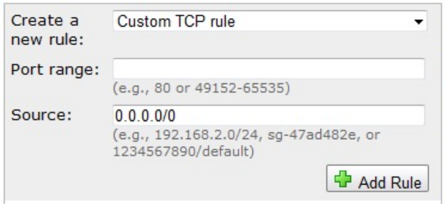
- EC2 instances can have an EBS volume as their root device ("EBS boot")
  - Result: Instance data persists independently from the lifetime of the instance
  - You can stop and restart the instance, similar to suspending and resuming a laptop
    - You won't be charged for the instance while it is stopped (only for EBS)
  - You can enable termination protection for the instance
    - Blocks attempts to terminate the instance (e.g., by accident) until termination protection is disabled again
- Alternative: Use instance store as the root
  - You can still store temporary data on it, but it will disappear when you terminate the instance
  - You can still create and mount EBS volumes explicitly

Security Groups



- Basically, a set of firewall rules
  - Can be applied to groups of EC2 instances
  - Each rule specifies a protocol, port numbers, etc...
  - Only traffic matching one of the rules is allowed through
- Sometimes need to explicitly open ports





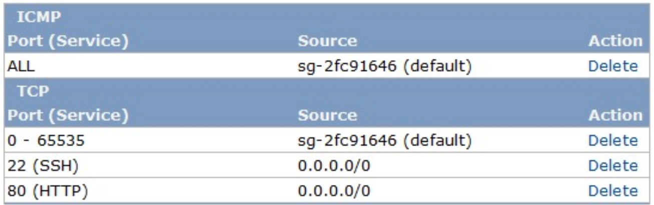
Create a new rule: Custom TCP rule

Port range:

(e.g., 80 or 49152-65535)

Source: 0.0.0.0/0

(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

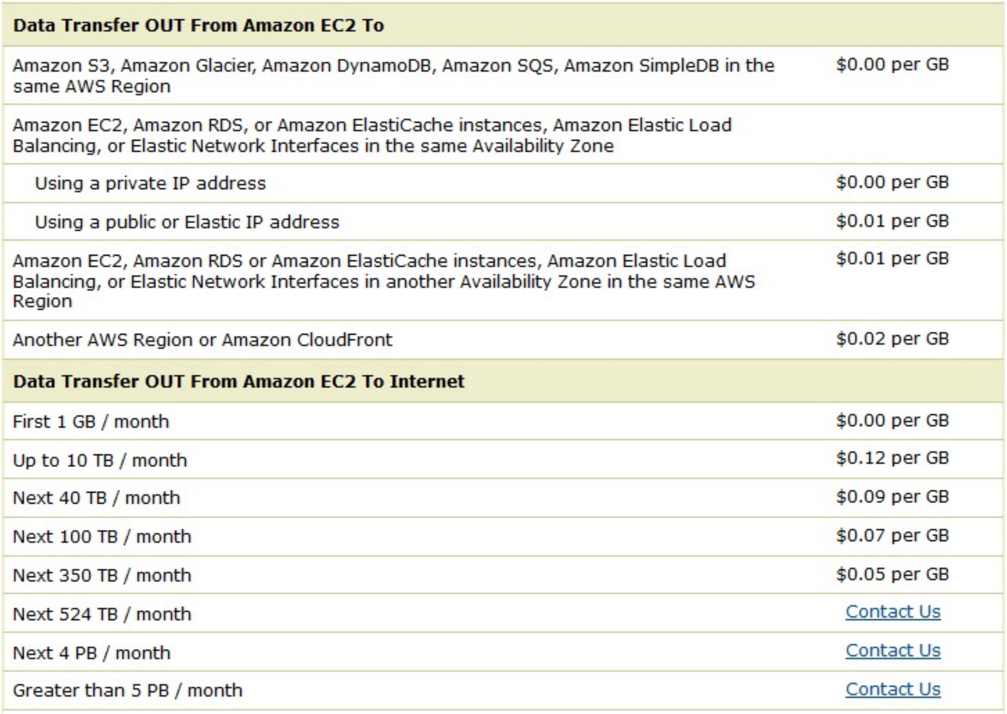


Port (Service)	Source	Action
ICMP		
ALL	sg-2fc91646 (default)	Delete
TCP		
0 - 65535	sg-2fc91646 (default)	Delete
22 (SSH)	0.0.0.0/0	Delete
80 (HTTP)	0.0.0.0/0	Delete

Configuring Firewalls

- Instances are put into “security groups”
- Each security groups defines a set of firewalls rules on who can connect to the instance
  - Make sure that port 22 (ssh) is open so you can log in the instance.
  - If you are running additional services you might need to add more rules
    - e.g., port 80 for HTTP traffic
  - format for IP ipaddress/length of the netmask
- Create your own security group with the group name in it

Network pricing



Data Transfer OUT From Amazon EC2 To	
Amazon S3, Amazon Glacier, Amazon DynamoDB, Amazon SQS, Amazon SimpleDB in the same AWS Region	$0.00 per GB
Amazon EC2, Amazon RDS, or Amazon ElastiCache instances, Amazon Elastic Load Balancing, or Elastic Network Interfaces in the same Availability Zone	
Using a private IP address	$0.00 per GB
Using a public or Elastic IP address	$0.01 per GB
Amazon EC2, Amazon RDS or Amazon ElastiCache instances, Amazon Elastic Load Balancing, or Elastic Network Interfaces in another Availability Zone in the same AWS Region	$0.01 per GB
Another AWS Region or Amazon CloudFront	$0.02 per GB
Data Transfer OUT From Amazon EC2 To Internet	
First 1 GB / month	$0.00 per GB
Up to 10 TB / month	$0.12 per GB
Next 40 TB / month	$0.09 per GB
Next 100 TB / month	$0.07 per GB
Next 350 TB / month	$0.05 per GB
Next 524 TB / month	Contact Us
Next 4 PB / month	Contact Us
Greater than 5 PB / month	Contact Us

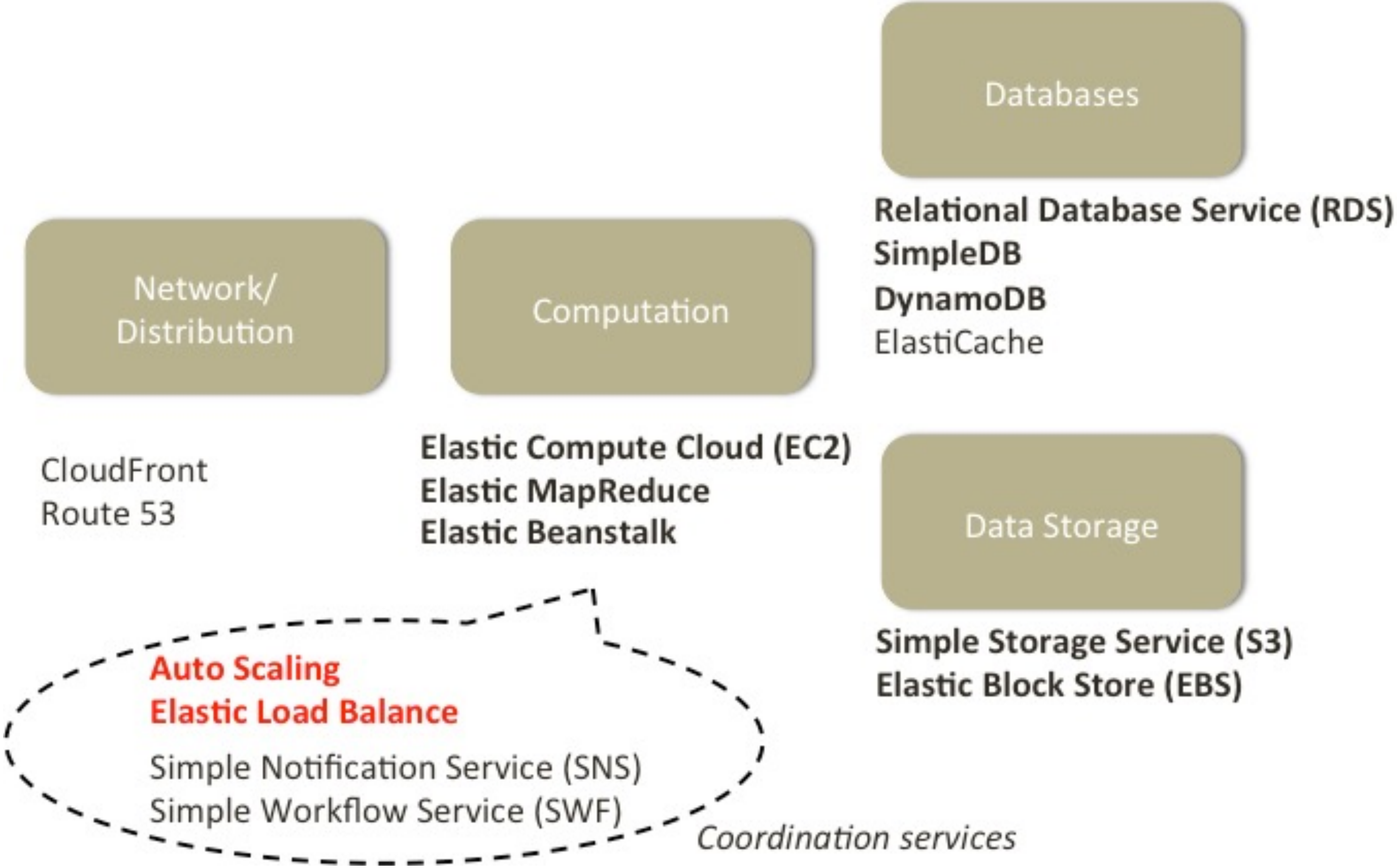
- AWS does charge for network traffic
  - Price depends on source and destination of traffic
  - Free within EC2 and other AWS svcs in same region (e.g., S3)
  - Remember: ISPs are typically charged for upstream traffic

Summary of EC2

- What EC2 is:
  - IaaS service - you can rent virtual machines
  - Various types: Very small to very powerful

- How to use EC2:
  - Ephemeral state - local data is lost when instance terminates
  - AMIs - used to initialize an instance (OS, applications, ...)
  - Security groups - "firewalls" for your instances
  - Regions and availability zones
  - On-demand/reserved/spot instances
  - Service level agreement (SLA)

Overview of AWS Services



Network/
Distribution

CloudFront
Route 53

Computation

Elastic Compute Cloud (EC2)
Elastic MapReduce
Elastic Beanstalk

Databases

Relational Database Service (RDS)
SimpleDB
DynamoDB
ElastiCache

Data Storage

Simple Storage Service (S3)
Elastic Block Store (EBS)

Auto Scaling
Elastic Load Balance

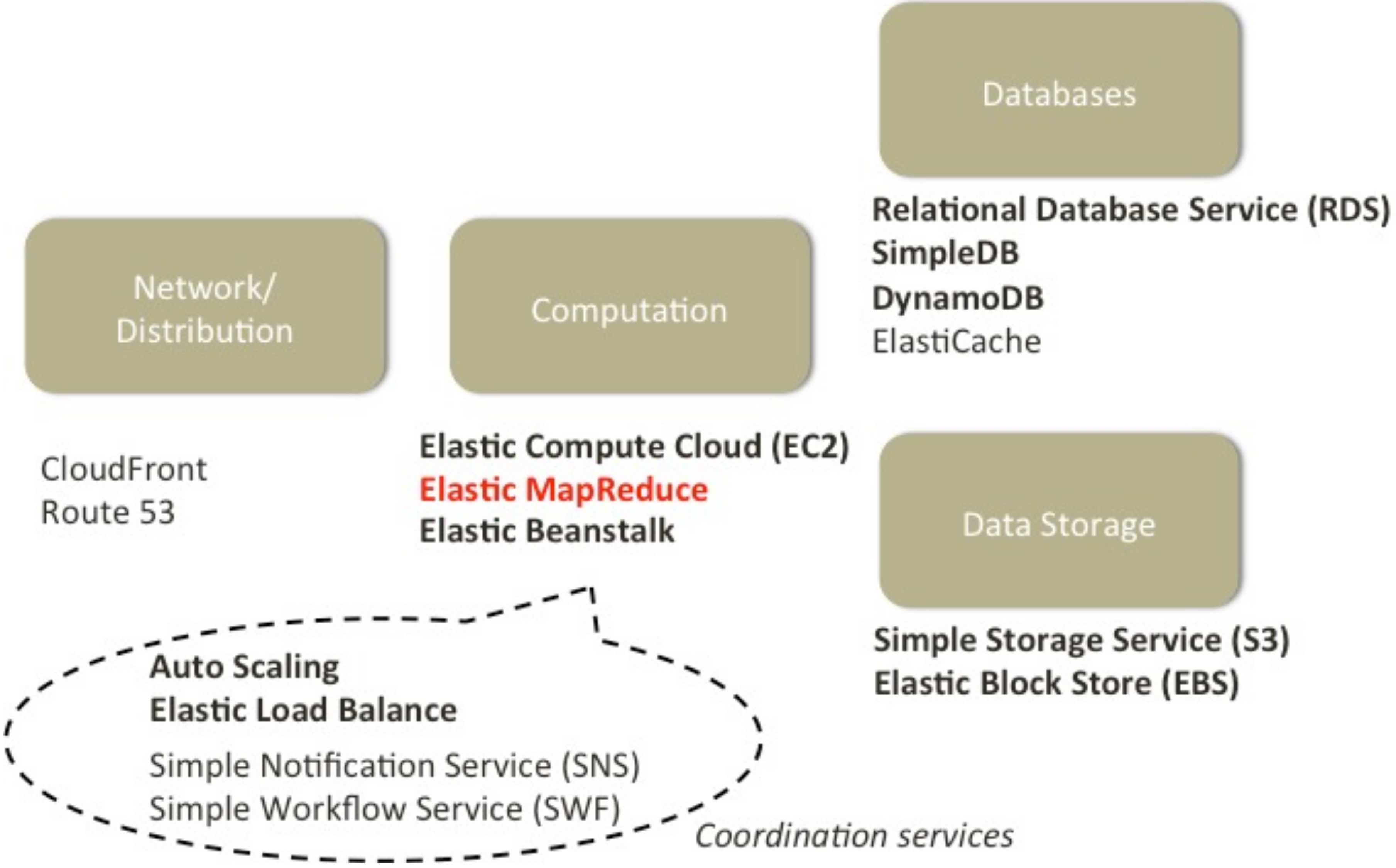
Simple Notification Service (SNS)
Simple Workflow Service (SWF)

Coordination services

Auto Scaling & Elastic Load Balance

- Auto Scaling
  - Monitor the load on EC2 instances using CloudWatch
  - Define Conditions
  - Spawn new instances when there is too much load or remove instances when not enough load
- Elastic Load Balance
  - Automatically distributes incoming application traffic across multiple EC2 instances
  - Detects EC2 instance health and divert traffic from bad ones
  - Support different protocols
    - HTTP, HTTPS, TCP, SSL, or Custom
- They can work together

Overview of AWS Services



Network/ Distribution

CloudFront
Route 53

Computation

Elastic Compute Cloud (EC2)
Elastic MapReduce
Elastic Beanstalk

Databases

Relational Database Service (RDS)
SimpleDB
DynamoDB
ElastiCache

Data Storage

Simple Storage Service (S3)
Elastic Block Store (EBS)

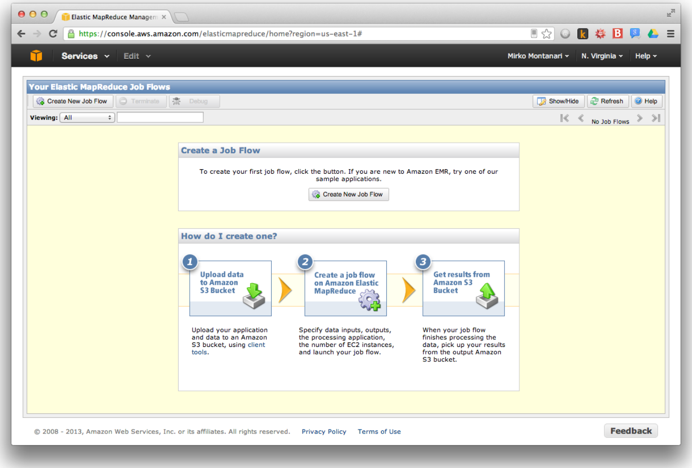
Auto Scaling
Elastic Load Balance
Simple Notification Service (SNS)
Simple Workflow Service (SWF)

Coordination services

Elastic MapReduce (EMR)

- EMR utilizes a hosted Hadoop framework running on the web-scale infrastructure of Amazon EC2 and Amazon S3

1. Write your Hadoop program in Java
2. Submit the jar for to EMR
3. Store the input in S3
4. Tell EMR to run it (web interface or CLI)
5. EMR runs it and stores the results back in S3


Elastic MapReduce Managem
https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#
Services
Edit
Mirko Montanari
N. Virginia
Help
Your Elastic MapReduce Job Flows
Create New Job Flow
Terminate
Debug
Show/Hide
Refresh
Help
Viewing: All
No Job Flows
Create a Job Flow
Create New Job Flow
How do I create one?
1 Upload data to Amazon S3 Bucket
2 Create a job flow on Amazon Elastic MapReduce
3 Get results from Amazon S3 Bucket
Privacy Policy
Terms of Use
Feedback

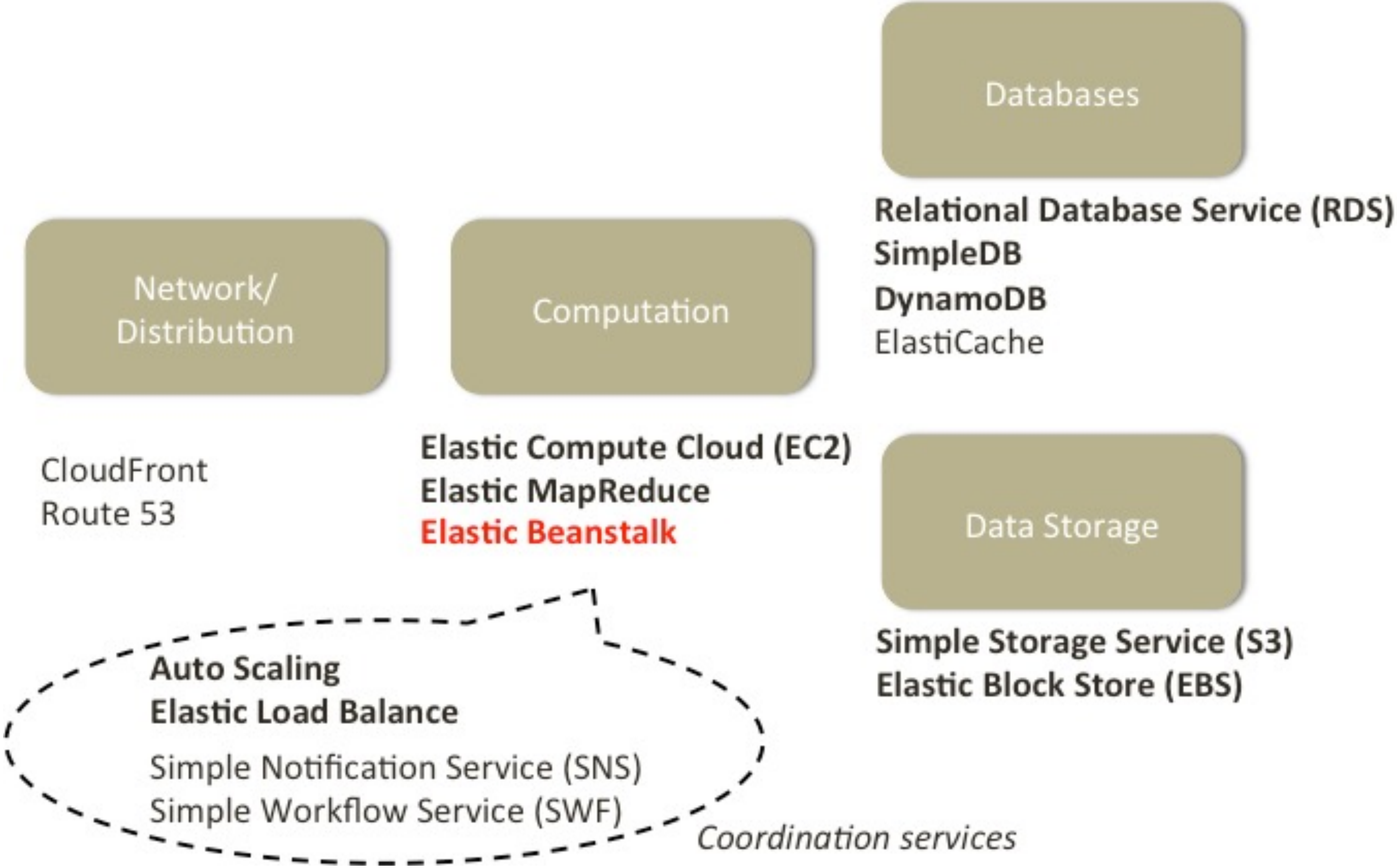
To create your first job flow, click the button. If you are new to Amazon EMR, try one of our sample applications.

1. Upload your application and data to an Amazon S3 bucket, using client tools.
2. Specify data inputs, outputs, the processing application, the number of EC2 instances, and launch your job flow.
3. When your job flow finishes processing the data, pick up your results from the output Amazon S3 bucket.

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved.

It takes up to 10 minutes to start your job, EMR looks for unused resources to minimize the costs

Overview of AWS Services



Databases

Relational Database Service (RDS)
SimpleDB
DynamoDB
ElastiCache

Network/
Distribution

CloudFront
Route 53

Computation

Elastic Compute Cloud (EC2)
Elastic MapReduce
Elastic Beanstalk

Data Storage

Simple Storage Service (S3)
Elastic Block Store (EBS)

Auto Scaling
Elastic Load Balance
Simple Notification Service (SNS)
Simple Workflow Service (SWF)

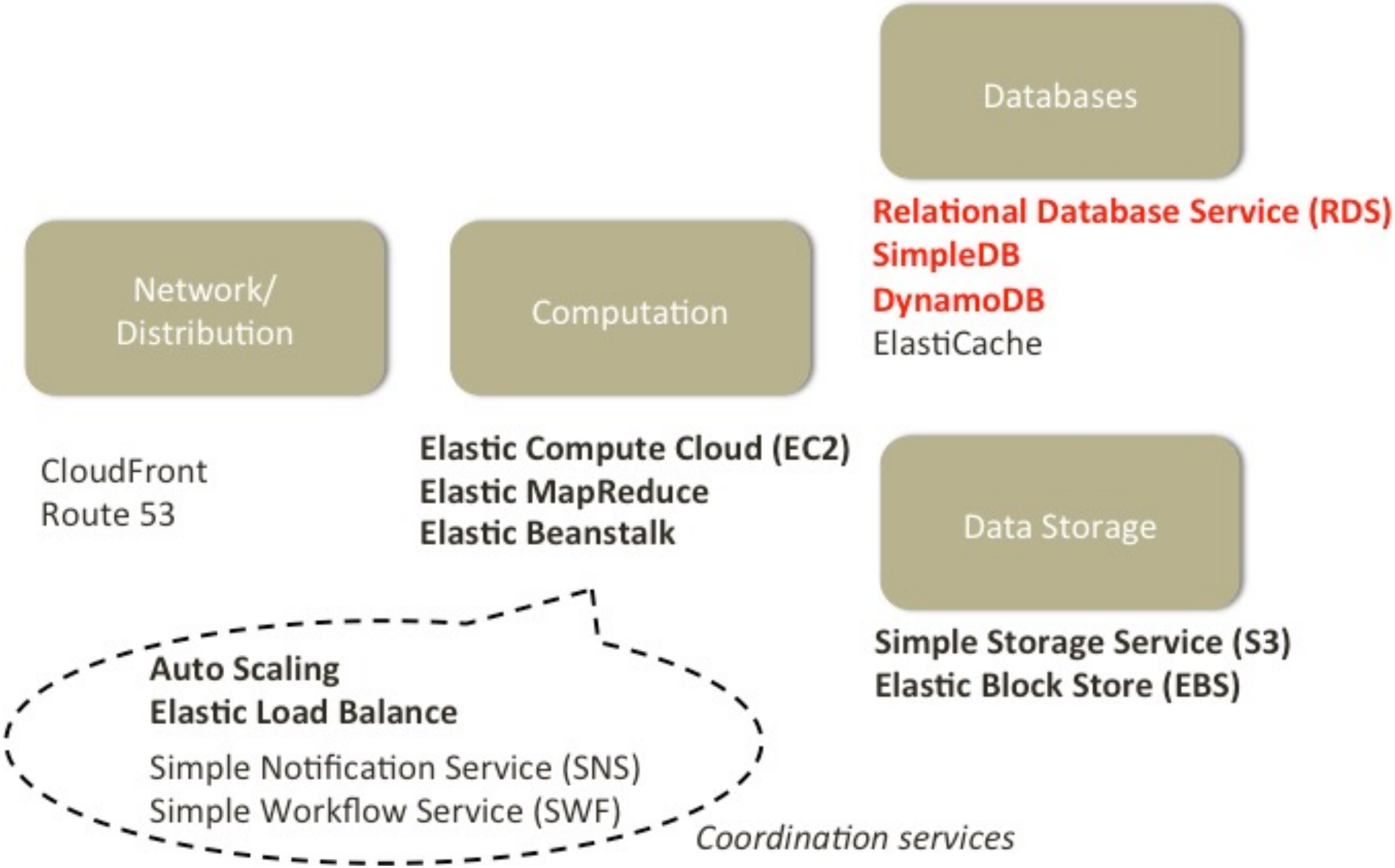
Coordination services

Elastic Beanstalk

- Solution for Enterprise server-side java application deployment
  - Write a Tomcat app, let Amazon deploy it, scale it when traffic increases, and detect failures.
- Create your normal Tomcat Java Web Application (e.g. Eclipse).
- Upload your application code in as WAR file.
- Deploy the application
  - Elastic Beanstalk handles the provisioning of a load balancer and the deployment of the WAR file to one or more EC2 instances running the Apache Tomcat application server
- Access the application at a customized URL

  (e.g. http://myapp.elasticbeanstalk.com/).

Overview of AWS Services



Databases

Relational Database Service (RDS)
SimpleDB
DynamoDB
ElastiCache

Network/ Distribution

CloudFront
Route 53

Computation

Elastic Compute Cloud (EC2)
Elastic MapReduce
Elastic Beanstalk

Data Storage

Simple Storage Service (S3)
Elastic Block Store (EBS)

Auto Scaling
Elastic Load Balance

Simple Notification Service (SNS)
Simple Workflow Service (SWF)

Coordination services

Relational Database Service (RDS)

- Preconfigured EC2 instances with MySQL or Oracle installed
  1. Create an RDS instance
  2. Dump your database into it
     - mysqldump acme | mysql --host=hostname --user=username --password acme
  3. Update SQL connection strings in your application (which might be running anywhere, including EC2 VMs)
- Features
  - Pre-configured
  - Monitoring and Metrics (CloudWatch)
  - Automatic Software Patching
  - Automated Backups
  - DB Snapshots
  - Changing the instance type (= increase computer power)
    - Through EBS snapshots
  - Multi-AZ Deployments
  - Read Replicas
    - Scaling for read-heavy database workloads
  - Isolation and Security

SimpleDB

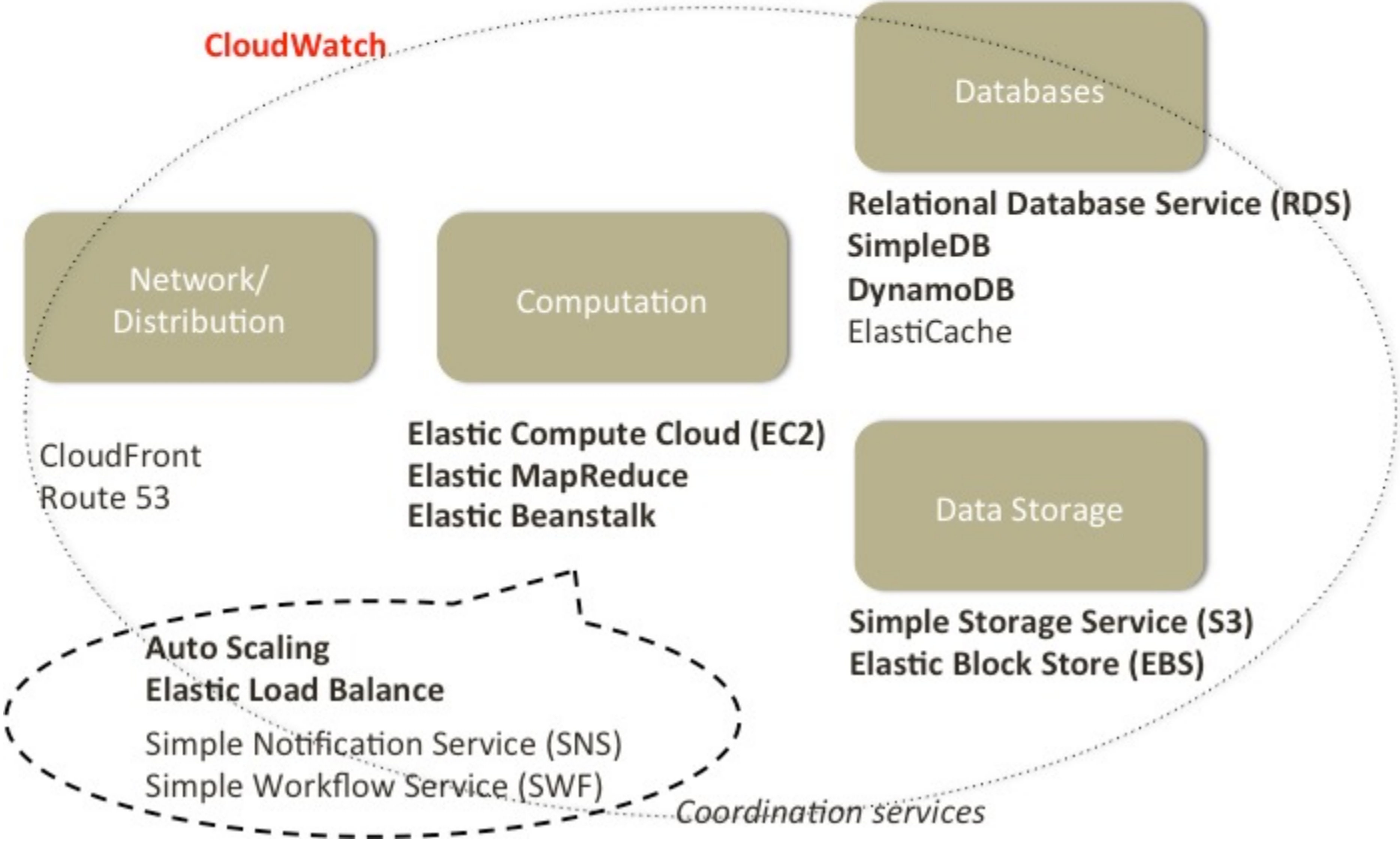
- A NoSQL database, non-relational
- Eventual consistency or strong consistency, depending on the request
- Data model is comprised of domains, items, attributes and values
  - Large collections of items organized into domains
  - Items are little hash tables containing attributes of key, value pairs
- Use Put, Batch Put, & Delete to create and manage the data set
- Use GetAttributes to retrieve a specific item
- Attributes can be searched with various lexicographical queries
- The service manages infrastructure provisioning, hardware and software maintenance, replication, indexing of data items, and performance tuning
- Tables limited to 10 GB, typically under 25 writes/second
- User manages partitioning and re-partitioning of data over additional SimpleDB tables

SimpleDB	**S3**
Indexes all the attributes	Stores raw data
Uses less dense drives	Uses dense storage drives
Better optimized for random access	Optimized for storing large objects

DynamoDB

- Amazon Dynamo paper (2007) → Open-source Apache Cassandra project → DynamoDB (1/2012)
  - Dynamo is a highly available, key-value structured storage system
- Fully managed NoSQL non-relational Database
- Data model is comprised of domains, items, attributes and values (similar to SimpleDB)
  - Domains are collections of items that are described by attribute-value pairs
- **Pay by reserved throughput + indexed storage**
- Integrates with Hadoop MapReduce using Elastic MapReduce
- Run on solid state disks (SSDs)
- There are no limits on the request capacity or storage size for a given table.
  - DynamoDB automatically partitions data and workload over a sufficient number of servers to meet the scale requirements

Overview of AWS Services



CloudWatch

Network/ Distribution

CloudFront
Route 53

Computation

Elastic Compute Cloud (EC2)
Elastic MapReduce
Elastic Beanstalk

Databases

Relational Database Service (RDS)
SimpleDB
DynamoDB
ElastiCache

Data Storage

Simple Storage Service (S3)
Elastic Block Store (EBS)

Auto Scaling
Elastic Load Balance
Simple Notification Service (SNS)
Simple Workflow Service (SWF)

Coordination services

Amazon CloudWatch

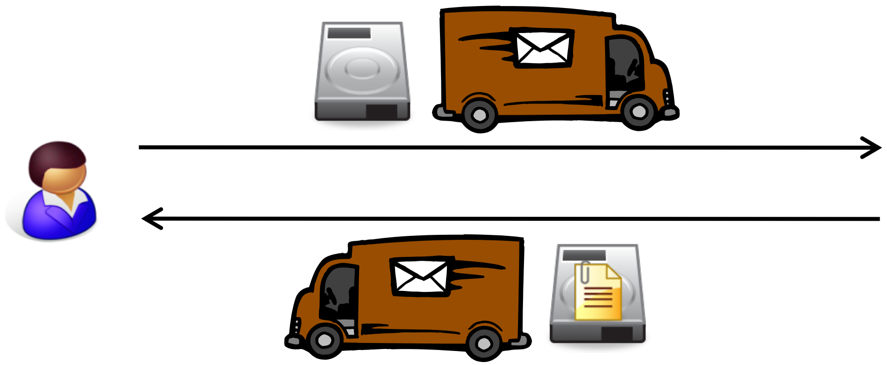
- Monitor AWS resources automatically
  - Monitoring for Amazon EC2 instances: seven pre-selected metrics at five-minute frequency
  - Amazon EBS volumes: eight pre-selected metrics at five-minute frequency
  - Elastic Load Balancers: four pre-selected metrics at one-minute frequency
  - Amazon RDS DB instances: thirteen pre-selected metrics at one-minute frequency
  - Amazon SQS queues: seven pre-selected metrics at five-minute frequency
  - Amazon SNS topics: four pre-selected metrics at five-minute frequency
- Custom Metrics generation and monitoring
- Set alarms on any of the metrics to receive notifications or take other automated actions
- Use Auto Scaling to add or remove EC2 instances dynamically based on CloudWatch metrics

Additional services from AWS

AWS Import/Export

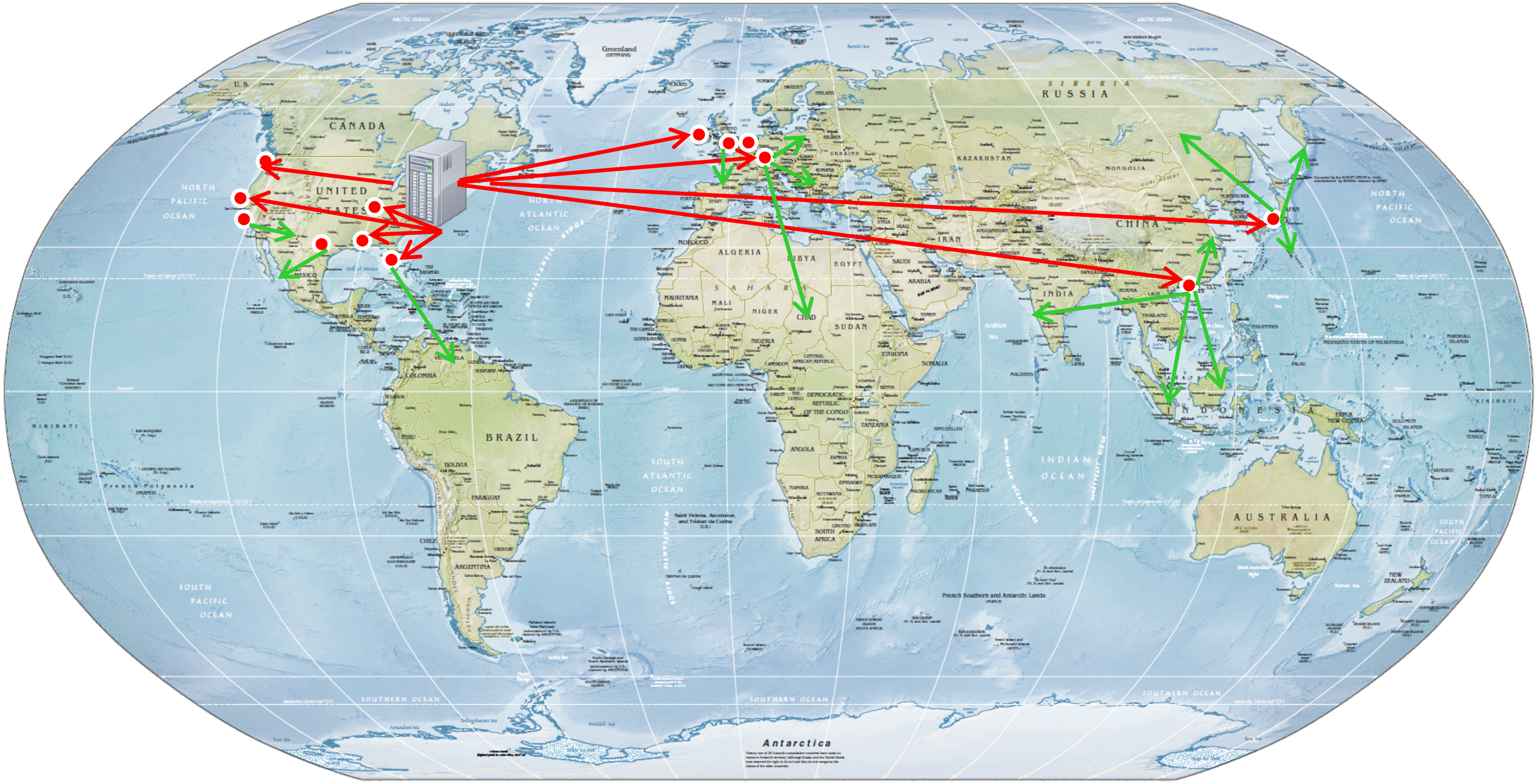
Method	Time
Internet (20Mbps)	45 days
FedEx	1 day

Time to transfer 10TB [AF10]



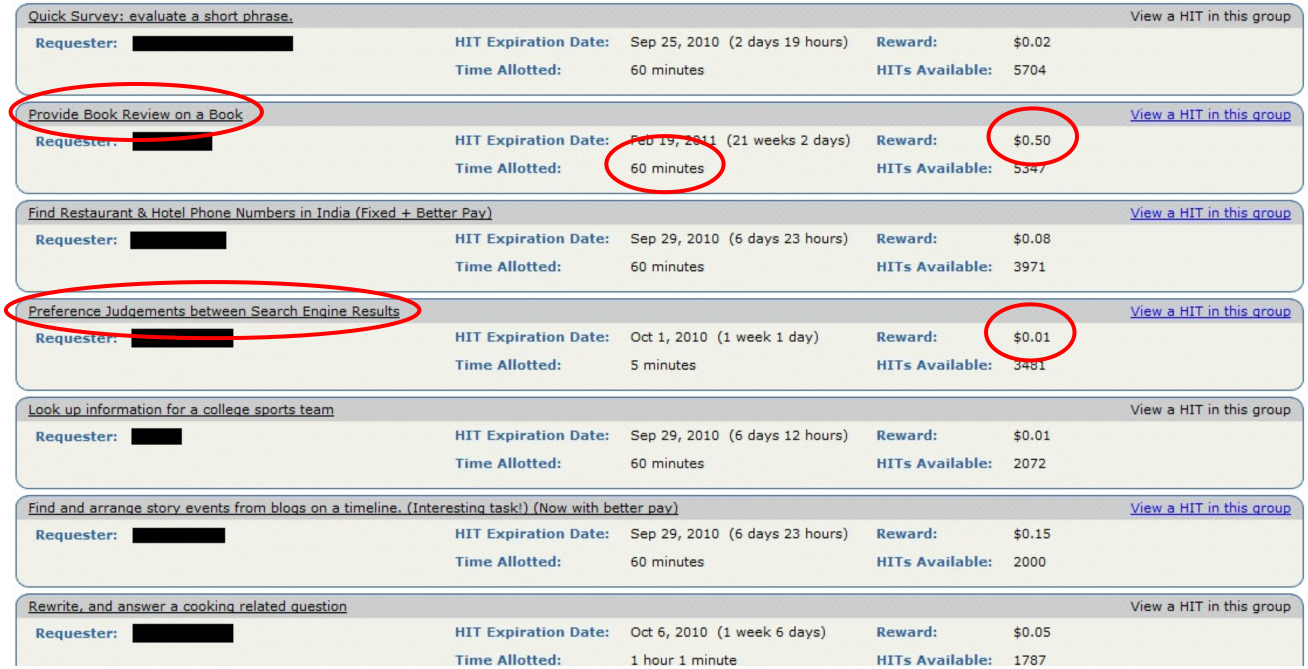
- Import/export large amounts of data to/from S3 buckets via physical storage device
  - Mail an actual hard disk to Amazon (power adapter, cables!)
  - Signature file for authentication
  - Discussion: Is this the Right Way for shipping data, or should we rather be using a network?

CloudFront



- Content distribution network
  - Caches S3 content at edge locations for low-latency delivery
  - Some similarities to other CDNs like Akamai, Limelight, ...

Mechanical Turk (MTurk)



Quick Survey: evaluate a short phrase. — View a HIT in this group
Requester: [redacted] — HIT Expiration Date: Sep 25, 2010 (2 days 19 hours) — Reward: $0.02
Time Allotted: 60 minutes — HITs Available: 5704

Provide Book Review on a Book — View a HIT in this group
Requester: [redacted] — HIT Expiration Date: Feb 19, 2011 (21 weeks 2 days) — Reward: $0.50
Time Allotted: 60 minutes — HITs Available: 5347

Find Restaurant & Hotel Phone Numbers in India (Fixed + Better Pay) — View a HIT in this group
Requester: [redacted] — HIT Expiration Date: Sep 29, 2010 (6 days 23 hours) — Reward: $0.08
Time Allotted: 60 minutes — HITs Available: 3971

Preference Judgements between Search Engine Results — View a HIT in this group
Requester: [redacted] — HIT Expiration Date: Oct 1, 2010 (1 week 1 day) — Reward: $0.01
Time Allotted: 5 minutes — HITs Available: 3481

Look up information for a college sports team — View a HIT in this group
Requester: [redacted] — HIT Expiration Date: Sep 29, 2010 (6 days 12 hours) — Reward: $0.01
Time Allotted: 60 minutes — HITs Available: 2072

Find and arrange story events from blogs on a timeline. (Interesting task!) (Now with better pay) — View a HIT in this group
Requester: [redacted] — HIT Expiration Date: Sep 29, 2010 (6 days 23 hours) — Reward: $0.15
Time Allotted: 60 minutes — HITs Available: 2000

Rewrite, and answer a cooking related question — View a HIT in this group
Requester: [redacted] — HIT Expiration Date: Oct 6, 2010 (1 week 6 days) — Reward: $0.05
Time Allotted: 1 hour 1 minute — HITs Available: 1787

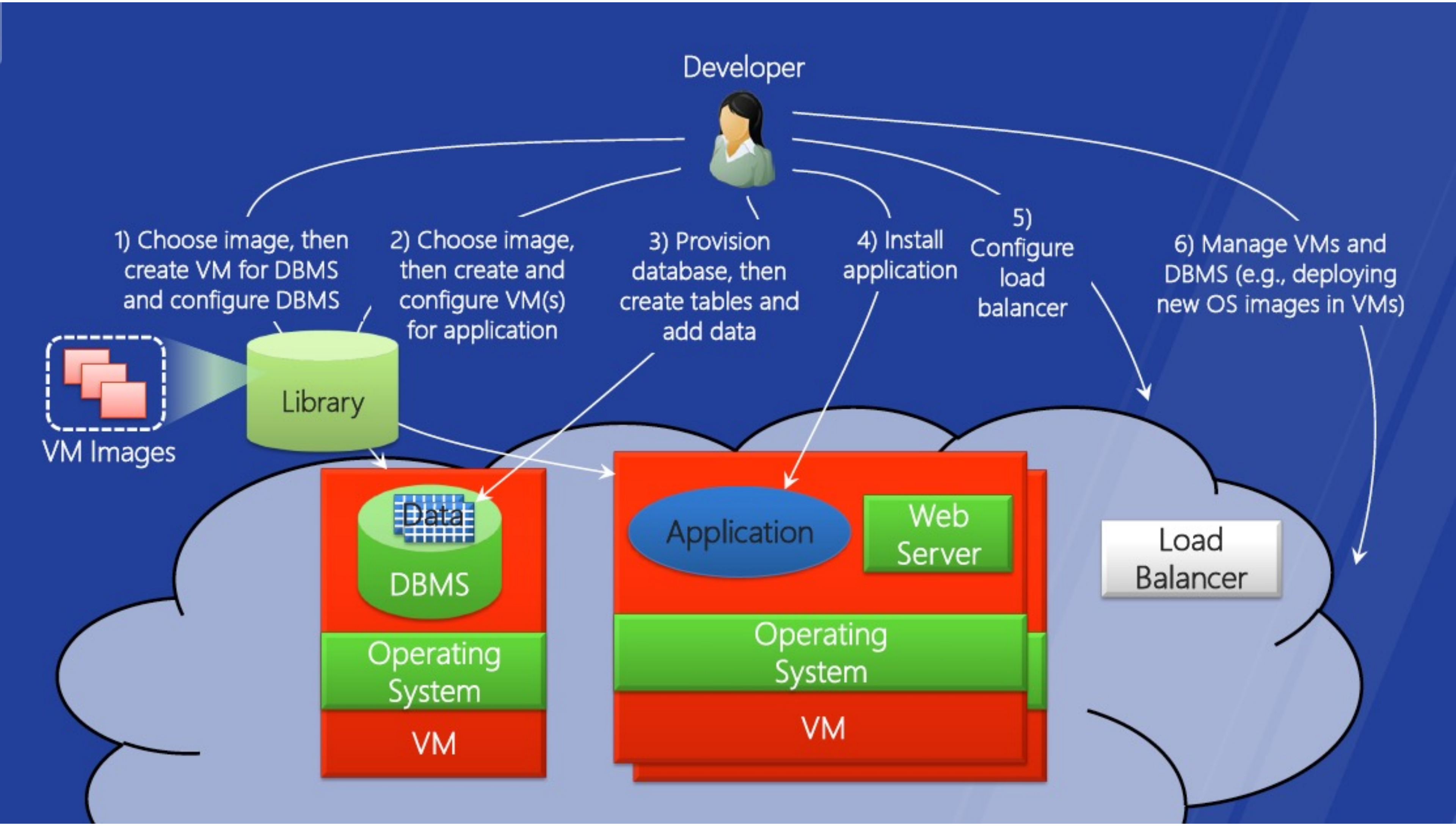
https://www.mturk.com/mturk/ (9/23/2010 1:58am)

- A crowdsourcing (Human) marketplace
  - Requesters post small jobs (HIT - Human Intelligence Task), offer small rewards ($0.01-$0.10)
  - e.g. Volunteer effort to search for Jim Gray

Summary of AWS

- AWS provides a diverse set of services that permits the creation of scalable applications
- Many of cloud providers provide similar services
  - **Storage; Computation; Databases; and other frameworks for building applications and hosting network-based services**

Example of IaaS: Using the IaaS from AWS


Developer
1) Choose image, then create VM for DBMS and configure DBMS
2) Choose image, then create and configure VM(s) for application
3) Provision database, then create tables and add data
4) Install application
5) Configure load balancer
6) Manage VMs and DBMS (e.g., deploying new OS images in VMs)
VM Images
Library
Data
DBMS
Operating System
VM
Application
Web Server
Operating System
VM
Load Balancer

Recap: Examples of *aaS

- Infrastructure as a Service (IaaS): basic compute and storage resources
  - On-demand servers
  - Amazon EC2, VMWare vCloud
- Platform as a Service (PaaS): cloud application infrastructure
  - On-demand application-hosting environment
  - E.g. Google AppEngine, Salesforce.com, Windows Azure, Amazon
- Software as a Service (SaaS): cloud applications
  - On-demand applications
  - E.g. GMail, Microsoft Office Web Companions

Case Studies on Platform as a Service (PaaS) Cloud Providers:

Our 1st PaaS Example:
Google App Engine (GAE)

Google App Engine (GAE)

- GAE was developed in 2008 as a PaaS by Google

- It supports multi-tenancy and offers automatic scaling for web applications

- It supports Python, Java and Go

GAE frameworks and tools

- GAE supports Django web framework and the Grails web app framework

- GAE provides infrastructure tools that enable users to deploy code without worrying about infrastructure challenges such as deployment, failover, scalability

- However, the GAE infrastructure limits the type of applications that can be run

GAE Security, Sandbox

- Applications run in a secure environment
- Isolates applications from hardware and operating system, and imposes security limitations
- Ex. Application code only runs in response to requests and a request handler cannot spawn potentially malicious sub-processes after response has been sent

Storing GAE data

- Users of GAE can use App Engine Datastore, Google Cloud SQL , and Google Cloud Storage

- Can harness Google's database technology like Bigtable

GAE’s use with Google Services

- Can take advantage of Google’s Single-Sign-On feature when other users want to access their gmail or google docs
- Build Chrome and Android games on GAE
- Google Cloud Endpoints to use access mobile services

Other Services supported

- App engine Map Reduce
- Search API
- SSL support
- Page Speed
- XMPP API
- Memcache API

Case Studies of GAE

- BugSense- An application error-reporting service, it used GAE to maintain logs of bugs in software and analyze them
- Ubisoft- used it to build their first web-based game, “From Dust” on Chrome browser
- Claritics- small social analytics company of 15 employees, used to analyze game datasets

GAE is great for Mobile

- Many cell phone apps use GAE for their backend like Ruzzle and Tap Zoo

- Fits GAE’s purpose well of being able to scale up for small teams of developers

A Case Study on the PaaS from Microsoft Azure

Microsoft Azure

- It was launched by Microsoft in 2010
- Provides both PaaS and IaaS services
  - But our discussion will focus on its PaaS side
- It is like a hybrid cloud provider that tries to do multiple things

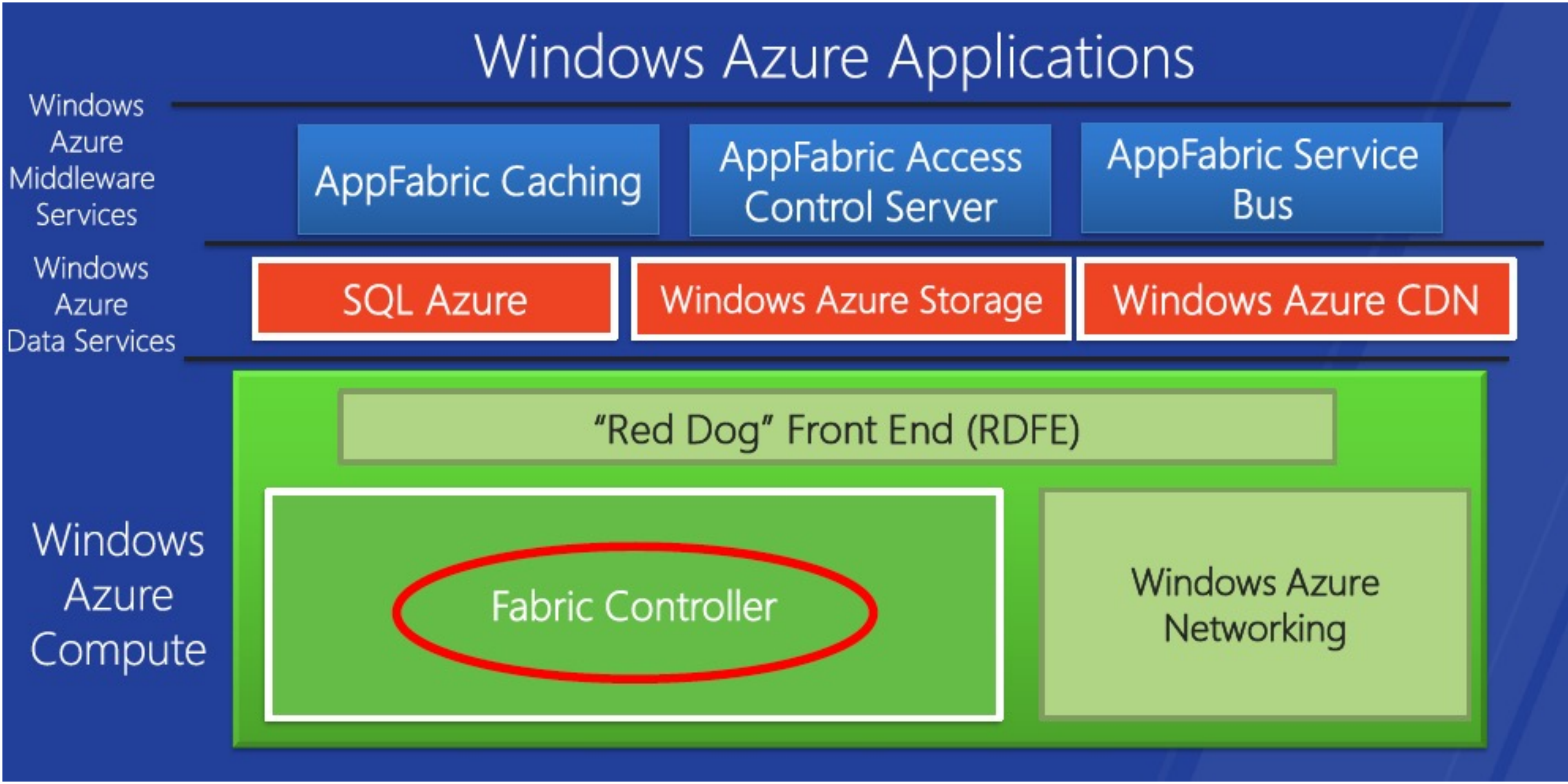
Windows Azure

- Windows Azure is the OS for the data center
  - Model: Treat the data center as a machine
  - Handles resource management, provisioning, and monitoring
  - Manages application lifecycle
  - Allows developers to concentrate on business logic
- Provides shared pool of compute, disk and network
  - Virtualized storage, compute and network
  - Illusion of boundless resources
- Provides common building blocks for distributed applications
  - Reliable queuing, simple structured storage, SQL storage
  - Application services like access control and connectivity

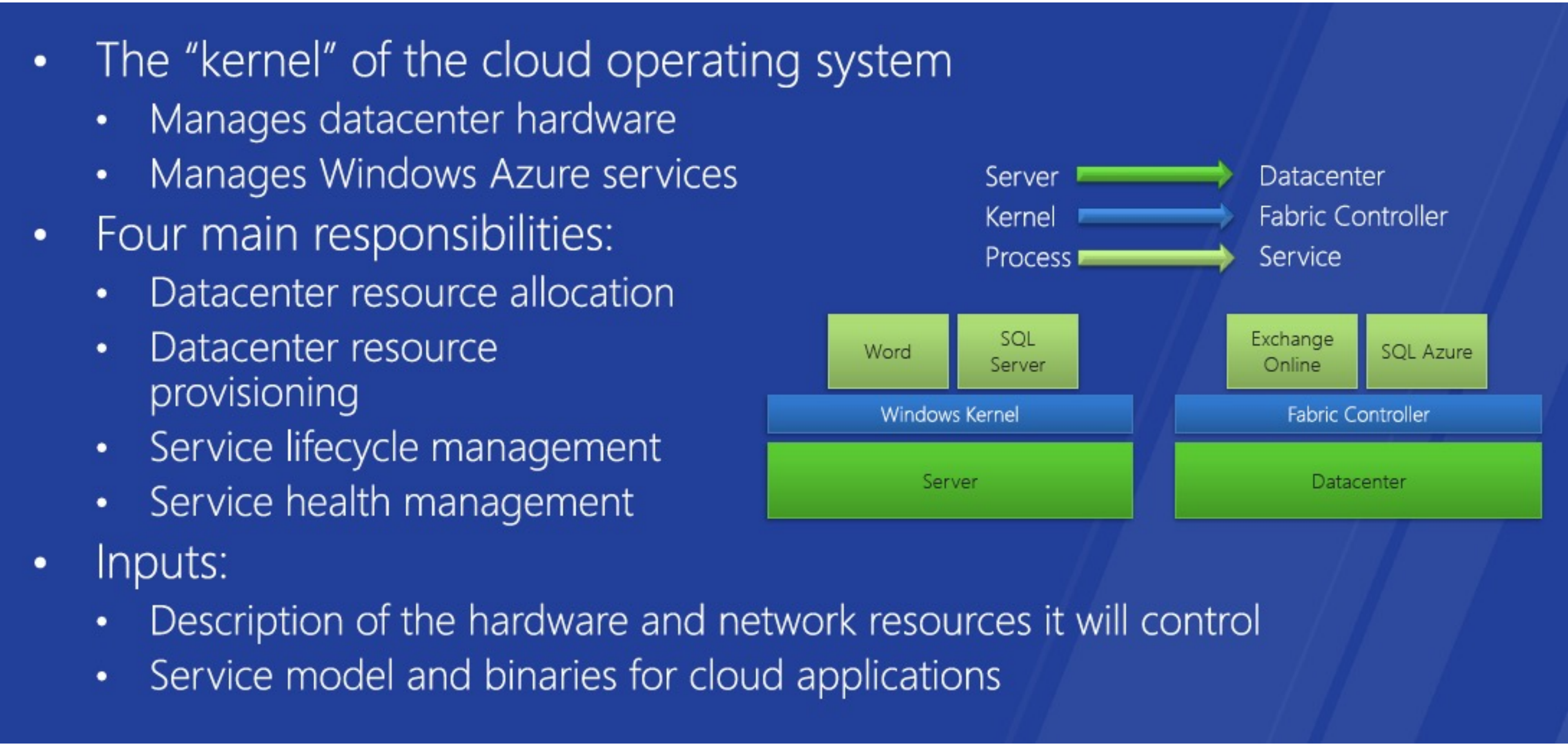
Windows Azure Components

	Windows Azure PaaS
Applications	Windows Azure Service Model
Runtimes	.NET 3.5/4, ASP .NET, PHP
Operating System	Windows Server 2008/R2-Compatible OS
Virtualization	Windows Azure Hypervisor
Server	Microsoft Blades
Database	SQL Azure
Storage	Windows Azure Storage (Blob, Queue, Table)
Networking	Windows Azure-Configured Networking

Windows Azure Platform

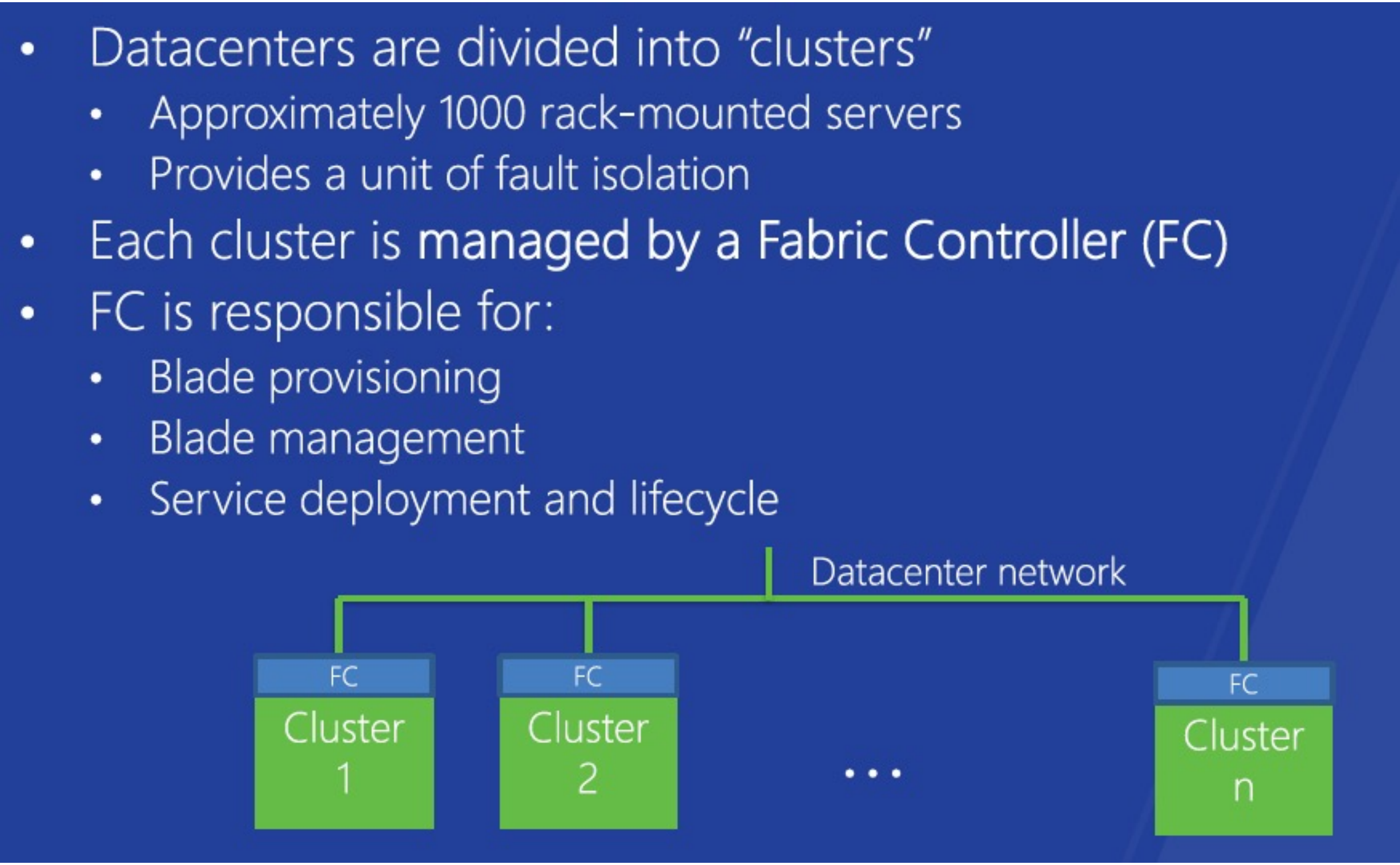


The Fabric Controller (FC)



- The "kernel" of the cloud operating system
  - Manages datacenter hardware
  - Manages Windows Azure services
- Four main responsibilities:
  - Datacenter resource allocation
  - Datacenter resource provisioning
  - Service lifecycle management
  - Service health management
- Inputs:
  - Description of the hardware and network resources it will control
  - Service model and binaries for cloud applications

Azure Datacenter Clusters



- Datacenters are divided into "clusters"
  - Approximately 1000 rack-mounted servers
  - Provides a unit of fault isolation
- Each cluster is managed by a Fabric Controller (FC)
- FC is responsible for:
  - Blade provisioning
  - Blade management
  - Service deployment and lifecycle

Datacenter network

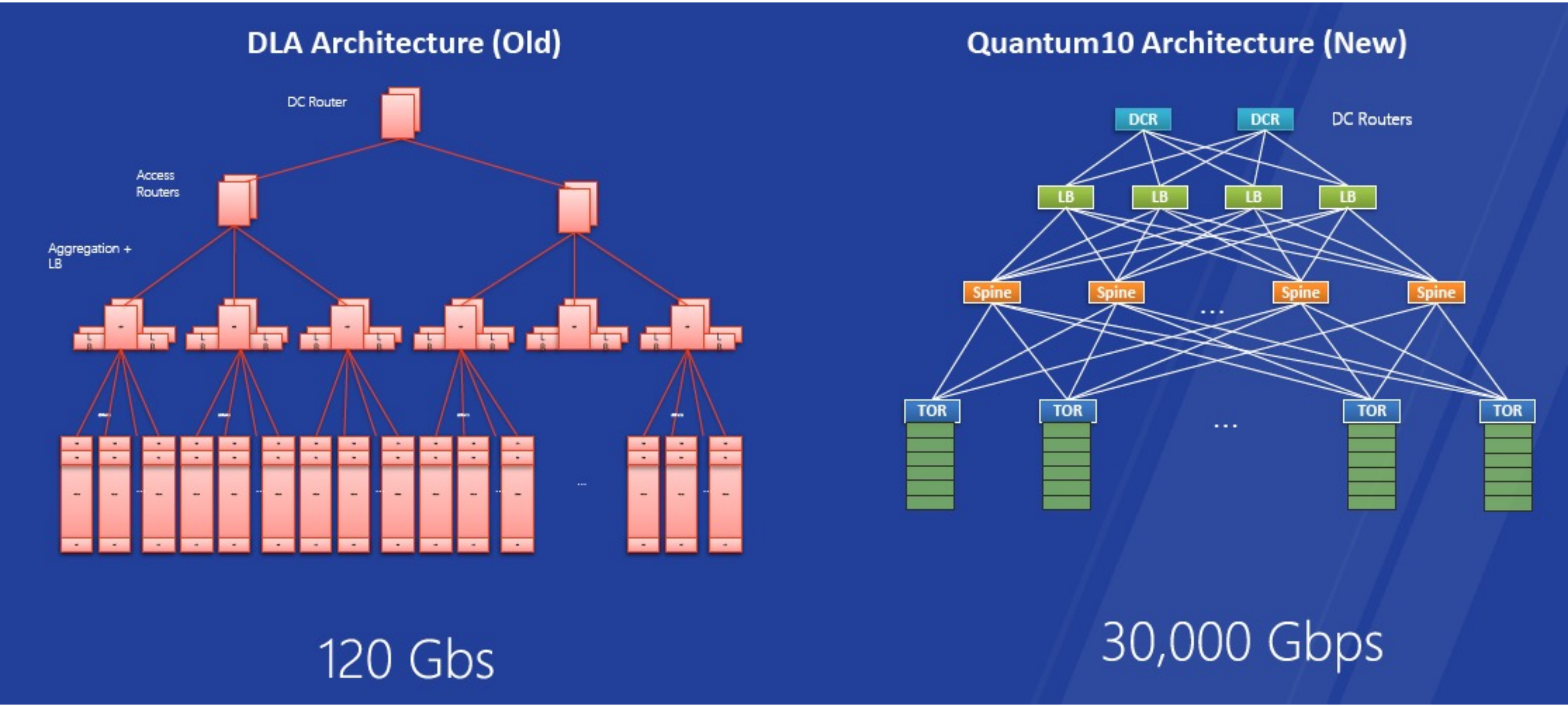
FC
Cluster 1

FC
Cluster 2

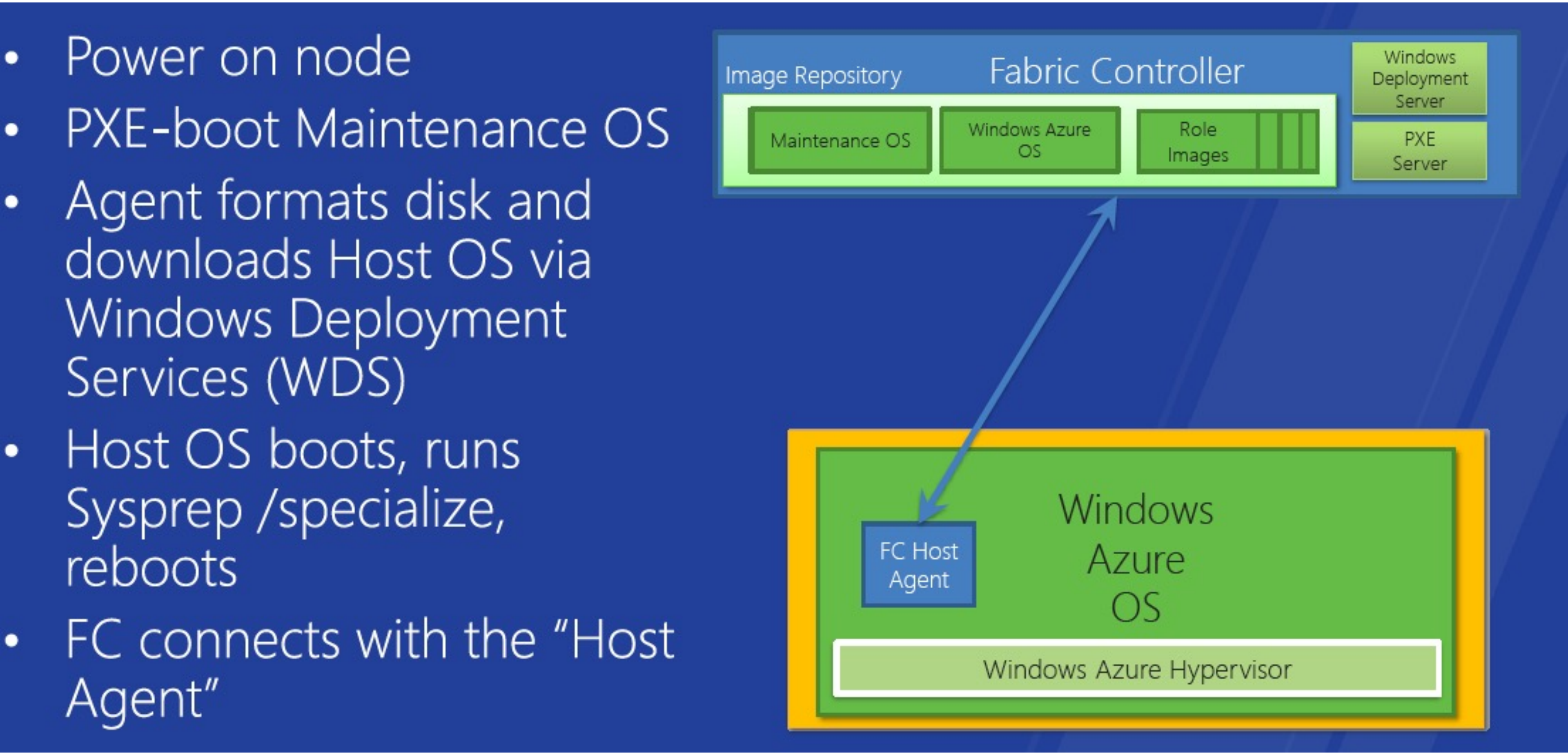
• • •

FC
Cluster n

Azure Datacenter Network Architecture

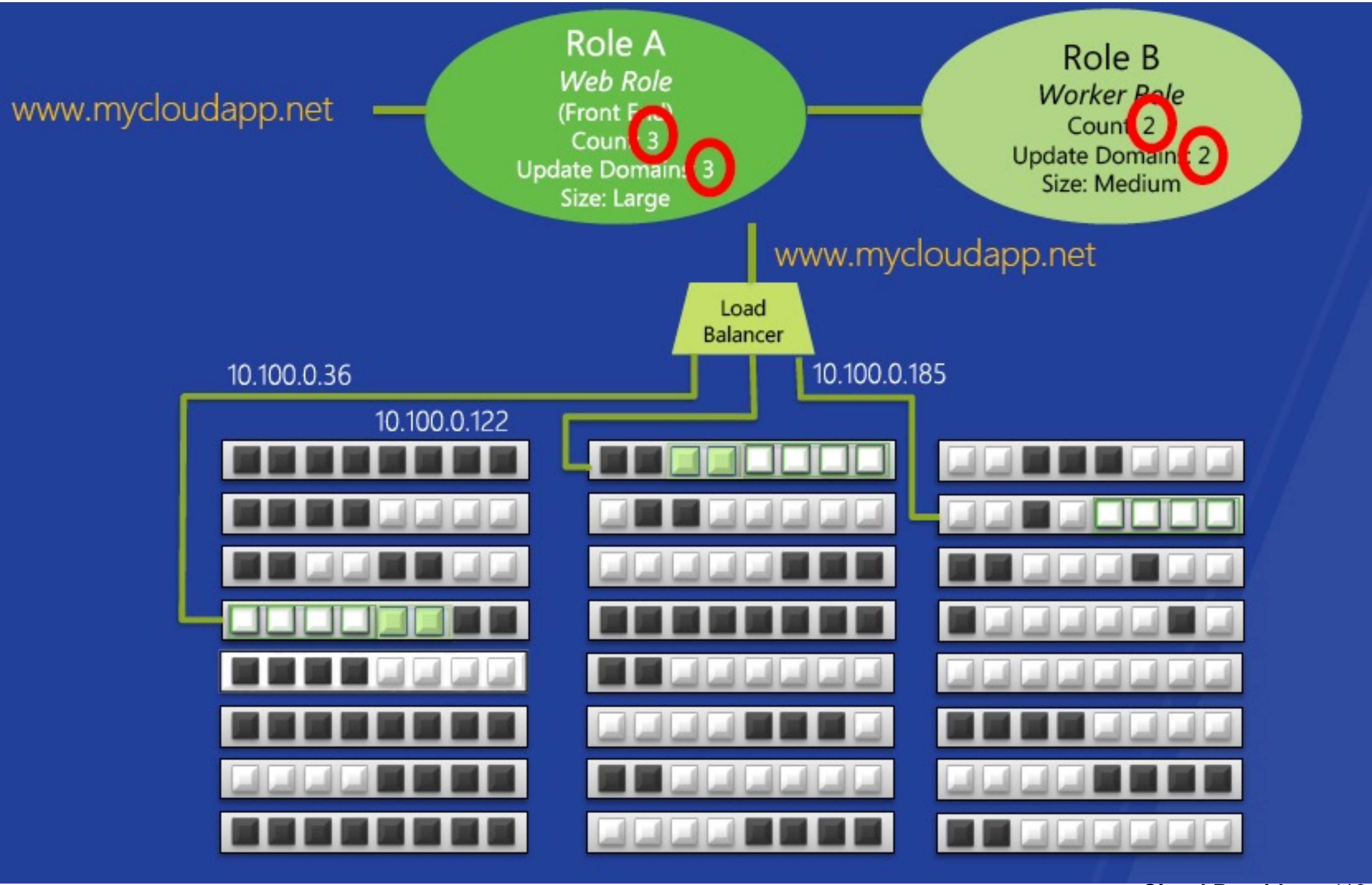


Provisioning a Node (takes 10 minutes!)



- Power on node
- PXE-boot Maintenance OS
- Agent formats disk and downloads Host OS via Windows Deployment Services (WDS)
- Host OS boots, runs Sysprep /specialize, reboots
- FC connects with the “Host Agent”

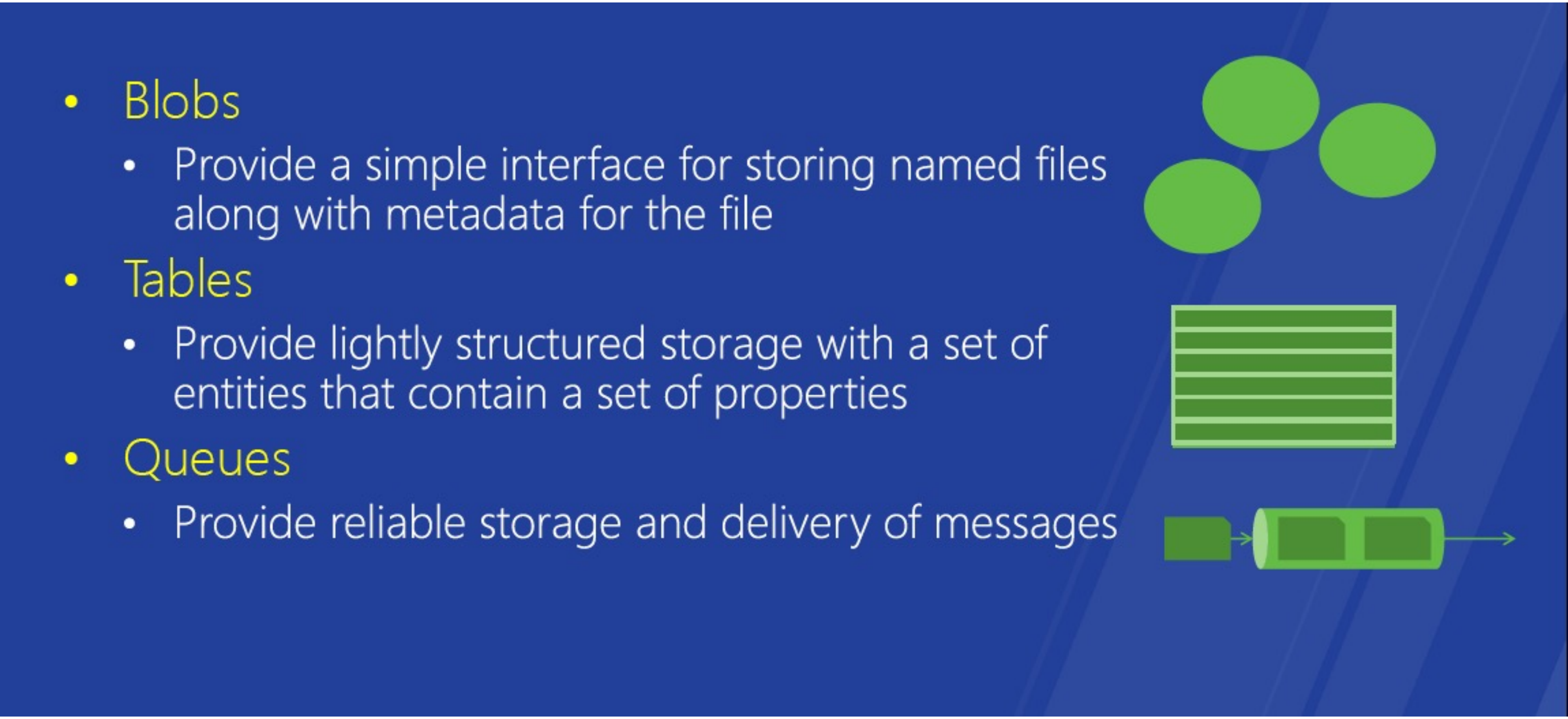
Deploying a Service



Azure Storage Fundamentals

- Storage Characteristics
  - Durable – replicated 3 times
  - Scalable (capacity and throughput)
  - Highly available
- Familiar Programming Interfaces
  - REST (HTTP and HTTPS)
  - .NET accessible

Azure Storage Objects



- Blobs
  - Provide a simple interface for storing named files along with metadata for the file
- Tables
  - Provide lightly structured storage with a set of entities that contain a set of properties
- Queues
  - Provide reliable storage and delivery of messages

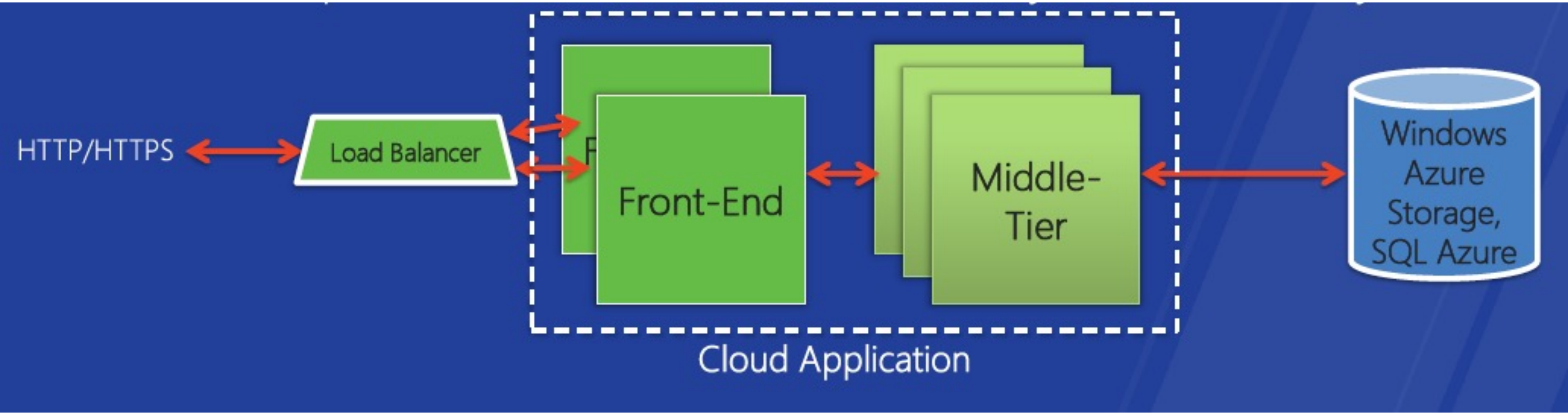
Comparison of Cloud Storage/DB Services

	Model	CAP	Scans	Sec. Indices	Queries	API	Scale-out	SLA
SimpleDB	Table-Store	CP	Yes (as queries)	Auto-matic	SQL-like (no joins, groups, …)	REST + SDKs	✖	✖
Dynamo-DB	Table-Store	CP	By range key / index	Local Sec. Global Sec.	Key+Cond. On Range Key(s)	REST + SDKs	Automatic over Prim. Key	✖
Azure Tables	Table-Store	CP	By range key	✖	Key+Cond. On Range Key	REST + SDKs	Automatic over Part. Key	99.9% uptime
AE/Cloud DataStore	Entity-Group	CP	Yes (as queries)	Auto-matic	Conjunct. of Eq. Predicates	REST/ SDK, JDO,JPA	Automatic over Entity Groups	✖
S3, Az. Blob, GCS	Blob-Store	AP	✖	✖	✖	REST + SDKs	Automatic over key	99.9% uptime (S3)

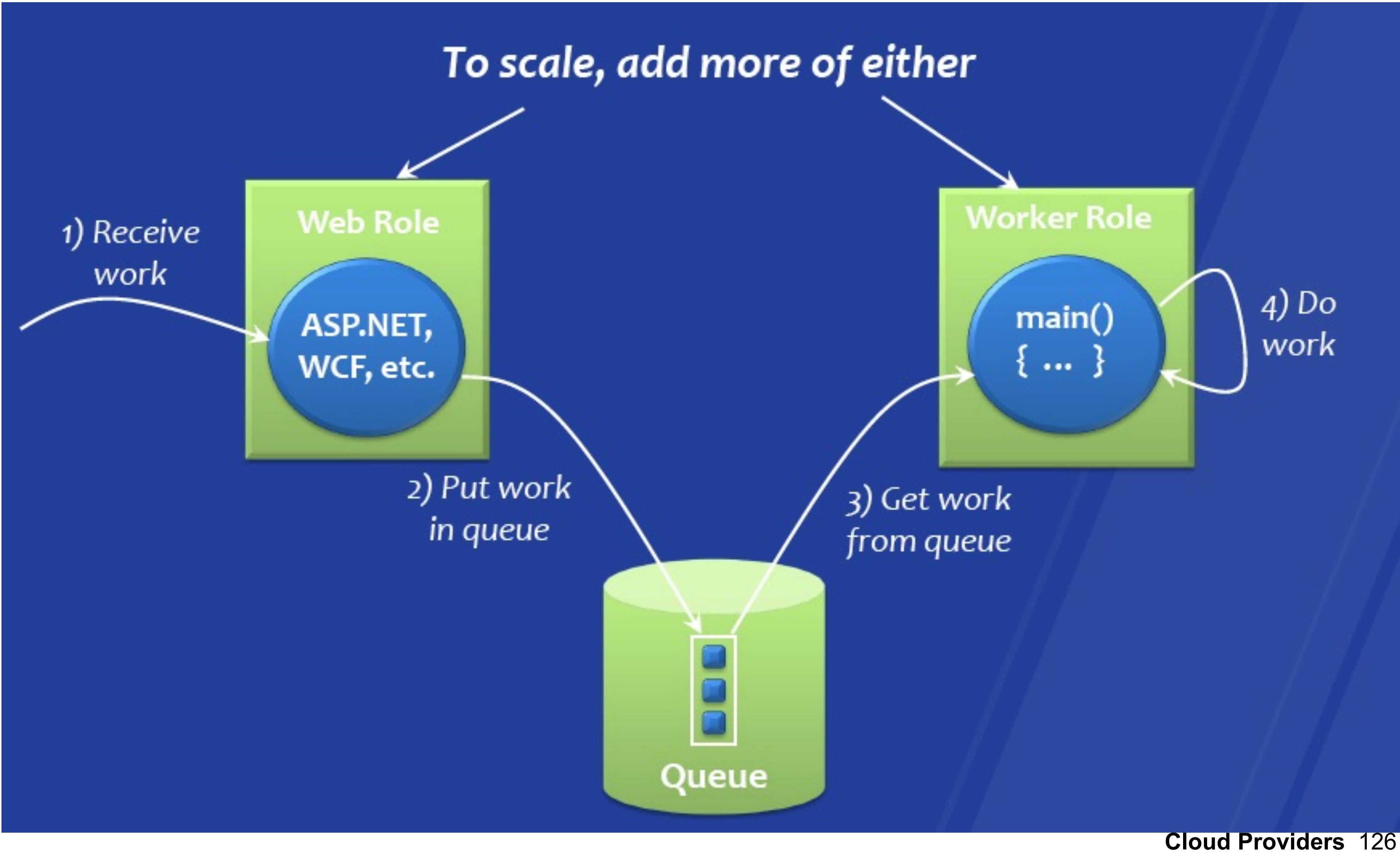
Source: Felix Gessert, "Cloud Databases in Research and Practice," Apr 2014, baqend.com/nosql.pdf

Modeling Cloud Applications under Azure

- A cloud application is typically made up of different components
  - Front end: e.g. load-balanced stateless web servers
  - Middle worker tier: e.g. order processing, encoding
  - Backend storage: e.g. SQL tables or files
  - Multiple instances of each for scalability and availability

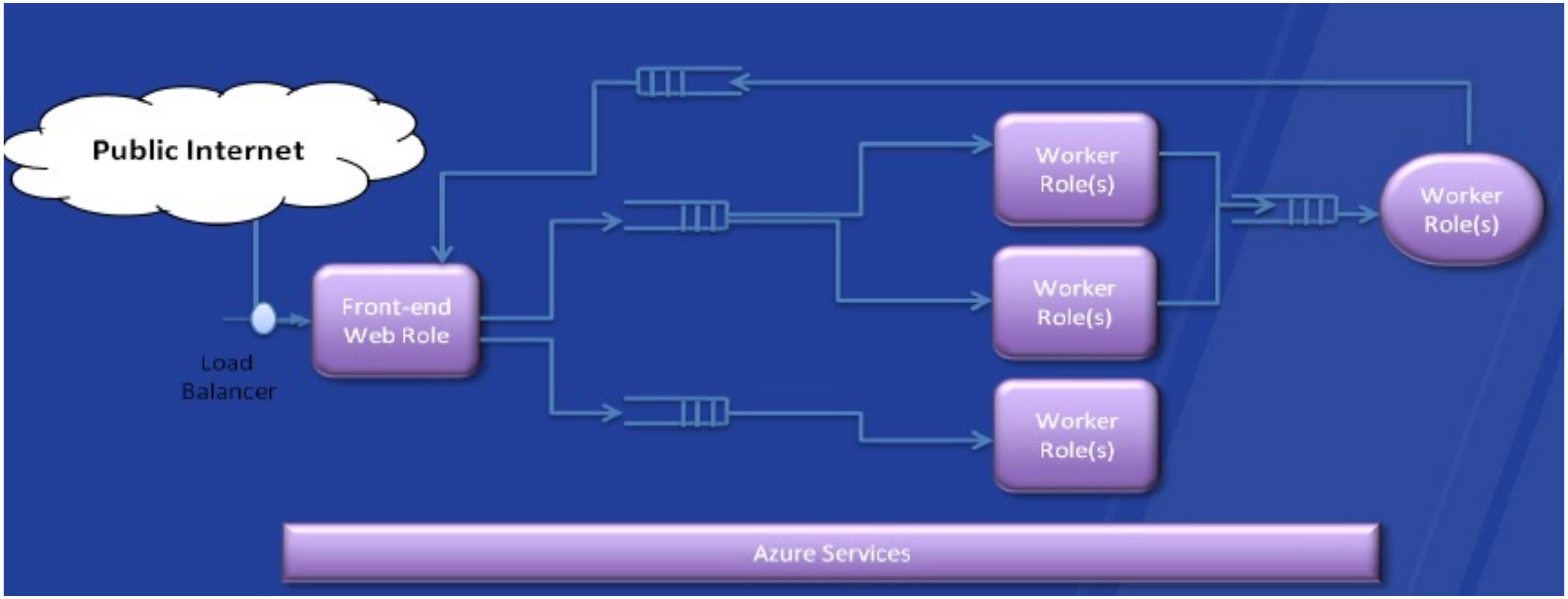


The Suggested Application Model under Azure (using Queues)



Scalable, Fault-tolerant Applications on Azure

- Queues are the application glue
  - Queues decouple different parts of application, making it easier to scale app parts independently
  - Flexible resource allocation, different priority queues and separation of backend servers to process different queues
  - Queues masks(i.e. hides from end-users) faults in worker roles



The Windows Azure Service Model

- A Windows Azure application is called a “service”
  - Definition information
  - Configuration information
  - At least one “role”
- Roles are like DLLs in the service “process”
  - Collection of code with an entry point that runs in its own virtual machine
- There are currently three role types:
  - Web Role: IIS7 and ASP.NET in Windows Azure-supplied OS
  - Worker Role: arbitrary code in Windows Azure-supplied OS
  - VM Role: uploaded VHD with customer-supplied OS

Node and Role Health Maintenance

- FC maintains service availability by monitoring the software and hardware health
  - Based primarily on heartbeats
  - Automatically “heals” affected roles
- Windows Azure compute SLA requires two instances of each role
  - 99.95% for connectivity to two instances
  - Achieved with update and fault domains

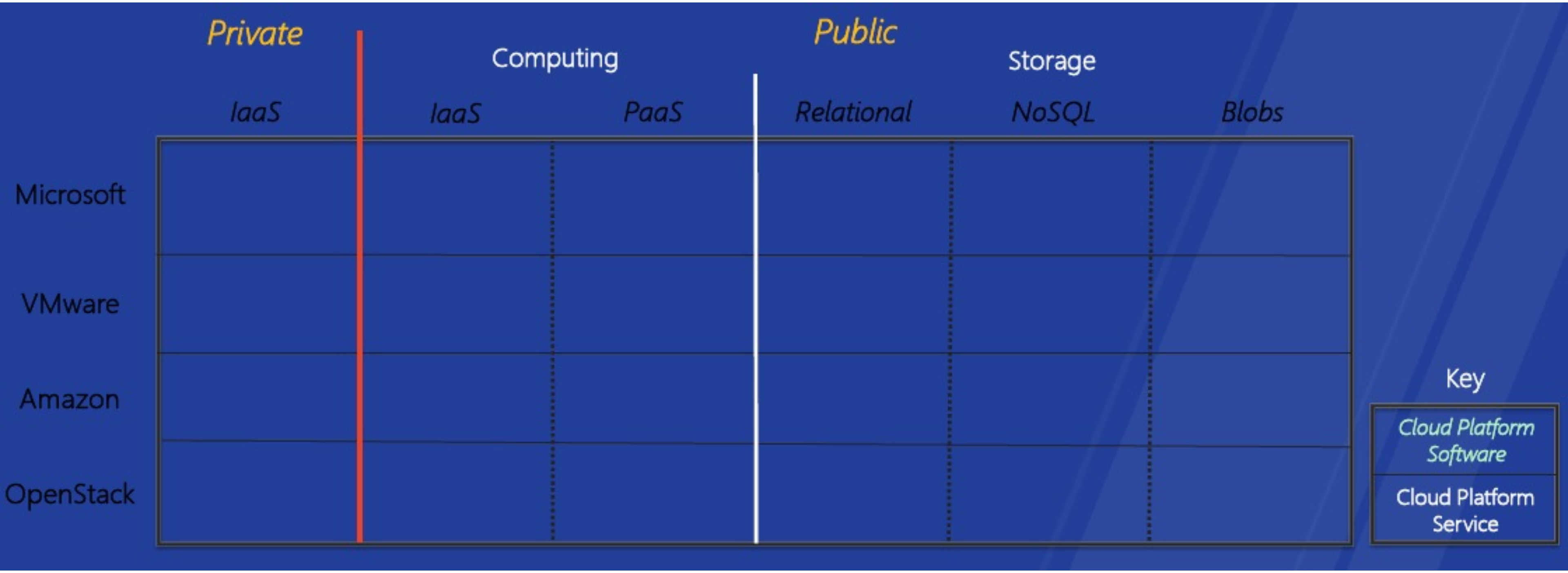
Problem	How to Detect	Fabric Response
Role instance crashes	FC guest agent monitors role termination	FC restarts role
Guest VM or agent crashes	FC host agent notices missing guest agent heartbeats	FC restarts VM and hosted role
Host OS or agent crashes	FC notices missing host agent heartbeat	Tries to recover node FC reallocates roles to other nodes
Detected node hardware issue	Host agent informs FC	FC migrates roles to other nodes Marks node “out for repair”

Azure Architecture Summary

- Platform as a Service is all about reducing management and operations overhead
- Require refactoring (recoding) of application into “roles”
- The Windows Azure Fabric Controller is the foundation for Windows Azure’s PaaS
  - Provisions machines
  - Deploys services
  - Configures hardware for services
  - Monitors service and hardware health
  - Performs service healing

Comparing Cloud Platforms

Cloud Platforms:
Leading Vendors and Technologies



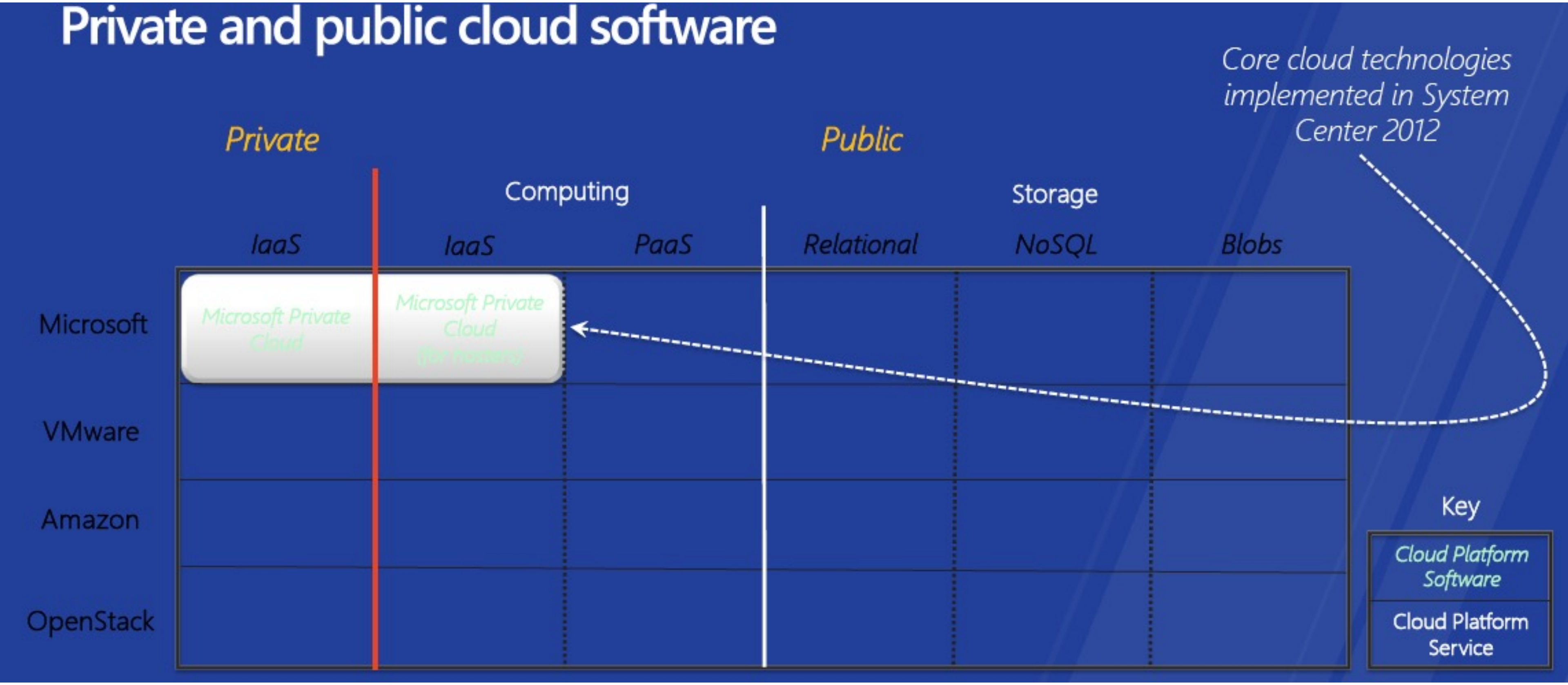
	Private	Public				
		Computing		Storage		
	IaaS	IaaS	PaaS	Relational	NoSQL	Blobs
Microsoft						
VMware						
Amazon						
OpenStack						

Key

Cloud Platform Software

Cloud Platform Service

Microsoft



Private and public cloud software

	Private	Public				
		Computing		Storage		
	IaaS	IaaS	PaaS	Relational	NoSQL	Blobs
Microsoft	Microsoft Private Cloud	Microsoft Private Cloud (for hosters)				
VMware						
Amazon						
OpenStack						

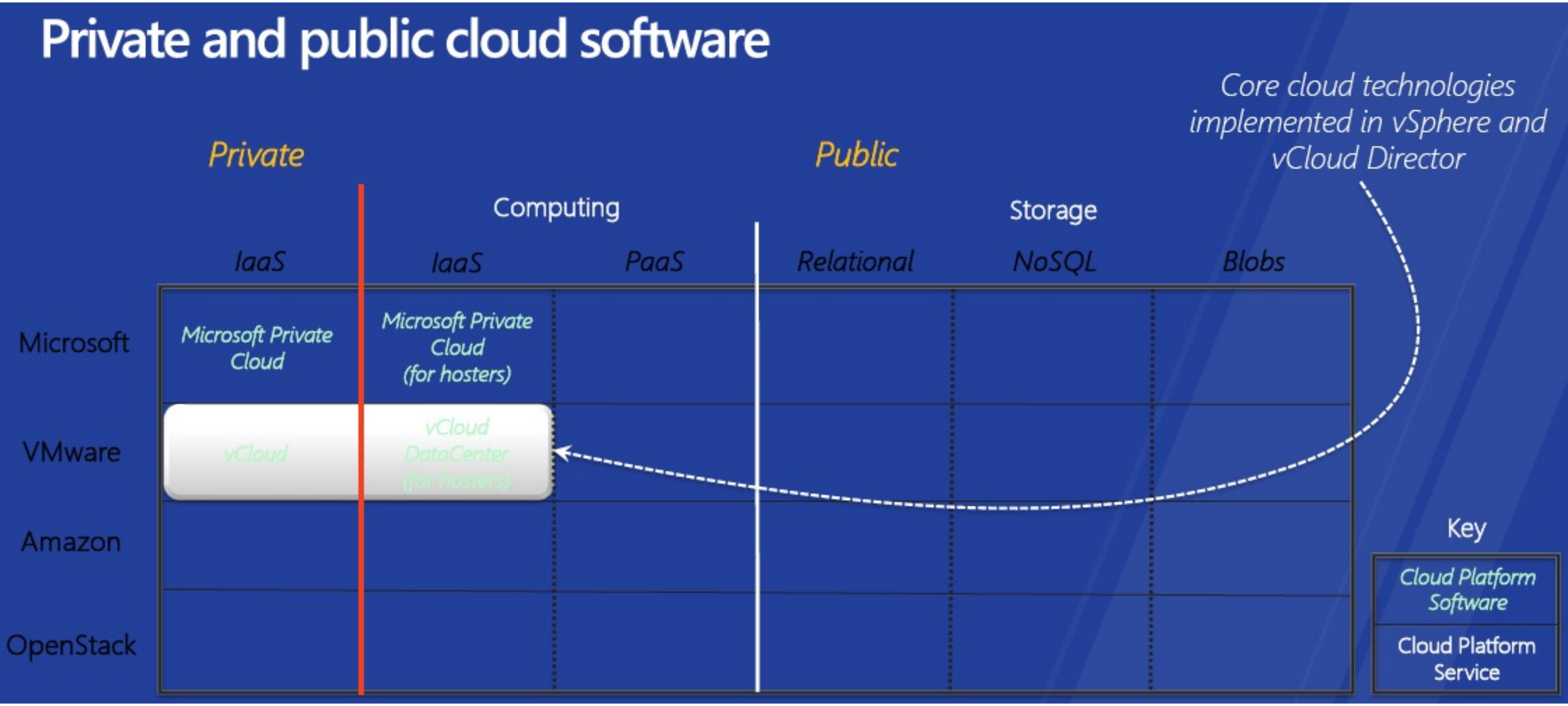
Core cloud technologies implemented in System Center 2012

Key

Cloud Platform Software

Cloud Platform Service

VMware



Private and public cloud software

	Private	Public				
		Computing		Storage		
	IaaS	IaaS	PaaS	Relational	NoSQL	Blobs
Microsoft	*Microsoft Private Cloud*	*Microsoft Private Cloud (for hosters)*				
VMware	*vCloud*	*vCloud DataCenter (for hosters)*				
Amazon						
OpenStack						

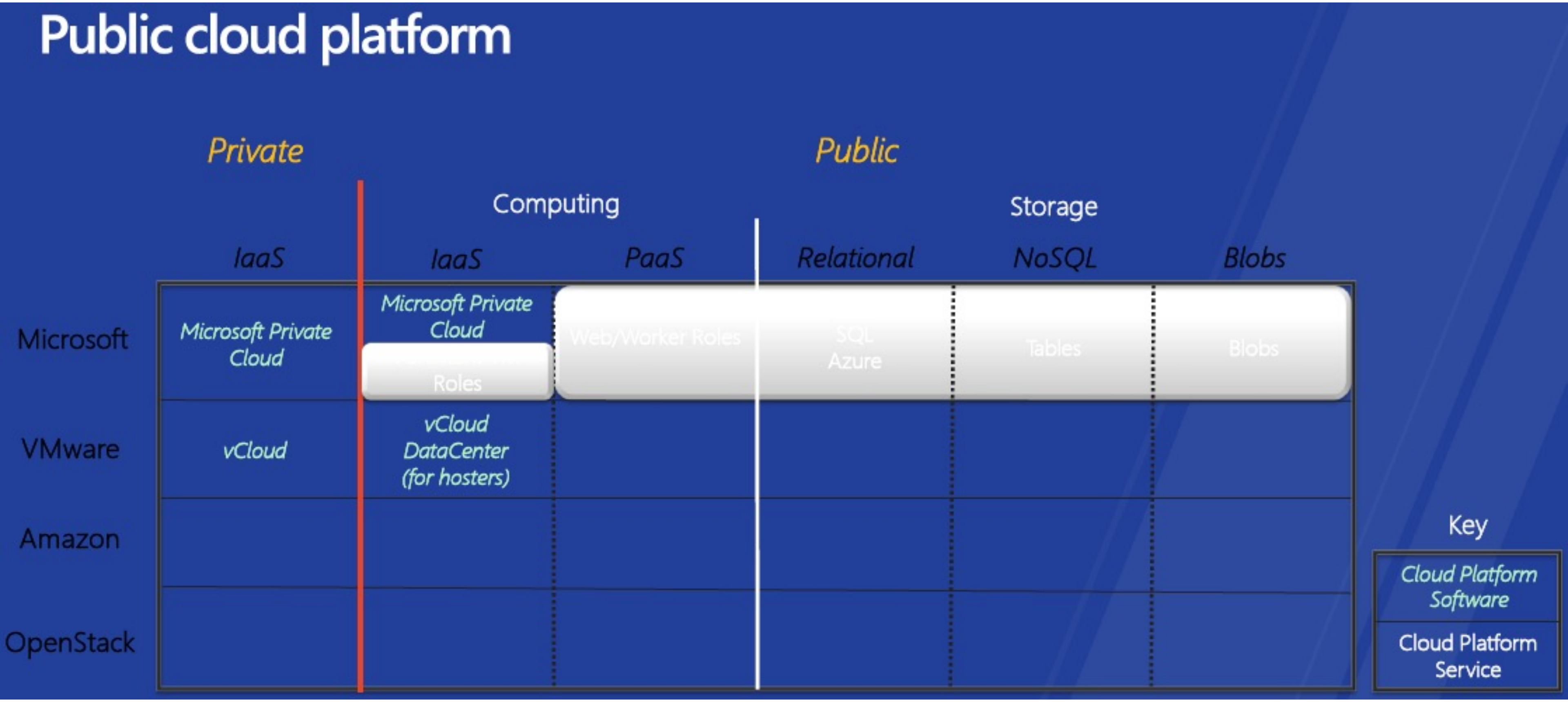
Core cloud technologies implemented in vSphere and vCloud Director

Key

Cloud Platform Software

Cloud Platform Service

Windows Azure



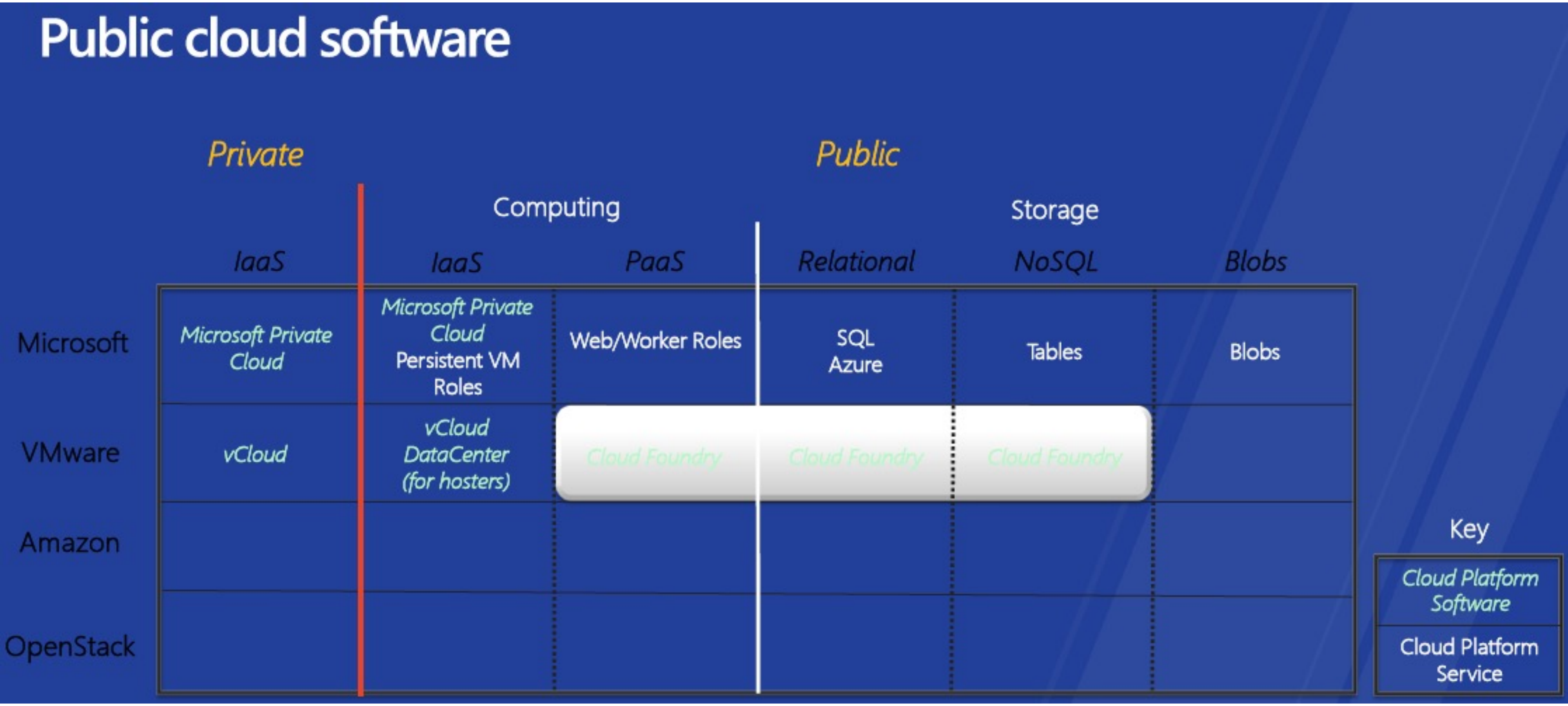
Public cloud platform

	Private	Public				
		Computing		Storage		
	IaaS	IaaS	PaaS	Relational	NoSQL	Blobs
Microsoft	Microsoft Private Cloud	Microsoft Private Cloud [illegible] Roles	Web/Worker Roles	SQL Azure	Tables	Blobs
VMware	vCloud	vCloud DataCenter (for hosters)				
Amazon						
OpenStack						

Key

- *Cloud Platform Software*
- Cloud Platform Service

Vmware Cloud Foundry



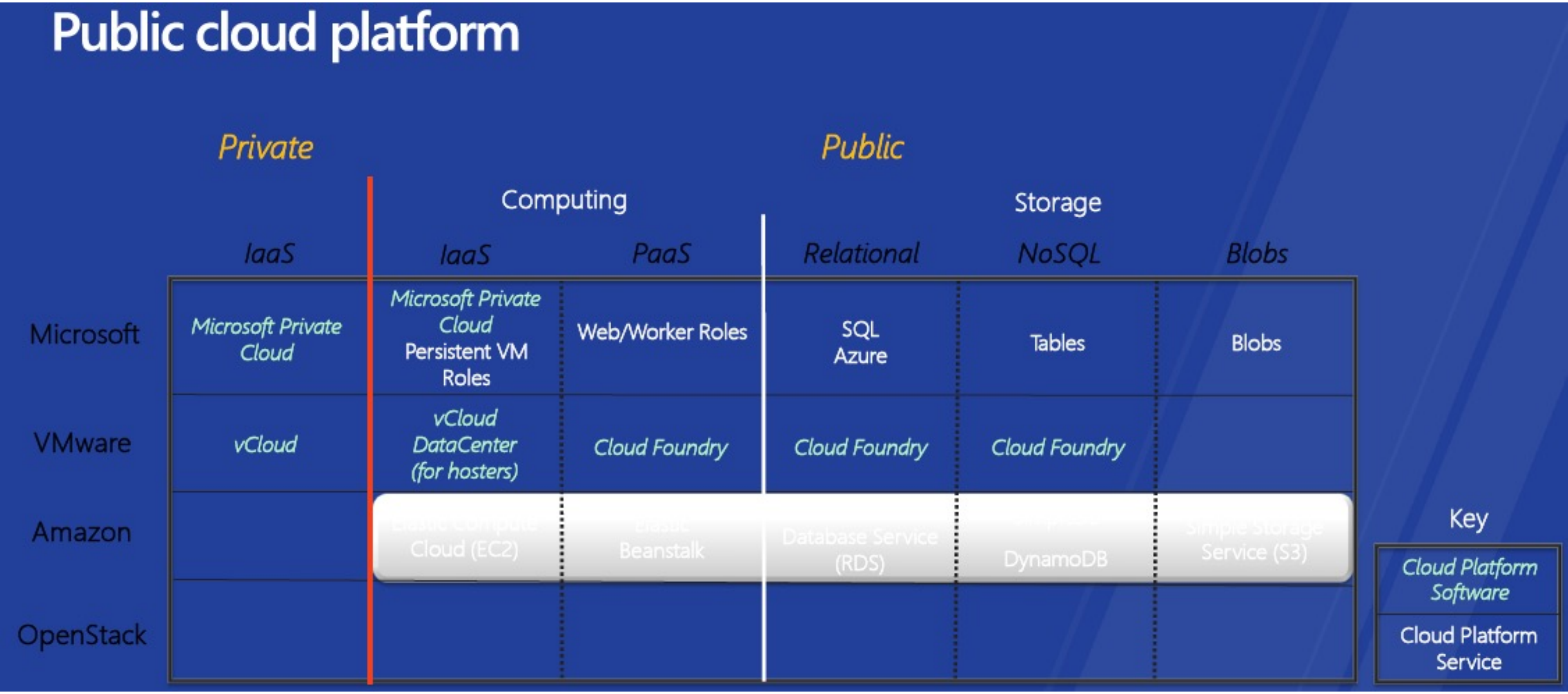
Public cloud software

	Private	*Public*				
		Computing		Storage		
	IaaS	*IaaS*	*PaaS*	*Relational*	*NoSQL*	*Blobs*
Microsoft	*Microsoft Private Cloud*	*Microsoft Private Cloud* Persistent VM Roles	Web/Worker Roles	SQL Azure	Tables	Blobs
VMware	*vCloud*	*vCloud DataCenter (for hosters)*	*Cloud Foundry*	*Cloud Foundry*	*Cloud Foundry*	
Amazon						
OpenStack						

Key

Cloud Platform Software
Cloud Platform Service

Amazon Web Services (AWS)



Public cloud platform

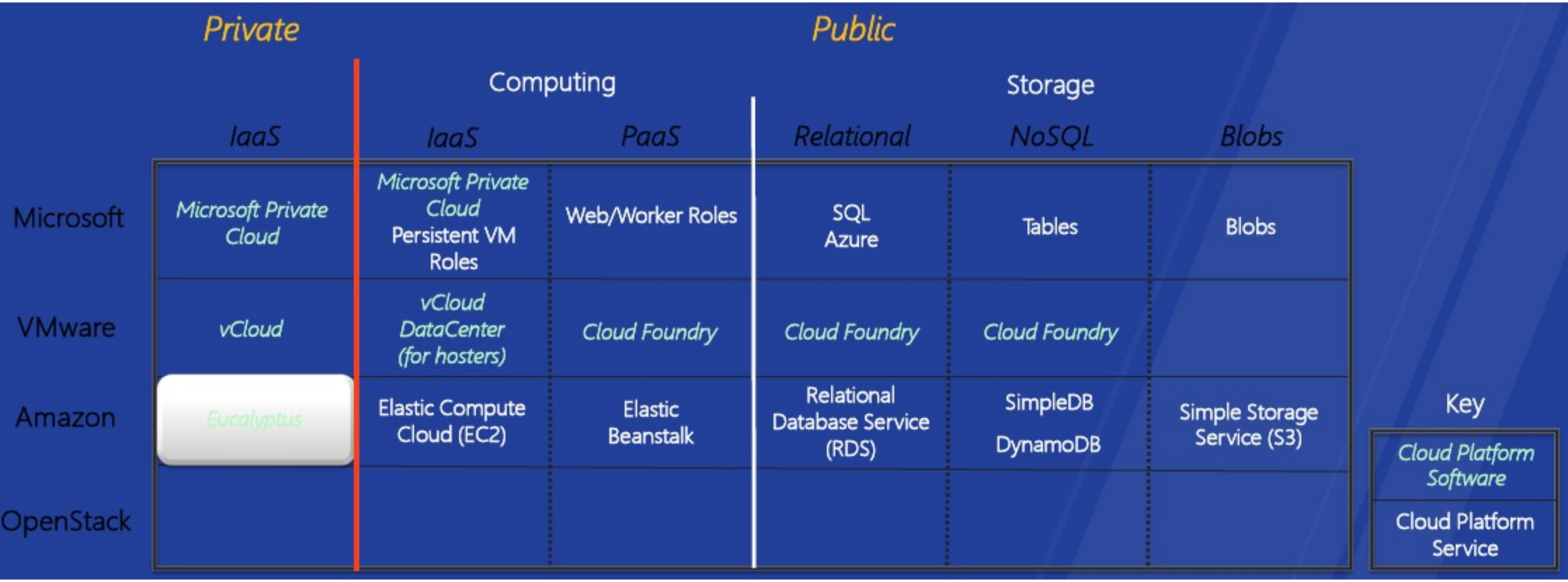
	Private	*Public*				
		Computing		Storage		
	IaaS	*IaaS*	*PaaS*	*Relational*	*NoSQL*	*Blobs*
Microsoft	*Microsoft Private Cloud*	*Microsoft Private Cloud* Persistent VM Roles	Web/Worker Roles	SQL Azure	Tables	Blobs
VMware	*vCloud*	*vCloud DataCenter (for hosters)*	*Cloud Foundry*	*Cloud Foundry*	*Cloud Foundry*	
Amazon		Elastic Compute Cloud (EC2)	Elastic Beanstalk	Database Service (RDS)	DynamoDB	Simple Storage Service (S3)
OpenStack						

Key

Cloud Platform Software

Cloud Platform Service

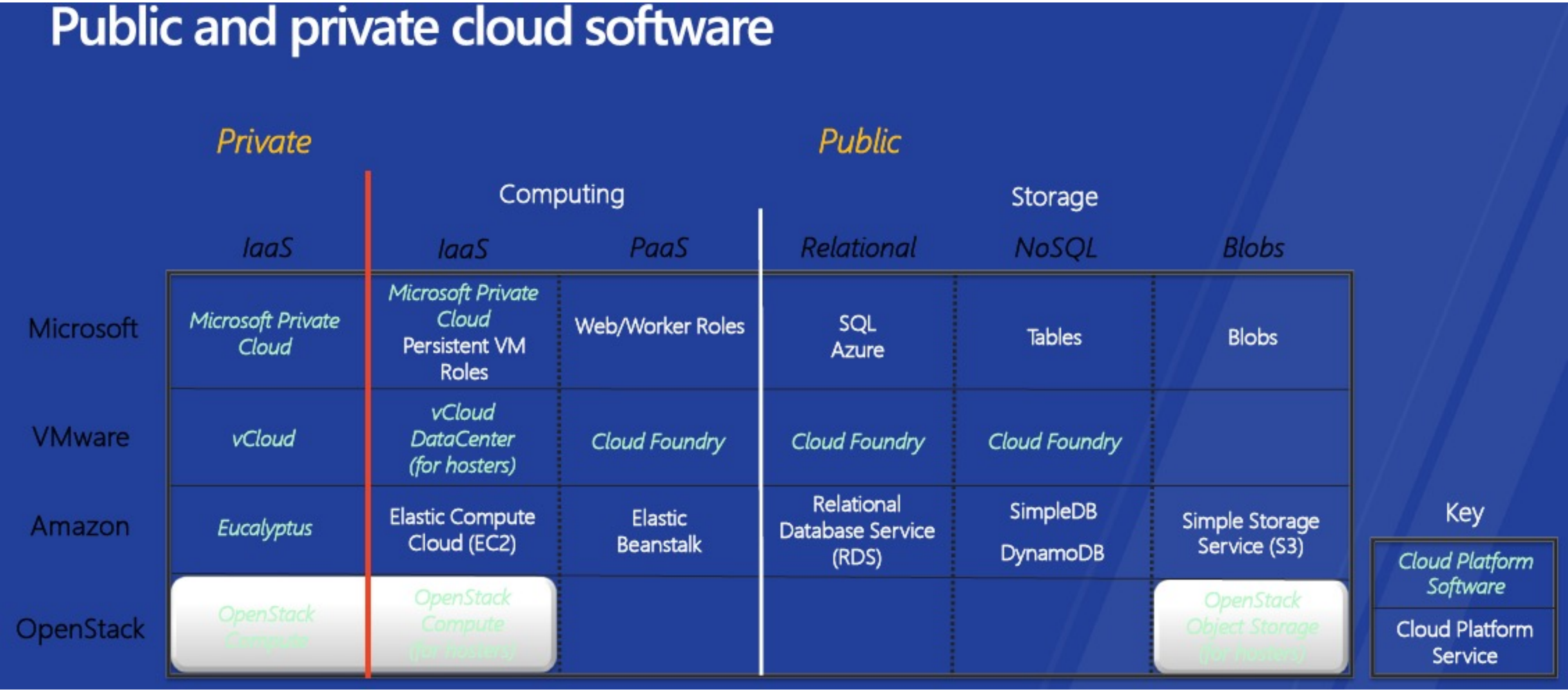
Eucalyptus



	Private	*Public*				
		Computing		Storage		
	IaaS	*IaaS*	*PaaS*	*Relational*	*NoSQL*	*Blobs*
Microsoft	*Microsoft Private Cloud*	*Microsoft Private Cloud* Persistent VM Roles	Web/Worker Roles	SQL Azure	Tables	Blobs
VMware	*vCloud*	*vCloud DataCenter (for hosters)*	*Cloud Foundry*	*Cloud Foundry*	*Cloud Foundry*	
Amazon	*Eucalyptus*	Elastic Compute Cloud (EC2)	Elastic Beanstalk	Relational Database Service (RDS)	SimpleDB DynamoDB	Simple Storage Service (S3)
OpenStack						

Key
Cloud Platform Software
Cloud Platform Service

Openstack

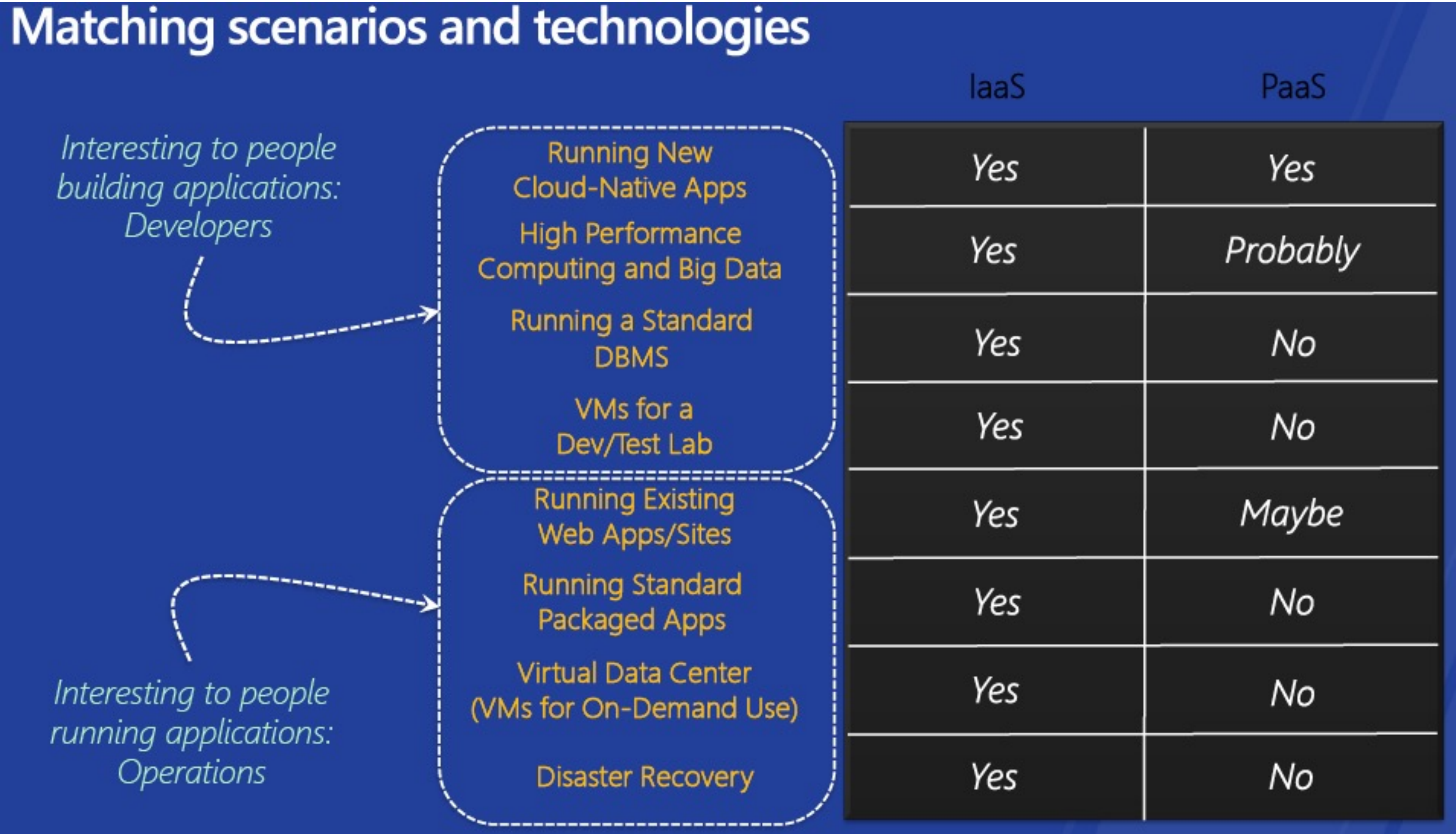


Public and private cloud software

	Private	*Public*				
		Computing		Storage		
	IaaS	*IaaS*	*PaaS*	*Relational*	*NoSQL*	*Blobs*
Microsoft	*Microsoft Private Cloud*	*Microsoft Private Cloud* Persistent VM Roles	Web/Worker Roles	SQL Azure	Tables	Blobs
VMware	*vCloud*	*vCloud DataCenter (for hosters)*	*Cloud Foundry*	*Cloud Foundry*	*Cloud Foundry*	
Amazon	*Eucalyptus*	Elastic Compute Cloud (EC2)	Elastic Beanstalk	Relational Database Service (RDS)	SimpleDB DynamoDB	Simple Storage Service (S3)
OpenStack	*OpenStack Compute*	*OpenStack Compute (for hosters)*				*OpenStack Object Storage (for hosters)*

Key
Cloud Platform Software
Cloud Platform Service

Typical Public Cloud Platform Use Cases



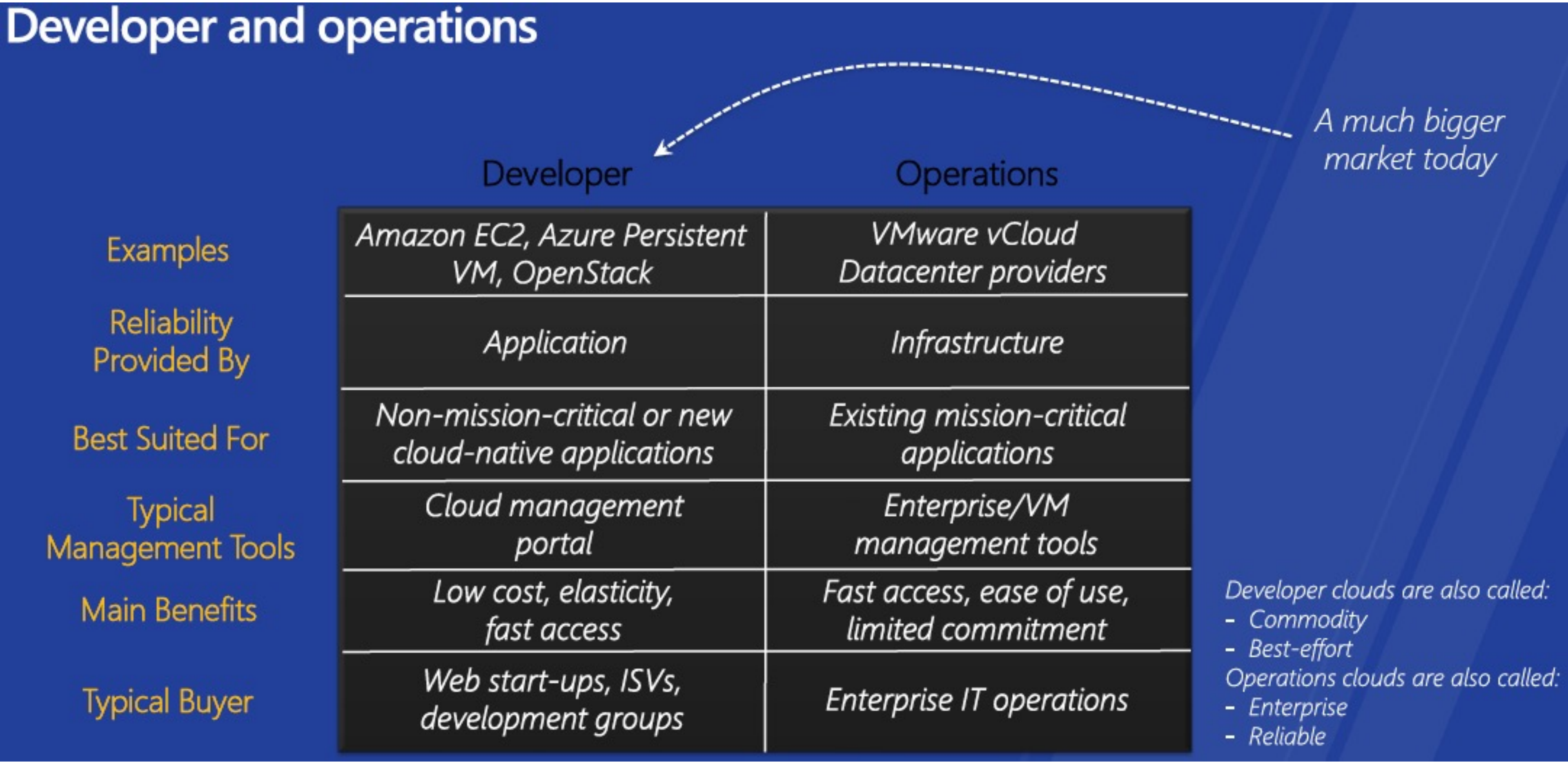
Matching scenarios and technologies

Interesting to people building applications: Developers

Interesting to people running applications: Operations

	IaaS	PaaS
Running New Cloud-Native Apps	*Yes*	*Yes*
High Performance Computing and Big Data	*Yes*	*Probably*
Running a Standard DBMS	*Yes*	*No*
VMs for a Dev/Test Lab	*Yes*	*No*
Running Existing Web Apps/Sites	*Yes*	*Maybe*
Running Standard Packaged Apps	*Yes*	*No*
Virtual Data Center (VMs for On-Demand Use)	*Yes*	*No*
Disaster Recovery	*Yes*	*No*

Categorizing Public IaaS Clouds



Developer and operations

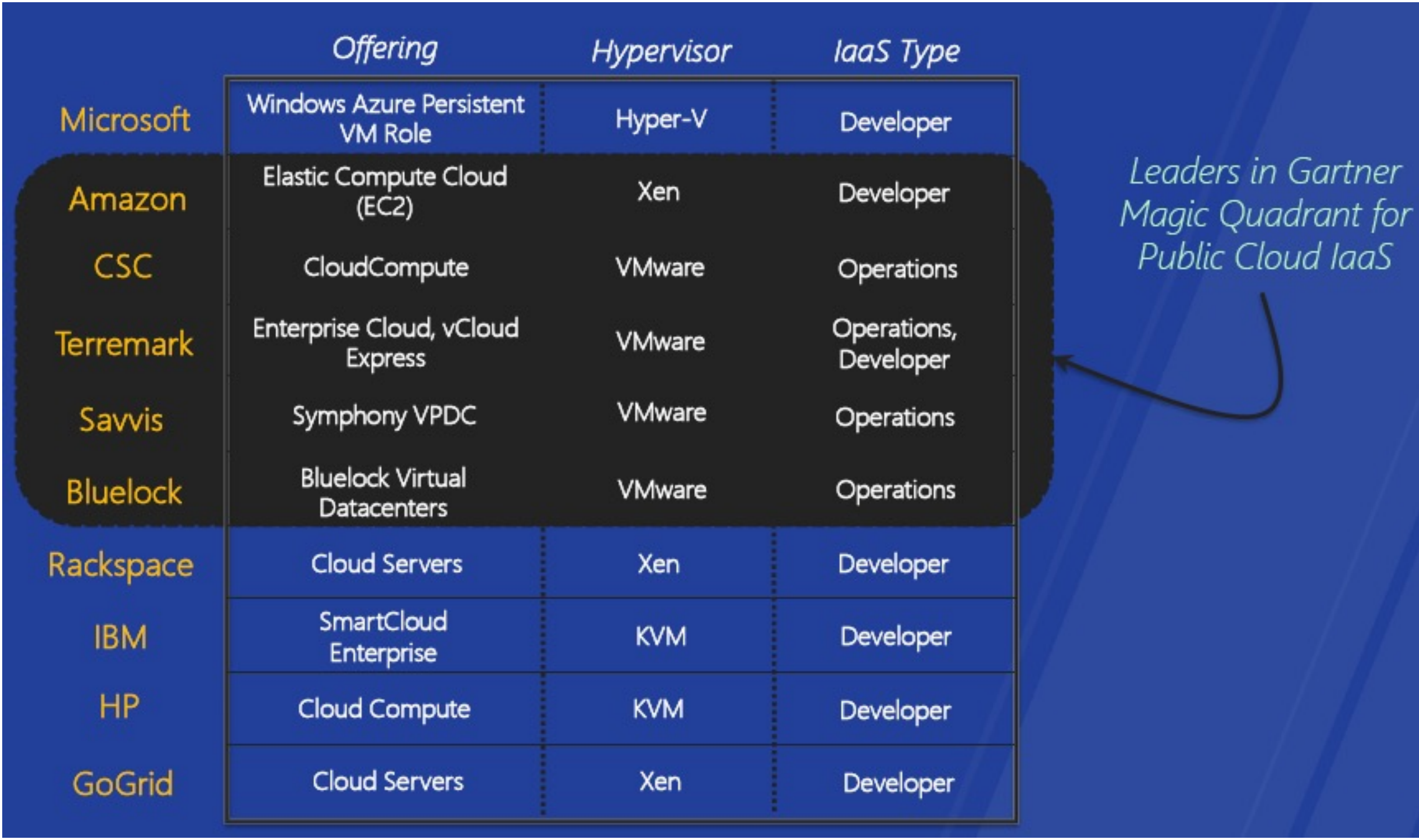
A much bigger market today

	Developer	Operations
Examples	*Amazon EC2, Azure Persistent VM, OpenStack*	*VMware vCloud Datacenter providers*
Reliability Provided By	*Application*	*Infrastructure*
Best Suited For	*Non-mission-critical or new cloud-native applications*	*Existing mission-critical applications*
Typical Management Tools	*Cloud management portal*	*Enterprise/VM management tools*
Main Benefits	*Low cost, elasticity, fast access*	*Fast access, ease of use, limited commitment*
Typical Buyer	*Web start-ups, ISVs, development groups*	*Enterprise IT operations*

Developer clouds are also called:
- *Commodity*
- *Best-effort*

Operations clouds are also called:
- *Enterprise*
- *Reliable*

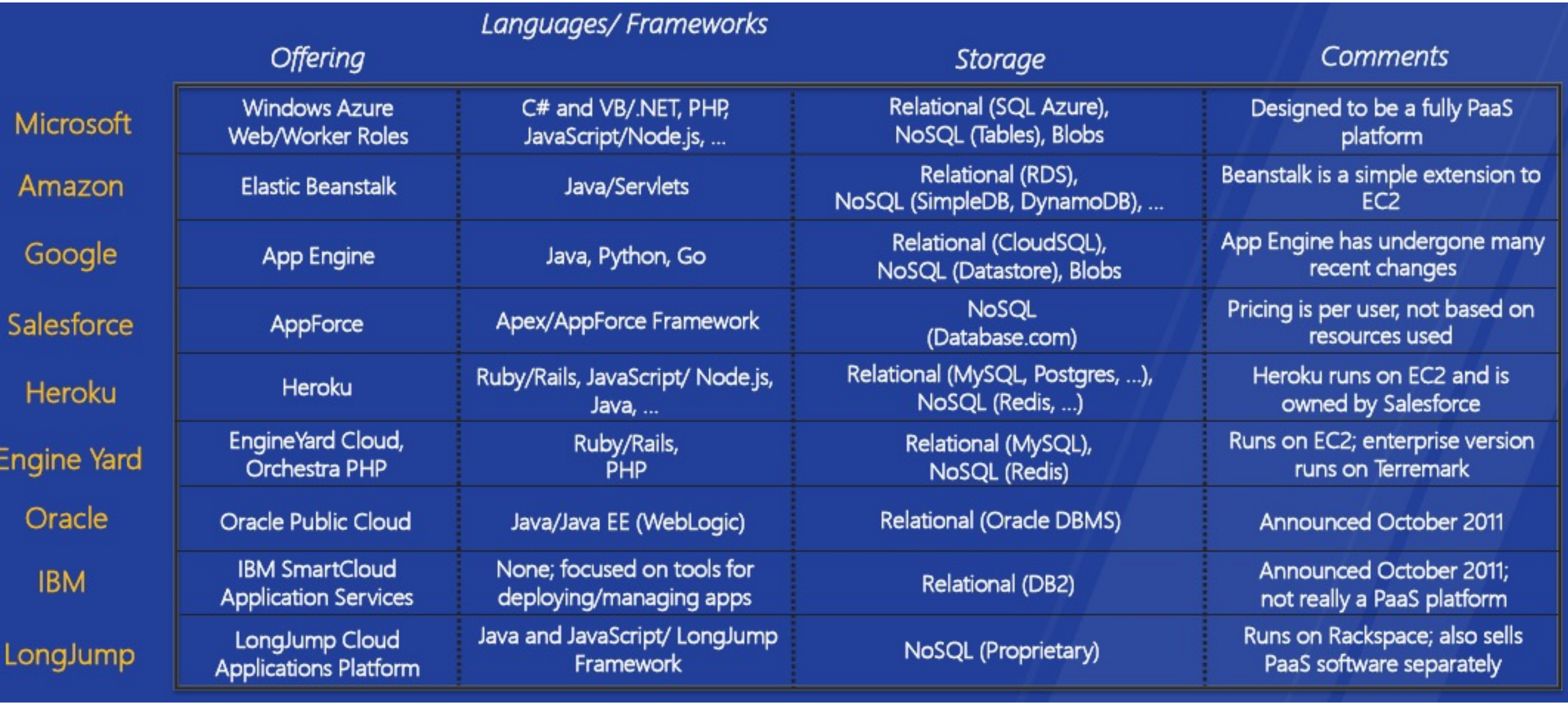
Public IaaS Offerings from Leading Cloud Vendors/Platforms



	Offering	*Hypervisor*	*IaaS Type*
Microsoft	Windows Azure Persistent VM Role	Hyper-V	Developer
Amazon	Elastic Compute Cloud (EC2)	Xen	Developer
CSC	CloudCompute	VMware	Operations
Terremark	Enterprise Cloud, vCloud Express	VMware	Operations, Developer
Savvis	Symphony VPDC	VMware	Operations
Bluelock	Bluelock Virtual Datacenters	VMware	Operations
Rackspace	Cloud Servers	Xen	Developer
IBM	SmartCloud Enterprise	KVM	Developer
HP	Cloud Compute	KVM	Developer
GoGrid	Cloud Servers	Xen	Developer

Leaders in Gartner Magic Quadrant for Public Cloud IaaS

Public PaaS Platform Offerings from Leading Cloud Vendors



	Offering	*Languages/ Frameworks*	*Storage*	*Comments*
Microsoft	Windows Azure Web/Worker Roles	C# and VB/.NET, PHP, JavaScript/Node.js, ...	Relational (SQL Azure), NoSQL (Tables), Blobs	Designed to be a fully PaaS platform
Amazon	Elastic Beanstalk	Java/Servlets	Relational (RDS), NoSQL (SimpleDB, DynamoDB), ...	Beanstalk is a simple extension to EC2
Google	App Engine	Java, Python, Go	Relational (CloudSQL), NoSQL (Datastore), Blobs	App Engine has undergone many recent changes
Salesforce	AppForce	Apex/AppForce Framework	NoSQL (Database.com)	Pricing is per user, not based on resources used
Heroku	Heroku	Ruby/Rails, JavaScript/ Node.js, Java, ...	Relational (MySQL, Postgres, ...), NoSQL (Redis, ...)	Heroku runs on EC2 and is owned by Salesforce
Engine Yard	EngineYard Cloud, Orchestra PHP	Ruby/Rails, PHP	Relational (MySQL), NoSQL (Redis)	Runs on EC2; enterprise version runs on Terremark
Oracle	Oracle Public Cloud	Java/Java EE (WebLogic)	Relational (Oracle DBMS)	Announced October 2011
IBM	IBM SmartCloud Application Services	None; focused on tools for deploying/managing apps	Relational (DB2)	Announced October 2011; not really a PaaS platform
LongJump	LongJump Cloud Applications Platform	Java and JavaScript/ LongJump Framework	NoSQL (Proprietary)	Runs on Rackspace; also sells PaaS software separately

Research Paper on Comparing Cloud Service Providers

- Research by Duke and Microsoft to compare cloud providers in 2010

- A. Li, X. Yang, S. Kandula, and M. Zhang.

CloudCmp: Comparing Public Cloud Providers. In ACM *Internet Measurement Conference,* 2010.

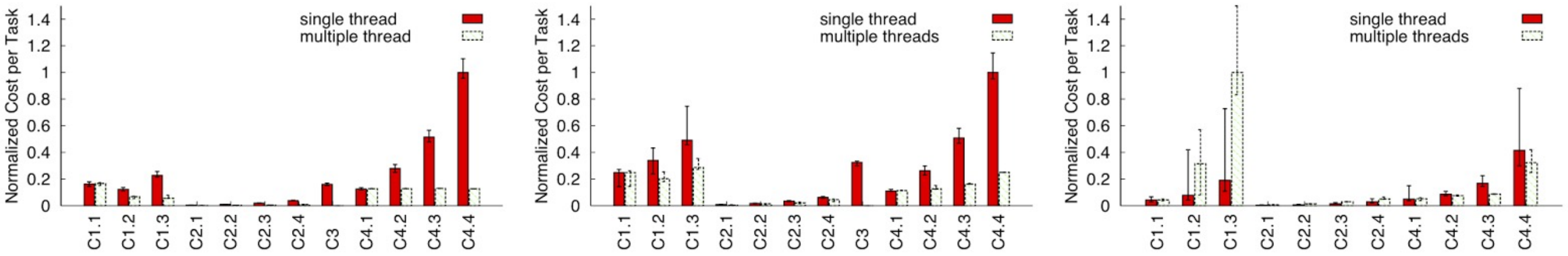
Comparison Methodology

- Test the performance of IaaS and PaaS providers
  - SaaS cannot be tested as it is too widely varied and using these benchmarks doesn't make sense
- Benchmark the service:
  - Per-Task Monetary Cost
  - Network Performance
  - Persistent Storage
  - Webpage Load Times

Which Providers?

- Amazon Web Services (C1)
    - Includes Beanstalk and EC2
- Rackspace CloudServers (C2)
- Google App Engine (C3)
- Microsoft Azure (C4)
- Not all providers offer all services, so some will not have values for certain benchmarks

Per-Task Monetary Cost



(a) CPU (b) Memory (c) Disk I/O

Figure 2: The per-task monetary cost on each type of cloud instance.

Amazon AWS C2: Rackspace CloudServers

C3: Google App Engine C4: Microsoft Azure

- Verdict:
  - Rackspace is the most cost-friendly provider
  - Microsoft Azure is the most expensive to use

Inter-Datacenter TCP Throughput

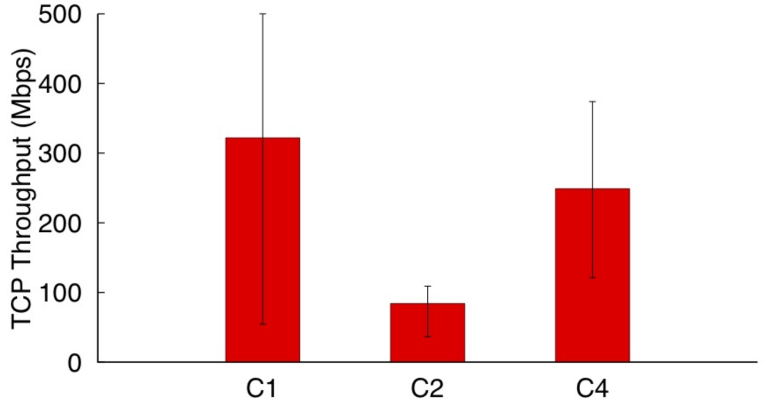
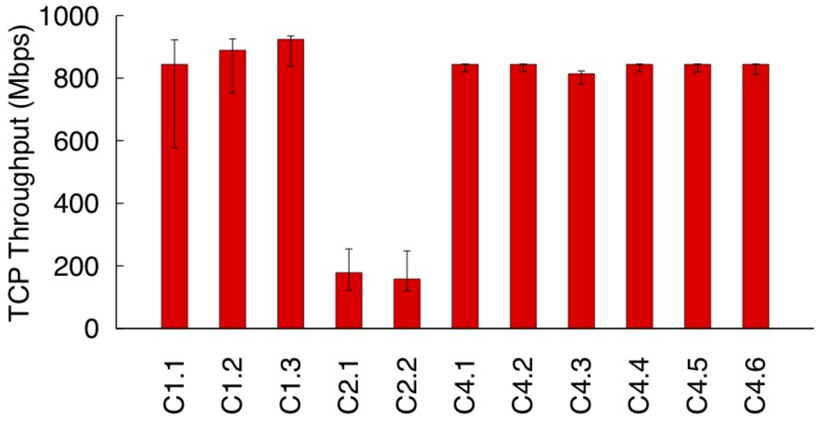


Figure 11: The TCP throughput between two different US data centers of a cloud provider.

C1: Amazon AWS
C2: Rackspace CloudServers
C4: Microsoft Azure

- Verdict:
  - Amazon has highest throughput between datacenters
  - Rackspace's is lamentably low

Intra-Datacenter TCP Throughput



C1: Amazon AWS
C2: Rackspace CloudServers
C4: Microsoft Azure

Figure 10: The intra-datacenter TCP throughput between two instances in all data centers we measure.

- Decimal indicates different datacenters for each service
- Verdict:
  - Amazon has highest throughput within datacenters
  - Rackspace's is lamentably low

Latency: Round Trip Time

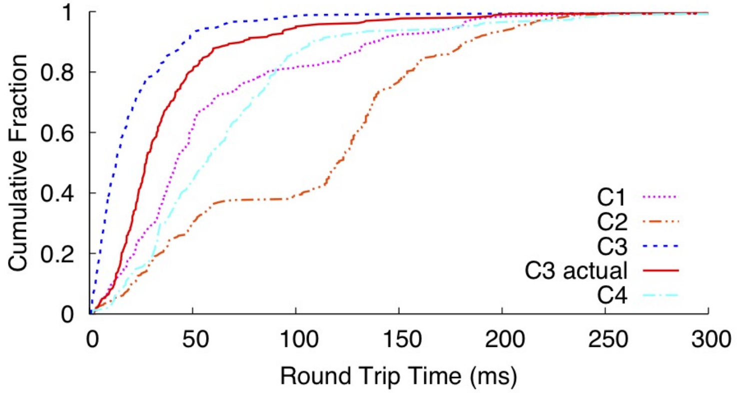


Figure 12: This figure shows the cumulative distribution of the optimal round trip time (RTT) to the instances deployed on a cloud provider from 260 global vantage points. For C_3 we also show the actual RTT from a vantage point to the instance returned by the cloud's DNS load balancing.

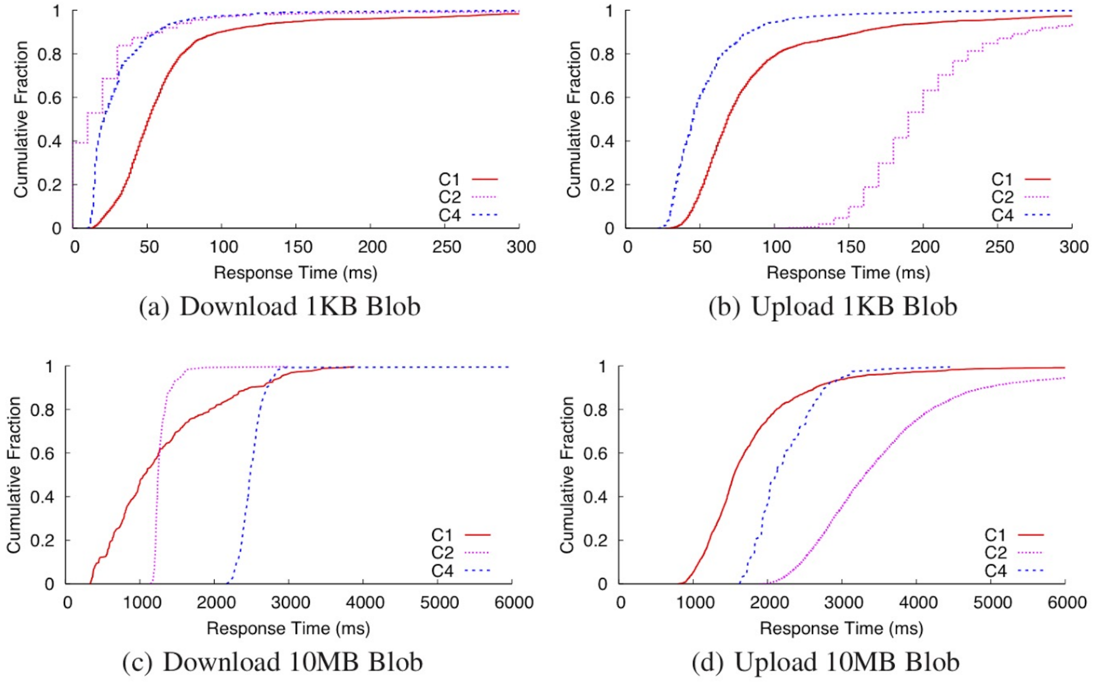
C1: Amazon AWS
C2: Rackspace CloudServers
C3: Google App Engine
C4: Microsoft Azure

- **C3 shows optimal performance and C3-actual shows average attainable performance**
- **Verdict:**
  - **Google is the fastest by far, even on average**
  - **Rackspace has the highest latency**

Persistent Storage

- Cloud providers offer persistent storage to share data between instances
- Two types of storage:
  - Blob Storage for unstructured data, regular files
  - Table Storage for structured data, databases
- Rackspace (C2) doesn't offer a table storage service
- Google App Engine (C3) does not offer a blob storage service

Blob Download/Upload Times



(a) Download 1KB Blob

(b) Upload 1KB Blob

(c) Download 10MB Blob

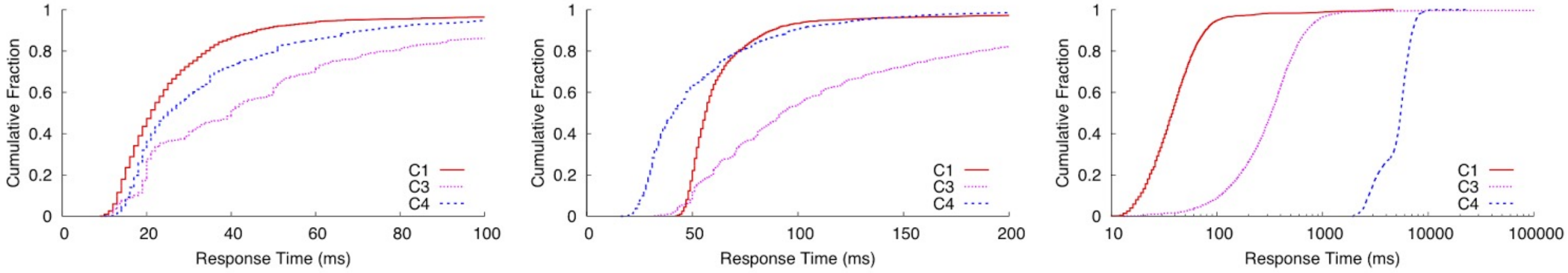
(d) Upload 10MB Blob

Figure 6: **The cumulative distribution of the response time to download or upload a blob using Java-based clients.**

C1: Amazon AWS
C2: Rackspace
CloudServers
C4: Microsoft Azure

- Verdict:
  - For small file sizes(1KB), Microsoft Azure is best
  - For large file sizes(10MB), Amazon AWS is best

Table Storage Operations



(a) Get (b) Put (c) Query

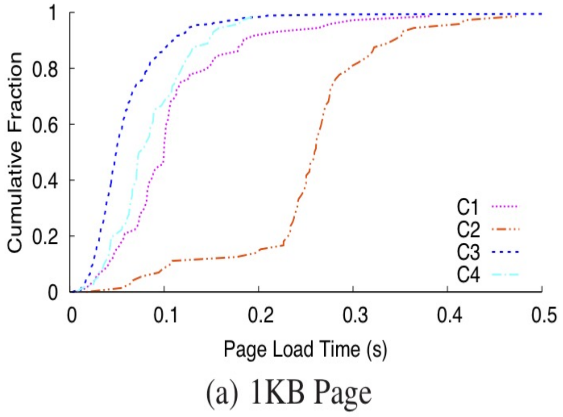
Figure 4: The cumulative distribution of the response time when using the large table with 100K entries. Note that for the query operation, the x-axis is in a logarithmic scale, due to the significant performance gaps between different services.

C1: Amazon AWS C3: Google App Engine

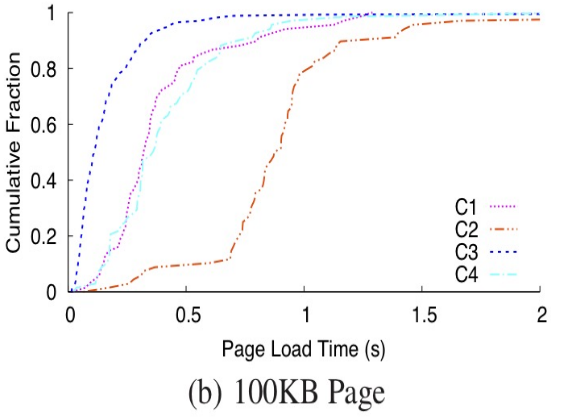
C4: Microsoft Azure

- Verdict:
  - Amazon has fastest table query times by a significant margin
  - Microsoft Azure has noticeably slow table query time

Webpage Load Time



(a) 1KB Page



(b) 100KB Page

Figure 15: The distribution of the page downloading time of our website. We show the results for two different page sizes: 1KB and 100KB.

C1: Amazon AWS

C2: Rackspace CloudServers

C3: Google App Engine

C4: Microsoft Azure

- Verdict:
  - ○ Google App Engine has fastest load times
  - ○ Rackspace is much slower than the rest

And the Winner Is…

- Not immediately clear
- Different providers cater to different needs, no one provider is best at everything
- Google App Engine has the fastest load times, but is less flexible than the other providers
- Amazon AWS has highest throughput and data access times
- Rackspace CloudServers is very cost-effective, but has low perfomance
- Microsoft Azure is rather middle-of-the-road in terms of service, but has a very high price point

NB: MANY THINGS MAY HAVE CHANGED SINCE 2010 !!