Ch 5 – **Cloud Service Models and Cloud Infrastructure**

Contents

- Jericho Cloud Cube Model
- Infrastructure-as-a-Service
- Platform-as-a-Service
- Software-as-a-Service
- Communication-as-a-Service
- Database-as-a-Service
- Cloud Computing at Amazon
- Amazon Web Services
- Cloud Computing from the Google Perspective
- Window Azure and Online Services
- Open Source Software Platforms for Private Clouds.

Jericho Cloud Cube Model

- In January 2004, an IT security association of companies, vendors, government groups, and academics “dedicated to advancing secure business in a global open-network environment” formed the Jericho Forum (www.jerichoforum.org) under the auspices of The Open Group.

- In Feb 2009, they delivered a practical framework geared toward creating the right collaboration-oriented architecture.

Jericho Cloud Cube Model

- Then, in April 2009, the forum published the Jericho Cloud Cube Model version 1.0. From their position paper, the purpose of the Cloud Cube Model is to:
  - Point out that not everything is best implemented in clouds; it may be best to operate some business functions using a traditional non-cloud approach.
  - Explain the different cloud formations that the Jericho Forum has identified.
  - Describe key characteristics, benefi ts and risks of each cloud formation.
  - Provide a framework for exploring in more detail the nature of different cloud formations and the issues that need answering to make them safe and secure places to work in.

Jericho Cloud Cube Model

- The Jericho Cloud Cube Model describes the model for cloud computing as having four “dimensions”:
- **Internal (I)/External (E)** — Defines the physical location of the data. If it is within your own physical boundary then it is Internal, if it is not within your own physical boundary then it is External.

- **Proprietary (P)/Open (O)** — Proprietary means that the organization providing the service is keeping the means of provision under their ownership. Clouds that are Open are using technology that is not proprietary, meaning that there are likely to be more suppliers.

Jericho Cloud Cube Model

- The Jericho Cloud Cube Model describes the model for cloud computing as having four “dimensions”:
- **Perimeterized (Per)/De-perimeterized (D-p) Architectures** — Inside your traditional IT perimeter or outside it? De-Perimeterization has always related to the gradual failure/removal/shrinking/collapse of the traditional silo-based IT perimeter.
- **Insourced/Outsourced** —
- Outsourced: the service is provided by a third party
- Insourced: the service is provided by your own staff under your control.

Jericho Cloud Cube Model

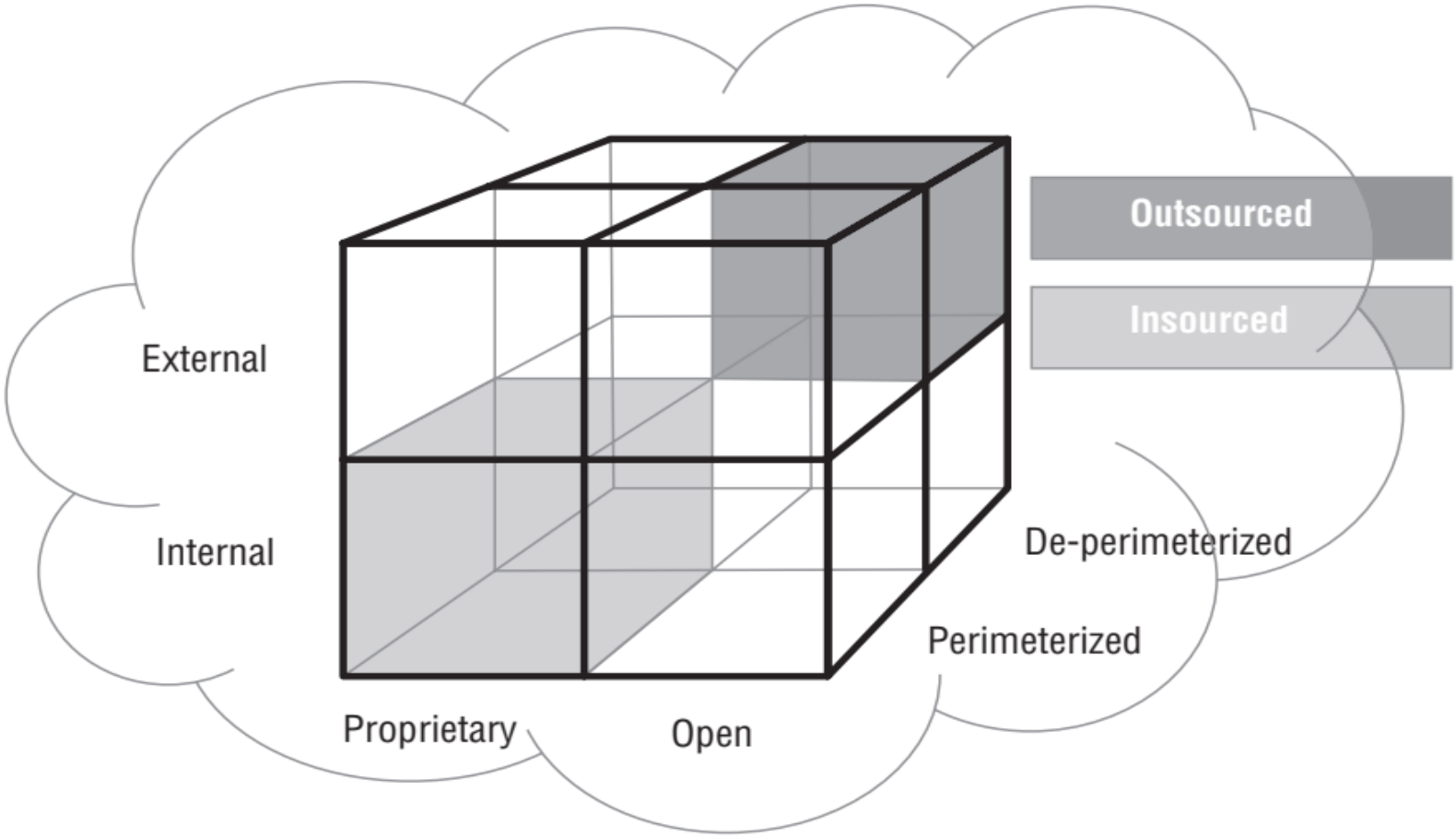


Figure 2-6: Jericho Cloud Cube Model

JCCM- Internal (I)/External (E)

- This is the dimension that defines the physical location of the data: where does the cloud form you want to use exist — inside or outside your organization's boundaries?

- If it is within your own physical boundary then it is Internal.
- If it is not within your own physical boundary then it is External.

- For example, virtualized hard disks in an organization's data center would be internal, while Amazon SC3 would be external at some location "off-site."

JCCM- Proprietary (P)/Open (O)

- This is the dimension that defines the state of ownership of the cloud technology, services, interfaces, etc.
- It indicates the degree of interoperability, as well as enabling “data/application transportability” between your own systems and other cloud forms.
- It also indicates any constraints on being able to share applications.
- Proprietary means that the organization providing the service is keeping the means of provision under their ownership. As a result, when operating in clouds that are proprietary, you may not be able to move to another cloud supplier without significant effort or investment. for example email (SMTP)

JCCM- Perimeterized (Per)/De-perimeterized (D-p) Architectures

- Perimeterized implies continuing to operate within the traditional IT perimeter, often signaled by “network firewalls.”
- De-perimeterized assumes that the system perimeter is architected following the principles outlined in the Jericho Forum’s Commandments and Collaboration oriented Architectures Framework.
- In a de-perimeterized environment an organization can collaborate securely with selected parties (business partner, customer, supplier, outworker) globally over any COA capable network.

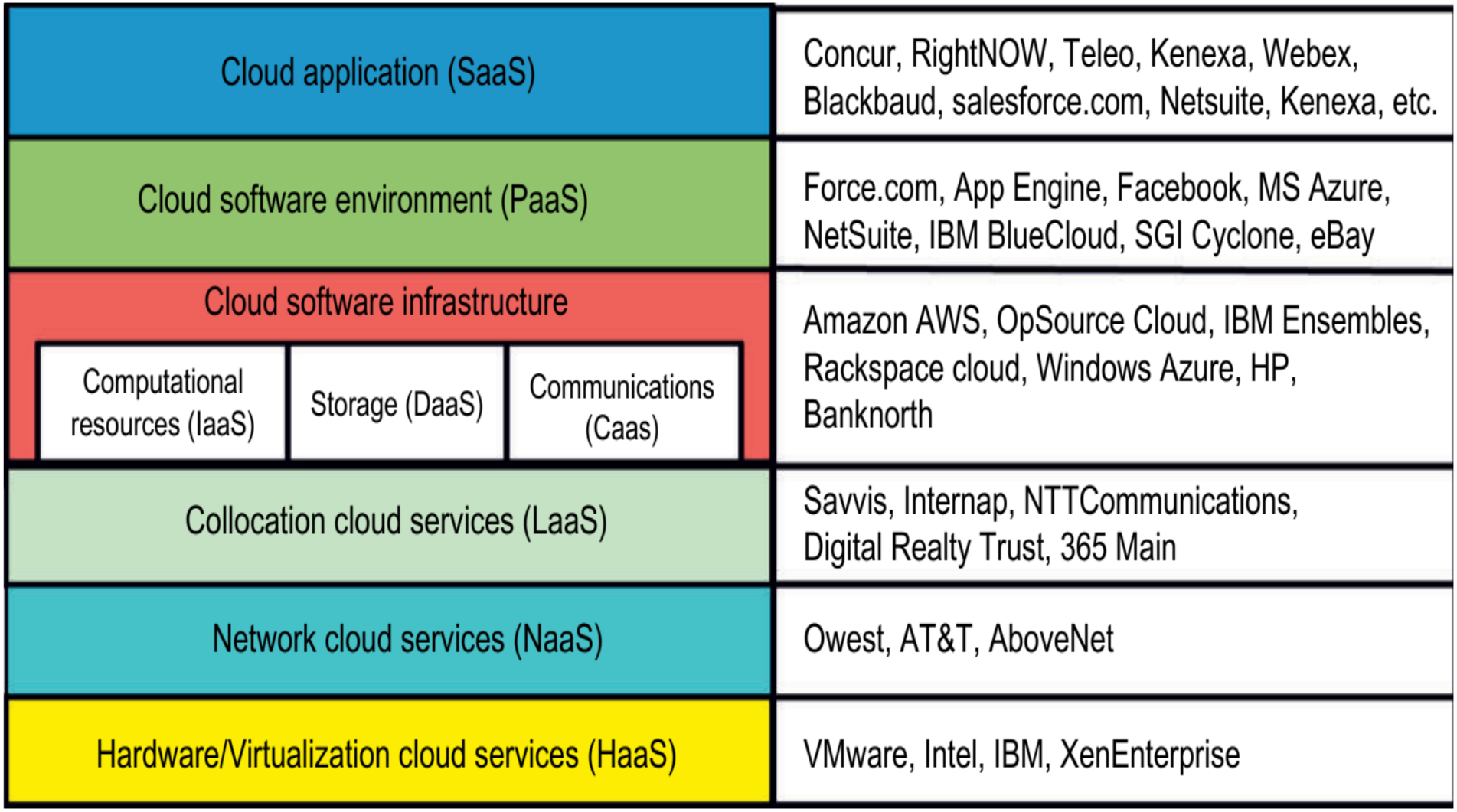
JCCM- Insourced / Outsource

- We define a fourth dimension that has two states in each of the eight cloud forms:
  → “Who do you want running your Clouds?”:
- Outsourced—The service is provided by a third party.
- Insourced—The service is provided by your own staff under your control.

Communication-as-a-Service

- Communication as a Service (CaaS), enables the consumer to utilize Enterprise level VoIP, VPNs, PBX and Unified Communications without the costly investment of purchasing, hosting and managing the infrastructure.
- With the TELCO/ISP cloud service provider responsible for the management and running of these services also, the other advantage the consumer has is that they needn’t require their own trained personnel, bringing significant OPEX as well as CAPEX costs.

Communication-as-a-Service



Database-as-a-Service

- Database as a service (DBaaS) is a cloud computing service model that provides users with some form of access to a database without the need for setting up physical hardware, installing software or configuring for performance.
- All of the administrative tasks and maintenance are taken care of by the service provider so that all the user or application owner needs to do is use the database. Of course, if the customer opts for more control over the database, this option is available and may vary depending on the provider.

Database-as-a-Service

- DBaaS consists of a database manager component, which controls all underlying database instances via an API.
- This API is accessible to the user via a management console, usually a web application, which the user may use to manage and configure the database and even provision or deprovision database instances.
- DBaaS- the ability to leverage the services of a remotely hosted database, sharing it with other users, and having it logically function as if the database were local.

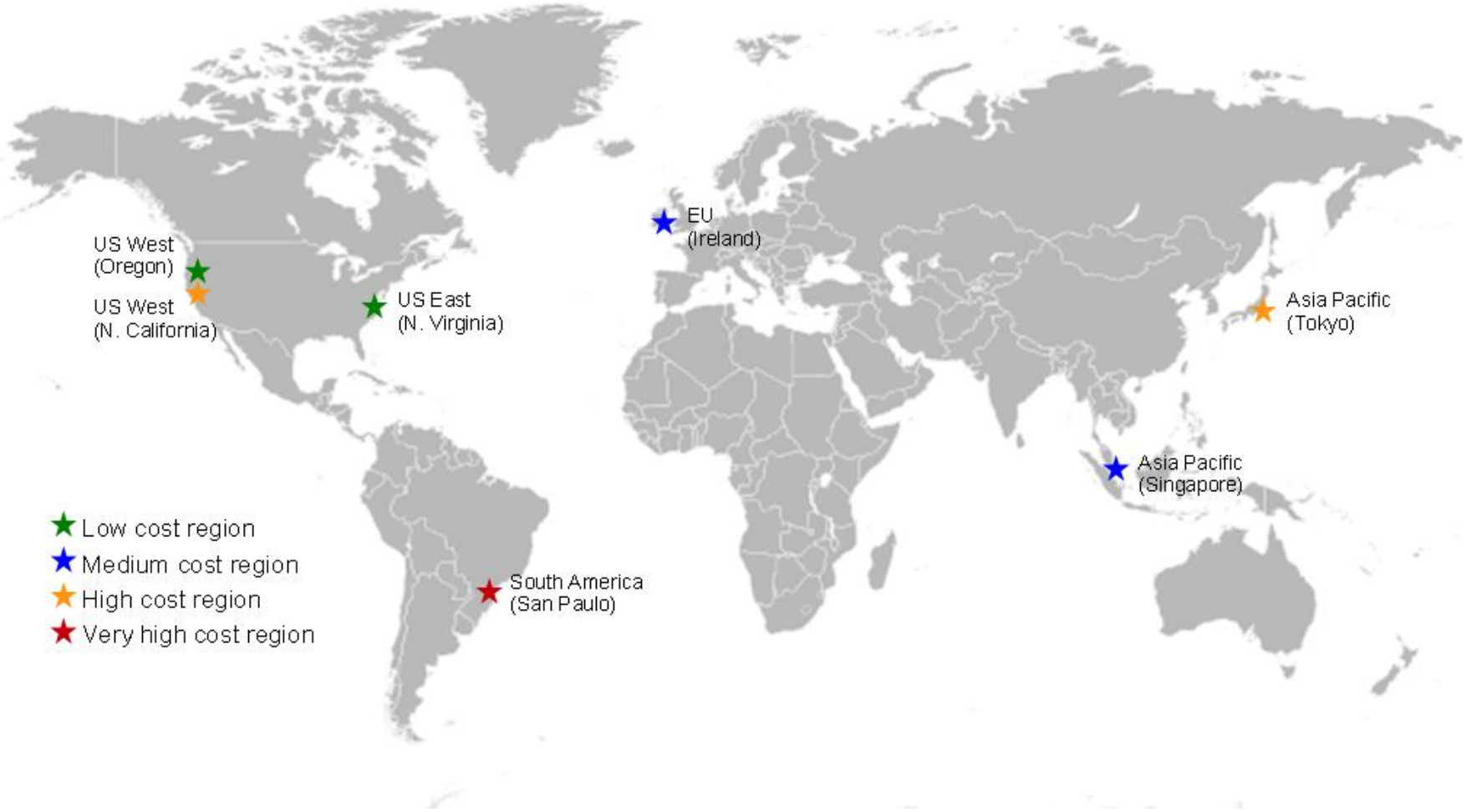
Cloud computing at Amazon

- Amazon introduced a computing platform that has changed the face of computing.
- First, it installed a powerful computing infrastructure to sustain its core business, e-commerce, selling a variety of goods ranging from books and CDs to good foods and home appliances.
- Then Amazon discovered that this infrastructure could be further extended to provide affordable and easy-to-use resources for enterprise computing as well as computing for the masses.

Cloud computing at Amazon

- In this model the cloud service provider offers an infrastructure consisting of compute and storage servers interconnected by high-speed networks that support a set of services to access these resources.
- An application developer is responsible for installing applications on a platform of his or her choice and managing the resources provided by Amazon.
- It is reported that in 2012, businesses in 200 countries used the *AWS*, demonstrating the international appeal of this computing paradigm.
- A significant number of large corporations as well as start-ups take advantage of computing services supported by the *AWS* infrastructure.

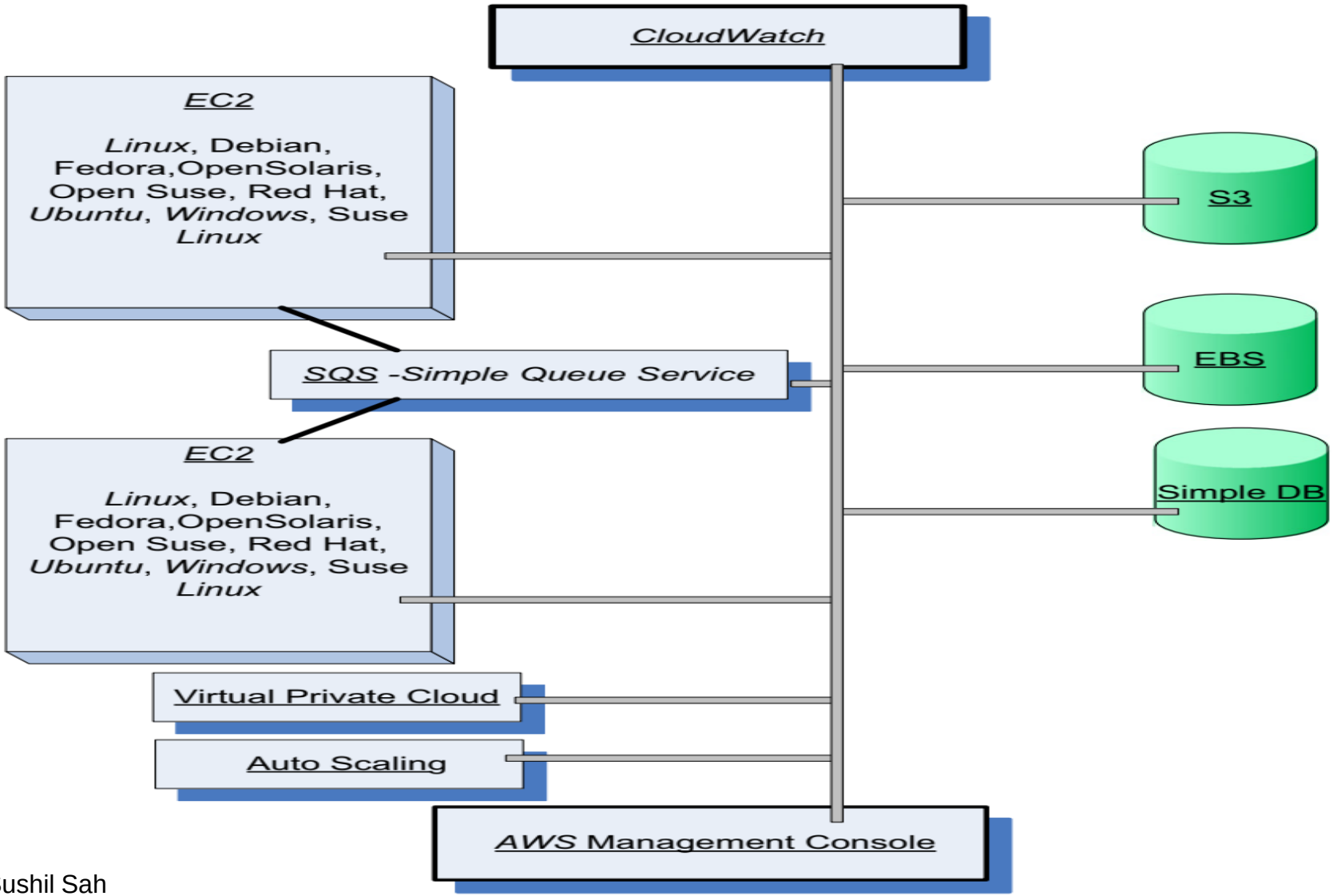
Cloud computing at Amazon



Amazon Web Services

- Amazon was the first provider of cloud computing; it announced a limited public beta release of its Elastic Computing platform called *EC2* in August 2006.
- Figure next shows the palette of *AWS* services accessible via the *Management Console* in late 2011.
- *EC2* - web service for launching instances of an application under several operating systems, such as:
  - Several Linux distributions.
  - Microsoft Windows Server 2003 and 2008.
  - OpenSolaris.
  - FreeBSD.
  - NetBSD.

Amazon Web Services



Amazon Web Services

- *AWS Management Console* - allows users to access the services offered by AWS .
- *Elastic Cloud Computing (EC2)* - allows a user to launch a variety of operating systems.
- *Simple Queuing Service (SQS)* - allows multiple *EC2* instances to communicate with one another.
- *Simple Storage Service (S3)*, *Simple DB*, and *Elastic Bloc Storage (EBS)* - storage services.
- *Cloud Watch* - supports performance monitoring.
- *Auto Scaling* - supports elastic resource management.
- *Virtual Private Cloud* - allows direct migration of parallel applications.

Amazon Web Services

- *A user can*
  - Load an *EC2* instance with a custom application environment.
  - Manage network's access permissions.
  - Run the image using as many or as few systems as desired.
- Import virtual machine (VM) images from the user environment to an instance through *VM import*.
- *EC2* instances boot from an AMI (Amazon Machine Image) digitally signed and stored in *S3*.
- *Users can access:*
  - Images provided by Amazon.
  - Customize an image and store it in *S3*.

Amazon Web Services

- *An EC2 instance is characterized by the resources it provides:*
  - VC (Virtual Computers) – virtual systems running the instance.
  - CU (Compute Units) – measure computing power of each system.
  - Memory.
  - I/O capabilities.
- Standard instances: micro (StdM), small (StdS), large (StdL), extra large (StdXL); small is the default.
- High memory instances: high-memory extra large (HmXL), high-memory double extra large (Hm2XL), and high-memory quadruple extra large (Hm4XL).
- High CPU instances: high-CPU extra large (HcpuXL).
- Cluster computing: cluster computing quadruple extra large (Cl4XL).

S3 – Simple Storage System

- Service designed to store large objects; an application can handle an unlimited number of objects ranging in size from 1 byte to 5 TB.
- An object is stored in a bucket and retrieved via a unique, developer-assigned key; a bucket can be stored in a Region selected by the user.
- Supports a minimal set of functions: write, read, and delete; it does not support primitives to copy, to rename, or to move an object from one bucket to another.
- The object names are global.
- S3 maintains for each object: the name, modification time, an access control list, and up to 4 KB of user-defined metadata.

S3 (cont'd)

- Authentication mechanisms ensure that data is kept secure.
- Objects can be made public, and rights can be granted to other users.
- *S3* computes the MD5 of every object written and returns it in a field called ETag.
- A user is expected to compute the MD5 of an object stored or written and compare this with the ETag; if the two values do not match, then the object was corrupted during transmission or storage.

Elastic Block Store (EBS)

- Provides persistent block level storage volumes for use with *EC2* instances; suitable for database applications, file systems, and applications using raw data devices.
- A volume appears to an application as a raw, unformatted and reliable physical disk; the range 1 GB -1 TB.
- An *EC2* instance may mount multiple volumes, but a volume cannot be shared among multiple instances.
- EBS supports the creation of snapshots of the volumes attached to an instance and then uses them to restart the instance.
- The volumes are grouped together in Availability Zones and are automatically replicated in each zone.

SimpleDB

- Non-relational data store. Supports store and query functions traditionally provided only by relational databases.
- Supports high performance Web applications; users can store and query data items via Web services requests.
- Creates multiple geographically distributed copies of each data item.
- It manages automatically:
  - The infrastructure provisioning.
  - Hardware and software maintenance.
  - Replication and indexing of data items.
  - Performance tuning.

SQS - Simple Queue Service

- Hosted message queues are accessed through standard SOAP and Query interfaces.
- Supports automated workflows - *EC2* instances can coordinate by sending and receiving SQS messages.
- Applications using SQS can run independently and asynchronously, and do not need to be developed with the same technologies.
- A received message is “locked" during processing; if processing fails, the lock expires and the message is available again.
- Queue sharing can be restricted by IP address and time-of-day.

CloudWatch

- Monitoring infrastructure used by application developers, users, and system administrators to collect and track metrics important for optimizing the performance of applications and for increasing the efficiency of resource utilization.
- Without installing any software a user can monitor either seven or eight pre-selected metrics and then view graphs and statistics for these metrics.
- When launching an Amazon Machine Image (AMI) the user can start the CloudWatch and specify the type of monitoring:
  - Basic Monitoring - free of charge; collects data at five-minute intervals for up to seven metrics.
  - Detailed Monitoring - subject to charge; collects data at one minute interval.

AWS services introduced in 2012

- *Route 53* - low-latency DNS service used to manage user's DNS public records.
- *Elastic MapReduce (EMR)* - supports processing of large amounts of data using a hosted Hadoop running on *EC2*.
- *Simple Workflow Service (SWF)* - supports workflow management; allows scheduling, management of dependencies, and coordination of multiple *EC2* instances.
- *ElastiCache* - enables web applications to retrieve data from a managed in-memory caching system rather than a much slower disk-based database.
- *DynamoDB* - scalable and low-latency fully managed NoSQL database service.

AWS services introduced in 2012 (cont'd)

- *CloudFront* - web service for content delivery.
- *Elastic Load Balancer* - automatically distributes the incoming requests across multiple instances of the application.
- *Elastic Beanstalk* - handles automatically deployment, capacity provisioning, load balancing, auto-scaling, and application monitoring functions.
- *CloudFormation* - allows the creation of a stack describing the infrastructure for an application.

Elastic Beanstalk

- Handles automatically the deployment, capacity provisioning, load balancing, auto-scaling, and monitoring functions.
- Interacts with other services including *EC2*, *S3*, SNS, Elastic Load Balance and AutoScaling.
- The management functions provided by the service are:
  - Deploy a new application version (or rollback to a previous version).
  - Access to the results reported by CloudWatch monitoring service.
  - Email notifications when application status changes or application servers are added or removed.
  - Access to server log files without needing to login to the application servers.
- The service is available using: a Java platform, the PHP server-side description language, or the .NET framework.

SaaS services offered by Google

- *Gmail* - hosts Emails on Google servers and provides a web interface to access the Email.
- *Google docs* - a web-based software for building text documents, spreadsheets and presentations.
- *Google Calendar* - a browser-based scheduler; supports multiple user calendars, calendar sharing, event search, display of daily/weekly/monthly views, and so on.
- *Google Groups* - allows users to host discussion forums to create messages online or via Email.
- *Picasa* - a tool to upload, share, and edit images.
- *Google Maps* - web mapping service; offers street maps, a route planner, and an urban business locator for numerous countries around the world

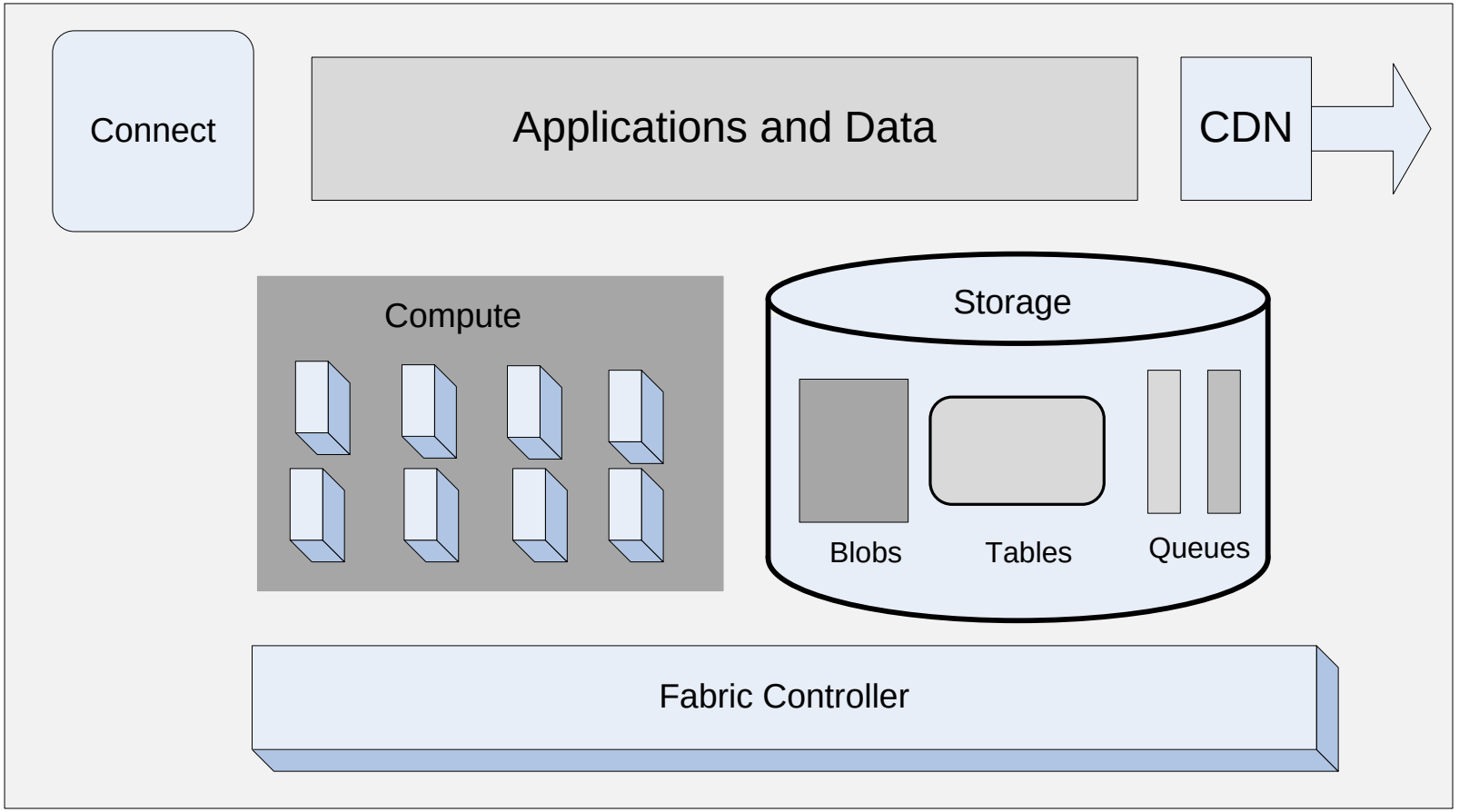
PaaS services offered by Google

- *AppEngine* - a developer platform hosted on the cloud.
  - Initially supported Python, Java was added later.
  - The database for code development can be accessed with GQL (Google Query Language) with a SQL-like syntax.
- *Google Co-op* - allows users to create customized search engines based on a set of facets/categories.
- *Google Drive* - an online service for data storage.
- *Google Base* - allows users to load structured data from different sources to a central repository, a very large, self-describing, semi-structured, heterogeneous database.

PaaS and SaaS services from Microsoft

- *Windows Azure* - an operating system; has 3 components:
  - Compute - provides a computation environment.
  - Storage - for scalable storage.
  - Fabric Controller - deploys, manages, and monitors applications.
- *SQL Azure* - a cloud-based version of the SQL Server.
- *Azure AppFabric*, formerly .NET Services - a collection of services for cloud applications.

Azure


Connect
Applications and Data
CDN
Compute
Storage
Blobs
Tables
Queues
Fabric Controller

Open-source platforms for private clouds

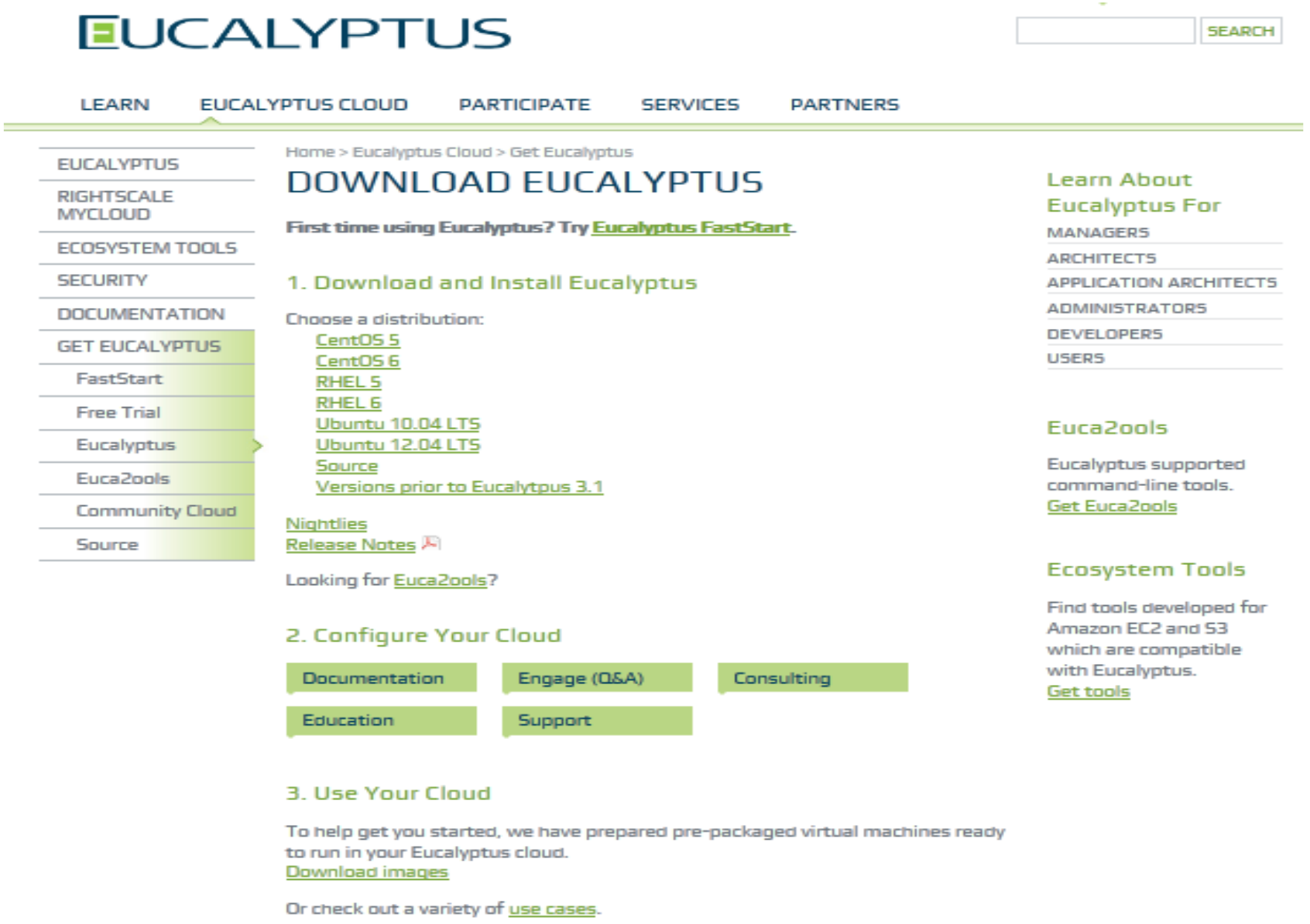
- *Eucalyptus* - can be regarded as an open-source counterpart of Amazon's EC2.

- *Open-Nebula* - a private cloud with users actually logging into the head node to access cloud functions. The system is centralized and its default configuration uses the NFS file system.

- *Nimbus* - a cloud solution for scientific applications based on Globus software; inherits from Globus:
  - The image storage.
  - The credentials for user authentication.
  - The requirement that a running Nimbus process can **ssh** into all compute nodes.

Eucalyptus

- *Virtual Machines* - run under several VMMs including Xen, KVM, and VMware.
- *Node Controller* - runs on server nodes hosting a VM and controls the activities of the node.
- *Cluster Controller* - controls a number of servers.
- *Cloud Controller* - provides the cloud access to end-users, developers, and administrators.
- *Storage Controller* - provides persistent virtual hard drives to applications. It is the correspondent of EBS.
- *Storage Service (Walrus)* - provides persistent storage; similar to S3, it allows users to store objects in buckets.


EUCALYPTUS
SEARCH
LEARN
EUCALYPTUS CLOUD
PARTICIPATE
SERVICES
PARTNERS
EUCALYPTUS
RIGHTSCALE MYCLOUD
ECOSYSTEM TOOLS
SECURITY
DOCUMENTATION
GET EUCALYPTUS
FastStart
Free Trial
Eucalyptus
Euca2ools
Community Cloud
Source
Home > Eucalyptus Cloud > Get Eucalyptus
DOWNLOAD EUCALYPTUS
First time using Eucalyptus? Try Eucalyptus FastStart.
1. Download and Install Eucalyptus
Choose a distribution:
CentOS 5
CentOS 6
RHEL 5
RHEL 6
Ubuntu 10.04 LTS
Ubuntu 12.04 LTS
Source
Versions prior to Eucalytpus 3.1
Nightlies
Release Notes
Looking for Euca2ools?
2. Configure Your Cloud
Documentation
Engage (Q&A)
Consulting
Education
Support
3. Use Your Cloud
To help get you started, we have prepared pre-packaged virtual machines ready to run in your Eucalyptus cloud.
Download images
Or check out a variety of use cases.
Learn About Eucalyptus For
MANAGERS
ARCHITECTS
APPLICATION ARCHITECTS
ADMINISTRATORS
DEVELOPERS
USERS
Euca2ools
Eucalyptus supported command-line tools.
Get Euca2ools
Ecosystem Tools
Find tools developed for Amazon EC2 and S3 which are compatible with Eucalyptus.
Get tools

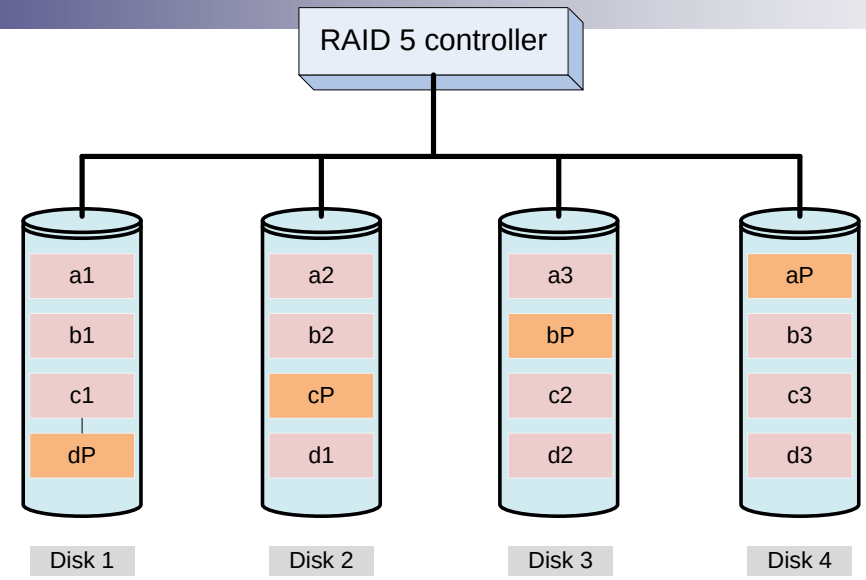
Open-source platforms for private clouds

Table 5: A side-by-side comparison of *Eucalyptus*, *OpenNebula*, and *Nimbus*.

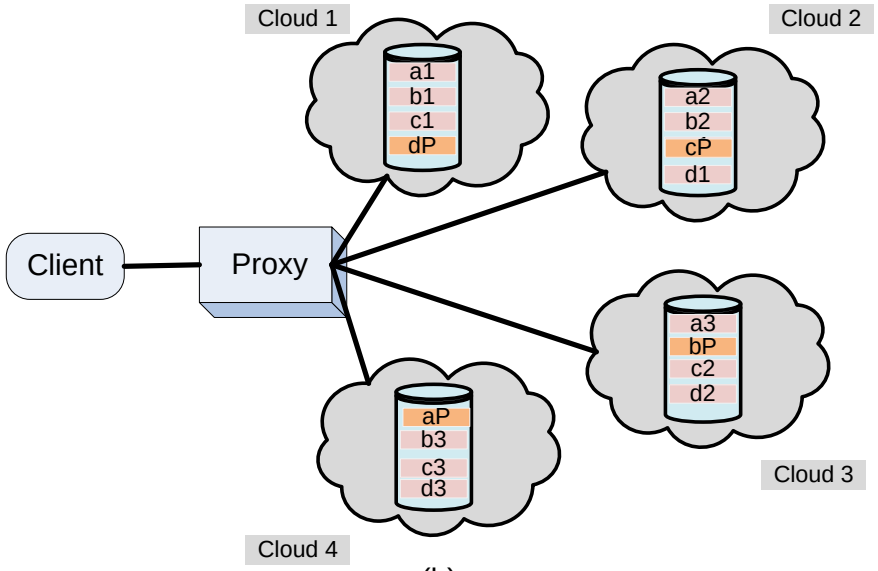
	Eucalyptus	*OpenNebula*	*Nimbus*
Design	Emulate EC2	Customizable	Based on Globus
Cloud type	Private	Private	Public/Private
User population	Large	Small	Large
Applications	All	All	Scientific
Customizability	Administrators limited users	Administrators and users	All but image storage and credentials
Internal security	Strict	Loose	Strict
User access	User credentials	User credentials	x509 credentials
Network access	To cluster controller	-	To each compute node

Cloud storage diversity and vendor lock-in

- Risks when a large organization relies on a single cloud service provider:
  - Cloud services may be unavailable for a short or an extended period of time.
  - Permanent data loss in case of a catastrophic system failure.
  - The provider may increase the prices for service.
- Switching to another provider could be very costly due to the large volume of data to be transferred from the old to the new provider.
- A solution is to replicate the data to multiple cloud service providers, similar to data replication in RAID.


RAID 5 controller
a1
b1
c1
dP
a2
b2
cP
d1
a3
bP
c2
d2
aP
b3
c3
d3
Disk 1
Disk 2
Disk 3
Disk 4

(a)


Cloud 1
Cloud 2
a1
b1
c1
dP
a2
b2
cP
d1
Client
Proxy
a3
bP
c2
d2
aP
b3
c3
d3
Cloud 3
Cloud 4

(b)

Cloud interoperability; the Intercloud

- An Intercloud ➔ a federation of clouds that cooperate to provide a better user experience.
- Is an Intercloud feasible?
- Not likely at this time:
  - There are no standards for either storage or processing.
  - The clouds are based on different delivery models.
  - The set of services supported by these delivery models is large and open; new services are offered every few months.
  - CSPs (Cloud Service Providers) believe that they have a competitive advantage due to the uniqueness of the added value of their services.
  - Security is a major concern for cloud users and an Intercloud could only create new threats.

Energy use and ecological impact

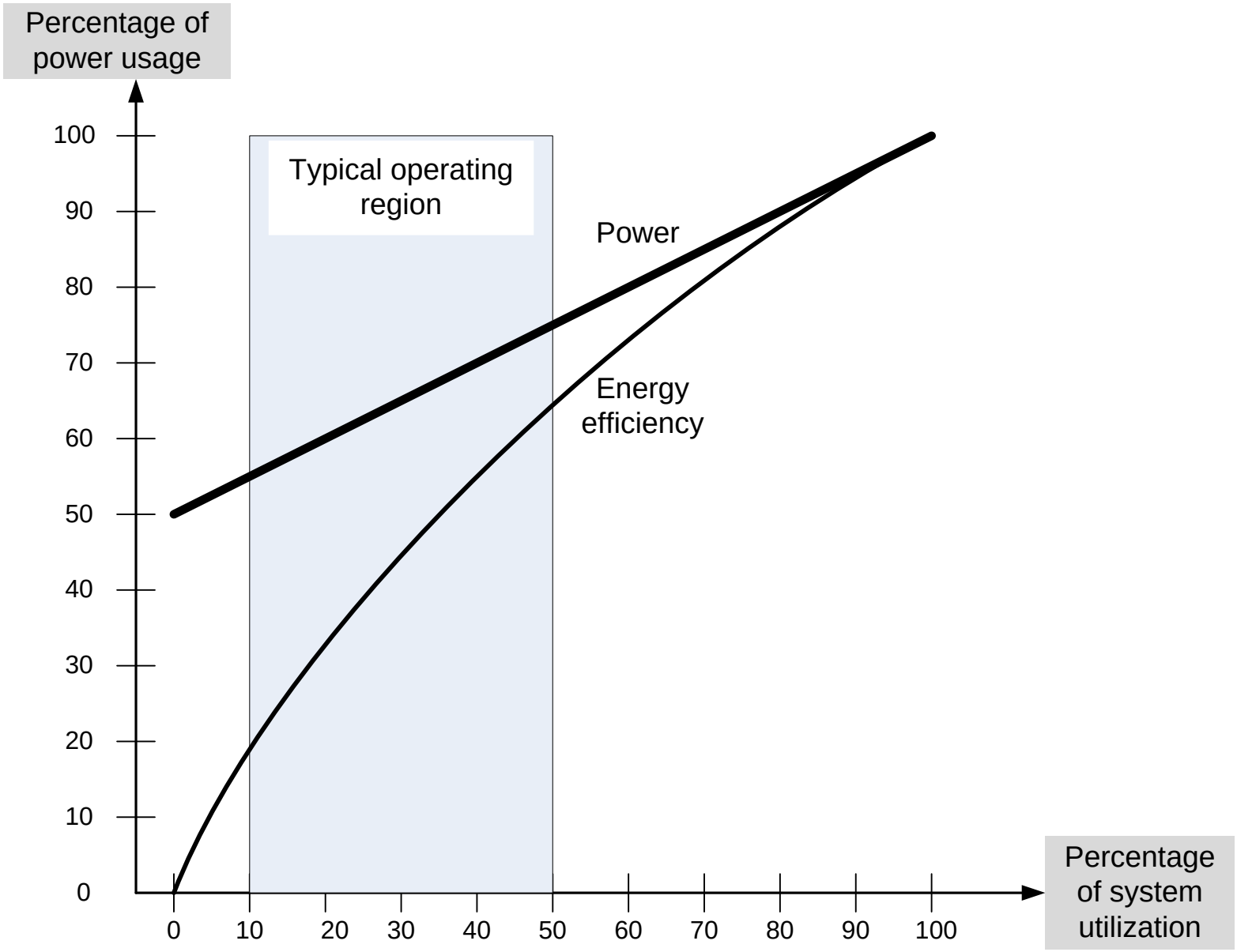
- The energy consumption of large-scale data centers and their costs for energy and for cooling are significant.
- In 2006, the 6,000 data centers in the U.S consumed 61×10^9 KWh of energy, 1.5% of all electricity consumption, at a cost of $4.5 billion.
- The energy consumed by the data centers was expected to double from 2006 to 2011 and peak instantaneous demand to increase from 7 GW to 12 GW.
- The greenhouse gas emission due to the data centers is estimated to increase from 116×10^9 tones of CO_2 in 2007 to 257 tones in 2020 due to increased consumer demand.
- The effort to reduce energy use is focused on computing, networking, and storage activities of a data center.

Energy use and ecological impact (cont'd)

- Operating efficiency of a system is captured by the *performance per Watt of power*.
- The performance of supercomputers has increased 3.5 times faster than their operating efficiency – 7,000% versus 2,000% during the period 1998 – 2007.
- A typical Google cluster spends most of its time within the 10-50% CPU utilization range; there is a mismatch between server workload profile and server energy efficiency.

Energy-proportional systems

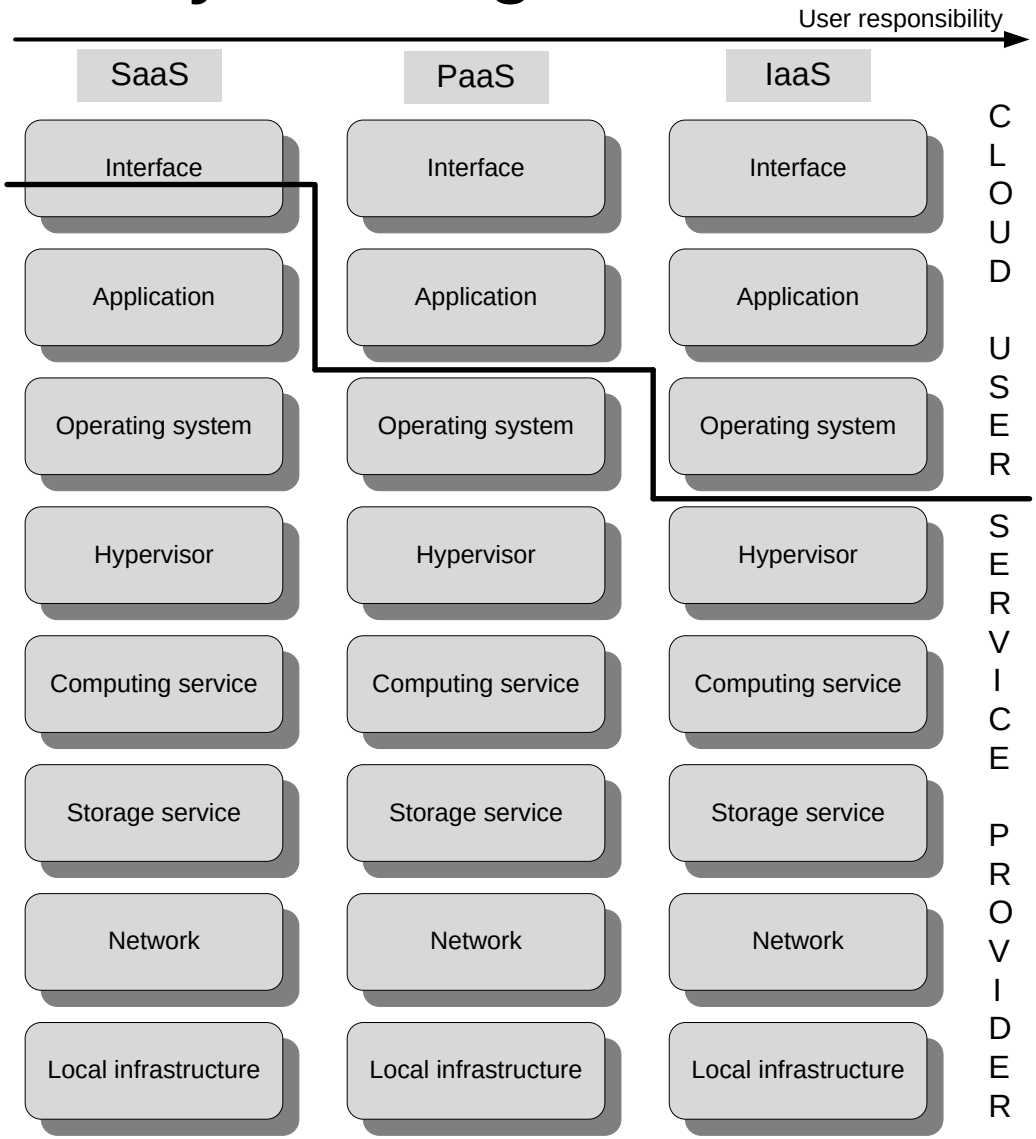
- An energy-proportional system consumes no power when idle, very little power under a light load and, gradually, more power as the load increases.
- By definition, an ideal energy-proportional system is always operating at 100% efficiency.
- Humans are a good approximation of an ideal energy proportional system; about 70 W at rest, 120 W on average on a daily basis, and can go as high as 1,000 – 2,000 W during a strenuous, short time effort.
- Even when power requirements scale linearly with the load, the energy efficiency of a computing system is not a linear function of the load; even when idle, a system may use 50% of the power corresponding to the full load.


Percentage of power usage
100
90
80
70
60
50
40
30
20
10
0
Typical operating region
Power
Energy efficiency
0
10
20
30
40
50
60
70
80
90
100
Percentage of system utilization

Service Level Agreement (SLA)

- SLA - a negotiated contract between the customer and CSP; can be legally binding or informal. Objectives:
  - Identify and define the customer's needs and constraints including the level of resources, security, timing, and QoS.
  - Provide a framework for understanding; a critical aspect of this framework is a clear definition of classes of service and the costs.
  - Simplify complex issues; clarify the boundaries between the responsibilities of clients and CSP in case of failures.
  - Reduce areas of conflict.
  - Encourage dialog in the event of disputes.
  - Eliminate unrealistic expectations.
- Specifies the services that the customer receives, rather than how the cloud service provider delivers the services.

Responsibility sharing between user and CSP



User security concerns

- Potential loss of control/ownership of data.
- Data integration, privacy enforcement, data encryption.
- Data remanence after de-provisioning.
- Multi tenant data isolation.
- Data location requirements within national borders.
- Hypervisor security.
- Audit data integrity protection.
- Verification of subscriber policies through provider controls.
- Certification/Accreditation requirements for a given cloud service.

Reasons driving decision to use public clouds

Reason	Percentage who agree
Improved system reliability and availability	50%
Pay only for what you use	50%
Hardware savings	47%
Software license saving	46%
Lower labor costs	44%
Lower maintenance costs	42%
Reduced IT support needs	40%
Ability to take advantage of the latest functionality	40%
Less pressure on internal resources	39%
Solve problems related to updating/upgrading	39%
Rapid deployment	39%
Ability to scale up resources to meet the needs	39%
Ability to focus on core competencies	38%
Take advantage of the improved economics of scale	37%
Reduced infrastructure management needs	37%
Lower energy costs	29%
Reduced space requirements	26%
Create new revenue streams	23%