Cloud Computing
(PECO8013T)
Unit-III
Platform as a Service (PaaS)

By:- Dr. D.R.Patil

Cloud Computing (PECO8013T)

Teaching Scheme
Lectures : 03 Hrs./week
Credits : 03

Examination Scheme
Term Test : 15 Marks
Teacher Assessment : 20 Marks
End Sem Exam : 65 Marks
Total Marks : 100 Marks

Prerequisite: Information Security, Distributed Computing, Web programming

Course Objectives:

- To capture the state-of-the-art in Cloud Computing technologies and applications.
- To cover a series of current cloud computing technologies, including technologies for Virtualization, Infrastructure as a Service, Platform as a Service and Software as a Service.

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Understand the fundamental concepts of cloud computing.	L2	Understand
CO2	Explore the virtualization at various layers of cloud infrastructure.	L6	Create
CO3	Analyse various cloud security concerns and mechanisms.	L3	Apply
CO4	Assess the need and then migrate to cloud.	L5	Evaluate
CO5	Explain Hadoop File System and role of HDFS in cloud.	L2	Understand

Syllabus

Course Contents

Unit-I — 04 Hrs.

Introduction to Cloud Computing

What is cloud computing?, Properties & Characteristics, Service models, Deployment models.

Unit-II — 08 Hrs.

Infrastructure as a Service (IaaS)

Introduction to IaaS, Resource Virtualization (Server, Storage, Network).

Unit-III — 06 Hrs.

Platform as a Service (PaaS)

Introduction to PaaS, Cloud platforms & Management (Computation and Storage), Case studies.

Unit-IV — 10 Hrs.

Software as a Service (SaaS)

Introduction to SaaS, Web services, Web 2.0, Web OS.

Syllabus

Unit-V — 10 Hrs.

Hadoop

Hadoop distributed file system, distributed computations with MapReduce, Hadoop's data and I/O building blocks. Hadoop in the cloud.

Unit-VI — 05 Hrs.

Cloud Security

Cloud Security reference model, governance and enterprise risk management, compliance and audit management,information management and data security.

Unit-VII — 02 Hrs.

Migration to Cloud

Cloud models suitable for different categories of users, Considerations for choosing applications suitable for cloud, Different phases to adopt the cloud.

Books

Text Books:

1. Raj Buyya, Christian Vecchiola, S. Selvi, "Mastering Cloud Computing", TMH, 2013.
2. RajkimarBuyya, James Broberg, Andrzej Goscinski, "Cloud Computing: Principles and Paradigms", Wiley India, 2013.

Reference Books:

1. Tom white, "Hadoop: The Definitive Guide", Ed. O'Reilly, 2012.
2. Chuck Lam, "Hadoop in action", Dreamtech Press, 2011.
3. Dr. Kumar Saurabh, "Cloud Computing: Insights into New-Era Infrastructure", 1st Edition, Wiley India, 2011.
4. Anthony T. Velte, "Cloud Computing: A Practical approach", TMH, 2009.
5. Halper Fern, Kaufman Marcia, Bloor Robin, Hurwit Judith, "Cloud Computing For Dummies", Wiley India, 2009.

Scheme

Evaluation Scheme:

Theory:

Continuous Assessment (A):

Subject teacher will declare Teacher Assessment criteria at the start of semester.

Continuous Assessment (B):

Conduction of Term Test

The two Term Tests of 15 marks will be conducted under Continuous Assessment(CA) out of which best performance among the two Term Tests will be considered.

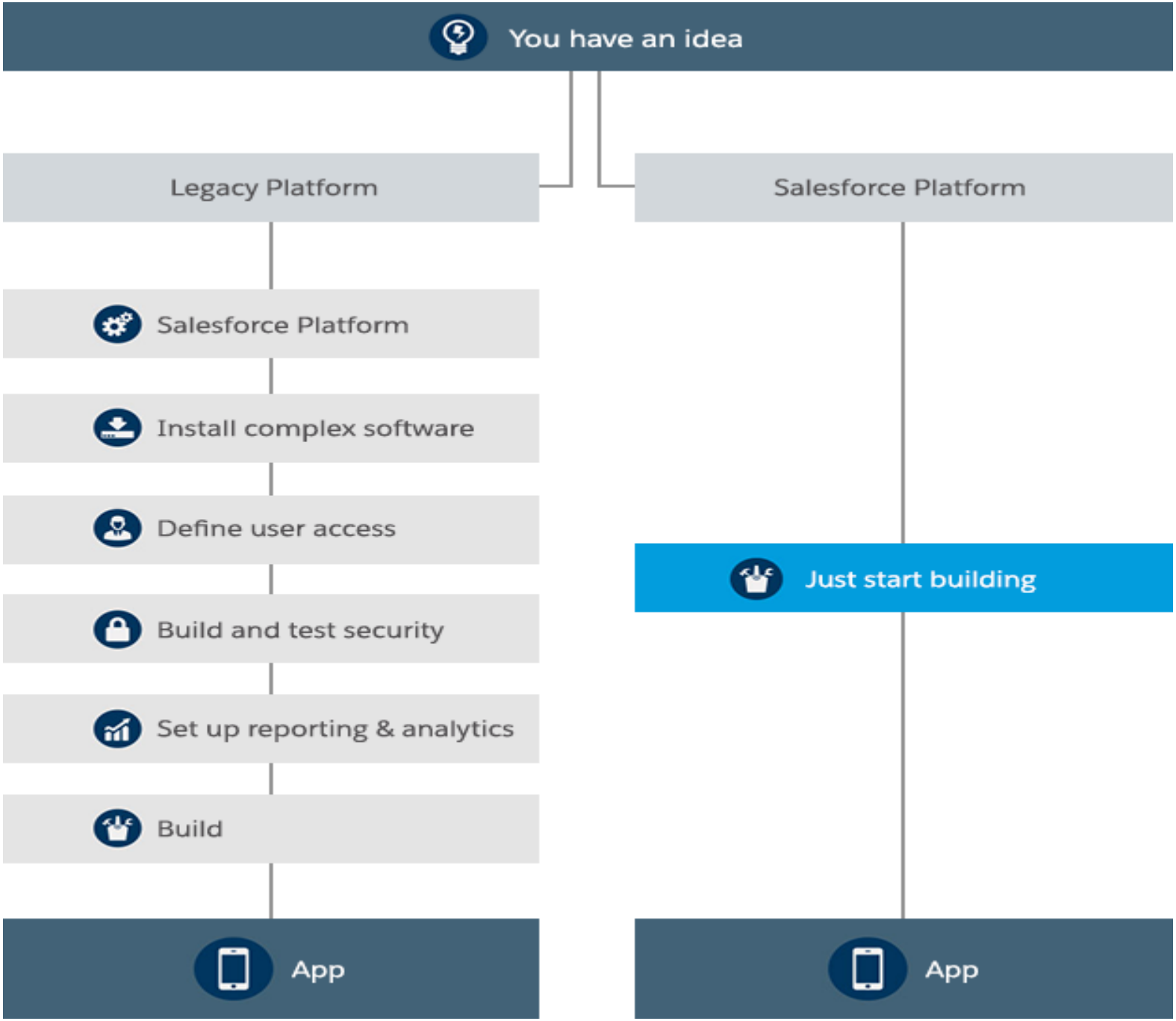
Term Test (TT) (for 15 Marks)

Best of Two (TT-1/TT-2)

End Semester Examination (C):

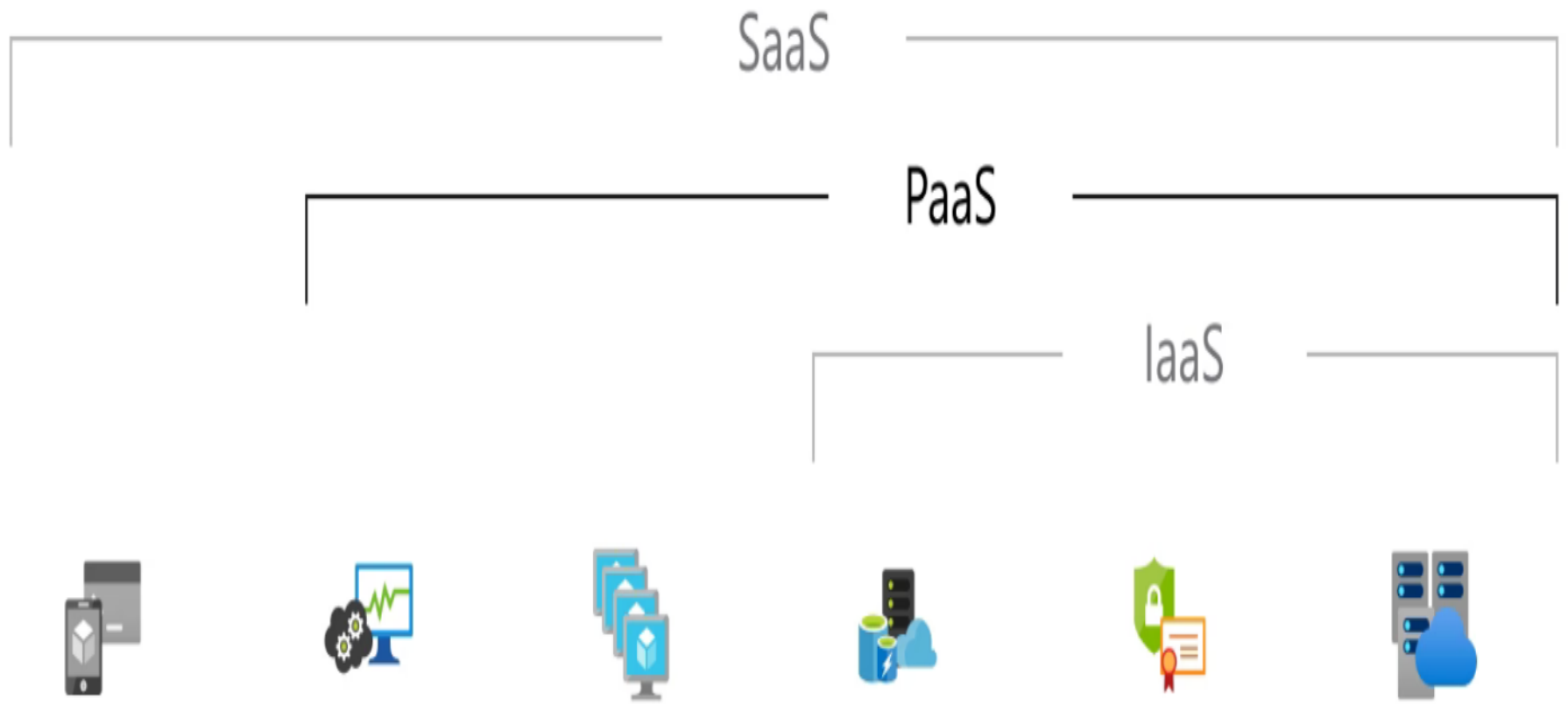
1. Question paper based on the entire syllabus, summing up to 65 marks.

2. Total duration allotted for writing the paper is 3 hrs.


You have an idea
Legacy Platform
Salesforce Platform
Salesforce Platform
Install complex software
Define user access
Build and test security
Set up reporting & analytics
Build
App
Just start building
App

- **Introduction to PaaS**
- Platform as a service (PaaS) is a cloud computing model that provides customers a complete cloud platform—hardware, software and infrastructure—for developing, running and managing applications without the cost, complexity and inflexibility that often comes with building and maintaining that platform on premises.
- Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.
- You purchase the resources you need from a cloud service provider on a pay-as-you-go basis and access them over a secure Internet connection.

- **Introduction to PaaS**
- Like IaaS, PaaS includes infrastructure—servers, storage, and networking—but also middleware, development tools, business intelligence (BI) services, database management systems, and more.
- PaaS is designed to support the complete web application lifecycle: building, testing, deploying, managing, and updating.
- PaaS allows you to avoid the expense and complexity of buying and managing software licenses, the underlying application infrastructure and middleware, container orchestrators such as Kubernetes, or the development tools and other resources.
- You manage the applications and services you develop, and the cloud service provider typically manages everything else.



Hosted applications/ apps

Development tools, database management, business analytics

Operating systems

Servers and storage

Networking firewalls/ security

Data center physical plant/ building

- **Introduction to PaaS**
- The PaaS provider hosts everything—servers, networks, storage, operating system software, databases, development tools—at their data center.
- Typically customers can pay a fixed fee to provide a specified amount of resources for a specified number of users, or they can choose 'pay-as-you-go' pricing to pay only for the resources they use.
- Either option enables PaaS customers to build, test, deploy run, update and scale applications more quickly and inexpensively than if they had had to build out and manage their own on-premises platform.

- **Introduction to PaaS**
- Every leading cloud service provider—including Amazon Web Services (AWS), Google Cloud, IBM Cloud and Microsoft Azure—has its own PaaS offering.
- Popular PaaS solutions are also available as open source projects (for example, Apache Stratos, Cloud Foundry) or from software vendors (for example, Red Hat OpenShift and Salesforce Heroku).

- **Benefits of PaaS**
- The most commonly cited benefits of PaaS, compared to an on-premises platform, include:
  - **Faster time to market.** With PaaS, there's no need to purchase and install the hardware and software you use to build and maintain your application development platform—and no need for development teams to wait while you do it. You simply tap into the cloud service provider's PaaS to begin provisioning resources and developing immediately.

- **Benefits of PaaS**
- The most commonly cited benefits of PaaS, compared to an on-premises platform, include:
  - **Affordable access to a wider variety of resources.** PaaS platforms typically offer access to a wider range of choices up and down the application stack—including operating systems, middleware, databases and development tools—than most organizations can practically or affordably maintain themselves.

- **Benefits of PaaS**
- **More freedom to experiment, with less risk.** PaaS also lets you try or test new operating systems, languages and other tools without having to make substantial investments in them or in the infrastructure required to run them.
- **Easy, cost-effective scalability.** With an on-premises platform, scaling is always expensive, often wasteful and sometimes inadequate:
  - You must purchase more compute, storage and networking capacity in anticipation of traffic spikes.
  - Much of that capacity sits idle during low-traffic periods, and none of it can be increased in time to accommodate unanticipated surges.
  - With PaaS, you can purchase more capacity and use it immediately, whenever you need it.

- **Benefits of PaaS**
- **Greater flexibility for development teams.** PaaS services provide a shared software development environment that allows development and operations teams access to all the tools they need, from any location with an internet connection.
- **Lower costs overall**.
  - Clearly PaaS reduces costs by enabling an organization to avoid capital equipment expense associated with building and scaling an application platform.
  - But PaaS can also reduce or eliminate software licensing costs.
  - And by handling patches, updates and other administrative tasks, PaaS can reduce your overall application management costs.

- **How PaaS works**
- In general, PaaS solutions have three main parts:
  - Cloud infrastructure including virtual machines (VMs), operating system software, storage, networking and firewalls.
  - Software for building, deploying and managing applications.
  - A graphic user interface or GUI where development or DevOps teams can do all their work throughout the entire application lifecycle.

- **How PaaS works**
- Because PaaS delivers all standard development tools through the GUI online interface, developers can log in from anywhere to collaborate on projects, test new applications or roll out completed products.
- Applications are designed and developed right in the PaaS using middleware.
- With streamlined workflows, multiple development and operations teams can work on the same project simultaneously.
- PaaS providers manage the bulk of your cloud computing services, such as servers, runtime and virtualization.
- As a PaaS customer, your company maintains management of applications and data.

- **PaaS, IaaS and SaaS**
- **IaaS:** is internet access to 'raw' IT infrastructure—physical servers, virtual machines, storage, networking and firewalls—hosted by a cloud provider.
- IaaS eliminates cost and the work of owning, managing and maintaining on-premises infrastructure.
- With IaaS, the organization provides its own application platform and applications.
- Any PaaS offering necessarily includes the IaaS resources required to host it, even if those resources aren't discretely broken out or referred to as IaaS.

- **PaaS, IaaS and SaaS**
- **SaaS:** is application software that you use via the cloud, as if it were installed on your computer (sometimes, parts of it are installed on your computer).
- SaaS enables your organization to use an application without the expense of setting up the infrastructure to run it and the effort and personnel to maintain it (apply bug fixes and updates, address outages, and more.)
- **Salesforce and Slack** are examples of popular SaaS offerings.
- Most web applications are considered SaaS.

- **Use cases for PaaS**
- By providing an integrated and ready-to-use platform and by enabling organizations to offload infrastructure management to the cloud provider and focus on building, deploying and managing applications, PaaS can ease or advance several IT initiatives, including:
  - **API development and management:** Because of its built-in frameworks, PaaS makes it much simpler for teams to develop, run, manage and secure APIs (application programming interfaces) for sharing data and functionality between applications.

- **Use cases for PaaS**
- **Internet of Things (IoT):** Out of the box, PaaS can support a range of programming languages (Java, Python, Swift and more), tools and application environments used for IoT application development and real-time processing of data generated by IoT devices.
- **Agile development and DevOps:** PaaS can provide fully-configured environments for automating the software application lifecycle including integration, delivery, security, testing and deployment.

- **Use cases for PaaS**
- **Cloud migration and cloud-native development:**
- With its ready-to-use tools and integration capabilities, PaaS can simplify migration of existing applications to the cloud—particularly via replatforming (moving an application to the cloud with modifications that take better advantage of cloud scalability, load balancing and other capabilities) or refactoring (rearchitecting some or all of an application by using microservices, containers and other cloud-native technologies).

- **Use cases for PaaS**
- **Hybrid cloud strategy:**
  - Hybrid cloud integrates public cloud services, private cloud services and on-premises infrastructure and provides orchestration, management and application portability across all three.
  - The result is a unified and flexible distributed computing environment, where an organization can run and scale its traditional (legacy) or cloud-native workloads on the most appropriate computing model.
  - The right PaaS solution allows developers to build once, then deploy and manage anywhere in a hybrid cloud environment.

- **Common PaaS scenarios**
- Organizations typically use PaaS for these scenarios:
  - **Development framework.**
  - PaaS provides a framework that developers can build upon to develop or customize cloud-based applications.
  - Similar to the way you create an Excel macro, PaaS lets developers create applications using built-in software components.
  - Cloud features such as scalability, high-availability, and multi-tenant capability are included, reducing the amount of coding that developers must do.

- **Common PaaS scenarios**
- Organizations typically use PaaS for these scenarios:
  - **Analytics or business intelligence.**
  - Tools provided as a service with PaaS allow organizations to analyze and mine their data, finding insights and patterns and predicting outcomes to improve forecasting, product design decisions, investment returns, and other business decisions.
  - **Additional services.**
  - PaaS providers may offer other services that enhance applications, such as workflow, directory, security, and scheduling.

- **Advantages of PaaS**
- By delivering infrastructure as a service, PaaS offers the same advantages as IaaS. But its additional features—middleware, development tools, and other business tools—give you more advantages:
  - **Cut coding time.**
  - PaaS development tools can cut the time it takes to code new apps with pre-coded application components built into the platform, such as workflow, directory services, security features, search, and so on.
  - **Add development capabilities without adding staff.**
  - Platform as a Service components can give your development team new capabilities without your needing to add staff having the required skills.

- **Advantages of PaaS**
- **Develop for multiple platforms**—including mobile—more easily. Some service providers give you development options for multiple platforms, such as computers, mobile devices, and browsers making cross-platform apps quicker and easier to develop.
- **Use sophisticated tools affordably**. A pay-as-you-go model makes it possible for individuals or organizations to use sophisticated development software and business intelligence and analytics tools that they could not afford to purchase outright.
- **Support geographically distributed development teams.** Because the development environment is accessed over the Internet, development teams can work together on projects even when team members are in remote locations.
- **Efficiently manage the application lifecycle.** PaaS provides all of the capabilities that you need to support the complete web application lifecycle: building, testing, deploying, managing, and updating within the same integrated environment.

PaaS

- Public solution stacks for web applications
  - OS, web server, language interpreters, provisions for automatic scaling, all shielded from the user
- Each system only has a few supported languages
  - Automatic deployment and scaling not trivial
- Offers development tools
  - Libraries for specific services
  - IDE plugins, deployment tools
- Two types – Instance PaaS, Framework PaaS

SaaS

- A new form of application delivery
- Global availability – web application, user data
- Multitenancy – users use the same installation
- Licensing – ideally pay per use
  - When monthly payment, it should be elastic
- Third layer of cloud doesn't imply use of underlying layers, but elastic infrastructure is advisable
  - <- No control over the number of users

PaaS properties

- Gives the programmer a solution stack
  - Web server, database engine, scripting language
- Simple deployment, no worries about servers, storage, network, scaling, updates, …
- Guarantees multitenancy for better security
- Users isolated by virtualization or OS means
- Accounting and billing of used resources
  - Different at every vendor
- Development tools

Comparison with IaaS

- IaaS better for migrating existing applications
  - More flexible, you install your environment
- PaaS has lower demands on administration
- PaaS will take care of scaling if applications use correct frameworks, also redundancy and CDN
- -> PaaS better for new applications
- BUT has dangers of vendor lock in if platform specific functions are used
  - IaaS instance can be copied to your server.

Comparison with Webhosting

- Webhosting essentially does the same – offers a platform for web sites / applications
- Minus scalability, multitenancy, accounting
- Plus personal contact – negotiation, support
- Different languages, cloud focuses on scalability
  - Hosting: PHP, ASP, some Perl and Python
  - Cloud: Java, Ruby, PHP (due to demand), Node.js
- Added value – e-mail and domain hosting
  - vs. development tools and web services in PaaS

PaaS types

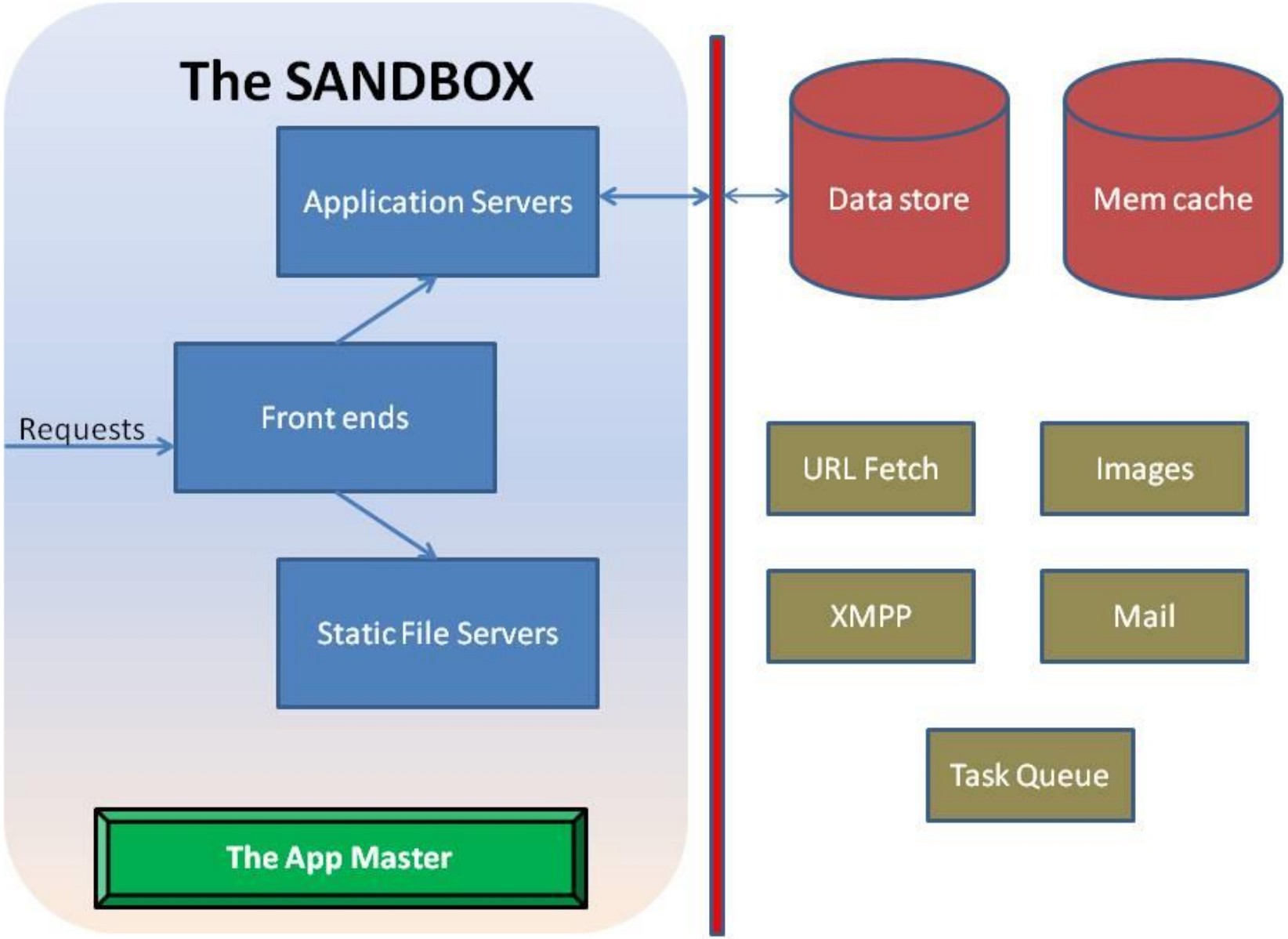
- Instance PaaS
  - Depends on IaaS layer for multitenancy
    - Better security and performance guarantees
  - Deploys applications to IaaS instances
- Framework PaaS
  - Uses OS capabilities for multitenancy
    - Better resource utilization and accounting granularity
  - Requires specific frameworks to be used
  - Can benefit from cloud infrastructure, but is not dependent on it
- Metadata PaaS
  - Client configures his service through metadata

Available systems (selected)

- Instance PaaS
  - Amazon Elastic Beanstalk
  - Microsoft Azure
- Framework PaaS
  - Google App Engine
  - VMware Cloud Foundry

Google App Engine

- Since Apr. 2008, commercial in Sep. 2011
- Languages (in order) – Python, Java, Go
- Typical Framework PaaS
  - Multitenancy by limiting system library functions
    - No filesystem writes and network sockets
    - Must use specific database and network services
      - High vendor lock-in potential (ex. emulation projects)
  - Quotas needed to support massive multitenancy
    - Daily quotas – billing, fair use
    - Per minute quotas – spike prevention



Architecture of Google App Engine

GAE - Quotas

- Basic quota enough for development
  - 28 h CPU time per day (granularity 15 min)
  - 1 GB datastore, 5 GB blobstore, 1 GB network traf.
- If billing enabled, minor service quotas raised
  - $0.08 CPU hour 600 MHz (of what?), larger inst.
  - $0.12 1 GB net traffic
  - $0.24 datastore, $0.13 blobstore, $0.1 100k writes
- If exceeded
  - Main quotas: user gets HTTP 403
  - Service quotas: OverQuotaError exception

GAE – Services

- Data storage based on Google File System
  - Over it BigTable noSQL database engine
    - Providing Datastore and Blobstore APIs (GQL lang.)
      - For Java JDO and JPA compatibility layers
        - Problem – not SQL: no indexes, no joins –> very limited
  - Datastore – Master-Slave or High Replication
- Google Accounts, HTTP fetch, e-mail send, XMPP, Memcache, image manip. and others
- Client libraries for all APIs in Eclipse plugin

GAE – Limitations

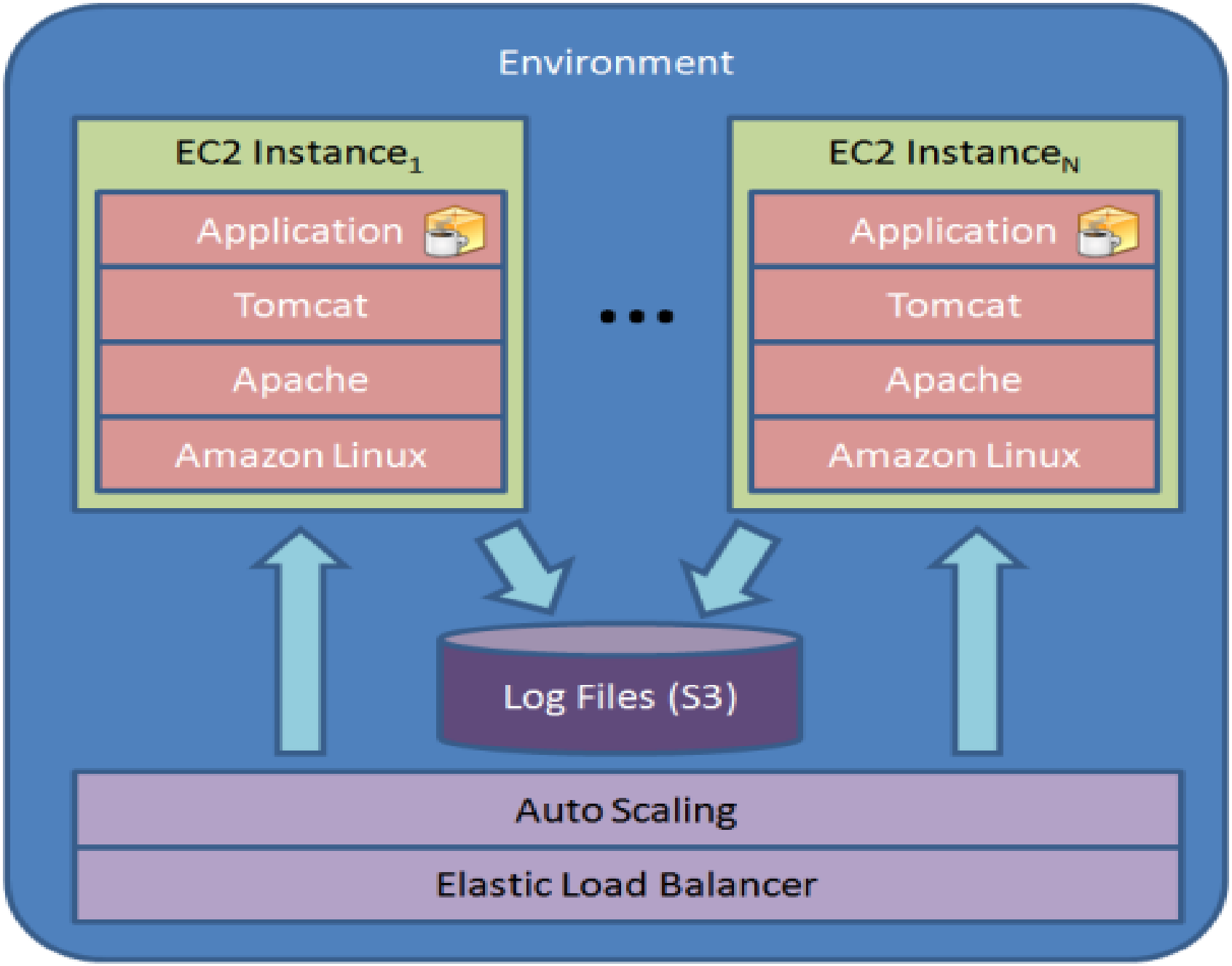
- No state information between HTTP requests
  - All sessions etc. in datastore
  - Good for instance unloading and scalability
- Event style programming, 30 s per request
  - Last year: Single-threaded instances default
    - Can be overriden with <threadsafe> in config
  - Task queues (60s tasks) and backend instances
- Specific programming style
  - Practically impossible to deploy existing apps.
  - Vendor lock-in

Amazon Elastic Beanstalk

- Since beginning of 2011, still Beta
- Initially only Java, extended by
  - PHP and Python, Ruby, .NET
- Amazon – major IaaS provider -> Instance PaaS
- Other PaaSs on Amazon: Heroku, AppFog, ..
- Multitenancy by virtualization
  - Each user gets his own VM instances
  - Some services shared, some need more instances
  - Minimal limitations – can migrate to IaaS

Beanstalk – Pricing

- Free – only a management tool for other services
  - Prepares images with solution stacks, w/upgrades
- Deploys images on EC2 instances:
  - m1.small – $0.085 (1,7 GB RAM, 1 ECU)
  - t1.micro – $0.02 (613 MB, 10% of 2 ECU)
- Uses Elastic Load Balancer:
  - $0.025 / 1 hour, $0.008 / 1 GB
- Applications and logs are on S3:
  - $0.14 / 1 GB and month
- Uses Elastic Autoscaler, which needs CloudWatch
  - $3.5 per instance and month
- Data transfers outbound from cloud
  - $0.12 / 1 GB basic price, degressive for larger volumes


Environment
EC2 Instance1
Application
Tomcat
Apache
Amazon Linux
EC2 InstanceN
Application
Tomcat
Apache
Amazon Linux
Log Files (S3)
Auto Scaling
Elastic Load Balancer

Beanstalk – Services

- The service itself has no database, Amazon has:
- Database VM instances
  - per hour licensing for IBM DB2 and Informix, Oracle, MS SQL, Sybase, Postgres Plus, normal instance price for free DBs
- RDS – automatically managed MySQL and Oracle
  - does updates, backups, scaling, HA between av. zones
  - Paid per hour, $0.11, 2x in hot-standby, $0.1 / 1 GB
- Non-relational databases with variable row format & sharding
  - SimpleDB - $0.25 / 1 GB, $0.14 / machine hour (shared)
    - Max size 10 GB, automatic indexing, website use, 1 GB & 25 h free
  - DynamoDB - $1 / 1 GB, $0.01 for 50 kB/s (granularity 1h)
    - On SSD storage, guaranteed IOPS, for data mining
- Other services: MapReduce, ElastiCache, Route 53, e–mail sending, Simple Queue Service, Simple Notification Service

Beanstalk – Limitations

- Minimal.
  - Applications are deployed to standard application servers (currently Apache Tomcat)
  - Standard relational databases available
  - Can disable the service and continue as IaaS only
- Local storage is ephemeral, design for scalability
- High price ~ $40 for minimal conf. + database
- Eclipse plugin provided, API and CLI tools

Microsoft Azure

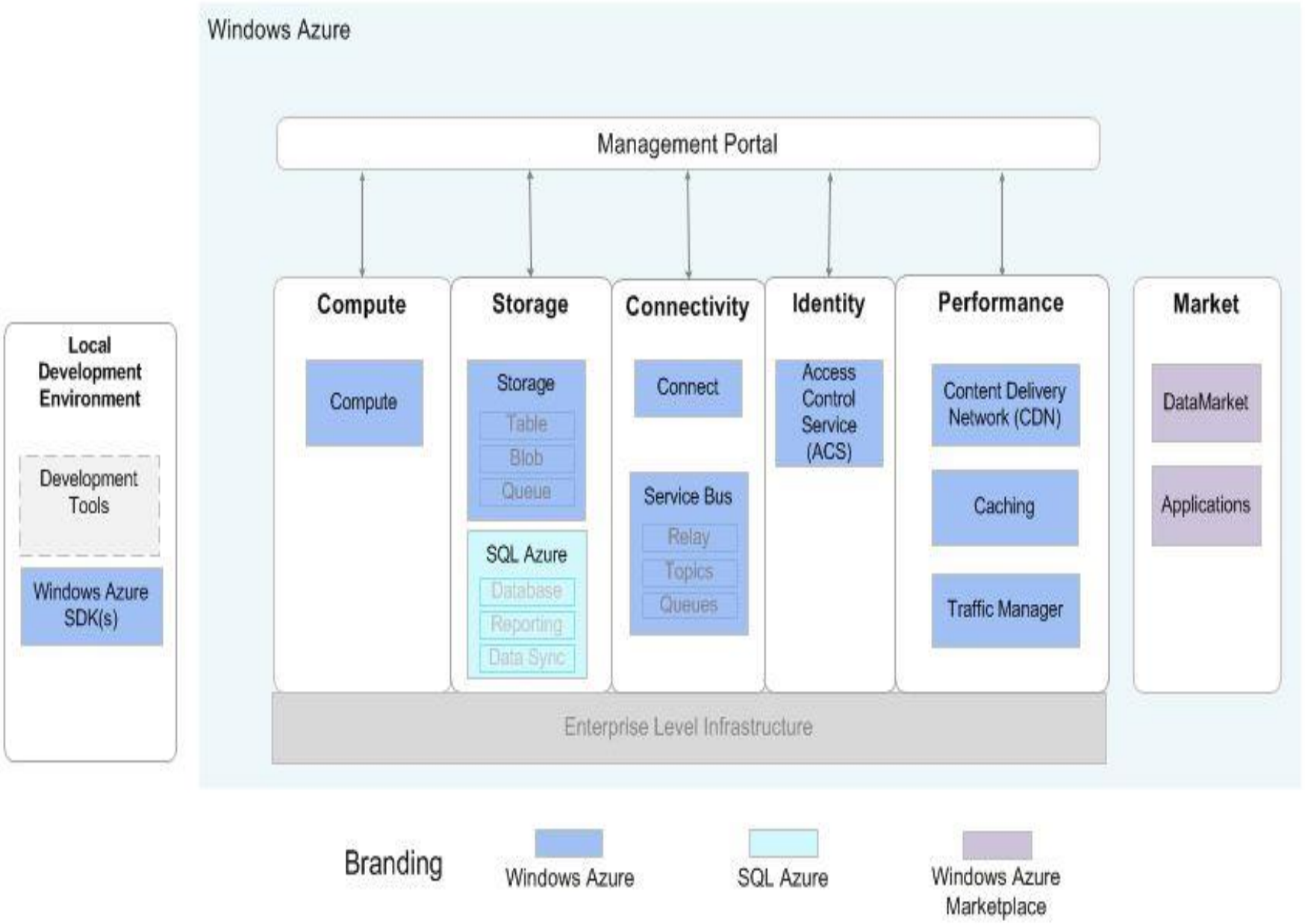
- Since Oct. 2008, commercial in Feb. 2010
- Languages: ASP.NET, PHP, Node.js, Java
- Internally Instance PaaS, VM Role only recently
  - Web Role – runs IIS server on W2k8 and Hyper-V
    - Application is a web server root directory
  - Worker Role – no IIS
    - Application is a package with EXE file
    - For services outside IIS, ie. Java, Node.js
  - VM Role – runs your VHD images of W2k8
    - Essentially IaaS

Azure – Pricing

- Instance PaaS – accounting by instance hours
- 5 instance types, prices exactly as Amazon + Win
  - Small: 1 CPU, 1,75 GB RAM, 230 GB disk, $0.12
  - ExtraSmall, 768GB RAM, 20 GB disk, $0.04
- Possible to buy discount packages
  - Like Amazon's Reserved Instances
- Interesting SLA – 99.95% only if you have 2 inst.
  - Automatic updates need reboot :-P

Azure – Services

- noSQL storage
  - Tables – variable row format, max. 255 attributes, manual sharding - “partition key”
  - Blobs – for storing large objects
    - Block blobs: 4 MB blocks, addressable individually or as a whole, max. size 200 GB, may use CDN
    - Page blobs: 512 B pages, sparse writes, snapshots
      - Azure Drives: page blobs containing VHD images
    - Queues: 8 kB XML messages for IPC
  - Counted together at $0.14 / 1 GB and $0.01 / 10k I/Os
- SQL Azure – modified MS SQL Server with clustering
  - Un-cloudlike flat fees, minimum $5 for max. 100 MB
- AppFabric – Access Control, SOA bus; Connect (VPN), Market (apps and datasets), Caching, MPI


Windows Azure
Management Portal
Compute
Compute
Storage
Storage
Table
Blob
Queue
SQL Azure
Database
Reporting
Data Sync
Connectivity
Connect
Service Bus
Relay
Topics
Queues
Identity
Access Control Service (ACS)
Performance
Content Delivery Network (CDN)
Caching
Traffic Manager
Market
DataMarket
Applications
Enterprise Level Infrastructure
Local Development Environment
Development Tools
Windows Azure SDK(s)
Branding
Windows Azure
SQL Azure
Windows Azure Marketplace

Azure – Limitations

- MS technologies.
  - Windows command line, T-SQL language, Caching is not memcached, MPI is not OpenMPI
  - Some may view it as an advantage
- A cloud classic – local storage is ephemeral
- No e-mail sending service
- Deployment requires writing an XML descriptor
  - Also needed for scaling – no built-in automation
- Prices higher than Amazon – need 2 instances
- Plugins for MS Visual Studio and Eclipse

Cloud Foundry

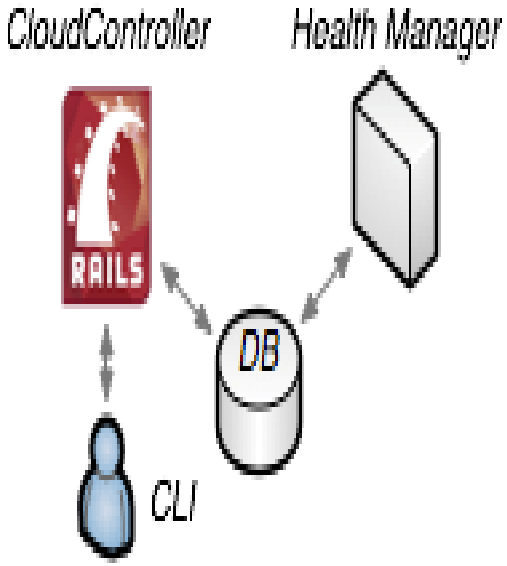
- Since beginning of 2011, still beta
- The only open source PaaS to date
  - Update – RedHat OpenShift now also available
  - There will be multiple providers, private clouds
- Supports Spring for Java, Ruby on Rails (and Sinatra), node.js, Grails and Scala on Lift
  - Typical Framework PaaS
    - Unlike Google, uses standard frameworks
  - More added by companies and community
    - PHP, Django on Python, Erlang
- Three versions: cloudfoundry.com, private, micro (in a VMware Player image)

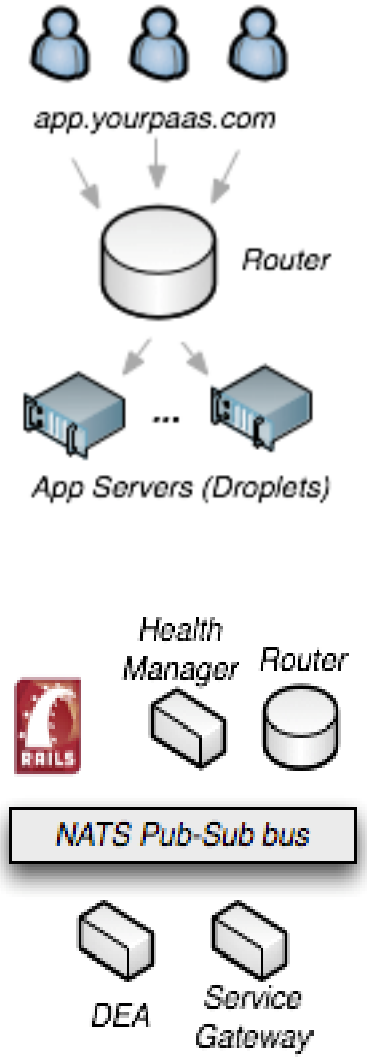
Cloudfoundry.com

- Public version is free in beta
- AppFog has a commercial public cloud already
- Limits:
  - 2 GB total memory per user
    - Apps can have from 64 MB to 2 GB per instance
  - Maximum 20 applications
  - 128 MB data in MySQL
  - 16 MB Redis, 240 MB MongoDB
- Which sums up the services, except RabbitMQ
  - More may be added later

CloudFoundry – Structure

- Droplet Execution Agent
- Router
- Cloud Controller
- Health Manager
- NATS message bus





Cloud Foundry – Limitations

- Not significant if programmer adheres to framework's design patterns
  - For Java – contains Tomcat application server
- Low Security – only standard UNIX user/group
  - A local exploit will compromise the DEA, exposing other users' apps and data
  - Nimbula IaaS – launches a whole CF for each user
  - ActiveState Stackato – a derivative using LXC
    - (Iron Foundry – a derivative running .NET)
- Scaling is manual, local storage ephemeral
- Eclipse plugin available, "vmc" CLI tool

- **What is Cloud Management?**
  - Cloud management is the organized management of cloud computing products and services that operate in the cloud.
  - It refers to the processes, strategies, policies, and technology used to help control and maintain public and private cloud, hybrid cloud, or multicloud environments.
  - Today, you are likely operating partially or entirely in the cloud.
  - As such, your organization needs a way to evaluate, monitor, and manage cloud computing resources, infrastructure, and services in the most efficient way possible.

- **What is Cloud Management?**
  - Keeping your cloud environments running smoothly requires many underlying tasks, from provisioning and orchestration of resources to automating cloud consumption and deployment, to lifecycle management of resources, cost optimization, performance monitoring, and security.

- **Cloud management defined**
  - Cloud management is the process of maintaining control and oversight of cloud computing infrastructure, resources, and services in public, private, or hybrid cloud environments.
  - With cloud management tools and technologies, IT administrators achieve control, visibility, and scalability while also being able to adapt rapidly to change.

- **Challenges of managing a cloud environment**
- As reliance on cloud computing continues to increase, many organizations are finding it difficult to manage increasingly complex architectures and gain the oversight needed to successfully implement their cloud strategies.
- To fully realize the potential of cloud services and solutions, it's also essential to understand the challenges with managing cloud environments you may encounter as you migrate to the cloud.
    - **Interoperability**
    - Today's modern IT environments are increasingly spread out across a mix of on-premises, public and private cloud, hybrid, or even multi cloud environments.
    - Teams must be able to consolidate the management of numerous, heterogeneous systems in a simple way.

- **Challenges of managing a cloud environment**
- **Cloud sprawl**
  - Self-service can boost productivity and development speed, but it can also become unmanageable.
  - You can start to lose track of cloud resources, leaving them to multiply and increase throughout your organization without oversight or control.
  - Cloud sprawl can cause costs to skyrocket, create security risks, and complicate management down the line.

- **Challenges of managing a cloud environment**
- **Cost assessment**
  - While cloud environments enable true cost management, estimating cloud costs to pinpoint waste can still pose many challenges.
  - Calculating the cost of a single service might require gathering details across accounts, regions, and other cloud-based tools and services.
  - This can be complicated further if there are any overlapping resources that must be paid by an individual business unit within your organization.

- **Challenges of managing a cloud environment**
- **Security and privacy**
  - Cloud security remains a top priority for all organizations and cloud service providers, including how to manage access, protect data, and how to secure cloud environments against internal and external threats.
  - Many companies struggle to balance effective use of resources for the highest performance with strong policy and risk mitigation.
  - Ultimately, cloud computing resources and services are not useful without cloud management features to control your infrastructure, applications, and data.

- **Cloud management features**
- Cloud management tools and technologies enable strategies and processes that let you effectively oversee cloud-based assets and applications, facilitate automation workflows, and make informed decisions so your cloud costs and usage are cost-effective.
- Here are some common cloud management features:
  - **Automation**
  - Application management can be repetitive, especially if you have a large environment.
  - Cloud management enables programmatic management of resources, providing automated capacity management, continuous integration, and orchestration to reduce operational burden.

- **Cloud management features**
- **Compliance and governance**
  - Moving to the cloud can mean relinquishing a certain amount of control over your underlying infrastructure and applications, but you'll still need to have oversight over access and be able to maintain compliance.
  - Cloud management enables you to ensure cloud users and configurations align with regulatory requirements and organizational policies by alerting you to issues so you can address them quickly.

- **Cloud management features**
  - **Security**
  - Since cloud security is a shared responsibility with your cloud service provider, cloud management plays a very active role in protecting your data, applications, and services in cloud environments.
  - Leading cloud management tools offer machine-learning capabilities for robust threat intelligence and detection and help streamline security monitoring and processes.

- **Cloud management features**
  - **Monitoring and logging**
  - Cloud management enables cloud observability with monitoring and logging of events that occur to help you inspect and understand what's going on across your environments.
  - Being able to aggregate, analyze, and correlate log files can help you to identify errors, automate incident management, and optimize performance.

- **Cloud management features**
  - **Performance and cost reporting**
  - Cloud computing costs can quickly balloon past your budget if you're not paying close attention from the start.
  - Cloud management provides cost management, reporting, and forecasting to help you gain an accurate picture of what and when resources are being used.
  - This can allow you to allocate resources more effectively, understand if you're using what you pay for, and eliminate unnecessary costs.

- **How does cloud management work?**
  - Cloud management tools provide a way to manage the deployment and operation of your cloud applications and services across public cloud, hybrid cloud, or multicloud environments.
  - A good cloud management platform will also help you track and monitor usage and performance for each part of your cloud infrastructure, enabling you to make informed decisions about the best way to run workloads.
  - Traditionally, cloud management tools and software are deployed as virtual machines with their own database and server.
  - Servers use APIs to communicate with resources interacting within your cloud environments.

- **How does cloud management work?**
  - You can capture activity in cloud-based applications, performance data, and other events, which can be analyzed and consolidated into a unified dashboard that can be accessed through a web interface.
  - Similar to your cloud resources, administrators can access cloud management tools from anywhere they are as long as they have a secure connection.
  - Most cloud service providers offer cloud management as services in their own platforms, with reporting capabilities across cloud applications to collect performance and usage data, generate insights, and send alerts.

- **What is the role of cloud management and why is it important?**
  - Cloud management is essential when it comes to managing hundreds, sometimes thousands, of different applications and servers in the cloud.
  - Doing everything manually is not just inefficient—it's impossible.
  - The move to modern IT environments has enabled organizations to scale their applications and innovate faster, but it has also introduced considerable complexity.
  - With cloud management, you can improve overall IT performance, improve security, and ensure your costs stay under control.
  - **Here are some of the primary cloud management benefits:**

- **What is the role of cloud management and why is it important?**
  - **It's Faster time to value**
  - Cloud management cuts down on the amount of time you spend on routine tasks and maintenance.
  - Self-service access for automated provisioning and deployments helps support agile development across the full application lifecycle to speed up delivery and reduce errors.
  - **Visibility across environments**
  - Stay ahead of cloud sprawl by keeping track of what resources you have, how they're being used, and who has access with cloud management.
  - It can also help you identify the optimal balance for your workloads and make the most of your cloud capacity.

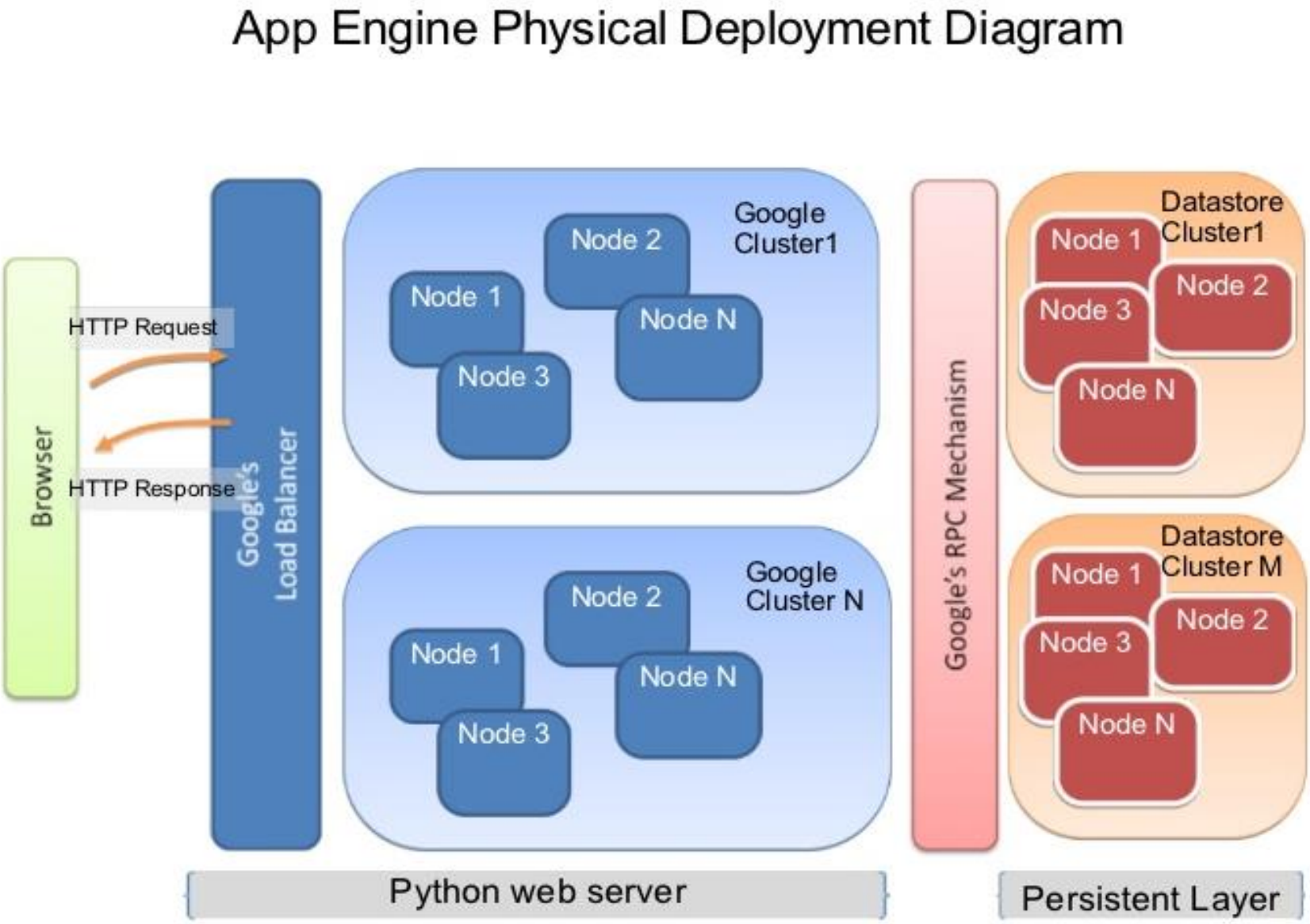
- **What is the role of cloud management and why is it important?**
  - **Centralized management**
  - Cloud management tools allow you to integrate applications, operating systems, storage, cloud security tools, and more.
  - The result is consistent and unified management over resource allocation, compliance, security, and cost management across multiple clouds and environments.
  - **Improved compliance & security**
  - Cloud management can help you adhere to cloud policies, guidelines, and regulatory requirements.
  - Administrators can also streamline security processes, automatically monitoring usage and configurations across environments to identify violations and other emerging incidents quickly.

- **What is the role of cloud management and why is it important?**
  - **Optimized cloud spend**
  - It's not always easy to track or predict cloud spend.
  - Cloud management solutions provide detailed reports with chargeback and showback to help you better understand usage, allocate resources more effectively, and accurately forecast spend.
  - **Environmental sustainability**
  - Cloud management can help you to drive and enforce strategies to help reduce wasted resources and lower the energy consumption of your cloud infrastructure.

- **Cloud management strategies**
- Effectively managing public, private, hybrid, or multicloud environments isn't just about the technologies and tools but also a strong cloud management strategy. Creating a solid strategy will vary between organizations—you'll need to find the best processes, technologies, and skill sets that best match your business needs.
- A strong cloud management strategy should:
  - Enable consistent control across environments, allowing you to deploy and integrate services where they are best suited.
  - Allow for agile change management while allowing you to continuously monitor, and analyze capacity, usage, costs, and security.

- **Cloud management strategies**
  - Create security and compliance guardrails that deliver security control and automated policy enforcement without stifling velocity.
  - Continuously review cloud resources and services, from costs to changes to users to violations, to spot issues before they become future problems and update business rules and processes accordingly.

- **Case Study – Google App Engine**



- **Google App Engine** (often referred to as GAE or simply App Engine, and also used by the acronym GAE/J) is a platform as a service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers.
- Applications are sandboxed and run across multiple servers.
- App Engine offers automatic scaling for web applications—as the number of requests increases for an application, App Engine automatically allocates more resources for the web application to handle the additional demand.
- Google App Engine is free up to a certain level of consumed resources.
- Fees are charged for additional storage, bandwidth, or instance hours required by the application.
- It was first released as a preview version in April 2008, and came out of preview in September 2011.

- **Runtimes and frameworks**
  - Currently, the supported programming languages are Python, Java (and, by extension, other JVM languages such as Groovy, JRuby, Scala, Clojure, Jython and PHP via a special version of Quercus), and Go.
  - Google has said that it plans to support more languages in the future, and that the Google App Engine has been written to be language independent.
- **Reliability and Support**
  - All billed High-Replication Datastore App Engine applications have a 99.95% uptime SLA.

- **Portability Concerns**
  - Developers worry that the applications will not be portable from App Engine and fear being locked into the technology.
  - In response, there are a number of projects to create open-source back-ends for the various proprietary/closed APIs of app engine, especially the datastore.
  - Although these projects are at various levels of maturity, none of them is at the point where installing and running an App Engine app is as simple as it is on Google's service. AppScale and TyphoonAE are two of the open source efforts.
  - AppScale can run Python, Java, and Go GAE applications on EC2 and other cloud vendors.
  - TyphoonAE can run python App Engine applications on any cloud that support linux machines.
  - Web2py web framework offers migration between SQL Databases and Google App Engine, however it doesn't support several App Engine-specific features such as transactions and namespaces.

- **Differences with other application hosting**
  - Compared to other scalable hosting services such as Amazon EC2, App Engine provides more infrastructure to make it easy to write scalable applications, but can only run a limited range of applications designed for that infrastructure.
  - App Engine's infrastructure removes many of the system administration and development challenges of building applications to scale to hundreds of requests per second and beyond.
  - Google handles deploying code to a cluster, monitoring, failover, and launching application instances as necessary.

- **Differences with other application hosting**
  - While other services let users install and configure nearly any *NIX compatible software, App Engine requires developers to use only its supported languages, APIs, and frameworks.
  - Current APIs allow storing and retrieving data from a BigTable non-relational database; making HTTP requests; sending e-mail; manipulating images; and caching.
  - Existing web applications that require a relational database will not run on App Engine without modification.