

Jaipur Engineering College & Research Centre, Jaipur



Notes

Cloud Computing

[6CS4-06]

**Prepared By:
Suniti Chouhan
Abhishek Jain**

VISION AND MISSION OF INSTITUTE

VISION

To become renowned centre of outcome based learning and work towards academic, professional, cultural and social enrichments of the lives of individual and communities”

MISSION

M1. Focus on evaluation of learning outcomes and motivate students to inculcate research aptitude by project based learning.

M2. Identify areas of focus and provide platform to gain knowledge and solutions based on informed perception of Indian, regional and global needs.

M3. Offer opportunities for interaction between academia and industry.

M4. Develop human potential to its fullest extent so that intellectually capable and imaginatively gifted leaders can emerge in a range of professions.

VISION AND MISSION OF DEPARTMENT

VISION

To become renowned Centre of excellence in computer science and engineering and make competent engineers & professionals with high ethical values prepared for lifelong learning.

MISSION

M1: To impart outcome based education for emerging technologies in the field of computer science and engineering.

M2: To provide opportunities for interaction between academia and industry.

M3: To provide platform for lifelong learning by accepting the change in technologies

M4: To develop aptitude of fulfilling social responsibilities.

COURSE OUTCOMES

CO1: Implement the cloud computing architecture i.e, the model, types of clouds, various service models and programming concepts.

CO2: Acquire knowledge about the recent trends in area of cloud computing like Hadoop, programming of Google app engine and virtualization technology and resource management.

CO3: Identify the various threats related to cloud and as well as disaster recovery related to same.

CO4:Analyze the cloud platforms in IT industry and various case studies on the industries providing cloud services.

Program Outcomes (PO)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Educational Objectives (PEO)

1. To provide students with the fundamentals of Engineering Sciences with more emphasis in Computer Science & Engineering by way of analyzing and exploiting Engineering challenge
2. To train students with good scientific and engineering knowledge so as to comprehend, analyze, design, and create novel products and solutions for the real life problems.
3. To inculcate professional and ethical attitude, effective communication skills, teamwork skills, multidisciplinary approach, entrepreneurial thinking and an ability to relate engineering issues with social issues.
4. To provide students with an academic environment aware of excellence, leadership, written ethical codes and guidelines, and the self-motivated life-long learning needed for a successful professional career.
5. To prepare students to excel in Industry and Higher education by Educating Students along with High moral values and Knowledge.

MAPPING CO-PO

Cos/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	3	2	2	2	1	2	2	3
CO2	3	3	3	3	3	2	2	2	1	2	2	3
CO3	3	3	3	3	2	2	2	2	2	2	2	3
CO4	3	3	3	3	2	2	2	2	2	2	2	3

PSO

PSO1: Ability to interpret and analyze network specific and cyber security issues, automation in real word environment.

PSO2: Ability to Design and Develop Mobile and Web-based applications under realistic constraints.

SYLLABUS

UNIT 1: Introduction: Objective, scope and outcome of the course.

UNIT 2: Introduction: Objective, scope and outcome of the course. Introduction Cloud Computing: Nutshell of cloud computing, Enabling Technology, Historical development, Vision, feature Characteristics and components of Cloud Computing. Challenges, Risks and Approaches of Migration into Cloud. Ethical Issue in Cloud Computing, Evaluating the Cloud's Business Impact and economics, Future of the cloud. Networking Support for Cloud Computing. Ubiquitous Cloud and the Internet of Things.

UNIT 3: Cloud Computing Architecture: Cloud Reference Model, Layer and Types of Clouds, Services models, Data centre Design and interconnection Network, Architectural design of Compute and Storage Clouds. Cloud Programming and Software: Fractures of cloud programming, Parallel and distributed programming paradigms-Map Reduce, Hadoop, High level Language for Cloud. Programming of Google App engine.

UNIT 4: Virtualization Technology: Definition, Understanding and Benefits of Virtualization. Implementation Level of Virtualization, Virtualization Structure/Tools and Mechanisms, Hypervisor VMware, KVM, Xen. Virtualization: of CPU, Memory, I/O Devices, Virtual Cluster and Resources Management, Virtualization of Server, Desktop, Network, and Virtualization of data-centre.

UNIT 5: Securing the Cloud: Cloud Information security fundamentals, Cloud security services, Design principles, Policy Implementation, Cloud Computing Security Challenges, Cloud Computing Security Architecture . Legal issues in cloud Computing. Data Security in Cloud: Business Continuity and Disaster Recovery , Risk Mitigation , Understanding and Identification of Threats in Cloud, SLA-Service Level Agreements, Trust Management

UNIT 6: Cloud Platforms in Industry: Amazon web services , Google AppEngine, Microsoft Azure Design, Aneka: Cloud Application Platform -Integration of Private and Public Clouds Cloud applications: Protein structure prediction, Data Analysis, Satellite Image Processing, CRM

Unit:-5 Cloud Platforms in Industry

Cloud computing has become one of the most discussed IT paradigms of recent years. It builds on many of the advances in the IT industry over the past decade and presents significant opportunities for organizations to shorten time to market and reduce costs. With cloud computing, organizations can consume shared computing and storage resources rather than building, operating, and improving infrastructure on their own. The speed of change in markets creates significant pressure on the enterprise IT infrastructure to adapt and deliver. Cloud computing provides fresh solutions to address these changes. As defined by Gartner¹, “Cloud computing is a style of computing where scalable and elastic IT-enabled capabilities are delivered as a service to external customers using Internet technologies.”

Cloud computing enables organizations to obtain a flexible, secure, and cost-effective IT infrastructure, in much the same way that national electric grids enable homes and organizations to plug into a centrally managed, efficient, and cost-effective energy source.

Amazon

- Amazon has a long history of using a decentralized IT infrastructure. This arrangement enabled our development teams to access compute and storage resources on demand, and it has increased overall productivity and agility.
- By 2005, Amazon had spent over a decade and millions of dollars building and managing the large-scale, reliable, and efficient IT infrastructure that powered one of the world’s largest online retail platforms.
- Amazon launched Amazon Web Services (AWS) so that other organizations could benefit from Amazon’s experience and investment in running a large-scale distributed, transactional IT infrastructure.
- AWS has been operating since 2006, and today serves hundreds of thousands of customers worldwide. Today Amazon.com runs a global web platform serving millions of customers and managing billions of dollars’ worth of commerce every year.
- Using AWS, you can requisition compute power, storage, and other services in minutes and have the flexibility to choose the development platform or programming model that

- makes the most sense for the problems they're trying to solve. You pay only for what you use, with no up-front expenses or long-term commitments, making AWS a cost-effective way to deliver applications.

Here are some of examples of how organizations, from research firms to large enterprises, use AWS today:

- A large enterprise quickly and economically deploys new internal applications, such as HR solutions, payroll applications, inventory management solutions, and online training to its distributed workforce.
- An e-commerce website accommodates sudden demand for a “hot” product caused by viral buzz from Facebook and Twitter without having to upgrade its infrastructure.
- A pharmaceutical research firm executes large-scale simulations using computing power provided by AWS.
- Media companies serve unlimited video, music, and other media to their worldwide customer base.

Characteristics of AWS

Flexible. AWS enables organizations to use the programming models, operating systems, databases, and architectures with which they are already familiar. In addition, this flexibility helps organizations mix and match architectures in order to serve their diverse business needs.

Building applications on AWS is very much like building applications using existing hardware resources. Since AWS provides a flexible, virtual IT infrastructure, you can use the services together as a platform or separately for specific needs. AWS run almost anything—from full web applications to batch processing to offsite data back-ups.

Cost-effective. With AWS, organizations pay only for what they use, without up-front or long-term commitments. the cloud provides an on-demand IT infrastructure that lets you consume only the amount of resources that you actually need. You are not limited to a set amount of storage, bandwidth, or computing resources.

You can access new resources almost instantly. The ability to respond quickly to changes, no matter how large or small, means that you can take on new opportunities and meet business challenges that could drive revenue and reduce costs. If you want to consult with AWS for deeper technical discussions, sales and solutions architecture teams are available.

Scalable and elastic. Organizations can quickly add and subtract AWS resources to their applications in order to meet customer demand and manage costs. The AWS cloud is also a useful resource for implementing short-term jobs, mission-critical jobs, and jobs repeated at regular intervals.

For example, when a pharmaceutical company needs to run drug simulations (a short-term job), it can use AWS to spin up resources in the cloud, and then shut them down when it no longer needs additional resources. When an enterprise has to quickly deal with the effects of natural disaster on its data center (a mission-critical job), it can use AWS to tap into new storage and computing resources to accommodate demand. Furthermore, AWS can preserve computing resources and reduce costs for regularly repeated tasks, such as month-end payroll or invoice processing.

Secure. In order to provide end-to-end security and end-to-end privacy, AWS builds services in accordance with security best practices, provides the appropriate security features in those services, and documents how to use those features.

Ensuring the confidentiality, integrity, and availability of your data is of the utmost importance to AWS, as is maintaining your trust and confidence. AWS takes the following approaches to secure the cloud infrastructure:

- **Certifications and accreditations.**

AWS will continue to obtain the appropriate security certifications and accreditations to demonstrate the security of our infrastructure and services.

- **Physical security.** Amazon has many years of experience designing, constructing, and operating large-scale data centers. The AWS infrastructure is located in Amazon-controlled data centers throughout the world. Knowledge of the location of the data centers is limited to those within Amazon who have a legitimate business reasons for this

- information. The data centers are physically secured in a variety of ways to prevent unauthorized access.

- **Secure services.** Each service in the AWS cloud is architected to be secure. The services contain a number of capabilities that restrict unauthorized access or usage without sacrificing the flexibility that customers demand.

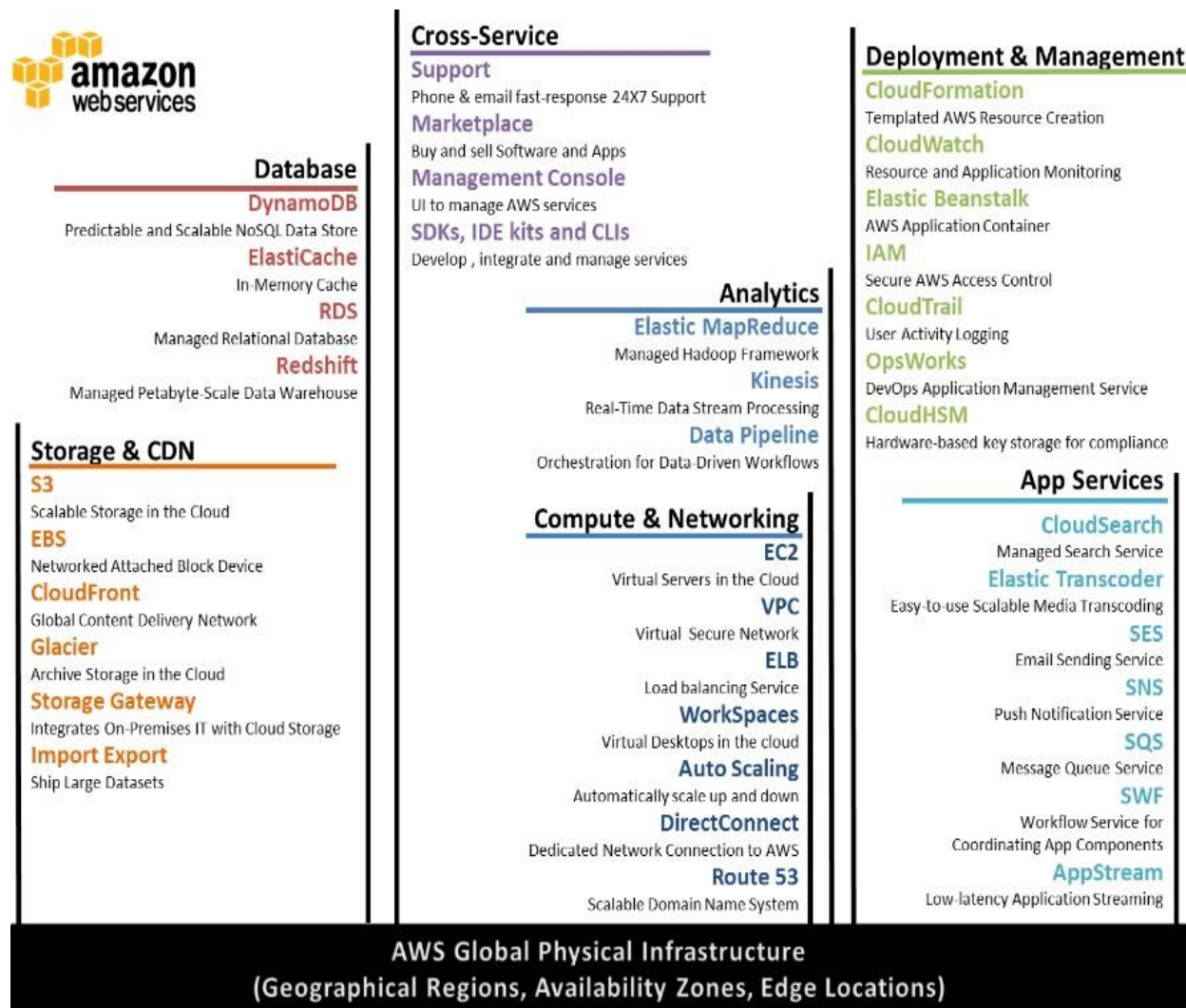
- **Data privacy.** You can encrypt personal and business data in the AWS cloud, and publish backup and redundancy procedures for services so that your customers can protect their data and keep their applications running.

Experienced. When using AWS, organizations can leverage Amazon's more than fifteen years of

experience delivering large-scale, global infrastructure in a reliable, secure fashion. You should hold your cloud-computing partner to the same high standards that you would expect of any hardware or software vendor. The trust that you place in your cloud-computing vendor will be critical as your organization grows and your customers continue to expect the best experience.

Amazon Web Services Cloud Platform

AWS is a comprehensive cloud services platform that offers compute power, storage, content delivery, and other functionality that organizations can use to deploy applications and services cost-effectively with flexibility, scalability, and reliability. AWS self-service means that you can proactively address your internal plans and react to external demands when you choose.



Compute & Networking

Amazon Elastic Compute Cloud (Amazon EC2)

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers and system administrators. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction.
- Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers and system administrators the tools to build failure resilient applications and isolate themselves from common failure scenarios.

Auto Scaling

- Auto Scaling allows you to scale your Amazon EC2 capacity up or down automatically according to conditions.
- Auto Scaling is particularly well suited for applications that experience hourly, daily, or weekly variability in usage.
- Auto Scaling is enabled by Amazon CloudWatch and available at no additional charge beyond Amazon CloudWatch fees.

Elastic Load Balancing

- Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic.
- Customers can enable Elastic Load Balancing within a single Availability Zone or across multiple zones for even more consistent application performance.

Amazon WorkSpaces

- Amazon WorkSpaces is a fully managed desktop computing service in the cloud. Amazon WorkSpaces allows customers to easily provision cloud-based desktops that allow end-users to access the documents, applications and resources they need with the device of their choice, including laptops, iPad, Kindle Fire, or Android tablets.
- With a few clicks in the AWS Management Console, customers can provision a high- quality desktop experience for any number of users at a cost that is highly competitive with

traditional desktops and half the cost of most virtual desktop infrastructure (VDI) solutions.

Amazon Virtual Private Cloud (Amazon VPC)

- Amazon Virtual Private Cloud lets you provision a logically isolated section of the Amazon Web Services (AWS) Cloud where you can launch AWS resources in a virtual network that you define.
- You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
- You can easily customize the network configuration for your Amazon VPC. For example, you can create a public-facing subnet for your web servers that has access to the Internet, and place your backend systems such as databases or application servers in a private-facing subnet with no Internet access.
- You can leverage multiple layers of security (including security groups and network access control lists) to help control access to Amazon EC2 instances in each subnet.
- Additionally, you can create a hardware virtual private network (VPN) connection between your corporate data center and your VPC and leverage the AWS cloud as an extension of your corporate data center.

Amazon Route 53

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to Internet applications by translating human readable names, such as www.example.com, into the numeric IP addresses, such as 192.0.2.1, that computers use to connect to each other.

- Route 53 effectively connects user requests to infrastructure running in AWS, such as an EC2 instance, an elastic load balancer, or an Amazon S3 bucket. Route 53 can also be used to route users to infrastructure outside of AWS.
- Amazon Route 53 is designed to be fast, easy to use, and cost effective. It answers DNS queries with low latency by using a global network of DNS servers. Queries for your domain are automatically routed to the nearest DNS server, and thus are answered with the best possible performance.

- With Route 53, you can create and manage your public DNS records with the AWS Management Console or with an easy-to-use API. It's also integrated with other Amazon Web Services. For instance, by using the AWS Identity and Access Management (IAM) service with Route 53, you can control who in your organization can make changes to your DNS records.
- Like other Amazon Web Services, there are no long-term contracts or minimum usage requirements for using Route 53—you pay only for managing domains through the service and the number of queries that the service answers.

AWS Direct Connect

- AWS Direct Connect makes it easy to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or co-location environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.
- AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations.
- Using industry standard 802.1Q virtual LANS (VLANs), this dedicated connection can be partitioned into multiple logical connections. This allows you to use the same connection to access public resources such as objects stored in Amazon S3 using public IP address space, and private resources such as Amazon EC2 instances running within an Amazon VPC using private IP space, while maintaining network separation between the public and private environments.
- Logical connections can be reconfigured at any time to meet your changing needs.

Storage & Content Delivery Network Amazon Simple Storage Service (Amazon S3)

- Amazon S3 is storage for the Internet. It is designed to make web-scale computing easier for developers.
- Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. The container for objects stored in Amazon S3 is called an Amazon S3 bucket.

- Amazon S3 gives any developer access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that Amazon uses to run its own global network of websites. The service aims to maximize benefits of scale and to pass those benefits on to developers.

Amazon Glacier

- Amazon Glacier is an extremely low-cost storage service that provides secure and durable storage for data archiving and backup.
- In order to keep costs low, Amazon Glacier is optimized for data that is infrequently accessed and for which retrieval times of several hours are suitable.
- With Amazon Glacier, customers can reliably store large or small amounts of data for as little as \$0.01 per gigabyte per month, a significant savings compared to on-premises solutions.
- Companies typically over-pay for data archiving.
- First, they're forced to make an expensive upfront payment for their archiving solution (which does not include the ongoing cost for operational expenses such as power, facilities, staffing, and maintenance).
- Second, since companies have to guess what their capacity requirements will be, they understandably over-provision to make sure they have enough capacity for data redundancy and unexpected growth. This set of circumstances results in under-utilized capacity and wasted money.
- With Amazon Glacier, you pay only for what you use. Amazon Glacier changes the game for data archiving and backup because you pay nothing up front, pay a very low price for storage, and can scale your usage up or down as needed, while AWS handles all of the operational heavy lifting required to do data retention well.
- It only takes a few clicks in the AWS Management Console to set up Amazon Glacier, and then you can upload any amount of data you choose.

Amazon Elastic Block Storage (EBS)

- Amazon Elastic Block Store (EBS) provides block level storage volumes for use with Amazon EC2 instances. Amazon EBS volumes are network-attached, and persist independently from the life of an instance.
- Amazon EBS provides highly available, highly reliable, predictable storage volumes that can be attached to a running Amazon EC2 instance and exposed as a device within the instance.

- Amazon EBS is particularly suited for applications that require a database, file system, or access to raw block level storage.

AWS Storage Gateway

- AWS Storage Gateway is a service connecting an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premises IT environment and AWS's storage infrastructure.
- The service enables you to securely upload data to the AWS cloud for cost-effective backup and rapid disaster recovery.
- AWS Storage Gateway supports industry-standard storage protocols that work with your existing applications.
- It provides low-latency performance by maintaining data on your on-premises storage hardware while asynchronously uploading this data to AWS, where it is encrypted and securely stored in Amazon Simple Storage Service (Amazon S3) or Amazon Glacier.
- Using AWS Storage Gateway, you can back up point-in-time snapshots of your on-premises application data to Amazon S3 for future recovery.
- In the event you need replacement capacity for disaster recovery purposes, or if you want to leverage Amazon EC2's on-demand compute capacity for additional capacity during peak periods, for new projects, or as a more cost-effective way to run your normal workloads, you can use AWS Storage Gateway to mirror your on-premises data to Amazon EC2 instances.

AWS Import/Export

- AWS Import/Export accelerates moving large amounts of data into and out of AWS using portable storage devices for transport.
- AWS transfers your data directly onto and off of storage devices using Amazon's high-speed internal network and bypassing the Internet.
- For significant data sets, AWS Import/Export is often faster than Internet transfer and more cost effective than upgrading your connectivity.
- AWS Import/Export supports importing and exporting data into and out of Amazon S3 buckets in the US East (N. Virginia), US West (Oregon), US West (Northern California), EU (Ireland), and Asia Pacific (Singapore) Regions.

- The service also supports importing data into Amazon EBS snapshots in the US East (N. Virginia), US West (Oregon), US West (Northern California), EU (Ireland), and Asia Pacific (Singapore) Regions.

Amazon Cloud Front

- Amazon Cloud Front is a content delivery web service. It integrates with other Amazon Web Services to give developers and businesses an easy way to distribute content to end users with low latency, high data transfer speeds, and no commitments.
- Amazon CloudFront can be used to deliver your entire website, including dynamic, static and streaming content using a global network of edge locations. Requests for objects are automatically routed to the nearest edge location, so content is delivered with the best possible performance.
- Amazon CloudFront is optimized to work with other Amazon Web Services, like Amazon S3 and Amazon EC2.
- Amazon CloudFront also works seamlessly with any origin server, which stores the original, definitive versions of your files. Like other Amazon Web Services, there are no contracts or monthly commitments for using Amazon CloudFront—you pay only for as much or as little content as you actually deliver through the service.

Database

Amazon Relational Database Service (Amazon RDS)

- Amazon Relational Database Service (Amazon RDS) is a web service that makes it easy to set up, operate, and scale a relational database in the cloud.
- It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business.
- Amazon RDS gives you access to the capabilities of a familiar MySQL, Oracle, SQL Server or PostgreSQL database. This means that the code, applications, and tools you already use today with your existing databases can be used with Amazon RDS.
- Amazon RDS automatically patches the database software and backs up your database, storing the backups for a retention period that you define and enabling point-in-time recovery. You benefit from the flexibility of being able to scale the compute resources or storage

capacity associated with your relational database instance by using a single API call.

- In addition, Amazon RDS makes it easy to use replication to enhance availability and reliability for production databases and to scale out beyond the capacity of a single database deployment for read-heavy database workloads.

Amazon DynamoDB

- Amazon DynamoDB is a fast, fully managed NoSQL database service that makes it simple and cost-effective to store and retrieve any amount of data, and serve any level of request traffic.
- All data items are stored on Solid State Drives (SSDs), and are replicated across 3 Availability Zones for high availability and durability. With DynamoDB, you can offload the administrative burden of operating and scaling a highly available distributed database cluster, while paying a low price for only what you use
- Amazon DynamoDB is designed to address the core problems of database management, performance, scalability, and reliability. Developers can create a database table that can store and retrieve any amount of data, and serve any level of request traffic.
- DynamoDB automatically spreads the data and traffic for the table over a sufficient number of servers to handle the request capacity specified by the customer and the amount of data stored, while maintaining consistent, fast performance.
- All data items are stored on solid state drives (SSDs) and are automatically replicated across multiple Availability Zones in a Region to provide built-in high availability and data durability.
- Amazon DynamoDB enables customers to offload the administrative burden of operating and scaling a highly available, distributed database cluster while only paying a low variable price for the resources they consume.

Amazon ElastiCache

Amazon ElastiCache is a web service that makes it easy to deploy, operate, and scale an in- memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from a fast, managed, in-memory caching system, instead of relying entirely on slower disk-based databases.

ElastiCache supports two open-source caching engines.

- Memcached - a widely adopted memory object caching system. ElastiCache is protocol compliant with Memcached, so popular tools that you use today with existing Memcached environments will work seamlessly with the service.
- Redis – a popular open-source in-memory key-value store that supports data structures such as sorted sets and lists. ElastiCache supports Redis master / slave replication which can be used to achieve cross AZ redundancy.

Amazon ElastiCache automatically detects and replaces failed nodes, reducing the overhead associated with self-managed infrastructures and provides a resilient system that mitigates the risk of overloaded databases, which slow website and application load times. Through integration with Amazon CloudWatch, Amazon ElastiCache provides enhanced visibility into key performance metrics associated with your Memcached or Redis nodes.

Amazon Redshift

- Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyze all your data using your existing business intelligence tools. It is optimized for datasets ranging from a few hundred gigabytes to a petabyte or more and costs less than \$1,000 per terabyte per year, a tenth the cost of most traditional data warehousing solutions.
- Amazon Redshift delivers fast query and I/O performance for virtually any size dataset by using columnar storage technology and parallelizing and distributing queries across multiple nodes.
- We've made Amazon Redshift easy to use by automating most of the common administrative tasks associated with provisioning, configuring, monitoring, backing up, and securing a data warehouse.
- Powerful security functionality is built-in. Amazon Redshift supports Amazon VPC out of the box and you can encrypt all your data and backups with just a few clicks. Once you've provisioned your cluster, you can connect to it and start loading data and running queries using the same SQL-based tools you use today.

Analytics

Amazon Elastic MapReduce (Amazon EMR)

- Amazon Elastic MapReduce (Amazon EMR) is a web service that makes it easy to quickly and

cost-effectively process vast amounts of data.

- Amazon EMR uses Hadoop, an open source framework, to distribute your data and processing across a resizable cluster of Amazon EC2 instances.
- Amazon EMR is used in a variety of applications, including log analysis, web indexing, data warehousing, machine learning, financial analysis, scientific simulation, and bioinformatics. Customers launch millions of Amazon EMR clusters every year.

Amazon Kinesis

- Amazon Kinesis is a fully managed service for real-time processing of streaming data at massive scale.
- Amazon Kinesis can collect and process hundreds of terabytes of data per hour from hundreds of thousands of sources, allowing you to easily write applications that process information in real-time, from sources such as web site click-streams, marketing and financial information, manufacturing instrumentation and social media, and operational logs and metering data.
- With Amazon Kinesis applications, you can build real-time dashboards, capture exceptions and generate alerts, drive recommendations, and make other real-time business or operational decisions.
- You can also easily send data to a variety of other services such as Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, or Amazon Redshift. In a few clicks and a couple of lines of code, you can start building applications which respond to changes in your data stream in seconds, at any scale, while only paying for the resources you use.

AWS Data Pipeline

- AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services as well as on-premise data sources at specified intervals.
- With AWS Data Pipeline, you can regularly access your data where it's stored, transform and process it at scale, and efficiently transfer the results to AWS services such as Amazon S3, Amazon RDS, Amazon DynamoDB, and Amazon Elastic MapReduce (EMR).
- AWS Data Pipeline helps you easily create complex data processing workloads that are fault tolerant, repeatable, and highly available. You don't have to worry about ensuring resource

availability, managing inter-task dependencies, retrying transient failures or timeouts in individual tasks, or creating a failure notification system.

- AWS Data Pipeline also allows you to move and process data that was previously locked up in on-premise data silos.

Amazon AppStream

- Amazon AppStream is a flexible, low-latency service that lets you stream resource intensive applications and games from the cloud. It deploys and renders your application on AWS infrastructure and streams the output to mass-market devices, such as personal computers, tablets, and mobile phones.
- Because your application is running in the cloud, it can scale to handle vast computational and storage needs, regardless of the devices your customers are using. You can choose to stream either all or parts of your application from the cloud.
- Amazon AppStream enables use cases for games and applications that wouldn't be possible running natively on mass-market devices.
- Using Amazon AppStream, your games and applications are no longer constrained by the hardware in your customer's hands.
- Amazon AppStream includes a SDK that currently supports streaming applications from Microsoft Windows Server 2008 R2 to devices running FireOS, Android, iOS, and Microsoft Windows. A Mac OS X SDK is planned for 2014.

Amazon Simple Queue Service (Amazon SQS)

- Amazon Simple Queue Service (Amazon SQS) is a fast, reliable, scalable, fully managed message queuing service.
- SQS makes it simple and cost-effective to decouple the components of a cloud application. You can use SQS to transmit any volume of data, at any level of throughput, without losing messages or requiring other services to be always available.
- With SQS, you can offload the administrative burden of operating and scaling a highly available messaging cluster, while paying a low price for only what you use.

Amazon Simple Notification Service (Amazon SNS)

- Amazon Simple Notification Service (Amazon SNS) Amazon Simple Notification Service

(SNS) is a fast, flexible, fully managed push messaging service. SNS makes it simple and cost-effective to push to mobile devices such as iPhone, iPad, Android, Kindle Fire, and internet connected smart devices, as well as pushing to other distributed services.

- Besides pushing cloud notifications directly to mobile devices, SNS can also deliver notifications by SMS text message or email, to Simple Queue Service (SQS) queues, or to any HTTP endpoint.
- To prevent messages from being lost, all messages published to Amazon SNS are stored redundantly across multiple availability zones.

Amazon Simple Workflow Service (Amazon SWF)

- Amazon Simple Workflow Service (Amazon SWF) is a task coordination and state management service for cloud applications. With Amazon SWF, you can stop writing complex glue-code and state machinery and invest more in the business logic that makes your applications unique.
- Our APIs, ease-of-use libraries, and control engine give developers the tools to coordinate, audit, and scale applications across multiple machines – in the AWS Cloud and other data centers. Whether automating business processes for finance applications, building big-data systems, or managing cloud infrastructure services,
- Amazon SWF helps you develop applications with processing steps that are resilient to failure – steps that can be scaled independent of each other and be audited even when they touch many different systems.
- Using Amazon SWF, you structure the various processing steps in an application that runs across one or more machines as a set of “tasks.”
- Amazon SWF manages dependencies between the tasks, schedules the tasks for execution, and runs any logic that needs to be executed in parallel. The service also stores the tasks, reliably dispatches them to application components, tracks their progress, and keeps their latest state.
- As your business requirements change, Amazon SWF makes it easy to change application logic without having to worry about the underlying state machinery, task dispatch, and flow control, and like other AWS Services, you only pay for what you use.

Amazon Simple Email Service (Amazon SES)

- Amazon Simple Email Service (Amazon SES) is a highly scalable and cost-effective bulk and transactional email sending service for organizations and developers.
- Amazon SES eliminates the complexity and expense of building an in-house email solution or licensing, installing, and operating a third-party email service.
- The service integrates with other AWS services, making it easy to send emails from applications that are hosted on services such as Amazon EC2.
- With Amazon SES there is no long-term commitment, minimum spend, or negotiation required. Organizations can utilize a free usage tier and after that enjoy low fees for the number of emails sent plus data transfer fees.
- Using SMTP or a simple API call, an organization can now access a high-quality, scalable email infrastructure to efficiently and inexpensively communicate to their customers.
- For high email deliverability, Amazon SES uses content filtering technologies to scan an organization's outgoing email messages to help ensure that the content meets ISP standards.
- The email message is then either queued for sending or routed back to the sender for corrective action.
- To help organizations further improve the quality of email communications with their customers, Amazon SES provides a built-in feedback loop, which includes notifications of bounce backs, failed and successful delivery attempts, and spam complaints.

Amazon CloudSearch

- Amazon CloudSearch is a fully-managed service in the AWS Cloud that makes it easy to set up, manage, and scale a search solution for your website or application.
- Amazon CloudSearch enables you to search large collections of data such as web pages, document files, forum posts, or product information.
- With Amazon CloudSearch, you can quickly add search capabilities to your website without having to become a search expert or worry about hardware provisioning, setup, and maintenance. With a few clicks in the AWS Management Console, you can create a search domain, upload the data you want to make searchable to Amazon CloudSearch, and the search service automatically provisions the required technology resources and deploys a highly tuned

search index.

- As your volume of data and traffic fluctuates, Amazon CloudSearch seamlessly scales to meet your needs. You can easily change your search parameters, fine tune search relevance, and apply new settings at any time without having to re-upload your data.
- Amazon CloudSearch enables customers to offload the administrative burden and expense of operating and scaling a search service. With Amazon CloudSearch, there's no need to worry about hardware provisioning, data partitioning, or software patches.

Amazon Elastic Transcoder

- Amazon Elastic Transcoder is media transcoding in the cloud.
- It is designed to be a highly scalable, easy to use and a cost effective way for developers and businesses to convert (or “transcode”) media files from their source format into versions that will playback on devices like smartphones, tablets and PCs.
- Amazon Elastic Transcoder manages all aspects of the transcoding process for you transparently and automatically.
- There’s no need to administer software, scale hardware, tune performance, or otherwise manage transcoding infrastructure.
- You simply create a transcoding “job” specifying the location of your source video and how you want it transcoded.
- Amazon Elastic Transcoder also provides transcoding presets for popular output formats, which means that you don’t need to guess about which settings work best on particular devices.
- All these features are available via service APIs and the AWS Management Console.

Deployment and Management

AWS Identity and Access Management (IAM)

- AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users.
- Using IAM, you can create and manage AWS users and groups and use permissions to allow and deny their access to AWS resources. IAM allows you to:
- Manage IAM users and their access - You can create users in IAM, assign them individual security credentials (i.e., access keys, passwords, and Multi-Factor Authentication devices) or

request temporary security credentials to provide users access to AWS services and resources. You can manage permissions in order to control which operations a user can perform.

- Manage IAM roles and their permissions - You can create roles in IAM, and manage permissions to control which operations can be performed by the entity, or AWS service, that assumes the role. You can also define which entity is allowed to assume the role.
- Manage federated users and their permissions - You can enable identity federation to allow existing identities (e.g. users) in your enterprise to access the AWS Management Console, to call AWS APIs, and to access resources, without the need to create an IAM user for each identity.

AWS CloudTrail

AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you.

The recorded information includes

- The identity of the API caller,
- The time of the API call,
- The source IP address of the API caller,
- The request parameters,
- The response elements returned by the AWS service.

With CloudTrail, you can get a history of AWS API calls for your account, including API calls made via the AWS Management Console, AWS SDKs, command line tools, and higher-level AWS services (such as AWS CloudFormation). The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing.

Amazon CloudWatch

- Amazon CloudWatch provides monitoring for AWS cloud resources and the applications customers run on AWS.
- Developers and system administrators can use it to collect and track metrics, gain insight, and react immediately to keep their applications and businesses running smoothly.
- Amazon CloudWatch monitors AWS resources such as Amazon EC2 and Amazon RDS DB Instances, and can also monitor custom metrics generated by a customer's applications and

services.

- With Amazon CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health.
- Amazon CloudWatch provides a reliable, scalable, and flexible monitoring solution that you can start using within minutes. You no longer need to set up, manage, or scale your own monitoring systems and infrastructure.
- Using Amazon CloudWatch, you can easily monitor as much or as little metric data as you need. Amazon CloudWatch lets you programmatically retrieve your monitoring data, view graphs, and set alarms to help you troubleshoot, spot trends, and take automated action based on the state of your cloud environment.

AWS Elastic Beanstalk

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with popular programming languages such as **Java, .NET, PHP, Node.js, Python and Ruby**.

- You simply upload your application and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling and application health monitoring.
- At the same time, with Elastic Beanstalk, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.
- Most existing application containers or platform-as-a-service solutions, while reducing the amount of programming required, significantly diminish developers' flexibility and control.
- Developers are forced to live with all the decisions pre-determined by the vendor with little to no opportunity to take back control over various parts of their application's infrastructure. However, with Elastic Beanstalk, you retain full control over the AWS resources powering your application.
- If you decide you want to take over some (or all) of the elements of their infrastructure, you can do so seamlessly by using Elastic Beanstalk's management capabilities.
- To ensure easy portability of your application, Elastic Beanstalk is built using familiar application/web servers such as Apache HTTP Server, Apache Tomcat, Nginx, Passenger and IIS 7.5/8.

AWS Cloud Formation

- AWS Cloud Formation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion.
- You can use AWS Cloud Formation's sample templates or create your own templates to describe the AWS resources, and any associated dependencies or runtime parameters, required to run your application.
- You don't need to figure out the order in which AWS services need to be provisioned or the subtleties of how to make those dependencies work. AWS Cloud Formation takes care of this for you.
- Once deployed, you can modify and update the AWS resources in a controlled and predictable way. This allows you to version control your AWS infrastructure in the same way as you version control your software.
- You can deploy and update a template and its associated collection of resources (called a stack) using the AWS Management Console, AWS CloudFormation command line tools, or CloudFormation API.
- AWS CloudFormation is available at no additional charge, and you pay only for the AWS resources needed to run your applications.

AWS OpsWorks

- AWS OpsWorks AWS OpsWorks is an application management service that makes it easy for DevOps users to model and manage the entire application from load balancers to databases.
- Start from templates for common technologies like Ruby, Node.JS, PHP, and Java, or build your own using Chef recipes to install software packages and perform any task that you can script.
- AWS OpsWorks can scale your application using automatic load-based or time-based scaling and maintain the health of your application by detecting failed instances and replacing them.
- You have full control of deployments and automation of each component.

AWS CloudHSM

- The AWS CloudHSM service helps you meet corporate, contractual and regulatory

compliance requirements for data security by using dedicated Hardware Security Module (HSM) appliances within the AWS cloud.

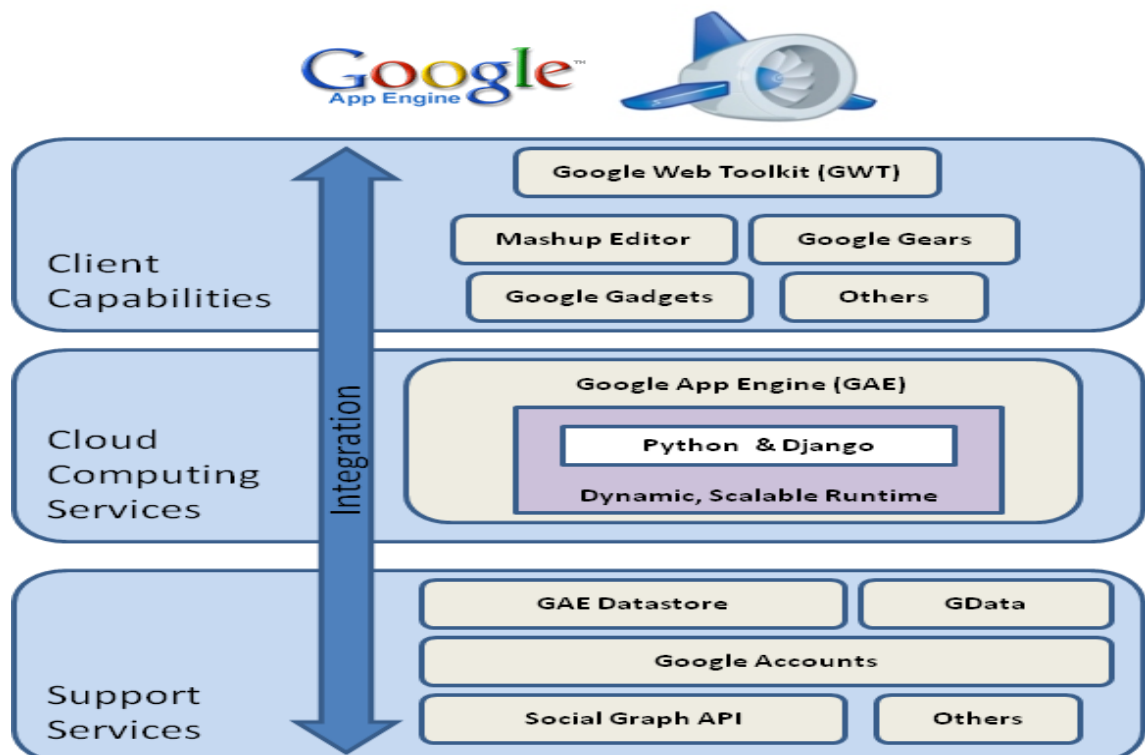
- AWS and AWS Marketplace partners offer a variety of solutions for protecting sensitive data within the AWS platform, but for applications and data subject to rigorous contractual or regulatory requirements for managing cryptographic keys, additional protection is sometimes necessary.
- Until now, your only option was to store the sensitive data (or the encryption keys protecting the sensitive data) in your on-premise datacenters.
- Unfortunately, this either prevented you from migrating these applications to the cloud or significantly slowed their performance.
- The AWS CloudHSM service allows you to protect your encryption keys within HSMs designed and validated to government standards for secure key management.
- You can securely generate, store, and manage the cryptographic keys used for data encryption such that they are accessible only by you.
- AWS CloudHSM helps you comply with strict key management requirements without sacrificing application performance.
- The AWS CloudHSM service works with Amazon Virtual Private Cloud (VPC).
- CloudHSMs are provisioned inside your VPC with an IP address that you specify, providing simple and private network connectivity to your Amazon Elastic Compute Cloud (EC2) instances.
- Placing CloudHSMs near your EC2 instances decreases network latency, which can improve application performance.
- AWS provides dedicated and exclusive access to CloudHSMs, isolated from other AWS customers. Available in multiple Regions and Availability Zones (AZs), AWS CloudHSM allows you to add secure and durable key storage to your Amazon EC2 applications.

Google App Engine

- Google AppEngine is a PaaS implementation that provides services for developing and hosting scalable Web applications.
- If you want to get an app on the cloud, the Google App Engine is the perfect tool to use and to make this dream reality. In essence, you write a bit of code in Python, tweak some HTML code, and then you've got your app built, and it only takes a few minutes. It takes a few

minutes for a simple, basic app. It'll take longer if you have something more complex in mind.

- you don't have to worry about buying servers, load balancers, or DNS tables Google handles all the heavy lifting for you.
- Developers can develop applications in Java, Python, and Go, a new programming language developed by Google to simplify the development of Web applications.
- Application usage of Google resources and services is metered by AppEngine, which bills users when their applications finish their free quotas.
- Google app engine is free for moderate levels of use. Every person that possesses a Gmail account can have a number of ten free applications running on the Google infrastructure and in case of one of them becoming very popular and the traffic going above the allowed levels of the free account, one can pay to use more of Google's resources.
- As the application scales, all the hardware, data storage, backup, and network provisioning for the customer are taken care of by Google's engineers.
- The payment for Google's resources is likely to be way lower than maintaining the same resources by the customer themselves. Google focuses on providing hardware and network, while the customer focuses on development and the user community around his application



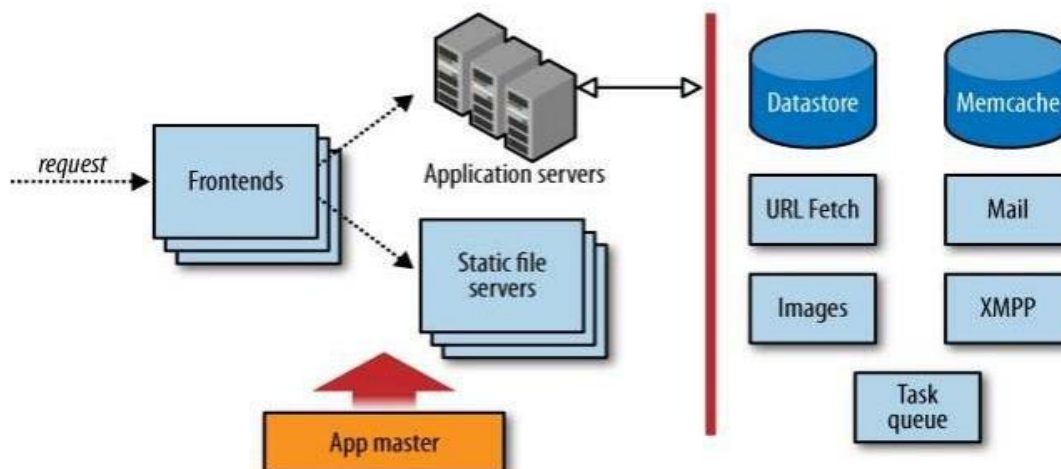
Architecture and core concepts

AppEngine is a platform for developing scalable applications accessible through the Web as shown in fig. The platform is logically divided into four major components:

- Infrastructure
- the runtime environment
- storage
- set of scalable services

Infrastructure

- AppEngine hosts Web applications, and its primary function is to serve users requests efficiently. To do so, AppEngine's infrastructure takes advantage of many servers available within Google datacenters.
- For each HTTP request, AppEngine locates the servers hosting the application that processes the request, evaluates their load, and, if necessary, allocates additional resources (i.e., servers) or redirects the request to an existing server.
- The particular design of applications, which does not expect any state information to be implicitly maintained between requests to the same application, simplifies the work of the infrastructure, which can redirect each of the requests to any of the servers hosting the target application or even allocate a new one.
- The infrastructure is also responsible for monitoring application performance and collecting statistics on which the billing is calculated.



Runtime environment

The runtime environment represents the execution of applications hosted on AppEngine. With reference to the AppEngine infrastructure code, which is always active and running, the run-time comes into existence when the request handler starts executing and terminates once the handler has completed.

- **Sandboxing**

One of the major responsibilities of the runtime environment is to provide the application environment with an isolated and protected context in which it can execute without causing a threat to the server and without being influenced by other applications. In other words, it provides applications with a sandbox.

Currently, AppEngine supports applications that are developed only with managed or interpreted languages, which by design require a runtime for translating their code into executable instructions.

Therefore, sandboxing is achieved by means of modified runtimes for applications that disable some of the common features normally available with their default implementations.

If an application tries to perform any operation that is considered potentially harmful, an exception is thrown and the execution is interrupted.

Some of the operations that are not allowed in the sandbox include writing to the server's file system;

- accessing computer through network besides using Mail, UrlFetch, and XMPP;
- executing code outside the scope of a request, a queued task, and a cron job;
- processing a request for more than 30 seconds.

- **Technology used**

Currently, it is possible to develop AppEngine applications using three different languages and related technologies:

Java

- AppEngine currently supports Java 6, and developers can use the common tools for Web application development in Java, such as the Java ServerPages(JSP), and the applications interact with the environment by using the Java Servlet standard.

- access to AppEngine services is provided by means of Java libraries that expose specific interfaces of provider-specific implementations of a given abstraction layer.
- Developers can create applications with the AppEngine Java SDK, which allows developing applications with either Java 5 or Java 6 and by using any Java library that does not exceed the restrictions imposed by the sandbox.

Python

- Support for Python is provided by an optimized Python 2.5.2 interpreter. As with Java, the run-time environment supports the Python standard library, but some of the modules that implement potentially harmful operations have been removed, and attempts to import such modules or to call specific methods generate exceptions.
- To support application development, AppEngine offers a rich set of libraries connecting applications to AppEngine services. In addition, developers can use a specific Python Web application framework, called webapp simplifying the development of Web applications.

Go

- The Go runtime environment allows applications developed with the Go programming language to be hosted and executed in AppEngine.
- Currently the release of Go that is supported by AppEngine is r58.1. The SDK includes the compiler and the standard libraries for developing applications in Go and interfacing it with AppEngine services.
- As with the Python environment, some of the functionalities have been removed or generate a runtime exception. In addition, developers can include third-party libraries in their applications as long as they are implemented in pure Go.

Storage

AppEngine provides various types of storage, which operate differently depending on the volatility of the data. There are three different levels of storage:

- MemCache that is storage in memory-cache
- Data Store that is storage for semi-structured data
- Static file servers that is long-term storage for static data.

data store and the use of static file servers described here and MemCache in the application services

section.

Static file servers

Web applications are composed of dynamic and static data. Dynamic data are a result of the logic of the application and the interaction with the user. Static data often are mostly constituted of the components that define the graphical layout of the application (CSS files, plain HTML files, JavaScript files, images, icons, and sound files) or data files. These files can be hosted on static file servers, since they are not frequently modified. Such servers are optimized for serving static content, and users can specify how dynamic content should be served when uploading their applications to AppEngine.

DataStore

- DataStore is a service that allows developers to store semi-structured data. The service is designed to scale and optimized to quickly access data.
- DataStore can be considered as a large object data base to store objects those can be retrieved by a specified key. Both the type of the key and the structure of the object can vary.
- The infrastructure of DataStore is based on Bigtable that is a redundant, distributed, and semistructured data store that organizes data in the form of tables queries, their implementation is substantially different.
- DataStore has been designed to be extremely fast in returning result sets; to do so it needs to know in advance all the possible queries that can be done for a given kind, because it stores for each of them a separate index. The indexes are provided by the user while uploading the application to AppEngine and can be automatically defined by the development server.
- When the developer tests the application, the server monitors all the different types of queries made against the simulated data store and creates an index for them. The structure of the indexes is saved in a configuration file and can be further changed by the developer before uploading the application. The use of precomputed indexes makes the query execution time-independent from the size of the stored data but only influenced by the size of the result set.
- DataStore also provides facilities for creating indexes on data and to update data within the context of a transaction. Indexes are used to support and speed up queries. Returned result sets can be sorted by key value or properties value. Even though the queries are quite similar to SQL queries, their implementation is substantially different.

- The implementation of transaction is limited in order to keep the store scalable and fast. AppEngine ensures that the update of a single entity is performed atomically. Multiple operations on the same entity can be performed within the context of a transaction. It is also possible to update multiple entities atomically.
- This is only possible if these entities belong to the same entity group. The entity group to which an entity belongs is specified at the time of entity creation and cannot be changed later.
- With regard to concurrency, AppEngine uses an optimistic concurrency control: If one user tries to update an entity that is already being updated, the control returns and the operation fails. Retrieving an entity never incurs into exceptions.

Application services

Applications hosted on AppEngine take the most from the services provided through the runtime environment. These services simplify most of the common operations that are performed in Web applications such as

- access to data
- account management
- integration of external resources
- messaging and communication
- image manipulation
- asynchronous computation.

UrlFetch

Web 2.0 has introduced the concept of composite Web applications. Different resources are put together and organized as meshes within a single Web page.

Meshes are fragments of HTML generated in different ways. They can be directly obtained from a remote server or from an XML document retrieved from a Web service, or they can be obtained by the browser as the result of an embedded and remote component.

A common characteristic of all these examples is the fact that the resource is not local to the server and often not even in the same administrative domain.

Therefore, it is fundamental for Web applications to be able to retrieve remote resources. The sandbox environment does not allow applications to open arbitrary connections through sockets, but it does provide developers with the capability of retrieving a remote resource through HTTP/HTTPS by means of the **urlfetch service**.

Applications can make synchronous and asynchronous Web requests and integrate the resources obtained in this way into the normal request handling cycle of the application. One of the interesting features of UrlFetch is the ability to set deadlines for requests so that they can be completed (or aborted) within a given time.

UrlFetch is not only used to integrate meshes into a Web page but also to hold remote Web services according to the SOA reference model for distributed applications.

MemCache

AppEngine provides developers with access to fast and reliable storage, which is DataStore. The main objective of the service is to serve as a scalable and long-term storage, where data are persisted to disk redundantly in order to ensure reliability and availability of data against failures.

This design poses a limit on how much faster the store can be compared to other solutions, especially for objects that are frequently accessed—for example, at each Web request. AppEngine provides caching services by means of **Memcache**.

This is a distributed in-memory cache that is optimized for fast access and provides developers with a volatile store for the objects that are frequently accessed. The caching algorithm implemented by MemCache will automatically remove the objects that are rarely accessed.

The use of MemCache can significantly reduce the access time to data that is developers can structure their applications so that each object is first looked up into MemCache and if there is a miss, it will be retrieved from DataStore and put into the cache for future lookups.

Mail and instant messaging

Communication is another important aspect of Web applications. It is common to use email for following up with users about operations performed by the application.

Email can also be used to trigger activities in Web applications. AppEngine provides developers with the ability to send and receive mails through mail. The service allows sending email on behalf of the

application to specific user accounts. It is also possible to include several types of attachments and to target multiple recipients.

Mail operates asynchronously, and in case of failed delivery the sending address is notified through an email detailing the error. AppEngine provides also another way to communicate with the external world that is the Extensible Messaging and Presence Protocol (XMPP).

Any chat service that supports XMPP, such as Google Talk, can send and receive chat messages to and from the Web application, which is identified by its own address.

The chat is a communication medium mostly used for human interactions, XMPP can be conveniently used to connect the Web application with chat bots or to implement a small administrative console.

Account management

Web applications often keep various data that customize their interaction with users. These data normally go under the user profile and are attached to an account.

App Engine simplifies account management by allowing developers to leverage Google account management by means of google accounts. The integration with the service also allows Web applications to offload the implementation of authentication capabilities to Google's authentication system.

Using Google Accounts, Web applications can conveniently store profile settings in the form of key value pairs, attach them to a given Google account, and quickly retrieve them once the user authenticates.

With respect to a custom solution, the use of Google Accounts requires users to have a Google account, but it does not require any further implementation. The use of Google Accounts is particularly advantageous for developing Web applications within a corporate environment using Google Apps. The applications can be easily integrated with all the other services (and profile settings) included in Google Apps.

Image manipulation

Web applications has pages with graphics. Often simple operations, such as adding watermarks or applying simple filters, are required.

AppEngine allows applications to perform image resizing, rotation, mirroring, and enhancement by means of image manipulation, a service that is also used in other Google products. Image Manipulation is mostly designed for lightweight image processing and is optimized for speed.

Compute services

Web applications are mostly designed to interface applications with users by means of a ubiquitous channel, that is, the Web. Most of the interaction is performed synchronously that means Users navigate the Web pages and get instantaneous feedback in response to their actions.

This feedback is often the result of some computation happening on the Web application, which implements the intended logic to serve the user request. Sometimes this approach is not applicable. for example, in long computations or when some operations need to be triggered at a given point in time. A good design for these scenarios provides the user with immediate feedback and a notification once the required operation is completed.

AppEngine offers additional services such as Task Queues and Cron Jobs that simplify the execution of computations.

Task queues

Task queues allow applications to submit a task for a later execution. This service is particularly useful for long computations that cannot be completed within the maximum response time of a request handler.

The service allows users to have up to 10 queues that can execute tasks at a configurable rate. In fact, a task is defined by a Web request to a given URL, and the queue invokes the request handler by passing the payload as part of the Web request to the handler.

It is the responsibility of the request handler to perform the “task execution,” which is seen from the queue as a simple Web request. The queue is designed to re-execute the task in case of failure in order to avoid transient failures preventing the task from a successful completion.

Cron jobs

Sometimes the length of computation might not be the primary reason that an operation is not performed within the scope of the Web request.

It might be possible that the required operation needs to be performed at a specific time of the day, which does not coincide with the time of the Web request. In this case, it is possible to schedule the required operation at the desired time by using the Cron jobs.

This service operates similarly to Task Queues but invokes the request handler specified in the task at a given time and does not reexecute the task in case of failure. This behavior can be useful to implement maintenance operations or send periodic notifications.

Application life cycle

AppEngine provides support for almost all the phases characterizing the life cycle of an application:

- testing and development,
- deployment and monitoring.

The SDKs released by Google provide developers with most of the functionalities required by these tasks. Currently there are two SDKs available for development: Java SDK and Python SDK.

Application development and testing

Developers can start building their Web applications on a local development server. This is a self contained environment that helps developers tune applications without uploading them to AppEngine.

The development server simulates the AppEngine runtime environment by providing a mock implementation of DataStore, MemCache, UrlFetch, and the other services leveraged by Web applications. Besides hosting Web applications, the development server contains a complete set of monitoring features that are helpful to profile the behavior of applications, especially regarding access to the DataStore service and the queries performed against it.

This is a particularly important feature that will be of relevance in deploying the application to AppEngine. The development server analyzes application behavior while running and traces all the queries made during testing and development, thus providing the required information about the indexes to be built.

Java SDK

The Java SDK provides developers with the facility for building applications with the Java 5 and Java 6 runtime environments. Alternatively, it is possible to develop applications within the Eclipse development environment by using the Google AppEngine plug-in, which integrates the features of

the SDK within the powerful Eclipse environment.

Using the Eclipse software installer, it is possible to download and install Java SDK, Google Web Toolkit, and Google AppEngine plug-ins into Eclipse. These three components allow developers to program powerful and rich Java applications for AppEngine.

The SDK supports the development of applications by using the servlet abstraction, which is a common development model. Together with servlets, many other features are available to build applications.

Developers can easily create Web applications by using the eclipse web platform which provides a set of tools and components. The plug-in allows developing, testing, and deploying applications on AppEngine. Other tasks, such as retrieving the log of applications, are available by means of command-line tools that are part of the SDK.

Python SDK

The Python SDK allows developing Web applications for App Engine with Python 2.5. It provides a standalone tool, called Google App Engine Launcher for managing Web applications locally and deploying them to App Engine.

The tool provides a convenient user interface that lists all the available Web applications, controls their execution, and integrates them with the default code editor for editing application files.

The launcher provides access to some important services for application monitoring and analysis, such as the logs, the SDK console, and the dashboard.

The log console captures all the information that is logged by the application while it is running. The console SDK provides developers with a Web interface via which they can see the application profile in terms of utilized resource. This feature is particularly useful because it allows developers to preview the behavior of the applications once they are deployed on AppEngine, and it can be used to tune applications made available through the runtime.

The Python implementation of the SDK also comes with an integrated Web application framework called webapp that includes a set of models, components, and tools that simplify the development of Web applications and enforce a set of coherent practices.

This is not the only Web framework that can be used to develop Web applications. There are dozens

of available Python Web frameworks that can be used. However, due to the restrictions enforced by the sandboxed environment, all of them cannot be used seamlessly.

The SDK is completed by a set of command-line tools that allows developers to perform all the operations available through the launcher and more from the command shell.

Application deployment and management

Once the application has been developed and tested, it can be deployed on AppEngine with a simple click or command-line tool.

Before performing such task, it is necessary to create an application identifier, which will be used to locate the application from the Web browser by typing the address **http://application-id.appspot.com**.

Alternatively, it is also possible to map the application with a registered DNS domain name. This is particularly useful for commercial development, where users want to make the application available through a more appropriate name.

An application identifier is mandatory because it allows unique identification of the application while it's interacting with AppEngine. Developers use an app identifier to upload and update applications.

It is possible to register an application identifier by logging into AppEngine and selecting the "Create application" option. It is also possible to provide an application title that is descriptive of the application; the title can be changed over time.

Once an application identifier has been created, it is possible to deploy an application on AppEngine. This task can be done using either the respective development environment (Google App Engine Launcher and Google App Engine plug-in) or the command-line tools. Once the application is uploaded, nothing else needs to be done to make it available. AppEngine will take care of everything.

Developers can then manage the application by using the administrative console. This is the primary tool used for application monitoring and provides users with insight into resource usage (CPU, bandwidth) and services and other useful counters. It is also possible to manage multiple versions of a single application, select the one available for the release, and manage its billing-related issues.

Cost model

AppEngine provides a free service with limited quotas that get reset every 24 hours. Once the

application has been tested and tuned for AppEngine, it is possible to set up a billing account and obtain more allowance and be charged on a pay-per-use basis. This allows developers to identify the appropriate daily budget that they want to allocate for a given application.

An application is measured against billable quotas, fixed quotas, and per-minute quotas. Google AppEngine uses these quotas to ensure that users do not spend more than the allocated budget and that applications run without being influenced by each other from a performance point of view. Billable quotas identify the daily quotas that are set by the application administrator and are defined by the daily budget allocated for the application.

AppEngine will ensure that the application does not exceed these quotas. Free quotas are part of the billable quota and identify the portion of the quota for which users are not charged. Fixed quotas are internal quotas set by AppEngine that identify the infrastructure boundaries and define operations that the application can carry out on the infrastructure (services and runtime).

The costing model also includes per-minute quotas, which are defined in order to avoid applications consuming all their credit in a very limited period of time, monopolizing a resource, and creating service interruption for other applications.

Once an application reaches the quota for a given resource, the resource is consumed and will not be available to the application until the quota is recharge again. Once a resource is consumed, subsequent requests to that resource will generate an error or an exception. Resources such as CPU time and incoming or outgoing bandwidth will return an “HTTP 403” error page to users.

Resources and services quotas are organized into

- free default quotas
- billing-enabled default quotas.

For these two categories, a daily limit and a maximum rate are defined. A detailed explanation of how quotas work, their limits, and the amount that is charged to the user can be found on the AppEngine Website at the following Internet address

<http://code.google.com/appengine/docs/quotas.html>.

Microsoft Azure Design

- Microsoft’s Azure Services Platform is a tool provided for developers who want to write

applications that are going to run partially or entirely in a remote data center.

- The Azure Services Platform (Azure) is an Internet-scale cloud services platform hosted in Microsoft datacenters, which provides an operating system and a set of developer services that can be used individually or together.
- Azure can be used to build new applications to run from the cloud or to enhance existing applications with cloud-based capabilities, and it forms the foundation of all Microsoft's cloud offerings.
- Its open architecture gives developers the choice to build web applications, applications running on connected devices, PCs, servers, or hybrid solutions offering the best of online and on premises.

Components

The components of the Azure Services Platform can be used by local applications running on a variety of systems, including various flavors of Windows, mobile devices, and others. Those components include:

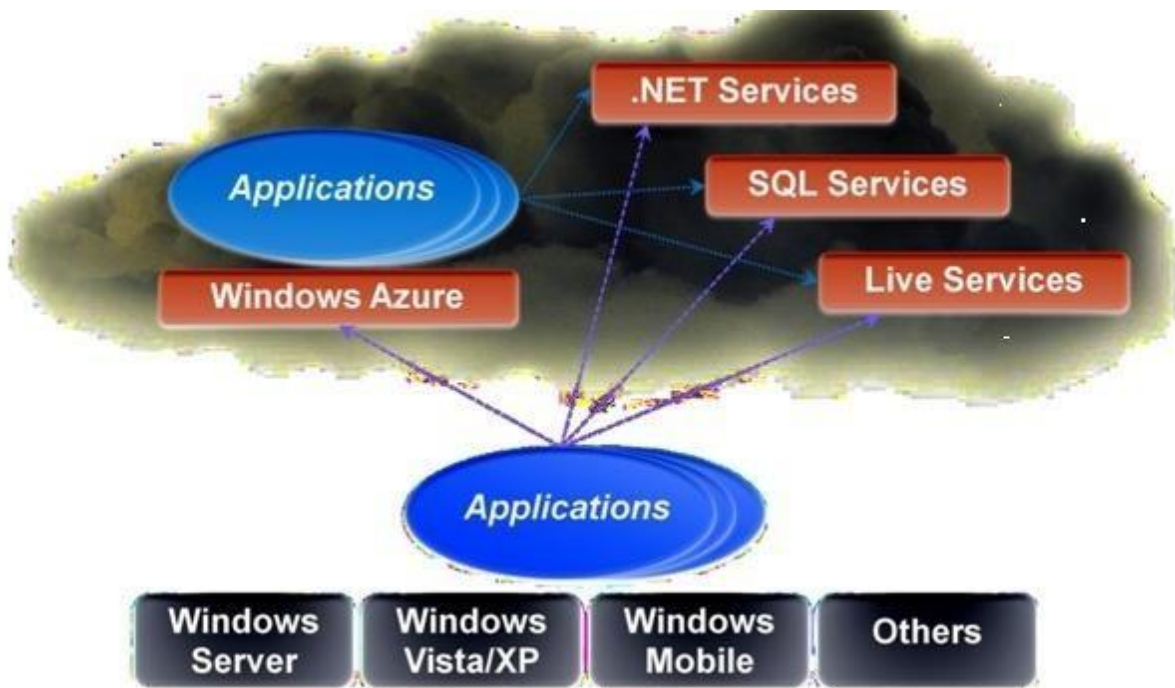


Figure 1: The Azure Services Platform supports applications running in the cloud and on local systems.

- *Windows Azure*: Provides a Windows-based environment for running applications and storing data on servers in Microsoft data centers.
- *Microsoft .NET Services*: Offers distributed infrastructure services to cloud-based and local applications.
- *Microsoft SQL Services*: Provides data services in the cloud based on SQL Server.
- *Live Services*: Through the Live Framework, provides access to data from Microsoft's live applications and others. The Live Framework also allows synchronizing this data across desktops and devices, finding and downloading applications, and many more.

Each component of the Azure Services Platform has its own role to play. These services can be managed and controlled through the Windows Azure Management Portal, which acts as an administrative console for all the services offered by the Azure platform. In this section, we present the core features of the major services available with Azure.

Windows Azure

“What is Windows Azure?”

Windows Azure is anything you want it to be...

Of course this kind of definition doesn't really explain what Windows Azure is, so let's look at the definition that Microsoft uses for describing Windows Azure:

“Windows Azure is an open and flexible cloud platform that enables you to quickly build, deploy, and manage applications across a global network of Microsoft-managed datacenters. You can build applications using any language, tool, or framework. And you can integrate your public cloud applications with your existing IT environment.”

This definition tells us that Windows Azure is a cloud platform, which means you can use it for running your business applications, services, and workloads in the cloud. It also includes some key words such as:

Open Windows Azure provides a set of cloud services that allow you to build and deploy cloud-based applications using almost any programming language, framework, or tool.

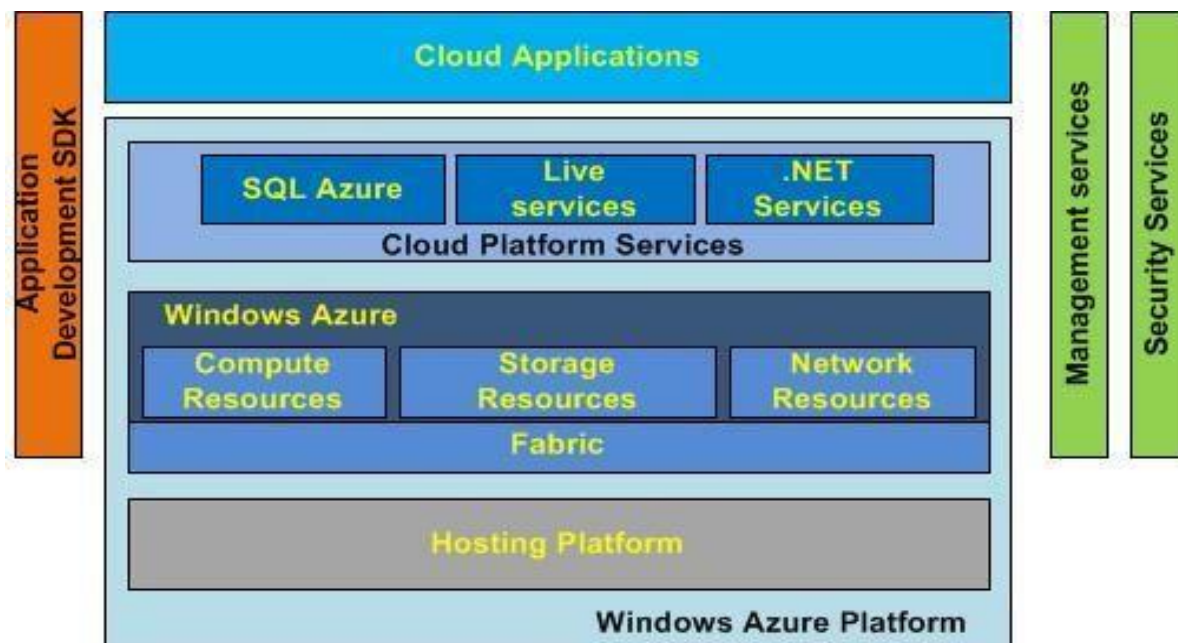
Flexible Windows Azure provides a wide range of cloud services that can let you do everything from hosting your company's website to running big SQL databases in the cloud. It also includes different features that can help deliver high performance and low latency for cloud-based applications.

Microsoft-managed Windows Azure services are currently hosted in several datacenters spread across the United States, Europe, and Asia. These datacenters are managed by Microsoft and provide expert global support on a 24x7x365 basis.

Compatible Cloud applications running on Windows Azure can easily be integrated with on-premises IT environments that utilize the Microsoft Windows Server platform.

Windows Azure platform

The Windows Azure platform is made up of a foundation layer and a set of developer services that can be used to build scalable applications. These services cover compute, storage, networking, and identity management, which are tied together by middleware called APPFabric

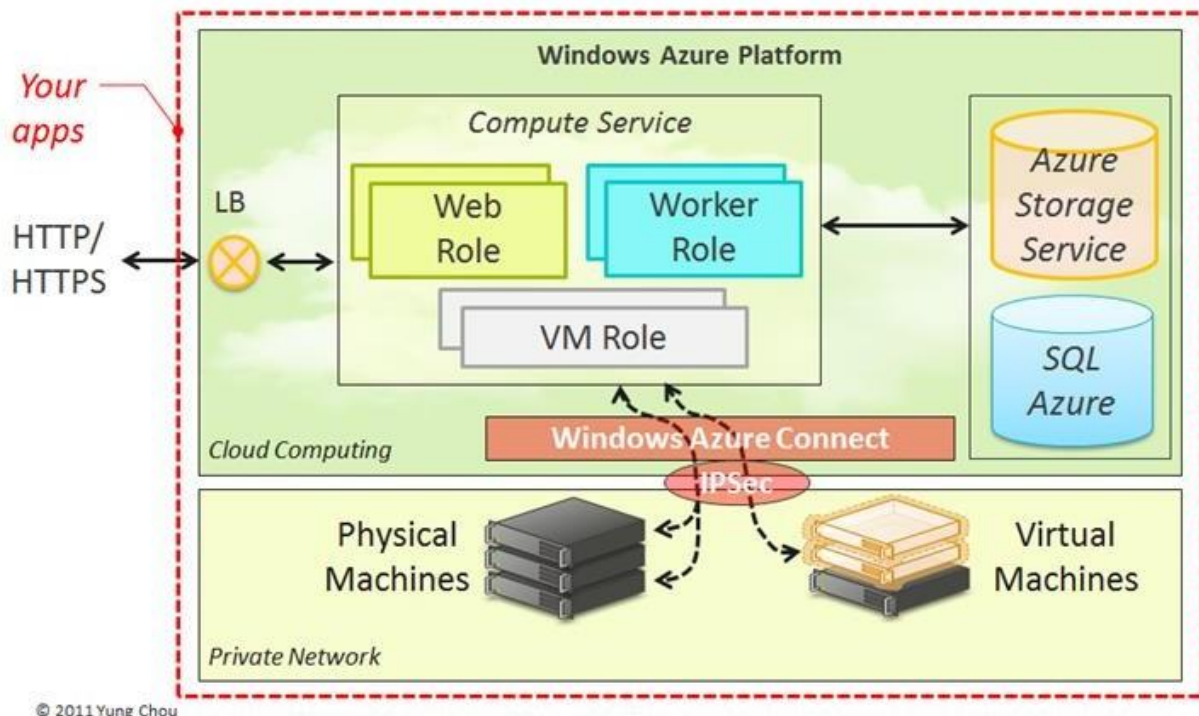


Windows Azure provides businesses with four basic categories of cloud-based services:

- Compute services
- Network services
- Data services
- App services

The number of different Windows Azure services available is expanding, with new services continually being added and existing services enhanced as the Windows Azure platform continues to evolve.

Windows Azure Connect Concept



Compute services

Compute services are the core components of Microsoft Windows Azure, and they are delivered by means of the abstraction of roles. A role is a run time environment that is customized for a specific compute task. Roles are managed by the Azure operating system and instantiated on demand in order to address surges in application demand. Currently, there are three different roles: Web role, Worker role, and Virtual Machine (VM) role.

- **Virtual Machines role:** This service provides you with a general purpose computing environment that lets you create, deploy, and manage virtual machines running in the Windows Azure cloud.
- **Web Role:** The Web role is designed to implement scalable Web applications. Web roles represent the units of deployment of Web applications within the Azure infrastructure. They are hosted on the IIS 7 Web Server, which is a component of the infrastructure that supports Azure.

- When Azure detects peak loads in the request made to a given application, it instantiates multiple Web roles for that application and distributes the load among them by means of a load balancer.
 - Since version 3.5, the .NET technology natively supports Web roles. Developers can directly develop their applications in Visual Studio, test them locally, and upload to Azure. It is possible to develop ASP.NET (ASP.NET Web Role and ASP.NET MVC2 Web Role) and WCF (WCF Service Web Role) applications.
 - Other Web technologies that are not integrated with IIS can still be hosted on Azure (i.e., Java Server Pages on Apache Tomcat), but there is no advantage to using a Web role over a Worker role.
- **Worker Role:** Worker roles are designed to host general compute services on Azure. They can be used to quickly provide compute power or to host services that do not communicate with the external world through HTTP.
 - A common practice for Worker roles is to use them to provide background processing for Web applications developed with Web roles. Developing a worker role is like a developing a service.
 - Compared to a Web role whose computation is triggered by the interaction with an HTTP client (i.e., a browser), a Worker role runs continuously from the creation of its instance until it is shut down.
 - The Azure SDK provides developers with convenient APIs and libraries that allow connecting the role with the service provided by the runtime and easily controlling its startup as well as being notified of changes in the hosting environment.
 - As with Web roles, the .NET technology provides complete support for Worker roles, but any technology that runs on a Windows Server stack can be used to implement its core logic. For example, Worker roles can be used to host Tomcat and serve JSP-based applications.

Network services

Windows Azure network services provide different options for how Windows Azure applications can be delivered to users and data centers. Windows Azure currently offers two different network services:

- **Virtual Network** This service allows you to treat the Windows Azure public cloud as if it is an extension of your on-premises datacenter.
- **Traffic Manager** This service allows you to route application traffic for the user who is using the application to Windows Azure data centers in three ways: for best performance, in round robin fashion, or using an Active/Passive failover configuration.

Data services

Windows Azure data services provide you with different ways of storing, managing, safe guarding, analyzing, and reporting business data. Windows Azure currently offers five different data services:

- **Data Management** This service store your business data in SQL databases, either with dedicated Microsoft SQL Server virtual machines, using Windows Azure SQL Database, using NoSQL Tables via REST, or using BLOB storage. Azure allows storing large amount of data in the form of binary large objects (BLOBs) by means of the blobs service This service is optimal to store large text or binary files. Two types of blobs are available:
 - **Block blobs** are composed of blocks and are optimized for sequential access; therefore they are appropriate for media streaming. Currently, blocks are of 4 MB, and a single block blob can reach 200 GB in dimension.
 - **Page blobs** are made of pages that are identified by an offset from the beginning of the blob. A page blob can be split into multiple pages or constituted of a single page. This type of blob is optimized for random access and can be used to host data different from streaming. Currently, the maximum dimension of a page blob can be 1 TB.
- **Business Analytics** This service enables ease of discovery and data enrichment using Microsoft SQL Server Reporting and Analysis Services or Microsoft SharePoint Server running in a virtual machine, Windows Azure SQL Reporting, the Windows Azure Marketplace, or HDInsight, a Hadoop implementation for Big Data.
- **HDInsight** This is Microsoft's Hadoop-based service which brings a 100 percent Apache Hadoop solution to the cloud.
- **Cache** This service provides a distributed caching solution that can help speed up your cloud-based applications and reduce database load.
- **Backup** This service helps you protect your server data offsite by using automated and manual backups to Windows Azure.
- **Recovery Manager** Windows Azure Hyper-V Recovery Manager helps you protect business

critical services by coordinating the replication and recovery of System Center 2012 private clouds at a secondary location.

App services

Windows Azure app services provide you with ways of enhancing the performance, security, discoverability, and integration of your cloud apps that are running. Windows Azure currently offers seven different app services:

- **Media Services** This service allows you to build workflows for the creation, management, and distribution of media using the Windows Azure public cloud.
- **Messaging** This consists of two services (Windows Azure Service Bus and Windows Azure Queue) that allow you to keep your apps connected across your private cloud environment and the Windows Azure public cloud.
- **Notification Hubs** This service provides a highly scalable, cross-platform push notification infrastructure for applications running on mobile devices.
- **BizTalk Services** This service provides Business-to-Business (B2B) and Enterprise Application Integration (EAI) capabilities for delivering cloud and hybrid integration solutions.
- **Active Directory** This service provides you with identity management and access control capabilities for your cloud applications.
- **Multifactor Authentication** This service provides an extra layer of authentication, in addition to the user's account credentials, in order to better secure access for both on-premises and cloud applications.

AZURE Design

Azure is designed in several layers, with different things going on under the hood.

Layer Zero

Layer Zero is Microsoft's Global Foundational Service. GFS is akin to the hardware abstraction layer (HAL) in Windows. It is the most basic level of the software that interfaces directly with the servers.

Layer One

Layer One is the base Azure operating system. It used to be code-named "Red Dog," and was designed by a team of operating system experts at Microsoft. Red Dog is the technology that

networks and manages the Windows Server 2008 machines that form the Microsoft hosted cloud.

Red Dog is made up of four pillars:

- Storage (a file system)
- The fabric controller, which is a management system for deploying and provisioning
- Virtualized computation/VM
- Development environment, which allows developers to emulate Red Dog on their Desktops Red Dog is designed by Microsoft such that it only has to be deployed on a single machine, and then multiple instances of it can be duplicated to the rest of the machines in the cloud.

Layer Two

Layer Two provides the building blocks that run on Azure. These services are Live Mesh platform. Developers build on top of these lower-level services when building cloud apps. SharePoint Services and CRM Services are not the same as SharePoint Online and CRM Online. They are just the platform basics that do not include user interface elements.

Layer Three

At Layer Three exist the Azure-hosted applications. Some of the applications developed by Microsoft include SharePoint Online, Exchange Online, Dynamics CRM, and Online. Third parties will create other applications.

.Net Services

Running applications in the cloud is an important aspect of cloud computing, but it's far from the whole story. It's also possible to provide cloud-based services that can be used by either on- premises applications or cloud applications. Filling this gap is the goal of .NET Services. Originally known as BizTalk Services, the functions provided by .NET Services address common infrastructure challenges in creating distributed applications. Figure shows its components.

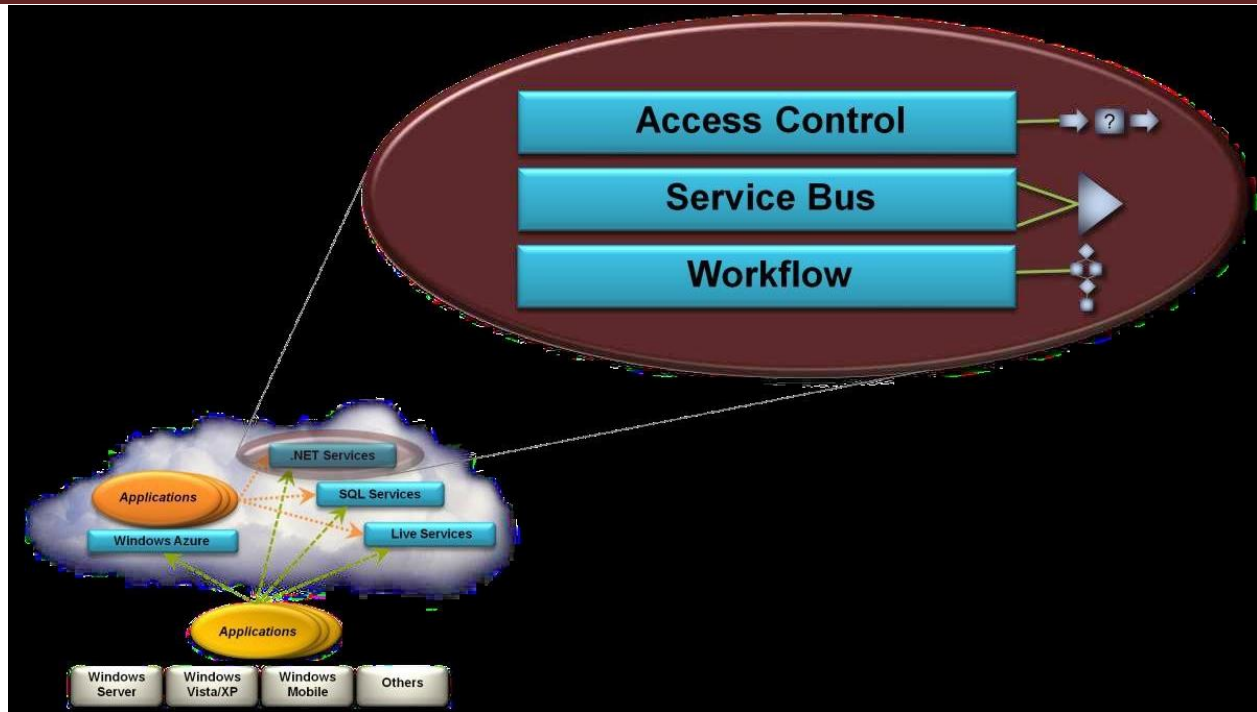


Figure 3: .NET Services provides cloud-based infrastructure that can be used by both cloud and on-premises applications.

The components of .NET Services are:

- **Access Control:** An increasingly common approach to identity is to have each user supply an application with a *token* containing some set of *claims*. The application can then decide what this user is allowed to do based on these claims. Doing this effectively across companies requires *identity federation*, which lets claims created in one identity scope be accepted in another. It might also require *claims transformation*, modifying claims when they're passed between identity scopes. The Access Control service provides a cloud-based implementation of both.
- **Service Bus:** Exposing an application's services on the Internet is harder than most people think. The goal of Service Bus is to make this simpler by letting an application expose Web services endpoints that can be accessed by other applications, whether on-premises or in the cloud. Each exposed endpoint is assigned a URI, which clients can use to locate and access the service. Service Bus also handles the challenges of dealing with network address translation and getting through firewalls without opening new ports for exposed applications.

□ **Workflow:** Creating composite applications, as in enterprise application integration, requires logic that coordinates the interaction among the various parts. This logic is sometimes best implemented using a workflow capable of supporting long-running processes. Built on Windows Workflow Foundation (WF), the Workflow service allows running this kind of logic in the cloud.

SQL Services

One of the most attractive ways of using Internet-accessible servers is to handle data. This means providing a core database, certainly, but it can also include more. The goal of SQL Services is to provide a set of cloud-based services for storing and working with many kinds of data, from unstructured to relational. Microsoft says that SQL Services will include a range of data-oriented facilities, such as reporting, data analytics, and others. The first SQL Services component to appear, however, is SQL Data Services. Figure 4 illustrates this idea.

- SQL Data Services, formerly known as SQL Server Data Services, provides a database in the cloud. As the figure suggests, this technology lets on-premises and cloud applications store and access data on Microsoft servers in Microsoft data centers.
- As with other cloud technologies, an organization pays only for what it uses, increasing and decreasing usage (and cost) as the organization's needs change. Using a cloud database also allows converting what would be capital expenses, such as investments in disks and database management systems (DBMSs), into operating expenses.

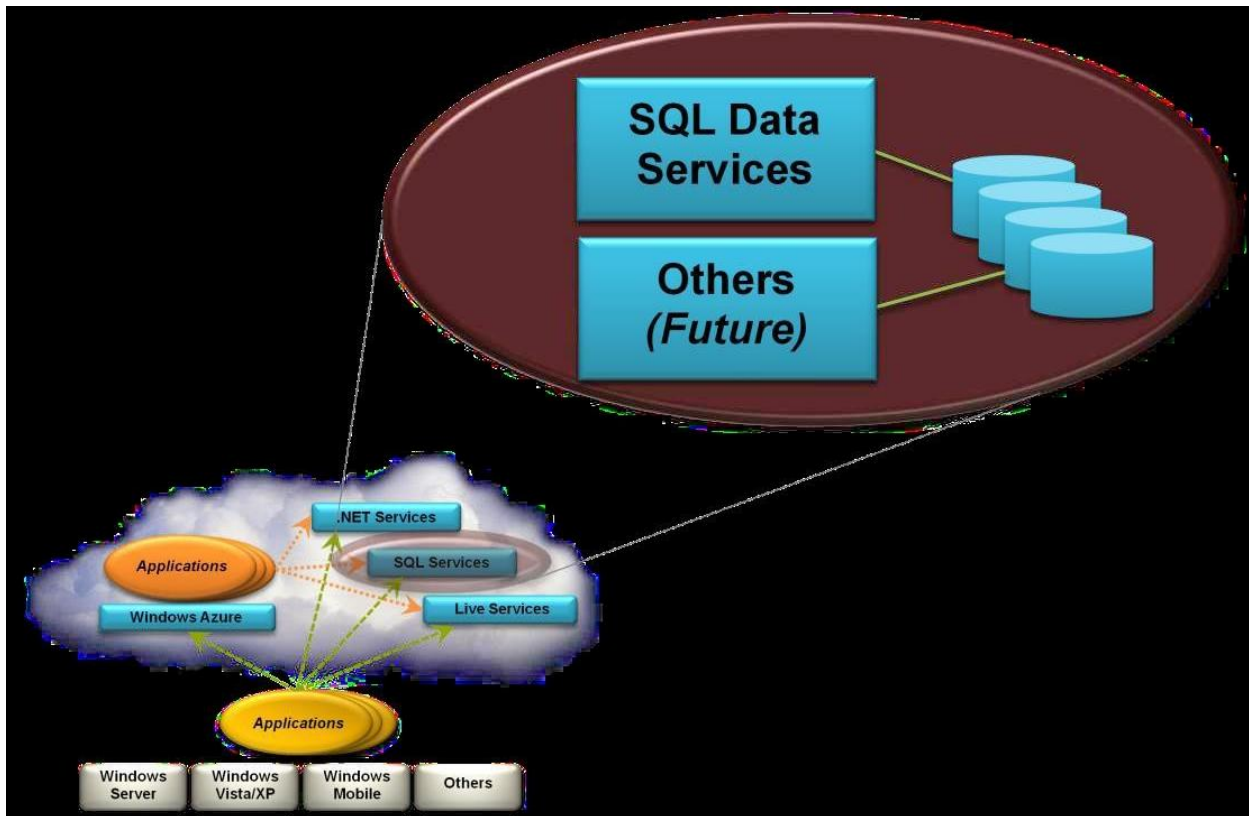


Figure 4: SQL Services provides data-oriented facilities in the cloud.

- A primary goal of SQL Data Services is to be broadly accessible. Toward this end, it exposes both SOAP and RESTful interfaces, allowing its data to be accessed in various ways. And because this data is exposed through standard protocols, SQL Data Services can be used by applications on any kind of system—it's not a Windows-only technology.
- Unlike the Windows Azure storage service, SQL Data Services is built on Microsoft SQL Server. Nonetheless, the service does not expose a traditional relational interface. Instead, SQL Data Services provides a hierarchical data model that doesn't require a pre-defined schema.
- Each data item stored in this service is kept as a property with its own name, type, and value. To query this data, applications can use direct RESTful access or a language based on the C# syntax defined by Microsoft's Language Integrated Query (LINQ).

advantages.

- SQL Data Services can provide better scalability, availability, and reliability than is possible by just running a relational DBMS in the cloud. The way it organizes and retrieves data makes replication and load balancing easier and faster than with a traditional relational approach.

- Another advantage is that SQL Data Services doesn't require customers to manage their own DBMS. Rather than worry about the mechanics, such as monitoring disk usage, servicing log files, and determining how many instances are required, a SQL Data Services customer can focus on what's important: the data.
- And finally, Microsoft has announced plans to add more relational features to SQL Data Services. Expect its functionality to grow. SQL Data Services can be used in a variety of ways.

Here are some examples:

□ An application might archive older data to SQL Data Services. For instance, think of an application that provides frequently updated RSS feeds. Information in these feeds that's more than, say, 30 days old probably isn't accessed often, but it still needs to be available. Moving this data to SQL Data Services could provide a low-cost, reliable alternative.

□ Suppose a manufacturer wishes to make product information available to both its dealer network and directly to customers. Putting this data in SQL Data Services would allow it to be accessed by applications running at the dealers and by a customer-facing Web application run by the manufacturer itself. Because the data can be accessed through RESTful and SOAP interfaces, the applications that use it can be written using any technology and run on any platform.

Like other components of the Azure Services Platform, SQL Data Services makes it simple to use its services: Just go to a Web portal and provide the necessary information. Whether it's for archiving data cheaply, making data accessible to applications in diverse locations, or other reasons, a cloud database can be an attractive idea. As new technologies become available under the SQL Services umbrella, organizations will have the option to use the cloud for more and more data-oriented tasks.

Live Services

Live Services is a set of building blocks within the Azure Services Platform that is used to handle user data and application resources. Live Services provides developers with a way to build social applications and experiences across a range of digital devices that can connect with one of the largest audiences on the Web.

The Live Framework offers a diverse set of functions that can be used in a variety of different ways.

Here are a few examples:

- A Java application running on Linux could rely on the Live Framework to access a user's contacts information. The application is unaware that the technology used to expose this information is the Live Framework; all it sees is a consistent HTTP interface to the user's data.
- A .NET Framework application might require its user to create a mesh, then use the Live Framework as a data caching and synchronization service. When the machine this application runs on is connected to the Internet, the application accesses a copy of its data in the cloud. When the machine is disconnected maybe it's running on a laptop that's being used on an airplane the application accesses a local copy of the same data. Changes made to any copy of the data are propagated by the Live Operating Environment.
- An ISV can create a mesh-enabled Web application that lets people keep track of what their friends are doing. This application, which can run unchanged on all of its user's systems, exploits several aspects of the Live Framework that support social applications. Because the Live Framework can expose information in a user's mesh as a feed, for example, the application can track updates from any of the user's friends. Because the Live Framework provides a delivery mechanism for mesh-enabled Web apps, viral distribution is possible, with each user inviting friends to use the application. And because the mesh automatically includes a user's Live Services contacts, the user can ask the application to invite friends by name, letting the application contact them directly.

The Live Framework provides a straightforward way to access Live Services data. Its data synchronization functions can also be applied in a variety of applications. For applications that need what it provides, this platform offers a unique set of supporting functions.

Windows Azure solutions

Microsoft has identified ten different solution categories where Windows Azure can bring significant benefits to businesses:

- Infrastructure Mobile
- Web Media Integration
- Identity & Access Management Big Data
- Dev & Test
- Storage, Backup, & Recovery Data Management

Let's briefly examine each solution category in more detail.

Infrastructure

Windows Azure can provide your business with on-demand infrastructure that can scale and adapt to your changing business needs. You can quickly deploy new virtual machines in minutes, and with pay-as-you-go billing you won't be penalized when you need to reconfigure your virtual machines. Windows Azure Virtual Machines even offers you a gallery of preconfigured virtual machine images you can choose from so you can get started as quickly as possible. You can also upload or download your virtual disks, load-balance your virtual machines, and integrate your virtual machines into your on-premises environment using virtual networks.

Mobile

Windows Azure lets you build and deploy a back-end cloud solution for your mobile device apps. You can use popular development platforms like .NET or NodeJS to create your solution, then deploy it to the cloud using Windows Azure Virtual Machines, Cloud Services, or Mobile Services. Windows Azure Mobile Services, in particular, provides cross-platform support for developing solutions for almost any platform including Windows Phone, Windows Store, Android, Apple iOS, and HTML5. Windows Azure Notification Hubs lets you push out notifications to users to enable real-time interactive applications, and you can use social media platforms from Microsoft, Google, Facebook, or Twitter for user authentication purposes.

Web

With support for both SNI and IP-based SSL certificates, and global datacenters with guaranteed SLA and 24/7 support available, Windows Azure can provide you with a robust and secure platform for giving your business website a global presence. You can use Windows Azure Active Directory for authentication and access control, and can securely store your website's business data in Windows Azure SQL Database, NoSQL Tables, BLOB storage. You can create your website using the language of your choice, such as ASP.NET, PHP, Node.js, Python, or even Classic ASP. And for even faster development, you can quickly build your site using a popular framework or template from the Windows Azure App Gallery, which includes WordPress, Umbraco, DotNetNuke, Drupal, Django, CakePHP, and Express.

Media

Windows Azure Media Services makes it easy to give your business a global media presence. You can quickly build end-to-end media workflows using services from both Microsoft and its partners. Your media can be protected using Digital Rights Management (DRM), and Advanced Encryption Standard (AES) or Playready can be used to protect it during playback.

Integration

Windows Azure provides several different options for integrating your existing on-premises infrastructure with your applications running in the Windows Azure public cloud. Windows Azure Service Bus can be used for communicating between your on-premises and cloud-based applications and services. Windows Azure BizTalk Services provides a robust business-to-business (B2B) and application integration PaaS in the cloud. And you can build your integration solution using the familiar tools of .NET and Visual Studio.

Identity & access management

Windows Azure Active Directory (Windows Azure AD) can provide you with identity services running in the cloud that you can use for managing access by employees, partners, and customers to your corporate assets, including both on-premises and cloud assets. You can even synchronize your on-premises Active Directory infrastructure with Windows Azure AD to provide single sign-on (SSO) for users to access your cloud applications, and Windows Azure Multifactor Authentication can be used to provide an additional layer of authentication to help protect your sensitive business data and applications.

Big data

Windows Azure enables you to quickly build a Hadoop cluster based on 100 percent Apache Hadoop. You can then use Windows Azure PowerShell and the Windows Azure Command-Line Interface to seamlessly integrate HDInsight into your existing analysis workflows and gain actionable insights from HDInsight by mining data with Microsoft Excel.

HDInsight supports a wide range of languages including .NET and Java, and .NET developers can also use LINQ and Hive for language-integrated query.

Dev & test

Windows Azure makes rapid application development and testing easy. Instead of having to go through a traditional procurement process and wait for the new hardware you ordered to arrive, you can simply use Windows Azure Virtual Machines to spin up as many virtual machines as you need and perform your application development and testing in the cloud. Then once your application has been validated, you can deploy it into a production environment that's identical to your test environment but also provides you with enhanced performance, infinite scalability, and global reach.

Storage, backup, & recovery

Windows Azure Storage can provide you with secure and reliable storage for all your business needs. Geo-replication across different geographical regions ensures redundancy so you can be sure of being able to access your data in the event that a local disaster occurs. Windows Azure Storage can not only scale to meet whatever needs your business might have, it's also very cost-effective since you only pay for what you use.

Data management

Windows Azure data services can provide you with a consistent experience whether you're working with relational or non-relational data and currently supports SQL databases up to 150 GB in size. You can utilize your existing data management skills, such as relational database design and Transact-SQL, and can mix and match data across a variety of different data services to create just the solution your business needs.

Aneka: Cloud Application Platform -Integration of Private and Public Clouds

- Aneka is Manjrasoft Pty. Ltd.'s solution for developing, deploying, and managing cloud applications.
- Aneka is an Application Platform-as-a-Service (Aneka PaaS) for Cloud Computing. It acts as a framework for building customized applications and deploying them on either public or private Clouds. One of the key features of Aneka is its support for provisioning resources on different public Cloud providers such as Amazon EC2, Windows Azure and GoGrid.
- Aneka is a market oriented Cloud development and management platform with rapid application development and workload distribution capabilities.

- Aneka is an integrated middleware package which allows you to seamlessly build and manage an interconnected network in addition to accelerating development, deployment and management of distributed applications using Microsoft .NET frameworks on these networks.
- It is market oriented since it allows you to build, schedule, provision and monitor results using pricing, accounting, QoS/SLA services in private and/or public (leased) network environments.
- One of Aneka's key advantages is its extensible set of application programming interfaces (APIs) associated with different types of programming models—such as Task, Thread, and MapReduce—used for developing distributed applications, integrating new capabilities into the cloud, and supporting different types of cloud deployment models: public, private, and hybrid.
- These features differentiate Aneka from infrastructure management software and characterize it as a platform for developing, deploying, and managing execution of applications on various types of clouds.

Framework

Aneka is a software platform for developing cloud computing applications. It allows harnessing of disparate computing resources and managing them into a unique virtual domain the Aneka Cloud in which applications are executed. According to the Cloud Computing Reference Model

Aneka is a pure PaaS solution for cloud computing. Aneka is a cloud middleware product that can be deployed on a heterogeneous set of resources: a network of computers, a multi-core server, datacenters, virtual cloud infrastructures, or a mixture of these.

The framework provides both middleware for managing and scaling distributed applications and an extensible set of APIs for developing them.

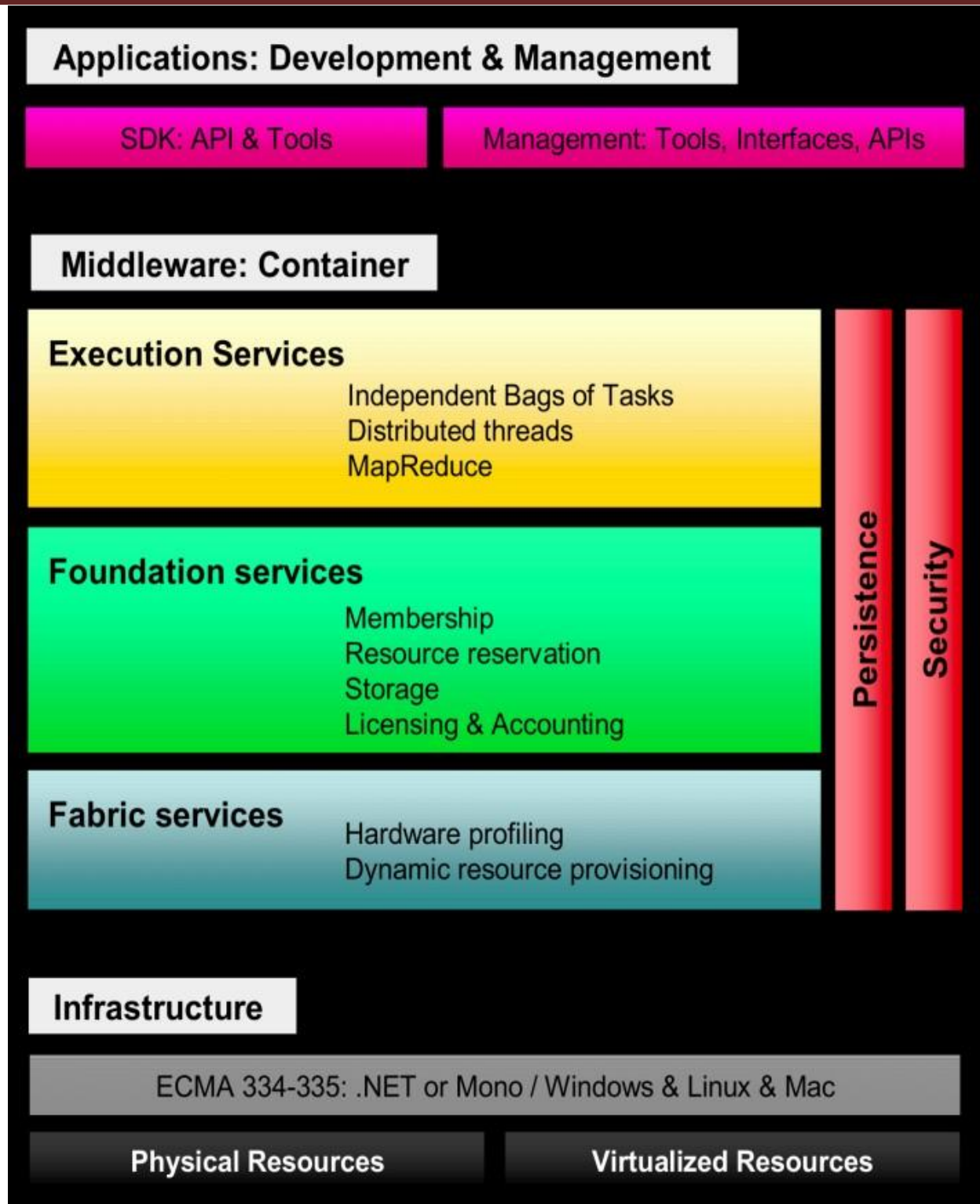


Figure provides a complete overview of the components of the Aneka framework. The core Infrastructure of the system provides a uniform layer that allows the framework to be deployed over different platforms and operating systems. The physical and virtual resources representing the bare metal of the cloud are managed by the Aneka container, which is installed on each node

A collection of interconnected containers constitute the Aneka Cloud: a single domain in which services are made available to users and developers. The container features three different classes of services: Fabric Services, Foundation Services, and Execution Services.

Aneka implements a service-oriented architecture (SOA), and services are the fundamental components of an Aneka Cloud. Services operate at container level and, except for the platform abstraction layer; they provide developers, users, and administrators with all features offered by the framework.

1. Build

Aneka includes a Software Development Kit (SDK) which includes a combination of APIs and Tools to enable you to express your application. Aneka also allows you to build different run- time environments and build new applications. Aneka provides APIs and tools that enable applications to be virtualized over a heterogeneous network.

Supported APIs include:

- Task Model for batch and legacy applications
- Thread Model for applications that use object oriented thread
- MapReduce Model for data intensive applications like data mining or analytics.
- Others such as MPI (Message Passing) and Actors (Distributive Active Objects/Agents) can be customized

Supported Tools include:

- Design Explorer for Parameter Sweep applications. Built on-top of task model with no additional requirements for programming.
- Work Flow applications. Built on-top of task model with some additional requirements for programming.

Build different types of Run-time environments:

- PC Grids (also called Enterprise Grids)
- Data Centres (Clusters)
- MultiCore Computers
- Public and/or private networks
- Virtual Machine or Physical

Use APIs and Tools to build new applications or enable existing applications over different Run-time environments. All the following services are available to specific interfaces and APIs on top of which the software development kit (SDK) and management kit are built. The SDK mainly relates to application development and modeling; it provides developers with APIs to develop applications with the existing programming models and an object model for creating new models.

These services involve:

- Elasticity and scaling. By means of the dynamic provisioning service, Aneka supports dynamically upsizing and downsizing of the infrastructure available for applications.
- Runtime management. The runtime machinery is responsible for keeping the infrastructure up and running and serves as a hosting environment for services. It is primarily represented by the container and a collection of services that manage service membership and lookup, infrastructure maintenance, and profiling.
- Resource management. Aneka is an elastic infrastructure in which resources are added and removed dynamically according to application needs and user requirements. To provide QoS-based execution, the system not only allows dynamic provisioning but also provides capabilities for reserving nodes for exclusive use by specific applications.
- Application management. A specific subset of services is devoted to managing applications. These services include scheduling, execution, monitoring, and storage management.
- User management. Aneka is a multitenant distributed environment in which multiple applications, potentially belonging to different users, are executed. The framework provides an extensible user system via which it is possible to define users, groups, and permissions. The services devoted to user management build up the security infrastructure of the system

and constitute a fundamental element for accounting management.

- QoS/SLA management and billing. Within a cloud environment, application execution is metered and billed. Aneka provides a collection of services that coordinate together to take into account the usage of resources by each application and to bill the owning user accordingly.

2. Manage

Aneka Management includes a Graphical User Interface (GUI) and APIs to set-up, monitor, manage and maintain remote and global Aneka compute clouds. Aneka also has an accounting mechanism and manages priorities and scalability based on SLA/QoS which enables dynamic provisioning.

Briefly, the set of operations that are performed through the Management Studio are the following:

- Quick setup of computing clouds;
- Remote installation and configuration of nodes;
- System load monitoring and tuning.
- Monitor aggregate dynamic statistics and probing individual nodes for CPU and memory load
- Extensible framework – add new features and services by implementing management plug-ins

Other management features include:

- Accounting and Pricing services provide flexible pricing strategies and keeping track of applications, reservations and users.
- Capacity Management – Dynamically provisioning to elastically scale up and down according to application requirements.
- Service Oriented allowing discovery of services and available nodes
- Automatic overflow and Failover giving high availability based on SLA

The management kit gives a complete view of Aneka Clouds and allows monitoring Aneka's status, whereas the SDK is more focused on the single application and provides means to control its execution from a single user.

3. Aneka container (ACCELERATE)

The Aneka container constitutes the building blocks of Aneka Clouds and represents the runtime machinery available to services and applications.

The container, the unit of deployment in Aneka Clouds, is a lightweight software layer designed to host services and interact with the underlying operating system and hardware. The services installed in the Aneka container can be classified into three major categories:

- Fabric Services
- Foundation Services
- Application Services

The services stack resides on top of the Platform Abstraction Layer (PAL), representing the interface to the underlying operating system and hardware. It provides a uniform view of the software and hardware environment in which the container is running.

Infrastructure

The core infrastructure of the system is based on the .NET technology and allows the Aneka container to be portable over different platforms and operating systems. The Common Language Infrastructure (CLI), which is the specification introduced in the standard, defines a common runtime environment and application model for executing programs but does not provide any interface to access the hardware or to collect performance data from the hosting operating system.

Moreover, each operating system has a different file system organization and stores that information differently. The Platform Abstraction Layer (PAL) addresses this heterogeneity and provides the container with a uniform interface for accessing the relevant hardware and operating system information, thus allowing the rest of the container to run unmodified on any

supported platform. The PAL provides the following features:

- Uniform and platform-independent implementation interface for accessing the hosting platform
- Uniform access to extended and additional properties of the hosting platform
- Uniform and platform-independent access to remote nodes
- Uniform and platform-independent management interfaces

The PAL is a small layer of software that comprises a detection engine, which automatically

configures the container at boot time, with the platform-specific component to access the above information and an implementation of the abstraction layer for the Windows, Linux, and Mac OS X operating systems.

The collectible data that are exposed by the PAL are the following:

- Number of cores, frequency, and CPU usage
- Memory size and usage
- Aggregate available disk space
- Network addresses and devices attached to the node

Fabric services

Fabric Services define the lowest level of the software stack representing the Aneka Container. They provide access to the resource-provisioning subsystem and to the monitoring facilities implemented in Aneka. Resource-provisioning services are in charge of dynamically providing new nodes on demand by relying on virtualization technologies, while monitoring services allow for hardware profiling and implement a basic monitoring infrastructure that can be used by all the services installed in the container.

5.2.2.1 Profiling and monitoring

Profiling and monitoring services are mostly exposed through the Heartbeat, Monitoring, and Reporting Services. The Heartbeat Service periodically collects the dynamic performance information about the node and publishes this information to the membership service in the Aneka Cloud. These data are collected by the index node of the Cloud, which makes them available for services such as reservations and scheduling in order to optimize the use of a heterogeneous infrastructure.

The set of built-in services for monitoring and profiling is completed by a generic monitoring Infrastructure, which allows any custom service to report its activity. This infrastructure is composed of the Reporting and Monitoring Services. Currently several built-in services provide information through this channel:

- The Membership Catalogue tracks the performance information of nodes.
- The Execution Service monitors several time intervals for the execution of jobs.
- The Scheduling Service tracks the state transitions of jobs.

- The Storage Service monitors and makes available information about data transfer, such as upload and download times, filenames, and sizes.
- The Resource Provisioning Service tracks the provisioning and lifetime information of virtual nodes.

Resource management

Resource management is another fundamental feature of Aneka Clouds. It comprises several tasks:

- resource membership,
- resource reservation, and
- resource provisioning.

Aneka provides a collection of services that are in charge of managing resources. These are the Index Service (or Membership Catalogue), Reservation Service, and Resource Provisioning Service.

The resource-provisioning infrastructure built into Aneka is mainly concentrated in the Resource Provisioning Service, which includes all the operations that are needed for provisioning virtual instances.

The implementation of the service is based on the idea of resource pools. A resource pool abstracts the interaction with a specific IaaS provider by exposing a common interface so that all the pools can be managed uniformly. The extend of the resource-provisioning scenarios can be handled by Aneka, which can also be used as a virtual machine manager.

Foundation services

Fabric Services are fundamental services of the Aneka Cloud and define the basic infrastructure management features of the system. Foundation Services are related to the logical management of the distributed system built on top of the infrastructure and provide supporting services for the execution of distributed applications.

All the supported programming models can integrate with and leverage these services to provide advanced and comprehensive application management. These services cover:

- Storage management for applications
- Accounting, billing, and resource pricing
- Resource reservation

Foundation Services provide a uniform approach to managing distributed applications and allow developers to concentrate only on the logic that distinguishes a specific programming model from the

others.

Together with the Fabric Services, Foundation Services constitute the core of the Aneka middleware.

Application services

Application Services manage the execution of applications and constitute a layer that differentiates according to the specific programming model used for developing distributed applications on top of Aneka.

The types and the number of services that compose this layer for each of the programming models may vary according to the specific needs or features of the selected model.

It is possible to identify two major types of activities that are common across all the supported models:

- scheduling and
- execution.

Aneka defines a reference model for implementing the runtime support for programming models that abstracts these two activities in corresponding services:

- the Scheduling Service and
- the Execution Service.

Moreover, it also defines base implementations that can be extended in order to integrate new models. Scheduling Services are in charge of planning the execution of distributed applications on top of Aneka and governing the allocation of jobs composing an application to nodes. They also constitute the integration point with several other Foundation and Fabric Services, such as the Resource Provisioning Service, the Reservation Service, the Accounting Service, and the **Reporting Service**.

Common tasks that are performed by the scheduling component are the following:

- Job to node mapping
- Rescheduling of failed jobs
- Job status monitoring
- Application status monitoring

Building Aneka clouds

Aneka is primarily a platform for developing distributed applications for clouds. As a software platform it requires infrastructure on which to be deployed; this infrastructure needs to be managed.

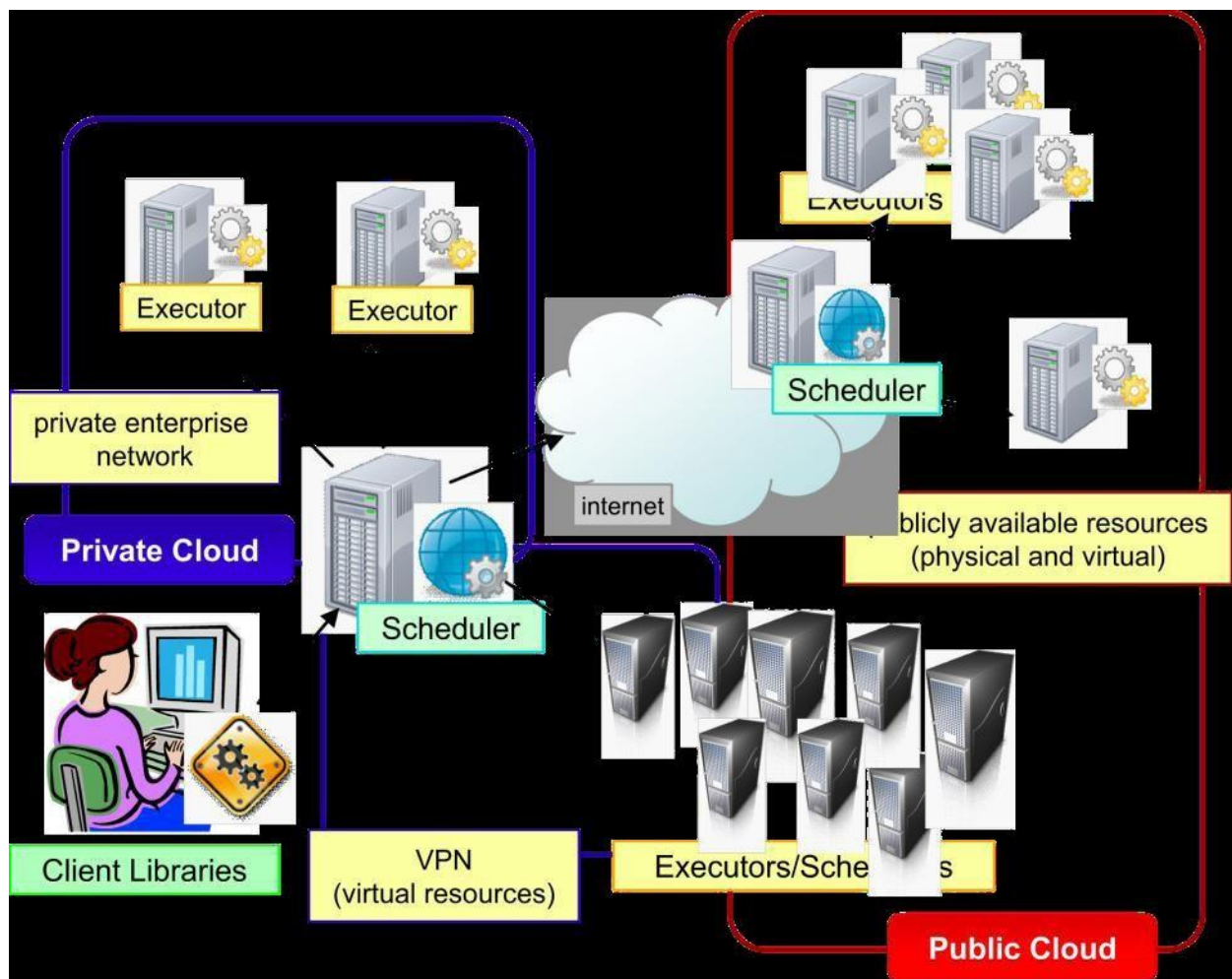
Infrastructure management tools are specifically designed for this task, and building clouds is one of the primary tasks of administrators. Aneka supports various deployment models for public, private, and hybrid clouds.

Infrastructure organization

Figure provides an overview of Aneka Clouds from an infrastructure point of view. The scenario is a reference model for all the different deployments Aneka supports.

A central role is played by the Administrative Console, which performs all the required management operations.

A fundamental element for Aneka Cloud deployment is constituted by repositories. A repository provides storage for all the libraries required to lay out and install the basic Aneka platform.



The libraries constitute the software image for the node manager and the container programs. Repositories can make libraries available through a variety of communication channels, such as HTTP, FTP, common file sharing, and so on.

The Management Console can manage multiple repositories and select the one that best suits the specific deployment.

Logical organization

The logical organization of Aneka Clouds can be very diverse, since it strongly depends on the configuration selected for each of the container instances belonging to the Cloud. The most common scenario is to use a master-worker configuration with separate nodes for storage, as shown in Figure The master node features all the services that are most likely to be present in one single copy and that provide the intelligence of the Aneka Cloud.

A common configuration of the master node is as follows: Index Service (master copy)

- Heartbeat Service
- Logging Service
- Reservation Service
- Resource Provisioning Service
- Accounting Service
- Reporting and Monitoring Service
- Scheduling Services for the supported programming models

The master node also provides connection to an RDBMS facility where the state of several services is maintained.

Deployment modes

1. Private cloud deployment mode

- A private deployment mode is mostly constituted by local physical resources and infrastructure management software providing access to a local pool of nodes, which might be virtualized.
- In this scenario Aneka Clouds are created by harnessing a heterogeneous pool of resources such as desktop machines, clusters, or workstations.
- These resources can be partitioned into different groups, and Aneka can be configured to leverage these resources according to application needs. Moreover, leveraging the Resource Provisioning Service, it is possible to integrate virtual nodes provisioned from a local resource

pool managed by systems such as XenServer, Eucalyptus, and OpenStack.

- Figure shows a common deployment for a private Aneka Cloud. This deployment is acceptable for a scenario in which the workload of the system is predictable and a local virtual machine manager can easily address excess capacity demand. Most of the Aneka nodes are constituted of physical nodes with a long lifetime and a static configuration and generally do not need to be reconfigured often.
- The different nature of the machines harnessed in a private environment allows for specific policies on resource management and usage that can be accomplished by means of the Reservation Service.
- For example, desktop machines that are used during the day for office automation can be exploited outside the standard working hours to execute distributed applications. Workstation clusters might have some specific legacy software that is required for supporting the execution of applications and should be preferred for the execution of applications with special requirements.

2. Public cloud deployment mode

Public Cloud deployment mode features the installation of Aneka master and worker nodes over a completely virtualized infrastructure that is hosted on the infrastructure of one or more resource providers such as Amazon EC2 or GoGrid.

In this case it is possible to have a static deployment where the nodes are provisioned beforehand and used as real machines. This deployment merely replicates a classic Aneka installation on a physical infrastructure without any dynamic provisioning capability.

More interesting is the use of the elastic features of IaaS providers and the creation of a Cloud that is completely dynamic. Figure provides an overview of this scenario.

In this scenario it is possible to deploy an Aneka Cloud composed of only one node and to completely leverage dynamic provisioning to elastically scale the infrastructure on demand.

- A fundamental role is played by the Resource Provisioning Service, which can be configured with different images and templates to instantiate.
- Other important services that have to be included in the master node are the Accounting and Reporting Services.

These provide details about resource utilization by users and applications and are fundamental in a multitenant Cloud where users are billed according to their consumption of Cloud capabilities.

3 Hybrid cloud deployment mode

The hybrid deployment model constitutes the most common deployment of Aneka. In many cases, there is an existing computing infrastructure that can be leveraged to address the computing needs of applications.

This infrastructure will constitute the static deployment of Aneka that can be elastically scaled on demand when additional resources are required. An overview of this deployment is presented in Figure.

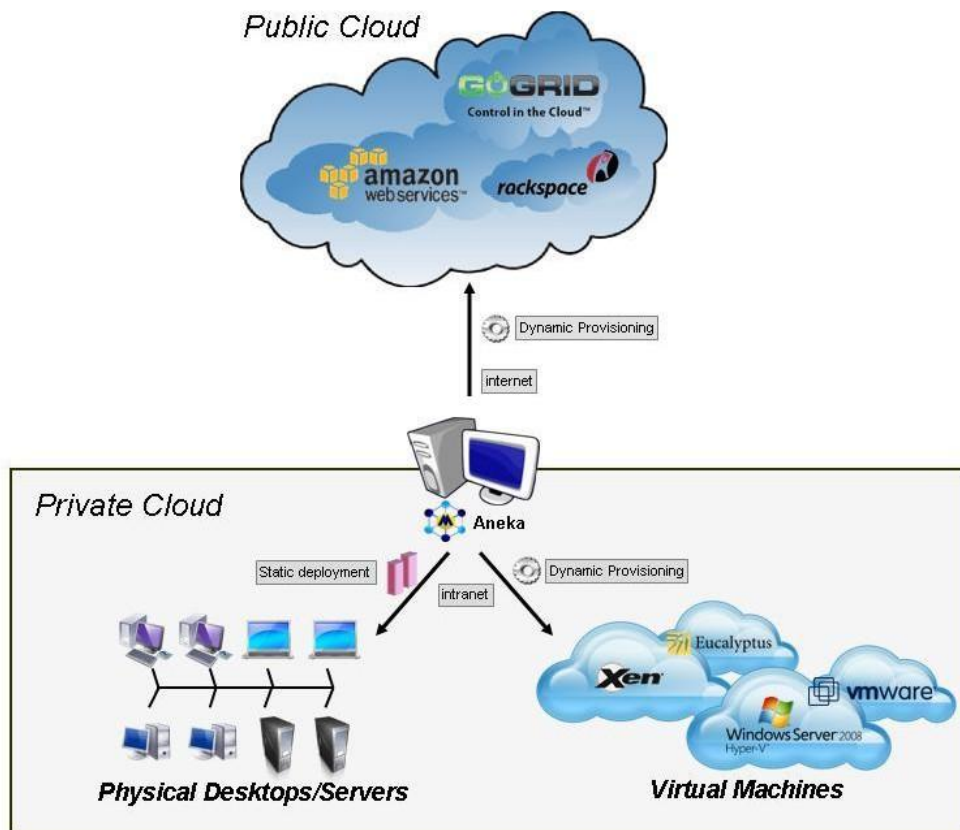
This scenario constitutes the most complete deployment for Aneka that is able to leverage all the capabilities of the framework:

- Dynamic Resource Provisioning
- Resource Reservation
- Workload Partitioning
- Accounting, Monitoring, and Reporting

Moreover, if the local premises offer some virtual machine management capabilities, it is possible to provide a very efficient use of resources, thus minimizing the expenditure for application execution.

In a hybrid scenario, heterogeneous resources can be used for different purposes. As we know, in the case of a private cloud deployment, desktop machines can be reserved for low priority workload outside the common working hours.

The majority of the applications will be executed on workstations and clusters, which are the nodes that are constantly connected to the Aneka Cloud. Any additional computing capability demand can be primarily addressed by the local virtualization facilities, and if more computing power is required, it is possible to leverage external IaaS providers.



Different from the Aneka Public Cloud deployment is the case in which it makes more sense to leverage a variety of resource providers to provision virtual resources. Since part of the infrastructure is local, a cost in data transfer to the external IaaS infrastructure cannot be avoided. It is then important to select the most suitable option to address application needs. The Resource Provisioning Service implemented in Aneka exposes the capability of leveraging several resource pools at the same time and configuring specific policies to select the most appropriate pool for satisfying a provisioning request. These features simplify the development of custom policies that can better serve the needs of a specific hybrid deployment.

Why Aneka?

Every CIO faces the need to optimize his budget to support the adoption of Cloud technology. Despite the economic downturn and the challenging business climate, considerable investments are allocated for infrastructure and software technology development. Cloud computing provides significant revenues with a minimum upfront commitment and the ability to comfortably control their expenditure according to the health status of the managed business. Choosing the right

option in such a vibrant market is a challenging task. Aneka Platform as a Service provides a comprehensive and unmatched solution for Cloud adoption by bringing several advantages to enterprises:

Reduced Costs. Any successful enterprise business knows the significance of appraising, managing, and optimizing the Capital Expenditure and Operational Expenditure and necessities to achieve cost economies. Aneka adeptly leverages your current infrastructure assets and Cloud management tools with a low cost guarantee.

Improved Reliability. As one of its kind, Manjrasoft's Aneka is the most comprehensive and mature path for Cloud adoption based on the .NET technology. Aneka's ability to design and create a solid fault tolerant system infrastructure without having to build topology level knowledge base into applications simplifies the application development and automatically manages application load over Clouds, Grids, clusters, or desktops. This gives a newfound level of resiliency with a guaranteed quality of service and effective metering and monitoring for all the service.

Simplicity. Aneka offers the most flexible and robust APIs framework that cleanly handles .NET based enterprise application management and development with lightweight technology and flexible application integration approaches.

Faster Time to Value. Aneka leverages a solid and widely adopted technology such as the Microsoft .NET framework; thus, enterprise customers can capitalize their existing .NET base application and engineering expertise over Cloud environment. With regards to the technical development team, Aneka's support of multiple programming and application environments helps them to leverage their current expertise over software development without having to sunk huge effort on capabilities to adopt new technology and learning new processes.

Seamless Scalability. Aneka helps enterprise customers to enrich their applications and services with support for distributed and scalable runtime environments for multicore desktops, servers and a network of computing systems that are presented as Clusters, Grid, and Clouds. This process is completely transparent to applications and relies on dynamic provisioning multiple virtual and/or physical machines for accelerating applications in a scalable manner from a single multi-core desktop computer to a large-scale elastic Cloud computing infrastructure such as Amazon EC2.