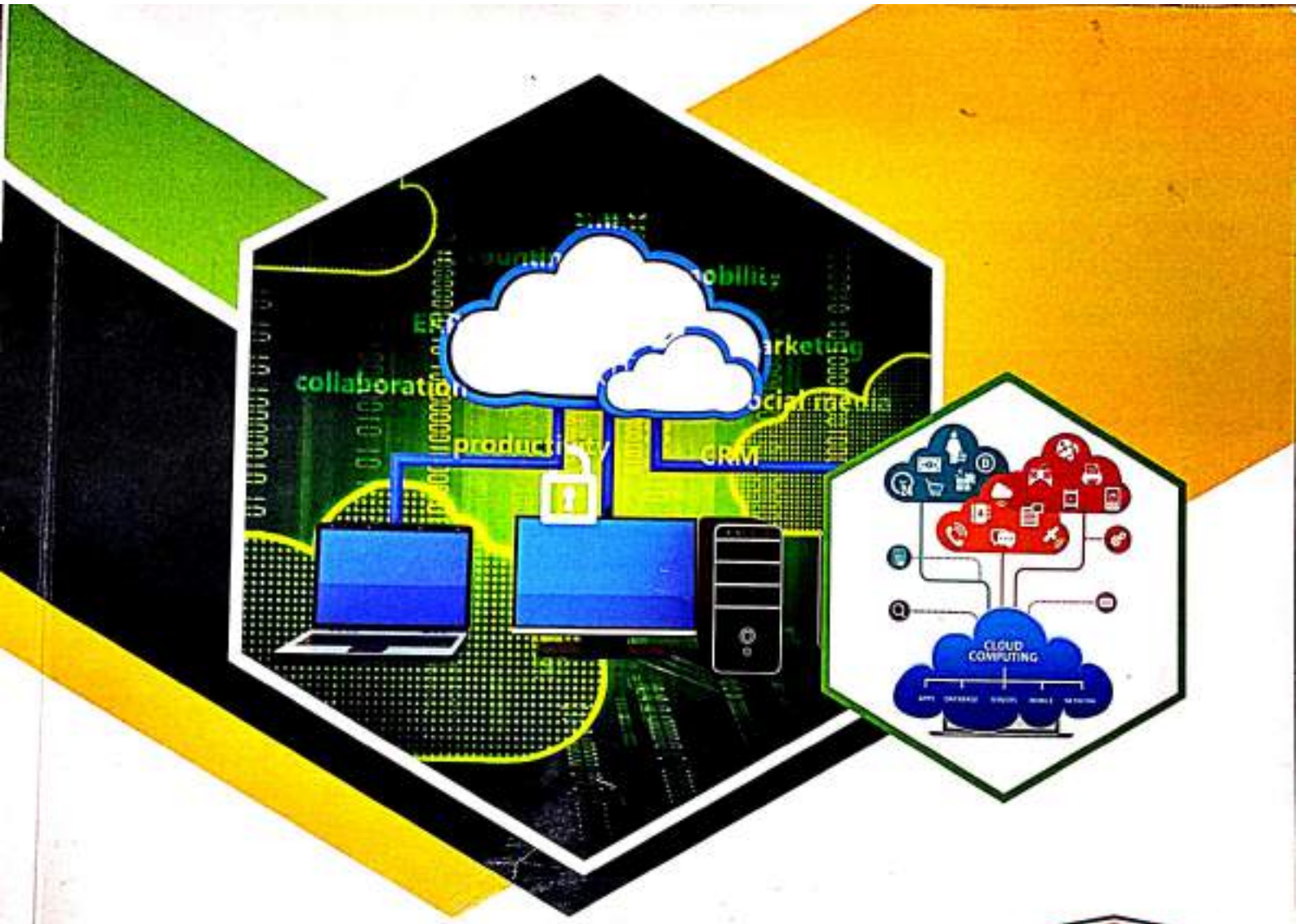


As Per New Revised Credit System Syllabus

DBATU

Final Year B.TECH. Semester - VII
Course in COMPUTER ENGINEERING

CLOUD COMPUTING

(PROGRAM ELECTIVE – IX 703(A))

NITIN N. SAKHARE

www.pragationline.com
niralipune@pragationline.com
www.facebook.com/niralibooks
@nirali.prakashan

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

...T BOOK OF

CLOUD COMPUTING

(PROGRAM ELECTIVE – IX 703(A))

FOR
SEMESTER – VII
FINAL YEAR B. TECH COURSE IN
COMPUTER ENGINEERING

Strictly According to New Revised Credit System Syllabus
of Dr. Babasaheb Ambedkar Technological University (DBATU),
Lonere, (Dist. Raigad) Maharashtra,
(w.e.f. June 2020-21)

NITIN N. SAKHARE
M. E. (Comp. Networks)
Assistant Professor,
Computer Engineering Department
Vishwakarma Institute of Information Technology
Kondhwa (Bk.), PUNE.

Price ₹ 210.00

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

Unit I : Introduction to Cloud

[6 Hrs]

Cloud Computing at a Glance, the Vision of Cloud Computing, Defining a Cloud, A Closer Look, Cloud Computing Reference Model, Characteristics and Benefits, Challenges Ahead, Historical Developments. **Virtualization:** Introduction, Characteristics of Virtualized Environment, Taxonomy of Virtualization Techniques, Virtualization and Cloud computing, Pros and Cons of Virtualization, Technology Examples- VMware and Microsoft Hyper-V. **Before the Move into the Cloud:** Know Your Software Licenses, The Shift to a Cloud Cost Model, Service Levels for Cloud Applications.

Unit II : Cloud Computing Architecture

[6 Hrs]

Introduction, Cloud Reference Model, Architecture, Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Interoperability and Standards, Scalability and Fault Tolerance.

Ready for the Cloud: Web Application Design, Machine Image Design, Privacy Design, Database Management, Data Security, Network Security, Host Security, Compromise Response.

Unit III : Defining the Clouds for Enterprise

[6 Hrs]

Storage as a service, Database as a service, Process as a service, Information as a service, Integration as a service and Testing as a service; Scaling a cloud infrastructure - Capacity Planning, Cloud Scale. **Disaster Recovery:** Disaster Recovery Planning, Disasters in the Cloud, Disaster Management.

Unit IV : Aneka: Cloud Application Platform

[6 Hrs]

Framework Overview, Anatomy of the Aneka Container, From the Ground Up: Platform Abstraction Layer, Fabric Services, Foundation Services, Application Services, Building Aneka Clouds, Infrastructure Organization, Logical Organization, Private Cloud Deployment Mode, Public Cloud Deployment Mode, Hybrid Cloud Deployment Mode, Cloud Programming and Management, Aneka SDK, Management Tools.

Unit V : Cloud Applications

[6 Hrs]

Scientific Applications – Health care, Geo-science and Biology; Business and Consumer Applications- CRM and ERP, Social Networking, Media Applications and Multiplayer Online Gaming. **Cloud Platforms in Industry:** Amazon Web Services- Compute Services, Storage Services, Communication Services and Additional Services. Google AppEngine-Architecture and Core Concepts, Application Life-Cycle, cost model. Microsoft Azure- Azure Core Concepts, SQL Azure.

CONTENTS

PREFACE TO THE SECOND EDITION

I am glad and excited to announce that the First Edition of this book received an overwhelming response from the engineering student community, compelling us to release its **Second Edition** within a very short period of time.

This thoroughly revised **Second Edition** has been updated with additional matter, many solved problems, including solutions to all University Examination Problems and Numerous Exercises for practice.

Special care has been taken to maintain high degree of accuracy in the theory and numericals throughout the book.

I take this opportunity to express my sincere thanks to Dineshbhai Furia of Nirali Prakashan, a reputed pioneer in the publication field. My special thanks to Jignesh Furia for their effective cooperation and great care in bringing out this revised edition.

I sincerely hope that this "**Second Edition**" will also be warmly received by all concerned as in the past.

Valuable suggestions from my esteemed readers to improve the book are most welcome and highly appreciated.

Author

Pune

UNIT I

INTRODUCTION TO CLOUD

1.1 CLOUD COMPUTING AT A GLANCE

The cloud computing can be defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models (IaaS, PaaS & SaaS), and four deployment models (public, private, hybrid and community cloud).

- Cloud computing is a specialized form of distributed computing that introduces utilization models for remotely provisioning scalable and measured resources. The cloud services are delivered by using internet technology on pay per use basis.
- Let's have a look on some of the motivations that has influenced the industry to adapt cloud computing.

Capacity Planning:

- Capacity planning is the process of determining and fulfilling future demands of an organization's IT resources, products, and services. Within this context, *capacity* represents the maximum amount of work that an IT resource is capable of delivering in a given period of time. Capacity planning is focused on minimizing this discrepancy to achieve predictable efficiency and performance.

Cost Reduction:

- A direct alignment between IT costs and business performance can be difficult to maintain. The growth of IT environments often corresponds to the assessment of their maximum usage requirements. This can make the support of new and expanded business automations an ever-increasing investment.
- Two costs need to be accounted for: the cost of acquiring new infrastructure, and the cost of its ongoing ownership.

Organizational Agility:

- Businesses need the ability to adapt and evolve to successfully face change caused by both internal and external factors. Organizational agility is the measure of an organization's responsiveness to change.
- An IT enterprise often needs to respond to business change by scaling its IT resources beyond the scope of what was previously predicted or planned for. For example, infrastructure may be subject to limitations that prevent the organization from responding to usage fluctuations even when anticipated if previous capacity planning efforts were restricted by inadequate budgets. In other cases, changing business needs and priorities may require IT resources to be more available and reliable than before.
- Following are the set of basic terms that represent the fundamental concepts and aspects pertaining to the notion of a cloud and its most primitive artifacts.
- It is important to distinguish the term "cloud" and the cloud symbol from the Internet. As a specific environment used to remotely provision IT resources, a cloud has a finite boundary. There are many individual clouds that are accessible via the Internet. Whereas the Internet provides open access to many Web-based IT resources, a cloud is typically privately owned and offers access to IT resources that is metered. Much of the Internet is dedicated to the access of content-based IT resources published via the World Wide Web.

1.1.1 IT Resource

- An *IT resource* is a physical or virtual IT-related artifact that can be either software based, such as a virtual server or a custom software program, or hardware-based, such as a physical server or a network device (Fig. 1.1).

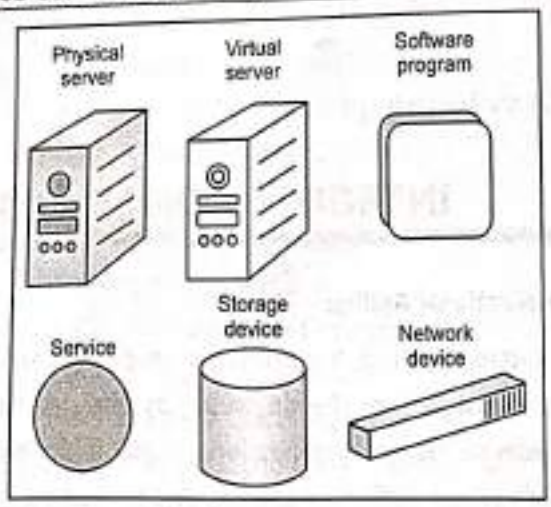


Fig. 1.1: Examples of common IT resources and their corresponding symbols

- Fig. 1.2 illustrates how the cloud symbol can be used to define a boundary for a cloud-based environment that hosts and provisions a set of IT resources. The displayed IT resources are consequently considered to be cloud-based IT resources.

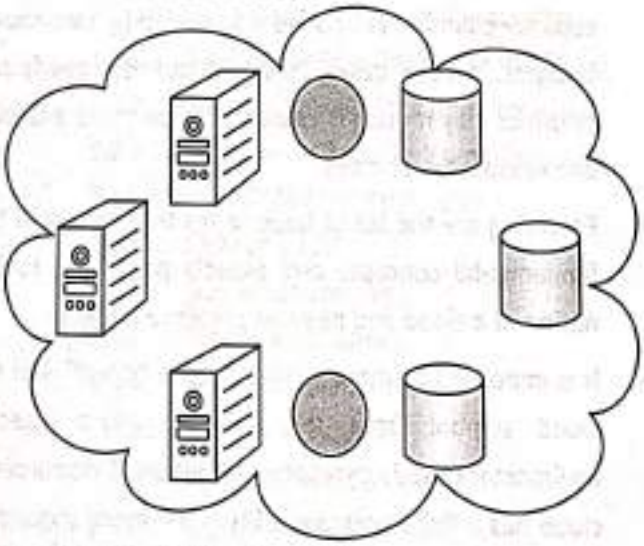


Fig. 1.2: A cloud is hosting eight IT resources: three virtual servers, two cloud services, and three storage devices.

1.1.2 On-Premise

- As a distinct and remotely accessible environment, a cloud represents an option for the deployment of IT resources. An IT resource that is hosted in a conventional IT enterprise within an organizational boundary (that does not specifically represent a cloud) is considered to be located on the premises of the IT enterprise, or *on-premise* for short. In other words, the term "on-premise" is another way of stating "on the premises of a controlled IT environment that is not cloud-based." This term is used to qualify an IT resource as an alternative to "cloud-based." An IT resource that is on-premise cannot be cloud-based and vice-versa.

1.1.3 Cloud Consumers and Cloud Providers

- The party that provides cloud-based IT resources is the *cloud provider*. The party that uses cloud-based IT resources is the *cloud consumer*. These terms represent roles usually assumed by organizations in relation to clouds and corresponding cloud provisioning contracts.

1.1.4 Scaling

- Scaling, from an IT resource perspective, represents the ability of the IT resource to handle increased or decreased usage demands.

  The following are types of scaling:

  1. **Horizontal Scaling :** scaling out and scaling in
  2. **Vertical Scaling :** scaling up and scaling down

1. Horizontal Scaling :

- The allocating or releasing of IT resources that are of the same type is referred to as *horizontal scaling* (Fig. 1.3). The horizontal allocation of resources is referred to as *scaling out* and the horizontal releasing of resources is referred to as *scaling in*. Horizontal scaling is a common form of scaling within cloud environments.

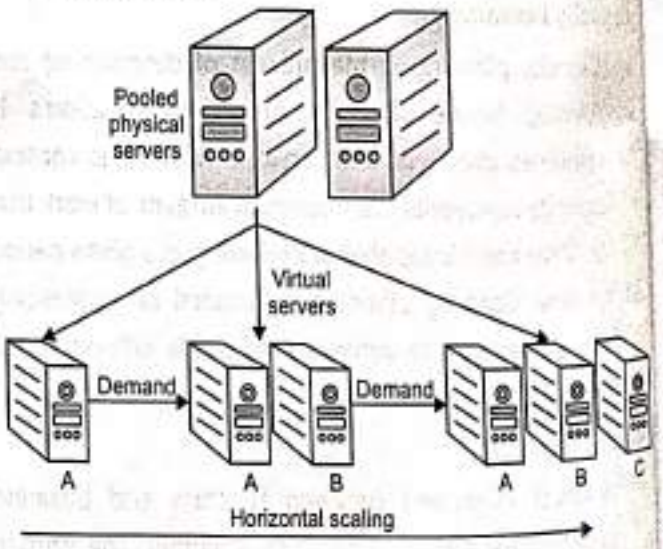


Fig. 1.3: Horizontal Scaling

- An IT resource (Virtual Server A) is scaled out by adding more of the same IT resources (Virtual Servers B and C).

2. Vertical Scaling :

- When an existing IT resource is replaced by another with higher or lower capacity, *vertical scaling* is considered to have occurred (Fig. 1.4). Specifically, the

replacing of an IT resource with another that has a higher capacity is referred to as *scaling up* and the replacing an IT resource with another that has a lower capacity is considered *scaling down*. Vertical scaling is less common in cloud environments due to the downtime required while the replacement is taking place.

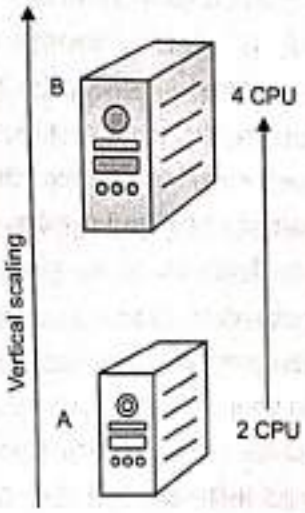


Fig. 1.4 : Vertical Scaling

- An IT resource (a virtual server with two CPUs) is scaled up by replacing it with a more powerful IT resource with increased capacity for data storage (a physical server with four CPUs).
- Table 1.1 provides a brief overview of common pros and cons associated with horizontal and vertical scaling.

Table 1.1: A Comparison of Horizontal and Vertical Scaling

Horizontal Scaling	Vertical Scaling
less expensive (through commodity hardware components)	more expensive (specialized servers)
IT resources instantly available	IT resources normally instantly available
resource replication and automated scaling	additional setup is normally needed
additional IT resources needed	no additional IT resources needed
not limited by hardware capacity	limited by maximum hardware capacity

1.2 THE VISION OF CLOUD COMPUTING

- Cloud computing is not a one-size-fits-all affair. Just as the hardware and software configuration you use in your organization is different from that of the company down the street, your cloud computing needs will be different as well.
- In this topic you will understand how your organization can best use cloud computing, and which solutions might be most appropriate for your needs. And while we talk about what cloud computing is good for, we also talk about cloud computing limitations.
- That is, cloud computing is not perfect, and there are times when you shouldn't turn to it.
- Whether or not you should use cloud computing depends on a number of factors, including
  - ➢ Cost/benefit ratio
  - ➢ Speed of delivery
  - ➢ How much capacity you will use
  - ➢ Whether your data is regulated
  - ➢ Your organization's corporate and IT structure
- There may be times when the need you have is a perfect match for cloud computing.
- But there may also be times when cloud computing is simply not a good match for your needs. In this section we'll take a look at both what you can use clouds for, and when you should steer clear of them.

Scenarios

- There are three different major implementations of cloud computing. How organizations are using cloud computing is quite different at a granular level, but the uses generally fall into one of these three solutions.

Compute Clouds

- Compute clouds allow access to highly scalable, inexpensive, on-demand computing resources that run the code that they're given. Three examples of compute clouds are
  1. Amazon's EC2
  2. Google App Engine
  3. Berkeley Open Infrastructure for Network Computing (BOINC)
- Compute clouds are the most flexible in their offerings and can be used for sundry purposes; it simply depends on the application the user wants to access.

- You could close this book right now, sign up for a cloud computing account, and get started right away. These applications are good for any size organization, but large organizations might be at a disadvantage because these applications don't offer the standard management, monitoring, and governance capabilities that these organizations are used to.

Enterprises aren't shut out, however.

- Amazon offers enterprise-class support and there are emerging sets of cloud offerings like
- Terremark's Enterprise Cloud, which are meant for enterprise use.

Cloud Storage

- One of the first cloud offerings was cloud storage and it remains a popular solution. Cloud storage is a big world. There are already in excess of 100 vendors offering cloud storage.
- This is an ideal solution if you want to maintain files off-site.
- Security and cost are the top issues in this field and vary greatly, depending on the vendor you choose. Currently, Amazon's S3 is the top dog.

Cloud Applications

- Cloud applications differ from compute clouds in that they utilize software applications that rely on cloud infrastructure. Cloud applications are versions of Software as a Service (SaaS) and include such things as web applications that are delivered to users via a browser or application like Microsoft Online Services. These applications offload hosting and IT management to the cloud.
- Cloud applications often eliminate the need to install and run the application on the customer's own computer, thus alleviating the burden of software maintenance, ongoing operation, and support.

Some Cloud Applications Include

- Peer-to-peer computing (like BitTorrent and Skype)
- Web applications (like MySpace or YouTube)
- SaaS (like Google Apps)
- Software plus services (like Microsoft Online Services)
- If you want to use cloud computing and post data covered by Health Insurance Portability and Accounting Act (HIPAA) on it, you are out of luck. Well, let's rephrase that if you want to put HIPAA data on a cloud, you shouldn't. That's sensitive healthcare information and the fact that HIPAA data could commingle on a server with another organization's data will likely get the attention of an observant HIPAA auditor.
- Even so, Google and Microsoft are both moving forward on health records services:
  - ➢ Microsoft is working on its Health Vault and Google Health promises to be a huge outpouring of private health data online. While the intent seems well-meaning to give consumers access to their healthcare data all it takes is one tiny breach to let sensitive data loose.
  - ➢ If you have data that is regulated like HIPAA or Sarbanes-Oxley you are well advised to be very careful in your plans to place data on a cloud. After all, if you have posted a customer's financial data and there's a breach, will they go after the cloud provider, or you?

Legislative Issues

- An issue of more concern for the sensitivity of private data is that there are laws and policy that allow the government freer access to data on a cloud than on a private server.
- For example, the Stored Communications Act allows the FBI access to data without getting a warrant or the owner's consent.

Geopolitical Concerns

- It may simply be illegal to post your information on a cloud. If you are in Canada (for instance) and you want to post your data on an American cloud, you're out of luck.
- The Canadian government has declared that government IT workers may not use network services that are operating within U.S. borders. The reason is that the Canadian data stored on those servers could be negatively impacted based on the Patriot Act.
- Sure, Canada might be the friendly neighbor of the United States to the north, but at this point in time, they have a great policy. All it would take is for the U.S. government to seize a server with foreign data on it, and before you can say "eh," we've got another international incident on our hands.

- And the same can be said of clouds operating outside the United States. You probably don't know the laws (if there are any) governing your privacy and protection in a foreign country. All it would take is the Generalissimo and his cadre of willing minions to roll into your provider's office and cart off the server with your data on it.

Hardware Dependencies

- If you have an application that requires specific hardware, chips, or drivers, a cloud solution might not be a good fit for you.
- First, if you have special hardware needs, the chances are lower that the service provider will have the precise hardware you need. That can significantly narrow your options when it comes to shopping around and finding a good deal.
- But let's say the planets are in perfect alignment, the provider you like has the hardware you need, and before long you are both humming away. This is all blissful now, but if the provider ever changes chipsets or other critical hardware, you might be out of luck.

Server Control

- If your application demands complete control over everything that is running, a cloud solution may not be right for you. If you need detailed control over the amount of memory, CPU, hard drive specs, or interfaces, then the cloud isn't an appropriate match for your application. After all, these are all things managed by the service provider.

Cost

- One of the big draws of cloud computing is cost. That is, it tends to be less expensive to run an application on a cloud than to invest in the infrastructure, buy the application outright, and then manage it day to day.
- However, over time, it may cost more to pay the cloud subscription than to have simply bought the servers yourself, so it is important to factor in everything from facilities, staff, software, and hardware.
- Cost and the way clouds operate are a moving target. Some have suggested that the cloud might bring servers into the client's datacenter. Another school of thought is a concept called *cloud bursting*. In this scenario, on-demand capacity can be provisioned to a cloud.

Lack of Need

- Anyone with a grandfather has heard the phrase, "if it ain't broke, don't fix it." And grandpa is right. If your current solution is getting the job done, why tinker with it?
- Now, there are certainly cases where cloud computing is advantageous for you. And in those cases, by all means use it. However, if you are just moving applications to the cloud for the fashion of it, take a look at some old pictures of "fashionable" people. You'll realize those polyester leisure suits and mullets may have been fashionable in their day, but not so much now.

Integration with Existing Applications

- If you mix oil and water, you get a lava lamp. Given the heading of this section, you already get what we're alluding to. The fact of the matter is that if you have two applications that need to integrate, it's best for one not to be located on-site and a second on the cloud.
- It creates problems with security, speed, and reliability. For instance, if you have two databases one with sensitive data housed locally, and one with non-sensitive data on a cloud the chances that the sensitive data will find its way to the cloud are very good.
- Also, if you are trying to run a high-speed application in-house and you rely on data from the cloud, the application will only run as fast as the cloud will allow. This also leads to questionable reliability. Will data be compromised or damaged from all the flying around it has to do?

Latency Concerns

- Since your data and application are located on a series of servers geographically disparate from your own site, it is going to take some time for the data to reach you. This isn't an issue of hours or days or even minutes. But if you require data instantaneously, the cloud might not be your best option.
- There's still travel time involved with your data. Now, it might be the case that a worker can request given data and it comes through in less than a second, and that speed is fine. However, if that same worker needs the data faster than a second, it might not be coming through fast enough.

Throughput Demands

- Since cloud computing is generally billed in a utility format, you pay for what you use.
- That's great and it seems fair, until you deploy applications that use a lot of throughput and costs start to rise. For instance, if you are streaming high-definition video over 100 sources, your costs are going to spike sharply.
- It's best to do the math on these sorts of things. Take into account what a server, power, and all other hardware will cost. Figure in the price of management and associated IT personnel costs and then compare that with what a service provider will charge you. If it's cheaper to buy the server, it might be best to forget about the cloud for now. But even if the cost is the same, you need to ask yourself what business you want to be in.

1.2.1 Benefits

- Your organization is going to have different needs from the company next door. However, cloud computing can help you with your IT needs. Let's take a closer look at what cloud computing has to offer your organization.

Scalability

- If you are anticipating a huge upswing in computing need (or even if you are surprised by a sudden demand), cloud computing can help you manage. Rather than having to buy, install, and configure new equipment, you can buy additional CPU cycles or storage from a third party.
- Since your costs are based on consumption, you likely wouldn't have to pay out as much as if you had to buy the equipment.

  Once you have fulfilled your need for additional equipment, you just stop using the cloud provider's services, and you don't have to deal with unneeded equipment. You simply add or subtract based on your organization's need.

Simplicity

- Again, not having to buy and configure new equipment allows you and your IT staff to get right to your business. The cloud solution makes it possible to get your application started immediately, and it costs a fraction of what it would cost to implement an on-site solution.

Knowledgeable Vendors

- Typically, when new technology becomes popular, there are plenty of vendors who pop up to offer their version of that technology. This isn't always good because a lot of those vendors tend to offer less than useful technology. By contrast, the first comers to the cloud computing party are actually very reputable companies.
- Companies like Amazon, Google, Microsoft, IBM, and Yahoo! have been good vendors because they have offered reliable service, plenty of capacity, and you get some brand familiarity with these well-known names.

1.2.2 Limitations

- There are other cases when cloud computing is not the best solution for your computing needs. This section looks at why certain applications are not the best to be deployed on the cloud. We don't mean to make these cases sound like deal-breakers, but you should be aware of some of the limitations. If you can work around them, that's great, but you should be aware of the issues before getting in too deep.

Your Sensitive Information

- We've talked about the concern of storing sensitive information on the cloud, but it can't be understated. Once data leaves your hands and lands in the lap of a service provider, you've lost a layer of control.

What's the Worry?

- Let's say a financial planner is using Google Spreadsheets to maintain a list of employee social security numbers. Now the financial planning company isn't the only one who should protect the data from hackers and internal data breaches. In a technical sense, it also becomes Google's problem. However, Google may absolve itself of responsibility in its agreement with you. So, it's no less complicated a task to sort out how sensitive information is genuinely secured. Also, the door is wide open for government investigators to subpoena that information. It has become much easier for the government to get information from third parties than from a privately owned server.
- Also, less scrupulous service providers might even share that data with a marketing firm. And other providers may, by way of their agreement with you, be allowed to access and catalog your information and

use it in ways you never intended. Again, be absolutely certain you understand fully your agreement with any service provider and that you approve and accept the terms of the agreement.

- What's important is that you realize what the provider's policies are governing the management and maintenance of your data. For example, Google's policy states that the company will share data with the government if it has a "good faith belief" that access is necessary to fulfill lawful requests.

Applications Not Ready

- In some cases the applications themselves are not ready to be used on the cloud. They may have little quirks that prevent them from being used to their fullest abilities, or they may not work whatsoever.
- First, the application might require a lot of bandwidth to communicate with users.
- Remember, since cloud computing is paid based on how much you use, it might turn out to be less expensive in the long run to simply house the application locally until it can be rewritten or otherwise modified to operate more efficiently.
- The application might also take a lot of effort to integrate with your other applications.
- If you try to relocate it to a cloud, you may find that the savings are erased by the additional effort required to maintain the integration. In this case it may end up being more cost effective to continue to host it locally.
- If the application has to talk with a database that you have onsite, it may be better to also have the application hosted locally until you can move the entire infrastructure to the cloud. Again, this helps you avoid the service cost of having to transfer to and from the cloud. It's also more efficient, because the application can talk to the database without having to reach out across the network to do so.
- Some applications may not be able to communicate securely across the Internet. If they cannot communicate securely or through a tunnel, then your data is at risk. In the event the application cannot communicate securely, you will need to host it locally where you can have other means of security to protect data as it is transported across networks.
- Also, since you are displaying the application results on an interface like a web browser, you need to ensure that your application is compatible with a variety of browsers and will operate properly using encryption, like SSL, for some or all of the interaction your user has within the application. If you are unable to display the application's results securely when necessary, then a cloud-based solution will be essentially worthless to you.
- If you are relying on applications to be available on the cloud, that may or may not be the case. It depends on whether the developer has created a cloud-friendly version of the application you want. In the event that your application is not ready, you might be out of luck.
- But that doesn't mean that you can't still get what you want done. It is still possible to write your own application.

1.3 DEFINING THE CLOUD

Cloud:

- A *cloud* refers to a distinct IT environment that is designed for the purpose of remotely provisioning scalable and measured IT resources. The term originated as a metaphor for the Internet which is, in essence, a network of networks providing remote access to a set of decentralized IT resources. Prior to cloud computing becoming its own formalized IT industry segment, the symbol of a cloud was commonly used to represent the Internet in a variety of specifications and mainstream documentation of Web-based architectures. This same symbol is now used to specifically represent the boundary of a cloud environment, as shown in Fig. 1.5:

Fig. 1.5 : The symbol used to denote the boundary of a cloud environment

1.4 THE CLOUD COMPUTING REFERENCE MODEL

- Cloud computing conceptual reference model identifies the major actors, their activities and functions in cloud computing.
- Below Fig. 1.6 presents an overview of the NIST cloud reference architecture.

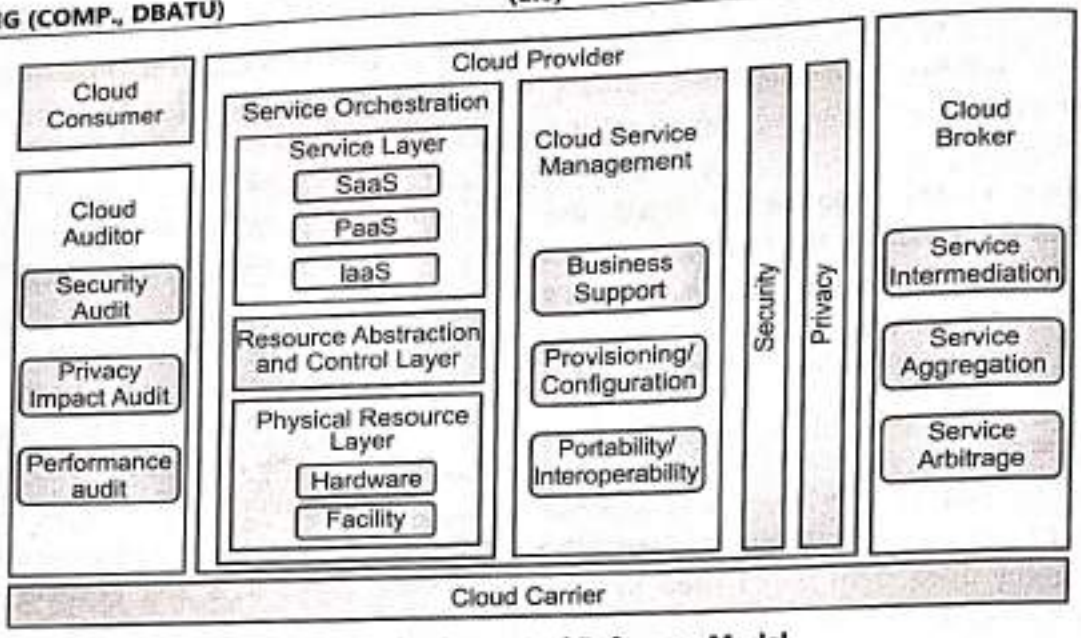


Fig. 1.6 : The Conceptual Reference Model

- As shown in Fig. 1.6, the NIST cloud computing reference architecture defines five major actors:
  1. Cloud consumer.
  2. Cloud provider.
  3. Cloud carrier.
  4. Cloud auditor.
  5. Cloud broker.

  Each actor is an entity (a person or an organization) that participates in a transaction or process and/or performs tasks in cloud computing.
- Table 1.2 briefly lists the actors defined in the NIST cloud computing reference architecture. The general activities of the actors are discussed in further tutorial.

Actors in Cloud Computing

Table 1.2 : Actors in Cloud Computing

Sr. No.	Actor	Definition
1.	Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, Cloud Providers.
2.	Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
3.	Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
4.	Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.
5.	Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.

- Fig. 1.7 illustrates the interactions among the actors. A cloud consumer may request cloud services from a cloud provider directly or via a cloud broker. A cloud auditor conducts independent audits and may contact the others to collect necessary information. The details will be discussed in the following sections and presented in increasing level of details in successive diagrams.

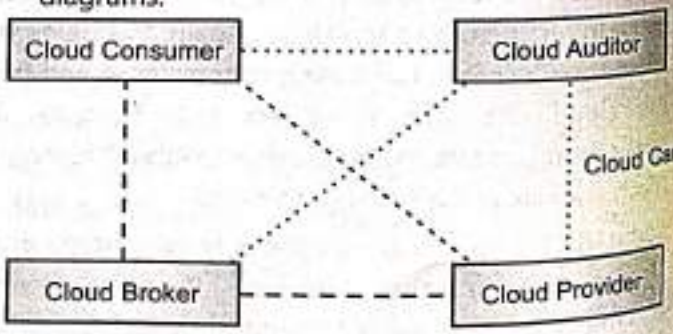


Fig. 1.7 : Interactions between the Actors in Cloud Computing

Usage Scenario 1: A cloud consumer may request service from a cloud broker instead of contacting a cloud provider directly. The cloud broker may create a new service by combining multiple services or by enhancing an existing service. In this example, the actual cloud providers are invisible to the cloud consumer and the cloud consumer interacts directly with the cloud broker.

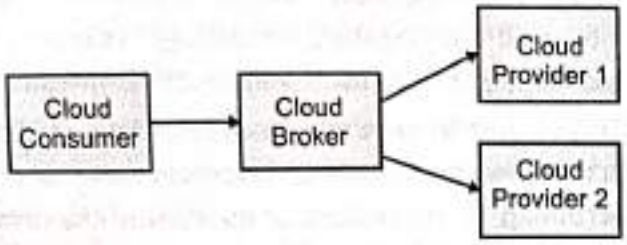


Fig. 1.8 : Usage Scenario for Cloud Brokers

Usage Scenario 2: Cloud carriers provide the connectivity and transport of cloud services from cloud providers to cloud consumers. As illustrated in Fig. 1.9, a cloud provider participates in and arranges for two unique Service Level Agreements (SLAs), one with a cloud carrier (e.g. SLA2) and one with a cloud consumer (e.g. SLA1).

- A cloud provider arranges Service Level Agreements (SLAs) with a cloud carrier and may request dedicated and encrypted connections to ensure the cloud services are consumed at a consistent level according to the contractual obligations with the cloud consumers. In this case, the provider may specify its requirements on capability, flexibility and functionality in SLA2 in order to provide essential requirements in SLA1.

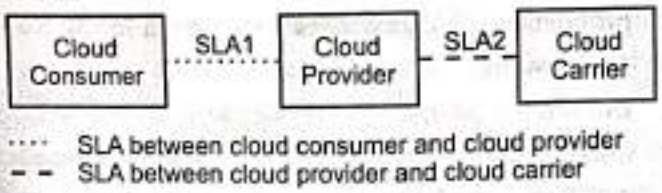


Fig. 1.9 : Usage Scenario for Cloud Carriers

- **Usage Scenario 3:** For a cloud service, a cloud auditor conducts independent assessments of the operation and security of the cloud service implementation. The audit may involve interactions with both the Cloud Consumer and the Cloud Provider.

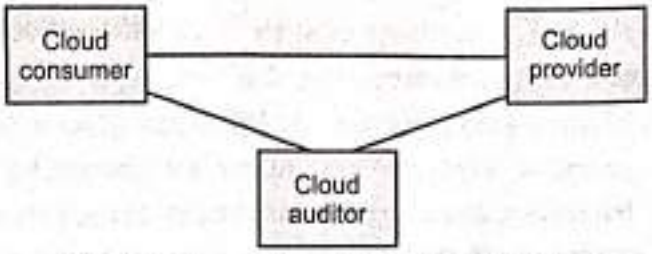


Fig. 1.10: Usage Scenario for Cloud Auditors

Cloud Consumer

- The cloud consumer is the principal stakeholder for the cloud computing service. A cloud consumer represents a person or organization that maintains a business relationship with, and uses the service from a cloud provider. A cloud consumer browses the service catalog from a cloud provider, requests the appropriate service, sets up service contracts with the cloud provider, and uses the service.
- The cloud consumer may be billed for the service provisioned, and needs to arrange payments accordingly. Cloud consumers need SLAs to specify the technical performance requirements fulfilled by a cloud provider. SLAs can cover terms regarding the quality of service, security, remedies for performance failures.

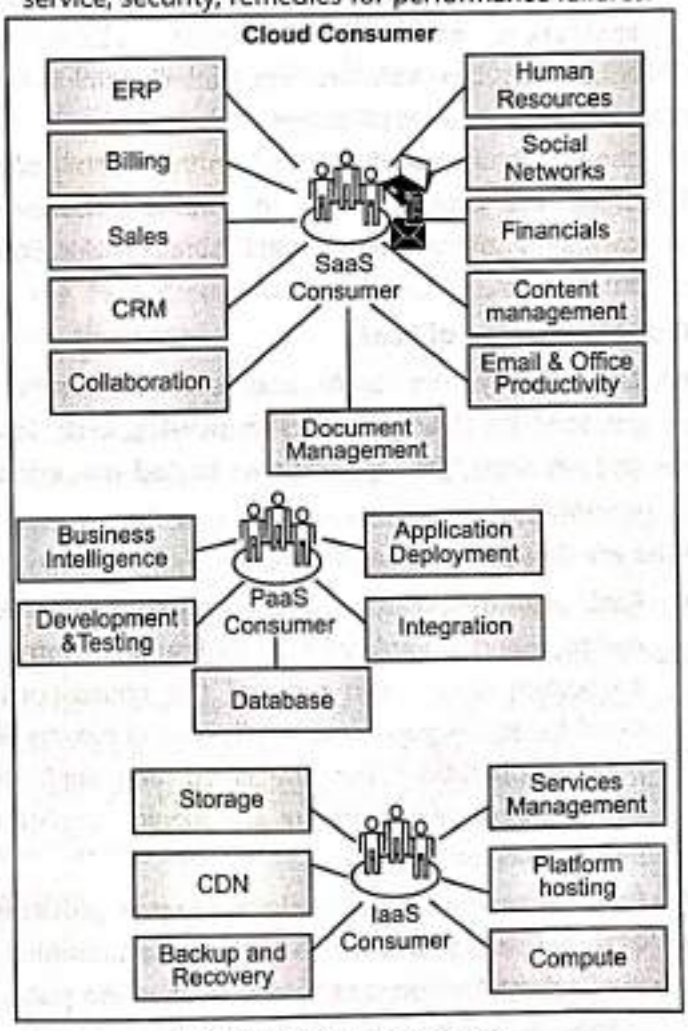


Fig. 1.11 : Cloud consumers

- Fig. 1.11 shows some example cloud services available to a cloud consumer.
- A cloud provider may also list in the SLAs a set of promises explicitly not made to consumers, i.e. limitations, and obligations that cloud consumers must accept. A cloud consumer can freely choose a cloud provider with better pricing and more favourable terms.
- Typically, a cloud provider's pricing policy and SLAs are non-negotiable, unless the customer expects heavy usage and might be able to negotiate for better contracts. Depending on the services requested, the activities and usage scenarios can be different among cloud consumers.

Cloud Consumers of SaaS:

- SaaS applications in the cloud and made accessible via a network to the SaaS consumers.

Who are Cloud Consumer of SaaS?

- The consumers of SaaS can be an organization that provide their members with access to software applications, end users who directly use software applications, or software application administrators who configure applications for end users. SaaS consumers can be billed based on the number of end users, the time of use, the network bandwidth consumed, the amount of data stored or duration of stored data.

Cloud Consumers of PaaS:

- Can employ the tools and execution resources provided by cloud providers to develop, test, deploy and manage the applications hosted in a cloud environment.

Who are Cloud Consumer of PaaS?

- PaaS consumers can be application developers who design and implement application software, application testers who run and test applications in cloud-based environments, application deployers who publish applications into the cloud, and application administrators who configure and monitor application performance on a platform.
- PaaS consumers can be billed according to, processing, database storage and network resources consumed by the PaaS application, and the duration of the platform usage.

Cloud Consumers of IaaS:

- Consumers of IaaS have access to virtual computers, network-accessible storage, network infrastructure components, and other fundamental computing resources on which they can deploy and run arbitrary software.

Who are Cloud Consumer of IaaS?

- The consumers of IaaS can be system developers, system administrators and IT managers who are interested in creating, installing, managing and monitoring services for IT infrastructure operations.
- IaaS consumers are provisioned with the capabilities to access these computing resources, and are billed according to the amount or duration of the resources consumed, such as CPU hours used by virtual computers, volume and duration of data stored, network bandwidth consumed, number of IP addresses used for certain intervals.

Examples of Cloud Services

- Some example cloud services available to a cloud consumer are listed below:

1. SaaS Services

- **Email and Office Productivity:** Applications for email, word processing, spreadsheets, presentations, etc.
- **Billing:** Application services to manage customer billing based on usage and subscriptions to products and services.
- **Customer Relationship Management (CRM):** CRM applications that range from call center applications to sales force automation.
- **Collaboration:** Tools that allow users to collaborate in workgroups, within enterprises, and across enterprises.
- **Content Management:** Services for managing the production of and access to content for web-based applications.
- **Document Management:** Applications for managing documents, enforcing document production workflows, and providing workspaces for groups or enterprises to find and access documents.
- **Financials:** Applications for managing financial processes ranging from expense processing and invoicing to tax management.
- **Human Resources:** Software for managing human resources functions within companies.

Sales: Applications that are specifically designed for sales functions such as pricing, commission tracking, etc.

Social Networks: Social software that establishes and maintains a connection among users that are tied in one or more specific types of interdependency.

Enterprise Resource Planning (ERP): Integrated computer-based system used to manage internal and external resources, including tangible assets, financial resources, materials, and human resources.

2. **PaaS Services**

- **Business Intelligence:** Platforms for the creation of applications such as dashboards, reporting systems, and data analysis.
- **Database:** Services offering scalable relational database solutions or scalable non-SQL datastores.
- **Development and Testing:** Platforms for the development and testing cycles of application development, which expand and contract as needed.
- **Integration:** Development platforms for building integration applications in the cloud and within the enterprise.
- **Application Deployment:** Platforms suited for general purpose application development. These services provide databases, web application runtime environments, etc.

3. **IaaS Services**

- **Backup and Recovery:** Services for backup and recovery of file systems and raw data stores on servers and desktop systems.
- **Compute:** Server resources for running cloud-based systems that can be dynamically provisioned and configured as needed.
- **Content Delivery Networks (CDNs):** CDNs store content and files to improve the performance and cost of delivering content for web-based systems.
- **Services Management:** Services that manage cloud infrastructure platforms. These tools often provide features that cloud providers do not provide or specialize in managing certain application technologies.
- **Storage:** Massively scalable storage capacity that can be used for applications, backups, archival, and file storage.

Cloud Providers

- A cloud provider is a person, or an organization; it is the entity responsible for making a service available to interested parties.
- A Cloud Provider acquires and manages the computing infrastructure required for providing the services, runs the cloud software that provides the services, and makes arrangement to deliver the cloud services to the Cloud Consumers through network access.

What Cloud Providers does for SaaS?

- For Software as a Service, the cloud provider deploys, configures, maintains and updates the operation of the software applications on a cloud infrastructure so that the services are provisioned at the expected service levels to cloud consumers. The provider of SaaS assumes most of the responsibilities in managing and controlling the applications and the infrastructure, while the cloud consumers have limited administrative control of the applications.

What Cloud Providers does for PaaS?

- For PaaS, the Cloud Provider manages the computing infrastructure for the platform and runs the cloud software that provides the components of the platform, such as runtime software execution stack, databases, and other middleware components.
- The PaaS Cloud Provider typically also supports the development
- For deployment and management process of the PaaS Cloud Consumer by providing tools such as Integrated Development Environments (IDEs), development version of cloud software, Software Development Kits (SDKs), deployment and management tools.
- The PaaS Cloud Consumer has control over the applications and possibly some of the hosting environment settings, but has no or limited access to the infrastructure underlying the platform such as network, servers, Operating Systems (OS), or storage.

What Cloud Providers does for IaaS?

For IaaS –

- The Cloud Provider acquires the physical computing resources underlying the service, including the servers, networks, storage and hosting infrastructure.

- The Cloud Provider runs the cloud software necessary to makes computing resources available to the IaaS Cloud Consumer through a set of service interfaces and computing resource abstractions, such as virtual machines and virtual network interfaces.
- The IaaS Cloud Consumer in turn uses these computing resources, such as a virtual computer, for their fundamental computing needs Compared to SaaS and PaaS Cloud Consumers, an IaaS Cloud Consumer has access to more fundamental forms of computing resources and thus has more control over the more software components in an application stack, including the OS and network.
- The IaaS Cloud Provider, on the other hand, has control over the physical hardware and cloud software that makes the provisioning of these infrastructure services possible, for example, the physical servers, network equipments, storage devices, host OS and hypervisors for virtualization.
- A Cloud Provider's activities can be described in five major areas, as shown in below Fig. 1.12, a cloud provider conducts its activities in the areas of service deployment, service orchestration, cloud service management, security, and privacy. We will see detail in respective sections.

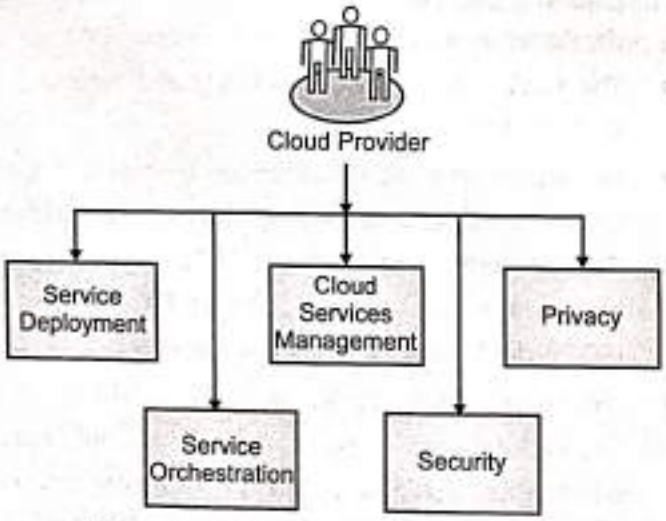


Fig. 1.12: Cloud Provider – Major Activities

Cloud Auditor

- A cloud auditor is a party that can perform an independent examination of cloud service controls with the intent to express an opinion thereon. Audits are performed to verify conformance to standards through review of objective evidence. A cloud auditor can evaluate the services provided by a cloud provider in terms of security controls, privacy impact, performance, etc.
- Auditing is especially important for federal agencies as "agencies should include a contractual clause enabling third parties to assess security controls of cloud providers".

Cloud Security Audit?

- Security controls are the management, operational, and technical safeguards or countermeasures employed within an organizational information system to protect the confidentiality, integrity, and availability of the system and its information.
- For security auditing, a cloud auditor can make an assessment of the security controls in the information system to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to the security requirements for the system.
- The security auditing should also include the verification of the compliance with regulation and security policy. For example, an auditor can be tasked with ensuring that the correct policies are applied to data retention according to relevant rules for the jurisdiction. The auditor may ensure that fixed content has not been modified and that the legal and business data archival requirements have been satisfied.
- A privacy impact audit can help Federal agencies comply with applicable privacy laws and regulations governing an individual's privacy, and to ensure confidentiality, integrity, and availability of an individual's personal information at every stage of development and operation.

Why we need Cloud Broker?

- As cloud computing evolves, the integration of cloud services can be too complex for cloud consumers to manage. A cloud consumer may request cloud services from a cloud broker, instead of contacting a cloud provider directly.

Who is Cloud Broker?

- A cloud broker is an entity that manages the use, performance and delivery of cloud services and negotiates relationships between cloud providers and cloud consumers.

- In general, a cloud broker can provide services in three categories:
  1. **Service Intermediation:** A cloud broker enhances a given service by improving some specific capability and providing value-added services to cloud consumers. The improvement can be managing access to cloud services, identity management, performance reporting, enhanced security, etc.
  2. **Service Aggregation:** A cloud broker combines and integrates multiple services into one or more new services. The broker provides data integration and ensures the secure data movement between the cloud consumer and multiple cloud providers.
  3. **Service Arbitrage:** Service arbitrage is like service aggregation except that the services being aggregated are not fixed. Service arbitrage means a broker has the flexibility to choose services from multiple agencies. The cloud broker, for example, can use a credit-scoring service to measure and select an agency with the best score.

Who is Cloud Carrier?

- A cloud carrier acts as an intermediary that provides connectivity and transport of cloud services between cloud consumers and cloud providers.
- Cloud carriers provide access to consumers through network, telecommunication and other access devices. For example, cloud consumers can obtain cloud services through network access devices, such as computers, laptops, mobile phones, Mobile Internet Devices (MIDs), etc.
- The distribution of cloud services is normally provided by network and telecommunication carriers or a transport agent, where a transport agent refers to a business organization that provides physical transport of storage media such as high-capacity hard drives.
- Note that a cloud provider will set up SLAs with a cloud carrier to provide services consistent with the level of SLAs offered to cloud consumers, and may require the cloud carrier to provide dedicated and secure connections between cloud consumers and cloud providers.

Scope of Control between Provider and Consumer

- The Cloud Provider and Cloud Consumer share the control of resources in a cloud system. As show in below Fig. 1.13, different service models affect an organization's control over the computational resources and thus what can be done in a cloud system.
- The Fig. 1.13 shows these differences using a classic software stack notation comprised of the application, middleware, and OS layers. This analysis of delineation of controls over the application stack helps understand the responsibilities of parties involved in managing the cloud application.

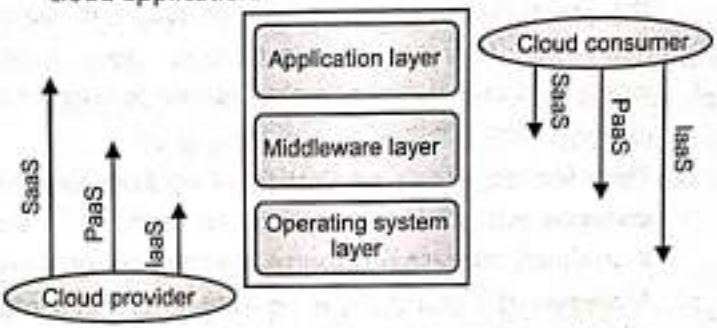


Fig. 1.13 : Scope of Controls between Provider and Consumer

- The application layer includes software applications targeted at end users or programs. The applications are used by SaaS consumers, or installed/managed/maintained by PaaS consumers, IaaS consumers, and SaaS providers.
- The middleware layer provides software building blocks (e.g., libraries, database, and Java virtual machine) for developing application software in the cloud. The middleware is used by PaaS consumers, installed/managed/maintained by IaaS consumers or PaaS providers, and hidden from SaaS consumers.
- The OS layer includes operating system and drivers, and is hidden from SaaS consumers and PaaS consumers. An IaaS cloud allows one or multiple guest OS"s to run virtualized on a single physical host. Generally, consumers have broad freedom to choose which OS to be hosted among all the OS"s that could be supported by the cloud provider. The IaaS consumers should assume full responsibility for the guest OS"s, while the IaaS provider controls the host OS.

1.5 CHARACTERISTICS AND BENEFITS

- The common benefits associated with adopting cloud computing are explained as below.

1.5.1 Reduced Investments and Proportional Costs

- Similar to a product wholesaler that purchases goods in bulk for lower price points, public cloud providers base their business model on the mass-acquisition of

IT resources that are then made available to cloud consumers via attractively priced leasing packages.

- This opens the door for organizations to gain access to powerful infrastructure without having to purchase it themselves. The most common economic rationale for investing in cloud-based IT resources is in the reduction or outright elimination of up-front IT investments, namely hardware and software purchases and ownership costs. A cloud's measured usage characteristic represents a feature-set that allows measured operational expenditures to replace anticipated capital expenditures. This is also referred to as proportional costs.
- This elimination or minimization of up-front financial commitments allows enterprises to start small and accordingly increase IT resource allocation as required.
- Moreover, the reduction of up-front capital expenses allows for the capital to be redirected to the core business investment. In its most basic form, opportunities to decrease costs are derived from the deployment and operation of large-scale data centers by major cloud providers. Such data centers are commonly located in destinations where real estate, IT professionals, and network bandwidth can be obtained at lower costs, resulting in both capital and operational savings.

Common measurable benefits to cloud consumers include:

- On-demand access to pay-as-you-go computing resources on a short-term basis (such as processors by the hour), and the ability to release these computing resources when they are no longer needed.
- The perception of having unlimited computing resources that are available on demand, thereby reducing the need to prepare for provisioning.
- The ability to add or remove IT resources at a fine-grained level, such as modifying available storage disk space by single gigabyte increments.
- Abstraction of the infrastructure so applications are not locked into devices or locations and can be easily moved if needed.

For example, a company with sizable batch-centric tasks can complete them as quickly as their application software can scale. Using 100 servers for one hour costs the same as using one server for 100 hours. This "elasticity" of IT resources, achieved without requiring steep initial investments to create a large-scale computing infrastructure, can be extremely compelling.

1.5.2 Increased Scalability

- By providing pools of IT resources, along with tools and technologies designed to leverage them collectively, clouds can instantly and dynamically allocate IT resources to cloud consumers, on demand or via the cloud consumer's direct configuration.
- This empowers cloud consumers to scale their cloud-based IT resources to accommodate processing fluctuations and peaks automatically or manually. Similarly, cloud-based IT resources can be released as processing demands decrease.
- The inherent, built-in feature of clouds to provide flexible levels of scalability to IT resources is directly related to the above-mentioned proportional cost benefit. Besides the evident financial gain to the automated reduction of scaling, the ability of IT resources to always meet and fulfill unpredictable usage demands avoids potential loss of business that can occur when usage thresholds are met.

1.5.3 Increased Availability and Reliability

- The availability and reliability of IT resources are directly associated with tangible business benefits. Outages limit the time an IT resource can be "open for business" for its customers, thereby limiting its usage and revenue generating potential. Runtime failures that are not immediately corrected can have a more significant impact during high-volume usage periods. Not only is the IT resource unable to respond to customer requests, its unexpected failure can decrease overall customer confidence.
- A hallmark of the typical cloud environment is its intrinsic ability to provide extensive support for increasing the availability of a cloud-based IT resource to minimize or even eliminate outages, and for increasing its reliability so as to minimize the impact of runtime failure conditions.

Specifically:

- An IT resource with increased availability is accessible for longer periods of time. Cloud providers generally offer "resilient" IT resources for which they are able to guarantee high levels of availability.
- An IT resource with increased reliability is able to better avoid and recover from exception conditions. The modular architecture of cloud environments provides extensive failover support that increases reliability.

1.6 CHALLENGES

- Several of the most critical cloud computing challenges pertaining mostly to cloud consumers that use IT resources located in public clouds are presented and examined as follows.

1.6.1 Increased Vulnerabilities in Security

- The moving of business data to the cloud means that the responsibility over data security becomes shared with the cloud provider.
- The remote usage of IT resources requires an expansion of trust boundaries by the cloud consumer to include the external cloud. It can be difficult to establish a security architecture that spans such a trust boundary without introducing vulnerabilities, unless cloud consumers and cloud providers happen to support the same or compatible security frameworks which is unlikely with public clouds. Another consequence of overlapping trust boundaries relates to the cloud provider's privileged access to cloud consumer data.
- The extent to which the data is secure is now limited to the security controls and policies applied by both the cloud consumer and cloud provider. Furthermore, there can be overlapping trust boundaries from different cloud consumers due to the fact that cloud-based IT resources are commonly shared.

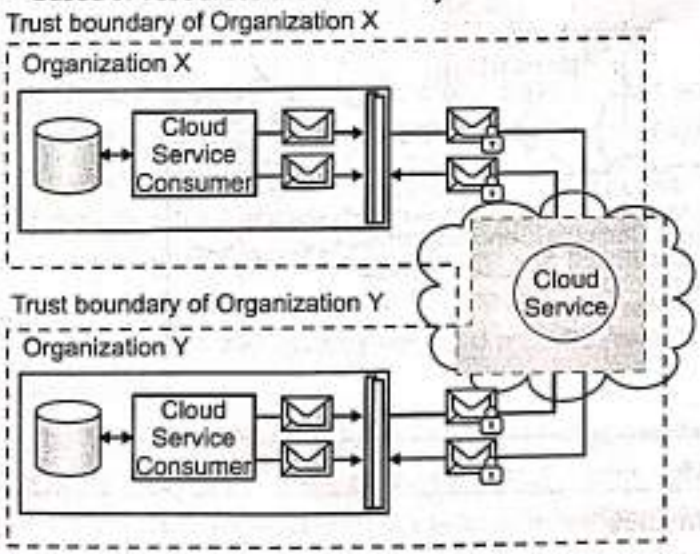


Fig. 1.14: The shaded area with diagonal lines indicates the overlap of two organizations trust boundaries

- The overlapping of trust boundaries and the increased exposure of data can provide malicious cloud consumers with greater opportunities to attack IT resources and steal or damage business data. Fig. 1.14 illustrates a scenario whereby two organizations accessing the same cloud service are required to extend their respective trust boundaries to the cloud, resulting in overlapping trust boundaries. It can be challenging for the cloud provider to offer security mechanisms that accommodate the security requirements of both cloud service consumers.

1.6.2 Reduced Operational Governance Control

- Cloud consumers are usually allotted a level of governance control that is lower than that over on-premise IT resources. This can introduce risks associated with how the cloud provider operates its cloud, as well as the external connections that are required for communication between the cloud and the cloud consumer.

Consider the Following Examples:

- An unreliable cloud provider may not maintain the guarantees it makes in the Service Level Agreements (SLAs) that were published for its cloud services.
- Longer geographic distances between the cloud consumer and cloud provider can require additional network hops that introduce fluctuating latency and potential bandwidth constraints.
- Legal contracts, when combined with SLAs, technology inspections, and monitoring, can mitigate governance risks and issues. A cloud governance system is established through SLAs, given the "as-a-service" nature of cloud computing. A cloud consumer must keep track of the actual service level being offered and the other warranties that are made by the cloud provider.

1.6.3 Limited Portability between Cloud Providers

- Due to a lack of established industry standards within the cloud computing industry, public clouds are commonly proprietary to various extents. For cloud consumers that have custom-built solutions with dependencies on these proprietary environments, it can be challenging to move from one cloud provider to another. Portability is a measure used to determine the impact of moving cloud consumer IT resources and data between clouds.

1.6.4 Multi-Regional Compliance and Legal Issues

- Third-party cloud providers will frequently establish data centers in affordable or convenient geographical locations. Cloud consumers will often not be aware of

the physical location of their IT resources and data when hosted by public clouds. For some organizations, this can pose serious legal concerns pertaining to industry or government regulations that specify data privacy and storage policies. For example, some UK laws require personal data belonging to UK citizens to be kept within the United Kingdom.

- Another potential legal issue pertains to the accessibility and disclosure of data. Countries have laws that require some types of data to be disclosed to certain government agencies or to the subject of the data. For example, a European cloud consumer's data that is located in the U.S. can be more easily accessed by government agencies (due to the U.S. Patriot Act) when compared to data located in many European Union countries.

1.7 HISTORIC DEVELOPMENTS

- Cloud Computing is not a latest technology. Cloud computing has evolved (develop gradually) through a number of phases which includes Grid Computing, Utility Computing, Application Service Provision, and Software as a Service etc. But the overarching (overall) concept of delivering Computing resource through a global network is started in the series.
- By 2020 The Cloud computing market is forecast to exceed $241 Billion. But how did we get here and where did all this started is the history of Cloud computing.
- The actual history of Cloud computing is not that old. The first business and consumer Cloud Computing services website (Salesforce.com and Google) was launched in 1999. Cloud computing is tied directly to the development of the Internet and Business technology since Cloud computing is the solution to the problem of how the Internet can help improve Business Technology.
- Business technology has long and fascinating history, one that is almost as long as business itself, but the development that most directly influenced the history of Cloud computing start with the emergence of computers as providers of real business solutions.

History of Cloud Computing

- Cloud computing is one the most innovative technology of our time. Following is a brief history of Cloud computing.

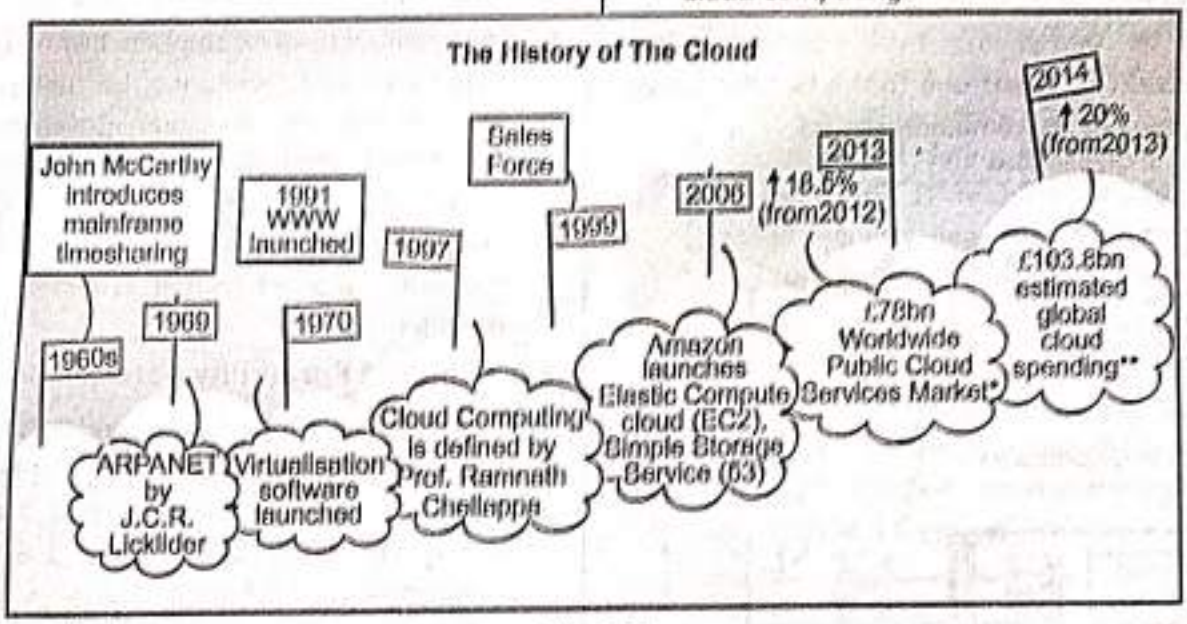


Fig. 1.15

Early 1960s

- The computer scientist John McCarthy, come up with concept of timesharing, and enabling Organization to simultaneously use an expensive mainframe. This computing is described as a significant contribution to the development of the Internet, and a pioneer of Cloud computing.

In 1969

- The idea of an "Intergalactic Computer Network" or "Galactic Network" (a computer networking concept similar to today's Internet) was introduced by J.C.R. Licklider, who was responsible for enabling the development of ARPANET (Advanced Research Projects Agency Network). His vision was for everyon

on the globe to be interconnected and being able to access programs and data at any site, from anywhere.

In 1970

- Using virtualization software like VMware, It become possible to run more than one Operating System simultaneously in an isolated environment. It was possible to run a completely different Computer (virtual machine) inside a different Operating System.

In 1997

- The first known definition of the term "Cloud Computing" seems to be by Prof. Ramnath Chellappa in Dallas in 1997 – "A computing paradigm where the boundaries of computing will be determined by economic rationale rather than technical limits alone."

In 1999

- The arrival of Salesforce.com in 1999 pioneered the concept of delivering enterprise applications via simple website. The services firm covered the way for both specialist and mainstream software firms to deliver applications over the Internet.

In 2003

- The first public release of Xen, which creates a Virtual Machine Monitor (VMM) also known as a hypervisor, a software system that allows the execution of multiple virtual guest operating systems simultaneously on a single machine.

In 2006

- In 2006, Amazon expanded its cloud services. First was its Elastic Compute cloud (EC2), which allowed people to access computers and run their own applications on them, all on the cloud. Then they brought out Simple Storage Service (S3). This introduced the pay-as-you-go model to both users and the industry as a whole, and it has basically become standard practice now.

In 2013

- The Worldwide Public Cloud Services Market totalled £78bn, up 18.5 per cent on 2012, with IaaS (infrastructure-as-a-service) the fastest growing market service.

In 2014

- In 2014, global business spending for infrastructure and services related to the cloud will reach an estimated £103.8bn, up 20% from the amount spent in 2013 (Constellation Research).

Vision of Cloud Computing

- We have seen how far Cloud computing has progressed in the short time since its initiation. Now lets have a look on what may become of Cloud computing technology in the future.

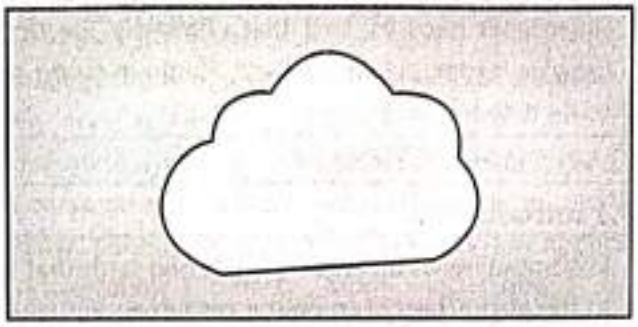


Fig. 1.16

Following are few forecasts of what we might expect in the coming future of Cloud computing:

- Cloud computing will become even more prominent in the coming years with rapid, continued growth of major global cloud data centres.
- 50% of all IT will be in the cloud within the next 5 – 10 years.
- There will be a greater use of cloud technology as a whole across emerging markets such as in the BRIC countries (Brazil, Russia, India and China) as they continue to develop and progress. The uptake will be particularly evident in Asia where there is already a trend to stay on the edge of the latest technology.
- Data for companies and personal use will be available everywhere in standardized formats, allowing us to easily consume and interact with one another at an even greater level.
- The security and reliability of cloud computing will continue to evolve, ensuring that data will be even more secure with numerous techniques employed.
- We will not even consider 'cloud' as the key technology, instead we will focus on the services and applications that it enables.
- Combining cloud technology with the Internet of Things (IOT), Wearables and Bring Your Own Device (BYOD) will become the norm in personal and working lives, so much so that the presence of cloud technology as an enabler will be overlooked. An estimated 50% of organizations will require employees to use their own devices by 2017.

- The total global cloud computing spend will reach $241 Billion in 2020.
- The future of the cloud is far from certain. The rapid pace at which technology has changed in the last 5 years makes the next 5 near impossible to predict. However it must be said that ultimately the cloud is growing exponentially and will continue to do so for some time to come.

1.8 VIRTUALIZATION

1.8.1 Introduction

- In computing, virtualization is a broad term that refers to the abstraction of computer resources. Virtualization hides the physical characteristics of computing resources from their users, be they applications, or end users. This includes making a single physical resource (such as a server, an operating system, an application, or storage device) appear to function as multiple virtual resources; it can also include making multiple physical resources (such as storage devices or servers) appear as a single virtual resource."

In layman's terms virtualization is often:

1. The creation of many virtual resources from one physical resource.
2. The creation of one virtual resource from one or more physical resource.

- The term is frequently used to convey one of these concepts in a variety of areas such as networking, storage, and hardware.

History

- Virtualization is not a new concept. One of the early works in the field was a paper by Christopher Strachey entitled "Time Sharing in Large Fast Computers". IBM began exploring virtualization with its CP-40 and M44/44X research systems. These in turn lead to the commercial CP-67/CMS. The virtual machine concept kept users separated while simulating a full stand-alone computer for each.
- In the 80's and early 90's the industry moved from leveraging singular mainframes to running collections of smaller and cheaper x86 servers. As a result the concept of virtualization become less prominent. That changed in 1999 with VMware's introduction of VMware workstation. This was followed by VMware's ESX Server, which runs on bare metal and does not require a host operating system.

Types of Virtualization

- Today the term virtualization is widely applied to number of concepts including:
  1. Server Virtualization
  2. Client / Desktop / Application Virtualization
  3. Network Virtualization
  4. Storage Virtualization
  5. Service / Application Infrastructure Virtualization
- In most of these cases, either virtualizing one physical resource into many virtual resources or turning many physical resources into one virtual resource is occurring.

1. Server Virtualization

- Server virtualization is the most active segment of the virtualization industry featuring established companies such as VMware, Microsoft, and Citrix. With server virtualization one physical machine is divided many virtual servers. At the core of such virtualization is the concept of a hypervisor (virtual machine monitor). A hypervisor is a thin software layer that intercepts operating system calls to hardware. Hypervisors typically provide a virtualized CPU and memory for the guests running on top of them. The term was first used in conjunction with the IBM CP-370.
- Hypervisors are classified as one of two types:

  Type 1 : This type of hypervisor is also known as native or bare-metal. They run directly on the hardware with guest operating systems running on top of them. Examples include VMware ESX, Citrix XenServer, and Microsoft's Hyper-V.

  Type 2 : This type of hypervisor runs on top of an existing operating system with guests running at a third level above hardware. Examples include VMware Workstation and SWSoft's Parallels Desktop.
- Related to type 1 hypervisors is the concept of paravirtualization. Paravirtualization is a technique in which a software interface that is similar but not identical to the underlying hardware is presented. Operating systems must be ported to run on top of a paravirtualized hypervisor. Modified operating systems use the "hypercalls" supported by the paravirtualized hypervisor to interface directly with the hardware. The popular Xen project makes use of this type of virtualization. Starting with version 3.0 however Xen

also able to make use of the hardware assisted virtualization technologies of Intel (VT-x) and AMD (AMD-V). These extensions allow Xen to run unmodified operating systems such as Microsoft Windows.

- Server virtualization has a large number of benefits for the companies making use of the technology. Among those frequently listed:
  - ➢ **Increased Hardware Utilization :** This results in hardware saving, reduced administration overhead, and energy savings.
  - ➢ **Security :** Clean images can be used to restore compromised systems. Virtual machines can also provide sandboxing and isolation to limit attacks.
  - ➢ **Development :** Debugging and performance monitoring scenarios can be easily setup in a repeatable fashion. Developers also have easy access to operating systems they might not otherwise be able to install on their desktops.
- Correspondingly there are a number of potential downsides that must be considered:
  - ➢ **Security** : There are now more entry points such as the hypervisor and virtual networking layer to monitor. A compromised image can also be propagated easily with virtualization technology.
  - ➢ **Administration** : While there are less physical machines to maintain there may be more machines in aggregate. Such maintenance may require new skills and familiarity with software that administrators otherwise would not need.
  - ➢ **Licensing/Cost Accounting :** Many software-licensing schemes do not take virtualization into account. For example running four copies of Windows on one box may require four separate licenses.
  - ➢ **Performance** : Virtualization effectively partitions resources such as RAM and CPU on a physical machine. This combined with hypervisor overhead does not result in an environment that focuses on maximizing performance.

2. Client / Desktop / Application Virtualization

- Virtualization is not only a server domain technology. It is being put to a number of uses on the client side at both the desktop and application level. Such virtualization can be broken out into four categories:
  - (i) Local Application Virtualization/Streaming
  - (ii) Hosted Application Virtualization
  - (iii) Hosted Desktop Virtualization
  - (iv) Local Desktop Virtualization
- Application virtualization is an umbrella term that describes software technologies that improve manageability and compatibility of legacy applications by encapsulating applications from the underlying operating system on which they are executed. A fully virtualized application is not installed in the traditional sense, although it is still executed as if it is. Application virtualization differs from operating system virtualization in that in the latter case, the whole operating system is virtualized rather than only specific applications.
- With streamed and local application virtualization an application can be installed on demand as needed. If streaming is enabled then the portions of the application needed for startup are sent first optimizing startup time. Locally virtualized applications also frequently make use of virtual registries and file systems to maintain separation and cleanness from the user's physical machine. Examples of local application virtualization solutions include Citrix Presentation Server and Microsoft SoftGrid. One could also include virtual appliances into this category such as those frequently distributed via VMware's VMware Player.
- Hosted application virtualization allows the user to access applications from their local computer that are physically running on a server somewhere else on the network. Technologies such as Microsoft's RemoteApp allow for the user experience to be relatively seamless include the ability for the remote application to be a file handler for local file types.

Benefits of Application Virtualization Include:

- **Security :** Virtual applications often run in user mode isolating them from OS level functions.
- **Management :** Virtual applications can be managed and patched from a central location.
- **Legacy Support :** Through virtualization technologies legacy applications can be run on modern operating systems they were not originally designed for.
- **Access** : Virtual applications can be installed on demand from central locations that provide failover and replication.

Disadvantages Include:

- **Packaging :** Applications must first be packaged before they can be used.
- **Resources :** Virtual applications may require more resources in terms of storage and CPU.
- **Compatibility :** Not all applications can be virtualized easily.
- Desktop virtualization (or Virtual Desktop Infrastructure) is a server-centric computing model that borrows from the traditional thin-client model but is designed to give administrators and end users the best of both worlds: the ability to host and centrally manage desktop virtual machines in the data center while giving end users a full PC desktop experience.
- Hosted desktop virtualization is similar to hosted application virtualization, expanding the user experience to be the entire desktop. Commercial products include Microsoft's Terminal Services, Citrix's XenDesktop, and VMware's VDI.

Benefits of Desktop Virtualization Include Most of Those with Application Virtualization as Well as:

- ***High Availability*** : Downtime can be minimized with replication and fault tolerant hosted configurations.
- ***Extended Refresh Cycles :*** Larger capacity servers as well as limited demands on the client PCs can extend their lifespan.
- ***Multiple Desktops :*** Users can access multiple desktops suited for various tasks from the same client PC.
- Disadvantages of desktop virtualization are similar to server virtualization. There is also the added disadvantage that clients must have network connectivity to access their virtual desktops. This is problematic for offline work and also increases network demands at the office.
- The final segment of client virtualization is local desktop virtualization. It could be said that this is where the recent resurgence of virtualization began with VMware's introduction of VMware Workstation in the late 90's. Today the market includes competitors such as Microsoft Virtual PC and Parallels Desktop. Local desktop virtualization has also played a key part in the increasing success of Apple's move to Intel processors since products like VMware Fusion and Parallels allow easy access to Windows applications. Some the benefits of local desktop virtualization include:
- **Security :** With local virtualization organizations can lock down and encrypt just the valuable contents of the virtual machine/disk. This can be more performant than encrypting a user's entire disk or operating system.
- **Isolation :** Related to security is isolation. Virtual machines allow corporations to isolate corporate assets from third party machines they do not control. This allows employees to use personal computers for corporate use in some instances.
- **Development/Legacy Support :** Local virtualization allows a users computer to support many configurations and environments it would otherwise not be able to support without different hardware or host operating system. Examples of this include running Windows in a virtualized environment on OS X and legacy testing Windows 98 support on a machine that's primary OS is Vista.

3. Network Virtualization

- Up to this point the types of virtualization covered have centered on applications or entire machines. These are not the only granularity levels that can be virtualized however. Other computing concepts also lend themselves to being software virtualized as well. Network virtualization is one such concept.
- In computing, network virtualization is the process of combining hardware and software network resources and network functionality into a single, software-based administrative entity, a virtual network. Network virtualization involves platform virtualization, often combined with resource virtualization. Network virtualization is categorized as either external, combining many networks, or parts of networks, into a virtual unit, or internal, providing network-like functionality to the software containers on a single system.
- Using the internal definition of the term, desktop and server virtualization solutions provide networking access between both the host and guest as well as between many guests. On the server side virtual switches are gaining acceptance as a part of the virtualization stack. The external definition of network

virtualization is probably the more used version of the term however. Virtual Private Networks (VPNs) have been a common component of the network administrators' toolbox for years with most companies allowing VPN use. Virtual LANs (VLANs) are another commonly used network virtualization concept. With network advances such as 10 gigabit Ethernet, networks no long need to be structured purely along geographical lines. Companies with products in the space include Cisco and 3Leaf.

In General Benefits of Network Virtualization Include:

- **Customization of Access** : Administrators can quickly customize access and network options such as bandwidth throttling and quality of service.
- **Consolidation** : Physical networks can be combined into one virtual network for overall simplification of management.

Similar to server virtualization, network virtualization can bring increased complexity, some performance overhead, and the need for administrators to have a larger skill set.

4. Storage Virtualization

- Another computing concept that is frequently virtualized is storage. Unlike the definitions we have seen up to this point that have been complex at times.
- Storage virtualization refers to the process of abstracting logical storage from physical storage.
- While RAID at the basic level provides this functionality, the term storage virtualization typically includes additional concepts such as data migration and caching. Storage virtualization is hard to define in a fixed manner due to the variety of ways that the functionality can be provided. Typically, it is provided as a feature of:
  - ➢ Host Based with Special Device Drivers
  - ➢ Array Controllers
  - ➢ Network Switchs
  - ➢ Stand Alone Network Appliances
- Each vendor has a different approach in this regard. Another primary way that storage virtualization is classified is whether it is in-band or out-of-band. In-band (often called symmetric) virtualization sits between the host and the storage device allowing caching. Out-of-band (often called asymmetric) virtualization makes use of special host based device drivers that first lookup the meta data (indicating where a file resides) and then allows the host to directly retrieve the file from the storage location. Caching at the virtualization level is not possible with this approach.

General Benefits of Storage Virtualization Include:

- **Migration** : Data can be easily migrated between storage locations without interrupting live access to the virtual partition with most technologies.
- **Utilization** : Similar to server virtualization, utilization of storage devices can be balanced to address over and under utilization.
- **Management** : Many hosts can leverage storage on one physical device that can be centrally managed.

Some of the Disadvantages Include:

- **Lack of Standards and Interoperability :** Storage virtualization is a concept and not a standard. As a result vendors frequently do not easily interoperate.
- **Metadata :** Since there is a mapping between logical and physical location, the storage metadata and its management becomes key to a working reliable system.
- **Backout :** The mapping between local and physical locations also makes the backout of virtualization technology from a system a less than trivial process.

5. Service / Application Infrastructure Virtualization

- Enterprise application providers have also taken note of the benefits of virtualization an begun offering solutions that allow the virtualization of commonly used applications such as Apache as well as application fabric platforms that allow software to easily be developed with virtualization capabilities from the ground up.
- Application infrastructure virtualization (sometimes referred to as application fabrics) unbundle an application from a physical OS and hardware. Application developers can then write to a virtualization layer. The fabric can then handle features such as deployment and scaling. In essence this process is the evolution of grid computing into a fabric form that provides virtualization level features. Companies such as Appistry and DataSynapse provides features including:
  - ➢ Virtualized Distribution
  - ➢ Virtualized Processing
  - ➢ Dynamic Resource Discovery

- IBM has also embraced the virtualization concept at the application infrastructure level with the rebranding and continued of enhancement of Websphere XD as Websphere Virtual Enterprise. The product provides features such as service level management, performance monitoring, and fault tolerance. The software runs on a variety of Windows, Unix, and Linux based operating systems and works with popular application servers such as WebSphere, Apache, BEA, JBoss, and PHP application servers. This lets administrators deploy and move application servers at a virtualization layer level instead of at the physical machine level.

Final Thoughts

- In summary it should now be apparent that virtualization is not just a server-based concept. The technique can be applied across a broad range of computing including the virtualization of:
  - ➢ Entire Machines on Both the Server and Desktop
  - ➢ Applications/Desktops
  - ➢ Storage
  - ➢ Networking
  - ➢ Application Infrastructure
- The technology is evolving in a number of different ways but the central themes revolve around increased stability in existing areas and accelerating adoption by segments of the industry that have yet to embrace virtualization. The recent entry of Microsoft into the bare-metal hypervisor space with Hyper-V is a sign of the technology's maturity in the industry.

1.8.2 Characteristics of Virtualized Environment

1. Increased Security :

- The ability to control the execution of a guest programs in a completely transparent manner opens new possibilities for delivering a secure, controlled execution environment. All the operations of the guest programs are generally performed against the virtual machine, which then translates and applies them to the host programs.
- A virtual machine manager can control and filter the activity of the guest programs, thus preventing some harmful operations from being performed. Resources exposed by the host can then be hidden or simply protected from the guest. Increased security is a requirement when dealing with untrusted code.

Example 1: Untrusted code can be analyzed in Cuckoo sandboxes environment.

- The term sandbox identifies an isolated execution environment where instructions can be filtered and blocked before being translated and executed in the real execution environment.

Example 2: The expression sandboxed version of the Java Virtual Machine (JVM) refers to a particular configuration of the JVM where, by means of security policy, instructions that are considered potentially harmful can be blocked.

2. Managed Execution :

- In particular, sharing, aggregation, emulation, and isolation are the most relevant features.

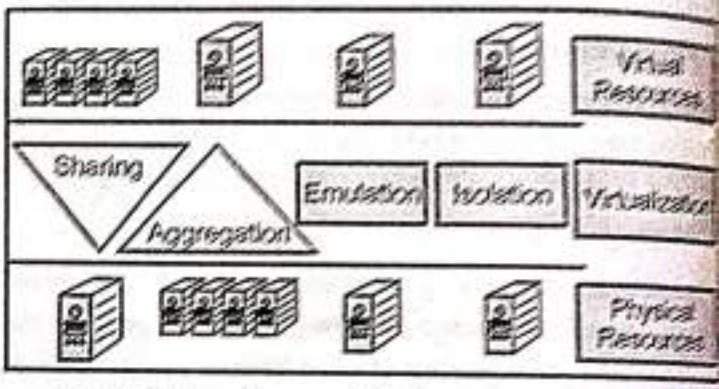


Fig. 1.17 : Functions enabled by managed execution

3. Sharing :

- Virtualization allows the creation of a separate computing environments within the same host. This basic feature is used to reduce the number of active servers and limit power consumption.

4. Aggregation :

- Not only it is possible to share physical resources among several guests, but virtualization also allows aggregation, which is the opposite process. A group of separate hosts can be tied together and represented to guests as a single virtual host. This functionality is implemented with cluster management software, which harnesses the physical resources of a homogeneous group of machines and represents them as a single resource.

5. Emulation :

- Guest programs are executed within an environment that is controlled by the virtualization layer, which ultimately is a program. Also a completely differ

environment with respect to the host can be emulated, thus allowing the execution of guest programs requiring specific characteristics that are not present in the physical host.

5. **Isolation :**

Virtualization allows providing guests whether they are operating systems, applications, or other entities with a completely separate environment, in which they are executed. The guest program performs its activity by interacting with an abstraction layer, which provides access to the underlying resources. The virtual machine can filter the activity of the guest and prevent harmful operations against the host.

Besides these characteristics, another important capability enabled by virtualization is performance tuning. This feature is a reality at present, given the considerable advances in hardware and software supporting virtualization. It becomes easier to control the performance of the guest by finely tuning the properties of the resources exposed through the virtual environment. This capability provides a means to effectively implement a Quality-of-Service (QoS) infrastructure.

7. **Portability :**

- The concept of portability applies in different ways according to the specific type of virtualization considered.
- In the case of a hardware virtualization solution, the guest is packaged into a virtual image that, in most cases, can be safely moved and executed on top of different virtual machines.
- In the case of programming-level virtualization, as implemented by the JVM or the .NET runtime, the binary code representing application components (jars or assemblies) can run without any recompilation on any implementation of the corresponding virtual machine.

1.9 TAXONOMY OF VIRTUALIZATION TECHNIQUES

Virtualization

- A technique for abstracting (or hiding) the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources. This includes making a single physical resource (such as a server, an operating system, an application, or storage device) appear to function as multiple logical resources; or it can include making multiple physical resources (such as storage devices or servers) appear as a single logical resource.

Hypervisor

- A relatively small software (or firmware) component that enables multiple 'guest' operating systems to dynamically share the resources of an underlying 'host' system, by allocating resources and providing an interface for all low-level compute requests (e.g., for CPU, memory, disk or network I/O, etc). A hypervisor can run directly on top of bare hardware to provide a server virtualization environment, or on top of a fully functioning operating system to provide an OS virtualization environment. Also known as a Virtual Machine Monitor or Manager (VMM). In common usage these terms are interchangeable, even though technically they provide different functions.

Hardware Virtualization

- A method of running multiple guest operating environments directly on top of base hardware, allocating fully discrete physical hardware resources (CPU, memory, I/O channels, etc,) separately to each guest, without requiring a complete host operating system.
- Typically used in older and larger server systems, but also recently adapted at chip-level for micro-level x86 environments, this method uses a single enclosure to house essentially isolated compute hardware components, which are not shared by any of the guest operating environments.

Server Virtualization

- A method of running multiple guest operating environments directly on top of base hardware, sharing fine-grained resources (CPU, memory, etc.), without requiring a complete host operating system. This method of virtualization runs standard operating systems such as Windows, UNIX, or Linux on top of a hypervisor that is installed directly onto a bare system. While this is most commonly used for server environments, it is equally capable of hosting desktop

environments. Also known as hardware emulation or as native, platform, system, or "Type 1" virtualization.

Para Virtualization

- A type of server virtualization where the guest OS makes some specific system requests (or 'hypercalls') intentionally to the hypervisor, rather than to the base hardware (to be intercepted and translated by the hypervisor). Hypercalls are typically made for resources that are difficult, impossible, or unsafe to virtualize (such as network and storage I/O calls, or privileged operations like updating page tables). Paravirtualization requires guest operating systems to be modified (paravirtualized or 'enlightened') with specific hypervisor-aware drivers, so that they are aware of this unique environment.

Operating System (OS) Virtualization

- A method of running multiple logical (or virtual) operating systems (or "guests") on top of a fully functioning base (or "host") operating system. This method of virtualization usually uses a standard operating system such as Windows, UNIX, or Linux as the host, plus a hypervisor, to run multiple guest operating systems. Sometimes referred to as "Type 2" virtualization.

Application Virtualization

- *A method of providing an individual application to an end user without needing to completely install this application on the user's local system. Unlike traditional client-server operations, the application itself is not necessarily designed to be used by multiple users at one time, and indeed is unlikely to be shared in the same way. Each user has their own, fully functional application environment, with few or no components actually being shared with other users*

1.10 CLOUD COMPUTING AND VIRTUALIZATION

- Virtualization is a technique of how to separate a service from the underlying physical delivery of that service. It is the process of creating a virtual version of something like computer hardware. It was initially developed during the mainframe era. It involves using specialized software to create a virtual or software-created version of a computing resource rather than the actual version of the same resource. With the help of Virtualization, multiple operating systems and applications can run on same machine and its same hardware at the same time, increasing the utilization and flexibility of hardware.
- In other words, one of the main cost effective, hardware reducing, and energy saving techniques used by cloud providers is virtualization. Virtualization allows to share a single physical instance of a resource or an application among multiple customers and organizations at one time. It does this by assigning a logical name to a physical storage and providing a pointer to that physical resource on demand. The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions for cloud computing. Moreover, virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory and networking.

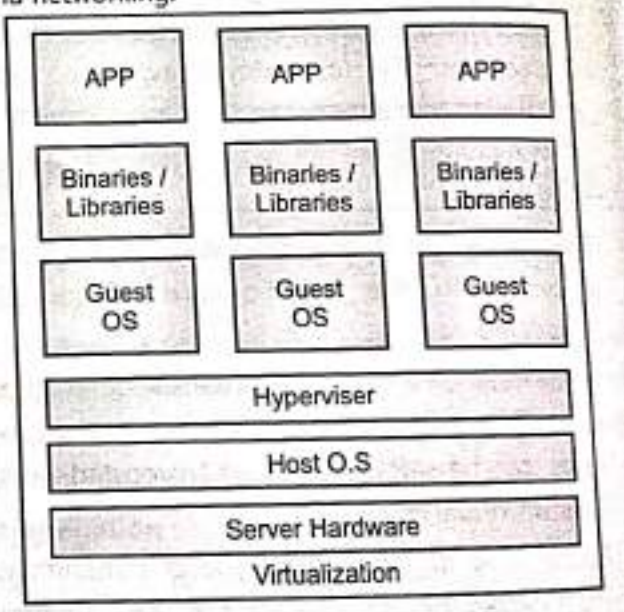


Fig. 1.18

- The machine on which the virtual machine is going be build is known as Host Machine and that virtual machine is referred as a Guest Machine.

Benefits of Virtualization

- More flexible and efficient allocation of resources.
- Enhance development productivity.
- It lowers the cost of IT infrastructure.
- Remote access and rapid scalibility.
- High availability and disaster recovery.

- Pay per use of the IT infrastructure on demand.
- Enables running multiple operating system.

Types of Virtualization:

1. Application Virtualization.
2. Network Virtualization.
3. Desktop Virtualization.
4. Storage Virtualization.

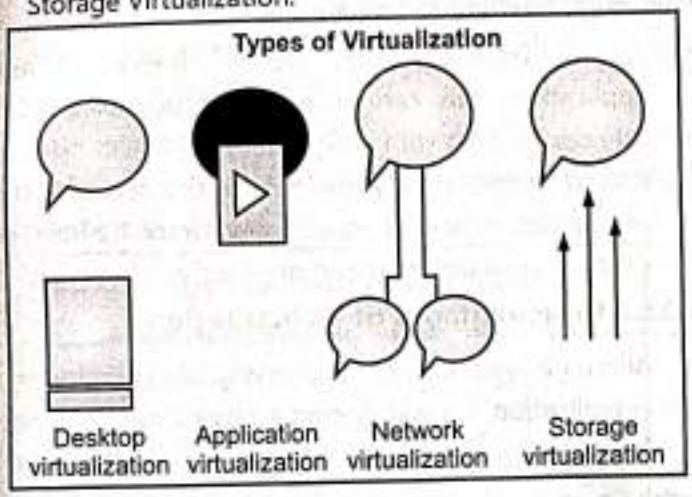


Fig. 1.19 : Types of virtualization

1. Application Virtualization:

- Application virtualization helps a user to have a remote access of an application from a server. The server stores all personal information and other characteristics of the application but can still run on a local workstation through internet. Example of this would be a user who needs to run two different versions of the same software. Technologies that use application virtualization are hosted applications and packaged applications.

2. Network Virtualization:

- The ability to run multiple virtual networks with each has a separate control and data plan. It co-exists together on top of one physical network. It can be managed by individual parties that potentially confidential to each other.
- Network virtualization provides a facility to create and provision virtual networks logical switches, routers, firewalls, load balancer, Virtual Private Network (VPN), and workload security within days or even in weeks.

3. Desktop Virtualization:

- Desktop virtualization allows the users' OS to be remotely stored on a server in the data center. It allows the user to access their desktop virtually, from any location by different machine. Users who wants specific operating systems other than Windows Server will need to have a virtual desktop. Main benefits of desktop virtualization are user mobility, portability, easy management of software installation, updates and patches.

4. Storage Virtualization:

- Storage virtualization is an array of servers that are managed by a virtual storage system. The servers aren't aware of exactly where their data is stored, and instead function more like worker bees in a hive. It makes managing storage from multiple sources to be managed and utilized as a single repository. storage virtualization software maintains smooth operations, consistent performance and a continuous suite of advanced functions despite changes, break down and differences in the underlying equipment.

1.11 PROS AND CONS OF VIRTUALIZATION

1.11.1 Advantages of Virtualization

Following are some of the most recognized advantages of Virtualization, which are explained in detail.

Using Virtualization for Efficient Hardware Utilization

- Virtualization decreases costs by reducing the need for physical hardware systems. Virtual machines use efficient hardware, which lowers the quantities of hardware, associated maintenance costs and reduces the power along with cooling the demand. You can allocate memory, space and CPU in just a second, making you more self-independent from hardware vendors.

Using Virtualization to Increase Availability

- Virtualization platforms offer a number of advanced features that are not found on physical servers, which increase uptime and availability. Although the vendor feature names may be different, they usually offer capabilities such as live migration, storage migration, fault tolerance, high availability and distributed resource scheduling. These technologies keep virtual machines chugging along or give them the ability to recover from unplanned outages.
- The ability to move a virtual machine from one server to another is perhaps one of the greatest single benefits of virtualization with far reaching uses. As the technology continues to mature to the point where can do long-distance migrations, such as being able

move a virtual machine from one data center to another no matter the network latency involved.

Disaster Recovery

- Disaster recovery is very easy when your servers are virtualized. With up-to-date snapshots of your virtual machines, you can quickly get back up and running. An organization can more easily create an affordable replication site. If a disaster strikes in the data center or server room itself, you can always move those virtual machines elsewhere into a cloud provider. Having that level of flexibility means your disaster recovery plan will be easier to enact and will have a 99% success rate.

Save Energy

- Moving physical servers to virtual machines and consolidating them onto far fewer physical servers' means lowering monthly power and cooling costs in the data center. It reduces carbon footprint and helps to clean up the air we breathe. Consumers want to see companies reducing their output of pollution and taking responsibility.

Deploying Servers too fast

- You can quickly clone an image, master template or existing virtual machine to get a server up and running within minutes. You do not have to fill out purchase orders, wait for shipping and receiving and then rack, stack, and cable a physical machine only to spend additional hours waiting for the operating system and applications to complete their installations. With virtual backup tools like Veeam, redeploying images will be so fast that your end users will hardly notice there was an issue.

Save Space in your Server Room or Datacenter

- Imagine a simple example: you have two racks with 30 physical servers and 4 switches. By virtualizing your servers, it will help you to reduce half the space used by the physical servers. The result can be two physical servers in a rack with one switch, where each physical server holds 15 virtualized servers.

Testing and Setting up Lab Environment

- While you are testing or installing something on your servers and it crashes, do not panic, as there is no data loss. Just revert to a previous snapshot and you can move forward as if the mistake did not even happen. You can also isolate these testing environments from end users while still keeping them online. When yo have completely done your work, deploy it in live.

Shifting all your Local Infrastructure to Cloud in a Day

- If you decide to shift your entire virtualize infrastructure into a cloud provider, you can do it in day. All the hypervisors offer you tools to export yo virtual servers.

Possibility to Divide Services

- If you have a single server, holding differen applications this can increase the possibility of th services to crash with each other and increasing the fa rate of the server. If you virtualize this server, you ca put applications in separated environments from eac other as we have discussed previously.

1.11.2 Disadvantages of Virtualization

- Although you cannot find many disadvantages fo virtualization, we will discuss a few prominent ones a follows –

Extra Costs

- Maybe you have to invest in the virtualization softwar and possibly additional hardware might be required t make the virtualization possible. This depends on you existing network. Many businesses have sufficien capacity to accommodate the virtualization withou requiring much cash. If you have an infrastructure tha is more than five years old, you have to consider a initial renewal budget.

Software Licensing

- This is becoming less of a problem as more softwar vendors adapt to the increased adoption o virtualization. However, it is important to check wit your vendors to understand how they view softwar use in a virtualized environment.

Learn the New Infrastructure

- Implementing and managing a virtualized environmen will require IT staff with expertise in virtualization. O the user side, a typical virtual environment will operate similarly to the non-virtual environment. There ar some applications that do not adapt well to th virtualized environment.

1.12 TECHNOLOGY EXAMPLES- VMWARE AND MICROSOFT HYPER-V.

- Server virtualization currently is a trending topic in the IT world. Its popularity and adoption keeps growing, especially in enterprise environments.

What Makes Virtualization Possible are Hypervisors?

- Server virtualization allows different operating systems running separate applications on one server while still using the same physical resources. These virtual machines make it possible for a system and network administrators to have a dedicated machine for every service they need to run.
- Not only does this reduce the number of physical servers required, but it also saves time when trying to pinpoint issues.

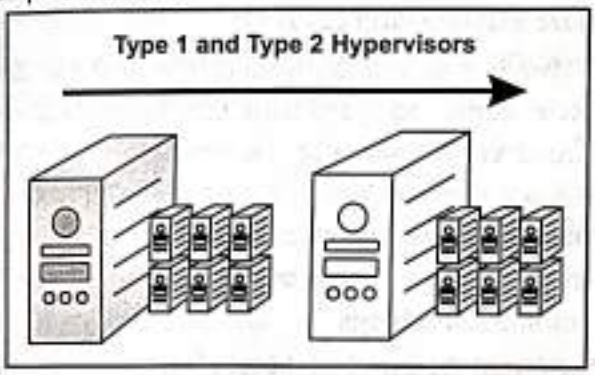


Fig. 1.20

What are Hypervisors?

- A hypervisor is a crucial piece of software that makes virtualization possible. It abstracts guest machines and the operating system they run on, from the actual hardware.
- Hypervisors create a virtualization layer that separates CPU / Processors, RAM and other physical resources from the virtual machines you create.
- The machine we install a hypervisor on is called a *host machine*, versus *guest virtual machines* that run on top of them.
- Hypervisors emulate available resources so that guest machines can use them. No matter what operating system you boot up with a virtual machine, it will think that actual physical hardware is at its disposal.
- From a VM's standpoint, there is no difference between the physical and virtualized environment. Guest machines do not know that the hypervisor created them in a virtual environment. Or that they are sharing available computing power. VMs run simultaneously with the hardware that powers them, and so they are entirely dependent on its stable operation.

Type 1 Hypervisor (also called bare metal or native)

Type 2 Hypervisor (also known as hosted hypervisors)

Type 1 Hypervisor

- A bare-metal hypervisor (Type 1) is a layer of software we install directly on top of a physical server and its underlying hardware.
- There is no software or any operating system in between, hence the name *bare-metal hypervisor*. A Type 1 hypervisor is proven in providing excellent performance and stability since it does not run inside Windows or any other operating system.
- Type 1 hypervisors are an OS themselves, a very basic one on top of which you can run virtual machines. The physical machine the hypervisor is running on serves virtualization purposes only. You cannot use it for anything else.
- Type 1 hypervisors are mainly found in enterprise environments.

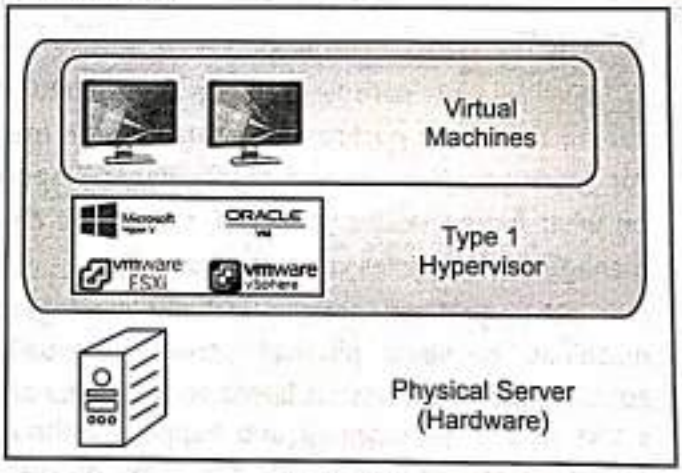


Fig. 1.21

Hypervisor Type 1 Performance

- Given that type 1 hypervisors are relatively simple, they do not offer many functionalities.
- Once you boot up a physical server with a bare-metal hypervisor installed, it displays a command prompt-like screen. If you connect a monitor to the server, what you get to see are some of the hardware and network details. This consists of the CPU type, the amount of memory, the IP address and the MAC address.
- Below is an example of a VMware type 1 hypervisor's screen after the server boots up.

- Another type 1 hypervisor may look quite different but they also only allow for simple server configuration. This consists of changing the date and time, IP address, password, etc. In order to create virtual instances, you need a management console set up on another machine. Using the console, you can connect to the hypervisor on the server, and manage your virtual environment.

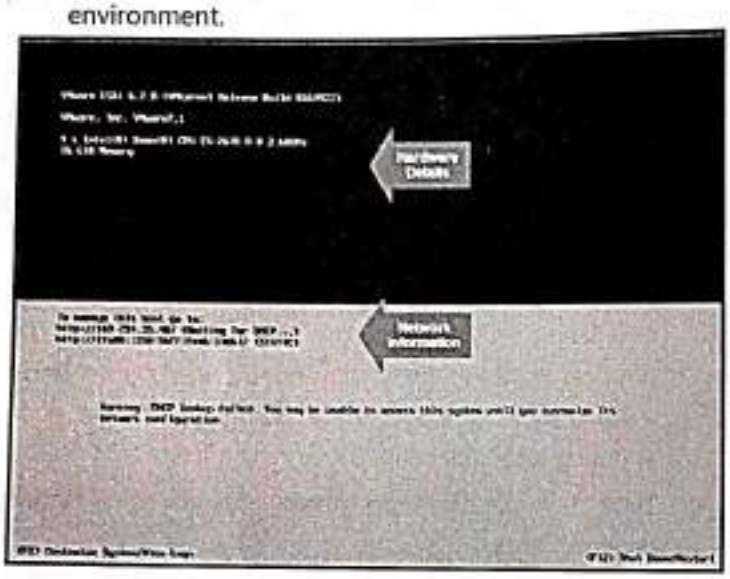


Fig. 1.22

- A management console can be web-based or a separate software package you install on the machine for which you want remote management. Depending on what functionalities you need, the license cost for management consoles varies substantially.
- One action you can perform includes moving virtual machines between physical servers, manually or automatically. This move is based on resource needs of a VM at a given moment and happens without any impact to the end-users. It's the same process if a piece of hardware or a whole server fails. Properly configured management software moves virtual machines to a working server as soon as an issue arises. The detection and restoration procedure takes place automatically and seamlessly.
- One of the best features of type 1 hypervisors is that they allow for over-allocation of physical resources.
- With type 1 hypervisors, you can assign more resources to your virtual machines than you have available. For example, if you have 128GB of RAM on your server and eight virtual machines, you can assign 24GB of RAM to each of them. This totals to 192GB of RAM, but VMs themselves will not actually consume all 24GB from the physical server. The VMs think they have 24GB when in reality they only use the amount of RAM they need to perform particular tasks.
- The hypervisor allocates only the amount of necessary resources for an instance to be fully functional. This is one of the reasons all modern enterprise data centers such as phoenixNAP, use type 1 hypervisors.

Type 1 Vendors

- There are many different hypervisor vendors available. Most provide trial periods to test out their services before you buy them.
- The licensing costs can be high if you want all the bells and whistles they have on offer.

These are the most common type 1 hypervisors:

VMware vSphere with ESX/ESXi

- VMware is an industry-leading vendor of virtualization technology, and many large data centers run on their products. It may not be the most cost-effective solution for smaller IT environments. If you do not need all the advanced features VMware vSphere offers, there is a free version of this hypervisor and multiple commercial editions.

KVM (Kernel-Based Virtual Machine)

- KVM is built into Linux as an added functionality. It lets you convert the Linux kernel into a hypervisor. It is sometimes confused with a type 2 hypervisor (see definition below). It has direct access to hardware along with virtual machines it hosts. KVM is an open-source hypervisor that contains all the features of Linux with the addition of many other functionalities. This makes it one of the top choices for enterprise environments. Some of the highlights include live migration, scheduling and resource control, alongside higher prioritization.

Microsoft Hyper-V

- Despite VMware's hypervisor being higher on the ladder with its numerous advanced features, Microsoft's Hyper-V has become a worthy opponent. Microsoft also offers a free edition of their hypervisor, but if you want a GUI and additional functionalities, you will have to go for one of the commercial versions. Hyper-V may not offer as many features as VMware's vSphere package, but you still get live migration, replication of virtual machines, dynamic memory and many other features.

Oracle VM

- This hypervisor has open-source Xen at its core and is free. Advanced features are only available in paid versions. Even though Oracle VM is essentially a stable product, it is not as robust as vSphere, KVM or Hyper-V.

Citrix Hypervisor (formerly known as Xen Server)

- This Server virtualization platform by Citrix is best suited for enterprise environments. It can handle all types of workloads and provides features for the most demanding tasks. Citrix is proud of its proprietary features, such as Intel and NVIDIA enhanced virtualized graphics and workload security with Direct Inspect APIs.

Type 2 Hypervisor

- This type of hypervisor runs inside of an operating system of a physical host machine.
- This is why we call type 2 hypervisors hosted hypervisors. As opposed to type 1 hypervisors that run directly on the hardware, hosted hypervisors have one software layer underneath. In this case we have:
  - ➢ A physical machine.
  - ➢ An operating system installed on the hardware (Windows, Linux, macOS).
  - ➢ A type 2 hypervisor software within that operating system.
  - ➢ The actual instances of guest virtual machines.

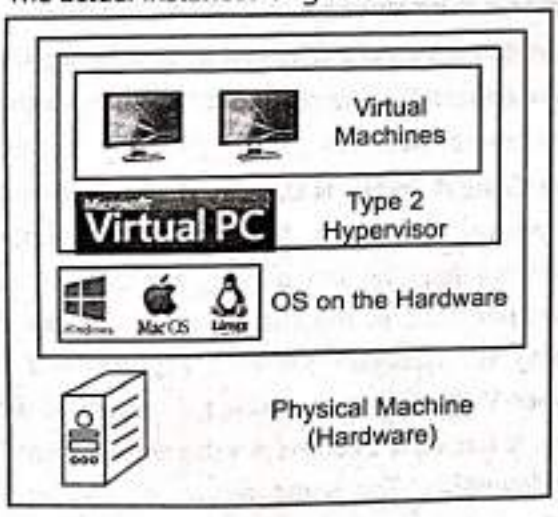


Fig. 1.23

- Type 2 hypervisors are typically found in environments with a small number of servers.
- What makes them convenient is that you do not need a management console on another machine to set up and manage virtual machines. You can do all of this on the server where you install the hypervisor. They are not any different from the other applications you have in your operating system.
- When you launch a virtual machine, you get another window to perform all tasks.

Hypervisor Type 2 Performance

- Hosted hypervisors essentially also act as management consoles for virtual machines, you can perform any task using the built-in functionalities.
- There is no need to install separate software on another machine to create and maintain your virtual environment. You simply install and run a type 2 hypervisor as you would any other application within your OS. With it, you can create snapshots or clone your virtual machines, import or export appliances, etc.
- Here is one example of a type 2 hypervisor interface (VirtualBox by Oracle):

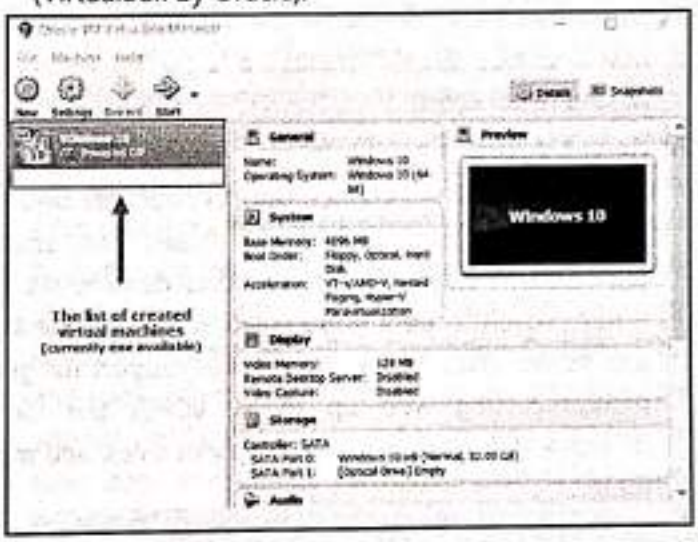


Fig. 1.24

- You do need to be careful when allocating actual resources with this type of hypervisor.
- Bare-metal hypervisors can dynamically allocate available resources depending on the current needs of a particular VM. A type 2 hypervisor occupies whatever you allocate to a virtual machine.
- When you assign 8GB of RAM to a VM, that amount will be taken up even if the VM is using only a fraction of it. If the host machine has 32GB of RAM and you create three VMs with 8GB each, you are left with 8GB of RAM to keep the physical machine running. Creating another VM with 8GB of ram would bring down your system. This is critical to keep in mind, so as to avoid

over-allocating resources and crashing the host machine.

- Type 2 hypervisors are convenient for testing new software and research projects.
- It is possible to use one physical machine to run multiple instances with different operating systems to test how an application behaves in each environment or to create a specific network environment. You only need to make sure that there are enough physical resources to keep both the host and the virtual machines running.

Type 2 Vendors

- As is the case with bare-metal hypervisors, you can choose between numerous vendors and products. Conveniently, many type 2 hypervisors are free in their basic versions and provide sufficient functionalities.
- Some even provide advanced features and performance boosts when you install add-on packages, free of charge. We will mention a few of the most used hosted hypervisors:

Oracle VM VirtualBox

- A free but stable product with enough features for personal use and most use cases for smaller businesses. VirtualBox is not resource demanding, and it has proven to be a good solution for both desktop and server virtualization. It provides support for guest multiprocessing with up to 32 vCPUs per virtual machine, PXE Network boot, snapshot trees, and much more.

VMware Workstation Pro/VMware Fusion

- VMware Workstation Pro is a type 2 hypervisor for Windows OS. It is full of advanced features and has seamless integration with vSphere. This allows you to move your apps between desktop and cloud environments.
- It does come with a price tag, as there is no free version. If you want to take a glimpse into VMware hosted hypervisors free of charge, you can try VMware Workstation Player. This is the basic version of the hypervisor suitable for small sandbox environments.
- For MacOS users, VMware has developed Fusion that is similar to their Workstation product. It comes with somewhat fewer features, but also carries a small price tag.

Windows Virtual PC

- It only supports Windows 7 as a host machine an Windows OS on guest machines. This includes multipl versions of Windows 7 and Vista, as well as XP Sp Virtual PC is completely free.

Parallels Desktop

- A competitor to VMware Fusion. It is primari intended for MacOS users and offers plenty of feature depending on the version you purchase. Some of th features are network conditioning, integration wit Chef/Ohai/Docker/Vagrant, support for up to 128G per VM, etc.

Type 1 vs. Type 2 Hypervisor

- **Choosing the Right Type of Hypervisor** strict depends on your individual needs.
- The first thing you need to keep in mind is the size the virtual environment you intend to run.
- For personal use and smaller deployments, you can g for one of the type 2 hypervisors. If budget is not a issue, VMware will provide every feature you nee Otherwise, Oracle VM VirtualBox is a hypervisor tha will provide most of the functionalities general needed.

Enterprise Environments

- Even though type 1 hypervisors are the way to go, yo do need to take into consideration many factors befor making a decision.
- **The Critical Factor is Usually the Licensing Cost.** Th is where you need to pay extra attention sinc licensing may be per server, per CPU or sometime even per core. In the current market, there is a batt going on between VMware vSphere and Microso Hyper-V. While Hyper-V was falling behind a few year ago, it has now become a valid choice, even for large deployments. The same argument can be made fo KVM.
- Many vendors offer multiple products and layers o licenses to accommodate any organization. You ma want to create a list of the requirements. Such as, ho many VMs you need, maximum allowed resources pe VM, nodes per cluster, specific functionalities, and the check which of these products best fits your needs.

1.12.1 Comparing VMware and Hyper-V

- Virtualization has long since become a significant part of modern-day businesses. Popularity of virtualization technology is attributable to the long list of benefits it provides, including flexibility, cost-efficiency, on-demand scalability, reliability, and portability. Essentially, hardware virtualization is made possible with the use of virtualization platforms, which help manage workloads in a virtual environment. Currently, there are two major players who dominate the virtualization market – Hyper-V vs VMware.

What Is Hyper-V?

- Hyper-V is a type-1, native hypervisor designed by Microsoft, which allows creating and managing multiple virtualized versions of a computer, called virtual machines (VMs). Hyper-V provides a virtualization platform on which you can build IT infrastructure of any level of complexity. Each VM, in this case, is running in its own isolated space, without interfering with the processes in other VMs.

What Is VMware?

- VMware vSphere is a server virtualization platform created by VMware. Essentially, vSphere encompasses a set of virtualization products, which include the ESXi hypervisor, vSphere Client, VMware Workstation, vCenter, and others. All of these products combined constitute the VMware infrastructure, which enables centralized management of the created virtual environment.

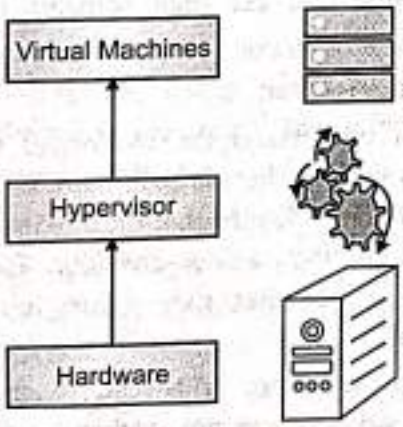


Fig. 1.25

Comparing Hyper-V vs VMware

- Hyper-V and VMware have a set of unique features which differentiate them from one another. Moreover, each virtualization platform entails a number of advantages and disadvantages to consider when deciding between the two. The following excerpt will outline the main components that both platforms offer as well as the differences and similarities between Hyper-V and VMware.

Architecture

Hyper-V vs. VMware vSphere

- Microsoft Hyper-V exists in two modes. The two variants are fairly similar in structure and perform the same functions:
  - ➢ As the Hyper-V role, which is an in-built Windows Server feature that can be enabled by a server administrator.
  - ➢ As a stand-alone Microsoft product (also known as Hyper-V Server), with limited functionality and Hyper-V management components.
- The architecture of Hyper-V is based upon micro-kernelized hypervisors, meaning that a host server (also called the parent partition) provides direct access to hardware and computing resources (virtualization stack). Hyper-V allows you to isolate VMs into logical units (partitions), including Operating Systems (OSes) and applications. Partitions are divided into the parent and the child partitions. Each Hyper-V environment can have only one parent partition, which should run a supported version of Windows Server.
- The parent partition can create multiple child partitions for hosting guest OSes. Child partitions can't directly access hardware resources but they can present a virtual view of the resources in the form of virtual devices. The communication between the parent and the child partitions is conducted via the VMBus, which lets you manage the requests directed to virtual devices. The parent partition also includes a Virtualization Service Provider (VSP), which enables the connection to the VMBus for managing device access requests from child partitions.
- Hyper-V can host two types of OSes in child partitions: enlightened and un-enlightened. The main difference is that the enlightened child partition has Hyper-V integration components and a Virtualization Service

Client (VSC), which lets you avoid device emulation and enable direct communication with the virtualization layer. At the same time, the unenlightened child partition does not have the same components and simply emulates software.

- VMware vSphere is a virtualization platform consisting of multiple components which need to be installed and set up. Essentially, vSphere is a suite of virtualization products, which, when combined, allow you to build a computing platform. At the core of VMware vSphere lies VMware ESXi, which is a type-1, native hypervisor used to directly manage host servers and run multiple guest VMs. VMware ESXi enables direct access to the physical machine's computing resources, which are shared by the VMs running in the system.
- The earlier version of VMware ESXi , VMware ESX ran on a Linux kernel, which acted as the primary VM. Later, ESXi hypervisor was introduced to minimize the overhead imposed by VMware ESX. VMware ESXi requires a minimum amount of hardware resources and enables a small footprint of 70 MB. High performance of ESXi is ensured by VMkernel, which forms the basis of the virtualization process. VMkernel runs directly on hosts and provides connection between VMs and the physical hardware.
- In order to manage the VMware virtualization platform, other components of vSphere, such as VMware vSphere Client or VMware vCenter Server, are provided. They function as the management tools used for running ESXi hosts.

1.13 BEFORE THE MOVE INTO THE CLOUD

Know Your Software Licenses, The Shift to a Cloud Cost Model, Service Levels for Cloud Applications.

1.13.1 Know your Software Licenses

- The cloud delivery models that present the most software-licensing challenges are Infrastructure as a Service (IaaS) and Platform as a Service (PaaS). Software as a Service (SaaS) is less likely to cause problems because, as the name suggests, the software is part of the cloud provider's services. With IaaS and PaaS, though, the customer has shared control over what is run in the cloud environment, including third-party software. In the case of IaaS, the customer does not manage or control the underlying cloud infrastructure but may have control over operating systems and deployed applications. With PaaS, wh the customer typically doesn't have control over t operating system, it may have control over th deployed applications.
- Where the complexity comes in is that softwa manufacturers are all over the map in how the address cloud use in their software licenses. Some ba their licensing on the number of users, and those use in turn may be named or concurrent. Others charg per processor or core that the software runs on. St others look at actual usage, a metric that is distin from number of users. The one thing that these vario licensing models have in common is that they a attempts to maximize revenue, and naturally, softwa makers view the use of their products in the cloud a an expansion of licensing rights that represents a opportunity for increased revenue.
- Can the customer argue that the cloud does no represent an expansion of licensing rights? It would b difficult. If the customer acquired its software license from the vendor under a long-standing agreemen chances are good that the agreement pre-dates th inception of cloud computing. Of course, contract generally do not address technology offerings tha don't exist at the time of the contract's drafting, so a pre-cloud software-licensing contract is highly unlike to contemplate the use of those licenses in a clou environment. Legally, any rights that aren't explicitl stated as being granted to the customer in the licens agreement are retained by the software manufacture In cases like this, customers do not have any pre existing rights to use their software licenses in th cloud.

Parsing the Clauses

- To better understand the challenges that the clou brings to software licensing, it might be helpful to tak a look at some clauses that one might see in a clou vendor's contract. Below are four, followed by m explanation of what they mean and why they pertinent.
- Customer authorizes [VENDOR] to copy, install an modify, when necessary and as required by th Agreement, all third-party software to be used in th Services.

What this Means: As part of providing the service, th cloud vendor may need to access the software in order t

reate redundant systems, and potentially to replicate or estore the customer environment in the event of an nplanned outage or other disaster. The above language ays that the customer gives the cloud vendor permission o do these things on its behalf.

Customer warrants to [VENDOR] that it has obtained any licenses or approvals required to give [VENDOR] such rights or licenses to access, copy, distribute, use and/or modify or install any third-party software to be used in the Services.

Vhat this Means: This affirms that the customer's license greement with the software manufacturer includes the ghts for the cloud vendor to access the software in the anner described above.

Some third-party software manufacturers' contract terms and conditions may become void if [VENDOR] provides services for or works on the software (such as providing maintenance services). [VENDOR] does not take responsibility for third-party warranties or for any effect that the [VENDOR'S] services may have on those warranties.

Vhat this Means: The cloud vendor is saying that if its use f the software in providing the services causes any oncompliance with the terms of the software-license greement, then the cloud vendor is not responsible for ny adverse consequences.

Third-party software shall be exclusively subject to the terms and conditions between the third-party software provider and Customer. [VENDOR] shall have no liability for third-party software.

Vhat this Means: The cloud vendor is saying that it has o responsibility regarding the effective functioning of the oftware, or any adverse impacts of any malfunctioning of e software.

ll this adds up to the fact that you need to clearly identify our license rights and usage needs before deploying ird-party software in the cloud, then effectively capture ose in your contract with the cloud vendor.

13.2 The Shift to a Cloud Cost Model

Cloud migration is the process of moving data, applications or other business elements to a cloud computing environment.

There are various types of cloud migrations an enterprise can perform. One common model is the transfer of data and applications from a local, on-premises data center to the public cloud. However, a cloud migration could also entail moving data and applications from one cloud platform or provider to another a model known as cloud-to-cloud migration. A third type of migration is a reverse cloud migration, cloud repatriation or cloud exit, where data or applications are moved off of the cloud and back to a local data center.

Benefits of Cloud Migration

- The general goal or benefit of any cloud migration is to host applications and data in the most effective IT environment possible, based on factors such as cost, performance and security.
- For example, many organizations perform the migration of on-premises applications and data from their local data center to public cloud infrastructure to take advantage of benefits such as greater elasticity, self-service provisioning, redundancy and a flexible, pay-per-use model.

Cloud Migration Strategies

- Moving workloads to the cloud requires a well-thought-out strategy that includes a complex combination of management and technology challenges as well as staff and resource realignment. There are choices in the type of migration to perform as well as the type of data that should move. It's important to consider the following cloud migration steps before taking action.

Applications:

- First, identify the application. Every company has a different reason to move a workload to the cloud, and goals for each organization will vary. Next steps are to figure out how much data needs to be moved, how quickly the work needs to be done and how to migrate that data. Take inventory of data and applications, look for dependencies and consider one of the many migration options.
- Remember that not every application should leave the enterprise data center. Among those that should stay are applications that are business-critical, have high throughput, require low latency or are applications that have strict geographic stewardship requirements such as GDPR that may be cause for concern.
- Consider your costs. An organization may have a bundle invested in hardware infrastructure and software licensing. If that investment is steep, it is worth weighing whether or not it's worth it to migrate the workload.

When to put workloads in the public cloud

Public cloud computing adoption continues to grow because of it various benefits, including decreased costs. In

this video, Steve Bigelow digs into these topic and how a cloud migration can benefit your enterprise.

Play
Mute
Current Time 0:00
/
Duration 5:14
Loaded: 3.13%

Picture-in-PictureFullscreen

Cloud Migration Types

- The next step is to identify the right cloud environment. Enterprises today have more than one cloud scenario from which to choose.
- The public cloud lets many users access compute resources through the internet or dedicated connections. A private cloud keeps data within the data center and uses a proprietary architecture. The hybrid cloud model mixes public and private cloud models and transfers data between the two. Finally, in a multi-cloud scenario, a business uses IaaS options from more than one public cloud provider.
- As you consider where the application should live, also consider how well it will perform once it's migrated. Be sure there is adequate bandwidth for optimal application performance. And investigate whether an application's dependencies may complicate a migration.

Cloud migration strategy checklist

- **Assess** the on-premises infrastructure and application fleet.
- **Map** application, network and data dependencies and topology.
- **Select** applications most suited for cloud migration, and assess cloud service options to map selected applications to the optimal choice of IaaS, PaaS or SaaS
- **Develop** a migration plan and process flow to account for important details, such as data replication, account logins and connections to dependent apps.
- **Create** the appropriate cloud hosting environment for an IaaS or PaaS legacy migration.
- **Replicate** application images and dependencies.
- **Stage** and test applications in a pilot environment. Migrage beta users to simulate real-world conditions.
- **Check** the final pilot environment for security and regulatory compliance before cutting over. Finally harden security and optimize performance as necessary.

Fig. 1.26

- Consider your options with this cloud migratio checklist.
- Now is a good time to review what's in the stack of th application that will make the move. Local applicatio may contain a lot of features that go unused, and it wasteful to pay to migrate and support thos nonessential items. Stale data is another concern wit cloud migration. Without a good reason, it's probab unwise to move historical data to the cloud.
- As you examine the application, it may be prudent t reconsider its strategic architecture to set it up fo what could potentially be a longer life. A handful o platforms are now mainstream among hybrid an multi-cloud environments, including the following:
  - ➢ Microsoft Azure Stack;
  - ➢ Google Cloud Anthos;
  - ➢ AWS Outposts;
  - ➢ VMware Cloud on AWS; and

- A container-based PaaS, such as Cloud Foundry or Red Hat OpenShift.

Staff Issues : Applications that live in the cloud require a different set of management skills, and, as such, IT leaders will need to ensure staffs are ready to handle a cloud migration. Consider employee skill sets, and make sure everyone is properly trained on how to control and manage those services. Cloud management is unlike working with local data centers and routine virtualized resources.

Regardless of the application, current staff must learn to adapt to new roles. In particular, data security requires a different approach in the cloud than on premises, so staff training will need to be a priority.

Cloud Migration Process

The steps or processes an enterprise follows during a cloud migration vary based on factors such as the type of migration it wants to perform and the specific resources it wants to move. That said, common elements of a cloud migration strategy include the following:

- Evaluation of performance and security requirements;
- Selection of a cloud provider;
- Calculation of costs; and
- Any reorganization deemed necessary.

At the same time, be prepared to address several common challenges during a cloud migration:

- Interoperability;
- Data and application portability;
- Data integrity and security; and
- Business continuity.

Without proper planning, a migration could degrade workload performance and lead to higher IT costs thereby negating some of the main benefits of cloud computing.

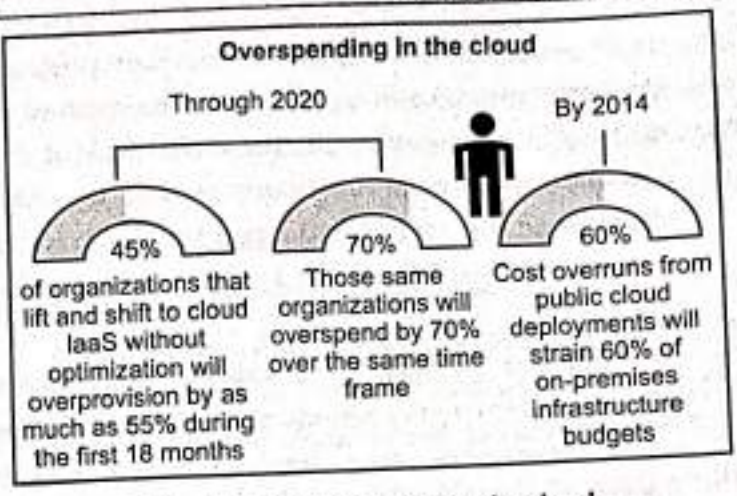


Fig. 1.27 : Overspending in the cloud

Don't Overspend in Cloud.

- Depending on the details of the migration, an enterprise may choose to move an application to its new hosting environment without any modifications a model sometimes referred to as a lift-and-shift migration. In this scenario, a workload moves directly from local servers to the cloud without any changes. This is essentially a 1-to-1 move done primarily as a short-term fix to save on infrastructure costs.
- In other cases, it might be more beneficial to change an application's code or architecture. This process is known as application refactoring or rearchitecting. It makes sense to refactor an application in advance of a cloud migration, but often it happens retroactively. This typically occurs once it is clear that a lift and shift has reduced an application's performance.
- Refactoring an application can be costly, so IT management should consider whether this makes financial sense. Don't forget to calculate cost, performance and security when you analyze your ROI. It's likely that an application will require at least some refactoring, whether the transformation is minimal or comprehensive.
- If a migration is done online, you will need to calculate how much bandwidth is necessary to make the move.
- Enterprises have several choices when it comes to transferring data from a local data center to the public cloud. These include the use of the public internet or private/dedicated network connection. Another option is an offline transfer, in which an organization upload

its local data onto an appliance and then physically ships that appliance to a public cloud provider, which then uploads the data to the cloud. The type of data migration an enterprise chooses online or offline depends on the amount and type of data it wants to move, as well as how fast it needs to complete the migration.

- It might not be realistic to sideline your internet connection for lengthy periods of time. In some cases, it will make more sense simply to use a truck to transfer data instead of an internet connection. There are services for this purpose Microsoft, AWS, Google and IBM have options for offline data shipping. Physical shipment may not eliminate the need for additional syncing, but it can cut time and expense to move the data.
- Before the workload moves to production, it should be stress tested and optimized to deliver acceptable performance. It's also important to test failure conditions as well as redundant systems.
- Once the cloud migration is complete, staff will shift its focus to data performance, usage and stability. Be sure to budget for these tools, as they are often forgotten in the initial planning.
- Here's where IT staff sees the biggest change in their support role. There is some reduction in overall hardware support. But cloud workloads must be managed, so it makes sense to add some cloud management training classes for the team. There may be some special considerations for the new security realities during a migration.
- Ensuring application security in the cloud is always a concern, particularly during a live migration to the cloud. VM migrations are essential to balance a workload's need for compute, storage and other application demands.
- Live migration through a network makes possible various types of attacks. An attacker can take a VM snapshot and create a VM in a different context than its original intent. Those stolen credentials can duplicate and steal the snapshot or install root kits or other malware for additional access. Thrashing is a persistent denial-of-service attack in which hacke force repeated migrations and disrupt computi processes by consuming system resources.

Cloud Migration Challenges

- Sometimes IT leaders discover that their application don't work as well in the cloud as they did o premises. They need to determine the reasons for th cloud migration failure. It could be poor latenc concerns about security or perhaps compliance challenges. Often, the reason is that the clou application has a higher cost than anticipated, or th application does not work as well as original anticipated.
- The reality is that not all applications are a good fit f the cloud. That's why it is so important for managers t scrutinize their on-premises applications careful when they make their initial choice about which shoul move to a cloud environment.
- Having a solid cloud exit strategy, where the apps an data move out of the cloud, is just as important a having a good cloud migration strategy. IT manager need to know where the data will go, how to manag the technical transition and then how to address an business or legal issues that may arise as a result of th reverse migration.
- Consider the changes you made prior to when yo moved the application to the cloud. Moving the ap back to its original platform might be one option. those older platforms no longer exist, it will mak sense to keep the application in the cloud until you ca come up with an alternate solution.
- The application does not need to return to its origina state or to a private cloud. If the application require additional infrastructure, a move to data centr virtualization might be the right choice.
- The most common alteration to an app that goes t the cloud tends to be horizontal scaling, the featur that lets applications access additional resources o the cloud when they need additional capacity or le users move apps to the cloud when needed. If yo make no provisions to retain that feature, be awa

that you will lose those benefits when you remove the app from the cloud.

- Cloud cost calculators and estimation tools help enterprises determine the cost of a cloud configuration before the team makes the migration.
- A thorough cloud exit strategy should include application lifecycle management. Test the apps in the new environment before the cutover. Last, double-check potential business and legal processes. Exiting the cloud is usually a last resort, so be sure your legal department checks your contract with the cloud provider, particularly if you transition before the contract expires.
- Since AWS is in such widespread use, it's worth noting some of the common mistakes made by AWS cloud administrators.
- Setting up the wrong instance type is common, as AWS provides several classes of instance types. You need to select the right amount of CPU and memory resources, as well as enough network connectivity for both your Amazon Elastic Block Storage (EBS) and app data transmission.
- Don't underestimate proper staff training. If staff cannot be trained prior to a cloud migration, it makes sense to hire an experienced AWS partner to manage the project.

Types of Cloud Migration Tools and Services

- Workload management undergoes a significant change when the application moves to the cloud. Cloud costs are often higher than originally anticipated. IT staff needs to change their management processes to work as well in the cloud as they do locally. This can be achieved by any number of services and tools.
- Tools like Microsoft Azure Advisor and AWS Trusted Advisor can offer optimization recommendations in a cloud environment in areas such as cost, performance and security.
- Cloud cost calculators and estimation tools help enterprises determine the cost of a cloud configuration before the team makes the migration. If you're an AWS shop, consider using the Simple Monthly Calculator for more detailed pricing estimates.
- Services are available to help users tackle the various phases of the cloud migration process, which can be broken into four steps: migration planning, data migration, server migration and database migration.
- Consider the following services as they apply to the aforementioned categories:
  - Data transfer services;
  - Network transfers; and
  - Cloud migration services and appliances.
- The big IaaS providers, AWS, Microsoft and Google, offer free-trial tiers, or low-cost cloud migration services. Here are a few of the most popular.
- Azure Migrate helps users assess how their VMware workloads would perform in an Azure public cloud before an actual migration takes place. Another tool, Azure Site Recovery, helps IT pros make the move. Customers can use the service to move VMs to Azure as well as take VMs on Azure and shift them to different cloud regions.
- Among AWS cloud migration services is AWS Migration Hub, which helps users monitor the progress of their app migration. It displays the status of all resources involved across every AWS migration in the user's portfolio. Another, AWS Application Discovery Service, maps out the planning stages of an app migration. It uses the data to provide insights about configuration, data utilization, dependencies, memory and resource usage.
- Google also has a host of cloud migration tools. Google Cloud Storage Transfer Service is used to move data into Google Cloud Storage. IT teams also use it to back up data and move it from one cloud storage entity to another. Google Transfer Appliance is an offline migration service for large data transfers.

A Breakdown of Cloud Migration Services

The following table lists migration services from AWS, Microsoft Azure and Google Cloud across five categories.

	AWS	AZURE	Google Cloud
Migration planning	• Application discovery service • Migration hub • TSO Logic • Cloud adoption readiness tool	• Azure migrate • Cloud adoption framework • App service migration assistant.	• Cloud adoption framework • Cloud maturity assessment
Bulk data migration	• Snowball • Snowball edge • Snowmobile	• Data box • Data box disk • Data box heavy	• Transfer appliance
Data migration over a network	• DataSync • Transfer for secure file transfer protocol • Storage gateway	• Azure stack edge • Data box gateway	• Cloud online data transfer • Storage transfer service
Server migration	• Server migration service • Cloud-Endure migration	• Site recovery	• Migrate for compute Engine (formerly velostrata) • Migrate for Anthos
Database migration	• Database migration service • Schema conversion tool	• Database migration service	• Bigquery data transfer service

- Pick the right cloud migration service that best fi your needs.
- There are a few automation options for lift-and-shi migrations, but most important is to understand ap performance and resource requirements prior to th move. The migration of composite apps that rely o databases can be partially automated, but users wi have to manually fix problems that may arise.

Why Migrate to the Cloud?

- Cloud computing ultimately frees an enterprise IT tea from the burden of managing uptime. Placing a application in the cloud is often the most logical ste for growth. A positive answer to some or all of thes questions may indicate your company's readiness t move an app to the cloud.

Should your Application Stay or Go?

- Legacy applications, or workloads that require lo latency or higher security and control, probably shoul stay on premises or move to a private cloud.

What's the Cost to Run an Application in the Cloud?

- One of the primary benefits of a cloud migration i workload flexibility. If a workload suddenly needs mor resources to maintain performance, its cost to run ma escalate quickly.

Which Cloud Model Fits Best?

- Public cloud provides scalability through a pay-pe usage model. Private cloud or on-premises provide extra control and security. A hybrid cloud mode provides the best of both, although performance an connectivity may suffer.

How do I Choose the Right Cloud Provider?

- The top three cloud providers AWS, Microsoft an Google generally offer comparable services to run a kinds of workloads in the cloud, as well as tools to hel you efficiently move apps there. Gauge your specif

needs for availability, support, security and compliance, and pricing to find the best fit.

1.13.3 The Service Levels for Cloud Applications

- A Service Level Agreement (SLA) is the bond for performance negotiated between the cloud services provider and the client. Earlier, in cloud computing all Service Level Agreements were negotiated between a client and the service consumer. Nowadays, with the initiation of large utility-like cloud computing providers, most Service Level Agreements are standardized until a client becomes a large consumer of cloud services. Service level agreements are also defined at different levels which are mentioned below:
  - ➢ Customer-based SLA
  - ➢ Service-based SLA
  - ➢ Multilevel SLA
- Few Service Level Agreements are enforceable as contracts, but mostly are agreements or contracts which are more along the lines of an Operating Level Agreement (OLA) and may not have the restriction of law. It is fine to have an attorney review the documents before making a major agreement to the cloud service provider. Service Level Agreements usually specify some parameters which are mentioned below:
  - ➢ Availability of the Service (uptime)
  - ➢ Latency or the response time
  - ➢ Service components reliability
  - ➢ Each party accountability
  - ➢ Warranties
- In any case, if a cloud service provider fails to meet the stated targets of minimums then the provider has to pay the penalty to the cloud service consumer as per the agreement. So, Service Level Agreements are like insurance policies in which the corporation has to pay as per the agreements if any casualty occurs.
- Microsoft publishes the Service Level Agreements linked with the Windows Azure Platform components, which is demonstrative of industry practice for cloud service vendors. Each individual component has its own Service Level Agreements. Below are two major Service Level Agreements (SLA) described:

1. Windows Azure SLA :

- Window Azure has different SLA's for compute and storage. For compute, there is a guarantee that when a client deploys two or more role instances in separate fault and upgrade domains, client's internet facing roles will have external connectivity minimum 99.95% of the time. Moreover, all of the role instances of the client are monitored and there is guarantee of detection 99.9% of the time when a role instance's process is not runs and initiates properly.

2. SQL Azure SLA :

- SQL Azure clients will have connectivity between the database and internet gateway of SQL Azure. SQL Azure will handle a "Monthly Availability" of 99.9% within a month. Monthly Availability Proportion for a particular tenant database is the ratio of the time the database was available to customers to the total time in a month. Time is measured in some intervals of minutes in a 30-day monthly cycle. Availability is always remunerated for a complete month. A portion of time is marked as unavailable if the customer's attempts to connect to a database are denied by the SQL Azure gateway.
- Service Level Agreements are based on the usage model. Frequently, cloud providers charge their pay-as-per-use resources at a premium and deploy standards Service Level Agreements only for that purpose. Clients can also subscribe at different levels.
- That guarantees access to a particular amount of purchased resources. The Service Level Agreements (SLAs) attached to a subscription many times offer various terms and conditions. If client requires access to a particular level of resources, then the client need to subscribe to a service. A usage model may not deliver that level of access under peak load condition.

2.1 INTRODUCTION TO CLOUD COMPUTING ARCHITECTURE

- The Cloud Computing Architecture is a broad and comprehensive modern concept, which includes the possibility to use the cloud to store large amounts of various data and applications, and providing them on demand, it is also the use of storage internet applications, as for example e-mails, it is the seamless access to powerful hardware, servers, storage and software technologies offered by datacenters without embedding significant investment to own infrastructure, software and hardware.
- The clouds are classified by location and by offered services. By location they can be:
  - ➢ **Private :** cloud which is build and exclusively used by a single organization.
  - ➢ **Public :** cloud hosted by cloud service providers.
  - ➢ **Hybrid :** combines both public and private cloud models.
- As for the offered services, the clouds can be:
  - ➢ **Infrastructure as a Service (IaaS) :** which offer the storage and database hosting;
  - ➢ **Platform as a Service (PaaS) :** which offer a development platform;
  - ➢ **Software as a Service (SaaS) :** which offer a complete ready-to-use application.

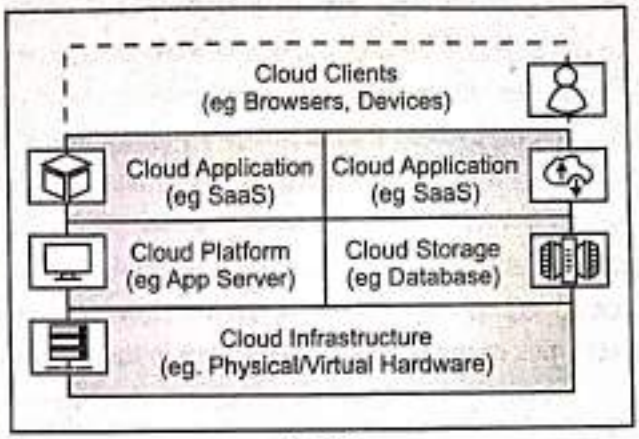


Fig. 2.1

Example: Introduction to Cloud Computing Architecture

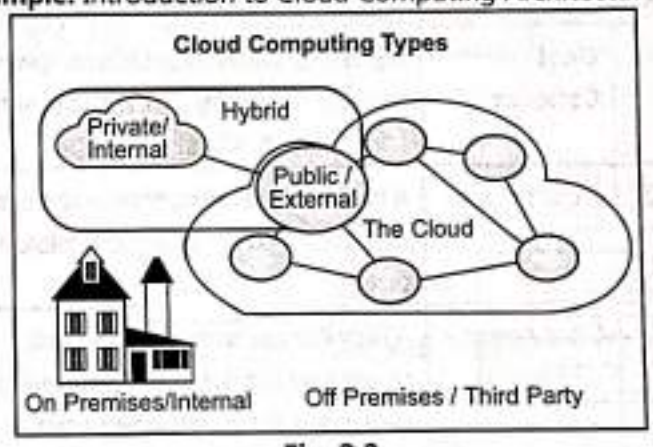


Fig. 2.2

Example : Cloud Computing Types

2.2 CLOUD REFERENCE MODEL

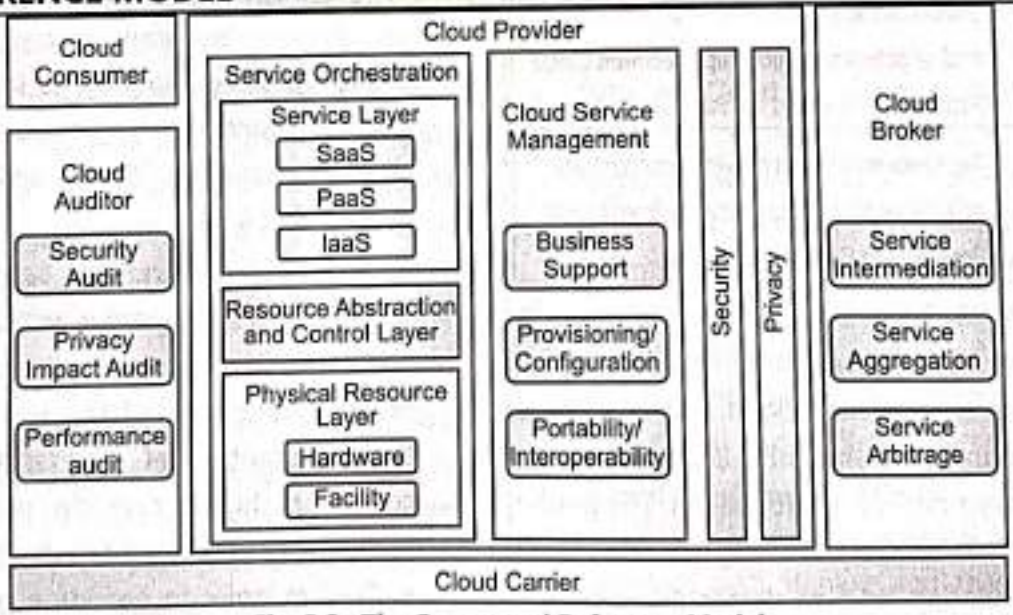


Fig. 2.3 : The Conceptual Reference Model

- Cloud computing conceptual reference model identifies the major actors, their activities and functions in cloud computing.
- Fig. 2.3 presents an overview of the NIST cloud reference architecture.
- As shown in Fig. 2.3, the NIST cloud computing reference architecture defines five major actors: cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker. Each actor is an entity (a person or an organization) that participates in a transaction or process and/or performs tasks in cloud computing.
- Table 2.1 briefly lists the actors defined in the NIST cloud computing reference architecture. The general activities of the actors are discussed in further tutorial.

Table 2.1 : Actors in Cloud Computing

Sr. No	Actor	Definition
1.	Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, Cloud Providers.
2.	Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
3.	Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
4.	Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.
5.	Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.

- Fig. 2.4 illustrates the interactions among the actors. A cloud consumer may request cloud services from a cloud provider directly or via a cloud broker. A cloud auditor conducts independent audits and may contact the others to collect necessary information. The details will be discussed in the following sections and presented in increasing level of details in successive diagrams.

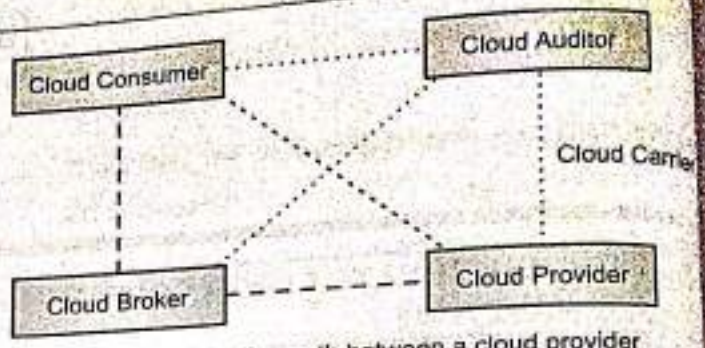


Fig. 2.4 : Interactions between the Actors in Cloud Computin

- **Usage Scenario 1:** A cloud consumer may reque service from a cloud broker instead of contacting cloud provider directly. The cloud broker may create new service by combining multiple services or e enhancing an existing service. In this example, th actual cloud providers are invisible to the clou consumer and the cloud consumer interacts direct with the cloud broker.

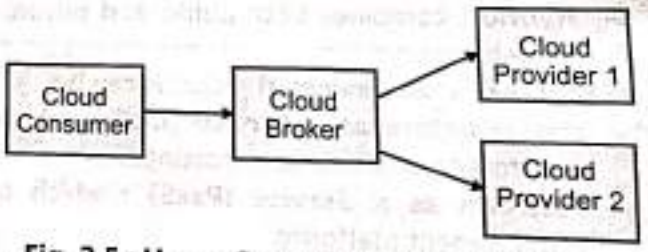


Fig. 2.5 : Usage Scenario for Cloud Brokers

- **Usage Scenario 2:** Cloud carriers provide th connectivity and transport of cloud services from clou providers to cloud consumers. As illustrated in Fig. 2. a cloud provider participates in and arranges for tw unique Service Level Agreements (SLAs), one with cloud carrier (e.g. SLA2) and one with a clou consumer (e.g. SLA1).
- A cloud provider arranges Service Level Agreemen (SLAs) with a cloud carrier and may request dedicate and encrypted connections to ensure the clou services are consumed at a consistent level accordin to the contractual obligations with the clou consumers. In this case, the provider may specify i requirements on capability, flexibility and functionali in SLA2 in order to provide essential requirements SLA1.

---- SLA between cloud consumer and cloud provider
– – SLA between cloud provider and cloud carrier

Fig. 2.6 : Usage Scenario for Cloud Carriers

- **Usage Scenario 3:** For a cloud service, a cloud auditor conducts independent assessments of the operation and security of the cloud service implementation. The audit may involve interactions with both the Cloud Consumer and the Cloud Provider.

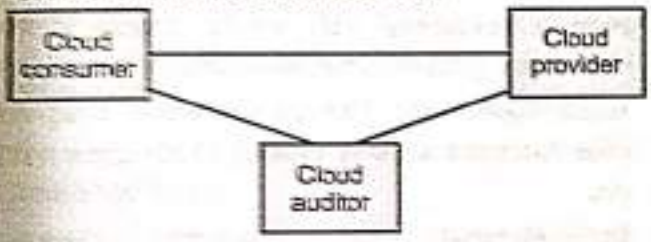


Fig. 2.7 : Usage Scenario for Cloud Auditors

Cloud Consumer

- The cloud consumer is the principal stakeholder for the cloud computing service. A cloud consumer represents a person or organization that maintains a business relationship with, and uses the service from a cloud provider. A cloud consumer browses the service catalog from a cloud provider, requests the appropriate service, sets up service contracts with the cloud provider, and uses the service.
- The cloud consumer may be billed for the service provisioned, and needs to arrange payments accordingly. Cloud consumers need SLAs to specify the technical performance requirements fulfilled by a cloud provider. SLAs can cover terms regarding the quality of service, security, remedies for performance failures.
- A cloud provider may also list in the SLAs a set of promises explicitly not made to consumers, i.e. limitations, and obligations that cloud consumers must accept. A cloud consumer can freely choose a cloud provider with better pricing and more favourable terms.
- Typically, a cloud provider's pricing policy and SLAs are non-negotiable, unless the customer expects heavy usage and might be able to negotiate for better contracts. Depending on the services requested, the activities and usage scenarios can be different among cloud consumers.

- Below Fig. 2.8 presents some example cloud services available to a cloud consumer (For details, see Appendix B: Examples of Cloud Services).

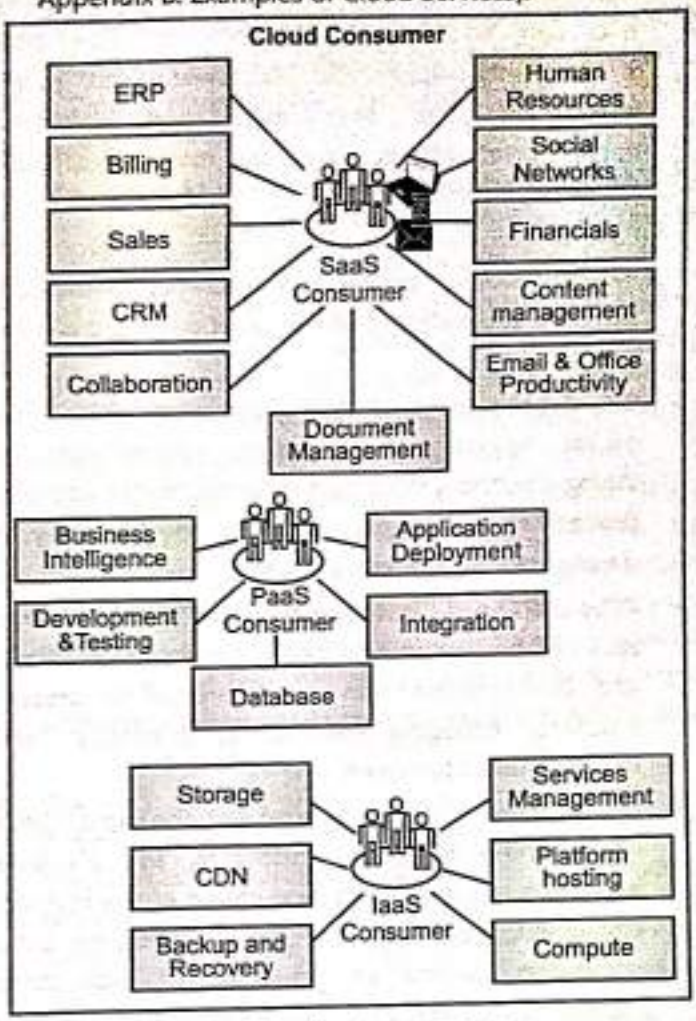


Fig. 2.8 : Cloud consumers

- **Cloud Consumers of SaaS:** SaaS applications in the cloud and made accessible via a network to the SaaS consumers.
- **Who are Cloud Consumer of SaaS?** The consumers of SaaS can be an organization that provide their members with access to software applications, end users who directly use software applications, or software application administrators who configure applications for end users. SaaS consumers can be billed based on the number of end users, the time of use, the network bandwidth consumed, the amount of data stored or duration of stored data.
- **Cloud Consumers of PaaS:** can employ the tools and execution resources provided by cloud providers to develop, test, deploy and manage the applications hosted in a cloud environment.

- **Who are Cloud Consumer of PaaS?** PaaS consumers can be application developers who design and implement application software, application testers who run and test applications in cloud-based environments, application deployers who publish applications into the cloud, and application administrators who configure and monitor application performance on a platform.
- PaaS consumers can be billed according to, processing, database storage and network resources consumed by the PaaS application, and the duration of the platform usage.
- **Cloud Consumers of IaaS:** Consumers of IaaS have access to virtual computers, network-accessible storage, network infrastructure components, and other fundamental computing resources on which they can deploy and run arbitrary software.
- **Who are Cloud Consumer of IaaS?** The consumers of IaaS can be system developers, system administrators and IT managers who are interested in creating, installing, managing and monitoring services for IT infrastructure operations.
- IaaS consumers are provisioned with the capabilities to access these computing resources, and are billed according to the amount or duration of the resources consumed, such as CPU hours used by virtual computers, volume and duration of data stored, network bandwidth consumed, number of IP addresses used for certain intervals.

Examples of Cloud Services

Some example cloud services available to a cloud consumer are listed below:

1. **SaaS Services**

- **Email and Office Productivity:** Applications for email, word processing, spreadsheets, presentations, etc.
- **Billing:** Application services to manage customer billing based on usage and subscriptions to products and services.
- **Customer Relationship Management (CRM):** CRM applications that range from call center applications to sales force automation.
- **Collaboration:** Tools that allow users to collaborate in workgroups, within enterprises, and across enterprises.
- **Content Management:** Services for managing t production of and access to content for web-bas applications.
- **Document Management:** Applications for manag documents, enforcing document produc workflows, and providing workspaces for groups enterprises to find and access documents.
- **Financials:** Applications for managing financ processes ranging from expense processing an invoicing to tax management.
- **Human Resources:** Software for managing huma resources functions within companies.
- **Sales:** Applications that are specifically designed f sales functions such as pricing, commission tracki etc.
- **Social Networks:** Social software that establishes an maintains a connection among users that are tied one or more specific types of interdependency.
- **Enterprise Resource Planning (ERP):** Integrate computer-based system used to manage internal an external resources, including tangible assets, financi resources, materials, and human resources.

2. **PaaS Services**

- **Business Intelligence:** Platforms for the creation applications such as dashboards, reporting system and data analysis.
- **Database:** Services offering scalable relation database solutions or scalable non-SQL datastores.
- **Development and Testing:** Platforms for th development and testing cycles of applicatio development, which expand and contract as needed.
- **Integration:** Development platforms for buildin integration applications in the cloud and within th enterprise.
- **Application Deployment:** Platforms suited for genera purpose application development. These service provide databases, web application runtim environments, etc.

3. **IaaS Services**

- **Backup and Recovery:** Services for backup an recovery of file systems and raw data stores on server and desktop systems.
- **Compute:** Server resources for running cloud-base systems that can be dynamically provisioned an configured as needed.

- **Content Delivery Networks (CDNs):** CDNs store content and files to improve the performance and cost of delivering content for web-based systems.
- **Services Management:** Services that manage cloud infrastructure platforms. These tools often provide features that cloud providers do not provide or specialize in managing certain application technologies.
- **Storage:** Massively scalable storage capacity that can be used for applications, backups, archival, and file storage.

Cloud Providers

A cloud provider is a person, or an organization; it is the entity responsible for making a service available to interested parties.

A Cloud Provider acquires and manages the computing infrastructure required for providing the services, runs the cloud software that provides the services, and makes arrangement to deliver the cloud services to the Cloud Consumers through network access.

What Cloud Providers does for SaaS?

- For Software as a Service, the cloud provider deploys, configures, maintains and updates the operation of the software applications on a cloud infrastructure so that the services are provisioned at the expected service levels to cloud consumers. The provider of SaaS assumes most of the responsibilities in managing and controlling the applications and the infrastructure, while the cloud consumers have limited administrative control of the applications.

What Cloud Providers does for PaaS?

- For PaaS, the Cloud Provider manages the computing infrastructure for the platform and runs the cloud software that provides the components of the platform, such as runtime software execution stack, databases, and other middleware components.
- The PaaS Cloud Provider typically also supports the development
- For deployment and management process of the PaaS Cloud Consumer by providing tools such as Integrated Development Environments (IDEs), development version of cloud software, Software Development Kits (SDKs), deployment and management tools.
- The PaaS Cloud Consumer has control over the applications and possibly some of the hosting environment settings, but has no or limited access to the infrastructure underlying the platform such as network, servers, Operating Systems (OS), or storage.

What Cloud Providers does for IaaS?

For IaaS –

- The Cloud Provider acquires the physical computing resources underlying the service, including the servers, networks, storage and hosting infrastructure.
- The Cloud Provider runs the cloud software necessary to makes computing resources available to the IaaS Cloud Consumer through a set of service interfaces and computing resource abstractions, such as virtual machines and virtual network interfaces.
- The IaaS Cloud Consumer in turn uses these computing resources, such as a virtual computer, for their fundamental computing needs Compared to SaaS and PaaS Cloud Consumers, an IaaS Cloud Consumer has access to more fundamental forms of computing resources and thus has more control over the more software components in an application stack, including the OS and network.
- The IaaS Cloud Provider, on the other hand, has control over the physical hardware and cloud software that makes the provisioning of these infrastructure services possible, for example, the physical servers, network equipments, storage devices, host OS and hypervisors for virtualization.
- A Cloud Provider's activities can be described in five major areas, as shown in below Fig. 2.9, a cloud provider conducts its activities in the areas of service deployment, service orchestration, cloud service management, security, and privacy. We will see detail in respective sections.

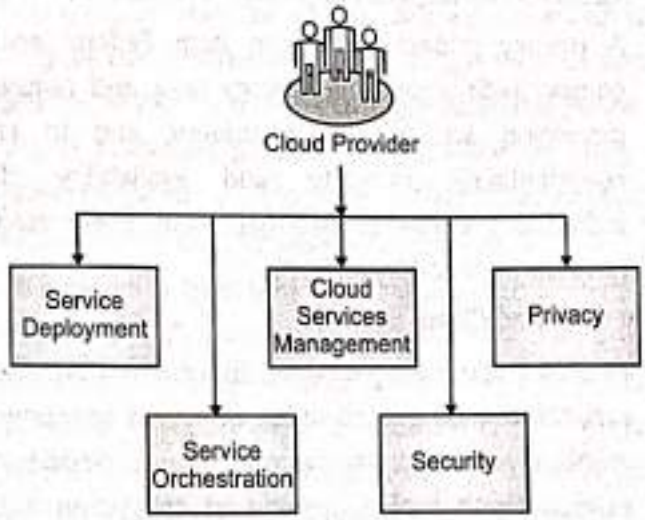


Fig. 2.9 : Cloud Provider – Major Activities

Cloud Auditor

- A cloud auditor is a party that can perform an independent examination of cloud service controls with the intent to express an opinion thereon. Audits are performed to verify conformance to standards through review of objective evidence. A cloud auditor can evaluate the services provided by a cloud provider in terms of security controls, privacy impact, performance, etc.
- Auditing is especially important for federal agencies as "agencies should include a contractual clause enabling third parties to assess security controls of cloud providers".

Cloud Security Audit?

- Security controls are the management, operational, and technical safeguards or countermeasures employed within an organizational information system to protect the confidentiality, integrity, and availability of the system and its information.
- For security auditing, a cloud auditor can make an assessment of the security controls in the information system to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to the security requirements for the system.
- The security auditing should also include the verification of the compliance with regulation and security policy. For example, an auditor can be tasked with ensuring that the correct policies are applied to data retention according to relevant rules for the jurisdiction. The auditor may ensure that fixed content has not been modified and that the legal and business data archival requirements have been satisfied.
- A privacy impact audit can help Federal agencies comply with applicable privacy laws and regulations governing an individual's privacy, and to ensure confidentiality, integrity, and availability of an individual's personal information at every stage of development and operation.

Why we Need Cloud Broker?

- As cloud computing evolves, the integration of cloud services can be too complex for cloud consumers to manage. A cloud consumer may request cloud services from a cloud broker, instead of contacting a cloud provider directly.

Who is Cloud Broker?

- A cloud broker is an entity that manages the performance and delivery of cloud services negotiates relationships between cloud providers cloud consumers.
- In general, a cloud broker can provide services in t categories:
  1. **Service Intermediation:** A cloud broker enha a given service by improving some spe capability and providing value-added service cloud consumers. The improvement can managing access to cloud services, ide management, performance reporting, enhan security, etc.
  2. **Service Aggregation:** A cloud broker comb and integrates multiple services into one or m new services. The broker provides data integra and ensures the secure data movement betw the cloud consumer and multiple cloud provide
  3. **Service Arbitrage:** Service arbitrage is like ser aggregation except that the services be aggregated are not fixed. Service arbitrage me a broker has the flexibility to choose services fr multiple agencies. The cloud broker, for exam can use a credit-scoring service to measure select an agency with the best score.

Who is Cloud Carrier?

- A cloud carrier acts as an intermediary that provi connectivity and transport of cloud services betwe cloud consumers and cloud providers.
- Cloud carriers provide access to consumers throu network, telecommunication and other access devic For example, cloud consumers can obtain clo services through network access devices, such computers, laptops, mobile phones, Mobile Inter Devices (MIDs), etc.
- The distribution of cloud services is normally provid by network and telecommunication carriers or transport agent, where a transport agent refers to business organization that provides physical transp of storage media such as high-capacity hard drives.
- Note that a cloud provider will set up SLAs with cloud carrier to provide services consistent with t level of SLAs offered to cloud consumers, and m require the cloud carrier to provide dedicated a

secure connections between cloud consumers and cloud providers.

Scope of Control between Provider and Consumer

- The Cloud Provider and Cloud Consumer share the control of resources in a cloud system. As show in below Fig. 2.10, different service models affect an organization's control over the computational resources and thus what can be done in a cloud system.
- The Fig. 2.10 shows these differences using a classic software stack notation comprised of the application, middleware, and OS layers. This analysis of delineation of controls over the application stack helps understand the responsibilities of parties involved in managing the cloud application.

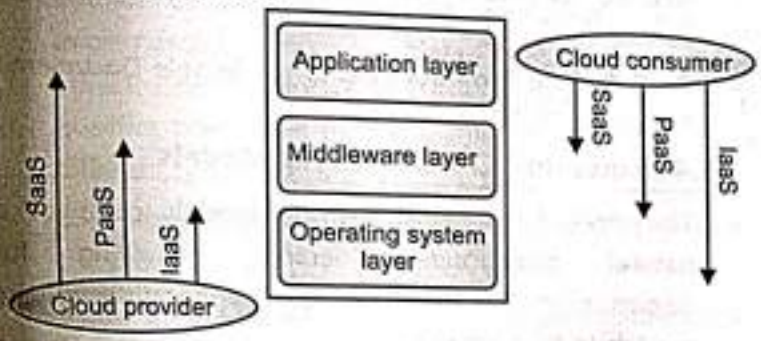


Fig. 2.10 : Scope of Controls between Provider and Consumer

- The application layer includes software applications targeted at end users or programs. The applications are used by SaaS consumers, or installed/managed/ maintained by PaaS consumers, IaaS consumers, and SaaS providers.
- The middleware layer provides software building blocks (e.g., libraries, database, and Java virtual machine) for developing application software in the cloud. The middleware is used by PaaS consumers, installed/managed/maintained by IaaS consumers or PaaS providers, and hidden from SaaS consumers.
- The OS layer includes operating system and drivers, and is hidden from SaaS consumers and PaaS consumers. An IaaS cloud allows one or multiple guest OS"s to run virtualized on a single physical host. Generally, consumers have broad freedom to choose which OS to be hosted among all the OS"s that could be supported by the cloud provider. The IaaS consumers should assume full responsibility for the guest OS"s, while the IaaS provider controls the host OS.

2.3 CLOUD DELIVERY MODELS

- A cloud delivery model represents a specific, pre-packaged combination of IT resources offered by a cloud provider.
- Following are the three common cloud delivery models have become widely established and formalized:
  1. Infrastructure-as-a-Service (IaaS)
  2. Platform-as-a-Service (PaaS)
  3. Software-as-a-Service (SaaS)
- Many specialized variations of the three base cloud delivery models have emerged, each comprised of a distinct combination of IT resources. Some examples include:
  - ➢ Storage-as-a-Service
  - ➢ Database-as-a-Service
  - ➢ Security-as-a-Service
  - ➢ Communication-as-a-Service
  - ➢ Integration-as-a-Service
  - ➢ Testing-as-a-Service
  - ➢ Process-as-a-Service

2.3.1 Infrastructure-as-a-Service (IaaS)

- The IaaS delivery model represents a self-contained IT environment comprised of infrastructure-centric IT resources that can be accessed and managed via cloud service-based interfaces and tools. This environment can include hardware, network, connectivity, operating systems, and other "raw" IT resources. In contrast to traditional hosting or outsourcing environments, with IaaS, IT resources are typically virtualized and packaged into bundles that simplify up-front runtime scaling and customization of the infrastructure.
- The general purpose of an IaaS environment is to provide cloud consumers with a high level of control and responsibility over its configuration and utilization. The IT resources provided by IaaS are generally not pre-configured, placing the administrative responsibility directly upon the cloud consumer.
- This model is therefore used by cloud consumers that require a high level of control over the cloud-based environment they intend to create. IT resources available through IaaS environments are generally offered as freshly initialized virtual instances. A central and primary IT resource within a typical IaaS

environment is the virtual server. Virtual servers are leased by specifying server hardware requirements, such as processor capacity, memory, and local storage space.

- **IaaS Service Provider Examples:** Amazon EC2, S3, Rightscale, vCloud.

2.3.2 Platform-as-a-Service (PaaS)

- The PaaS delivery model represents a pre-defined "ready-to-use" environment typically comprised of already deployed and configured IT resources. Specifically, PaaS relies on the usage of a ready-made environment that establishes a set of pre-packaged products and tools used to support the entire delivery lifecycle of custom applications.
- Common reasons a cloud consumer would use and invest in a PaaS environment include:
  - ➢ The cloud consumer wants to extend on-premise environments into the cloud for scalability and economic purposes.
  - ➢ The cloud consumer uses the ready-made environment to entirely substitute an on-premise environment.
  - ➢ The cloud consumer wants to become a cloud provider and deploys its own cloud services to be made available to other external cloud consumers.
- By working within a ready-made platform, the cloud consumer is spared the administrative burden of setting up and maintaining the bare infrastructure IT resources provided via the IaaS model. Conversely, the cloud consumer is granted a lower level of control over the underlying IT resources that host and provision the platform.
- PaaS products are available with different development stacks. For example, Google App Engine offers a Java and Python-based environment.
- **PaaS Service Provider Examples:** Windows Azure, Hadoop, Google AppEngine, and Aneka.

2.3.3 Software-as-a-Service (SaaS)

- A software program positioned as a shared cloud service and made available as a "product" or generic utility represents the typical profile of a SaaS offering. The SaaS delivery model is typically used to make a reusable cloud service widely available (often commercially) to a range of cloud consumers. An entire marketplace exists around SaaS products that can be leased and used for different purposes and different terms.
- A cloud consumer is generally granted very limited administrative control over a SaaS implementation. It is most often provisioned by the cloud provider, but can be legally owned by whichever entity assumes the cloud service owner role. For example, an organization acting as a cloud consumer while using and working with a PaaS environment can build a cloud service that it decides to deploy in that same environment as a SaaS offering. The same organization then effectively assumes the cloud provider role as the SaaS-based cloud service is made available to other organizations that act as cloud consumers when using that cloud service.
- **SaaS Service Provider Examples:** Google Documents, Facebook, Flickr, Salesforce.

2.3.4 Combining Cloud Delivery Models

- The three base cloud delivery models comprise a natural provisioning hierarchy, allowing for opportunities for the combined application of the models to be explored.
- IaaS + PaaS
- A PaaS environment will be built upon an underlying infrastructure comparable to the physical and virtual servers and other IT resources provided in an IaaS environment. A cloud provider would not normally need to provision an IaaS environment from its own cloud in order to make a PaaS environment available to cloud consumers.
- The motivation for such an arrangement is influenced by economics or maybe because the first cloud provider is close to exceeding its existing capacity for serving other cloud consumers. And, a particular cloud consumer imposes a legal requirement for data to be physically stored in a specific region i.e. different from where the first cloud provider's cloud resides.
- IaaS + PaaS + SaaS
- All three cloud delivery models can be combined to establish layers of IT resources that build upon each other. For example, by adding on to the preceding layered architecture, the ready-made environment provided by the PaaS environment can be used by the cloud consumer organization to develop and deploy its

own SaaS cloud services that it can then make available as commercial products.

2.4 TYPES OF CLOUDS

Clouds constitute the primary outcome of cloud computing. They are a type of parallel and distributed system harnessing physical and virtual computers presented as a unified computing resource. Clouds build the infrastructure on top of which services are implemented and delivered to customers. Such infrastructures can be of different types and provide useful information about the nature and the services offered by the cloud.

A more useful classification is given according to the administrative domain of a cloud. It identifies the boundaries within which cloud computing services are implemented, provides hints on the underlying infrastructure adopted to support such services, and qualifies them. It is then possible to differentiate four different types of cloud:

1. **Public Clouds:** The cloud is open to the wider public.
2. **Private Clouds:** The cloud is implemented within the private premises of an institution and generally made accessible to the members of the institution or a subset of them.
3. **Hybrid or Heterogeneous Clouds:** The cloud is a combination of the public and private cloud and most likely identifies a private cloud that has been augmented with resources or services hosted in a public cloud.
4. **Community Clouds:** The cloud is characterized by a multi-administrative domain involving different deployment models, and it is specifically designed to address the needs of a specific industry.

2.4.1 Public Clouds

- Public clouds constitute the first expression of cloud computing. They are a realization of the canonical view of cloud computing in which the services offered are made available to anyone, from anywhere, and at any time through the Internet. From a structural point of view they are a distributed system, most likely composed of one or more datacenters connected together, on top of which the specific services offered by the cloud are implemented. Any customer can easily sign in with the cloud provider, enter her credential and billing details, and use the services offered.
- A fundamental characteristic of public clouds is multi tenancy. A public cloud is meant to serve a multitude of users, not a single customer. Any customer requires a virtual computing environment that is separated, and most likely isolated, from other users. A public cloud can offer any kind of service: infrastructure, platform, or applications. For example, Amazon EC2 is a public cloud that provides infrastructure as a service; Google App Engine is a public cloud that provides an application development platform as a service; and SalesForce.com is a public cloud that provides software as a service. What makes public clouds peculiar is the way they are consumed: They are available to everyone and are generally architected to support a large quantity of users. What characterizes them is their natural ability to scale on demand and sustain peak loads.

2.4.2 Private Clouds

- Public clouds are appealing and provide a viable option to cut IT costs and reduce capital expenses, but they are not applicable in all scenarios. For example, a very common analysis to the use of cloud computing in its canonical implementation is the loss of control. In the case of public clouds, the provider is in control of the infrastructure and, eventually, of the customers' core logic and sensitive data. Even though there could be regulatory procedure in place that guarantees fair management and respect of the customer's privacy, this condition can still be perceived as a threat or as an unacceptable risk that some organizations are not willing to take. In particular, institutions such as government and military agencies will not consider public clouds as an option for processing or storing their sensitive data. The risk of a breach in the security infrastructure of the provider could expose such information to others; this could simply be considered unacceptable.
- Private clouds are virtual distributed systems that rely on a private infrastructure and provide internal users with dynamic provisioning of computing resources. Instead of a pay-as-you-go model as in public clouds, there could be other schemes in place, taking into account the usage of the cloud and proportionally

billing the different departments or sections of an enterprise. Private clouds have the advantage of keeping the core business operations in-house by relying on the existing IT infrastructure and reducing the burden of maintaining it once the cloud has been set up. In this scenario, security concerns are less critical, since sensitive information does not flow out of the private infrastructure. Moreover, existing IT resources can be better utilized because the private cloud can provide services to a different range of users. Another interesting opportunity that comes with private clouds is the possibility of testing applications and systems at a comparatively lower price rather than public clouds before deploying them on the public virtual infrastructure.

- **Customer Information Protection:** Despite assurances by the public cloud leaders about security, few provide satisfactory disclosure or have long enough histories with their cloud offerings to provide warranties about the specific level of security put in place on their systems. In-house security is easier to maintain and rely on.
- **Infrastructure Ensuring SLAs :** Quality of service implies specific operations such as appropriate clustering and failover, data replication, system monitoring and maintenance, and disaster recovery, and other uptime services can be commensurate to the application needs. Although public cloud vendors provide some of these features, not all of them are available as needed.
- **Compliance with Standard Procedures and Operations :** If organizations are subject to third-party compliance standards, specific procedures have to be put in place when deploying and executing applications. This could be not possible in the case of the virtual public infrastructure.

- All these aspects make the use of cloud-based infrastructures in private premises an interesting option.
- Private clouds can provide in-house solutions for cloud computing, but if compared to public clouds they exhibit more limited capability to scale elastically on demand.

2.4.3 Hybrid Clouds

- Public clouds are large software and hardwa infrastructures that have a capability that is hu enough to serve the needs of multiple users, but th suffer from security threats and administrative pitfa Although the option of completely relying on a pub virtual infrastructure is appealing for companies th did not incur IT capital costs and have just start considering their IT needs, in most cases the priva cloud option prevails because of the existing infrastructure.
- Private clouds are the perfect solution when it necessary to keep the processing of information with an enterprise's premises or it is necessary to use t existing hardware and software infrastructure. One the major drawbacks of private deployments is t inability to scale on demand and to efficiently addr peak loads. In this case, it is important to levera capabilities of public clouds as needed. Hence, a hybr solution could be an interesting opportunity for takin advantage of the best of the private and public world This led to the development and diffusion of hybr clouds.
- Hybrid clouds allow enterprises to exploit existing infrastructures, maintain sensitive information with the premises, and naturally grow and shrink b provisioning external resources and releasing the when they're no longer needed. Security concerns a then only limited to the public portion of the clo that can be used to perform operations with le stringent constraints but that are still part of th system workload.
- Fig. 2.11 provides a general overview of a hybrid clou It is a heterogeneous distributed system resulting fro a private cloud that integrates additional services resources from one or more public clouds. For th reason they are also called heterogeneous clouds. depicted in the diagram, dynamic provisioning is fundamental component in this scenario. Hybr clouds address scalability issues by leveraging extern resources for exceeding capacity demand.

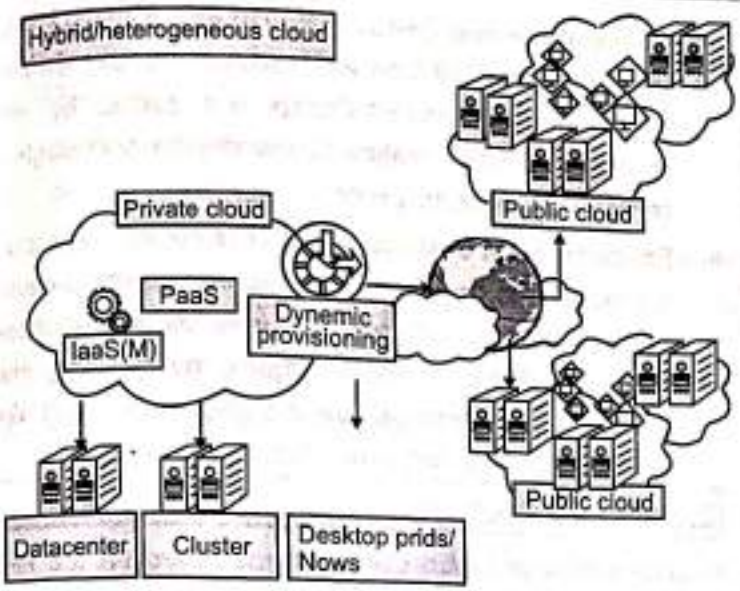


Fig. 2.11 : Hybrid/heterogeneous cloud overview

- These resources or services are temporarily leased for the time required and then released. This practice is also known as cloud bursting.
- Whereas the concept of hybrid cloud is general, it mostly applies to IT infrastructure rather than software services. Service-oriented computing already introduces the concept of integration of paid software services with existing application deployed in the private premises. In an IaaS scenario, dynamic provisioning refers to the ability to acquire on demand virtual machines in order to increase the capability of the resulting distributed system and then release them. Infrastructure management software and PaaS solutions are the building blocks for deploying and managing hybrid clouds. In particular, with respect to private clouds, dynamic provisioning introduces a more complex scheduling algorithm and policies, the goal of which is also to optimize the budget spent to rent public resources.

2.4.4 Community Clouds

- Community clouds are distributed systems created by integrating the services of different clouds to address the specific needs of an industry, a community, or a business sector.
- The National Institute of Standards and Technologies (NIST) characterize community clouds as follows:

"The infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise."

- Fig. 2.12 provides a general view of the usage scenario of community clouds, together with reference architecture. The users of a specific community cloud fall into a well-identified community, sharing the same concerns or needs; they can be government bodies, industries, or even simple users, but all of them focus on the same issues for their interaction with the cloud. This is a different scenario than public clouds, which serve a multitude of users with different needs. Community clouds are also different from private clouds, where the services are generally delivered within the institution that owns the cloud.
- From an architectural point of view, a community cloud is most likely implemented over multiple administrative domains. This means that different organizations such as government bodies, private enterprises, research organizations, and even public virtual infrastructure providers contribute with their resources to build the cloud infrastructure.

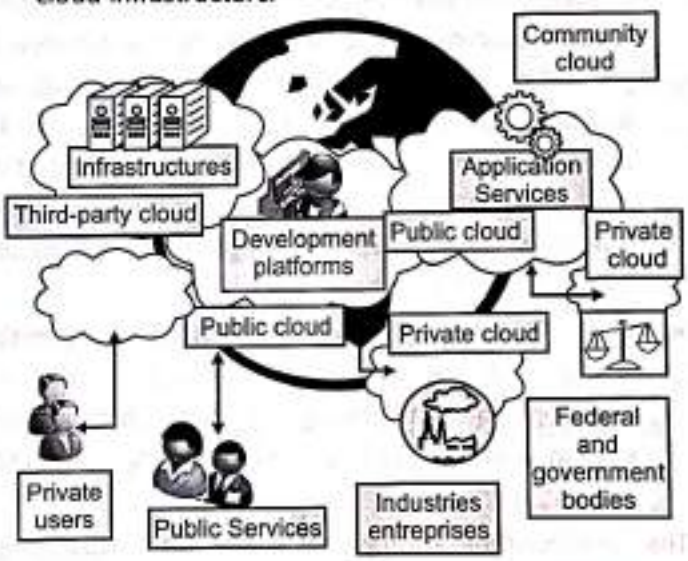


Fig. 2.12 : A community cloud

Candidate sectors for community clouds are as follows:

- **Media Industry :** In the media industry, companies are looking for low-cost, agile, and simple solutions to improve the efficiency of content production. Most media productions involve an extended ecosystem of partners. In particular, the creation of digital content is the outcome of a collaborative process that includes movement of large data, massive compute-intensive rendering tasks, and complex workflow executions. Community clouds can provide a shared environment

where services can facilitate business-to-business collaboration and offer the horsepower in terms of aggregate bandwidth, CPU, and storage required to efficiently support media production.

- **Healthcare Industry :** In the healthcare industry, there are different scenarios in which community clouds could be of use. In particular, community clouds can provide a global platform on which to share information and knowledge without revealing sensitive data maintained within the private infrastructure.
- **Energy and other Core Industries :** In these sectors, community clouds can bundle the comprehensive set of solutions that together vertically address management, deployment, and orchestration of services and operations. Since these industries involve different providers, vendors, and organizations, a community cloud can provide the right type of infrastructure to create an open and fair market.
- **Public Sector :** Legal and political restrictions in the public sector can limit the adoption of public cloud offerings. Moreover, governmental processes involve several institutions and agencies and are aimed at providing strategic solutions at local, national, and international administrative levels. They involve business-to-administration, citizen-to-administration, and possibly business-to-business processes. Some examples include invoice approval, infrastructure planning, and public hearings.
- **Scientific Research :** Science clouds are an interesting example of community clouds. In this case, the common interest driving different organizations sharing a large distributed infrastructure is scientific computing.

The benefits of these community clouds are the following:

- **Openness :** By removing the dependency on cloud vendors, community clouds are open systems in which fair competition between different solutions can happen.
- **Community :** Being based on a collective that provides resources and services, the infrastructure turns out to be more scalable because the system can grow simply by expanding its user base.
- **Graceful Failures :** Since there is no single provider or vendor in control of the infrastructure, there is no single point of failure.
- **Convenience and Control :** Within a community clo there is no conflict between convenience and cont because the cloud is shared and owned by a community, which makes all the decisions through collective democratic process.
- **Environmental Sustainability :** The community clo is supposed to have a smaller carbon footprint becau it harnesses underutilized resources. Moreover, thi clouds tend to be more organic by growing a shrinking in a symbiotic relationship to support t demand of the community, which in turn sustains it.

2.5 ECONOMICS OF THE CLOUD

- Economics of Cloud Computing is based on the Pa AS YOU GO method. Users/Customers must have t pay only for their way of usage of the cloud services definitely beneficial for the users. So that Cloud economically very convenient for all. Another side is t eliminate some indirect cost which is generated b assets such as license of software and their support cloud, users can use software application o subscription basis without any cost because th property of the software providing service remains t the cloud provider.
- Economical background of cloud is more useful fo developers in the following ways:
  - ➢ Pay as you go model offered by cloud providers.
  - ➢ Scalable and Simple.
- Cloud Computing Allows:
  - ➢ Reduces the capital costs of infrastructure.
  - ➢ Removes the maintenance cost.
  - ➢ Removes the administrative cost.

What is Capital Cost ?

- It is cost occurred in the purchasing infrastructure the assets that is important in the production of goods It takes long time to generate profit.
- In the case of start-ups, there is no extra budget fo the infrastructure and its maintenance. So cloud co minimizes expenses of any small organization in term of economy. It leads to the developers can only focu on the development logic and not on the maintenance of the infrastructure.
- There are three different Pricing Strategies which ar introduced by the Cloud Computing: Tiered Pricing Per-unit Pricing, and Subscription based Pricing. These are explained as following below.

1. **Tired Pricing:**

   Cloud Services are offered in the various tiers. Each tier offers fix service agreements at specific cost. Amazon EC2 uses this kind of pricing.

2. **Per-Unit Pricing:**

   The model is based upon the unit specific service concept. Data transfer and memory allocation includes in this model for specific units. GoGrid uses this kind of pricing in terms of RAM/hour.

3. **Subscription based Pricing:**

   In this model users are paying periodic subscription fee for the usage of software.

   So these models gives more flexible solutions about cloud economy.

2.6 OPEN CHALLENGES

- Cloud computing is used for enabling global access to mutual pools of resources such as services, apps, data, servers, and computer networks. It is done on either a third-party server located in a data center or a privately owned cloud. This makes data-accessing contrivances more reliable and efficient, with nominal administration effort.
- Because cloud technology depends on the allocation of resources to attain consistency and economy of scale, similar to a utility, it is also fairly cost-effective, making it the choice for many small businesses and firms.
- But there are also many challenges involved in cloud computing, and if you're not prepared to deal with them, you won't realize the benefits. Here are six common challenges you must consider before implementing cloud computing technology.

1. **Cost**

- Cloud computing itself is affordable, but tuning the platform according to the company's needs can be expensive. Furthermore, the expense of transferring the data to public clouds can prove to be a problem for short-lived and small-scale projects.
- Companies can save some money on system maintenance, management, and acquisitions. But they also have to invest in additional bandwidth, and the absence of routine control in an infinitely scalable computing platform can increase costs.

2. **Service Provider Reliability**

- The capacity and capability of a technical service provider are as important as price. The service provider must be available when you need them. The main concern should be the service provider's sustainability and reputation. Make sure you comprehend the techniques via which a provider observes its services and defends dependability claims.

3. **Downtime**

- Downtime is a significant shortcoming of cloud technology. No seller can promise a platform that is free of possible downtime. Cloud technology makes small companies reliant on their connectivity, so companies with an untrustworthy internet connection probably want to think twice before adopting cloud computing.

4. **Password Security**

- Industrious password supervision plays a vital role in cloud security. However, the more people you have accessing your cloud account, the less secure it is. Anybody aware of your passwords will be able to access the information you store there.
- Businesses should employ multi-factor authentication and make sure that passwords are protected and altered regularly, particularly when staff members leave. Access rights related to passwords and usernames should only be allocated to those who require them.

5. **Data Privacy**

- Sensitive and personal information that is kept in the cloud should be defined as being for internal use only, not to be shared with third parties. Businesses must have a plan to securely and efficiently manage the data they gather.

6. **Vendor Lock-In**

- Entering a cloud computing agreement is easier than leaving it. "Vendor lock-in" happens when altering providers is either excessively expensive or just not possible. It could be that the service is nonstandard or that there is no viable vendor substitute.
- It comes down to buyer carefulness. Guarantee the services you involve are typical and transportable to other providers, and above all, understand the requirements.

- Cloud computing is a good solution for many businesses, but it's important to know what you're getting into. Having plans to address these six prominent challenges first will help ensure a successful experience.

2.7 CLOUD INTEROPERABILITY AND STANDARDS

- Portability and interoperability relate to the ability to build systems from re-usable components that will work together "out of the box".
- A particular concern for cloud computing is cloud on-boarding – the deployment or migration of systems to a cloud service or set of cloud services. A common scenario is that some components cannot be moved to the cloud; for example, because of requirements for the enterprise to have complete control over personal data. On-boarding requires portability of those components that can be moved to the cloud, and interoperability with them of components that remain on in-house systems.

The Important Categories of Cloud Computing Portability and Interoperability

- A system that involves cloud computing typically includes data, application, platform, and infrastructure components, where:
  - ➢ Data is the machine-processable representation of information, held in computer storage.
  - ➢ Applications are software programs that perform functions related to business problems.
  - ➢ Platforms are programs that support the applications and perform generic functions that are not business-related.
  - ➢ Infrastructure is a collection of physical computation, storage, and communication resources.
- The application, platform, and infrastructure components can be as in traditional enterprise computing, or they can be cloud resources that are (respectively) software application programs (SaaS), software application platforms (PaaS), and virtual processors and data stores (IaaS).

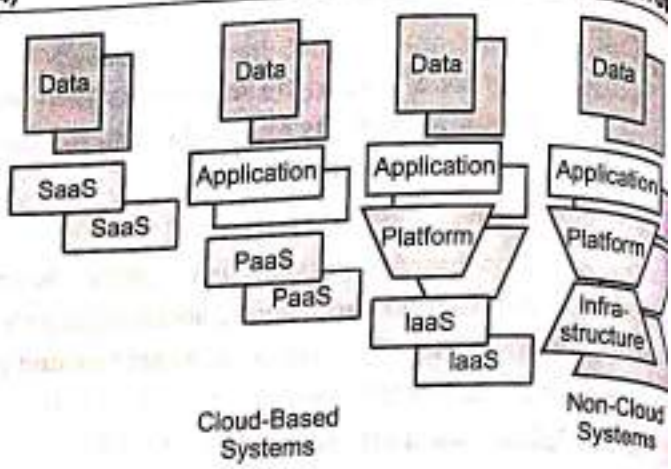


Fig. 2.13

Data, Applications, Platforms, and Infrastructure

- Non-cloud systems include mainframe minicomputers, personal computers, and mobi devices owned and used by enterprises an individuals.
- Data components interoperate via applicatio components rather than directly. There are no "dat interoperability" interfaces.
- Portability and interoperability of infrastructu components are achieved by hardware an virtualization architectures. The interfaces are most internal to the IaaS and infrastructure component shown in Data, Applications, Platforms, an Infrastructure. The interfaces exposed by thes components are physical communications interface these are important, but are the same as for traditiona computing. For these reasons, infrastructure portabilit and interoperability are not discussed further in th Guide.
- The main kinds of cloud computing portability t consider are data portability, application portabilit and platform portability. These are the portabilit respectively of data, application, and platfor components.
- Application interoperability between SaaS services an applications, and platform interoperability betwee PaaS services and platforms are important kinds o cloud computing interoperability to consider.
- Applications can include programs concerned with the deployment, configuration, provisioning, and operatio of cloud resources. Interoperability between thes programs and the cloud resource environments i important. This is management interoperability.

- Applications can also include programs such as app stores (for applications), data markets (for, e.g., openly available data) and cloud catalogues (e.g., reserved capacity exchanges, cloud service catalogs) from which users can acquire software products, data and cloud services, and to which developers can publish applications, data, and cloud services. In this Guide, all such programs are referred to as marketplaces. Publication and acquisition of products is performed by platforms, including PaaS services, that interface to the marketplaces. This is the final important cloud interoperability interface.

The cloud computing portability and interoperability categories to consider are thus:

1. Data Portability
2. Application Portability
3. Platform Portability
4. Application Interoperability
5. Platform Interoperability
6. Management Interoperability
7. Publication and Acquisition Interoperability

1. Data Portability

- Data portability enables re-use of data components across different applications.
- Suppose that an enterprise uses a SaaS product for Customer Relations Management (CRM), for example, and the commercial terms for use of that product become unattractive compared with other SaaS products or with use of an in-house CRM solution. The customer data held by the SaaS product may be crucial to the enterprise's operation. How easy will it be to move that data to another CRM solution?
- In many cases, it will be very difficult. The structure of the data is often designed to fit a particular form of application processing, and a significant transformation is needed to produce data that can be handled by a different product.
- This is no different from the difficulty of moving data between different products in a traditional environment. But, in a traditional environment, the customer is more often able to do nothing; to stay with an old version of a product, for example, rather than upgrading to a newer, more expensive one. With SaaS, the vendor can more easily force the customer to pay more or lose the service altogether.
- Cloud introduces no new technical problems, but its different commercial arrangements can make the old technical problems much more serious.

2. Application Portability

- Application portability enables the re-use of application components across cloud PaaS services and traditional computing platforms.
- Suppose that an enterprise has an application built on a particular cloud PaaS service and, for cost, performance, or other reasons, wishes to move it to another PaaS service or to in-house systems. How easy will this be?
- If the application uses features that are specific to the platform, or if the platform interface is non-standard, then it will not be easy.
- Application portability requires a standard interface exposed by the supporting platform. As discussed under Application Interoperability, this must enable the application to use the service discovery and information communication protocols implemented by the platform, as well as providing access to the platform capabilities that support the application directly. On a cloud PaaS platform, or a platform running on a cloud IaaS service, it may also enable applications to manage the underlying resources.
- A particular application portability issue that arises with cloud computing is portability between development and operational environments. Cloud PaaS is particularly attractive for development environments from a financial perspective, because it avoids the need for investment in expensive systems that will be unused once the development is complete. But, where a different environment is to be used at run time – either on in-house systems or on different cloud services – it is essential that the applications can be moved unchanged between the two environments. Cloud computing is bringing development and operations closer together, and indeed increasingly leading to the two being integrated as devops. This can only work if the same environment is used for development and operation, or if there is application portability between development and operation environments.

3. Platform Portability

There are Two Kinds of Platform Portability:

(i) Re-use of platform components across cloud IaaS services and non-cloud infrastructure – platform source portability

(ii) Re-use of bundles containing applications and data with their supporting platforms – machine image portability

- The UNIX operating system provides an example of platform source portability. It is mostly written in the C programming language, and can be implemented on different hardware by re-compiling it and re-writing a few small hardware-dependent sections that are not coded in C. Some other operating systems can be ported in a similar way. This is the traditional approach to platform portability. It enables applications portability because applications that use the standard operating system interface can similarly be re-compiled and run on systems that have different hardware. It is illustrated in Platform Source Portability.

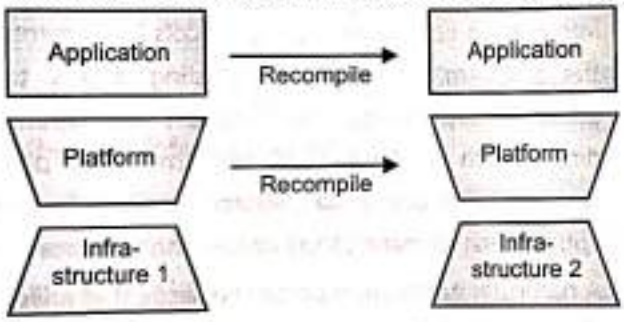


Fig. 2.14

Platform Source Portability

- Machine image portability gives enterprises and application vendors a new way of achieving applications portability, by bundling the application with its platform and porting the resulting bundle, as illustrated in Machine Image Portability. It requires a standard program representation that can be deployed in different IaaS use environments.

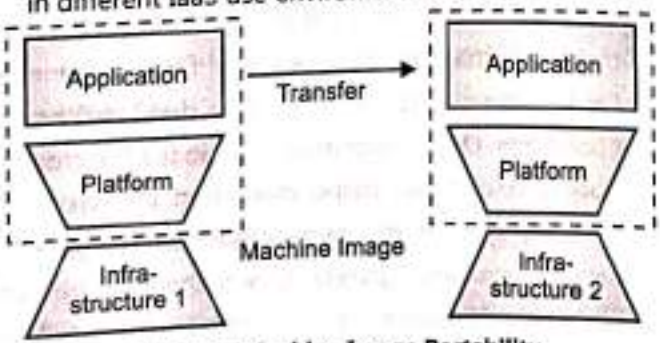


Fig. 2.15: Machine Image Portability

4. Application Interoperability

- Application interoperability is interoperability be
- application components deployed as SaaS,
- applications using PaaS, as applications on platf
- using IaaS, in a traditional enterprise IT environm
- or on client devices. An application component ma
- a complete monolithic application, or a part o
- distributed application.
- Interoperability is required, not just between diffe
- components, but between identical compone
- running in different clouds. For example, in a hyb
- cloud solution, an application component may
- deployed in a private cloud, with provision for a co
- to be run in a public cloud to handle traffic peaks.
- two components must work together.
- Data synchronization is a particular issue wh
- components in different clouds or internal resour
- work together, whether or not they are identical. Su
- components often keep copies of the same data, a
- these copies must be maintained in a consistent sta
- Communication between clouds typically has a hi
- latency, which makes synchronization difficult. Also, t
- two clouds may have different access control regim
- complicating the task of moving data between the
- The design approach must address:
  - ➢ Management of "system of record" sources
  - ➢ Management of data at rest and data in tran
    across domains that may be under control of
    cloud service consumer or provider
  - ➢ Data visibility and transparency
- Full interoperability includes dynamic discovery a
- composition: the ability to discover instances
- application components, and combine them with oth
- application component instances, at run time.
- Cloud SaaS gives enterprises the possibility
- acquiring new application capabilities quickly a
- easily, but much of the benefit of this is lost if cos
- integration work is needed to make the SaaS servi
- interoperate with other applications and services th
- the enterprise uses.
- Application components typically intercommunicate b
- invoking their respective platforms, which implemen
- the necessary communications protocols. Protoc
- standards enable platform interoperability directly a
- are discussed under that heading. They are indire
- enablers of application interoperability.

- Application interoperability requires more than communications protocols. It requires that the interoperating applications share common process and data models. These are not appropriate subjects for generic standards, although there are specific standards for some particular applications and business areas.
- There are, however, some design principles that improve application interoperability. Integration of applications designed following these principles still requires some effort, but is much less difficult and expensive than integration of applications that do not follow them.

5. Platform Interoperability

- Platform interoperability is interoperability between platform components, which may be deployed as PaaS, as platforms on IaaS, in a traditional enterprise IT environment, or on client devices.
- Platform interoperability requires standard protocols for service discovery and information exchange. As discussed above, these indirectly enable interoperability of the applications that use the platforms. Application interoperability cannot be achieved without platform interoperability.
- Service discovery is currently used by a minority of applications, but is essential to achieve the highest levels of service integration maturity [OSIMM]. Standard service discovery protocols should be supported by platforms used by service registries and other applications.
- Protocols for information exchange between platforms should support the establishment of sessions and transfer of session information, as well as information transport. (In the case of IaaS, the platform in question is not part of the infrastructure service but implemented on top of it.) Session information might, for example, include the user's identity, the authorization level established by the user for access control purposes, the user's time-zone, the user's language, and the user's preferred cultural environment.

6. Management Interoperability

- Management interoperability is interoperability between cloud services (SaaS, PaaS, or IaaS) and programs concerned with the implementation of on-demand self-service.
- As cloud computing grows, enterprises will want to manage cloud services together with their in-house systems, using generic off-the-shelf systems management products. This can only be achieved if cloud services have standard interfaces.
- These interoperability interfaces may provide the same functionality as the management interfaces mentioned under Application Portability.

Publication and Acquisition Interoperability

- Publication and acquisition interoperability is interoperability between platforms, including cloud PaaS services, and marketplaces (including app stores).
- Cloud service providers often maintain marketplaces from which their cloud services can be obtained. Some also make associated components available. For example, an IaaS supplier may make available machine images that run on its infrastructure services. Some large user organizations, including governments, maintain app stores to which approved suppliers can publish programs, which can then be downloaded by the organization's departments. Some mobile device suppliers maintain app stores from which users can obtain apps to run on their devices.
- Standard interfaces to these stores would lower the cost of cloud computing for software providers and users.

2.8 SCALABILITY AND FAULT TOLERANCE

- Scalability in cloud computing is the ability to quickly and easily increase or decrease the size or power of an IT solution. A scalable cloud is why you can sign up and use most cloud solutions in just a few minutes – if not seconds. It's why you can add resources like storage to an existing account just as quickly.
- There are usually two ways to scale a cloud solution up or down:
  1. Contact the cloud provider to request it
  2. Add the resources yourself via an online portal
- Some cloud solutions can also be auto-scaled. This means you can set them up to scale up or down automatically based on certain conditions, like when your cloud solution is running out of storage space.

Key Features of Cloud Scalability

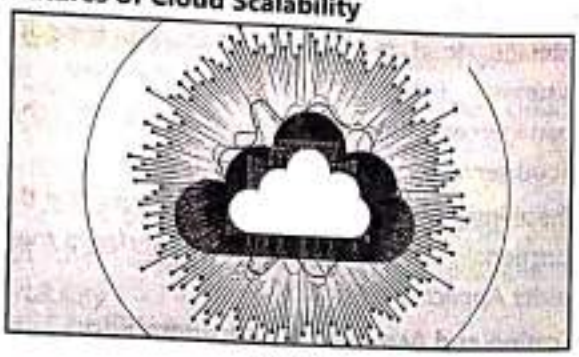


Fig. 2.16

- **Grow or Shrink:** Scaling is a change in size. It can mean increasing or decreasing
- **Sizeable Difference:** You don't use a cool word like "scalability" to describe a minor change. It should mean adding a significant amount of users or data, or hardware-like assets such as vCPUs and vRAM.
- **Non-Disruptive:** Scaling doesn't mean replacing. You're adding resources to an existing deployment, so there should be minimal downtime or learning curve. Adding seats to your Google Apps deployment as you grow – that's scaling. Switching to Office 365 because Google Apps can't support you any longer, not so much.
- **Relatively Fast:** Not at all cloud solutions scale up in minutes or with the click of a button, but scaling with the cloud should at least be faster than buying and setting up the hardware yourself.
- **Relatively Easy:** If scalability was easy in every case, we'd all be AWS Architects making a sweet $100,000+ per year. But the architecture of the cloud still makes things easier than scaling locally. Without virtualization, you'd have to run your largest apps on expensive, difficult-to-maintain mainframes.
- Even if you've never heard the term scalability before, you've probably done some scaling of your own without realizing it. If you've ever done any of these things:
  - ➢ Created a Gmail account
  - ➢ Added storage to your Dropbox account
  - ➢ Watched something on Netflix
- Then you've done some scaling, at least in a limited, frontend sense. What you've done is create an IT resource an email account, storage, or a streaming video without buying any additional hardware.
- There's a lot happening behind the scenes here, to With the Gmail example, Google's cloud automatic sets aside space for your new email account – which probably does millions of times per day, for its millions of other new users. Google is probably doing th without purchasing any new hardware, either. probably has a surplus of hundreds or thousands o servers, all set up and ready to host millions of Gma accounts.

6 Key Benefits of Cloud Scalability

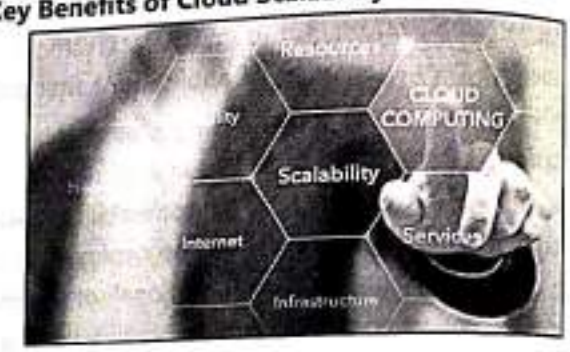


Fig. 2.17

1. It makes you feel like a god. There's something pret cool about being able to deploy thousands of serve or terabytes of data with a single click. You start t think, "Today, I deployed and configured all the serve I needed. Tomorrow...the world."
2. It makes your job easier. Don't tell your boss about th one, but adding resources with the cloud takes le time than doing it locally with physical hardware – a l less time. You can then spend a lot of this extra ti "working."
3. It makes disaster recovery easier. Not every busine can afford a hot or cold site. But scalability allows an business to rebuild their IT in just a few hours: you ju have to deploy new servers and copy over your data. can take weeks, on the other hand, to rebuild yo local IT with new physical servers.
4. It gives your business incredible speed and flexibil Want to open a new branch? Add a new team? Start new project or campaign? Scalability lets you add th IT resources for initiatives like this in minutes, no months.
5. It lets you avoid costly, disruptive migrations. Yo don't want to deploy your IT on a platform, only to fi that it can't support you after several years of suc growth. With a scalable platform, you only migrat when you want to – not when your underlyin platform lets you down.

6. It saves you money. There are no large upfront costs with the cloud. No $5,000+ servers, SANs, or networking equipment to buy. And you only pay for what you use. On a large scale, scalability reduces waste. It's why cloud providers can offer secure, reliable business email hosting for $5 per month and still be in business.

- Virtualization is what makes scalability in cloud computing possible. Virtual Machines (VMs) are scalable. They're not like physical machines, whose resources are relatively fixed. You can add any amount of resources to VMs at any time. You can scale them up by:
  - ➢ Moving them to a server with more resources
  - ➢ Hosting them on multiple servers at once (clustering)
- The other reason cloud computing is scalable? Cloud providers already have all the necessary hardware and software in place. Individual businesses, in contrast, can't afford to have surplus hardware on standby.
- Fault tolerance in cloud computing is about designing a blueprint for continuing the ongoing work whenever a few parts are down or unavailable. This helps the enterprises to evaluate their infrastructure needs and requirements, and provide services when the associated devices are unavailable due to some cause. It doesn't mean that the alternate arrangement can provide 100% of the full service, but this concept keeps the system in running mode at a useable, and most importantly, at a reasonable level. This is important if the enterprises are to keep growing in a continuous mode and increase their productivity levels.

Main Concepts behind Fault Tolerance in Cloud Computing System

- **Replication:** The fault tolerant system works on the concept of running several other replicates for each and every service. Thus, if one part of the system goes wrong, it has other instances that can be placed instead of it to keep it running. Take for example, a database cluster that has three servers with the same information on each of them. All the actions like data insertion, updates, and deletion get written on each of them. The servers, which are redundant, would be in the inactive mode unless and until any fault tolerance system doesn't demand the availability of them.
- **Redundancy:** When any system part fails or moves towards a downstate, then it is important to have backup type systems. For example, a website program that has MS SQL as its database may fail in between due to some hardware fault. Then a new database has to be availed in the redundancy concept when the original is in offline mode. The server operates with the emergency database which comprises of several redundant services within.

Techniques for Fault Tolerance in Cloud Computing

- All the services have to be given priority when designing a fault tolerance system. The database has to be given special preference because it powers several other units.
- After deciding the priorities, the enterprise has to work on the mock test. Take for example, the enterprise has a forum website that enables users to log in and posts comments. When the authentication services fail due to some problem, the users will not be able to log in. Then, the forum becomes a read-only one and does not serve the purpose. But with the fault tolerant systems, remediation will be ensured and the user can search for information with minimal impact.

Major Attributes of Fault Tolerance in Cloud Computing

- **None Point Failure:** The concepts of redundancy and replication defines that fault tolerance can be had but with some minor impacts. If there isn't even a single point failure then the system is not a fault tolerant one.
- **Accept the Fault Isolation Concept:** The fault occurrence has to be handled separately from other systems. This helps the enterprise to isolate it from the existing system failure.

Existence of Fault Tolerance in Cloud Computing

- **System Failure:** This may be either software or hardware issue. The software failure results in system crash or hanging situation that may be due to stack overflow or other reasons. Any improper maintenance of the physical hardware machines will result in hardware system failure.
- **Security Breach Occurrences:** There are several reasons why fault tolerance occurs due to security failures. The hacking of the server negatively impacts the server and results in data breach. Other reasons for the necessity of fault tolerance in the form of security

breaches include ransomware, phishing, virus attack etc.

- **Ready for the Cloud:** Web Application Design, Machine Image Design, Privacy Design, Database Management, Data Security, Network Security, Host Security, Compromise Response.

2.9 READY FOR THE CLOUD

2.9.1 Web Application Design

- Whether written in .NET, Ruby, Java, PHP, or anything else, web applications share a similar general architecture and architecture makes or breaks an application in the cloud.
- Fig. 2.18 illustrates the generic application architecture that web applications share.

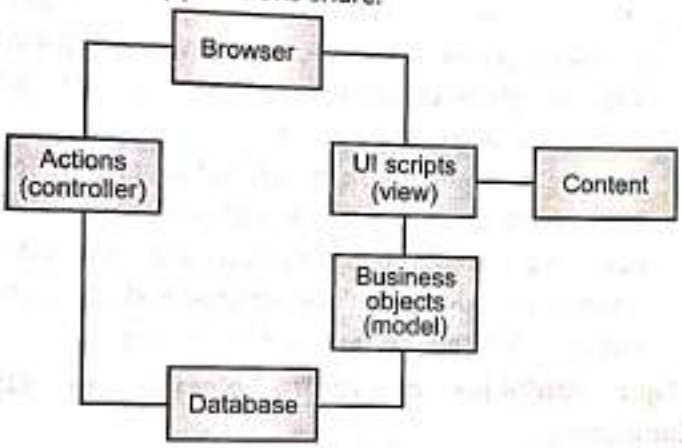


Fig. 2.18 : Most web applications share the same basic architecture

- You may move around or combine the boxes a bit, but you are certain to have some kind of (most often scripting) language that generates content from a combination of templates and data pulled from a model backed by a database. The system updates the model through actions that execute transactions against the model.

System State and Protecting Transactions

- The defining issue in moving to the cloud is how your application manages its state. Let's look at the problem of booking a room in a hotel.
- The architecture from Fig 2.18 suggests that you have represented the room and the hotel in a model. For the purposes of this discussion, it does not matter whether you have a tight separation between model, view, and data, or have mixed them to some degree. The key point is that there is some representation of the hotel and room data in your application space that mirrors their respective states in the database.
- How does the application state in the application change between the time the user makes the request and the transaction is changed in the database?

The Process Might Look Something Like This Basic Sequence:

- Lock the data associated with the room.
- Check the room to see whether it is currently available.
- If currently available, mark it as "booked" and thus no longer available.
- Release the lock.

The Problem with Memory Locks

- You can implement this logic in many different ways, not all of which will succeed in the cloud. A common Java approach that works well in a single-server environment but fails in a multiserver context might use the following code:

```
public void book(Customer customer, Room room, Date[] days)
  throws BookingException {
  synchronized( room )
  {
            // synchronized "locks" the room object
   if( !room.isAvailable(days) )
   {
   throw new BookingException("Room unavailable.");
    }
    room.book(customer, days);
  }
}
```

- Because the code uses the Java locking keyword synchronized, no other threads in the current process can make changes to the room object. If you are on a single server, this code will work under any locking supported by the server. Unfortunately, it will fail miserably in a multi server context.
- The problem with this example is the memory-based lock that the application grabs. If you had two clients making two separate booking requests against the same server, Java would allow only one of them to execute the synchronized block at a time. As a result you would not end up with a double booking.

- On the other hand, if you had each customer making a request against different servers (or even distinct processes on the same server), the synchronized blocks on each server could execute concurrently. As a result, the first customer to reach the room.book() call would lose his reservation because it would be overwritten by the second. Fig. 2.19 illustrates the double-booking problem.
- The non-Java way of expressing the problem is that if your transactional logic uses memory-based locking to protect the integrity of a transaction, that transaction will fail in a multi server environment and thus it won't be able to take advantage of the cloud's ability to dynamically scale application processing.
- One way around this problem is to use clustering technologies or cross-server shared memory systems. Another way to approach the problem is to treat the database as the authority on the state of your system.

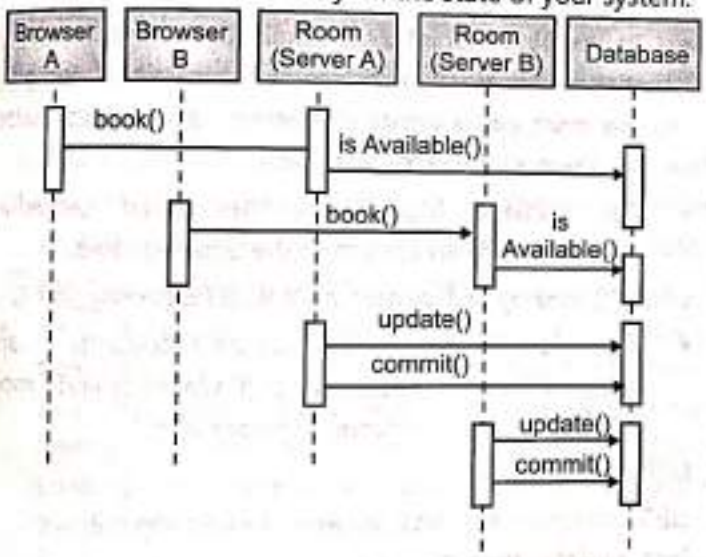


Fig. 2.19 : The second client overwrites the first, causing a double-booking

What if my application uses memory locks?

- I can hear a lot of you especially those of you who have massively multithreaded applications cursing me right now. If you find yourself in a situation in which you use memory-based locking and reworking the application away from that model is impractical, you can still move into the cloud. You simply won't be able to scale your application across multiple application servers.
- The way around this concern is to lock everything using a shared locking mechanism typically your database engine. The only other common alternative is to write some kind of home-grown distributed transaction management system. But why do that when your database already has one?

Transactional Integrity Through Stored Procedures

- I am not a huge fan of stored procedures. A key benefit of stored procedures, however, is that they enable you to leverage the database to manage the integrity of your transactions. After all, data integrity is the main job of your database engine!
- Instead of doing all of the booking logic in Java, you could leverage a MySQL stored procedure:

```
DELIMITER |

CREATE PROCEDURE book
(
 IN customerId BIGINT,
 IN roomId BIGINT,
 IN startDate DATE,
 IN endDate DATE,
 OUT success CHAR(1)
)
BEGIN
 DECLARE n DATE;
 DECLARE cust BIGINT;
 SET success = 'Y';
 SET n = startDate;
 bookingAttempt:
 REPEAT
  SELECT customer INTO cust FROM booking
  WHERE room_id = roomId AND booking_date = n;
  IF cust IS NOT NULL AND cust <> customerId
  THEN
   SET success = 'N';
   LEAVE bookingAttempt;
  END IF;
  UPDATE booking SET customer = customerId
  WHERE room_id = roomId AND booking_date = n;
  SET n = DATE_ADD(n, INTERVAL 1 DAY);
 UNTIL n > endDate
 END REPEAT;
 IF success = 'Y' THEN
  COMMIT;
 ELSE
  ROLLBACK;
 END IF;
END
```

- This method goes through each row of the booking table in your MySQL database and marks it booked by the specified customer. If it encounters a date when the room is already booked, the transaction fails and rolls back.

An example using the stored procedure follows, using Python:

```
def book(customerId, roomId, startDate, endDate):
  conn = getConnection();
  c = conn.cursor();
  c.execute("CALL book(%s, %s, %s, %s, @success)", \
        (customerId, roomId, startDate, endDate));
  c.execute("SELECT @success");
  row = c.fetchone();
  success = row[0];
  if success == "y":
    return 1
  else:
    report 0
```

- Even if you have two different application servers running two different instances of your Python application, this transaction will fail, as desired, for the second customer, regardless of the point at which the second customer's transaction begins.

Two Alternatives to Stored Procedures

- As I noted earlier, I am not a fan of stored procedures. They have the advantage of executing faster that the same logic in an application language. Furthermore, multi server transaction management through stored procedures is very elegant. But I have three key objections:
  1. Stored procedures are not portable from one database to another.
  2. They require an extended understanding of database programming—something that may not be available to all development teams.
  3. They don't completely solve the problem of scaling transactions across application servers under all scenarios. You still need to write your applications to use them wisely, and the result may, in fact, make your application more complicated.
- In addition to these core objections, I personally strongly prefer a very strict separation of presentation, business modeling, business logic, and data.
- The last objection is subjective and perhaps a matter of personal quirk. The first two objections, however, are real problems. After all, how many of you reading this book have found yourselves stuck with Oracle applications that could very easily work in MySQL if it weren't for all the stored procedures? You are paying a huge Oracle tax just because you used stored procedures to build your applications!
- The second objection is a bit more esoteric. If you have the luxury of a large development staff with a diverse skill set, you don't see this problem. If you are in a small company that needs each person to wear multiple hats, it helps to have an application architecture that requires little or no database programming expertise.
- To keep your logic at the application server level while still maintaining multi server transactional integrity, you must either create protections against dirty writes or create a lock in the database.
- The booking logic from the stored procedure essentially was an update to the booking table:

```
UPDATE booking SET customer = ? WHERE booking_id = ?
```

- If you add last_update_timestamp and last_update_user fields, that SQL would operate more effectively in a multiserver environment:

```
UPDATE booking
SET customer = ?, last_update_timestamp = ?,
last_update_user = ?
WHERE booking_id = ? AND last_update_timestamp = ?
AND last_update_user = ?;
```

- In this situation, the first client will attempt to book the room for the specified date and succeed. The second client then attempts to update the row but gets no matches since the timestamp it reads as well as the user ID of the user on the client will not match the values updated by the first client. The second client realizes it has updated zero rows and subsequently displays an error message. No double booking!
- This approach works well as long as you do not end up structuring transactions in a way that will create *deadlocks*. A deadlock occurs between two transactions

when each transaction is waiting on the other to release a lock. Our reservations system example is an application in which a deadlock is certainly possible.

- Because we are booking a range of dates in the same transaction, poorly structured application logic could cause two clients to wait on each other as one attempts to book a date already booked by the other, and vice versa. For example, if you and I are looking to book both Tuesday and Wednesday, but for whatever reason your client first tries Wednesday and my client first tries Tuesday, we will end up in a deadlock where I wait on you to commit your Wednesday booking and you wait on me to commit my Tuesday booking.
- This somewhat contrived scenario is easy to address by making sure that you move sequentially through each day. Other application logic, however, may not have as obvious a solution.
- Another alternative is to create a field for managing your locks. The room table, for example, might have two extra columns for booking purposes: locked_by and locked_timestamp. Before starting the transaction that books the rooms, update the room table and commit the update. Once your booking transaction completes, release the lock by nulling out those fields prior to committing that transaction.
- Because this approach requires two different database transactions, you are no longer executing the booking as a single atomic transaction. Consequently, you risk leaving an open lock that prevents others from booking any rooms on any dates. You can eliminate this problem through two tricks:
  1. The room is considered unlocked not only when the fields are NULL, but also when the locked_timestamp has been held for a long period of time.
  2. When updating the lock at the end of your booking transaction, use the locked_by and locked_timestamp fields in the WHERE clause. Thus, if someone else steals a lock out from under you, you only end up rolling back your transaction.
- Both of these approaches are admittedly more complex than taking advantage of stored procedures. Regardless of what approach you use, however, the important key for the cloud is simply making sure that you are not relying on memory locking to maintain your application state integrity.

When Servers Fail

- The ultimate architectural objective for the cloud is to set up a running environment where the failure of any given application server ultimately doesn't matter. If you are running just one server, that failure will obviously matter at some level, but it will still matter less than losing a physical server.
- One trick people sometimes use to get around the problems described in the previous section is data segmentation also known as *sharding*. Fig. 2.20 shows how you might use data segmentation to split processing across multiple application servers.

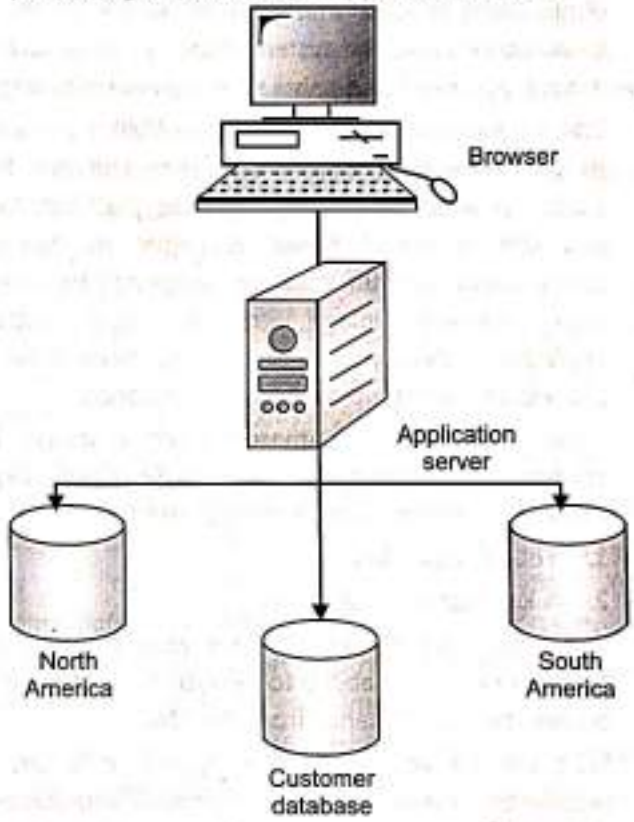


Fig. 2.20 : Supporting different hotels on different servers guarantees no double bookings

- In other words, each application server manages a subset of data. As a result, there is never any risk that another server will overwrite the data. Although segmentation has its place in scaling applications, that place is not at the application server in a cloud cluster. A segmented application server cluster ultimately has a very low availability rating, as the failure of any individual server does matter.
- The final corollary to all of this discussion of application state and server failure is that application servers in a cloud cannot store any state data beyond

caching data. In other words, if you need to back up your application server, you have failed to create a solid application server architecture for the cloud. All state information including binary data belongs in the database, which must be on a persistent system.

2.9.2 Machine Image Design

Two indirect benefits of the cloud are:

1. It forces discipline in deployment planning
2. It forces discipline in disaster recovery

- Thanks to the way virtualized servers launch from machine images, your first step in moving into any cloud infrastructure is to create a repeatable deployment process that handles all the issues that could come up as the system starts up. To ensure that it does, you need to do some deployment planning.
- The machine image (in Amazon, the AMI) is a raw copy of your operating system and core software for a particular environment on a specific platform. When you start a virtual server, it copies its operating environment from the machine image and boots up. If your machine image contains your installed application, deployment is nothing more than the process of starting up a new virtual instance.
- When you create an Amazon machine image, it is encrypted and stored in an Amazon S3 bundle. One of two keys can subsequently decrypt the AMI:
  1. Your Amazon key
  2. A key that Amazon holds
- Only your user credentials have access to the AMI. Amazon needs the ability to decrypt the AMI so it can actually boot an instance from the AMI.
- Even though your AMI is encrypted, it is strongly recommend never storing any sensitive information in an AMI. Not only does Amazon have theoretical access to decrypt the AMI, but there also are mechanisms that enable you to make your AMI public and thus perhaps accidentally share whatever sensitive data you were maintaining in the AMI.
- For example, if one company sues another Amazon customer, a court may subpoena the other Amazon customer's data. Unfortunately, it is not uncommon for courts to step outside the bounds of common sense and require a company such as Amazon to make available all Amazon customer data. If you want to make sure your data is never exposed as the result of a third-party subpoena, you should not store that data in an Amazon AMI.
- Instead, encrypt it separately and load it into your instance at launch so that Amazon will not have the decryption keys and thus the data cannot be accessed, unless you are a party to the subpoena.
- A machine image should include all of the software necessary for the runtime operation of a virtual instance based on that image *and nothing more*. The starting point is obviously the operating system, but the choice of components is absolutely critical. The full process of establishing a machine image consists of the following steps:
  - ➢ Create a component model that identifies what components and versions are required to run the service that the new machine image will support.
  - ➢ Separate out stateful data in the component model. You will need to keep it out of your machine image.
  - ➢ Identify the operating system on which you will deploy.
  - ➢ Search for an existing, trusted baseline public machine image for that operating system.
  - ➢ Harden your system using a tool such as Bastille.
  - ➢ Install all of the components in your component model.
  - ➢ Verify the functioning of a virtual instance using the machine image.
  - ➢ Build and save the machine image.
- The starting point is to know exactly what components are necessary to run your service. Fig. 2.21 shows a sample model describing the runtime components for a MySQL database server.

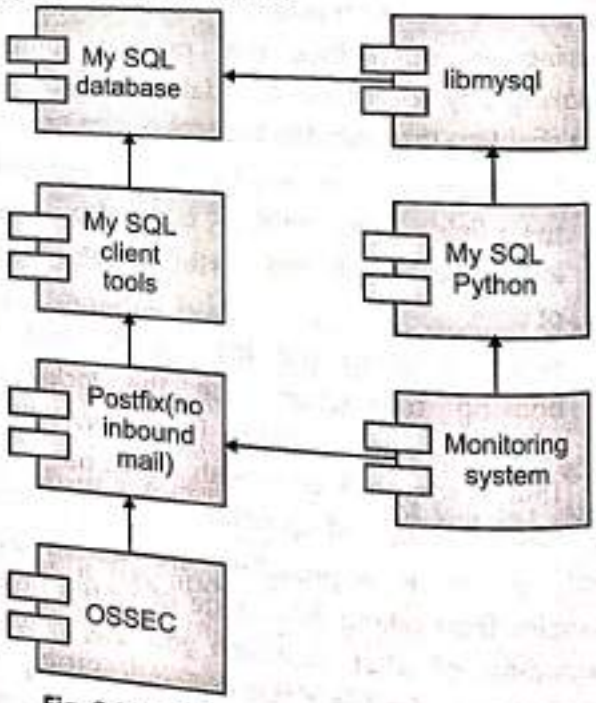


Fig. 2.21 : Software necessary to support a MySQL database server

- In this case, the stateful data exists in the MySQL directory, which is externally mounted as a block storage device. Consequently, you will need to make sure that your startup scripts mount your block storage device before starting MySQL.
- Because the stateful data is assumed to be on a block storage device, this machine image is useful in starting *any* MySQL databases, not just a specific set of MySQL databases.
- The services you want to run on an instance generally dictate the operating system on which you will base the machine image. If you are deploying a .NET application, you probably will use one of the Amazon Windows images. A PHP application, on the other hand, probably will be targeting a Linux environment.
- Hardening an operating system is the act of minimizing attack vectors into a server. Among other things, hardening involves the following activities:
  - ➢ Removing unnecessary services.
  - ➢ Removing unnecessary accounts.
  - ➢ Running all services as a role account (not root) when possible.
  - ➢ Running all services in a restricted jail when possible.
  - ➢ Verifying proper permissions for necessary system services.
- Now that you have a secure base from which to operate, it is time to actually install the software that this system will support. In the case of the current example, it's time to install MySQL.
- When installing your server-specific services, you may have to alter the way you think about the deployment thanks to the need to keep stateful data out of the machine image. For a MySQL server, you would probably keep stateful data on a block device and mount it at system startup. A web server, on the other hand, might store stateful media assets out in a cloud storage system such as Amazon S3 and pull it over into the runtime instance on startup.
- Different applications will definitely require different approaches based on their unique requirements. Whatever the situation, you should structure your deployment so that the machine image has the intelligence to look for its stateful data upon startup and provide your machine image components with access to that data before they need it.
- Once you have the deployment structured the right way, you will need to test it. That means testing the system from launch through shutdown and recovery. Therefore, you need to take the following steps:
  - ➢ Build a temporary image from your development instance.
  - ➢ Launch a new instance from the temporary image.
  - ➢ Verify that it functions as intended.
  - ➢ Fix any issues.
  - ➢ Repeat until the process is robust and reliable.
- The trick to creating a machine image that supports database servers is knowing how your database engine of choice stores its data. In the case of MySQL, the database engine has a data directory for its stateful data. This data directory may actually be called any number of things(*/usr/local/mysql/data*, */var/lib/mysql*, etc.), but it is the only thing other than the configuration file that must be separated from your machine image. In a typical custom build, the data directory is */usr/local/mysql/data*.
- Once you start an instance from a standard image and harden it, you need to create an elastic block storage volume and mount it. The standard Amazon approach is to mount the volume off of */mnt* (e.g., */mnt/database*). Where you mount it is technically unimportant, but it can help reduce confusion to keep the same directory for each image.
- You can then install MySQL, making sure to install it within the instance's root file system (e.g., */usr/local/mysql*). At that point, move the data over into the block device using the following steps:
  - ➢ Stop MySQL if the installation process automatically started it.
  - ➢ Move your data directory over into your mount and give it a name more suited to mounting on a separate device (e.g., */mnt/database/mysql*).
  - ➢ Change your *my.cnf* file to point to the new data directory.
- You now have a curious challenge on your hands: MySQL cannot start up until the block device has been mounted, but a block device under Amazon EC2 cannot be attached to an instance of a virtual machine

until that instance is running. As a result, you cannot start MySQL through the normal boot-up procedures. However, you can end up where you want by enforcing the necessary order of events: boot the virtual machine, mount the device, and finally start MySQL. You should therefore carefully alter your MySQL startup scripts so that the system will no longer start MySQL on startup, but will still shut the MySQL engine down on shutdown.

- The best way to effect this change is to edit the MySQL startup script to wait for the presence of the MySQL data directory before starting the MySQL executable.
- In approaching AMI design, you can follow one of two core philosophies:
  1. A minimalist approach in which you build a few multipurpose machine images.
  2. A comprehensive approach in which you build numerous purpose-specific machine images.
- The minimalist approach has the advantage of being easier for rolling out security patches and other operating-system-level changes. On the flip side, it takes a lot more planning and EC2 skills to structure a multipurpose AMI capable of determining its function after startup and self-configuring to support that function. If you are just getting started with EC2, it is probably best to take the comprehensive approach and use cloud management tools to eventually help you evolve into a library of minimalist machine images.
- For a single application installation, you won't likely need many machine images, and thus the difference between a comprehensive approach and a minimalist approach is negligible. SaaS applications especially ones that are not multitenant require a runtime deployment of application software.
- Runtime deployment means uploading the application software such as the MySQL executable discussed in the previous section to a newly started virtual instance after it has started, instead of embedding it in the machine image. A runtime application deployment is more complex (and hence the need for cloud management tools) than simply including the application in the machine image, but it does have a number of major advantages:
  - ➢ You can deploy and remove applications from a virtual instance while it is running. As a result, in a multi application environment, you can easily move an application from one cluster to another.
  - ➢ You end up with automated application restoration. The application is generally deployed at runtime using the latest backup image. When you embed the application in an image, on the other hand, your application launch is only as good as the most recent image build.
  - ➢ You can avoid storing service-to-service authentication credentials in your machine image and instead move them into the encrypted backup from which the application is deployed.

2.9.3 Privacy Design

Privacy in the Cloud

- The key to privacy in the cloud or any other environment is the strict separation of sensitive data from non sensitive data followed by the encryption of sensitive elements. The simplest example is storing credit cards. You may have a complex e-commerce application storing many data relationships, but you need to separate out the credit card data from the rest of it to start building a secure e-commerce infrastructure.
- Fig. 2.22 provides an application architecture in which credit card data can be securely managed.

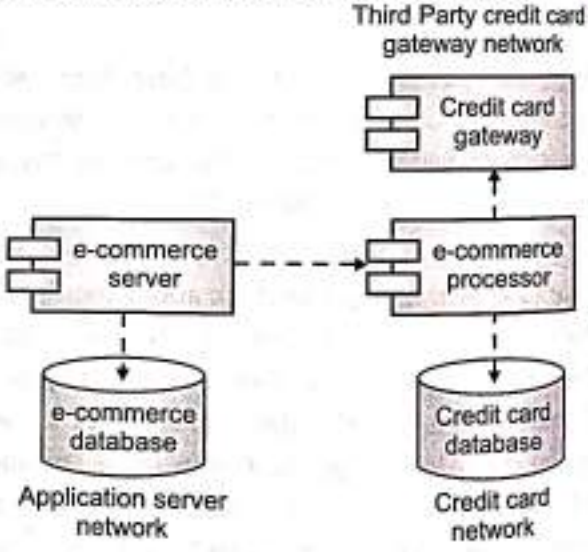


Fig. 2.22 : Host credit card data behind a web service that encrypts credit card data

- It's a pretty simple design that is very hard to compromise as long as you take the following precautions:
  - ➢ The application server and credit card server sit in two different security zones with only web service

traffic from the application server being allowed into the credit card processor zone.

- ➢ Credit card numbers are encrypted using a customer-specific encryption key.
- ➢ The credit card processor has no access to the encryption key, except for a short period of time (in memory) while it is processing a transaction on that card.
- ➢ The application server never has the ability to read the credit card number from the credit card server.
- ➢ No person has administrative access to both servers.

- Under this architecture, a hacker has no use for the data on any individual server; he must hack both servers to gain access to credit card data. Of course, if your web application is poorly written, no amount of structure will protect you against that failing.
- You therefore need to minimize the ability of a hacker to use one server to compromise the other. Because this problem applies to general cloud security. For now, I'll just list a couple rules of thumb:
  - ➢ Make sure the two servers have different attack vectors. In other words, they should not be running the same software. By following this guideline, you guarantee that whatever exploit compromised the first server is not available to compromise the second server.
  - ➢ Make sure that neither server contains credentials or other information that will make it possible to compromise the other server. In other words, don't use passwords for user logins and don't store any private SSH keys on either server.

Managing the Credit Card Encryption

- In order to charge a credit card, you must provide the credit card number, an expiration date, and a varying number of other data elements describing the owner of the credit card. You may also be required to provide a security code.
- This architecture separates the basic capture of data from the actual charging of the credit card. When a person first enters her information, the system stores contact info and some basic credit card profile information with the e-commerce application and sends the credit card number over to the credit card processor for encryption and storage.
- The first trick is to create a password on the e-commerce server and store it with the customer record. It's not a password that any user will ever see or use, so you should generate something complex using the strongest password guidelines. You should also create a credit card record on the e-commerce server that stores everything except the credit card number. Fig. 2.23 shows a sample e-commerce data model.

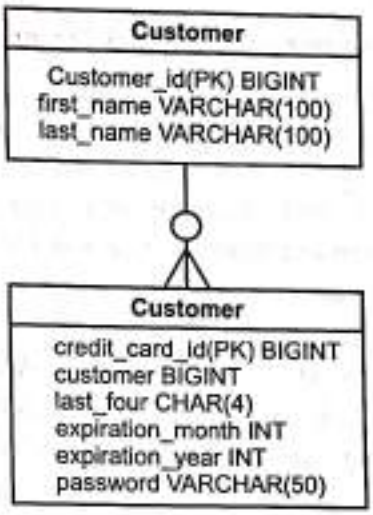


Fig. 2.23 : The e-commerce system stores everything but the credit card number and security code

- With that data stored in the e-commerce system database, the system then submits the credit card number, credit card password, and unique credit card ID from the e-commerce system to the credit card processor.
- The credit card processor does not store the password. Instead, it uses the password as salt to encrypt the credit card number, stores the encrypted credit card number, and associates it with the credit card ID. Fig. 2.24 shows the credit card processor data model.

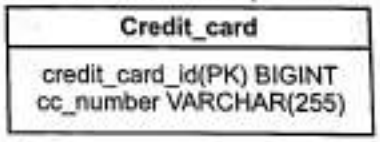


Fig. 2.24 : The credit card processor stores the encrypted credit card number and associates it with the e-commerce credit card ID

- Neither system stores a customer's security code, because the credit card companies do not allow you to store this code.

Processing a Credit Card Transaction

- When it comes time to charge the credit card, the e-commerce service submits a request to the credit card processor to charge the card for a specific amount. The e-commerce system refers to the credit card on the

credit card processor using the unique ID that was created when the credit card was first inserted. It passes over the credit card password, the security code, and the amount to be charged. The credit card processor then decrypts the credit card number for the specified credit card using the specified password. The unencrypted credit card number, security code, and amount are then passed to the bank to complete the transaction.

If the e-Commerce Application is Compromised

- If the e-commerce application is compromised, the attacker has access only to the non sensitive customer contact info. There is no mechanism by which he can download that database and access credit card information or otherwise engage in identity theft. That would require compromising the credit card processor separately.
- Having said all of that, if your e-commerce application is insecure, an attacker can still assume the identity of an existing user and place orders in their name with deliveries to their address. In other words, you still need to worry about the design of each component of the system.

If the Credit Card Processor is Compromised

- Compromising the credit card processor is even less useful than compromising the e-commerce application. If an attacker gains access to the credit card database, all he has are random unique IDs and strongly encrypted credit card numbers each encrypted with a unique encryption key. As a result, the attacker can take the database offline and attempt to brute-force decrypt the numbers, but each number will take a lot of time to crack and, ultimately, provide the hacker with a credit card number that has no individually identifying information to use in identity theft.
- Another attack vector would be to figure out how to stick a Trojan application on the compromised server and listen for decryption passwords. However, if you are running intrusion detection software as , even this attack vector becomes unmanageable.

When the Amazon Cloud Fails to Meet Your Needs

- The architecture I described in the previous section matches traditional noncloud deployments fairly closely. You may run into challenges deploying in the Amazon cloud, however, because of a couple of critical issues involving the processing of sensitive data:
  1. Some laws and specifications impose conditions on the political and legal jurisdictions where such data is stored. In particular, companies doing business in the EU may not store private data about EU citizens on servers in the U.S. (or any other nation falling short of EU privacy standards).
  2. Some laws and specifications were not written with virtualization in mind. In other words, they specify physical servers in cases where virtual servers would do identically well, simply because a server meant a physical server at the time the law or standard was written.
- The first problem has a pretty clear solution: if you are doing business in the EU and managing private data on EU citizens, that data must be handled on servers with a physical presence in the EU, stored on storage devices physically in the EU, and not pass through infrastructure managed outside the EU.
- Amazon provides a presence in both the U.S. and EU. As a result, you can solve the first problem by carefully architecting your Amazon solution. It requires, however, that you associate the provisioning of instances and storage of data with your data management requirements.
- The second issue is especially problematic for solutions such as Amazon that rely entirely on virtualization. In this case, however, it's for fairly stupid reasons. You can live up to the spirit of the law or specification, but because the concept of virtualization was not common at the time, you cannot live up to the letter of the law or specification. The workaround for this scenario is similar to the workaround for the first problem.
- In solving these challenges, you want to do everything to realize as many of the benefits of the cloud as possible without running private data through the cloud and without making the overall complexity of the system so high that it just isn't worth it. Cloud providers such as Rackspace and GoGrid tend to make such solutions easier than attempting a hybrid solution with Amazon and something else.
- To meet this challenge, you must route and store all private information outside the cloud, but execute as much application logic as possible inside the cloud.

You can accomplish this goal by following the general approach I described for credit card processing and abstracting the concepts out into a privacy server and a web application server:

- The privacy server sits outside the cloud and has the minimal support structures necessary to handle your private data.
- The web application server sits inside the cloud and holds the bulk of your application logic.

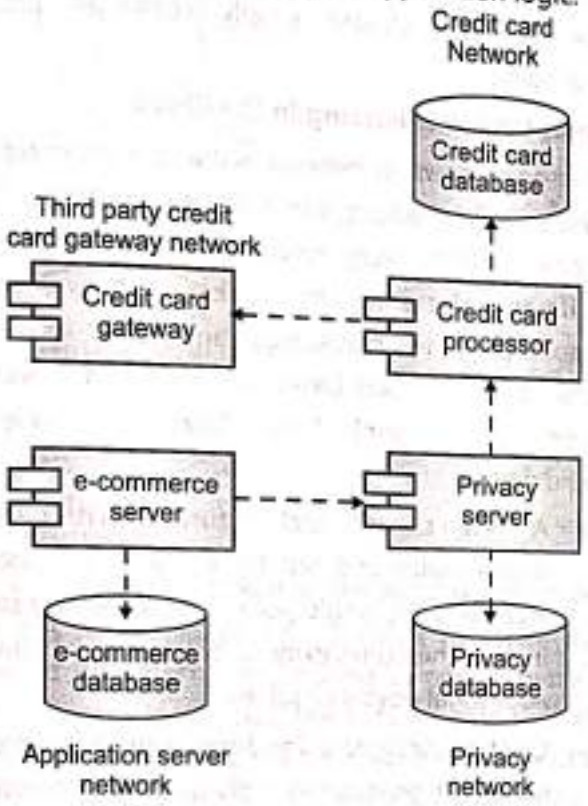


Fig. 2.25: Pulling private data out of the cloud creates three different application components

- Because the objective of a privacy server is simply to physically segment out private data, you do not necessarily need to encrypt everything on the privacy server. Fig. 2.25 illustrates how the e-commerce system might evolve into a privacy architecture designed to store all private data outside of the cloud.
- As with the cloud-based e-commerce system, you store credit card data on its own server in its own network segment. The only difference for the credit card processor is that this time it is outside of the cloud.
- The new piece to this puzzle is the customer's personally identifying information. This data now exists on its own server outside of the cloud, but still separate from credit card data. When saving user profile information, those actions execute against the privacy server instead of the main web application. Under no circumstances does the main web application have any access to personally identifying information, unless that data is aggregated before being presented to the web application.
- How useful this architecture is depends heavily on how much processing you are doing that has nothing to do with private data. If all of your transactions involve the reading and writing of private data, you gain nothing by adding this complexity. On the other hand, if the management of private data is just a tiny piece of the application, you can gain all of the advantages of the cloud for the other parts of the application while still respecting any requirements around physical data location.

2.9.4 Database Management

- The trickiest part of managing a cloud infrastructure is the management of your persistent data. Persistent data is essentially any data that needs to survive the destruction of your cloud environment. Because you can easily reconstruct your operating system, software, and simple configuration files, they do not qualify as persistent data. Only the data that cannot be reconstituted qualify. If you are following my recommendations, this data lives in your database engine.
- The problem of maintaining database consistency is not unique to the cloud. The cloud simply brings a new challenge to an old problem of backing up your database, because your database server in the cloud will be much less reliable than your database server in a physical infrastructure. The virtual server running your database will fail completely and without warning. Count on it.
- Whether physical or virtual, when a database server fails, there is the distinct possibility that the files that comprise the database state will get corrupted. The likelihood of that disaster depends on which database engine you are using, but it can happen with just about any engine out there.
- Absent of corruption issues, dealing with a database server in the cloud is very simple. In fact, it is much easier to recover from the failure of a server in a virtualized environment than in the physical world: simply launch a new instance from your database

machine image, mount the old block storage device, and you are up and running.

- The most effective mechanism for avoiding corruption is leveraging the capabilities of a database engine that supports true *clustering*. In a clustered database environment, multiple database servers act together as a single logical database server. The mechanics of this process vary from database engine to database engine, but the result is that a transaction committed to the cluster will survive the failure of any one node and maintain full data consistency. In fact, clients of the database will never know that a node went down and will be able to continue operating.
- Unfortunately, database clustering is very complicated and generally quite expensive.
  - ➢ Unless you have a skilled DBA on hand, you should not even consider undertaking the deployment of a clustered database environment.
  - ➢ A clustered database vendor often requires you to pay for the most expensive licenses to use the clustering capabilities in the Database Management System (DBMS). Even if you are using MySQL clustering, you will have to pay for five machine instances to effectively run that cluster.
  - ➢ Clustering comes with significant performance problems. If you are trying to cluster across distinct physical infrastructures in other words, across availability zones you will pay a hefty network latency penalty.
- The alternative to clustering is replication. A replication-based database infrastructure generally has a main server, referred to as the database master. Client applications execute write transactions against the database master. Successful transactions are then replicated to database slaves.

Replication has Two Key Advantages Over Clustering:

- It is generally much simpler to implement.
- It does not require an excessive number of servers or expensive licenses.
- Unfortunately, replication is not nearly as reliable as clustering. A database master can, in theory, fail after it has committed a transaction locally but before the database slave has received it. In that event, you would have a database slave that is missing data. In fact, when a database master is under a heavy load, the database slave can actually fall quite far behind the master. If the master is somehow corrupted, it can also begin replicating corrupted data.
- Apart from reliability issues, a replicated environment does not failover as seamlessly as a clustered solution. When your database master fails, clients using the master for write transactions cannot function until the master is recovered. On the other hand, when a node in a cluster fails, the clients do not notice the failure because the cluster simply continues processing transactions.

Using Database Clustering in the Cloud

- The good news, in general, is that the cloud represents few specific challenges to database clustering. The bad news is that every single database engine has a different clustering mechanism (or even multiple approaches to clustering) and thus an in-depth coverage of cloud-based clustering is beyond the scope of this book. I can, however, provide a few guidelines:
  - ➢ A few cluster architectures exist purely for performance and not for availability. Under these architectures, single points of failure may still exist. In fact, the complexity of clustering may introduce additional points of failure.
  - ➢ Clusters designed for high availability are often slower at processing individual write transactions, but they can handle much higher loads than standalone databases. In particular, they can scale to meet your read volume requirements.
  - ➢ Some solutions such as MySQL may require a large number of servers to operate effectively. Even if the licensing costs for such a configuration are negligible, the cloud costs will add up.
  - ➢ The dynamic nature of IP address assignment within a cloud environment may add new challenges in terms of configuring clusters and their failover rules.

Using Database Replication in the Cloud

- For most non mission-critical database applications replication is a "good enough" solution that can save you a lot of money and potentially provide you with opportunities for performance optimization. In fact, a MySQL replication system in the cloud can provide you with a flawless backup and disaster recovery system as

well as availability that can almost match that of a cluster. Because the use of replication in the cloud can have such a tremendous impact compared to replication in a traditional data center, we'll go into a bit more detail on using replication in the cloud than we did with clustering.

- Fig. 2.26 shows a simple replication environment.

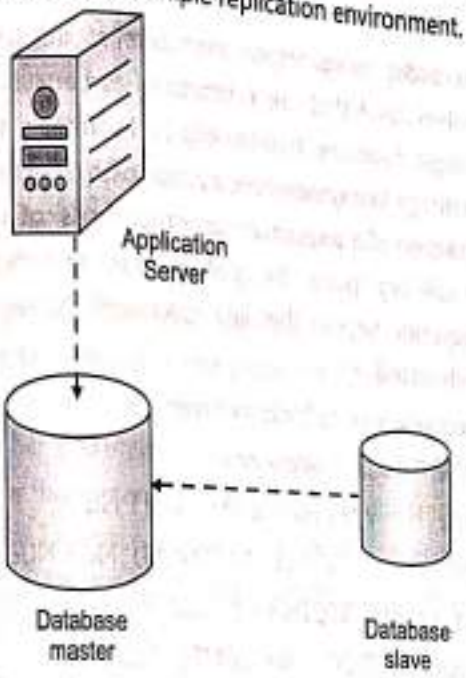


Fig. 2.26 : A simple replication (arrows show dependency)

- In this structure, you have a single database server of record (the master) replicating to one or more copies (the slaves). In general, the process that performs the replication from the master to the slave is not atomic with respect to the original transaction. In other words, just because a transaction successfully commits on the master does not mean that it successfully replicated to any slaves. The transactions that do make it to the slaves are generally atomic, so although a slave may be out of sync, the database on the slave should always be in an internally consistent state (uncorrupted).
- Under a simple setup, your web applications point to the database master. Consequently, your database slave can fail without impacting the web application. To recover, start up a new database slave and point it to the master.
- Recovering from the failure of a database master is much more complicated. If your cloud provider is Amazon, it also comes with some extra hurdles you won't see in a standard replication setup.
- Ideally, you will recover your database master by starting a new virtual server that uses your database machine image and then mounting the volume that was formerly mounted by the failed server. The failure of your master, however, may have resulted in the corruption of the files on that volume. At this point, you will turn to the database slave.
- A database can recover using a slave in one of two ways:
  1. Promotion of a slave to database master (you will need to launch a replacement slave)
  2. Building a new database master and exporting the current state from a slave to a new master
- Promotion is the fastest mechanism for recovery and the approach you almost certainly want to take, unless you have a need for managing distinct database master and database slave machine images. If that's the case, you may need to take the more complex recovery approach.
- As with other components in your web application architecture, putting your database in a replication architecture gives it the ability to rapidly recover from a node failure and, as a result, significantly increases overall system availability rating.

Replication for Performance

- Another reason to leverage replication is performance. Without segmenting your data, most database engines allow you to write against only the master, but you can read from the master or any of the slaves. An application heavy on read operations can therefore see significant performance benefits from spreading reads across slaves. Fig. 2.27 illustrates the design of an application using replication for performance benefits.

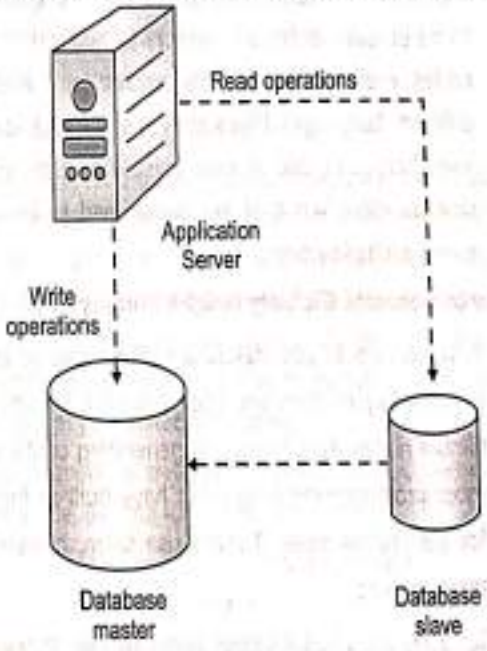


Fig. 2.27 : By separating read operations to execute against slaves, your applications can scale without clustering

- The rewards of using replication for performance are huge, but there are also risks. The primary risk is that you might accidentally execute a write operation against one of the slaves. When you do that, replication falls apart and your master and slaves end up in inconsistent states. Two approaches to solving this problem include:
  1. Clearly separating read logic from write logic in your code and centralizing the acquisition of database connections.
  2. Making your slave nodes read-only.
- The second one is the most foolproof, but it complicates the process of promoting a slave to master because you must reconfigure the server out of read-only mode before promoting it.

Primary Key Management

- With a web application operating behind a load balancer in which individual nodes within the web application do not share state information with each other, the problem of cross-database primary key generation becomes a challenge. The database engine's auto-increment functionality is specific to the database you are using and not very flexible; it often is guaranteed to be unique only for a single server.
- In *Java Database Best Practices* (O'Reilly; http://oreilly.com/catalog/9780596005221/index.html), I describe in detail a mechanism for generating keys in memory in an application server that are guaranteed to be unique across any number of application server nodes even across multiple applications written in different languages. I'll cover that technique at a high level here and add a new twist: the generation of random identifiers that are guaranteed to be unique across multiple nodes.

How to Generate Globally Unique Primary Keys

- First, you could use standard UUIDs to serve as your primary key mechanism. They have the benefit of an almost nonexistent chance of generating conflicts, and most programming languages have built-in functions for generating them. I don't use them, however, for three reasons:
  1. They are 128-bit values and thus take more space and have longer lookup times than the 64-bit primary keys I prefer.
  2. Cleanly representing a 128-bit value in Java and some other programming languages is painful. In fact, the best way to represent such a value is through two separate values representing the 64 high bits and the 64 low bits, respectively.
  3. The possibility of collisions, although not realistic, does exist.
- In order to generate identifiers at the application server level that are guaranteed to be unique in the target database, traditionally I rely on the database to manage key generation. I accomplish this through the creation of a sequencer table that hands out a key with a safe key space. The application server is then free to generate keys in that key space until the key space is exhausted.

The sequencer table looks like this:

```
CREATE TABLE sequencer (
    name        VARCHAR(20)     NOT NULL,
    next_key    BIGINT UNSIGNED NOT NULL,
    last_update BIGINT UNSIGNED NOT NULL,
    spacing     INT UNSIGNED NOT NULL;
    PRIMARY KEY ( name, last_update ),
    UNIQUE INDEX ( name )
);
```

- The first thing of note here is that there is nothing specific to any database in this table structure and your keys are not tied to a particular table. If necessary, multiple tables can share the same primary key space. Similarly, you can generate unique identifiers that have nothing to do with a particular table in your database.
- To generate a unique person_id for your person table:
  1. Set up a next_key value in memory and initialize it to 0.
  2. Grab the next spacing and last_update for the sequencer record with the name = 'person.person_id'.
  3. Add 1 to the retrieved next_key and update the sequencer table with the name and retrieved last_update value in the WHERE clause.
  4. If no rows are updated (because another server beat you to the punch), repeat steps 2 and 3.
  5. Set the next person ID to next_key.
  6. Increment the next_key value by 1.

7. The next time you need a unique person ID, simply execute steps 5 and 6 as long as next_key < and repeat the entire process.

- Within the application server, this entire process must be locked against multithreaded access.

Support for Globally Unique Random Keys

- The technique for unique key generation just described generates (more or less) sequential identifiers. In some cases, it is important to remove reasonable predictability from identifier generation. You therefore need to introduce some level of randomness into the equation.
- To get a random identifier, you need to multiply your next_key value by some power of 10 and then add a random number generated through the random number generator of your language of choice. The larger the random number possibility, the smaller your overall key space is likely to be. On the other hand, the smaller the random number possibility, the easier your keys will be to guess.
- The following Python example illustrates how to generate a pseudorandom unique person ID:

```
import thread
import random

nextKey = -1;
spacing = 100;
lock = thread.allocate_lock();

def next():
  try:
    lock.acquire();
    # make sure only one thread at a time can access
    if nextKey == -1 or nextKey > spacing:
      loadKey();
    nextId = (nextKey * 100000);
    nextKey = nextKey + 1;
  finally:
    lock.release();
  rnd = random.randint(0,99999);
  nextId = nextId + rnd;
  return nextId;
```

- You can minimize the wasting of key space by tracking the allocation of random numbers and incrementing the nextKey value only after the random space has been sufficiently exhausted. The further down that road you go, however, the more likely you are to encounter the following challenges:
  - ➢ The generation of unique keys will take longer.
  - ➢ Your application will take up more memory.
  - ➢ The randomness of your ID generation is reduced.

Database Backups

- Throughout this book, I have hinted at the challenge of database backup management and its relationship to disaster recovery. But for now, I will deal with the specific problem of performing secure database backups in the cloud.
- A good database backup strategy is hard, regardless of whether or not you are in the cloud. In the cloud, however, it is even more important to have a working database backup strategy.

Types of Database Backups

- Most database engines provide multiple mechanisms for executing database backups. The rationale behind having different backup strategies is to provide a trade-off between the impact that executing a backup has on the production environment and the integrity of the data in the backup. Typically, your database engine will offer at least these backup options (in order of reliability):
  - ➢ Database export/dump backup
  - ➢ File system backup
  - ➢ Transaction log backup
- The most solid backup you can execute is the database export/dump. When you perform a database export, you dump the entire schema of the database and all of its data to one or more export files. You can then store the export files as the backup. During recovery, you can leverage the export files to restore into a pristine install of your database engine.
- To execute a database export on SQL Server, for example, use the following command:

```
BACKUP DATABASE website to disk =
'D:\db\website.dump'
```

- The result is an export file you can move from one SQL Server environment to another SQL Server environment.

- The downside of the database export is that your database server must be locked against writes in order to get a complete export that is guaranteed to be in an internally consistent state. Unfortunately, the export of a large database takes a long time to execute. As a result, full database exports against a production database generally are not practical.
- Most databases provide the option to export parts of the database individually. For example, you could dump just your access_log table every night. In MySQL:

```
$ mysqldump website access_log >
/backups/db/website.dump
```

- If the table has any dependencies on other tables in the system, however, you can end up with inconsistent data when exporting on a table-by-table basis. Partial exports are therefore most useful on data from a data warehouse.
- File system backups involve backing up all of the underlying files that support the database. For some database engines, the database is stored in one big file. For others, the tables and their schemas are stored across multiple files. Either way, a backup simply requires copying the database files to backup media.
- Though a filesystem backup requires you to lock the database against updates, the lock time is typically shorter. In fact, the snapshotting capabilities of block storage devices generally reduce the lock time to under a second, no matter how large the database is.
- The following SQL will freeze MySQL and allow you to snapshot the filesystem on which the database is stored:

```
FLUSH TABLES WITH READ LOCK
```

- With the database locked, take a snapshot of the volume, and then release the lock.
- The least disruptive kind of backup is the transaction log backup. As a database commits transactions, it writes those transactions to a transaction log file. Because the transaction log contains only committed transactions, you can back up these transaction log files without locking the database or stopping. They are also smaller files and thus back up quickly. Using this strategy, you will create a full database backup on a nightly or weekly basis and then back up the transaction logs on a more regular basis.
- Restoring from transaction logs involves restoring from the most recent full database backup and then applying the transaction logs. This approach is a more complex backup scheme than the other two because you have a number of files created at different times that must be managed together. Furthermore, restoring from transaction logs is the longest of the three restore options.

Applying a Backup Strategy for the Cloud

- The best backup strategy for the cloud is a file-based backup solution. You lock the database against writes, take a snapshot, and unlock it. It is elegant, quick, and reliable. The key cloud feature that makes this approach possible is the cloud's ability to take snapshots of your block storage volumes. Without snapshot capabilities, this backup strategy would simply take too long.
- Your backup strategy cannot, however, end with a file-based backup. Snapshots work beautifully within a single cloud, but they cannot be leveraged outside your cloud provider. In other words, an Amazon EC2 elastic block volume snapshot cannot be leveraged in a cloud deployment. To make sure your application is portable between clouds, you need to execute full database exports regularly.
- How regularly you perform your database exports depends on how much data you can use. The underlying question you need to ask is, "If my cloud provider suddenly goes down for an extended period of time, how much data can I afford to lose when launching in a new environment?"
- For a content management system, it may be OK in such an extreme situation to lose a week of data. An e-commerce application, however, cannot really afford to lose any data even under such extreme circumstances.
- My approach is to regularly execute full database exports against a MySQL slave, as shown in Fig 2.28.

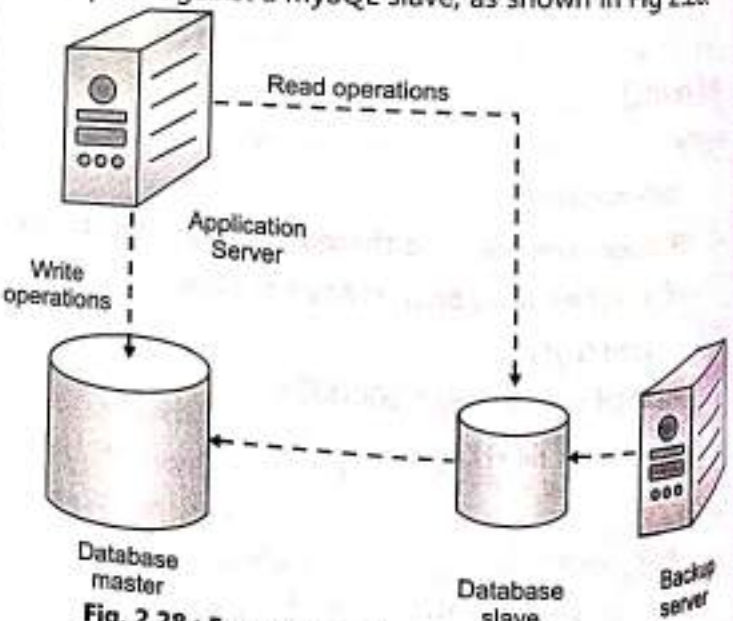


Fig. 2.28 : Execute regular full database exports against a replication slave

- For the purposes of a backup, it does not matter if your database slave is a little bit behind the master. What matters is that the slave represents the consistent state of the entire database at a relatively reasonable point in time. You can therefore execute a very long backup against the slave and not worry about the impact on the performance of your production environment. Because you can execute long backups, you can also execute numerous backups bounded mostly by your data storage appetite.
- If your database backups truly take such a long time to execute that you risk having your slaves falling very far behind the master, it makes sense to configure multiple slaves and rotate backups among the slaves. This rotation policy will give a slave sufficient time to catch up with the master after it has executed a backup and before it needs to perform its next backup.
- Once the backup is complete, you should move it over to S3 and regularly copy those backups out of S3 to another cloud provider or your own internal file server.
- Your application architecture should now be well structured to operate not only in the Amazon cloud, but in other clouds as well.

2.10 SECURITY

2.10.1 Data Security

- Moving to the public cloud or using a hybrid cloud means the potential for cloud security issues is everywhere along the chain. It can happen as the data is prepped for migration, during migration, or potentially within the cloud after the data arrives. And you need to be prepared to address this every step of the way.
- Data security has been incumbent on the cloud service providers, and they have risen to the occasion. No matter which platform you select in the debate between AWS vs. Azure vs. Google, all sport various compliances to standards like HIPAA, ISO, PCI DSS, and SOC.
- However, just because the providers offer compliance doesn't give customers the right to abdicate their responsibilities. They have some measure of responsibility as well, which creates a significant cloud computing challenge. So here are eight critical concepts for data security in the cloud.

Privacy Protection

- Your data should be protected from unauthorized access regardless of your cloud decisions, which includes data encryption and controlling who sees and can access what. There may also situations where you want to make data available to certain personnel under certain circumstances. For example, developers need live data for testing apps but they don't necessarily need to see the data, so you would use a redaction solution. Oracle, for example, has a Data Redact tool for its databases.
- The first step is something you should have done already: identify the sensitive data types and define them. Discover where the sensitive data resides, classify and define the data types, and create policies based on where the data is and which data types can go into the cloud and which cannot. Too many early adopters of the cloud rushed to move all their data there, only to realize it needed to be kept on-premises in a private cloud.
- There are automated tools to help discover and identify an organization's sensitive data and where it resides. Amazon Web Services has Macie while Microsoft Azure has Azure Information Protection (AIP) to classify data by applying labels. Third party tools include Tableau, Fivetran, Logikcull, and Looker.

Preserve Data Integrity

- Data integrity can be defined as protecting data from unauthorized modification or deletion. This is easy in a single database, because there is only one way in or out of the database, which you can control. But in the cloud, especially a multi cloud environment, it gets tricky.
- Because of the large number of data sources and means to access, authorization becomes crucial in assuring that only authorized entities can interact with data. This means stricter means of access, like two-factor authorization, and logging to see who accessed what. Another potential means of security is a Trusted Platform Module (TPM) for remote data checks.

Data Availability

- Downtime is a fact of life and all you can do is minimize the impact. That's of considerable importance with cloud storage providers because your data is on someone else's servers. This is where the

Service-Level Agreement (SLA) is vital and paying a close eye to details really matters.

- For example, Microsoft offers 99.9% availability for major Azure storage options but AWS offers 99.99% availability for stored objects. This difference is not trivial. Also, make sure your SLA allows you to specify where the data is stored. Some providers, like AWS, allow you to dictate in what region data is stored. This can be important for issues of compliance and response time/latency.
- Every cloud storage service has a particular strength: Amazon's Glacier is ideal for mass cold storage of rarely accessed data, Microsoft's Azure blob storage is ideal for most unstructured data, while Google Cloud's SQL is tuned for MySQL databases.

Data Privacy

- A huge raft of privacy laws, national and international, have forced more than a few companies to say no to the cloud because they can't make heads or tails of the law or it's too burdensome. And it's not hard to see why.
- Many providers may store data on servers not physically located in a region as the data owner and the laws may be different. This is a problem for firms under strict data residency laws. Not to mention that the cloud service provider will likely absolve themselves of any responsibility in the SLA. That leaves the customers with full liability in the event of a breach.
- As said above, there are national and international data residency laws that are your responsibility to know. In the U.S. that includes the Health Information Portability and Accountability Act (HIPAA), The Payment Card Industry Data Security Standards (PCI DSS), the International Traffic in Arms Regulations (ITAR) and the Health Information Technology for Economic and Clinical Health Act (HITECH).
- In Europe you have the very burdensome General Data Protection Regulation (GDPR) with its wide ranging rules and stiff penalties, plus many European Union (EU) countries now that dictate that sensitive or private information may not leave the physical boundaries of the country or region from which they originate. There are also the United Kingdom Data Protection Law, the Swiss Federal Act on Data Protection, the Russian Data Privacy Law and the Canadian Personal Information Protection and Electronic Documents Act (PIPEDA).
- All of these protect the interest of the data owner, so it is in your best interest to know them and know how well your provider complies with them.

Encryption

- Encryption is the means for which data privacy is protected and insured, and encryption technologies are fairly mature. Encryption is done via key-based algorithms and the keys are stored by the cloud provider, although some business-related apps, like Salesforce and Dynamix, use tokenization instead of keys. This involves substituting specific token fields for anonymous data tokens.
- Virtually every cloud storage provider encrypts the data while it is in transfer. Most do it through browser interfaces, although there are some cloud storage providers like Mega and SpiderOak that use a dedicated client to perform the encryption. This should all be spelled out in the SLA.
- Many cloud services offer key management solutions that allow you to control access because the encryption keys are in your hands. This may prove to be a better or at least more reassuring risk because you are in control over who has the keys. Again, this should be spelled out in the SLA.

Threats

- If you are online you are under threat of attack, that is a fact of life. The old style of attacks, like DDoS attacks, SQL injection, and cross-site scripting, have faded into new fears. Cloud service providers have a variety of security tools and policies in place but problems still happen, usually originating in human error.
  - ➢ **Data Breaches:** This can happen any number of ways, from the usual means a hacked account or a lost password/laptop to means unique to the cloud. For example, it is possible for a user on one virtual machine to listen for the signal that an encryption key has arrived on another VM on the same host. It's called the "side channel timing exposure," and it means the victim's security credentials in the hands of someone else.
  - ➢ **Data Loss:** While the chance of data loss is minimal short of someone logging in and erasing everything, it is possible. You can mitigate this by insuring your applications and data are distributed across several zones and you backup your data using off-site storage.

- **Hijacked Accounts:** All it takes is one lost notebook for someone to get into your cloud provider. Secure, tough passwords and two-factor authentication can prevent this. It also helps to have policies that look for and alert to unusual activity, like copying mass amounts of data or deleting it.
- **Cryptojacking:** Cryptojacking is the act of surreptitiously taking over a computer to farm cryptocurrency, which is a very compute-intensive process. Cryptojacking spiked in 2017 and 2018 and the cloud was a popular target because there is more compute resources available. Monitoring for unusual compute activity is the key way to stop this.

Data Security and Staff

- Most employee-related incidents are not malicious. According to the Ponemon Institute's 2016 Cost of Insider Threats Study, 598 of the 874 insider related incidents in 2016 were caused by careless employees or contractors.
- However, it also found 85 incidents due to imposters stealing credentials and 191 were by malicious employees and criminals. Bottom line: your greatest threat is inside your walls. Do you know your employees well enough?

Contractual Data Security

- The SLA should include a description of the services to be provided and their expected levels of service and reliability, along with a definition of the metrics by which the services are measured, the obligations and responsibilities of each party, remedies or penalties for failure to meet those metrics, and rules for how to add or remove metrics.
- Don't just sign your SLA. Read it, have a lot of people read it, including in-house attorneys. Cloud service providers are not your friend and are not going to fall on their sword for liability. There are multiple checkmarks for a SLA.
  - Specifics of services provided, such as uptime and response to failure.
  - Definitions of measurement standards and methods, reporting processes, and a resolution process.
  - An indemnification clause protecting the customer from third-party litigation resulting from a service level breach.
- This last point is crucial because it means the service provider agrees to indemnify the customer company for any breaches, so the service provider is on the hook for any third-party litigation costs resulting from a breach. This gives the provider a major incentive to hold up their end of the security bargain.

2.10.2 Network Security

- Network security deals with protecting the computer network from any type of unauthorized access. The important terminologies in network security are: Access Control List De-Militarized Zone Virtual Private Network Intruders Intrusion Detection System Distributed Denial-Of-Service Ping of Death Ping Flood Attack In the next slide, let us understand the first terminology which is Access Control List.

Access Control List

- Network security is one of the most significant steps to be initiated in a production environment. It takes care of authorization, authentication, and accounting. Access Control Lists (ACLs) are the network filters used by some switches and routers to allow and limit the flow of data in and out of network interfaces. They protect high-speed interfaces where the line rate speed is significant and the firewalls are constricting. ACLs limit the updates from the network peers for routing. They define the flow control for the network traffic. To filter the traffic against some known vulnerable protocols and less desirable networks, ACLs should be placed on external routers. ACLs for routers offer a noteworthy amount of firewall capability. During the configuration of an ACL on an interface, the network device examines the data passed through the interface, compares it with the described criteria in the ACL, and permits or prohibits the data to flow. Setting up a DMZ or de-militarized zone in the network is one of the most common methods in network security. To implement DMZ architecture, two separate network devices are used. We will discuss this in the next slide.

De Militarized Zone

- In networking, De-Militarized Buffer Zone (DMZ) is named after the de-militarized zones, the prohibition area for the military of either countries. DMZ, a secure

server, provides an additional layer of security to the network that works as a barrier between a Local Area Network (LAN) and a less secure network, like the Internet. It provides secure services to LAN users who require the Internet for web applications, FTP, email, and other applications. We will continue our discussion on DMZ in the next slide.

- As it provides buffer between LAN and the Internet, the DMZ management server network is considered a less secure network compared to the internal network. Therefore, few additional security measures are taken for a DMZ host, which include disabling unnecessary services, running the necessary services with the reduced privileges, eliminating any unnecessary user account, and ensuring the DMZ has the latest security patches and updates. Though few computers have access to the DMZ, it will compromise the entire network in case there is a security breach. It would be otherwise, if the entire network could be accessed through the Internet. Most IT professionals place systems in DMZ which can be accessed from outside, for example, web servers, DNS servers, and VPN systems or remote access. Next, we will discuss DMZ configuration.

De Militarized Zone Configuration

- The steps to configure DMZ is given on the slide. The external router provides access to connections outside the network. The DMZ internal router contains restrictive ACLs that protect the internal network from well-defined threats. ACLs are often configured with explicit permit and they deny the statements for specific addresses and protocol services. The ACLs are less restrictive, however, it provides large protection access blocks to global routing table areas that is to be restricted. It protects from well-known protocols that provide access to the network or outside the network. In addition, ACLs should be configured to limit the peer access network and can be used in combination with the routing protocol. This should be done to limit the extent of routes that the network peers received or sent and to confine the number of updates. In the following slide, we will learn about another network security concept – VPN.

Virtual Private Network

- A Virtual Private Network (VPN) uses a public connection, the Internet, to provide access to remote offices and users to the network of their organization. To access public network as a private network, it employs the mechanism of authentication, encryption and integration protection. VPN can connect distant networks of an organization, for instance, it can allow the employee travelling abroad access the organization's intranet, i.e., private network, remotely. It can also create a private network over the public network, for instance, the Internet. This depends on the use of virtual connections, which are temporary and do not have any presence, made of packets.
- These networks should be physically separate from each other and the user needs to connect the networks using VPN. In such case, the user sets up two firewalls, Firewall1 and Firewall2, for encryption and decryption purposes. The Network 1 connects to the Internet via Firewall1, and Network 2 connects to the Internet via Firewall2. Note the two firewalls are virtually connected to each other through the Internet with the help of a VPN tunnel. Let us understand how the VPN protects the traffic that passes between two hosts on different networks. Assume that host X of Network 1 sends a data to host Y of Network 2. The host X creates a packet, inserts its IP address as the source address and IP address of host Y as the destination address. The packet reaches Firewall that adds new headers to it. In the new header, Firewall alters the source address to that of Firewall 1, say, FA1. Similarly, it alters destination address of host Y to that of Firewall 2, say FA2. It encrypts and authenticates the packet, as per the settings and forwards the modified packet over the Internet. The packet reaches to Firewall 2 via one or more routers. The Firewall 2 then discards the outer header and performs decryption and other cryptographic functions as per the requirement. This in turn gives the original packet, created by host X. While going through the packet contents, Firewall 2 realizes packet was addressed to host Y. Therefore, it delivers the packet to host Y. Next, we will discuss the objectives and classes of intruders.

Intruders

- The general meaning of an intruder is a person who enters the territory that does not belong to him/her. An intruder's aim is to access the system or increase the range of privileges accessed on a system. It is in fact the most publicized of all threats to security. The intruder attack may or may not harm the infrastructure. However, it is the responsibility of a network administrator to perform and configure the monitoring system, which can detect any type of intrusion. This is where an Intrusion Detection System (IDS) comes into picture. In the next slide, we will continue the discussion on network security concepts and learn in detail about IDS. In the next slide, we will discuss the types of intruders.

Types of Intruders

- There are three classes of intruders which are explained as follows:
  1. The first is the masquerader: Masquerader is the one who seeks unauthorized access to the system to use a legal user's account.
  2. The second intruder is the misfeasor: Misfeasor is a legitimate user who seeks access to areas that are unauthorized to him/her, misusing the privileges.
  3. The third type of intruder is the clandestine or a secret user: Clandestine is a user who can exert managerial control over the system and use it to avoid access and auditing controls or to suppress the collection of audit.
- Usually, the masquerader is an outsider and the misfeasor is an insider. Clandestine can be either an insider or an outsider. We will learn about an Intrusion Detection System in the next slide.

Intrusion Detection System

- An Intrusion Detection System (IDS) is responsible for inspecting all outbound and inbound activities. It identifies any doubtful pattern when a system or a network is attacked by someone who tries to break in. It performs a variety of functions, which include monitoring the user and system activities, auditing system configurations for vulnerabilities and misconfigurations, accessing the integrity of critical system and data files, recognizing known attack patterns in the system activity, and identifying abnormal activities through statistical analysis. In the following slide, we will discuss the types of IDS.

Types of Intrusion Detection System

- There are two types of IDS:
  1. Host based IDS (HIDS)
  2. Network Based IDS (NIDS)
- Which will be covered in the subsequent slides. Let us begin with the first type, i.e., HIDS, in the next slide.

1. Host Based Intrusion Detection System

- A Host-Based Intrusion Detection System (HIDS) checks on the log files, audit trails, and network traffic that enter or exit the host. HIDS can operate both in real time as the activity arises and in batch mode by checking on a periodic basis. Generally, host-based systems are self-contained, that is, every essential element that needs to be secured is available. Many new commercial products are designed to both report and be managed by a central system. Local system resources are used by these systems to operate. In the following slide, we will discuss between the old versus new HIDS.

Old Versus New Host Based Intrusion Detection System

- The table on the slide distinguishes between old and new host-based intrusion detection system. The older versions of HIDS were operated in batch mode. It looked for suspicious activities in particular events in the system's log files on an hourly or daily basis. However, in the new versions of HIDS, the processor speed is increased and IDS performs a real-time check on the log files. They also have the ability to examine the data traffic, which is generated and received by the host. Many HIDSs focus on the log files or the audit trails, which are produced by the local operating system. The examined logs are the application, system, and security event logs on the Windows systems. Whereas on the UNIX system, the examined logs are usually the message, kernel, and error logs. Some HIDSs have the ability to consistently monitor, either on a single application (payroll application) or on a dedicated protocol (FTP or HTTP, etc.). HIDSs search activities in the log file, like, use of certain programs; login authentication failure; adding new user account; modification or access of critical system files; logins at odd hours; starting or stopping processes; privilege

escalation; and modification or removal of binary files. Next, we will look into the components of HIDS.

Components of Host Based Intrusion Detection System

- HIDS contains various components as follows: Critical files contains information references about the critical files, for example, OS files. Log files are generated as a report by the Host Operating System. Network traffic component accepts the network traffic. Traffic collector is used to collect the traffic that is generated from any of the three components, that is, log files, critical files, and network traffic. Traffic analyzer is used to analyze the traffic and find the signature, which is then matched with the signature database. The signature database contains all malicious signatures. It is the heart component of the IDS. If any signature matches with the signature database, the traffic analyzer triggers the alarm, flashes the warning box in the user interface, and maintains the log of the threat. Threat is usually stored in the report. Next, we will look into NIDS.

2. Network Based Intrusion Detection System

- Network-Based Intrusion Detection System (NIDS) monitors the network traffic that interconnects the system, such as the bits and bytes travelling along the cables and wires. It analyzes the traffic as per the protocol, destination, type, source, content, amount, traffic already seen, and so on, that occurs quickly. To be effective, the IDS handles the traffic at speed on which the network operates. Network-based IDSs are generally deployed so that they can monitor traffic in and out of the organization's major links, like, connections to the Internet, remote offices, partner, etc. NIDS searches activities like, malicious content in the data payload of packet/packets; vulnerability scanning; Trojans, viruses, or worms; denial of service attacks; tunneling; port scans; and brute-force attacks. We will discuss NIDS layout in the next slide.

Network Based Intrusion Detection System Layout

- The working of NIDS is similar to that of HIDS; however, it does not scan critical files or log files. Let us discuss the Distributed Denial-of-Service (DDoS) in the next slide.

Distributed Denial of Service

- A Distributed Denial-of-Service (DDoS) attack affects the availability of the services to the legitimate user. As shown in the image, the attacker prevents the efficient functioning of the internet site or a service. This can be set either on a temporary basis or for an indefinite period. The motto is to affect the reputation of the service provider. This malfunctioning occurs by collecting all the zombies present in the public network. The zombies are the vulnerable computers in the public network that are breached by the attacker and are in complete control. The attacker then puts forward a request to perform bombing of unlimited packets in the victim's computer through the zombies. Since the attack is performed simultaneously by multiple zombies, it is called distributed denial-of-service attack. Next, we will discuss ping of death attack.

Ping of Death

- Ping of death is a type of attack for denial-of-service, where the attacker sends a packet of size larger than 65536 bytes. Earlier, an operating system could not handle larger packets. Therefore, the attacker would deliberately create a packet of larger size and send it to the target system. Consequently, the operating system would either hang or be restarted. This in turn resulted in the system downtime. However, nowadays all the latest operating systems and NIC cards are being patched for such type of attack. In the following slide, we will discuss ping flood attack.

Ping Flood Attack

- In the ping flood attack, the attacker sends a huge number of "ICMP Echo Requests" to the victim. Here, ICMP stands for Internet Control Message Protocol. Ping flood attack is supported by many ping utilities and the attacker does not have to be highly knowledgeable about it. Since it overloads the network links, it is damaging for both attacker and the victim, unless the attacker has link faster than that of the victim. Filtering the incoming packets can help if the Ethernet speed is flooded. If the speed is slow, the options are less; you can either hang up or reconnect with a different IP address. To filter the incoming ICMP Echo Request packets, the victim can use a firewall. This allows the computer to refuse sending ICMP Echo Reply packets. We will explore some of the best security practices in the next slide.

Best Security Practices

- The best security practices are as follows: Review and audit logs are generated in the server to hunt malicious or suspected activity. As an admin, you need to ensure the encryption of data during storage and transit. This can be implemented using methods like VPN, etc. Obfuscation is a practice of using defined pattern to mask a sensitive data. Obfuscation protects the data stored in the physical storage device. Zoning controls access from isolation of a single server to a group of storage devices or a single storage device and one node to the other. It also associates one or more storage devices with a set of multiple servers. LUN masking provides detailed security than zoning as LUNs allow sharing of storage at the port level. We have learned about the various network security concepts and best practices, so far. In the subsequent slides, we will move on to understand the encryption technologies and methods.

Encryption Technologies and Methods

- Encryption is a process wherein information is coded. This is to ensure that only authorized individuals will be able to read the data contained. The various technologies and methods used for encryption are: Cryptography, Cipher, Public Key Infrastructure, IPSec, SSL Protocol, and TLS Protocol. We will begin with cryptography in the next slide.

Cryptography

- Cryptography is the science of converting the human-readable data into a scrambled data using a pre-defined algorithm and vice-versa. Cryptography provides confidentiality, integrity, and non-repudiation of data. Confidentiality refers to the data being confidential and read only by the intended recipient. Integrity refers to the data not being modified or fabricated during the data exchange process between the sender and the intended receiver. Non-repudiation refers to the proof of the sender being the genuine intended sender. Next, we will discuss the types of cryptography.

Types of Cryptography

- Cryptography can be classified into three types; they are secret-key cryptography, public-key cryptography, and hashing. Secret-key cryptography is also referred to as symmetric-key cryptography. In this, only one key is used for encryption and decryption process. Public-key cryptography is also called as asymmetric-key cryptography. In this, two keys are used, one key is used for encryption, and the other key is used for decryption process. The two keys are referred to as public key of sender, which is used for encryption; and private key of sender, which is used for decryption. Hashing is also called as one-way encryption. This is basically used to check the integrity of the data, by checking the integrity of the decryption. In the next slide, we will learn about the differences between plain text and cipher text.

Cipher Text versus Plain Text

- Cipher is an algorithm in cryptography that performs encryption and decryption. In a non-technical usage, a 'cipher' is similar to a 'code'; however, the concepts vary in cryptography. Before we discuss encryption and decryption, let's understand the meaning of a plain text and a cipher text. Plain text is a message or a file that is human-readable, whereas a cipher text is a message or a file that is encrypted. Technically, the process of encoding plain text message into cipher is known as encryption and the reverse is known as decryption. In communication, the computer forwards the encrypted message from the sender's end, which is received over a network, for instance, the Internet, by the receiver. The receiver's computer in turn decrypts the message to obtain the original plain text message. To encrypt the message, sender applies encryption algorithm and to decrypt the message recipient applies decryption algorithm. We will look into some of the examples of cipher in the following slide.

Examples of Cipher

- Data Encryption Standard (DES) is also called as Data Encryption Algorithm (DEA). It was developed in 1977, by National Bureau of Standards (NBS) (US Government) for the secure transmission of data within the systems. DES is a block cipher which uses a private key or a secret key for encryption. There are 72 quadrillion or more possible keys that can be used for this process. Like other private key encryption, both the sender and the receiver must know and use the same private key. Triple DES is an extended version of DES, with a difference that the cipher text generated first time, after using DES, is re-accepted as the input

for cipher text generation for two more times. It makes the process far more secure. Digital Signature Algorithm (DSA) is a standard for digital signature proposed by National Institute of Standards and Technologies and is adopted by FIPS (Federal Information Processing Standard). Advanced Encryption Standard (AES) is another type of secret key cryptography. There are three types of algorithms in AES; they are AES128, AES192, and AES256. AES is created to remove some flaws that were found in the DES algorithm. Later, this algorithm was accepted as standard for defense communications. Rivest-Shamir-Adleman (RSA) algorithm is a public key cryptography where the keys for encryption and decryption are different. The sender uses the receiver's public key to encrypt the message and the receiver uses his/her private key to decrypt the message. RC4 is a type of cipher that Rivest created for the purpose of securing data. RC4 is based on total random permutation of the data to be protected. It is used for file-encryption products, like password managers. RC5 is a type of cipher, which has total block size of 32, 64, or 128 bits unlike RC4. RC5 uses the concept of data-dependent rotations for performing cryptographic operations. We will discuss Public Key Infrastructure in the next slide.

Public Key Infrastructure

- A Public Key Infrastructure (PKI) is made up of different components like hardware, applications, policies, services, programming interfaces, cryptographic algorithms, protocols, users, and utilities. Such components work together and allow communications using public key cryptography and symmetric keys for digital signatures, data encryption, and integrity. There is no need of constructing and implementing a PKI application and protocol because the same type of functionality is provided by different application and protocols. We will continue our discussion on PKI.
- It is similar to that of issuing Social Security Number. It is issued by the formal authority called Social Security Administration. When User A applies for issuing certificate, the registration authority will ask for a proof of identity and validate it. Once the proof has been validated, it requests certification authority for issuing the certificate. The certificate authority will use the private key to sign the certificate digitally. When User B receives User A's certificate and verifies that it was signed digitally by a certificate authority that he or she trusts, then he or she will believe the certificate to be valid and not because he or she trusts User A. This is referred to as a third-party model. The process allows User A to authenticate himself to User B and communicate with User B through encryption process without prior communication or a pre-existing relationship. Once User B is convinced of the legitimacy of User A's public key, he or she can use it to encrypt and decrypt messages between himself or herself and User A. The job role of Validation Authority is to validate the user in the public key infrastructure and provide the result to the user who needs verification. Let us next discuss IPSec.

IPSec

- IPSec (Internet Protocol security) is a set of protocols developed by the Internet Engineering Task Force (IETF) for the secure exchange of packets at the network layer of the OSI model. This protocol works only in combination with IP networks. It is possible to tunnel across other networks at lower levels of the OSI model, once an IPSec connection is established. The set of security services, which is provided by IPSec, takes place at the network level of the OSI model. Because of this, the higher-level protocols, like TCP, UDP, BGP, etc., are not affected by the implementations of IPSec services. The IPSec protocol is designed to provide a comprehensive array of services; but it is not limited to connectionless integrity, access control, rejection of replayed packets, traffic-flow confidentiality, data origin authentication and data security, etc. In the following slide, we will learn about the two modes of IPSec.

IPSec Modes

- There are two modes of IPSec. They are transport mode and tunnel mode. The transport mode method encrypts the data portion of the packet only, which enables an outsider to see the source and destination IP addresses. Protection of data portion of the packet is referred to as content protection. The tunnel mode protects source and destination IP addresses as well as data. Though this provides the greatest security, it is possible only between IPSec servers, since the final destination needs to be known for delivery. Protecting

header information is known as target protection. Different security levels are provided by these methods. It is possible to use both methods at the same time, such as, using transport within one's own network to reach an IPSec server, and using the transport method from the target network's IPSec server to the target host. We will discuss the SSL protocol in the next slide.

SSL Protocol

- SSL was first developed by Netscape. Later, they gained popularity among other companies like Microsoft and became a compulsory standard until TLS (Transport Layer Security) was evolved. SSL uses public key infrastructure concepts. It uses a program layer which is located between the Internet's Transport Control Protocol (TCP) and Hypertext Transfer Protocol (HTTP) layers. Netscape's SSL Ref program library can enable any web server. It is available for download for non-marketable use or is licensed for commercial use. SSL protocol is used to provide a secure communication interface between the user's web browser and the server. It provides two levels of security services, namely, authentication and confidentiality. The main feature of providing secure communication helps in maintaining the confidentiality of the data. It verifies the user by performing the authentication process. Next, we will discuss two important concepts of SSL.

Concepts of SSL

- Following are the two important concepts of SSL:
  1. **First is the SSL Connection:** It is a transport, which provides a suitable type of service, for instance, peer-to-peer relationships. SSL connections are transitory and each connection is related with a single session.
  2. **Second concept is the SSL Session:** In this session, a client and a server are associated. They are created by Handshake Protocol. SSL Session defines a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid repeated authentication to maintain connectivity, since repeated authentication requires high bandwidth of network, which may result in slower service. We will learn more about the TLS protocol in the next slide.

TLS Protocol

- Transport Layer Security (TLS) protocol, the successor to the Secure Sockets Layer (SSL), ensures privacy between collaborating applications as well as their users on the web. When a client and a server communicate, TLS ensures any third party does not pry or interfere with any message. TLS comprises two layers: the TLS Handshake Protocol and the TLS Record Protocol. The TLS Handshake Protocol allows authentication between the client and the server. It also allows negotiation with cryptographic keys and an encryption algorithm prior to data exchange. The TLS Record Protocol secures connection with encryption method, such as, the Data Encryption Standard (DES). It can be used without encryption. The TLS protocol is based on Netscape's SSL 3.0 protocol; however, SSL and TLS are not interoperable. The TLS protocol does not contain a mechanism that allows TLS implementation to back down to SSL 3.0. TLS is supported by the recent browser versions. TLS is an Internet Engineering Task Force (IETF) standardization initiative whose goal is to produce an SSL internet standard version. In the following slide, we will discuss Discretionary Access Control Methods.

Discretionary Access Control Method

- The DAC method was formerly used by the military to describe the two approaches to control system access. One approach is user-based and the other is machine-based. User-based approach refers to the access right, mapped with the individual's assigned username. Machine-based approach refers to the access right mapped with either IP address or MAC address of the machine. Thus, in machine-based approach, the access is given with respect to the machine identification and not user identification. It is a preferred practice in DAC to follow user-based approach. DAC is meant to limit the access to objects; it is based on the identification of groups or subjects to which they belong. The controls are discretionary. "If a system has discretionary access control, the owner of an object decides on the subjects that will have access, also the specific access they will be given." The permission used in UNIX-based systems is the common method to accomplish this. The owner of a file can specify what permissions (read/ write/ execute) can be given to the

members of the same group and the permissions all others may have. We will learn more about the second method of access control, that is, MAC in the next slide.

Mandatory Access Control

- Like DAC, Mandatory Access Control (MAC) was originally used by military to describe the two approaches used to control the access an individual had on a system. MAC system is used in environments where different levels of security classifications exist; and is more restrictive of user independence to work. MAC restricts the access to objects based on the sensitivity of the information contained in the objects as well as the formal authorization of subjects. In MAC, the operating system decides whether to grant access to another subject, not the owner or subject. In this system, the security mechanism controls how the objects are accessed. This access cannot be changed by any individual subject. Here, the label attached to each object and subject is the key. The label will identify the classification level for that specific subject and object that is entitled to. We will discuss the third method of access control, that is, RBAC in the next slide.

Role Based Access Control

- In RBAC, instead of being assigned permissions for specific actions for the objects associated with the computer system or network, the user is assigned a set of responsibilities that he or she is expected to perform. The roles are sequentially assigned with access authorizations, which are required to perform the tasks related with the role. Thus, permissions to the objects in terms of their specific duties will be granted to the users. The advantage of RBAC is it encapsulates all the access needed by an entity in one set, called 'role'. This makes it easier to establish or remove access for an entity as the specific access needs are easily identified. RBAC does not provide much flexibility. A certain entity is bound to the access provided by the role they are in. The access needs of an entity is most often determined by several exceptions. It rarely occurs that large entity groups need the exact access. We will look into other access control methods in the next slide.

Other Access Control Methods

- The other access control methods are multifactor authentication, single sign-on, and federated server which will be covered in the subsequent slides. Let us begin with multifactor authentication.

Multifactor Authentication

- Multi-factor authentication is a security system that verifies the validity of a transaction. It is more than one form of authentication. It is an important function within the organization since it protects the users' identities, secures corporate network accesses, and ensures the users' authenticity. The aim of multifactor authentication is to increase the level of security, since more than one mechanism would have to be spoofed for an unauthorized individual to gain access to a computer system or network. Let us discuss the single sign-on method in the next slide.

Single Sign On Method and Federated Server

- Single sign-on is a feature introduced in the cloud in which a user can use the credentials provided by the federated server and can access the multiple service providers' website, without officially registering in the website. Nowadays, almost in all the updated website, the user can authenticate himself or herself with the help of their Google accounts or Facebook accounts. Let's take a look at the website: www.4shared.com as an example. This website provides the user with an option to login by using the credentials of Google or Facebook or Twitter accounts. The intermediate server which provides token to the user to access service provider's website directly is called federated server. One of the practical examples is the use of open-id. Open-id is the name of the product which is free to be used by the web developers. Open-id enables the API to perform single sign-on process for the developer's website. You can get details on open-id on the website www.openid.net. In the next slide, we will see the single sign-on and federated server.

Working of the Single Sign On and Federated Server

- The working of the single sign-on and federated server is shown in the image on the slide. First, the user sends a request to the service provider to authenticate by using other service provider's credentials. Next, the service provider redirects the user to the associated federated server to get a token. The federated server

provides the login page where the user provides his or her credentials. After that, the federated server validates the user's credentials and provides token with respect to the service provider. The user then tries to access service provider through the token that he or she has received. The service provider further verifies the token from the federated server. Once the federated server sends a positive feedback, the user gets the access to the service provider server. In the next slide, we will move on to understand the guest and host hardening techniques.

Guest and Host Hardening Techniques

- Hardening refers to providing security to the systems. To avoid any malicious activity from the attackers, it is the best way to secure the servers. Though it is impossible to make any system 100% secure, this technique will reduce the possibility of high security threat to a system. The first technique is to disable all the unwanted ports. The ports are the entry points to the system. It is recommended that only those ports which are essential should be kept open while others should be closed. However, there are situations where a system administrator cannot close a free port. In such situations, use the second technique. In the second technique, the users can use the firewalls to monitor the traffic. The firewall can perform various activities like deciding what should be denied and what should be accessible. The firewall can monitor the system based on ports, services, protocols, signature patterns, etc. The next method is changing the default passwords. Whenever an operating system or software is developed, the developer may create dummy or default accounts for beta testing. However, when a user installs the software, he or she is in a situation where he or she is not aware of the default account. The attacker initially tries to use the default passwords to bypass the credentials. Therefore, it is ideal to disable all the default accounts. The next method is using antivirus software. The antivirus software helps the system to run smoothly and also to determine if any malicious program is running. The next method is the patching process. It is essential that the operating system and software must be patched and updated regularly. This helps a system to stay protected with all the threats that have been found after the release of software or operating system. The next method is disabling default user accounts. Almost every software and operating system will have default user accounts to provide simplicity to the customer. However, these default accounts may be used by an attacker to compromise the system. Thus, it is recommended to deactivate all the default accounts for more security. The last method is to enable user and host authentication when it comes to storage and compute. This will ensure the data or information to be passed to the client only when they are authenticated with user-specific and device-specific codes.

2.10.3 Host Security

- Host security describes how your server is set up for the following tasks:
  - ➢ Preventing attacks.
  - ➢ Minimizing the impact of a successful attack on the overall system.
  - ➢ Responding to attacks when they occur.
- It always helps to have software with no security holes. Good luck with that! In the real world, the best approach for preventing attacks is to assume your software has security holes. As I noted earlier in this chapter, each service you run on a host presents a distinct attack vector into the host. The more attack vectors, the more likely an attacker will find one with a security exploit. You must therefore minimize the different kinds of software running on a server.
- Given the assumption that your services are vulnerable, your most significant tool in preventing attackers from exploiting a vulnerability once it becomes known is the rapid rollout of security patches. Here's where the dynamic nature of the cloud really alters what you can do from a security perspective. In a traditional data center, rolling out security patches across an entire infrastructure is time-consuming and risky. In the cloud, rolling out a patch across the infrastructure takes three simple steps:
  1. Patch your AMI with the new security fixes.
  2. Test the results.
  3. Relaunch your virtual servers.
- Here a tool such as enStratus or RightScale for managing your infrastructure becomes absolutely critical. If you have to manually perform these three steps, the cloud can become a horrible maintenance headache.

- Systems hardening is a collection of tools, techniques, and best practices to reduce vulnerability in technology applications, systems, infrastructure, firmware, and other areas. The goal of systems hardening is to reduce security risk by eliminating potential attack vectors and condensing the system's attack surface. By removing superfluous programs, accounts functions, applications, ports, permissions, access, etc. attackers and malware have fewer opportunities to gain a foothold within your IT ecosystem.
- Systems hardening demands a methodical approach to audit, identify, close, and control potential security vulnerabilities throughout your organization. There are several types of system hardening activities, including:
  - ➢ Application hardening
  - ➢ Operating system hardening
  - ➢ Server hardening
  - ➢ Database hardening
  - ➢ Network hardening
- Although the principles of system hardening are universal, specific tools and techniques do vary depending on the type of hardening you are carrying out. System hardening is needed throughout the lifecycle of technology, from initial installation, through configuration, maintenance, and support, to end-of-life decommissioning. Systems hardening is also a requirement of mandates such as PCI DSS and HIPAA.

Systems Hardening to Reduce the "Attack Surface"

- The "attack surface" is the combination of all the potential flaws and backdoors in technology that can be exploited by hackers. These vulnerabilities can occur in multiple ways, including:
  - ➢ Default and hardcoded passwords
  - ➢ Passwords and other credentials stored in plain text files
  - ➢ Unpatched software and firmware vulnerabilities
  - ➢ Poorly configured BIOS, firewalls, ports, servers, switches, routers, or other parts of the infrastructure
  - ➢ Unencrypted network traffic or data at rest
  - ➢ Lack of privileged access

9 Best Practices for Systems Hardening

- The type of hardening you carry out depends on the risks in your existing technology, the resources you have available, and the priority for making fixes.

1. **Audit Your Existing Systems:**

- Carry out a comprehensive audit of your existing technology. Use penetration testing, vulnerability scanning, configuration management, and other security auditing tools to find flaws in the system and prioritize fixes. Conduct system hardening assessments against resources using industry standards from NIST, Microsoft, CIS, DISA, etc.

2. **Create a Strategy for Systems Hardening:**

- You do not need to harden all of your systems at once. Instead, create a strategy and plan based on risks identified within your technology ecosystem, and use a phased approach to remediate the biggest flaws.

3. **Patch Vulnerabilities Immediately:**

- Ensure that you have an automated and comprehensive vulnerability identification and patching system in place.

4. **Network Hardening:**

- Ensure your firewall is properly configured and that all rules are regularly audited; secure remote access points and users; block any unused or unneeded open network ports; disable and remove unnecessary protocols and services; implement access lists; encrypt network traffic.

5. **Server Hardening:**

- Put all servers in a secure datacenter; never test hardening on production servers; always harden servers before connecting them to the internet or external networks; avoid installing unnecessary software on a server; segregate servers appropriately; ensure super user and administrative shares are properly set up, and that rights and access are limited in line with the principle of least privilege.

6. **Application Hardening:**

- Remove any components or functions you do not need; restrict access to applications based on user roles and context (such as with application controls); remove all sample files and default passwords. Application passwords should then be managed via an

application password management/privileged password management solution, that enforces password best practices (password rotation, length, etc.). Hardening of applications should also entail inspecting integrations with other applications and systems, and removing, or reducing, unnecessary integration components and privileges.

7. **Database Hardening:**

- Create admin restrictions, such as by controlling privileged access, on what users can do in a database; turn on node checking to verify applications and users; encrypt database information both in transit and at rest; enforce secure passwords; introduce role-based access control (RBAC) privileges; remove unused accounts;

8. **Operating System Hardening:**

- Apply OS updates, service packs, and patches automatically; remove unnecessary drivers, file sharing, libraries, software, services, and functionality; encrypt local storage; tighten registry and other systems permissions; log all activity, errors, and warnings; implement privileged user controls.

9. **Eliminate Unnecessary Accounts and Privileges:**

- Enforce least privilege by removing unnecessary accounts (such as orphaned accounts and unused accounts) and privileges throughout your IT infrastructure.

Benefits of Systems Hardening

- Systems hardening recovers continuous effort, but the diligence will pay off in substantive ways across your organization via:
  - ➢ **Enhanced System Functionality:** Since fewer programs and less functionality means there is less risk of operational issues, misconfigurations, incompatibilities, and compromise.
  - ➢ **Significantly Improved Security:** A reduced attack surface translates into a lower risk of data breaches, unauthorized access, systems hacking, or malware.
  - ➢ **Simplified Compliance and Auditability:** Fewer programs and accounts coupled with a less complex environment means auditing the environment will usually be more transparent and straightforward.

2.10.4 Compromise Response

- Because you should be running an intrusion detection system, you should know very quickly if and when an actual compromise occurs. If you respond rapidly, you can take advantage of the cloud to eliminate exploit-based downtime in your infrastructure.
- When you detect a compromise on a physical server, the standard operating procedure is a painful, manual process:
  - ➢ Remove intruder access to the system, typically by cutting the server off from the rest of the network.
  - ➢ Identify the attack vector. You don't want to simply shut down and start over, because the vulnerability in question could be on any number of servers. Furthermore, the intruder very likely left a rootkit or other software to permit a renewed intrusion after you remove the original problem that let him in. It is therefore critical to identify how the intruder compromised the system, if that compromise gave him the ability to compromise other systems, and if other systems have the same vulnerability.
  - ➢ Wipe the server clean and start over. This step includes patching the original vulnerability and rebuilding the system from the most recent uncompromised backup.
  - ➢ Launch the server back into service and repeat the process for any server that has the same attack vector.
- This process is very labor intensive and can take a long time. In the cloud, the response is much simpler.
- First of all, the forensic element can happen after you are operating.
- You simply copy the root file system over to one of your block volumes, snapshot your block volumes, shut the server down, and bring up a replacement.
- Once the replacement is up (still certainly suffering from the underlying vulnerability, but at least currently uncompromised), you can bring up a server in a dedicated security group that mounts the compromised volumes. Because this server has a different root file system and no services running on it, it is not compromised.

UNIT III

DEFINING THE CLOUDS FOR ENTERPRISE

3.1 INTRODUCTION

- Anything as a Service (XaaS) is a cloud computing term for the extensive variety of services and applications emerging for users to access on demand over the Internet.

Fig. 3.1 : XaaS – Anything as a Service

- XaaS term refers to delivery of anything as a service. In this model of cloud computing products, tools and technologies are delivered to users as a service over a network; typically the Internet, rather than on-premises.

Example of XaaS

- Every single day, many services are getting added in XaaS. The most common encompass the three general cloud computing models:
  1. Software as a Service (SaaS).
  2. Platform as a Service (PaaS).
  3. Infrastructure as a Service (IaaS).
- We have seen services and advantages of above three cloud service models.
- There are many other examples of XaaS as following.
  - ➢ Network as a Service.
  - ➢ Storage as a Service.
  - ➢ Database as a Service.
  - ➢ Information as a Service.
  - ➢ Integration as a Service.
  - ➢ Security as a Service.
  - ➢ Malware as a Service.
  - ➢ Disaster Recovery as a Service(DRaaS).
  - ➢ Communication as a Service.

3.2 STORAGE AS A SERVICE

- Storage as a Service (SaaS) is a cloud business model in which a company leases or rents its storage infrastructure to another company or individuals to store data.
- Small companies and individuals often find this to be a convenient methodology for managing backups, and providing cost savings in personnel, hardware and physical space.
- As an alternative to storing magnetic tapes offsite in a vault, IT administrators are meeting their storage and backup needs by Service Level Agreements (SLAs) with an SaaS provider, usually on a cost-per-gigabyte-stored and cost-per-data-transferred basis. The client transfers the data meant for storage to the service provider on a set schedule over the SaaS provider's wide area network or over the Internet.

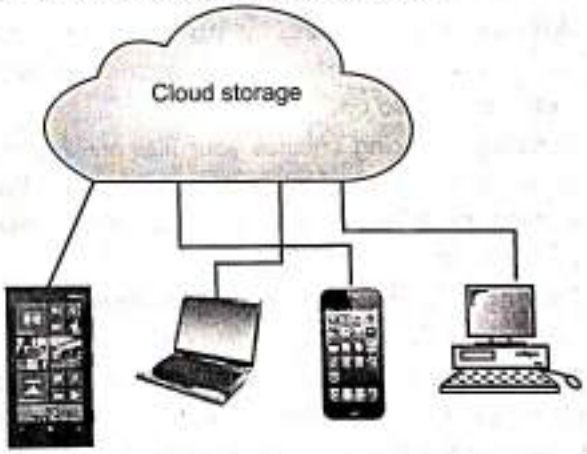


Fig. 3.2

- The storage provider provides the client with the software required to access their stored data. Clients use the software to perform standard tasks associated with storage, including data transfers and data backups. Corrupted or lost company data can easily be restored.
- Storage as a service is prevalent among small to mid-sized businesses, as no initial budget is required to set up hard drives, servers and IT staff. SaaS is also

marketed as an excellent technique to mitigate risks in disaster recovery by providing long-term data storage and enhancing business stability.

- Storage as a service is fast becoming the method of choice to all small and medium scale businesses. This is because storing files remotely rather than locally boasts an array of advantages for professional users.

Top Ten Advantage of Storage as a Services

1. **Cost :** factually speaking, backing up data isn't always cheap, especially when take the cost of equipment into account. Additionally, there is the cost of the time it takes to manually complete routine backups. Storage as a service reduces much of the cost associated with traditional backup methods, providing ample storage space in the cloud for a low monthly fee.
2. **Invisibility :** Storage as a service is invisible, as no physical presence of it is seen in its deployment and so it doesn't take up valuable office space.
3. **Security :** In this service type, data is encrypted both during transmission and while at rest, ensuring no unauthorized user access to files.
4. **Automation :** Storage as a service makes the tedious process of backing up easy to accomplish through automation. Users can simply select what and when they want to backup, and the service does all the rest.
5. **Accessibility :** By going for storage as a service, users can access data from smart phones, netbooks to desktops and so on.
6. **Syncing :** Syncing ensures your files are automatically updated across all of your devices. This way, the latest version of a file a user saved on their desktop is available on your smart phone.
7. **Sharing :** Online storage services allow the users to easily share data with just a few clicks
8. **Collaboration :** Cloud storage services are also ideal for collaboration purposes. They allow multiple people to edit and collaborate on a single file or document. Thus, with this feature users need not worry about tracking the latest version or who has made what changes.
9. **Data Protection :** By storing data on cloud storage services, data is well protected by all kind of catastrophes such as floods, earthquakes and human errors.
10. **Disaster Recovery :** As said earlier, data stored in cloud is not only protected from catastrophes by having the same copy at several places, but can also favor disaster recovery to ensure business continuity.

3.3 DATABASE AS A SERVICE IN CLOUD COMPUTING

- Database as a Service (DBaaS) is a cloud business model in which a company leases or rents Database services to another company or individuals to store their data.
- Database-as-a-Service (DBaaS) is the fastest growing cloud service.

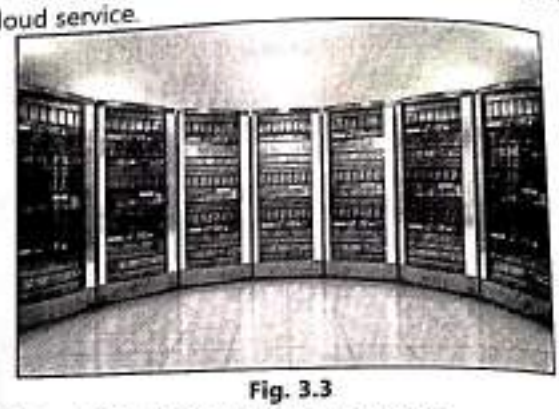


Fig. 3.3

Definition of the Database-as-a-Service

- The term "Database-as-a-Service" (DBaaS) refers to software that enables users to provision, manage, consume, configure, and operate database software using a common set of abstractions (primitives), without having to either know nor care about the exact implementations of those abstractions for the specific database software.
- In other words, a DBaaS user could provision a MySQL database, manage, configure and operate it using the same set of API calls as he (or she) would use if it were an Oracle or MongoDB database. The user would be able, for example, to request a backup of the database using an API call which did the right thing(s) for the database that was being used. Similarly, the user could request a MySQL cluster or a MongoDB cluster, and then resize that cluster using the same API call(s), without having to know exactly how that operation was being performed for each of those database technologies.

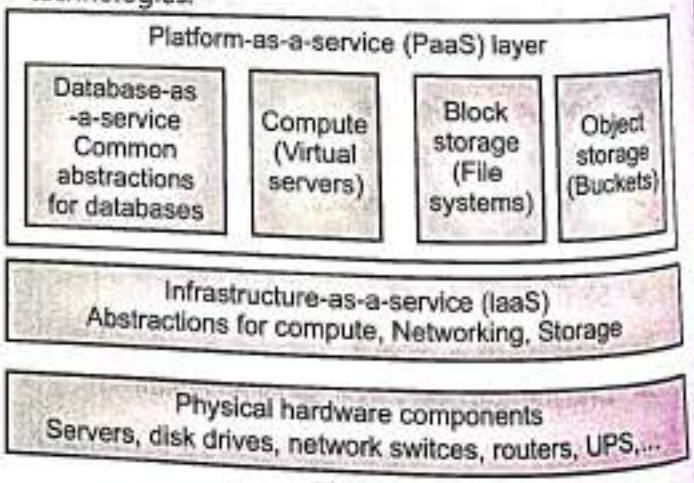


Fig. 3.4

- DBaaS is often considered to be a component of a Platform-as-a-Service, the "platform" in this case being the database (or a number of databases). The DBaaS solution would consume resources of the underlying Infrastructure-as-a-Service (IaaS), for example provisioning compute, storage and networking from that IaaS.

The Benefits of DBaaS

- A DBaaS solution provides an organization a number of benefits, the chief among them being:
  - Developer agility
  - DBA productivity
  - Application reliability, performance and security

Application Reliability, Performance and Security

- Databases are often the "system of record" and are the repository of valuable information in the organization. A database outage could have catastrophic impact. Through automation and standardization, DBaaS ensures that all common workflows involved in the provisioning, configuration, management, and operation of databases are consistent.
- Through this standardization, a DBaaS ensures that all databases are operated in the same way, and in keeping with the best practices established by the IT organization. This frees up the developer and DBA to work on more important things like the application and innovation rather than the boring minutiae of running a database.
- It is important to realize that most enterprises today operate applications that require many different database technologies, a departure from recent years where the 'corporate standard' mandated a single database solution for all application needs. With this diversity in database technologies, DBaaS solutions allow IT organizations to ensure application reliability, performance and data security no matter what database solution is in use, without requiring that the IT organization or the developer team have deep knowledge of the finer points of each of the technologies. DBaaS solutions encapsulate those best practices and codify the proper way(s) to deploy, manage and operate all of the different technologies thereby freeing up the DBAs and developers from these chores.

Comparison of Some DBaaS Solutions

- The most widely used DBaaS in the market today is Amazon Relational Database Service (RDS). RDS provides support for a number of databases including MySQL, MariaDB, Oracle, PostgreSQL and SQL Server. In addition, Amazon also provides Aurora and DynamoDB. Aurora is a scalable relational database compatible with MySQL or PostgreSQL while DynamoDB is a scalable NoSQL database.
- Microsoft offers SQL Database as part of the Azure Cloud platform, and Google offers Cloud SQL, a fully managed MySQL database service.
- With the exception of DynamoDB, all of these are DBaaS solutions that provide management abstractions but no data API. The application that uses the database interacts directly with the managed database in these cases. In DynamoDB however, the service offers a data API as well.
- In the OpenStack ecosystem, the Trove project offers a DBaaS that supports a number of relational and non-relational database packages including most commonly used FOSS databases.
- The value of a DBaaS comes through the standardization of the abstractions, and through the common API. Since the most widely used Cloud API in the world today is Amazon's AWS API , there is considerable value in implementing a solution that exposes its services using that same API. This is the approach that Stratoscale Symphony has adopted. Symphony exposes the same APIs defined by AWS and allows you to provision an AWS region and RDS-compatible DBaaS in your own data center.

3.4 PROCESS AS A SERVICE

- Business Process as a Service, or BPaaS, is a type of Business Process Outsourcing (BPO) delivered based on a cloud services model. BPaaS is connected to other services, including SaaS, PaaS and IaaS, and is fully configurable. BPaaS provides companies with the people, processes and technology they need to operate as a pay-per-use service by making use of the availability and efficiency of a cloud-based system. This approach to operations greatly reduces total cost of ownership by providing an on-demand solution based on services needed as opposed to purchasing a package deal tied into a single application.
- BPaaS keeps companies in lockstep with industry best practices and technology advancements. Companies can also easily increase service levels during peak periods and bring new products and services to market faster with BPaaS's unique operating flexibility and agility.

BPaaS Offers Many Business Benefits, Including:

- **Product/Service Deliverability:** From managing inventory to organizing email and customer records, BPaaS helps companies facilitate the delivery of products and services in an automated, streamlined way with help of cloud technologies. BPaaS is

standardized for use across industries and organizations, so it's flexible and repeatable, resulting in higher efficiency and, ultimately, better service and experience for customers.

- **Cutting Edge at Reduced Cost:** BPaaS provides a business with the latest digital tools, technologies, processes and talent to improve its efficiency, service and the customer experience, without the large capital investment traditionally required. By implementing BPaaS, companies can shift to a pay-per-use consumption model and reduce total cost of ownership.
- **Accommodates Fluctuating Business Needs:** BPaaS can scale on-demand when a company experiences a peak workload. Due to its innate configurability applicable across multiple business areas, and its interaction with other foundational cloud services like SaaS, the service can make use of its cloud foundation to scale to accommodate large fluctuations in business process needs.
- Any business process (for example, payroll, printing, ecommerce) delivered as a service over the Internet and accessible by one or more web enabled interfaces (PC, smart devices and phones) can be considered as a *Business Process as a Service (BPaaS)*. Advertising services such as Google Adsense, IBM Blueworks Live for business process management are some of the several publicly available services, whereas there are number of other services that today IT departments provide to their users within the firewall or to the trusted partners.
- When these services are delivered from the cloud, it changes the economic dynamics. Particularly from the user perspective, one moves away capital expenditure (CAPEX) discussion to the operational expenditure (OPEX) discussion during business case calculations. This in the IT world is a paradigm shift, shift towards utility computing, where you pay as you go even within the walls of the enterprise and extended value chain (the shift to which we are now used to in the consumer world with many services available through apps on the smartphones that we use to manage many business or family processes).
- When one looks at the BPaaS in the Common Cloud Management Platform Reference Architecture (Fig. 3.5) proposed by IBM to The Open Group, the BPaaS sits above Software as a Service (SaaS) layer. In my opinion BPaaS is about intelligently consuming services from the SaaS, PaaS, and IaaS layer.

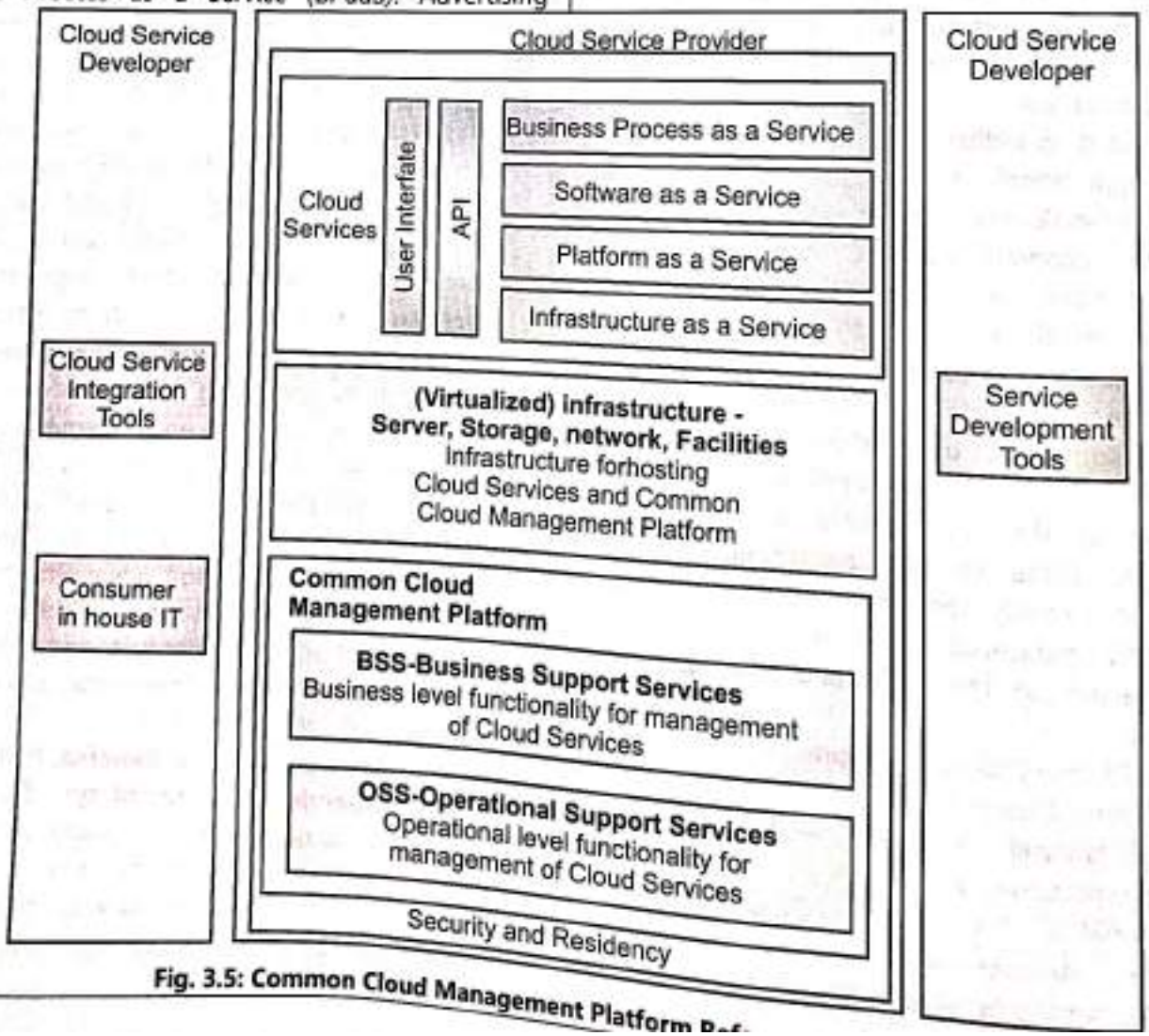


Fig. 3.5: Common Cloud Management Platform Ref...

standardized for use across industries and organizations, so it's flexible and repeatable, resulting in higher efficiency and, ultimately, better service and experience for customers.

- **Cutting Edge at Reduced Cost:** BPaaS provides a business with the latest digital tools, technologies, processes and talent to improve its efficiency, service and the customer experience, without the large capital investment traditionally required. By implementing BPaaS, companies can shift to a pay-per-use consumption model and reduce total cost of ownership.
- **Accommodates Fluctuating Business Needs:** BPaaS can scale on-demand when a company experiences a peak workload. Due to its innate configurability applicable across multiple business areas, and its interaction with other foundational cloud services like SaaS, the service can make use of its cloud foundation to scale to accommodate large fluctuations in business process needs.
- Any business process (for example, payroll, printing, ecommerce) delivered as a service over the Internet and accessible by one or more web enabled interfaces (PC, smart devices and phones) can be considered as a *Business Process as a Service (BPaaS)*. Advertising services such as Google Adsense, IBM Blueworks Live for business process management are some of the several publicly available services, whereas there are number of other services that today IT departments provide to their users within the firewall or to the trusted partners.
- When these services are delivered from the cloud, it changes the economic dynamics. Particularly from the user perspective, one moves away capital expenditure (CAPEX) discussion to the operational expenditure (OPEX) discussion during business case calculations. This in the IT world is a paradigm shift, shift towards utility computing, where you pay as you go even within the walls of the enterprise and extended value chain (the shift to which we are now used to in the consumer world with many services available through apps on the smartphones that we use to manage many business or family processes).
- When one looks at the BPaaS in the Common Cloud Management Platform Reference Architecture (Fig. 3.5) proposed by IBM to The Open Group, the BPaaS sits above Software as a Service (SaaS) layer. In my opinion BPaaS is about intelligently consuming services from the SaaS, PaaS, and IaaS layer.

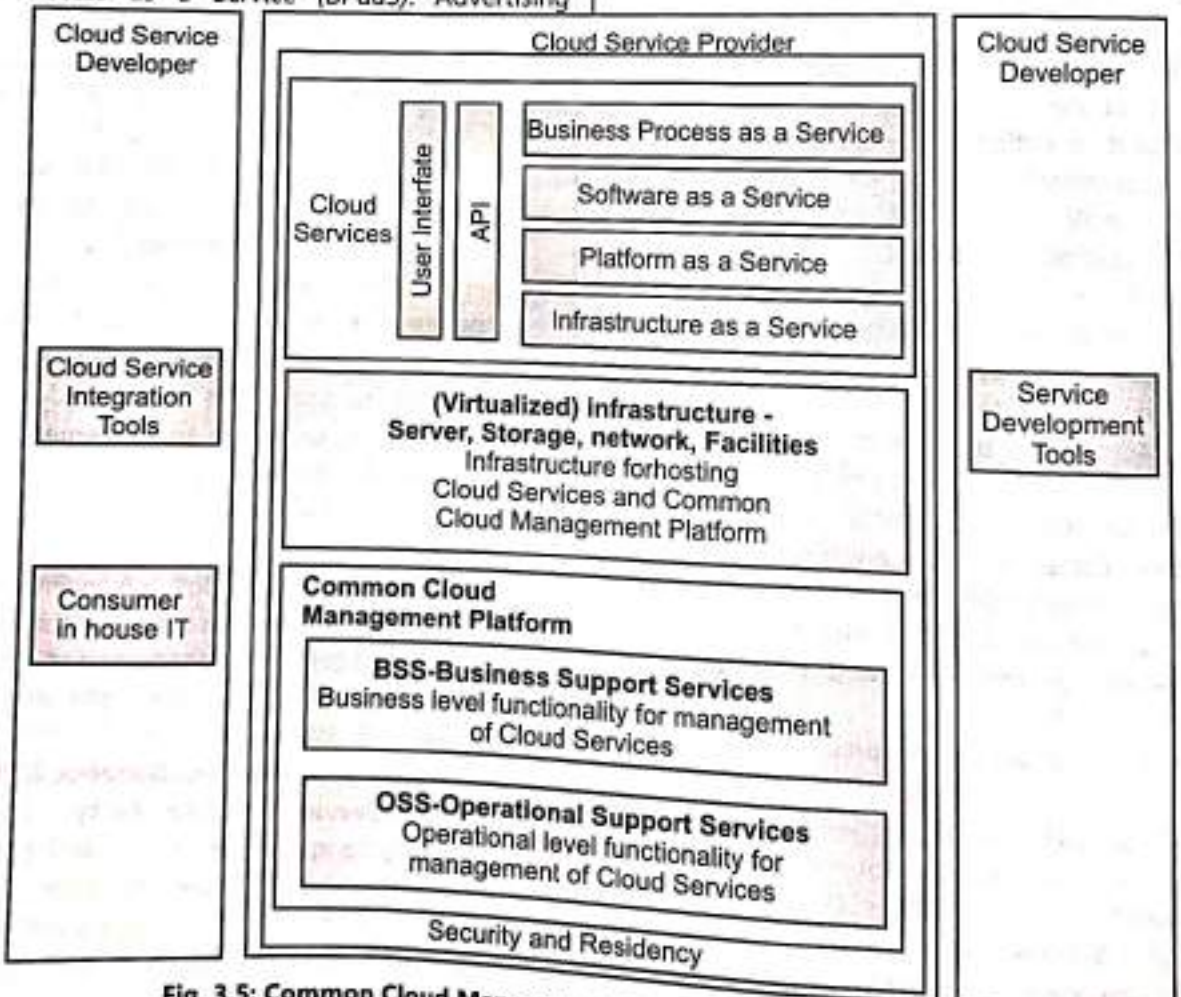


Fig. 3.5: Common Cloud Management Platform Reference Architecture

- Along with the logical economic benefit and ease-of-use of delivering BPaaS from cloud, a key benefit going forward I see is the opportunity a prospective user has to take it out for the proverbial test drive, that is, users can try out various services before integrating in their larger business process or replace an existing service rapidly. The concepts such as virtual desktop make these services consumable from any device. This is a complete paradigm shift with which everybody including internal IT departments will have to live with, it's no more the IT departments that are creating and sourcing the services. Services are now available in the market place that line-of-business executives can search, use for testing, and then decide to buy or keep looking for something else. As depicted in Fig. 3.6, the services consumed by the business process can be sourced from internal private cloud, partners cloud, or some public cloud. Alternatively, a BPaaS through cloud delivery model creates entirely different markets for companies and for the CIO organizations where existing delivery models don't facilitate access to these markets.
  - ➢ What are the key attributes of a winning cloud provider business strategy and model?
  - ➢ How can partnering across the ecosystem accelerate my success?
  - ➢ What are the implications if I do not act now?

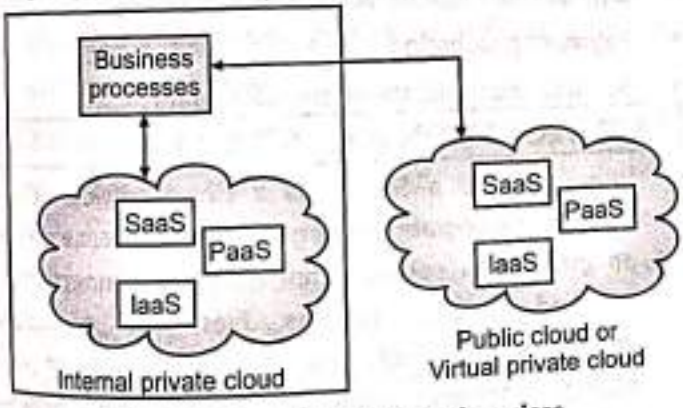


Fig. 3.6: Internal and external services

- Let's look at an example of a BPaaS offering from IBM, namely IBM Blueworks Live. Whether you are involved in simple processes that are today run over email, are managing a small team and want to run more efficiently, or are involved in a corporate BPM program around process modeling and governing, Blueworks Live is an offering that brings everyone in the process conversation, thus leading to increased visibility and more consistent results. Following are the key features that Blueworks Live provides around enterprise BPM.

Discovery and Documentation

- The first step in any process improvement effort is getting a clear picture of where you are today. With Blueworks Live, your whole team can engage collaboratively in the process discovery exercise while your process experts can use complex BPMN to document the most intricate aspects of your processes.

Centralized Collaboration

- Social networking meets BPM in Blueworks Live. Built-in communication features, such as instant messaging, live news feeds, and the ability to leave comments enable you to immediately react to process changes that affect your projects. By consolidating all your organization's process knowledge in one place, you can ensure that the right people are informed of the right changes at the right time.
  - ➢ **Private Communities:** Blueworks Live's main login page immediately tells you what changes have occurred to processes you are either directly or indirectly involved in. Learn which new process have been "shared" or published, which processes have been edited, and who has commented on which process. Choose to "follow" work and processes that involve you. Real-time status updates increase visibility and accountability while keeping everyone engaged with their work.
  - ➢ **Public Communities:** Use the built-in expert community to gain industry, process improvement, and BPM knowledge. A curated Twitter stream is part of the Community view to provide you with relevant blogs, white papers, and other content that can help you improve your business operations. The Library houses a variety of material such as tutorials, analyst reports, and videos.
  - ➢ **Access Control:** Blueworks Live offers both Editor and Contributor licenses. As an editor, you are able to create, modify, and share a Process Blueprint. Contributors can view and comment on processes but aren't able to modify or create them. Both editors and contributors are able to configure and run Process Apps with Blueworks Live. Blueworks Live allows the account administrator to set the level of access, editor or contributor, for each

individual process. Administration is handled through a simple intuitive interface. You can allocate licenses by simply typing the user's email address and inviting them to join. If at some point, licenses are not being used, you can re-allocate those to new users. To obtain new seats, you can simply order them online by providing your credit card information and selecting the number of Editor and Contributor licenses you need.

Automating Simple Processes

- Every day you waste countless hours working on the wrong thing at the wrong time because the wrong person is on the approval email thread or because a deadline was lost at the bottom of an email chain. With Blueworks Live, configuring a repeatable Process App that gives you back all those wasted hours is as close as 90 seconds and a few form fields away. By adding a layer of governance to these types of ad-hoc processes your organization will regain control not only over the business but also over its inboxes.
- IBM Blueworks Live has one example where an entire organization can collaborate and make decisions using a third-party business service. The Blueworks Live provides 30 day free trial, after which you decide whether to subscribe to this service.
- IBM Blueworks Live is just one example and I guess of great value for small and medium enterprises (SMEs) who will not be able to consume or afford such a powerful business process management tool on their own.
- I think the option to test drive, pay per use, and of course, subscription and not purchase (so that you can easily change the service as your business requirements changes) is a good value proposition for all types of enterprises (including the large ones). We all know many instances of the enterprises that are stuck with their enterprise applications and tools because of some wrong assumptions, short-sighted decision-making during application selection or implementation process.
- When one looks at IBM's recent announcement regarding SaaS offerings, one sees that IBM currently offers number of software products under SaaS portfolio that can be integrated to create very different and customized BPaaS scenarios. The following critical enterprise business processes are currently covered by IBM's SaaS offerings:
  - ➢ Social collaboration
  - ➢ Business process management
  - ➢ Web analytics
  - ➢ Enterprise marketing management
  - ➢ Business-to-business integration
  - ➢ Supply-chain management
  - ➢ Security governance, risk management and compliance
  - ➢ Business service management
- Also, many ERP vendors and service providers are offering or in the process to offer the ERP functionalities as a BPaaS.

3.5 INFORMATION AS A SERVICE

- Information as a Service is an emerging cloud business model in which a company share or sells relevant Information to another company or individuals to perform their business.
- In the world of cloud computing where storage and compute are accessed using well-defined APIs, we also have available to us information that can be as easily accessed, also using well-defined APIs.

Following are General Example of Information as a Service:

- Zip code or Address validation and lookup
- Payment processing
- Services that validate or complete data

3.6 INTEGRATION AS A SERVICE

- Integration as a Service is a cloud service delivery model for integration. Integration-as-a-Service delivers an integration solution that provides connectivity to backend systems, sources, files, and operational applications through the implementation of well-defined interfaces, web services, and calls between applications and data sources.
- Integration Platform as a Service (iPaaS) is an emerging cloud integration solution that is capable of handling tough integration scenarios. As a platform for building and developing integrations in the cloud, iPaaS can be used for simple point-to-point integrations but also includes developer tools for creating custom integrations, seamlessly scaling up as the number of endpoints increases.

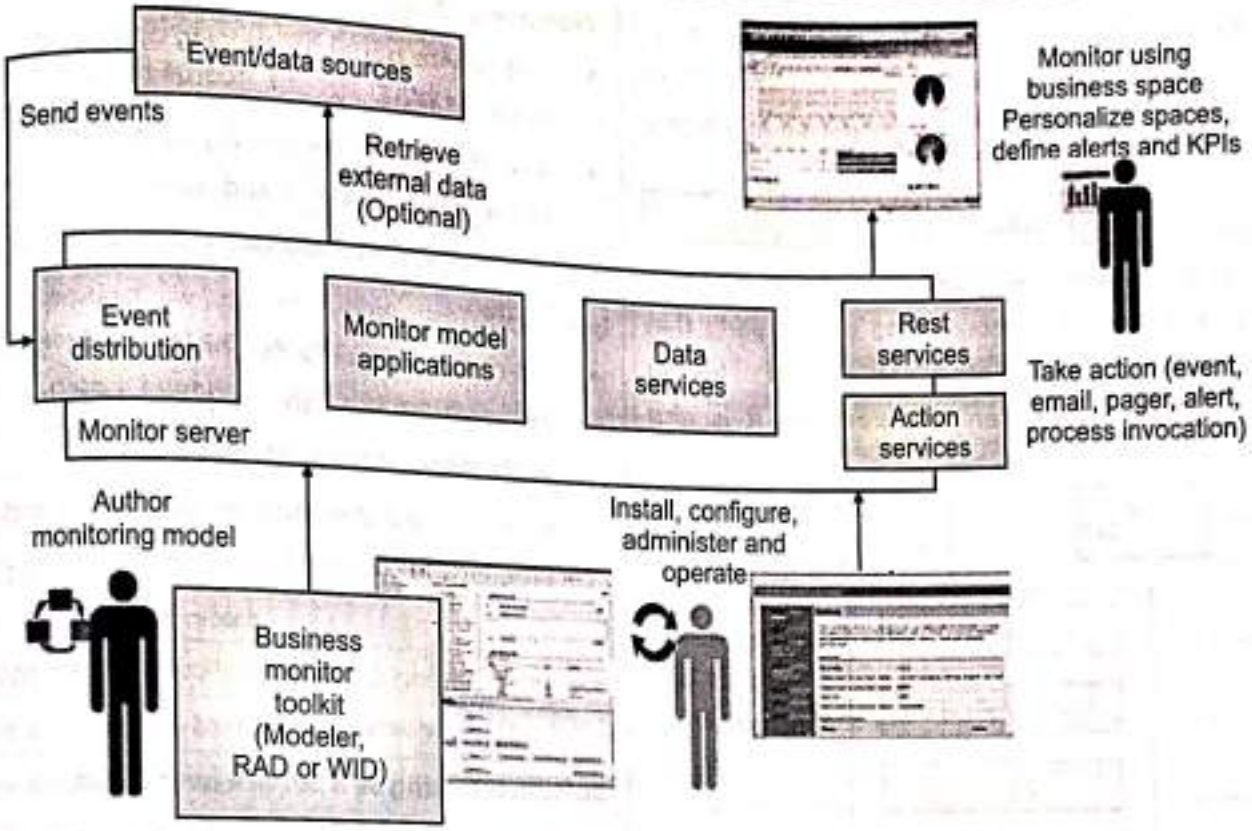


Fig. 3.7

Integration as a Service

- This provides users with a more loosely coupled environment, safe from complex interdependencies. The Integration-as-a-Service delivery model enables integration across the cloud, making it possible to share data between systems as well as third-party vendors in real-time.
- Businesses employ Integration-as-a-Service initiatives in order to gain agility and automate business processes. Moreover, with the change in the way the IT landscape is structured, businesses seek a flexible solution that is capable of delivering connectivity between databases, applications, files, and sources residing on-premises and in the cloud.

Point-to-Point Integration

- Businesses have taken other approaches to ensure connectivity across the enterprise, including point-to-point integration. This approach calls upon experienced developers to build custom integrations between endpoints in order to allow seamless communication. Point-to-point integration, although appearing to be simple at first, quickly becomes complicated. As the number of applications, services, and systems increases, custom code integration becomes an intertwined web of connectivity, quickly becoming "spaghetti architecture". This tightly coupled infrastructure creates dependencies that result in complications when changes need to be implemented. The fragile structure is vulnerable to breaking at even the slightest modifications. With inevitable change, businesses need to be flexible, scalable, and future-proof.

Example of Integration as a Service

- CloudHub is the world's leading cloud-based integration platform as a service that gives businesses the ability to integrate applications, systems, and services on-premises or in the cloud. It helps organizations stay connected through cloud to cloud and cloud to enterprise integration, as well as social and API enablement. The cloud-based solution also gives businesses flexibility and control to overcome the number one sales barrier that SaaS vendors face –

integration. CloudHub, working with the other components of MuleSoft's Anypoint Platform™, gives businesses all they need to stay connected in a highly fragmented ecosystem.

3.7 TESTING AS SERVICE

- Testing as a Service is an outsourcing model, in which testing activities are outsourced to a third party that specializes in simulating real world testing environments as per client requirements. It is also abbreviated as TaaS.

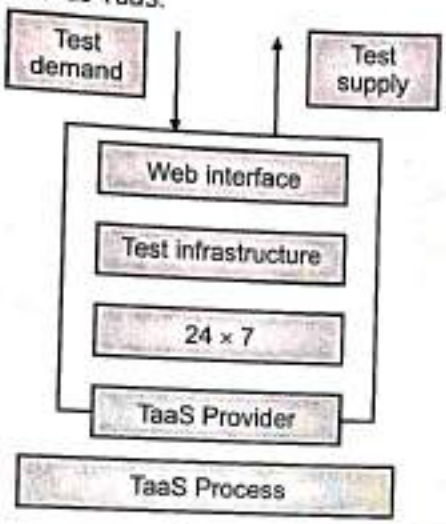


Fig. 3.8

Types of TaaS

1. Functional Testing.
2. Performance Testing.
3. Security Testing.

1. Functional Testing as a Service

- TaaS Functional Testing may include UI/GUI Testing, regression, integration and automated User Acceptance Testing (UAT) but not necessary to be part of functional testing.

2. Performance Testing as a Service

- Multiple users are accessing the application at the same time. TaaS mimic as a real world users environment by creating virtual users and performing the load and stress test.

3. Security Testing as a Service

- TaaS scans the applications and websites for any vulnerability

Features of TaaS

- Self-service portal for running application for functio and load tests.
- Test library with full security controls that saves all th test assets available to end users.
- To maximize the hardware utilization, sharing of Clou hardware
- On-demand availability for complete test labs tha includes ability to deploy complex multi-tie applications, test scripts, and test tools
- It ensures the detection of bottlenecks and solve th problems for the application under test by monitoring it
- The metering capabilities allows tracking and charging for that the services used by customer

Software Testing as a Service over Cloud

- Once user scenarios are created, and the test is designed, these service providers delivers servers to generate virtual traffic across the globe.
- In Cloud, software testing occurs in following steps
  - ➢ Develop users scenarios
  - ➢ Design test cases
  - ➢ Select cloud service provider
  - ➢ Set up infrastructure
  - ➢ Leverage cloud service
  - ➢ Start testing
  - ➢ Monitor goals
  - ➢ Deliver

TaaS is Useful When

- Testing of applications that require extensive automation and with short test execution cycle.
- Performing testing task that don't ask for in-depth knowledge of the design or the system.
- For ad-hoc or irregular testing activities that require extensive resources.

Benefits of Cloud Testing

- Flexible Test Execution and Test Assets.
- Some users claim 40-60% savings in the cloud testing vs. the traditional testing model.
- Achieve a fast return of investments by eliminating the investment made after hardware procurement, management, and maintenance, software licensing, etc.
- Deliver product in quicker time through rapid procurement, project set-up, and execution.
- Ensure data integrity and anytime anywhere accessibility.
- Reduce operational costs, maintenance costs and investments.
- Pay as you use

Traditional vs. TaaS Services

No.	Approach	Traditional	TaaS
1.	Test Environment	Manually Created	On demand
2.	Test Assets	Manually Generated	Dynamic
3.	Test Data	Manually Generated	Dynamic Sanitized
4.	Test Tools	Manually Purchased	On demand
5.	Test Documentation	Manually Generated	Dynamically Generated
6.	Business Domain Knowledge	Manually Extracted	Extracted Dynamically

3.8 SCALING A CLOUD INFRASTRUCTURE

- One of the biggest advantages of cloud computing is its ability to scale on demand. While considering the numerous benefits of cloud, it becomes difficult to conceptualize the extent and power of scaling on-demand. Organizations of different categories benefit tremendously when auto-scaling is implemented correctly. Issues faced earlier are now non-existent as engineers working on cloud implementation remember that a massive influx of traffic once caused servers to fail. No longer.
- AWS auto-scaling significantly reduces risks associated with traffic overflow leading to server failure. Auto-scaling can lead to cost reduction as well. Resources that match to actual usage are run on a 'moment-to-moment' basis. The advantages of scale and pricing come with their own set of complexities. While on-demand scaling in cloud computing is absolutely possible, applications too need to be able to scale up with the environment. This may look quite simple - running a website backed up by an elastic load balancer that distributes traffic in the case of increased demand.
- Yet, there are other considerations that need to be made while accounting for scaling session information, uploads, and data. When compared to the legacy IT management systems, the single most important change in cloud computing is that it needs to be totally replaceable. AWS provides various tools for facilitating this process. For example, instead of local data storage, S3, which is AWS' storage solution, can be used. There is also a distributed file system which can be made use of just in case your business systems and data cannot be moved onto S3. RDS or Elasti Cache should be used for saving your sessions in place of local files.
- Issues related to scalability are not new. But with the advent of AWS and cloud-based engineering techniques, solutions are available and old problems can be solved with ease. Previously, approaches like forking and threads were used to scale up applications. Generally, programmers focus on developing applications that are scalable, without much consideration about the data that is stored in the memory. But now, due to auto-scaling, systems get scaled to become the Memory and CPU and developers are able to write data for long term storage. Configuration management is popularly used but there are many other ways that can be utilized to scale up a system from 0 to 100%. Irrespective of the approach, scaling ability is dependent on the application's capability of scaling up.
- But why is scalability in cloud computing so important? Would you want your cloud to be scalable at all? And if so then, why? Here are a few industry related instances:
  - **Managing Seasonality in e-Commerce**: Your e-commerce business would have a peak season during festivities and holidays for which you need to be prepared by increasing processing power. Then you would need to scale back post the shopping season to avoid exceeding your budget

and infrastructure sitting idle. Here is where the scalability of cloud comes to play- you can scale up and down with ease.

- ➢ **Data Analysis**: If numbers and statistical analysis forms the basis of your business, you might require running minimal infrastructure when there is less processing of data sets, and also increase storage and processing capability when the quantum of data is more. Flexibility is a factor that is advantageous for most businesses.
- ➢ **Managing Social Media Applications:** A scalable storage and infrastructure is a must if development of social media applications is your forte. These applications are often driven by Big Data which is created through social interactions. From a scalability viewpoint, you would require to scale up to deal with growth. Here, again scalability plays a major role in adding more resources and processing capabilities.

Summing up the Major Benefits of Scalability in Cloud Computing

- Efficient usage of resources as there is no tie up with hardware that you are not using actively.
- Control panel management and Open API that allows the setting of parameters without having to take care of the servers.
- Since it is powered by Open Stack, any application can be moved to another cloud or to the onsite data center. This can be achieved without rebuilding the application code.

3.8.1 Capacity Planning

- For available resources, capacity planning seeks a heavy demand. It determines whether the systems are working properly, used to measure their performance, determine the usage of patterns and predict future demand of cloud-capacity. This also adds an expertise planning for improvement and optimizes performance. The goal of capacity planning is to maintain the workload without improving the efficiency. Tuning of performance and work optimization is not the major target of capacity planners.
- It measures the maximum amount of task that it can perform. The capacity planning for cloud technology offers the systems with more enhanced capabilities including some new challenges over a purely physical system.
- The goal of capacity planners is to identify significant & vital resources that have resource ceiling & add more resources to move the restricted access to high levels of demand. Network capacity is one of th hardest factors to resolve & the performance of th network is affected by I/O of the network at the serv & network traffic from cloud to ISPs (Internet Servic Providers).
- Capacity planners try to find the solution to mee future demands on a system by providing addition capacity to fulfill those demands. Capacity planning system optimization are two both different concept and you mustn't mix them as one. Performance capacity are two different attributes of a system. Clou 'capacity' measures & concerns about how muc workload a system can hold whereas 'performance deals with the rate at which a task get performed.
  1. Determine the distinctiveness of the presen system.
  2. Determine the working load for different resource in the system such as CPU, RAM, network, etc.
  3. Load the system until it gets overloaded; & state what's requiring to uphold acceptable performance.
  4. Predict the future based on older statistical reports & other factors.
  5. Deploy resources to meet the predictions & calculations.
  6. Repeat step 1. through 5. as a loop.

Cloud Application Baseline

- The first thing that strikes in mind while dealing with the business issue is the system's capacity or working load as a measurable quantity over time, since many developers build their cloud-based applications & websites based on the LAMP. The full-form is extracted below:
  1. Linux - operating system
  2. Apache - Apache Software Foundation's Web server
  3. MySQL - database server
  4. PHP - Hypertext Preprocessor
- The above four technologies are open-source although the distribution may vary from cloud to cloud. There are other slight variations of the LAMP that are available for development. These are:
  - ➢ OpAMP (OpenBSD Apache MySQL PHP)
  - ➢ SAMP (Solaris Apache MySQL PHP)
  - ➢ WAMP (Windows Apache MySQL PHP)

Baseline Measurement

- There are two important work-load matrices in the LAMP system. These are:
  1. **Page View:** is the number of hits on a website & is measured in hits per second
  2. **Transactions:** is measured by transactions per second and is the number of queries the database server completes per second

Load Testing

- Server administrator checks for servers under load for system metrics to give capacity planners enough information to do significant capacity planning. Capacity planners should know about the increase in load to the system. Load-testing needs to query the following questions:
  - What is the optimum load that the system can support?
  - What system blocks the current system & limits the system's performance?
  - Can the configuration be altered in the server to use capacity?
  - How will the server react concerning performance with other servers having different characteristics?

3.8.2 Cloud Scale

- The problem of scaling can be solved by adding resources to a specific instance (vertical scaling) or adding a few more instances (horizontal scaling).

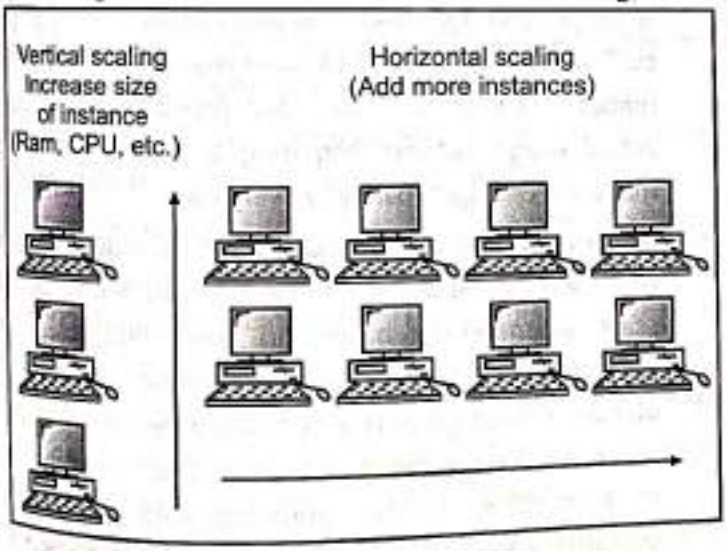
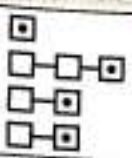
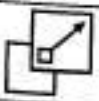


Fig. 3.9

- **Vertical Scaling in the Cloud :** Adding more CPU, memory, or I/O resources to an existing instance. Amazon Web Services (AWS) vertical scaling and Microsoft Azure vertical scaling can be performed when you change instance sizes. Scaling vertically in the cloud is possible for everything from EC2 instances to RDS databases. AWS and Azure cloud services offer many different instance sizes. The key advantage of vertical scaling is that it's fast. The key disadvantage of it is that it offers limited scalability. Therefore, the vertical scaling strategy is widely used by small and mid-sized companies.
- **Horizontal Scaling in the Cloud** : Adding more instances instead of moving to a larger instance size. That often means splitting workloads between instances to limit the number of requests any individual instance is getting, and it is good for performance, no matter how large the instance.
- Scaling horizontally is usually the best practice if your business deals with very high traffic spikes. Also, It's much easier to accomplish without downtime as scaling vertically (even in the cloud) usually calls for making the application unavailable for a certain period of time. And there even may be cases that that particular instance may be unavailable after the server is up, and you need to find another one to replace it. Therefore, horizontal scaling is the best choice when a high availability of services is required. Also, horizontal scaling is easier to manage automatically.

- Many global tech giants such as Google, Yahoo, Facebook , eBay, Amazon, etc. use horizontal scaling to meet high load demands.

Scaling Type	Advantage		Disadvantage	
Horizontal		Much lover cost than vertical scaling.		Software has to handle all the data distribution and parallel processing complexities.

Scaling Type	Advantage		Disadvantage	
		Easier to run fault-tolerance.		Limited number of software are available that can take advantage of horizontal scaling
		Ability to scale out as much as possible.		
		High availability		Higher utility cost (Electricity and cooling)
Vertical		Most of the software can easily take advantage of vertical scaling		Requires huge amount of financial investment
		Less power consumption than running multiple servers.		Greater risk of hardware failure causing bigger outages
		Easy to manage and install hardware within a single machine.		Generally vendor lock-in and limited upgradeability in the future.
				Low availability

What is the best choice for the majority of big companies? A hybrid approach.

- Large companies usually prefer horizontal scaling, but they also take advantage of vertical scaling by adding very powerful machines when scaling horizontally. This way, you can benefit from the speed of vertical scaling, combined with the infinite scalability and the resilience of horizontal scaling.

Scaling Cloud Infrastructure can be Manual, Scheduled or Automated.

- **Manual Scaling :** Both vertical and horizontal scaling can be accomplished manually by an individual. However, manual scaling is not very efficient as it can't take into account all the minute-by-minute changes in demand and traffic. This also can bring about human errors as an individual might forget to scale back down when the demand goes down. And that will lead to extra charges.
- **Scheduled Scaling :** Based on your typical demand curve during the day or certain periods of time, you can scale out to, for example, five instances from 5 pm to 10 pm and then back into two instances from 10 pm to 7 am, then back out to five instances at 5 pm. This makes it easier to tailor your provisioning to your actual usage without requiring a team member to make the changes manually every day.
- **Automatic Scaling :** (also known as autoscaling) is when your compute, database, and storage resources scale automatically based on predefined rules. For example, when metrics like CPU, memory, and network utilization rates go above or below a certain threshold, you can scale up, down, out, or in. The key advantage of autoscaling is that your application is always available and has enough resources provisioned to prevent performance problems or outages without paying for far more resources than you are actually using.
- Infrastructure as Code (IaC) is a key enabler for efficient migration of legacy systems to the cloud. Thanks to it, you can automatically manage and

provision computers and networks (physical and/or virtual) through scripts instead of manually configuring them.

Optimizing Costs when Scaling Cloud Infrastructure

- There are many best practices for cloud cost optimization. Here are the most common ones.
  - Deleting underutilized instances;
  - Rightsizing your workloads;
  - Taking advantage of autoscaling;
  - Moving infrequently accessed storage to cheaper tiers;
  - Setting alerts for crossing predetermined spend thresholds;
  - Exploring whether hosting in a different region could reduce costs;
  - Investing in reserved instances;
  - Leveraging spot instances for server less and things that don't require high availability;
  - Making use of discounts.

How to Choose Specialists for Effective Infrastructure Scaling.

- Specialists need to have a solid understanding of servers, networking, and protocols.
- They must be proficient in concurrency, performance, and resource constraints.
- They must be able to anticipate future issues and potential risks, offer and implement solutions.
- They must have long-standing experience in cloud migration and have the corresponding certifications.
- The company needs to have a large portfolio of cases that show solving different architecture and infrastructure challenges for various types of businesses.
- The company can undertake a Discovery Phase, conduct assessment of the present state and devise the road map that fits a specific business case.
- Infrastructure specialists need to have a high level of communication and problem-solving skills.

3.9 DISASTER RECOVERY

- Data is the most valuable asset of modern-day organizations. Its loss can result in irreversible damage to your business, including the loss of productivity, revenue, reputation, and even customers. It is hard to predict when a disaster will occur and how serious its impact will be. However, what you can control is the way you respond to a disaster and how successfully your organization will recover from it. Get to discover post how you can use disaster recovery in cloud computing for your benefit.
- As organizations more often use cloud storage, they're taking advantage of additional cloud services that can help them reduce costs and improve operational efficiencies. One cloud service type that's gaining popularity is cloud disaster recovery. The term "cloud disaster recovery" is often used interchangeably with disaster recovery as a service or DRaaS. This article refers to the former term.
- Cloud disaster recovery is a backup and restore strategy that applies not only to data, but also entire virtual machines, servers and corporate networks. The operative word is "strategy" because businesses need to decide for themselves how best to use such a service.
- Rather than simply subscribing to a service option and hoping for the best, it's important to understand what the company's priorities are in advance, so an appropriate disaster recovery plan can be put in place. Otherwise, the business may lack timely access to critical resources when disaster strikes. The results of such as mistake could be extremely costly or even fatal, so it's important to understand the trade offs among the available cloud disaster recovery options.
- Cloud disaster recovery is an attractive option for any size organization, regardless of where they fall along the cloud maturity spectrum. This is because cloud disaster recovery provides speed and cost advantages over traditional disaster recovery approaches. As enterprise infrastructures continue to become more virtual, an increasing amount of data and IT operations are moving into the cloud. Therefore, their disaster recovery strategies need to evolve.

Pros and Cons of Different Cloud DR Approaches

- Cloud disaster recovery is more flexible than traditional forms of disaster recovery because subscribers have more DR recovery solutions. Instead of backing data up from a data center to tape, for example, cloud subscribers have many more options including:

- Backing up from a data center.
- Backing up from a private cloud.
- Backing up from the same cloud service used to store data.
- Backing up from a hybrid cloud environment.
- Restoring to the original environment.
- Restoring to the cloud versus on-premise.

- Given the greater flexibility and the relatively low cost of cloud disaster recovery, it's an attractive option, although businesses are wise to consider their priorities first. In fact, part of any good cloud disaster recovery plan prioritizes the recovery of assets. If disaster strikes, what's critical and what isn't?
- For example, when it comes to data, there's the concept of hot, warm, and cool storage. "Hot" data is that which needs to be readily available. Of the three types of data, hot data is the data that is accessed most often. "Warm" data is data that's accessed less often than hot data, such as historical data used for reporting purposes. "Cold" data is data that is rarely used but must be retained. The stratified approach to data informs the associated Service Level Agreements (SLAs) and also the associated costs of storing the data.
- A similar temperature-type concept applies to the disaster recovery of entire sites:
  - A hot site is a complete copy of the production site. Its purpose is to minimize downtime in the event of a natural or man made disaster.
  - A warm site has established connections between a primary site and a secondary remote backup site. Recovery is delayed, but not as much as a cold site.
  - A cold site is one that is essentially unprepared for disaster recovery so when disaster strikes, it takes considerable time to get the site back online. Not surprisingly, a cold site is the cheapest option, although when disaster strikes it may prove to be an expensive option from a total cost perspective.
- Cloud disaster recovery uses a comparable model that ranges from cold to hot. Specifically, customers can choose backup (the slowest and cheapest option), a minimal version of an environment, a partial version of the environment or full (multi-site) disaster recovery which is a SAN-to-SAN enterprise back up replication method that runs in the cloud and on-site. The benefit of multi-site is that traffic is rerouted to the cloud during recovery.
- How does disaster recovery in cloud computing differ from traditional disaster recovery?

  Traditional disaster recovery involves building a remote Disaster Recovery (DR) site, which requires constant maintenance and support on your part. In this case, data protection and disaster recovery are performed manually, which can be a time-consuming and resource-intensive process. Disaster recovery in cloud computing entails storing critical data and applications in cloud storage and failing over to a secondary site in case of a disaster. Cloud computing services are provided on a pay-as-you-go basis and can be accessed from anywhere and at any time. Backup and disaster recovery in cloud computing can be automated, requiring minimum input on your part.
- How does disaster recovery planning work in cloud computing?

  Creating, testing, and updating a DR plan can prepare your organization for an unexpected disaster and ensure safety and continuity for your business. A comprehensive DR plan should take into account your infrastructure, potential threats and vulnerabilities, most critical assets and the order of their recovery, and workable DR strategies. Integration of cloud computing services in disaster recovery allows you to design a DR plan and automate each step of the recovery process.

Backup and Disaster Recovery in Cloud Computing

- Cloud computing is the on-demand delivery of computing services over the internet (more often referred to as 'the cloud') which operates on a pay-as-you-go basis. Cloud computing vendors generally provide access to the following services:
- Infrastructure as a Service (IaaS) allows you to rent IT infrastructure, including servers, storages and network component, from the cloud vendor.
- Platform as a Service (PaaS) allows you to rent a computing platform from the cloud provider for developing, testing, and configuring software applications.
- Software as a Service (SaaS) allows you to access software applications which are hosted on the cloud.

- As you can see, each cloud computing service is designed to help you achieve different business needs. More so, cloud computing can considerably improve data the security and high availability of your virtualized workloads. Let's discuss how you can approach disaster recovery in the cloud computing environment.

Cloud Disaster Recovery vs. Traditional Disaster Recovery

- Cloud disaster recovery is a cloud computing service which allows for storing and recovering system data on a remote cloud-based platform. To better understand what disaster recovery in cloud computing entails, let's compare it to traditional disaster recovery.
- The essential element of traditional disaster recovery is a secondary data center, which can store all redundant copies of critical data, and to which you can fail over production workloads. A traditional on-premises DR site generally includes the following:
- A dedicated facility for housing the IT infrastructure, including maintenance employees and computing equipment.
- Sufficient server capacity to ensure a high level of operational performance and allow the data center to scale up or scale out depending on your business needs.
- Internet connectivity with sufficient bandwidth to enable remote access to the secondary data center.
- Network infrastructure, including firewalls, routers, and switches, to ensure a reliable connection between the primary and secondary data centers, as well as provide data availability.
- However, traditional disaster recovery can often be too complex to manage and monitor. Moreover, support and maintenance of a physical DR site can be extremely expensive and time-consuming. When working with an on-premises data center, you can expand your server capacity only by purchasing additional computing equipment, which can require a lot of money, time, and effort.

Disaster Recovery in Cloud Computing

- Disaster recovery in cloud computing can effectively deal with most issues of traditional disaster recovery. The benefits include the following:
- You don't need to build a secondary physical site, and buy additional hardware and software to support critical operations. With disaster recovery in cloud computing, you get access to cloud storage, which can be used as a secondary DR site.
- Depending on your current business demands, you can easily scale up or down by adding required cloud computing resources.
- With its affordable pay-as-you go pricing model, you are required to pay only for the cloud computing services you actually use.
- Disaster recovery in cloud computing can be performed in a matter of minutes from anywhere. The only thing you need is a device that is connected to the internet.
- You can store your backed up data across multiple geographical locations, thus eliminating a single point of failure. You can always have a backup copy, even if one of the cloud data centers fails.
- State-of-the-art network infrastructure ensures that any issues or errors can be quickly identified and taken care of by a cloud provider. Moreover, the cloud provider ensures 24/7 support and maintenance of your cloud storage, including hardware and software upgrades.

Why Choose Disaster Recovery in Cloud Computing

- The primary goal of disaster recovery is to minimize the overall impact of a disaster on business performance. Disaster recovery in cloud computing can do just that. In case of disaster, critical workloads can be failed over to a DR site in order to resume business operations. As soon as your production data center gets restored, you can fail back from the cloud and restore your infrastructure and its components to their original state. As a result, business downtime is reduced and service disruption is minimized.
- Due to its cost-efficiency, scalability, and reliability, disaster recovery in cloud computing has become the most lucrative option for small and Medium-Sized Businesses (SMBs). Generally, SMBs don't have a sufficient budget or resources to build and maintain their own DR site. Cloud providers offer you access to cloud storage, which can become a cost-effective and long-lasting solution to data protection as well as disaster recovery.

- As you can see, each cloud computing service is designed to help you achieve different business needs. More so, cloud computing can considerably improve data the security and high availability of your virtualized workloads. Let's discuss how you can approach disaster recovery in the cloud computing environment.

Cloud Disaster Recovery vs. Traditional Disaster Recovery

- Cloud disaster recovery is a cloud computing service which allows for storing and recovering system data on a remote cloud-based platform. To better understand what disaster recovery in cloud computing entails, let's compare it to traditional disaster recovery.
- The essential element of traditional disaster recovery is a secondary data center, which can store all redundant copies of critical data, and to which you can fail over production workloads. A traditional on-premises DR site generally includes the following:
- A dedicated facility for housing the IT infrastructure, including maintenance employees and computing equipment.
- Sufficient server capacity to ensure a high level of operational performance and allow the data center to scale up or scale out depending on your business needs.
- Internet connectivity with sufficient bandwidth to enable remote access to the secondary data center.
- Network infrastructure, including firewalls, routers, and switches, to ensure a reliable connection between the primary and secondary data centers, as well as provide data availability.
- However, traditional disaster recovery can often be too complex to manage and monitor. Moreover, support and maintenance of a physical DR site can be extremely expensive and time-consuming. When working with an on-premises data center, you can expand your server capacity only by purchasing additional computing equipment, which can require a lot of money, time, and effort.

Disaster Recovery in Cloud Computing

- Disaster recovery in cloud computing can effectively deal with most issues of traditional disaster recovery. The benefits include the following:
- You don't need to build a secondary physical site, and buy additional hardware and software to support critical operations. With disaster recovery in cloud computing, you get access to cloud storage, which can be used as a secondary DR site.
- Depending on your current business demands, you can easily scale up or down by adding required cloud computing resources.
- With its affordable pay-as-you go pricing model, you are required to pay only for the cloud computing services you actually use.
- Disaster recovery in cloud computing can be performed in a matter of minutes from anywhere. The only thing you need is a device that is connected to the internet.
- You can store your backed up data across multiple geographical locations, thus eliminating a single point of failure. You can always have a backup copy, even if one of the cloud data centers fails.
- State-of-the-art network infrastructure ensures that any issues or errors can be quickly identified and taken care of by a cloud provider. Moreover, the cloud provider ensures 24/7 support and maintenance of your cloud storage, including hardware and software upgrades.

Why Choose Disaster Recovery in Cloud Computing

- The primary goal of disaster recovery is to minimize the overall impact of a disaster on business performance. Disaster recovery in cloud computing can do just that. In case of disaster, critical workloads can be failed over to a DR site in order to resume business operations. As soon as your production data center gets restored, you can fail back from the cloud and restore your infrastructure and its components to their original state. As a result, business downtime is reduced and service disruption is minimized.
- Due to its cost-efficiency, scalability, and reliability, disaster recovery in cloud computing has become the most lucrative option for small and Medium-Sized Businesses (SMBs). Generally, SMBs don't have a sufficient budget or resources to build and maintain their own DR site. Cloud providers offer you access to cloud storage, which can become a cost-effective and long-lasting solution to data protection as well as disaster recovery.

How to Design a Cloud-Based Disaster Recovery Plan

- After considering the benefits of cloud computing in disaster recovery, it is time to design a comprehensive DR plan. In fact, you can read one of our blog posts which walks you through the entire process of a creating a DR plan. Below, we are going to discuss how to create a DR plan which works in the cloud environment.
- As a rule, an effective cloud-based DR plan should include the following steps:
  - ➢ Perform a risk assessment and business impact analysis.
  - ➢ Choose prevention, preparedness, response, and recovery measures.
  - ➢ Test and update your cloud-based DR plan.
  - ➢ Let's discuss how disaster recovery planning works in cloud computing.

Perform a Risk Assessment and Business Impact Analysis

- The first step in a disaster recovery planning in cloud computing is to assess your current IT infrastructure, as well as identify potential threats and risk factors that your organization is most exposed to.
- A risk assessment helps you discover vulnerabilities of your IT infrastructure and identify which business functions and components are most critical. At the same time, a business impact analysis allows you to estimate how unexpected service disruption might affect your business.
- Based on these estimations, you can also calculate the financial and non-financial costs associated with a DR event, particularly Recovery Time Objective (RTO) and Recovery Point Objective (RPO). The RTO is the maximum amount of time that IT infrastructure can be down before any serious damage is done to your business. The RPO is the maximum amount of data which can be lost as a result of service disruption. Understanding the RTO and RPO can help you decide which data and applications to protect, how many resources to invest in achieving DR objectives, and which DR strategies to implement in your cloud-based DR plan.

Implement Prevention, Preparedness, Response, and Recovery Measures

- The next step is to decide which Prevention, Preparedness, Response, and Recovery (PPRR) measures should be implemented in disaster recovery of the cloud computing environment. In a nutshell, PPRR measures can accomplish the following:
- Prevention allows you to reduce possible threats and eliminate system vulnerabilities in order to prevent a disaster from occurring in the first place.
- Preparedness entails creating the outline of a DR plan which states what to do during an actual DR event. Remember to document every step of the process to ensure that the DR plan is properly executed during a disaster.
- Response describes which DR strategies should be implemented when a disaster strikes in order to address an incident and mitigate its impact.
- Recovery determines what should be done to successfully recover your infrastructure in case of a disaster and how to minimize the damage.
- After you have determined which approach to disaster recovery to implement, you should choose a data protection solution capable of putting your DR plan into action and achieving DR objectives. Choose the solution which meets your business needs and complies with your infrastructure requirements. For this purpose, consider the following criteria:
  - ➢ Available Services
  - ➢ Hardware capacity
  - ➢ Bandwidth
  - ➢ Data security
  - ➢ Ease of use
  - ➢ Service scalability
  - ➢ Cost
  - ➢ Reputation
  - ➢ Test and update your cloud-based DR plan
- After you have created and documented the DR plan, you should run regular tests to see if your plan actually works. You can test whether business-critical data and applications can be recovered within the expected time frame.

Testing a cloud-based DR plan can help you identify any issues and inconsistencies in your current approach to disaster recovery in cloud computing. After the test run, you can decide what your DR plan lacks and how it should be updated in order to achieve the required results and eliminate existing issues.

3.9.1 Disaster Recovery Planning

- Cloud disaster recovery planning begins with the prioritization of many things, including applications, data, and services. Each asset should have an acceptable recovery target associated with it. Understanding the top tips for disaster recovery planning as a service is essential.
- There are two reasons why prioritization is necessary: cost and operational prudence. Treating all assets equally is neither cost-effective nor practical because companies would over pay or under pay for services. What is the business impact if a particular application, other IT asset, or certain type of data becomes unavailable? Some assets are more critical than others, so a cloud disaster recovery plan should reflect those priorities.
- Two important cloud disaster recovery metrics are Recovery Time Objective (RTO) which is the time it takes to restore a business process to its target level and Recovery Point Objective (RPO) which defines the acceptable level of data loss. The target metrics need to be defined as part of the plan.
- It's also important to understand the scope of threats to business continuity. That is, the kinds of natural or man made disasters might cause business disruption. For example, when Hurricane Harvey struck Houston, many businesses in the area lacked a disaster recovery plan.
- Other prudent elements of a plan include who needs to be notified or involved in the event of a disaster and what the budget is for cloud disaster recovery. IDC estimates that the total cost of unplanned application downtime per hour is about $100,000 per hour for a non-critical application for a Fortune 1000 company. A critical application failure can cost between $500,000 and $1 million per hour. Knowing the trade offs between costs and recovery time is essential for enterprise backup and recovery management.

Benefits of Cloud Disaster Recovery

- **Flexibility:** Organizations are not tied to any particular type of architecture, so regardless of where they are on their cloud journey, they can choose an option that meets their needs.
- **Cost:** A cloud disaster recovery service is cheaper than physically duplicating an environment. This fact has enabled smaller organizations to take advantage of disaster recovery options that they couldn't afford otherwise.
- **Faster Recovery Times:** Backing up from the cloud is faster than backing up from tape. Hosting both sites or storing data on the same cloud as the cloud disaster recovery service can have additional time advantages. The virtualized nature of the cloud also offers advantages over a physical twin. For example, if a virtualized server fails, another can be spun up in minutes. Since virtualized servers are technology-independent, different applications, operating systems, data, and patches can be store on them, so all of that is automatically restored with the virtual server. In a non-virtualized environment, each element has to be restored individually.
- **Elasticity:** This general benefit of cloud computing applies to cloud disaster recovery. As data grows and environments become more complex, scalability is not an issue.
- **Compliance:** Faster recovery times may help avoid fines for missing deadlines.

Risks of Cloud Disaster Recovery

- **Security:** The biggest concern is the multi-tenant nature of cloud environments on which backup environments or data are hosted.
- **Recovery:** Can take longer than desired without comprehensive planning. Asset prioritization, SLAs, and bandwidth of the connections between the original environment and the backup environment should be considered.
- **Control over Data:** Data is easy to get into a cloud environment, but it may not be so easy to get out. Make sure to read the terms of use and pay attention to the details.
- **Outage:** Cloud environments aren't perfect. It's wise to consider this risk as part of an overall disaster recovery strategy.

3.9.2 Disasters in the Cloud

- Although, Google Doc is a public cloud platform where users can upload, share and access information, it still can face troubleshooting issues. An industry example can help you explain entire Google September outage in a better way.

  Example: A managed cloud provider that used Google Docs extensively. As Google Docs is known for its flexible public cloud architecture, most of the company's daily tasks were performed with the help of using Google Docs. Activities like arranging important events, conferences, sharing files and documents amongst team members or clients at the time employees are out of office, Google Docs helped out big time.

- However, there came a day when Google Docs suffered approximately an hour outage due to which work in terms of daily tasks came to a standstill. The word processor faced a downtime at 10 pm in the UK. It also made US organizations to suffer a lot from it as they could not access or share files with others.
- This proved to be a major setback for Managed Cloud Provider in terms of monetary losses company suffered as well as their reputation in the market.
- Solutions to this problem are that organizations should be always prepared for a disaster recovery and keep their clients well informed about the possibilities of cloud downtime so that they can get away from the embarrassment at later stages.
- Yet another cloud outage erupted with a bang when Google Docs collapsed in the Google HQ. Although, the problem that affected the San Francisco and Budapest regions of the US were instantly handled and rectified, there is no specific reason why such an outage happened for the second time.
- As conveyed to Cloud Pro, Google work force said that "Please rest assured that system reliability is a top priority at Google and we are making continuous improvements to make our systems better."

Microsoft launched its Office 365 cloud productivity suite, but just few months after its launch media broke the news of its collapse that shattered hopes of Microsoft applications users. The company also experienced a global outage with DNS servers falling into trash.

- With all these examples, it is proved that outages or technical faults like these is a common affair with even a flexible environment like cloud computing. The only thing organizations need to take care is the fact keep working on improving and enhancing their in-house IT infrastructures.
- According to the reports, Dublin invites several European and US countries to enter its territories for availing cloud services. Nevertheless, with all the facts in mind, the country's prevailing bad weather conditions are also not hidden.
- It all happened in the month of August that both Microsoft and Amazon's cloud data centers had blown off by a thunder lighting strike. Both the cloud servers collapsed because of it, which led big and small organizations to suffer hugely. Servers didn't give any access and remained non-functional for two consecutive days. Companies not only suffered monetary losses, but also had hard time in recovering.
- Yet another cloud disaster occurred when Amazon EC2 or Elastic Cloud Compute hit the East coast of the US making big time players like the Reddit, Hootsuite, Quora and Sqaurefoot suffer tremendously. To add more to its numbers, approximately 170 SMBs also suffered a major setback as they found it extremely tough to run their businesses during an 8 hour downtime that Amazon EC2 cloud showed to them.
- As per the reports there were frequent and timely updates regarding the troubleshooting that erupted from Amazon, it put almost all the IT organizations running their businesses or daily task on that platform to a standstill. Thus, one thing or lesson organizations must take from this incidence is that always keep a back plan for important clients in order to keep and maintain a healthy relationship with them.
- After knowing so much on cloud and its cons, one is sure that it is the organizations that have to put in extra efforts for running their businesses smooth.

Better disaster recovery and data backup plans and working efficiently with ample presence of mind is the only trick to win the race of business competition and set an example for others at the same time.

- Managed Cloud Services help you plan a better disaster recovery, so that no business hours are lost. Spreading the data and backup in different geographical regions and multiple cloud providers are the most easy way to reduce downtime and ensure 100% Service level even at the time of disaster.

3.9.3 Disaster Management

- When companies fail at cloud disaster management, it's often because they fail at imagination. Either they assume disasters are too unpredictable to prepare for, or else they assume everything will go as planned no matter what befalls them.
- Most DR scenarios aren't all that dramatic you're more likely to deal with a service outage than a major earthquake but that doesn't mean you can get complacent, or assume you won't suffer a bigger catastrophe at some point. Organizations that take the time to figure out all the things that could go wrong are much more likely to survive a disaster than those that just purchase a cloud backup service and call it a day.

The State of Cloud Disaster Management

- While the data shows that more companies are taking disaster recovery seriously, it's likely not enough. According to Forrester Research and the Disaster Recovery Journal, 40% of organizations had a formal enterprise risk management program that reports to the board or the C-suite in 2015, and only 19% of organizations lacked any kind of risk management program. A full 77% of organizations had a plan for data and information tampering, and 53% had a customer privacy breach plan. The majority of those tested their information tampering and privacy breach plans at least once a year.
- Those numbers are an improvement but it's clear that not all companies are taking disaster recovery seriously. 60% of organizations don't have an enterprise-wide risk management program, and almost half of organizations aren't ready for a customer privacy breach. And organizations tend to be too confident in the preparations they've made especially around data loss. According to a 2015 survey, although 77% are "at least fairly confident" that they can recover from data loss, 68.8% actually test their DR solutions less than once a year.
- The problem with cloud disaster management is that it's not very important until suddenly, it is. Outages are costly, but most companies can weather a few hours offline without too much of a problem. However, a major incident can destroy an unprepared organization, sometimes literally overnight. That's what happened to Code Spaces, a company offering services for developers. They were targeted by a hacker who took over their control panel and demanded ransom. When they tried to regain control, the hacker started deleting data at random. By the time they got their system back, the hacker had deleted so much data that they were forced to close.
- And Code Spaces did everything right at least on paper. They replicated their services, they backed up their data. Unfortunately, their backup could be controlled from the same panel as their primary system. Because they hadn't fully understood the risks that design decision created, they lost everything.

Many Cloud DR Providers aren't Helping the Situations.

- Collecting rent is a good way to make money in IT, and there's nothing wrong with that. If you take good care of your customers' data and keep them happy, you'll be able to make good profit off your investment, save your customers money, and provide better reliability than they'd achieve in-house.
- But if you're providing cloud DR, you need to earn your keep by testing your disaster management procedures, while making sure your team and your customers know what to do when something goes wrong. Many cloud disaster recovery providers simply don't do that. They may provide reliable hosting, data replication and failover procedures that should work, but they don't make sure they do work. From the customer perspective, it seems fine they trust the

company to host the software, and it looks like everything is ready to go. But without running actual simulations that test disaster preparedness, they never know what risks they're taking until it's too late.

- Cloud Disaster Recovery Solutions Are Complex

There's More to Weathering A Disaster Than Your It

- When people talk about DR, they're often referring to the core services and infrastructure that replicate data and (in theory) can be used to rollover to a backup service in an emergency. But successful recovery doesn't just require that a copy of your data exists somewhere — it means that you can actually get your business up and online, and restore access to your stakeholders within a certain maximum amount of lost time and data.
- That means you need a trained team, able to execute the rollover and connect all your stakeholders. It also takes a lot of planning. You need to examine your business requirements for not just the rollover, but the entire disaster process. Creating RTO and RPO targets isn't enough. You need to be looking at the Maximum Tolerable Period of Disruption (MTPD) the amount of time from when the disaster is declared until your business resumes normal operations.
- To do that, you need to list out all your stakeholders, and examine what services they need, and how those services interact. Will your DR solution give access to your contractors? Will it automatically hook into your financial services provider? Does your cloud disaster management team have a method to get a hold of everyone and make sure they can successfully login to the backup system? Otherwise, you could end up with a backup production landscape that no one can use for a week.

Cloud Disaster Management Needs to Go Beyond It.

- IT issues like data center failure and service outages are at the center of DR for a good reason: they're far and away the most common disasters businesses face. But while having a strong data center solution is key there are plenty of other scenarios that can inflict catastrophic damage.
- Earthquakes, fires, major power outages, super storms, and security breaches all create unpredictable risks for businesses that can't be addressed by a backup system alone. Your disaster management approach does need to at least run through these scenarios, evaluate the risk to your particular business, and decide whether or not they're worth addressing, and how to do it. Realistically, you can't address everything that could possibly go wrong that's life but you can (and should) create a good, risk-based model that mitigates the most serious threats.

You Need a Complete Disaster Management Program

- Sometimes the universe just has it out for you. Your data center goes down while key members of your team are sick with a bad flu. You suffer a power outage in your West Coast office while your East Coast team is struggling through a major storm. Your landscape is attacked while you're updating your software, or is attacked repeatedly by the same hacker group, using different vectors.
- We're big believers that the right team and the right technology can lead you through a disaster. We use a cutting edge next generation cloud DR solution, and practice regularly to make sure we're prepared. But we also recognize that things don't always go as planned. If you're in cloud disaster management mode, it means something has already gone wrong, and there's a risk of other things breaking. Having backup plans to your backup plans, and an experienced team who can stay cool, calm, and collected is crucial, because you never know exactly what situation you're going to face.

When Disaster Strikes

- Let's imagine that a major storm or hurricane hits the region your head office and main data center are located in. If you have warning, you're lucky there's time for some extra preparation. Your data center team can double-check backup generators and other equipment, get extra supplies in place in case your team is stranded inside, and position emergency equipment, such as pumps to mitigate possible flooding in any vulnerable areas.

- They'll need to secure the area to minimize the risks of equipment being knocked loose, and make sure the right staff are on hand or on call. If it's going to be bad, your team may need to move mission-critical applications to a different facility, or even evacuate the data center in the most extreme cases.
- And, of course, they'll need to ensure they have a communication plan allowing them to coordinate their cloud DR solution with the backup center, even if cell service and connectivity go down. They'll have to balance the costs and risks of rolling over before the storm hits with those of trying to wait it out.
- In-House Cloud DR Solutions Rarely Work Things aren't going to be any easier for your home office. You need to make sure your staff stay safe, and that your business is prepared to minimize downtime during the disaster. You may need to close certain offices and temporary change the chain of command to keep your business running.
- Now imagine it's a freak storm. It starts off as just an ordinary thunderstorm, and grows rapidly into torrential rain, with massive winds, continuous lightning and flooding throughout the region. There's no time to batten down the data center or shuffle leadership you have nothing but your cloud disaster management plan and the preparations you made ahead of time.
- The first challenge is simply activating the disaster recovery plan. You need to convene leadership, declare a disaster, and activate the various stakeholders. This can be extremely challenging in a major storm. Cell service may be sporadic or down altogether. You may have members who are stranded or unaccounted for. You might not be able access to facilities or resources you counted on, such as computer systems and/or log books with DR contact information. Hope you've got backups extra copies around.
- Activating and coordinating the stakeholders gets more challenging as you go. Your communications leadership team needs to coordinate with all the parts of your business, which can include:
  - ➢ Management
  - ➢ Technical/IT
  - ➢ Customer relations
  - ➢ HR
  - ➢ Marketing
- Each of these leaders needs to be able to restore normal business operations for their own teams, from the board, all the way down to the ground floor worker and customers. And in many cases, they'll need to coordinate across teams as well. The board may need to coordinate with network managers to handle communications with key customers. HR, technical leadership and facilities teams may have to work together to ensure workers can get back to their jobs as soon as possible, and work out solutions to keep core processes running in the meantime.
- To make it all work, you need to assume the worst. If a key member of the team isn't available, or a system your DR solution depends on fails, you need a backup plan, no matter what. Furthermore, your cloud disaster recovery plan needs to spell out every phase of the process for everyone involved, from beginning to end. You won't have time for misunderstandings or mistakes. Your whole DR team needs to work together to get the business running again.

EXERCISE

1. What is XaaS? Give the example.
2. Describe in brief following approaches of XaaS.
   a. Storage as a Service
   b. Database as a Service
   c. Process as a Service
   d. Information as a Service
   e. Integration as a Service
   f. Testing as a Service
3. Mention the benefits of storage as a service
4. What are the benefits of cloud testing?
5. Write short note on "Scaling a cloud infrastructure"
6. Explain the concept of "capacity planning" in the context of cloud scaling.

UNIT IV

ANEKA: CLOUD APPLICATION PLATFORM

4.1 FRAMEWORK OVERVIEW

Aneka is a platform and a framework for developing distributed applications on the Cloud. It harnesses the spare CPU cycles of a heterogeneous network of desktop PCs and servers or datacenters on demand. Aneka provides developers with a rich set of APIs for transparently exploiting such resources and expressing the business logic of applications by using the preferred programming abstractions. System administrators can leverage on a collection of tools to monitor and control the deployed infrastructure. This can be a public cloud available to anyone through the Internet, or a private cloud constituted by a set of nodes with restricted access.

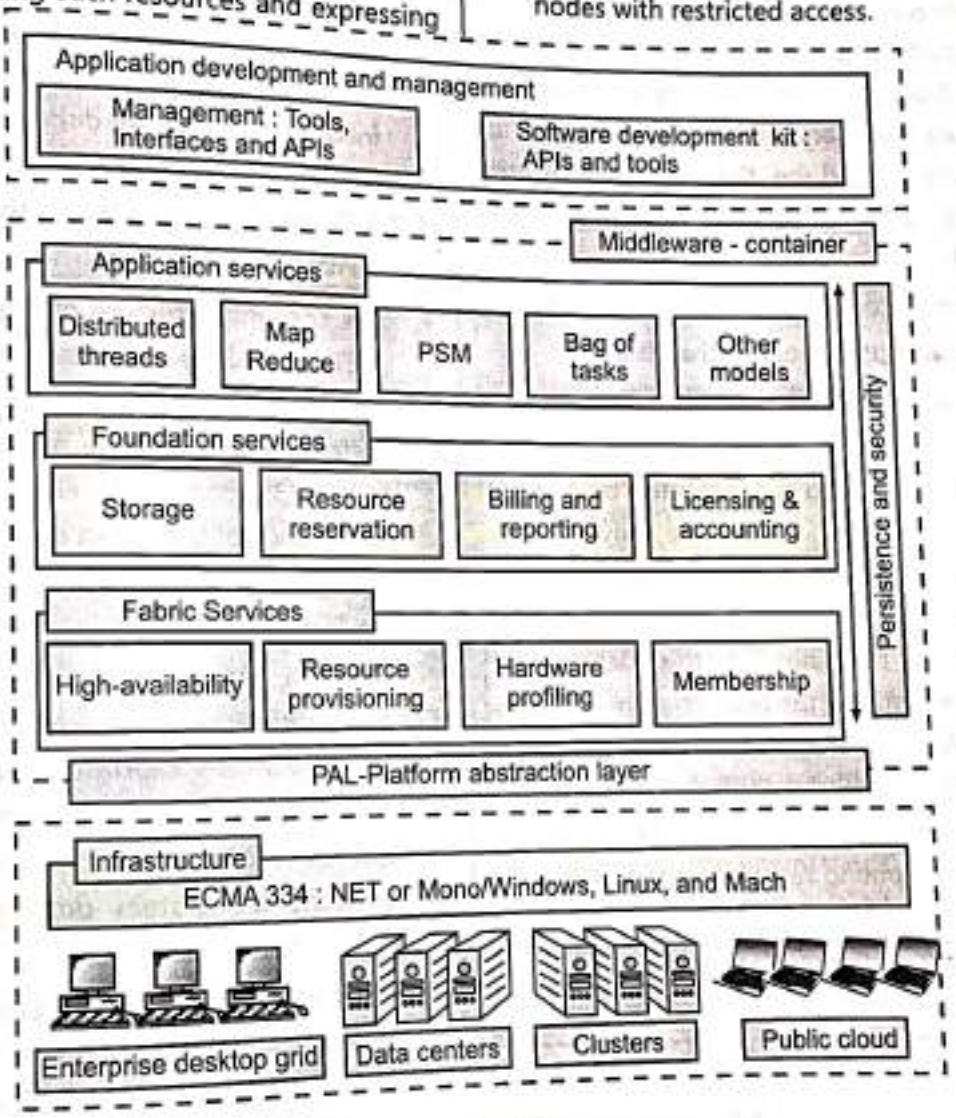


Fig. 4.1

The Aneka based computing cloud is a collection of physical and virtualized resources connected through a network, which are either the Internet or a private intranet. Each of these resources hosts an instance of the Aneka Container representing the runtime environment where the distributed applications are executed. The container provides the basic management features of the single node and leverages all the other operations on the services that it is hosting. The services are broken up into fabric,

foundation, and execution services. Fabric services directly interact with the node through the Platform Abstraction Layer (PAL) and perform hardware profiling and dynamic resource provisioning. Foundation services identify the core system of the Aneka middleware, providing a set of basic features to enable Aneka containers to perform specialized and specific sets of tasks. Execution services directly deal with the scheduling and execution of applications in the Cloud.

- One of the key features of Aneka is the ability of providing different ways for expressing distributed applications by offering different programming models; execution services are mostly concerned with providing the middleware with an implementation for these models. Additional services such as persistence and security are transversal to the entire stack of services that are hosted by the Container. At the application level, a set of different components and tools are provided to:
  1. Simplify the development of applications (SDK).
  2. Porting existing applications to the Cloud.
  3. Monitoring and managing the Aneka Cloud.
- A common deployment of Aneka is presented at the side. An Aneka based Cloud is constituted by a set of interconnected resources that are dynamically modified according to the user needs by using resource virtualization or by harnessing the spare CPU cycles of desktop machines. If the deployment identifies a private Cloud all the resources are in house, for example within the enterprise. This deployment is extended by adding publicly available resources on demand or by interacting with other Aneka public clouds providing computing resources connected over the Internet.

4.2 THE ANATOMY OF THE ANEKA CONTAINER

- 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 main role of the container is to provide a lightweight environment in which to deploy services and some basic capabilities such as communication channels through which it interacts with other nodes in the Aneka Cloud. Almost all operations performed within Aneka are carried out by the services managed by the container. The services installed in the Aneka container can be classified into three major categories:
  1. Fabric Services
  2. Foundation Services
  3. 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. Persistence and security traverse all the services stack to provide a secure and reliable infrastructure. In the following sections we discuss the components of these layers in more detail.

4.2.1 From the Ground Up: the Platform Abstraction Layer

- 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. Any platform featuring an ECMA-334 and ECMA-335 compatible environment can host and run an instance of the Aneka container.
- The Common Language Infrastructure (CLI), which is the specification introduced in the ECMA-334 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 is responsible for detecting the supported hosting environment and providing the corresponding implementation to interact with it to support the activity of the container. 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

Moreover, additional custom information can be retrieved by querying the properties of the hardware. The PAL interface provides means for custom implementations to pull additional information by using name-value pairs that can host any kind of information about the hosting platform. For example, these properties can contain additional information about the processor, such as the model and family, or additional data about the process running the container.

4.2.2 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.

1. Profiling and Monitoring

- Profiling and monitoring services are mostly exposed through the Heartbeat, Monitoring, and Reporting Services. The first makes available the information that is collected through the PAL; the other two implement a generic infrastructure for monitoring the activity of any service in the Aneka Cloud.
- 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.
- As already discussed, basic information about memory, disk space, CPU, and operating system is collected. Moreover, additional data are pulled into the "alive" message, such as information about the installed software in the system and any other useful information. More precisely, the infrastructure has been designed to carry over any type of data that can be expressed by means of text-valued properties. As previously noted, the information published by the Heartbeat Service is mostly concerned with the properties of the node.
- A specific component, called Node Resolver, is in charge of collecting these data and making them available to the Heartbeat Service. Aneka provides different implementations for such component in order to cover a wide variety of hosting environments. A variety of operating systems are supported with different implementations of the PAL, and different node resolvers allow Aneka to capture other types of data that do not strictly depend on the hosting operating system.
- For example, the retrieval of the public IP of the node is different in the case of physical machines or virtual instances hosted in the infrastructure of an IaaS provider such as EC2 or GoGrid. In virtual deployment, a different node resolver is used so that all other components of the system can work transparently. 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. The Reporting Service manages the store for monitored data and makes them accessible to other services or external applications for analysis purposes. On each node, an instance of the Monitoring Service acts as a gateway to the Reporting Service and forwards to it all the monitored data that has been collected on the node. Any service that wants to publish monitoring data can leverage the local monitoring service without knowing the details of the entire infrastructure. 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.
- All this information can be stored on a Relational Database Management System (RDBMS) or a flat file and can be further analyzed by specific applications. For example, the Management Console provides a view on such data for administrative purposes.

2. 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 Membership Catalogue is Aneka's fundamental component for resource management; it keeps track of the basic node information for all the nodes that are connected or disconnected. The Membership Catalogue implements the basic services of a directory service, allowing the search for services using attributes such as names and nodes. During container startup, each instance publishes its information to the Membership Catalogue and updates it constantly during its lifetime. Services and external applications can query the membership catalogue to discover the available services and interact with them. To speed up and enhance the performance of queries, the membership catalogue is organized as a distributed database: All the queries that pertain to information maintained locally are resolved locally; otherwise the query is forwarded to the main index node, which has a global knowledge of the entire Cloud.
- The Membership Catalogue is also the collector of the dynamic performance data of each node, which are then sent to the local monitoring service to be persisted in the long term. Indexing and categorizing resources are fundamental to resource management. On top of the basic indexing service, provisioning completes the set of features that are available for resource management within Aneka. Deployment of container instances and their configuration are performed by the infrastructure management layer and are not part of the Fabric Services.
- Dynamic resource provisioning allows the integration and management of virtual resources leased from IaaS providers into the Aneka Cloud. This service changes the structure of the Aneka Cloud by allowing it to scale up and down according to different needs: handling node failures, ensuring the quality of service for applications, or maintaining a constant performance and throughput of the Cloud.
- Aneka defines a very flexible infrastructure for resource provisioning whereby it is possible to change the logic that triggers provisioning, support several back-ends and change the runtime strategy with which a specific back-end is selected for provisioning.
- 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.
- A resource pool does not necessarily map to an IaaS provider but can be used to expose as dynamic

resources a private cloud managed by a Xen Hypervisor or a collection of physical resources that are only used sporadically.

- The system uses an open protocol, allowing for the use of metadata to provide additional information for describing resource pools and to customize provisioning requests. This infrastructure simplifies the implementation of additional features and the support of different implementations that can be transparently integrated into the existing system.
- Resource provisioning is a feature designed to support QoS requirements-driven execution of applications. Therefore, it mostly serves requests coming from the Reservation Service or the Scheduling Services. Despite this, external applications can directly leverage Aneka's resource provisioning capabilities by dynamically retrieving a client to the service and interacting with the infrastructure to it.
- This extends the resource-provisioning scenarios that can be handled by Aneka, which can also be used as a virtual machine manager.

4.2.3 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:
  1. Storage management for applications.
  2. Accounting, billing, and resource pricing.
  3. 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. These services are mostly consumed by the execution services and Management Consoles. External applications can leverage the exposed capabilities for providing advanced application management.

1. Storage Management

- Data management is an important aspect of any distributed system, even in computing clouds. Applications operate on data, which are mostly persisted and moved in the format of files. Hence, any infrastructure that supports the execution of distributed applications needs to provide facilities for file/data transfer management and persistent storage.
- Aneka offers two different facilities for storage management: a centralized file storage, which is mostly used for the execution of compute intensive applications, and a distributed file system, which is more suitable for the execution of data-intensive applications.
- The requirements for the two types of applications are rather different. Compute-intensive applications mostly require powerful processors and do not have high demands in terms of storage, which in many cases is used to store small files that are easily transferred from one node to another.
- In this scenario, a centralized storage node, or a pool of storage nodes, can constitute an appropriate solution. In contrast, data-intensive applications are characterized by large data files (gigabytes or terabytes), and the processing power required by tasks does not constitute a performance bottleneck. In this scenario, a distributed file system harnessing the storage space of all the nodes belonging to the cloud might be a better and more scalable solution.
- Centralized storage is implemented through and managed by Aneka's Storage Service. The service constitutes Aneka's data-staging facilities. It provides distributed applications with the basic file transfer facility and abstracts the use of a specific protocol to end users and other components of the system, which are dynamically configured at runtime according to the facilities installed in the cloud.
- The option that is currently installed by default is normal File Transfer Protocol (FTP). To support different protocols, the system introduces the concept of a file channel that identifies a pair of components: a file channel controller and a file channel handler. The file channel controller constitutes the server

component of the channel, where files are stored and made available; the file channel handler represents the client component, which is used by user applications or other components of the system to upload, download, or browse files.

- The storage service uses the configured file channel factory to first create the server component that will manage the storage and then create the client component on demand. User applications that require support for file transfer are automatically configured with the appropriate file channel handler and transparently upload input files or download output files during application execution.
- In the same way, worker nodes are configured by the infrastructure to retrieve the required files for the execution of the jobs and to upload their results. An interesting property of the file channel abstraction is the ability to chain two different channels to move files by using two different protocols.
- Each file in Aneka contains metadata that helps the infrastructure select the appropriate channel for moving the file.
- For example, an output file whose final location is an S3 bucket can be moved from the worker node to the Storage Service using the internal FTP protocol and then can be staged out on S3 by the FTP channel controller managed by the service. The Storage Service supports the execution of task-based programming such as the Task and the Thread Model as well as Parameter Sweep-based applications. Storage support for data-intensive applications is provided by means of a distributed file system.
- The reference model for the distributed file system is the Google File System, which features a highly scalable infrastructure based on commodity hardware.
- The architecture of the file system is based on a master node, which contains a global map of the file system and keeps track of the status of all the storage nodes, and a pool of chunk servers, which provide distributed storage space in which to store files. Files are logically organized into a directory structure but are persisted on the file system using a flat namespace based on a unique ID. Each file is organized as a collection of chunks that are all of the same size.
- File chunks are assigned unique IDs and stored on different servers, eventually replicated to provide high availability and failure tolerance.
- The model proposed by the Google File System provides optimized support for a specific class of applications that expose the following characteristics:
  - ➢ Files are huge by traditional standards (multi-gigabytes).
  - ➢ Files are modified by appending new data rather than rewriting existing data.
  - ➢ There are two kinds of major workloads: large streaming reads and small random reads.
  - ➢ It is more important to have a sustained bandwidth than a low latency.
- Moreover, given the huge number of commodity machines that the file system harnesses together, failure (process or hardware failure) is the norm rather than an exception. These characteristics strongly influenced the design of the storage, which provides the best performance for applications specifically designed to operate on data as described.
- Currently, the only programming model that makes use of the distributed file system is MapReduce, which has been the primary reason for the Google File System implementation. Aneka provides a simple Distributed File System (DFS), which relies on the file system services of the Windows operating system.

2. Accounting, Billing, and Resource Pricing

- Accounting Services keep track of the status of applications in the Aneka Cloud. The collected information provides a detailed breakdown of the distributed infrastructure usage and is vital for the proper management of resources. The information collected for accounting is primarily related to infrastructure usage and application execution.
- A complete history of application execution and storage as well as other resource utilization parameters is captured and minded by the Accounting Services. This information constitutes the foundation on which users are charged in Aneka.
- Billing is another important feature of accounting. Aneka is a multitenant cloud programming platform in which the execution of applications can involve provisioning additional resources from commercial

IaaS providers. Aneka Billing Service provides detailed information about each user's usage of resources, with the associated costs. Each resource can be priced differently according to the set of services that are available on the corresponding Aneka container or the installed software in the node.

- The accounting model provides an integrated view of budget spent for each application, a summary view of the costs associated to a specific user, and the detailed information about the execution cost of each job.
- The accounting capabilities are concentrated within the Accounting Service and the Reporting Service. The former keeps track of the information that is related to application execution, such as the distribution of jobs among the available resources, the timing of each of job, and the associated cost.
- The latter makes available the information collected from the monitoring services for accounting purposes: storage utilization and CPU performance. This information is primarily consumed by the Management Console.

3. Resource Reservation

- Aneka's Resource Reservation supports the execution of distributed applications and allows for reserving resources for exclusive use by specific applications. Resource reservation is built out of two different kinds of services: Resource Reservation and the Allocation Service.
- Resource Reservation keeps track of all the reserved time slots in the Aneka Cloud and provides a unified view of the system. The Allocation Service is installed on each node that features execution services and manages the database of information regarding the allocated slots on the local node.
- Applications that need to complete within a given deadline can make a reservation request for a specific number of nodes in a given timeframe. If it is possible to satisfy the request, the Reservation Service will return a reservation identifier as proof of the resource booking.
- During application execution, such an identifier is used to select the nodes that have been reserved, and they will be used to execute the application. On each reserved node, the execution services will check with the Allocation Service that each job has valid permissions to occupy the execution timeline by verifying the reservation identifier. Even though this is the general reference model for the reservation infrastructure, Aneka allows for different implementations of the service, which mostly vary in the protocol that is used to reserve resources or the parameters that can be specified while making a reservation request.
- Different protocol and strategies are integrated in a completely transparent manner, and Aneka provides extensible APIs for supporting advanced services. At the moment, the framework supports three different implementations:
  - (i) **Basic Reservation :** Features the basic capability to reserve execution slots on nodes and implements the alternate offers protocol, which provides alternative options in case the initial reservation requests cannot be satisfied.
  - (ii) **Libra Reservation :** Represents a variation of the previous implementation that features the ability to price nodes differently according to their hardware capabilities.
  - (iii) **Relay Reservation :** Constitutes a very thin implementation that allows a resource broker to reserve nodes in Aneka Clouds and control the logic with which these nodes are reserved. This implementation is useful in integration scenarios in which Aneka operates in an inter cloud environment. Resource reservation is fundamental to ensuring the quality of service that is negotiated for applications. It allows Aneka to have a predictable environment in which applications can complete within the deadline or not be executed at all. The assumptions made by the reservation service for accepting reservation requests are based on the static allocation of such requests to the existing physical (or virtual) infrastructure available at the time of the requests and by taking into account the current and future load. This solution is sensitive to node failures that could make Aneka unable to fulfill the Service-Level Agreement (SLA) made with users. Specific implementations of the service tend to delay the allocation of nodes to reservation requests as late as possible in order to cope with temporary

failures or limited outages, but in the case of serious outages in which the remaining available nodes are not able to cover the demand, this strategy is not enough. In this case, resource provisioning can provide an effective solution: Additional nodes can be provisioned from external resource providers in order to cover the outage and meet the SLA defined for applications. The current implementation of the resource reservation infrastructure leverages the provisioning capabilities of the fabric layer when the current availability in the system is not able to address the reservation requests already confirmed. Such behavior solves the problems of both insufficient resources and temporary failures.

4.2.4 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.

1. Scheduling

- 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
- Aneka does not provide a centralized scheduling engine, but each programming model features its own scheduling service that needs to work in synergy with the existing services of the middleware. As already mentioned, these services mostly belong to the fabric and the foundation layers of the architecture shown in Figure 4.1.
- The possibility of having different scheduling engines for different models gives great freedom in implementing scheduling and resource allocation strategies but, at the same time, requires a careful design of use of shared resources. In this scenario, common situations that have to be appropriately managed are the following: multiple jobs sent to the same node at the same time; jobs without reservations sent to reserved nodes; and jobs sent to nodes where the required services are not installed. Aneka's Foundation Services provide sufficient information to avoid these cases, but the runtime infrastructure does not feature specific policies to detect these conditions and provide corrective action.
- The current design philosophy in Aneka is to keep the scheduling engines completely separate from each other and to leverage existing services when needed. As a result, it is possible to enforce that only one job per programming model is run on each node at any given time, but the execution of applications is not mutually exclusive unless Resource Reservation is used.

2. Execution

- Execution Services control the execution of single jobs that compose applications. They are in charge of setting up the runtime environment hosting the execution of jobs. As happens for the scheduling services, each programming model has its own requirements, but it is possible to identify some common operations that apply across all the range of supported models:
  - Unpacking the jobs received from the scheduler
  - Retrieval of input files required for job execution
  - Sandboxed execution of jobs

- Execution failure management (i.e., capturing sufficient contextual information useful to identify the nature of the failure) at the end of execution
- Performance monitoring
- Packing jobs and sending them back to the scheduler

- Execution Services constitute a more self-contained unit with respect to the corresponding scheduling services. They handle less information and are required to integrate themselves only with the Storage Service and the local Allocation and Monitoring Services. Aneka provides a reference implementation of execution services that has built-in integration with all these services, and currently two of the supported programming models specialize on the reference implementation.
- Application Services constitute the runtime support of the programming model in the Aneka Cloud. Currently there are several supported models:
  - **Task Model :** This model provides the support for the independent "bag of tasks" applications and many computing tasks. In this model, an application is modeled as a collection of tasks that are independent from each other and whose execution can be sequenced in any order.
  - **Thread Model :** This model provides an extension to the classical multithreaded programming to a distributed infrastructure and uses the abstraction of Thread to wrap a method that is executed remotely.
  - **MapReduce Model :** This is an implementation of MapReduce as proposed by Google on top of Aneka.
  - **Parameter Sweep Model :** This model is a specialization of the Task Model for applications that can be described by a template task whose instances are created by generating different combinations of parameters, which identify a specific point into the domain of interest.
- Other programming models have been developed for internal use and are at an experimental stage. These are the Dataflow Model , the Message-Passing Interface, and the Actor Model .

4.3 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.

4.3.1 Infrastructure Organization

- Figure 4.2 provides an overview of Aneka Clouds from an infrastructure point of view.

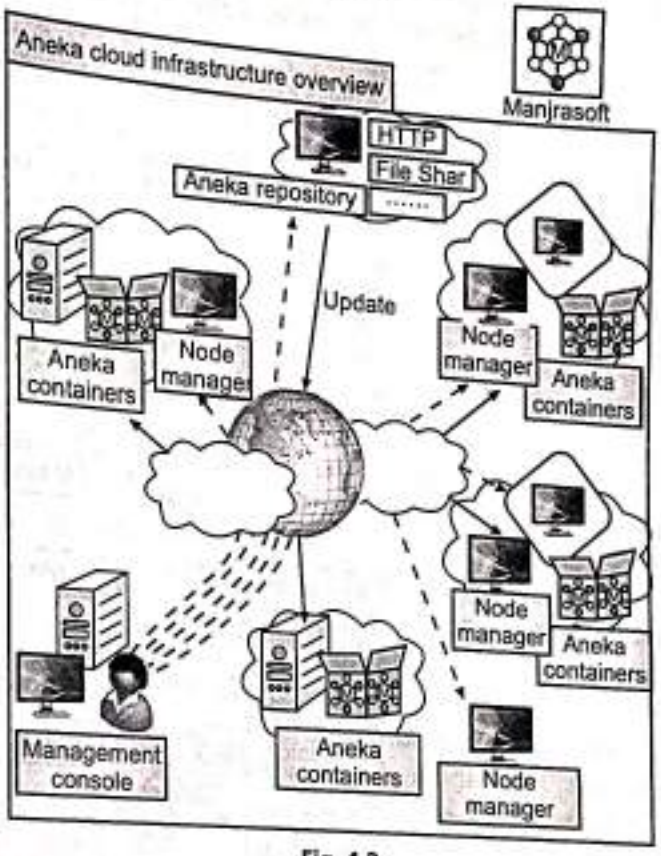


Fig. 4.2

- 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 layout and install the basic Aneka platform. These 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. The infrastructure is deployed by harnessing a collection of nodes and installing on them the Aneka node manager, also called the Aneka

daemon. The daemon constitutes the remote management service used to deploy and control container instances. The collection of resulting containers identifies the Aneka Cloud.

- From an infrastructure point of view, the management of physical or virtual nodes is performed uniformly as long as it is possible to have an Internet connection and remote administrative access to the node. A different scenario is constituted by the dynamic provisioning of virtual instances; these are generally created by prepackaged images already containing an installation of Aneka, which only need to be configured to join a specific Aneka Cloud. It is also possible to simply install the container or install the Aneka daemon, and the selection of the proper solution mostly depends on the lifetime of virtual resources.

4.3.2 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 4.3

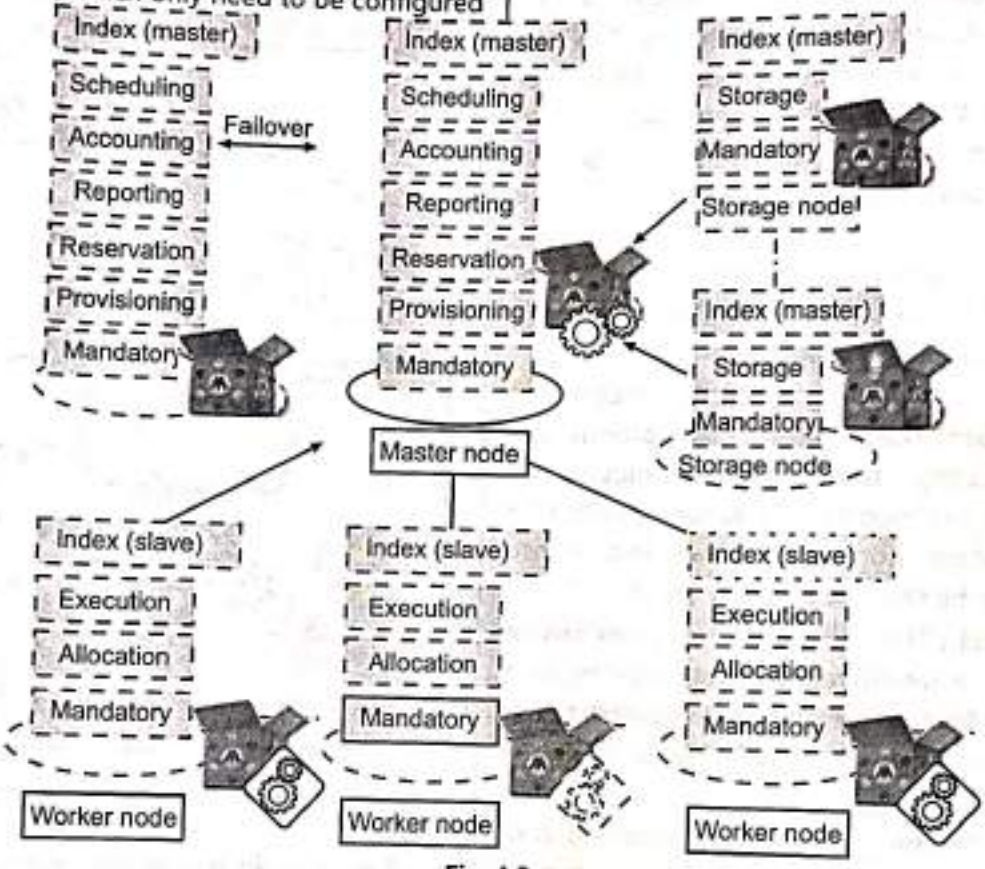


Fig. 4.3

- 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. What specifically characterizes a node as a master node is the presence of the Index Service (or Membership Catalogue) configured in master mode; all the other services, except for those that are mandatory, might be present or located in other nodes. 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. For the same reason, all the scheduling services are maintained in the master node. They share the application store that is normally persisted on the RDBMS in order to provide a fault-tolerant infrastructure. The master configuration can then be replicated in several nodes to provide a highly available infrastructure based on the failover mechanism.
- The worker nodes constitute the workforce of the Aneka Cloud and are generally configured for the execution of applications. They feature the mandatory services and the specific execution services of each of

the supported programming models in the Cloud. A very common configuration is the following:

- Index Service
- Heartbeat Service
- Logging Service
- Allocation Service
- Monitoring Service
- Execution Services for the supported programming models

- A different option is to partition the pool of worker nodes with a different selection of execution services in order to balance the load between programming models and reserve some nodes for a specific class of applications.
- Storage nodes are optimized to provide storage support to applications. They feature, among the mandatory and usual services, the presence of the Storage Service. The number of storage nodes strictly depends on the predicted workload and storage consumption of applications. Storage nodes mostly reside on machines that have considerable disk space to accommodate a large quantity of files. The common configuration of a storage node is the following:
  - Index Service
  - Heartbeat Service
  - Logging Service
  - Monitoring Service
  - Storage Service
- In specific cases, when the data transfer requirements are not demanding, there might be only one storage node. In some cases, for very small deployments, there is no need to have a separate storage node, and the Storage Service is installed and hosted on the master node.
- All nodes are registered with the master node and transparently refer to any failover partner in the case of a high-availability configuration.

4.3.3 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 has 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.

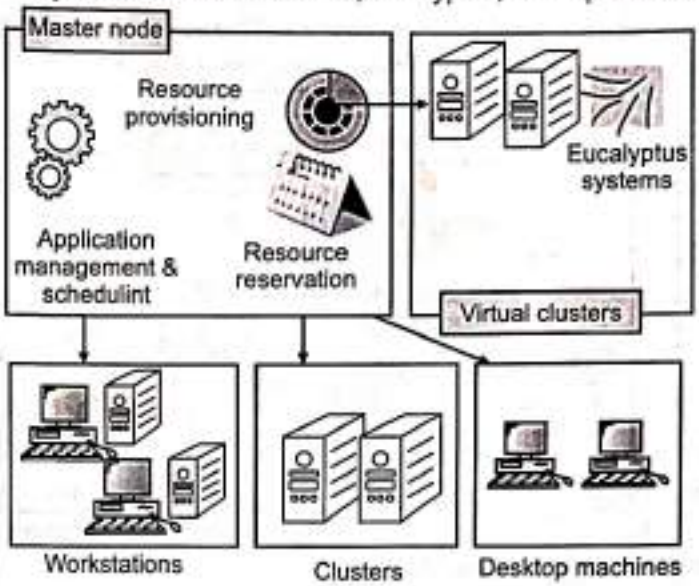


Fig. 4.4 : shows a common deployment for a private Aneka Cloud.

- 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.

4.3.4 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 though they were 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.

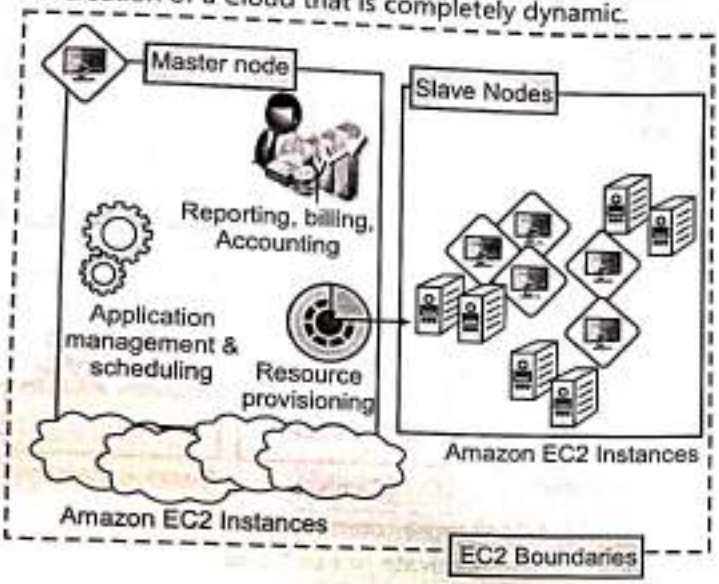


Fig. 4.5

- Figure 4.5 provides an overview of this scenario. The deployment is generally contained within the infrastructure boundaries of a single IaaS provider.
- The reasons for this are to minimize the data transfer between different providers, which is generally priced at a higher cost, and to have better network performance. 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.
- Dynamic instances provisioned on demand will mostly be configured as worker nodes, and, in the specific case of Amazon EC2, different images featuring a different hardware setup can be made available to instantiate worker containers.
- Applications with specific requirements for computing capacity or memory can provide additional information to the scheduler that will trigger the appropriate provisioning request. Application execution is not the only use of dynamic instances; any service requiring elastic scaling can leverage dynamic provisioning. Another example is the Storage Service.
- In multitenant Clouds, multiple applications can leverage the support for storage; in this scenario it is then possible to introduce bottlenecks or simply reach the quota limits allocated for storage on the node. Dynamic provisioning can easily solve this issue as it does for increasing the computing capability of an Aneka Cloud.
- Deployments using different providers are unlikely to happen because of the data transfer costs among providers, but they might be a possible scenario for federated Aneka Clouds.
- In this scenario resources can be shared or leased among providers under specific agreements and more convenient prices. In this case the specific policies installed in the Resource Provisioning Service can discriminate among different resource providers, mapping different IaaS providers to provide the best solution to a provisioning request.

4.3.5 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 4.6.

Aneka Based Hybrid Cloud Architecture

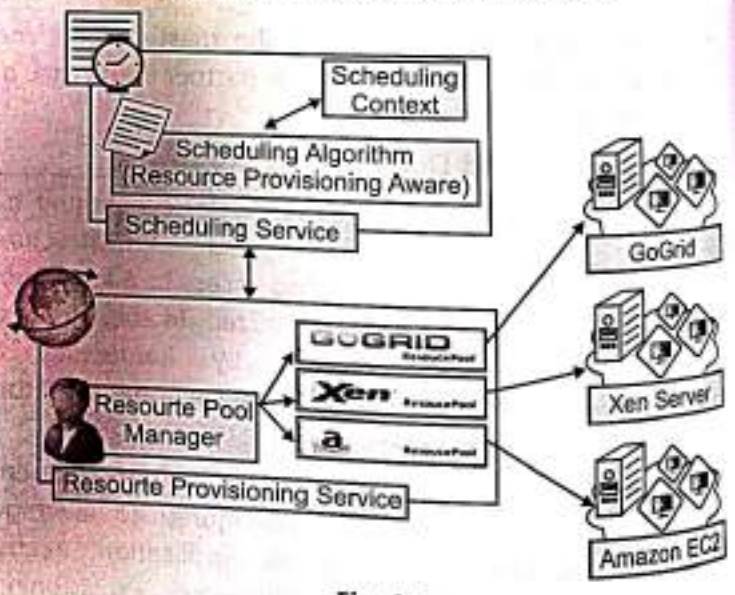


Fig. 4.6

- 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 discussed 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.
- 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.
- 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.

4.4 CLOUD PROGRAMMING AND MANAGEMENT

- Aneka's primary purpose is to provide a scalable middleware product in which to execute distributed applications. Application development and management constitute the two major features that are exposed to developers and system administrators.
- To simplify these activities, Aneka provides developers with a comprehensive and extensible set of APIs and administrators with powerful and intuitive management tools. The APIs for development are mostly concentrated in the Aneka SDK; management tools are exposed through the Management Console.

4.4.1 Aneka SDK

- Aneka provides APIs for developing applications on top of existing programming models, implementing new programming models, and developing new services to integrate into the Aneka Cloud. The development of applications mostly focuses on the use of existing features and leveraging the services of the middleware, while the implementation of new programming models or new services enriches the features of Aneka.
- The SDK provides support for both programming models and services by means of the Application Model and the Service Model. The former covers the development of applications and new programming models; the latter defines the general infrastructure for service development.

1. Application Model

- Aneka provides support for distributed execution in the Cloud with the abstraction of programming models. A programming model identifies both the abstraction used by the developers and the runtime support for the execution of programs on top of Aneka. The Application Model represents the minimum set of APIs that is common to all the programming models for representing and programming distributed applications on top of Aneka. This model is further specialized according to the needs and the particular features of each of the programming models.
- Each distributed application running on top of Aneka is an instance of the ApplicationBase < M > class, where M identifies the specific type of application manager used to control the application. Application classes constitute the developers' view of a distributed application on Aneka Clouds, whereas application managers are internal components that interact with Aneka Clouds in order to monitor and control the execution of the application. Application managers are

also the first element of specialization of the model and vary according to the specific programming model used.

- Whichever the specific model used, a distributed application can be conceived as a set of tasks for which the collective execution defines the execution of the application on the Cloud. Aneka further specializes applications into two main categories:
  1. Applications whose tasks are generated by the user
  2. Applications whose tasks are generated by the runtime infrastructure.
- These two categories generally correspond to different application base classes and different implementations of the application manager. The first category is the most common and it is used as a reference for several programming models supported by Aneka: the Task Model, the Thread Model, and the Parameter Sweep Model. Applications that fall into this category are composed of a collection of units of work submitted by the user and represented by the WorkUnit class. Each unit of work can have input and output files, the transfer of which is transparently managed by the runtime.
- The specific type of WorkUnit class used to represent the unit of work depends on the programming model used (Aneka Task for the Task Model and Aneka Thread for the Thread Model). All the applications that fall into this category inherit or are instances of AnekaApplication , W,M ., where W is the specific type of WorkUnit class used, and M is the type of application manager used to implement the IManualApplicationManager interface.
- The second category covers the case of MapReduce and all those other scenarios in which the units of work are generated by the runtime infrastructure rather than the user. In this case there is no common unit-of-work class used, and the specific classes used by application developers strictly depend on the requirements of the programming model used.
- For example, in the case of the MapReduce programming model, developers express their distributed applications in terms of two functions, map and reduce; hence, the MapReduceApplication class provides an interface for specifying the Mapper , K,V . and Reducer , K,V . types and the input files required by the application. Other programming models might have different requirements and expose different interfaces.
- For this reason there are no common base types for this category except for ApplicationBase , M ., where M implements IAutoApplicationManager. A set of additional classes completes the object model. Among these classes, the most notable are the Configuration class, which is used to specify the settings required to initialize the application and customize its behavior, and the ApplicationData class, which contains the runtime information of the application.
- Table 4.1 summarizes the features that are available in the Aneka Application Model and the way they reflect into the supported programming model. The model has been designed to be extensible, and these classes can be used as a starting point to implement a new programming model. This can be done by augmenting the features (or specializing) an existing implementation of a programming model or by using the base classes to define new models and abstractions.
- For example, the Parameter Sweep Model is a specialization of the Task Model, and it has been implemented in the context of management of applications on Aneka. It is achieved by providing a different interface to end users who just need to define a template task and the parameters that customize it.

Table 4.1 Aneka's Application Model Features

Category	Description	Base Application Type	Work Units?	Programming Models
Manual	Units of work are generated by the user and submitted through the application	AnekaApplication<W,M> IManualApplicationManager<W> ManualApplicationManager<W>	Yes	Task Model Thread Model Parameter Sweep Model
Auto	Units of work are generated by the runtime infrastructure and managed internally.	ApplicationBase,<M> IAutoApplicationManager	No	MapReduce Model

2. Service Model

- The Aneka Service Model defines the basic requirements to implement a service that can be hosted in an Aneka Cloud. The container defines the runtime environment in which services are hosted. Each service that is hosted in the container must be compliant with the IService interface, which exposes the following methods and properties:

- Name and status
- Control operations such as Start, Stop, Pause, and Continue methods
- Message handling by means of the HandleMessage method

Specific services can also provide clients if they are meant to directly interact with end users. Examples of such services might be Resource Provisioning and Resource Reservation Services, which ship their own clients for allowing resource provisioning and reservation. Apart from control operations, which are used by the container to set up and shut down the service during the container life cycle, the core logic of a service resides in its message-processing functionalities that are contained in the HandleMessage method. Each operation that is requested to a service is triggered by a specific message, and results are communicated back to the caller by means of messages.

Figure 4.7 describes the reference life cycle of each service instance in the Aneka container.

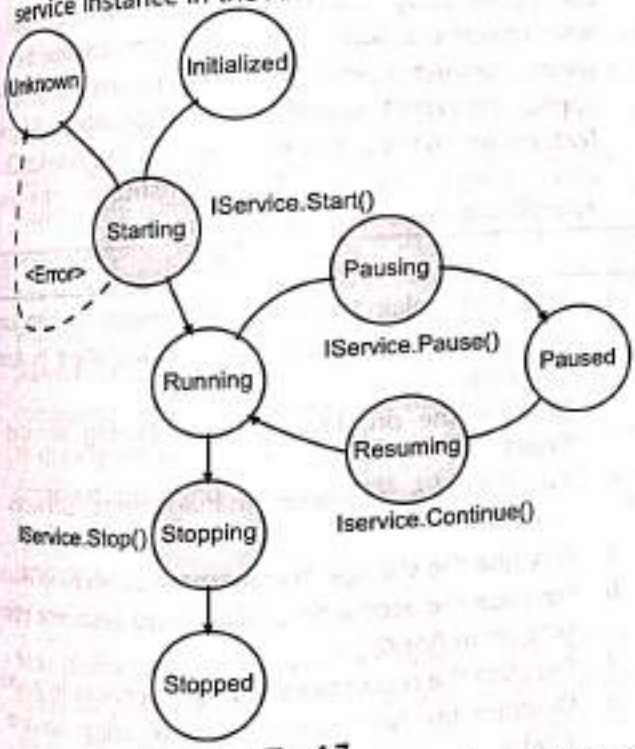


Fig. 4.7

The shaded balloons indicate transient states; the white balloons indicate steady states. A service instance can initially be in the Unknown or Initialized state, a condition that refers to the creation of the service instance by invoking its constructor during the configuration of the container. Once the container is started, it will iteratively call the Start method on each service method. As a result the service instance is expected to be in a Starting state until the startup process is completed, after which it will exhibit the Running state. This is the condition in which the service will last as long as the container is active and running.

- This is the only state in which the service is able to process messages. If an exception occurs while starting the service, it is expected that the service will fall back to the Unknown state, thus signaling an error. When a service is running it is possible to pause its activity by calling the Pause method and resume it by calling Continue. As described in the figure, the service moves first into the Pausing state, thus reaching the Paused state.
- From this state, it moves into the Resuming state while restoring its activity to return to the Running state. Not all the services need to support the pause/continue operations, and the current implementation of the framework does not feature any service with these capabilities. When the container shuts down, the Stop method is iteratively called on each service running, and services move first into the transient Stopping state to reach the final Stopped state, where all resources that were initially allocated have been released.
- Aneka provides a default base class for simplifying service implementation and a set of guidelines that service developers should follow to design and implement services that are compliant with Aneka. In particular, the guidelines define a ServiceBase class that can be further extended toprovide a proper implementation. This class is the base class of several services in the framework and provides some built-in features:
  - Implementation of the basic properties exposed by IService
  - Implementation of the control operations with logging capabilities and state control
  - Built-in infrastructure for delivering a service specific client
  - Support for service monitoring
- Developers are provided with template methods for specializing the behavior of control operations, implementing their own message-processing logic, and providing a service-specific client.
- Aneka uses a strongly typed message-passing communication model, whereby each service defines its own messages, which are in turn the only ones that the service is able to process. As a result, developers who implement new services in Aneka need also to define the type of messages that the services will use to communicate with services and clients. Each message type inherits from the base class Message defining common properties such as:
  - Source node and target node
  - Source service and target service

Security Credentials

- Additional properties are added to carry the specific information for each type. Messages are generally used inside the Aneka infrastructure. In case the service exposes features directly used by applications, they may expose a service client that provides an object-oriented interface to the operations exposed by the service.
- Aneka features a ready-to-use infrastructure for dynamically injecting service clients into applications by querying the middleware.
- Services inheriting from the ServiceBase class already support such a feature and only need to define an interface and a specific implementation for the service client. Service clients are useful to integrate Aneka services into existing applications that do not necessarily need support for the execution of distributed applications or require access to additional services.
- Aneka also provides advanced capabilities for service configuration. Developers can define editors and configuration classes that allow Aneka's management tools to integrate the configuration of services within the common workflow required by the container configuration.

4.4.2 Management Tools

- Aneka is a pure PaaS implementation and requires virtual or physical hardware to be deployed. Hence, infrastructure management, together with facilities for installing logical clouds on such infrastructure, is a fundamental feature of Aneka's management layer. This layer also includes capabilities for managing services and applications running in the Aneka Cloud.

1. Infrastructure Management

- Aneka leverages virtual and physical hardware in order to deploy Aneka Clouds. Virtual hardware is generally managed by means of the Resource Provisioning Service, which acquires resources on demand according to the need of applications, while physical hardware is directly managed by the Administrative Console by leveraging the Aneka management API of the PAL. The management features are mostly concerned with the provisioning of physical hardware and the remote installation of Aneka on the hardware.

2. Platform Management

- Infrastructure management provides the basic layer on top of which Aneka Clouds are deployed. The creation of Clouds is orchestrated by deploying a collection of services on the physical infrastructure that allows the installation and the management of containers.
- A collection of connected containers defines the platform on top of which applications are executed. The features available for platform management are mostly concerned with the logical organization and structure of Aneka Clouds. It is possible to partition the available hardware into several Clouds variably configured for different purposes. Services implement the core features of Aneka Clouds and the management layer exposes operations for some of them, such as Cloud monitoring, resource provisioning and reservation, user management, and application profiling.

3. Application Management

- Applications identify the user contribution to the Cloud. The management APIs provide administrators with monitoring and profiling features that help them track the usage of resources and relate them to users and applications. This is an important feature in a cloud computing scenario in which users are billed for their resource usage. Aneka exposes capabilities for giving summary and detailed information about application execution and resource utilization. All these features are made accessible through the Aneka Cloud Management Studio, which constitutes the main Administrative Console for the Cloud.

EXERCISE

1. Draw and explain the Aneka framework architecture
2. Give the classification of services installed in Aneka Container
3. Describe the profiling and monitoring services of Aneka
4. Describe the resource management services in Aneka.
5. Describe the storage management services in Aneka.
6. Describe the accounting, billing and resource pricing services in Aneka.
7. Describe the resource reservation services in Aneka
8. Describe the Scheduling and execution services in Aneka/
9. Draw and explain infrastructural organization of Aneka Clouds.
10. Draw and explain logical organization of Aneka clouds.
11. State and explain the Aneka Cloud deployment models.
12. What is Aneka SDK? Describe the Aneka SDK component models.

✠ ✠ ✠

UNIT V

CLOUD APPLICATIONS

5.1 INTRODUCTION TO CLOUD COMPUTING APPLICATIONS

Cloud computing consists of three distinct types of computing services delivered remotely to clients via the internet. Clients typically pay a monthly or annual service fee to providers, to gain access to systems that deliver software as a service, platforms as a service and infrastructure as a service to subscribers. Clients who subscribe to cloud computing services can reap a variety of benefits, depending on their particular business needs at a given point in time. The days of large capital investments in software and IT infrastructure are now a thing of the past for any enterprise that chooses to adopt the cloud computing model for procurement of IT services. The ability to access powerful IT resources on an incremental basis is leveling the playing field for small and medium sized organizations, providing them with the necessary tools and technology to compete in the global marketplace, without the previously requisite investment in on premise IT resources. Clients who subscribe to computing services delivered via the "cloud" are able to greatly reduce the IT service expenditures for their organizations; and gain access to more agile and flexible enterprise level computing services, in the process.

SaaS

- Saas (Software as a Service) provides clients with the ability to use software applications on a remote basis via an internet web browser. Software as a service is also referred to as "software on demand" .
- Clients can access SaaS applications from anywhere via the web because service providers host applications and their associated data at their location. The primary benefit of SaaS, is a lower cost of use, since subscriber fees require a much smaller investment than what is typically encountered under the traditional model of software delivery. Licensing fees, installation costs, maintenance fees and support fees that are routinely associated with the traditional model of software delivery can be virtually eliminated by subscribing to the SaaS model of software delivery. Examples of SaaS include: Google Applications and internet based email applications like Yahoo! Mail, Hotmail and Gmail.

PaaS

- PaaS (Platform as a Service) provides clients with the ability to develop and publish customized applications in a hosted environment via the web. It represents a new model for software development that is rapidly increasing in its popularity. An example of PaaS is Salesforce.com. PaaS provides a framework for agile software development, testing, deployment and maintenance in an integrated environment. Like SaaS, the primary benefit of PaaS, is a lower cost of use, since subscriber fees require a much smaller investment than what is typically encountered when implementing traditional tools for software development, testing and deployment. PaaS providers handle platform maintenance and system upgrades, resulting in a more efficient and cost effective solution for enterprise software development.

IaaS

- IaaS (Infrastructure as a Service) allows clients to remotely use IT hardware and resources on a "pay-as-you-go" basis. It is also referred to as HaaS (hardware as a service). Major IaaS players include companies like IBM, Google and Amazon.com. IaaS employs virtualization, a method of creating and managing infrastructure resources in the "cloud" . IaaS provides small start up firms with a major advantage, since it allows them to gradually expand their IT infrastructure without the need for large capital investments in hardware and peripheral systems.

5.2 SCIENTIFIC APPLICATIONS OF CLOUD COMPUTING

- Scientific applications are a sector that is increasingly using cloud computing systems and technologies. The immediate benefit seen by researchers and academics is the potentially infinite availability of computing resources and storage at sustainable prices compared to a complete in-house deployment. Cloud computing systems meet the needs of different types of applications in the scientific domain: High-Performance Computing (HPC) applications, High-Throughput Computing (HTC) applications, and data-intensive applications. The opportunity to use cloud resources is even more appealing because minimal changes need to be made to existing applications in order to leverage cloud resources.
- The most relevant option is IaaS solutions, which offer the optimal environment for running bag-of-tasks applications and workflows. Virtual machine instances are opportunely customized to host the required software stack for running such applications and coordinated together with distributed computing middleware capable of interacting with cloud-based infrastructures. PaaS solutions have been considered as well. They allow scientists to explore new programming models for tack- ling computationally challenging problems. Applications have been redesigned and implemented on top of cloud programming application models and platforms to leverage their unique capabilities. For instance, the MapReduce programming model provides scientists with a very simple and effective model for building applications that need to process large datasets. Therefore it has been widely used to develop data-intensive scientific applications. Problems that require a higher degree of flexibility in terms of structuring of their computation model can leverage platforms such as Aneka, which supports MapReduce and other programming models. We now discuss some interesting case studies in which Aneka has been used.

5.2.1 Healthcare: ECG Analysis in the Cloud

- Healthcare is a domain in which computer technology has found several and diverse applications: from supporting the business functions to assisting scientists in developing solutions to cure diseases.
- An important application is the use of cloud technologies to support doctors in providing more effective diagnostic processes. In particular, here we discuss electrocardiogram (ECG) data analysis on the cloud.
- The capillary development of Internet connectivity and its accessibility from any device at any time has made cloud technologies an attractive option for developing health-monitoring systems. ECG data analysis and monitoring constitute a case that naturally fits into this scenario. ECG is the electrical manifestation of the contractile activity of the heart's myocardium. This activity produces a specific waveform that is repeated over time and that represents the heartbeat. The analysis of the shape of the ECG waveform is used to identify arrhythmias and is the most common way to detect heart disease. Cloud computing technologies allow the remote monitoring of a patient's heartbeat data, data analysis in minimal time, and the notification of first-aid personnel and doctors should these data reveal potentially dangerous conditions. This way a patient at risk can be constantly monitored without going to a hospital for ECG analysis. At the same time, doctors and first-aid personnel can instantly be notified of cases that require their attention.
- An illustration of the infrastructure and model for supporting remote ECG monitoring is shown in Figure 5.1. Wearable computing devices equipped with ECG sensors constantly monitor the patient's heartbeat. Such information is transmitted to the patient's mobile device, which will eventually forward it to the cloud-hosted Web service for analysis. The Web service forms the front-end of a platform that is entirely hosted in the cloud and that leverages the three layers of the cloud computing stack: SaaS, PaaS, and IaaS. The Web service constitute the SaaS application that will store ECG data in the Amazon S3 service and issue a processing request to the scalable cloud plat- form. The runtime platform is composed of a dynamically sizable number of instances running the workflow engine and Aneka. The number of workflow engine instances is controlled according to the number of requests in the queue of each instance, while Aneka controls the number of EC2 instances used to execute the single tasks defined by the workflow engine for a single ECG processing job. Each of these jobs consists

of a set of operations involving the extraction of the waveform from the heartbeat data and the comparison of the waveform with a reference waveform to detect anomalies. If anomalies are found, doctors and first-aid personnel can be notified to act on a specific patient.

- Even though remote ECG monitoring does not necessarily require cloud technologies, cloud computing introduces opportunities that would be otherwise hardly achievable. The first advantage is the elasticity of the cloud infrastructure that can grow and shrink according to the requests served. As a result, doctors and hospitals do not have to invest in large computing infrastructures designed after capacity planning, thus making more effective use of budgets. The second advantage is ubiquity. Cloud computing technologies have now become easily accessible and promise to deliver systems with minimum or no downtime. Computing systems hosted in the cloud are accessible from any Internet device through simple interfaces (such as SOAP and REST-based Web services). This makes these systems not only ubiquitous, but they can also be easily integrated with other systems maintained on the hospital's premises. Finally, cost savings constitute another reason for the use of cloud technology in healthcare. Cloud services are priced on a pay-per-use basis and with volume prices for large numbers of service requests. These two models provide a set of flexible options that can be used to price the service, thus actually charging costs based on effective use rather than capital costs.

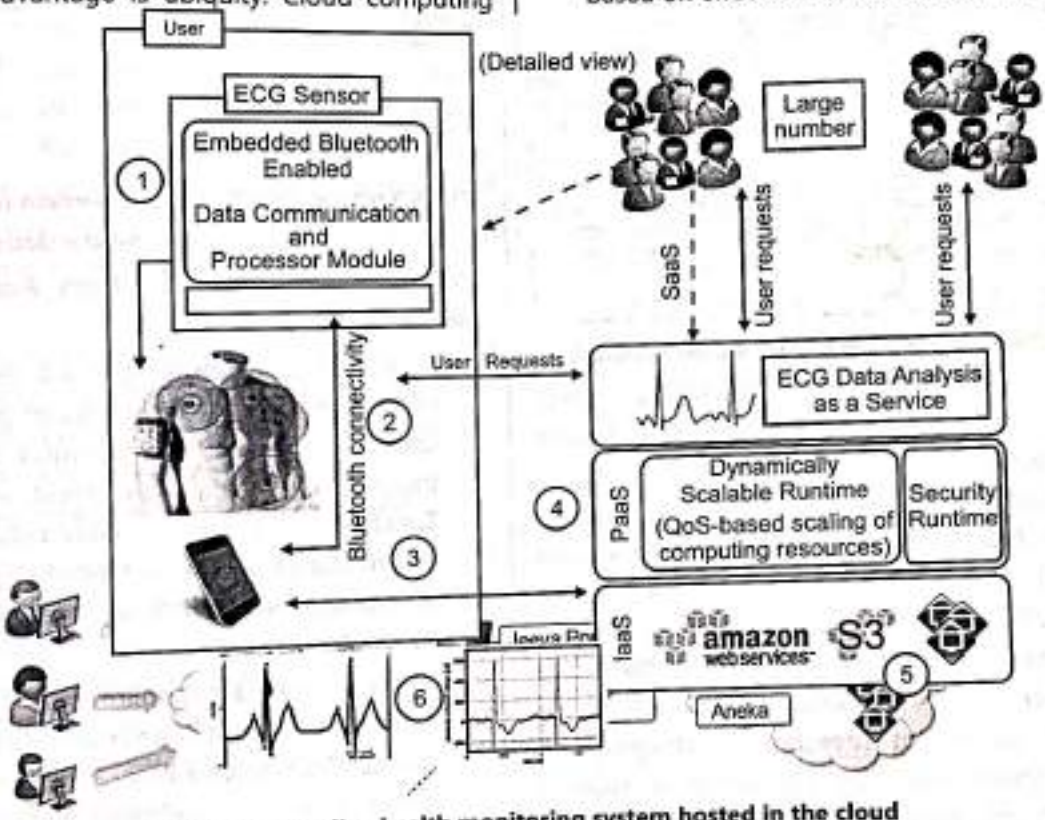


Fig. 5.1 : An online health monitoring system hosted in the cloud

5.2.2 Biology: Protein Structure Prediction

- Applications in biology often require high computing capabilities and often operate on large data- sets that cause extensive I/O operations. Because of these requirements, biology applications have often made extensive use of supercomputing and cluster computing infrastructures. Similar capabilities can be leveraged on demand using cloud computing technologies in a more dynamic fashion, thus opening new opportunities for bioinformatics applications.
- Protein structure prediction is a computationally intensive task that is fundamental to different types of research in the life sciences. Among these is the design of new drugs for the treatment of diseases. The geometric structure of a protein cannot be directly inferred from the sequence of genes that compose its structure, but it is the result of complex computations aimed at identifying the structure that minimizes the required energy. This task requires the investigation of

a space with a massive number of states, consequently creating a large number of computations for each of these states. The computational power required for protein structure prediction can now be acquired on demand, without owning a cluster or navigating the bureaucracy to get access to parallel and distributed computing facilities. Cloud computing grants access to such capacity on a pay-per-use basis.

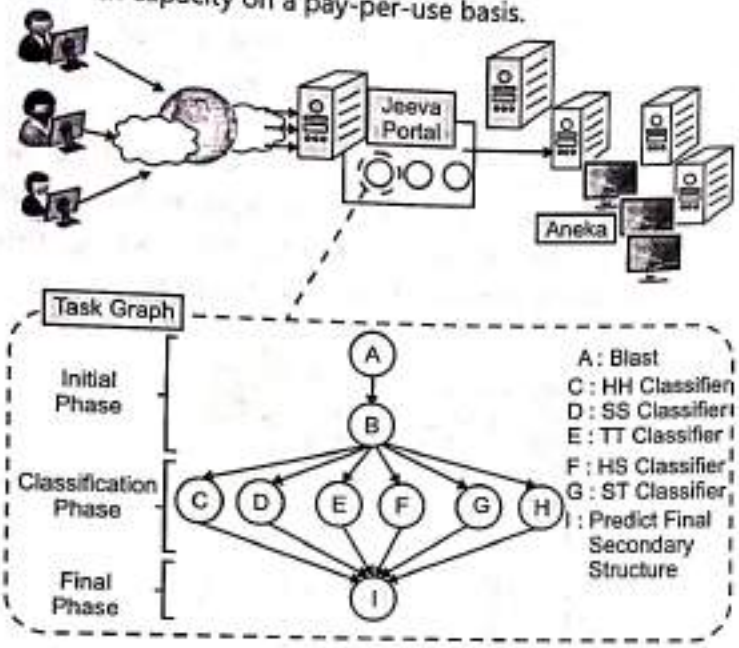


Fig. 5.2 : Architecture and overview of the Jeeva Portal.

- One project that investigates the use of cloud technologies for protein structure prediction is Jeevaan integrated Web portal that enables scientists to offload the prediction task to a computing cloud based on Aneka (see Figure 5.2). The prediction task uses machine learning techniques (support vector machines) for determining the secondary structure of proteins. These techniques translate the problem into one of pattern recognition, where a sequence has to be classified into one of three possible classes (E, H, and C). A popular implementation based on support vector machines divides the pattern recognition problem into three phases:
  1. Initialization.
  2. Classification.
  3. Final phase.
- Even though these three phases have to be executed in sequence, it is possible to take advantage of parallel execution in the classification phase, where multiple classifiers are executed concurrently. This creates the opportunity to sensibly reduce the computational time of the prediction. The prediction algorithm is then translated into a task graph that is submitted to Aneka. Once the task is completed, the middleware makes the results available for visualization through the portal.
- The advantage of using cloud technologies (i.e., Aneka as scalable cloud middleware) versus conventional grid infrastructures is the capability to leverage a scalable computing infrastructure that can be grown and shrunk on demand. This concept is distinctive of cloud technologies and constitute strategic advantage when applications are offered and delivered as a service.

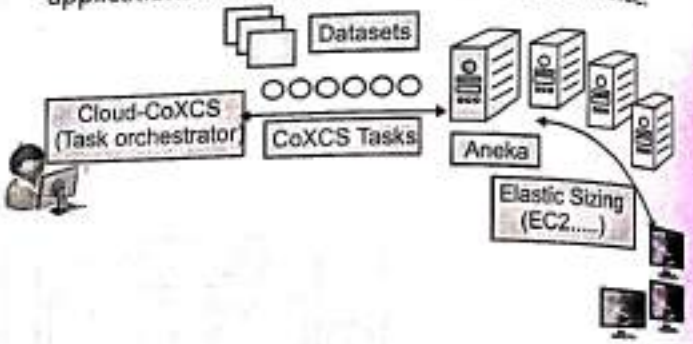


Fig. 5.3 :Cloud-CoXCS: An environment for micro array data processing on the cloud.

Biology: Gene Expression Data Analysis for Cancer Diagnosis

- Gene expression profiling is the measurement of the expression levels of thousands of genes at once. It is used to understand the biological processes that are triggered by medical treatment at a cellular level. Together with protein structure prediction, this activity is a fundamental component of drug design, since it allows scientists to identify thee effect sofa specific treatment.
- Another important application of gene expression profiling is cancer diagnosis and treatment. Cancer is a disease characterized by uncontrolled cell growth and proliferation. This behavior occurs because genes regulating the cell growth mutate. This means that all the cancerous cells contain mutated genes. In this context, gene expression profiling is utilized to provide a more accurate classification of tumors. The classification of gene expression data samples into distinct classes is a challenging task. The dimensionality of typical gene expression datasets ranges from several thousands to over tens of thousands of genes. However, only small sample sizes are typically available for analysis.

- This problem is often approached with learning classifiers, which generate a population of condition-action rules that guide the classification process. Among these, the eXtended Classifier System (XCS) has been successfully utilized for classifying large datasets in the bioinformatics and computer science domains. However, the effectiveness of XCS, when confronted with high dimensional datasets (such as micro array gene expression data sets), has not been explored in detail. A variation of this algorithm, CoXCS has proven to be effective in these conditions. CoXCS divides the entire search space into sub-domains and employs the standard XCS algorithm in each of these sub-domains. Such a process is computationally intensive but can be easily parallelized because the classifications problems on the sub-domains can be solved concurrently. Cloud-CoXCS (see Fig. 5.3) is a cloud-based implementation of CoXCS that leverages Aneka to solve the classification problems in parallel and compose their outcomes. The algorithm is controlled by strategies, which define the way the outcomes are composed together and whether the process needs to be iterated.
- Because of the dynamic nature of XCS, the number of required compute resources to execute it can vary over time. Therefore, the use of scalable middleware such as Aneka offers a distinctive advantage.

5.2.3 Geoscience: Satellite Image Processing

- Geoscience applications collect, produce, and analyze massive amounts of geospatial and non spatial data. As the technology progresses and our planet becomes more instrumented (i.e., through the deployment of sensors and satellites for monitoring), the volume of data that needs to be processed increases significantly. In particular, the Geographic Information System (GIS) is a major element of geoscience applications. GIS applications capture, store, manipulate, analyze, manage, and present all types of geographically referenced data. This type of information is now becoming increasingly relevant to a wide variety of application domains: from advanced farming to civil security and natural resources management. As a result, a considerable amount of geo-referenced data is ingested into computer systems for further processing and analysis. Cloud computing is an attractive option for executing these demanding tasks and extracting meaningful information to support decision makers.
- Satellite remote sensing generates hundreds of gigabytes of raw images that need to be further processed to become the basis of several different GIS products. This process requires both I/O and compute-intensive tasks. Large images need to be moved from a ground station's local storage to compute facilities, where several transformations and corrections are applied. Cloud computing pro- vides the appropriate infrastructure to support such application scenarios. A cloud-based implementation of such a workflow has been developed by the Department of Space, Government of India. The system shown in Fig. 5.4 integrates several technologies across the entire computing stack. A SaaS application provides a collection of services for such tasks as geocode generation and data visualization. At the PaaS level, Aneka controls the importing of data into the virtualized infrastructure and the execution of image-processing tasks that produce the desired outcome from raw satellite images. The platform leverages a Xen private cloud and the Aneka technology to dynamically provision the required resources (i.e., grow or shrink) on demand.
- The project demonstrates how cloud computing technologies can be effectively employed to off- load local computing facilities from excessive workloads and leverage more elastic computing infrastructures.

5.3 BUSINESS AND CONSUMER APPLICATIONS

- The business and consumer sector is the one that probably benefits the most from cloud computing technologies. On one hand, the opportunity to transform capital costs into operational costs makes clouds an attractive option for all enterprises that are IT-centric. On the other hand, the sense of ubiquity that the cloud offers for accessing data and services makes it interesting for end users as well. Moreover, the elastic nature of cloud technologies does not require huge up-front investments, thus allowing new ideas to be quickly translated into products and services that can comfortably grow with the demand. The combination of all these elements has made cloud

computing the preferred technology for a wide range of applications, from CRM and ERP systems to productivity and social-networking applications.

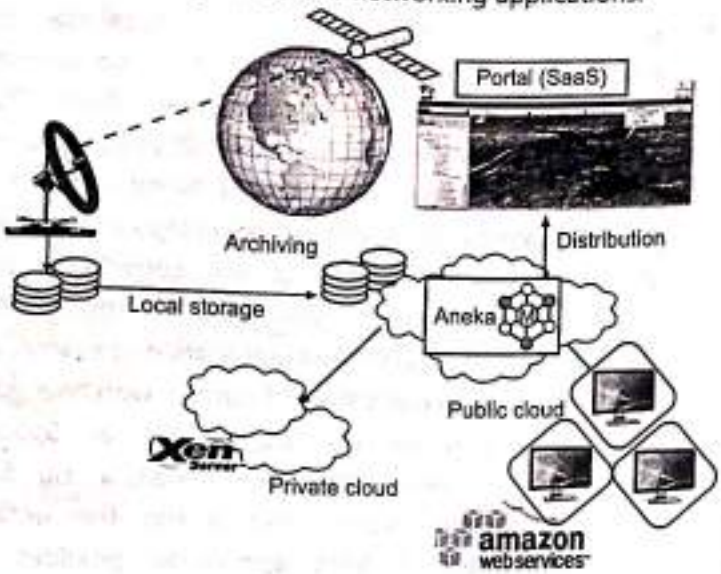


Fig. 5.4 : A cloud environment for satellite data processing

5.3.1 CRM and ERP

- Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP) applications are market segments that are flourishing in the cloud, with CRM applications the more mature of the two. Cloud CRM applications constitute a great opportunity for small enterprises and start-ups to have fully functional CRM software without large up-front costs and by paying subscriptions. Moreover, CRM is not an activity that requires specific needs, and it can be easily moved to the cloud. Such a characteristic, together with the possibility of having access to your business and customer data from everywhere and from any device, has fostered the spread of cloud CRM applications. ERP solutions on the cloud are less mature and have to compete with well-established in-house solutions. ERP systems integrate several aspects of an enterprise: finance and accounting, human resources, manufacturing, supply chain management, project management, and CRM. Their goal is to provide a uniform view and access to all operations that need to be performed to sustain a complex organization. Because of the organizations that they target, the transition to cloud-based models is more difficult : he cost advantage over the long term might not be clear, and the switch to the cloud could be difficult if organizations already have large ERP installations. For this reason cloud ERP solutions are less popular than CRM solutions at this time.

Salesforce.com

- Salesforce.com is probably the most popular and developed CRM solution available today. As of today more than 100,000 customers have chosen Safesforce.com to implement their CRM solutions. The application provides customizable CRM solutions that can be integrated with additional features developed by third parties. Salesforce.com is based on the Force.com cloud development platform. This represents scalable and high-performance middleware executing all the operations of all Salesforce.com applications.
- The architecture of the Force.complatform is shown in Fig. 5.5. Initially designed to sup- port scalable CRM applications, the platform has evolved to support the entire life cycle of a wider range of cloud applications by implementing a flexible and scalable infrastructure. At the core of the platform resides its metadata architecture, which provides the system with flexibility and scalability. Rather than being built on top of specific components and tables, application core logic and business rules are saved as metadata into the Force.comstore.
- Both application structure and application data are stored in the store. A runtime engine executes application logic by retrieving its metadata and then performing the operations on the data. Although running in isolated containers, different applications logically share the same database structure, and the runtime engine exe- cutes all of them uniformly. A full-text search engine supports the runtime engine. This allows application users to have an effective user experience despite the large amounts of data that need to be crawled. The search engine maintains its indexing data in a separate store and is constantly updated by background processes triggered by user interaction.

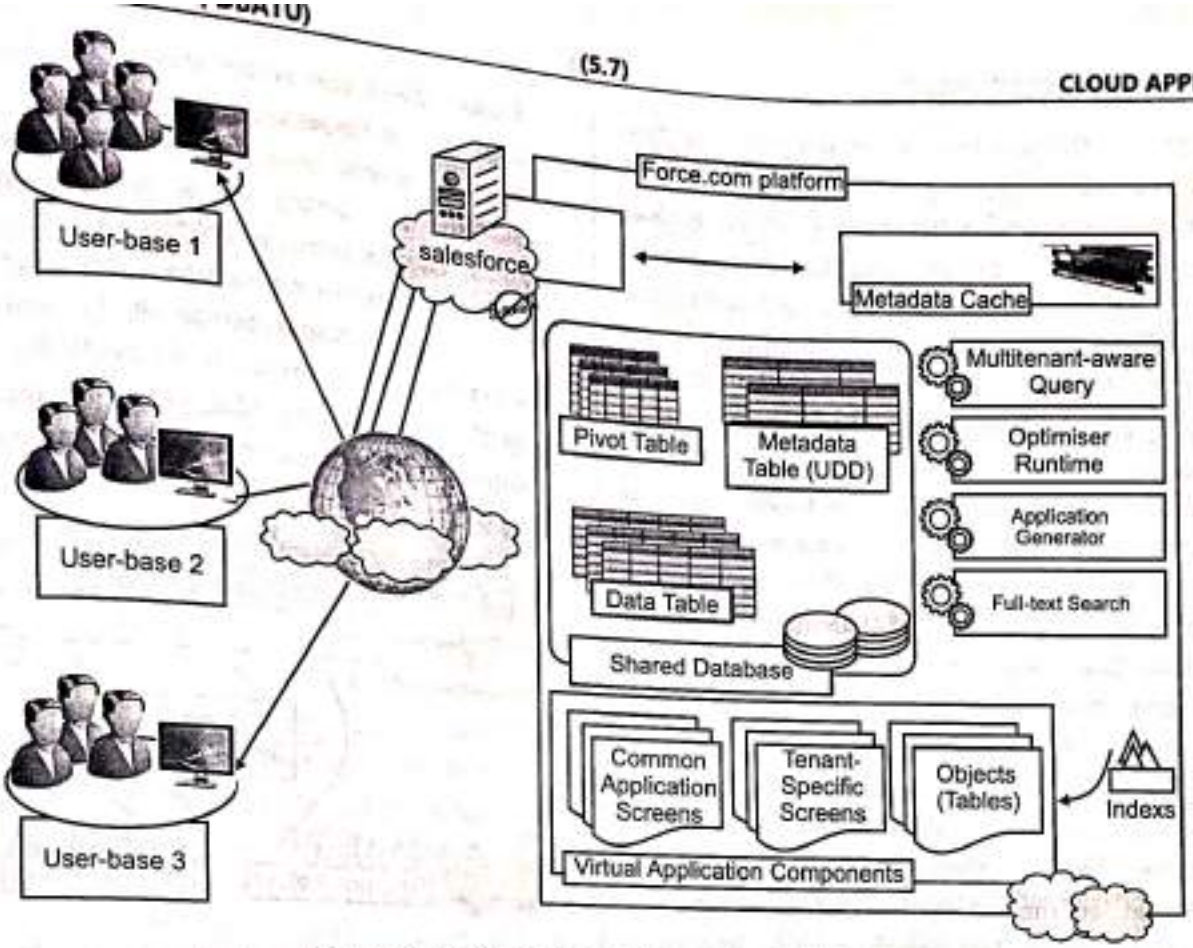


Fig. 5.5 : Salesforce.com and Force.com architecture.

- Users can customize their application by leveraging the "native" Force.com application framework or by using programmatic APIs in the most popular programming languages. The application framework allows users to visually define either the data or the core structure of a Force.com application, while the programmatic APIs provide them with a more conventional way for developing applications that relies on Web services to interact with the platform. Customization of application processes and logic can also be implemented by developing scripts in APEX. This is a Java-like language that provides object-oriented and procedural capabilities for defining either scripts executed on demand or triggers. APEX also offers the capability of expressing searches and queries to have complete access to the data managed by the Force.com platform.

Microsoft Dynamics CRM

- Microsoft Dynamics CRM is the solution implemented by Microsoft for customer relationship management. Dynamics CRM is available either for installation on the enterprise's premises or as an online solution priced as a monthly per-user subscription.
- The system is completely hosted in Microsoft's datacenters across the world and offers to customers a 99.9% SLA, with bonus credits if the system does not fulfill the agreement. Each CRM instance is deployed on a separate database, and the application provides users with facilities for marketing, sales, and advanced customer relationship management. Dynamics CRM Online features can be accessed either through a Web browser interface or programmatically by means of SOAP and RESTful Web services. This allows Dynamics CRM to be easily integrated with both other Microsoft products and line-of-business applications. Dynamics CRM can be extended by developing plug-ins that allow implementing specific behaviors triggered on the occurrence of given events. Dynamics CRM can also leverage the capability of Windows Azure for the development and integration of new features.

NetSuite

- NetSuite provides a collection of applications that help customers manage every aspect of the business enterprise. Its offering is divided into three major products: NetSuite Global ERP, NetSuite Global CRM1 , and NetSuite Global Ecommerce. Moreover, an all-in-

one solution: NetSuite One World, integrates all three products together.

- The services NetSuite delivers are powered by two large datacenters on the East and West coasts of the United States, connected by redundant links. This allows NetSuite to guarantee 99.5% uptime to its customers. Besides the prepackaged solutions, NetSuite also provides an infrastructure and a development environment for implementing customized applications. The NetSuite Business Operating System (NS-BOS) is a complete stack of technologies for building SaaS business applications that leverage the capabilities of NetSuite products. On top of the SaaS infrastructure, the NetSuite Business Suite components offer accounting, ERP, CRM, and ecommerce capabilities.
- An online development environment, SuiteFlex, allows integrating such capabilities into new Web applications, which are then packaged for distribution by SuiteBundler. The entire infrastructure is hosted in the NetSuite datacenters, which provide warranties regarding application uptime and availability.

Productivity

- Productivity applications replicate in the cloud some of the most common tasks that we are used to performing on our desktop: from document storage to office automation and complete desktop environments hosted in the cloud.

Dropbox and iCloud

- One of the core features of cloud computing is availability anywhere, at any time, and from any Internet-connected device. Therefore, document storage constitutes a natural application for such technology. Online storage solutions preceded cloud computing, but they never became popular. With the development of cloud technologies, online storage solutions have turned into SaaS applications and become more usable as well as more advanced and accessible.
- Perhaps the most popular solution for online document storage is Dropbox, an online application that allows users to synchronize any file across any platform and any device in a seamless manner (see Fig. 5.6). Dropbox provides users with a free amount of storage that is accessible through the abstraction of a folder. Users can either access their Dropbox folder through a browser or by downloading and installing a Dropbox client, which provides access to the online storage by means of a special folder. All the modifications into this folder are silently synched so that changes are notified to all the local instances of the Dropbox folder across all the devices. The key advantage of Dropbox is its availability on different platforms (Windows, Mac, Linux, and mobile) and the capability to work seamlessly and transparently across all of them.

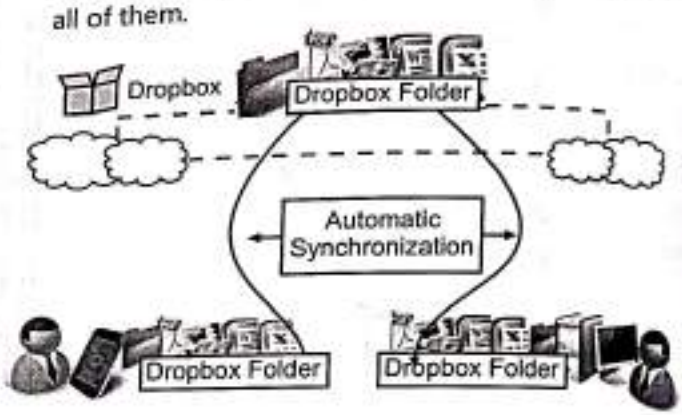


Fig. 5.6 : Dropbox usage scenario.

- Another interesting application in this area is iCloud, a cloud-based document-sharing application provided by Apple to synchronize iOS-based devices in a completely transparent manner. Unlike Dropbox, which provides synchronization through the abstraction of a local folder, iCloud has been designed to be completely transparent once it has been set up. Documents, photos, and videos are automatically synched as changes are made, without any explicit operation. This allows the system to efficiently automate common operations without any human intervention: taking a picture with your iPhone and having it automatically available in iPhoto on your Mac at home; editing a document on the iMac at home and having the changes updated in your iPad. Unfortunately, this capability is limited to iOS devices, and currently there are no plans to provide iCloud with a Web-based interface that would make user content accessible from even unsupported platforms.
- There are other solutions for online document sharing such as Windows Live, Amazon Cloud Drive, and CloudMe, that are popular and that we did not cover. These solutions offer more or less the same capabilities of those we've discussed, with different levels of integration between platform and devices.

Google Docs

- Google Docs is a SaaS application that delivers the basic office automation capabilities with sup- port for collaborative editing over the Web. The application is executed on top of the Google distributed computing infrastructure, which allows the system to dynamically scale according to the number of users using the service.
- Google Docs allows users to create and edit text documents, spreadsheets, presentations, forms, and drawings. It aims to replace desktop products such as Microsoft Office and OpenOffice and provide similar interface and functionality as a cloud service. It supports collaborative editing over the Web for most of the applications included in the suite. This eliminates tedious emailing and synchronization tasks when documents need to be edited by multiple users. By being stored in the Google infrastructure, these documents are always available from anywhere and from any device that is connected to the Internet. Moreover, the suite allows users to work offline if Internet connectivity is not available. Support for various formats such as those that are produced by the most popular desktop office solutions allows users to easily import and move documents in and out of GoogleDocs, thus eliminating barriers to the use of this application.
- Google Docs is a good example of what cloud computing can deliver to end users: ubiquitous access to resources, elasticity, absence of installation and maintenance costs, and delivery of core functionalities as a service.

Cloud Desktops: EyeOS and XIOS/3

- Asynchronous JavaScript and XML (AJAX) technologies have considerably augmented the capabilities that can be implemented in Web applications. This is a fundamental aspect for cloud computing, which delivers a considerable amount of its services through the Web browser. Together with the opportunity to leverage large-scale storage and computation, this technology has made possible the replication of complex desktop environments in the cloud and made them available through the Web browser. These applications, called cloud desktops, are rapidly gaining in popularity.

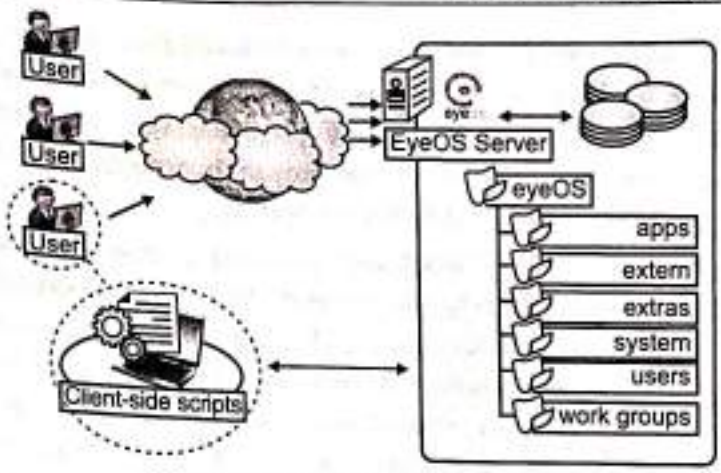


Fig. 5.7 : EyeOS architecture

- EyeOS is one of the most popular Web desktop solutions based on cloud technologies. It replicates the functionalities of a classic desktop environment and comes with pre-installed applications for the most common file and document management tasks (see Fig. 5.7). Single users can access the EyeOS desktop environment from anywhere and through any Internet-connected device, whereas organizations can create a private EyeOS Cloud on their premises to virtualize the desktop environment of their employees and centralize their management.
- The EyeOS architecture is quite simple: On the server side, the EyeOS application maintains the information about user profiles and their data, and the client side constitutes the access point for users and administrators to interact with the system. EyeOS stores the data about users and applications on the server file system. Once the user has logged in by providing credentials, the desktop environment is rendered in the client's browser by downloading all the JavaScript libraries required to build the user interface and implement the core functionalities of EyeOS. Each application loaded in the environment communicates with the server by using AJAX; this communication model is used to access user data as well as to perform application operations: editing documents, visualizing images, copying and saving files, sending email, and chatting.
- EyeOS also provides APIs for developing new applications and integrating new capabilities into the system. EyeOS applications are server-side components that are defined by at least two files (stored in the eyeos/apps/appnamedirectory):

appname.phpand appname.js. The first file defines and implements all the operations that the application exposes; the JavaScript file contains the code that needs to be loaded in the browser in order to provide user interaction with the application.

- Xcerion XML Internet OS/3 (XIOS/3) is another example of a Web desktop environment. The service is delivered as part of the CloudMe application, which is a solution for cloud document storage. The key differentiator of XIOS/3 is its strong leverage of XML, used to implement many of the tasks of the OS: rendering user interfaces, defining application business logics, structuring file system organization, and even application development. The architecture of the OS concentrates most of the functionalities on the client side while implementing server-based functionalities by means of XML Web services. The client side renders the user interface, orchestrates processes, and provides data-binding capabilities on XML data that is exchanged with Web services. The server is responsible for implementing core functions such as transaction management for documents edited in a collaborative mode and core logic of installed applications into the environment. XIOS/3 also provides an environment for developing applications (XIDE), which allows users to quickly develop complex applications by visual tools for the user interface and XML documents for business logic.
- XIOS/3 is released as open-source software and implements a marketplace where third parties can easily deploy applications that can be installed on top of the virtual desktop environment. It is possible to develop any type of application and feed it with data accessible through XML Web services: developers have to define the user interface, bind UI components to service calls and operations, and provide the logic on how to process the data. XIDE will package this information into a proper set of XML documents, and the rest will be performed by an XML virtual machine implemented in XIOS.
- XIOS/3 is an advanced Web desktop environment that focuses on the integration of services into the environment by means of XML-based services and that simplifies collaboration with peers.

5.3.2 Social Networking

- Social networking applications have grown considerably in the last few years to become the most active sites on the Web. To sustain their traffic and serve millions of users seamlessly, services such as Twitter and Facebook have leveraged cloud computing technologies. The possibility of continuously adding capacity while systems are running is the most attractive feature for social networks, which constantly increase their user base.

Facebook

- Facebook is probably the most evident and interesting environment in social networking. With more than 800 million users, it has become one of the largest Websites in the world. To sustain this incredible growth, it has been fundamental that Facebook be capable of continuously adding capacity and developing new scalable technologies and software systems while maintaining high performance to ensure a smooth user experience.
- Currently, the social network is backed by two data centers that have been built and optimized to reduce costs and impact on the environment. On top of this highly efficient infrastructure, built and designed out of inexpensive hardware, a completely customized stack of opportunely modified and refined open-source technologies constitutes the back-end of the largest social network. Taken all together, these technologies constitute a powerful platform for developing cloud applications.
- This platform primarily supports Facebook itself and offers APIs to integrate third-party applications with Facebook's core infrastructure to deliver additional services such as social games and quizzes created by others.
- The reference stack serving Facebook is based on LAMP (Linux, Apache, MySQL, and PHP). This collection of technologies is accompanied by a collection of other services developed in-house. These services are developed in a variety of languages and implement specific functionalities such as search, news feeds, notifications, and others. While serving page requests, the social graph of the user is composed. The social graph identifies a collection of interlinked information that is of relevance for a given user. Most of the user

data are served by querying a distributed cluster of MySQL instances, which mostly contain key-value pairs. These data are then cached for faster retrieval. The rest of the relevant information is then composed together using the services mentioned before. These services are located closer to the data and developed in languages that provide better performance than PHP.

- The development of services is facilitated by a set of internally developed tools. One of the core elements is Thrift. This is a collection of abstractions (and language bindings) that allow cross-language development. Thrift allows services developed in different languages to communicate and exchange data. Bindings for Thrift in different languages take care of data serialization and deserialization, communication, and client and server boilerplate code. This simplifies the work of the developers, who can quickly prototype services and leverage existing ones. Other relevant services and tools are Scribe, which aggregates streaming log feeds, and applications for alerting and monitoring.

5.3.3 Media Applications

- Media applications are a niche that has taken a considerable advantage from leveraging cloud computing technologies. In particular, video-processing operations, such as encoding, transcoding, com- position, and rendering, are good candidates for a cloud-based environment. These are computationally intensive tasks that can be easily offloaded to cloud computing infrastructures.

Animoto

- Animoto is perhaps the most popular example of media applications on the cloud. The Website provides users with a very straightforward interface for quickly creating videos out of images, music, and video fragments submitted by users. Users select a specific theme for a video, upload the photos and videos and order them in the sequence they want to appear, select the song for the music, and render the video. The process is executed in the background and the user is notified via email once the video is rendered.
- The core value of Animoto is the ability to quickly create videos with stunning effects without user intervention. A proprietary Artificial Intelligence (AI) engine, which selects the animation and transition effects according to pictures and music, drives the rendering operation. Users only have to define the storyboard by organizing pictures and videos into the desired sequence. If users don't like the result, the video can be rendered again and the engine will select a different composition, thus producing a different out come every time. The service allows users to create 30 second videos for free. By paying a monthly or a yearly subscription it is possible to produce videos of any length and to choose among a wider range of templates.
- The infrastructure supporting Animoto is complex and is composed of different systems that all need to scale (see Fig. 5.8). The core function is implemented on top of the Amazon Web Services infrastructure. In particular, it uses Amazon EC2 for the Web front-end and the worker nodes; Amazon S3 for the storage of pictures, music, and videos; and Amazon SQS for connecting all the components. The system's auto-scaling capabilities are managed by Rightscale, which monitors the load and controls the creation of new worker instances as well as their reclaim. Front-end nodes collect the components required to make the video and store them in S3. Once the storyboard of the video is completed, a video-rendering request is entered into a SQS queue. Worker nodes pick up rendering requests and perform the rendering. When the process is completed, another message is entered into a different SQS queue and another request is served. This last queue is cleared routinely and users are notified about the completion. The life of EC2 instances is controlled by Rightscale, which constantly monitors the load and the performance of the system and decides whether it is necessary to grow or shrink.
- The architecture of the system has proven to be very scalable and reliable by using up to 4,000 servers on EC2 in peak times without dropping requests but simply causing acceptable temporary delays for the rendering process.

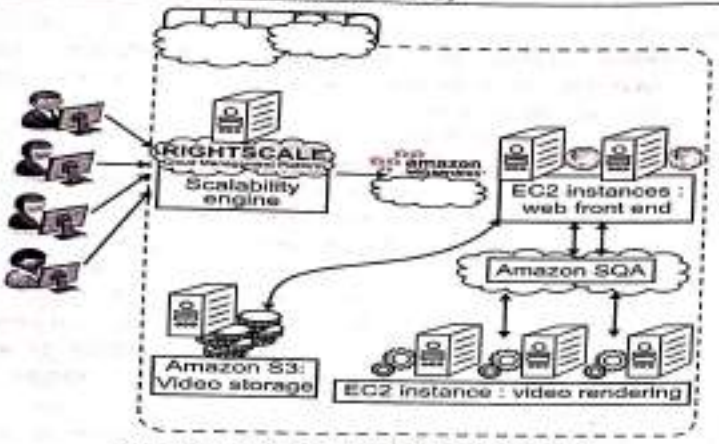


Fig. 5.8 : Animoto reference architecture

Maya Rendering with Aneka

- Interesting applications of media processing are found in the engineering disciplines and the movie production industry. Operations such as rendering of models are now an integral part of the design workflow, which has become computationally demanding. The visualization of mechanical models is not only used at the end of the design process, it is iteratively used to improve the design. It is then fundamental to perform such tasks as fast as possible. Cloud computing provides engineers with the necessary computing power to make this happen.
- A private cloud solution for rendering train designs has been implemented by the engineering department of GoFront group, a division of China Southern Railway (see Fig. 5.9). The department is responsible for designing models of high-speed electric locomotives, metro cars, urban transportation vehicles, and motor trains. The design process for prototypes requires high-quality, Three-Dimensional (3D) images. The analysis of these images can help engineers identify problems and correct their design. Three-dimensional rendering tasks take considerable amounts of time, especially in the case of huge numbers of frames, but it is critical for the department to reduce the time spent in these iterations. This goal has been achieved by leveraging cloud computing technologies, which turned the network of desktops in the department into a desktop cloud managed by Aneka.

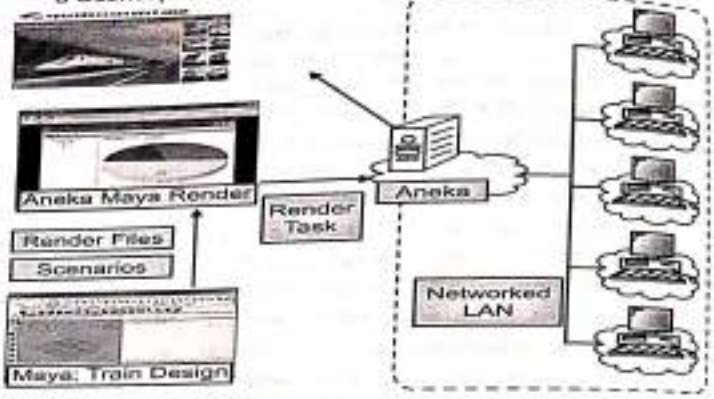


Fig. 5.9 : 3D rendering on private clouds

- The implemented system includes a specialized client interface that can be used by GoFrontengineers to enter all the details of the rendering process (the number of frames, the number of cameras, and other parameters). The application is used to submit the rendering tasks to the Aneka Cloud, which distributes the load across all the available machines. Every rendering task triggers the execution of the local Maya batch renderer and collects the result of the execution. The renders are then retrieved and put all together for visualization.
- By turning the local network into a private cloud, the resources of which can be used off-peak (i.e., at night, when desktops are not utilized), it has been possible for GoFront to sensibly reduce the time spent in the rendering process from days to hours.

Video Encoding on the Cloud: Encoding.com

- Video encoding and transcoding are operations that can greatly benefit from using cloud technologies: They are computationally intensive and potentially require considerable amounts of storage. Moreover, with the continuous improvement of mobile devices as well as the diffusion of the Internet, requests for video content have significantly increased. The variety of devices with video playback capabilities has led to an explosion of formats through which a video can be delivered. Software and hardware for video encoding

and transcoding often have prohibitive costs or are not flexible enough to support conversion from any format to any format. Cloud technologies present an opportunity for turning these tedious and often demanding tasks into services that can be easily integrated into a variety of work flow so made available to everyone according to their needs.

- Encoding.com is a software solution that offers video-transcoding services on demand and leverages cloud technology to provide both the horsepower required for video conversion and the storage for staging videos. The service integrates with both Amazon Web Services technologies (EC2, S3, and CloudFront) and Rackspace (Cloud Servers, Cloud Files, and Limelight CDNaccess). Users can access the services through a variety of interfaces: the Encoding.com Website, Web service XML APIs, desktop applications, and watched folders. To use the service, users have to specify the location of the video to transcode, the destination format, and the target location of the video. Encoding.com also offers other video-editing operations such as the insertion of thumb- nails, watermarks, or logos. Moreover, it extend sits capabilities to audio and image conversion.
- The service provides various pricing options: monthly fee, pay-as-you-go (by batches), and special prices for high volumes. Encoding.com now has more than 2,000 customers and has already processed more than 10 million videos.

5.3.4 Multiplayer Online Gaming

- Online multiplayer gaming attracts millions of gamers around the world who share a common experience by playing together in a virtual environment that extends beyond the boundaries of a normal LAN. Online games support hundreds of players in the same session, made possible by the specific architecture used to forward interactions, which is based on game log processing. Players update the game server hosting the game session, and the server integrates all the updates into a log that is made available to all the players through a TCP port. The client software used for the game connects to the log port and, by reading the log, updates the local user interface with the actions of other players.

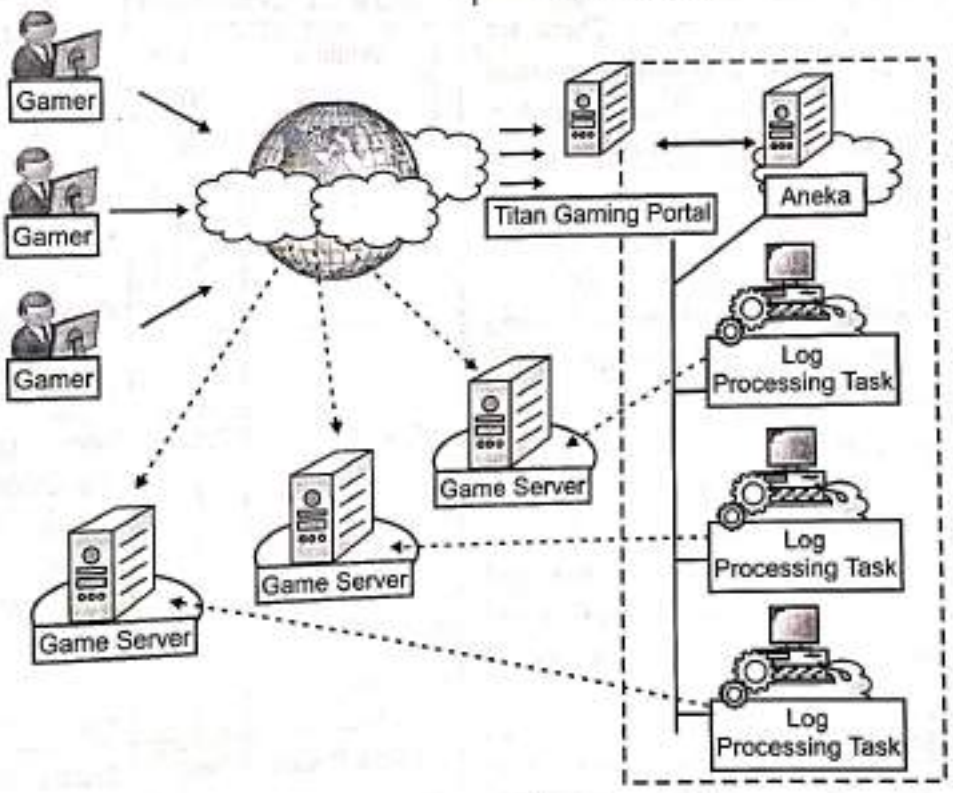


Fig. 5.10 : Scalable processing of logs for network games.

- Game log processing is also utilized to build statistics on players and rank them. These features constitute the additional value of online gaming portals that attract more and more gamers. The processing of game logs is a potentially compute-intensive operation that strongly depends on the number of players online

and the number of games monitored. Moreover, gaming portals are Web applications and therefore might suffer from the spiky behavior of users that can randomly generate large amounts of volatile work loads that do not justify capacity planning.

- The use of cloud computing technologies can provide the required elasticity for seamlessly processing these workloads and scale as required when the number of users increases. A prototypal implementation of cloud-based game log processing has been implemented by Titan Inc. (now Xfire), a company based in California that extended its gaming portal for offload game log processing to an Aneka Cloud. The prototype (shown in Fig. 5.10) uses a private cloud deployment that allowed TitanInc .to process concurrently multiple log sand sustain a larger number of users.

5.4 CLOUD PLATFORMS IN INDUSTRY

- Cloud computing allows end users and developers to leverage large distributed computing infrastructures. This is made possible thanks to infrastructure management software and distributed computing platforms offering on-demand compute, storage, and, on top of these, more advanced services. There are several different options for building enterprise cloud computing applications or for using cloud computing technologies to integrate and extend existing industrial applications. An overview of a few prominent cloud computing platforms and a brief description of the types of service they offer are shown in Table 5.1. A cloud computing system can be developed using either a single technology and vendor or a combination of them.
- This chapter presents some of the representative cloud computing solutions offered as Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) services in the market. It pro- vides some insights into and practical issues surrounding the architecture of the major cloud computing technologies and their service offerings.

5.4.1 Amazon Web Services

- Amazon Web Services (AWS) is a platform that allows the development of flexible applications by providing solutions for elastic infrastructure scalability, messaging, and data storage. The platform is accessible through SOAP or RESTful Web service interfaces and provides a Web-based console where users can handle administration and monitoring of the resources required, as well as their expenses computed on a pay-as-you-gobasis.
- Fig. 5.11 shows all the services available in the AWS ecosystem. At the base of the solution stack are services that provide raw compute and raw storage. Amazon Elastic Compute (EC2) and Amazon Simple Storage Service (S3). These are the two most popular services, which are generally complemented with other offerings for building a complete system. At the higher level, Elastic MapReduce and Auto Scaling provide additional capabilities for building smarter and more elastic computing systems. On the data side, Elastic Block Store (EBS), Amazon SimpleDB, Amazon RDS, and Amazon ElastiCache provide solutions for reliable data snapshots and the management of structured and semi structured data. Communication needs are covered at the networking level by Amazon Virtual Private Cloud (VPC), Elastic Load Balancing, Amazon Route 53, and Amazon Direct Connect. More advanced services for connecting applications are Amazon Simple Queue

Table 5.1 : Some Example Cloud Computing Offerings

Vendor / Product	Service Type	Description
Amazon Web Services	IaaS, PaaS, SaaS	Amazon Web Services (AWS) is a collection of Web services that provides developers with compute, storage, and more advanced services. AWS is mostly popular for IaaS services and primarily for its elastic compute serviceEC2.
Google App Engine	PaaS	Google App Engine is a distributed and scalable runtime for developing scalable Web applications based on Java and Python runtime environments. These are enriched with access to services that simplify the development of applications in a scalable manner.
Microsoft Azure	PaaS	Microsoft Azure is a cloud operating system that provides services for developing scalable applications based on the proprietary Hyper-V virtualization technology and the .NET framework.

SalesForce.com and Force.com	SaaS, PaaS	SalesForce.com is a Software-as-a-Service solution that allows prototyping of CRM applications. It leverages the Force.com platform, which is made available for developing new components and capabilities for CRM applications.
Heroku	PaaS	Heroku is a scalable runtime environment for building applications based on Ruby.
Right Scale	IaaS	Right scale is a cloud management platform with a single dash board to manage public and hybrid clouds.

- Service (SQS), Amazon Simple Notification Service (SNS), and Amazon Simple E-mail Service (SES). Other services include:
  - ➢ Amazon Cloud Front content delivery network solution
  - ➢ Amazon Cloud Watch monitoring solution for several Amazon services
  - ➢ Amazon Elastic Bean Stalk and Cloud Formation flexible application packaging and deployment
- As shown, AWS comprise a wide set of services. We discuss the most important services by examining the solutions proposed by AWS regarding compute, storage, communication, and complementary services.

5.4.2 Compute Services

- Compute services constitute the fundamental element of cloud computing systems. The fundamental service in this space is Amazon EC2, which delivers an IaaS solution that has served as a reference model for several offerings from other vendors in the same market segment. Amazon EC2 allows deploying servers in the form of virtual machines created as instances of a specific image. Images come with a preinstalled operating system and a software stack, and instances can be configured for memory, number of processors, and storage. Users are provided with credentials to remotely access the instance and further configure or install software if needed.

Amazon Machine Images

- Amazon Machine Images (AMIs) are templates from which it is possible to create a virtual machine. They are stored in Amazon S3 and identified by a unique identifier in the form of ami-xxxxxxand a manifest XML file. An AMI contains a physical file system layout with a predefined operating system installed.

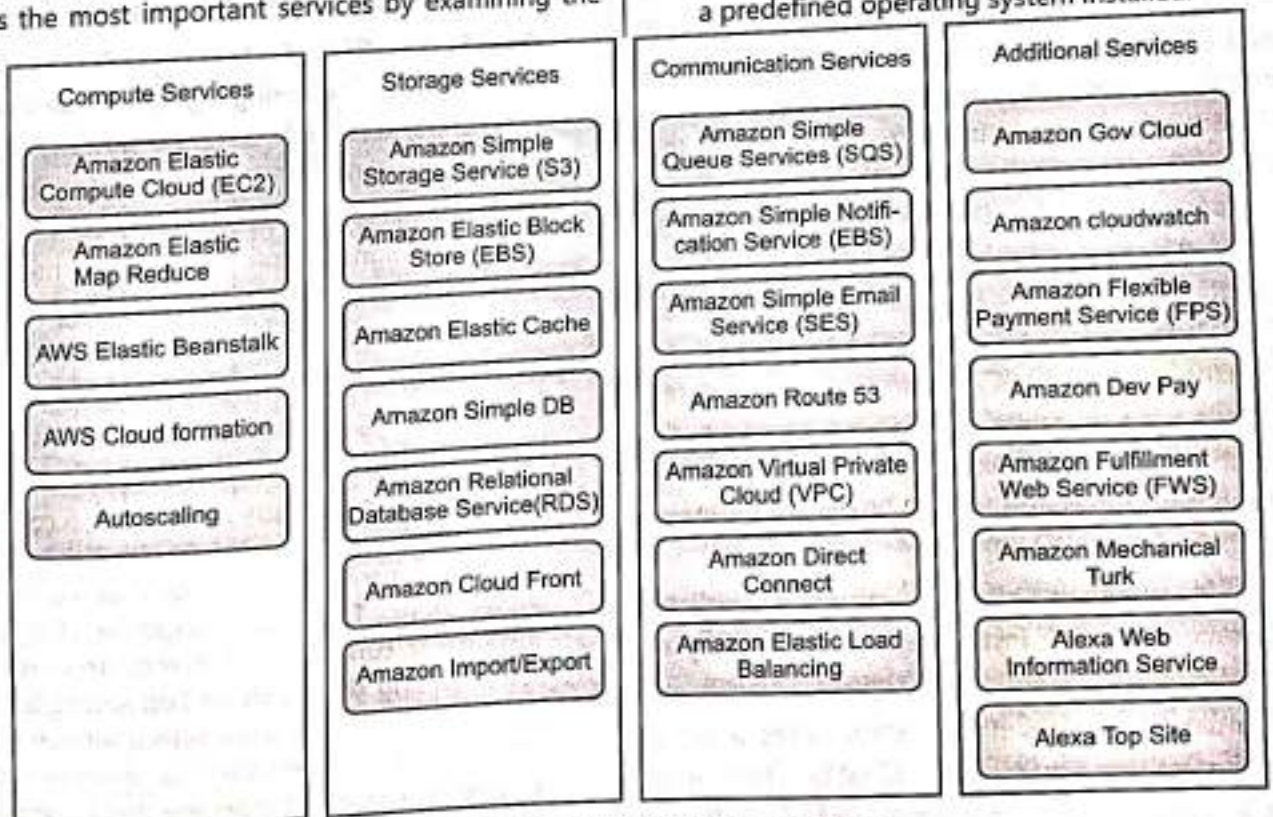


Fig. 5.11 : Amazon Web Services ecosystem

- These are specified by the Amazon Ramdisk Image (ARI, id: ari-yyyyyy) and the Amazon Kernel Image (AKI, id: aki-zzzzzz), which are part of the configuration of the template. AMIs are either created from scratch or "bundled" from existing EC2 instances. A common practice is to prepare new AMIs to create an instance from a preexisting AMI, log into it once it is booted and running, and install all the software needed. Using the tools provided by Amazon, we can convert the instance into a new image. Once an AMI is created, it is stored in an S3 bucket and the user can decide whether to make it available to other users or keep it for personal use. Finally, it is also possible to associate a product code with a given AMI, thus allowing the owner of the AMI to get revenue every time this AMI is used to create EC2 instances.

EC2 Instances

- EC2 instances represent virtual machines. They are created using AMI as templates, which are specialized by selecting the number of cores, their computing power, and the installed memory. The processing power is expressed in terms of virtual cores and EC2 Compute Units (ECUs). The ECU is a measure of the computing power of a virtual core; it is used to express a predictable quantity of real CPU power that is allocated to an instance. By using compute units instead of real frequency values, Amazon can change over time the mapping of such units to the underlying real amount of computing power allocated, thus keeping the performance of EC2 instances consistent with standards set by the times. Over time, the hardware supporting the underlying infrastructure will be replaced by more powerful hardware, and the use of ECUs helps give users a consistent view of the performance offered by EC2 instances. Since users rent computing capacity rather than buying hardware, this approach is reasonable. One ECU is defined as giving the same performance as a 1.0 1.2GHz 2007 Opteron or 2007 Xeon processor.
- Table 5.2 : shows all the currently available configurations for EC2 instances. We can identify six major categories:
  1. **Standard Instances :** This class offers a set of configurations that are suitable for most applications. EC2 provides three different categories of increasing computing power, storage, and memory.
  2. **Micro Instances :** This class is suitable for those applications that consume a limited amount of computing power and memory and occasionally need bursts in CPU cycles to process surges in the workload. Micro instances can be used for small Web applications with limited traffic.
  3. **High-Memory Instances :** This class targets applications that need to process huge workloads and require large amounts of memory. Three-tier Web applications characterized by high traffic are the target profile. Three categories of increasing memory and CPU are available, with memory proportionally larger than computing power.
  4. **High-CPU Instances :** This class targets compute-intensive applications. Two configurations are available where computing power proportionally increases more than memory.
  5. **Cluster Compute Instances :** This class is used to provide virtual cluster services. Instances in this category are characterized by high CPU compute power and large memory and an extremely high I/O and network performance, which makes it suitable for HPC applications.
  6. **Cluster GPU Instances :** This class provides instances featuring graphic processing units (GPUs) and high compute power, large memory, and extremely high I/O and network performance. This class is particularly suited for cluster applications that perform heavy graphic computations, such as rendering clusters. Since GPU can be used for general-purpose computing, users of such instances can benefit from additional computing power, which makes this class suitable for HPC applications.
- EC2 instances are priced hourly according to the category they belong to. At the beginning of every hour of usage, the user will be charged the cost of the entire hour. The hourly expense charged for one instance is constant. Instance owners are responsible for providing their own backup strategies, since there is no guarantee that the instance will run for the entire hour. Another alternative is represented by spot instances. These instances are much more dynamic in terms of pricing and lifetime since they are made

available to the user according to the load of EC2 and the availability of resources. Users define an upper bound for a price they want to pay for these instances; as long as the current price (the spot price) remains under the given bound, the instance is kept running. The price is sampled at the beginning of each hour. Spot instances are more volatile than normal instances; where as for normal instances EC2 will try as much as possible to keep them active, there is no such guarantee for spot instances. Therefore, implementing backup and check pointing strategies is inevitable.

Table 5.2 : Amazon EC2 (On-Demand) Instances Characteristics

Instance Type	ECU	Platform	Memory	Disk Storage	Price (U.S. East) (USD/hour)
Standard instances					
Small	1(1 3 1)	32 bit	1.7 GB	160 GB	$0.085 Linux $0.12 Windows
Large	4(2 3 2)	64 bit	7.5 GB	850 GB	$0.340 Linux $0.48 Windows
Extra Large	8(4 3 2)	64 bit	15 GB	1,690 GB	$0.680 Linux $0.96 Windows
Micro instances					
Micro	,52	32/64 bit	613 MB	EBS Only	$0.020 Linux $0.03 Windows
High-Memory instances					
Extra Large	6.5(233.25)	64 bit	17.1 GB	420 GB	$0.500 Linux $0.62 Windows
Double Extra Large	13(433.25)	64 bit	34.2 GB	850 GB	$1.000 Linux $1.24 Windows
Quadruple Extra Large	26(833.25)	64 bit	68.4 GB	1,690 GB	$2.000 Linux $2.48 Windows
High-CPU Instances					
Medium	5(2 3 2.5)	32 bit	1.7 GB	350 GB	$0.170 Linux $0.29 Windows
Extra Large	20(8 3 2.5)	64 bit	7 GB	1,690 GB	$0.680 Linux $1.16 Windows
Cluster Instances					
Quadruple Extra Large	33.5	64 bit	23 GB	1,690 GB	$1.600 Linux $1.98 Windows
Cluster GPU Instances					
Quadruple Extra Large	33.5	64 bit	22 GB	1,690 GB	$2.100 Linux $2.60 Windows

- EC2 instances can be run either by using the command-line tools provided by Amazon, which connects the Amazon Web Service that provides remote access to the EC2 infrastructure, or via the AWS console, which allows the management of other services, such as S3. By default an EC2 instance is created with the kernel and the disk associated to the AMI. These define the architecture (32 bit or 64 bit) and the space of disk available to the instance. This is an ephemeral disk; once the instance is shut down, the content of the disk will be lost. Alternatively, it is possible to attach an EBS volume to the instance, the content of which will be stored in S3. If the default AKI and ARI are not suitable, EC2 provides capabilities to run EC2 instances by specifying a different AKI and ARI, thus giving flexibility in the creation of instances.

EC2 Environment

- EC2 instances are executed within a virtual environment, which provides them with the services they require to host applications. The EC2 environment is in charge of allocating addresses, attaching storage volumes, and configuring security in terms of access control and net work connectivity.
- By default, instances are created with an internal IP address, which makes them capable of communicating within the EC2 network and accessing the Internet as clients. It is possible to associate an Elastic IP to each instance, which can then be remapped to a different instance over time. Elastic IPs allow instances running in EC2 to act as servers reachable from the Internet and, since they are not strictly bound to specific instances, to implement failover capabilities. Together with an external IP, EC2 instances are also given a domain name that generally is in the form ec2-xxx-xxx-xxx.compute-x.amazonaws.com, where xxx-xxx-xxx normally represents the four parts of the external IP address separated by a dash, and compute-x gives information about the availability zone where instances are deployed. Currently, there are five availability zones that are priced differently: two in the United States (Virginia and Northern California), one in Europe (Ireland), and two in Asia Pacific (Singapore and Tokyo).
- Instance owners can partially control where to deploy instances. Instead, they have a finer control over the security of the instances as well as their network accessibility. Instance owners can associate a key pair

to one or more instances when these instances are created. A key pair allows the owner to remotely connect to the instance once this is running and gain root access to it. Amazon EC2 controls the accessibility of a virtual instance with basic firewall configuration, allowing the specification of source address, port, and protocols (TCP, UDP, ICMP). Rules can also be attached to security groups, and instances can be made part of one or more groups before their deployment. Security groups and firewall rules constitute a flexible way of providing basic security for EC2 instances, which has to be complemented by appropriate security configuration within the instance itself.

Advanced Compute Services

- EC2 instances and AMIs constitute the basic blocks for building an IaaS computing cloud. On top of these, Amazon Web Services provide more sophisticated services that allow the easy packaging and deploying of applications and a computing platform that supports the execution of MapReduce based applications.

AWS CloudFormation

- AWS CloudFormation constitutes an extension of the simple deployment model that characterizes EC2 instances. CloudFormation introduces the concepts of templates, which are JSON formatted text files that describe the resources needed to run an application or a service in EC2 together with the relations between them. CloudFormation allows easily and explicitly linking EC2 instances together and introducing dependencies among them. Templates provide a simple and declarative way to build complex systems and integrate EC2 instances with other AWS services such as S3, Simple DB, SQS, SNS, Route53, Elastic Beanstalk, and others.

AWS Elastic Beanstalk

- AWS Elastic Beanstalk constitutes a simple and easy way to package applications and deploy them on the AWS Cloud. This service simplifies the process of provisioning instances and deploying application code and provides appropriate access to them. Currently, this service is available only for Web applications developed with the Java/Tomcat technology stack. Developers can conveniently package their Web application into a WAR file and use Beanstalk to automate its deployment on the AWS Cloud.
- With respect to other solutions that automate cloud deployment, Beanstalk simplifies tedious tasks without removing the user's capability of accessing and taking over control of the underlying EC2 instances that make up the virtual infrastructure on top of which the application is running. With respect to AWS CloudFormation, AWS Elastic Beanstalk provides a higher-level approach for application deployment on the cloud, which does not require the user to specify the infrastructure in terms of EC2 instances and their dependencies.

Amazon Elastic MapReduce

- Amazon Elastic MapReduce provides AWS users with a cloud computing platform for MapReduce applications. It utilizes Hadoop as the MapReduce engine, deployed on a virtual infrastructure composed of EC2 instances, and uses Amazon S3 for storage needs.
- Apart from supporting all the application stack connected to Hadoop (Pig, Hive, etc.), Elastic MapReduce introduces elasticity and allows users to dynamically size the Hadoop cluster according to their needs, as well as select the appropriate configuration of EC2 instances to compose the cluster (Small, High-Memory, High-CPU, Cluster Compute, and Cluster GPU). On top of these ser- vices, basic Web applications allowing users to quickly run data-intensive applications without writing code are offered.

5.4.3 Storage Services

- AWS provides a collection of services for data storage and information management. The core ser- vice in this area is represented by Amazon Simple Storage Service (S3). This is a distributed object store that allows users to store information in different formats. The core components of S3 are two: buckets and objects. Buckets represent virtual containers in which to store objects; objects rep- resent the content that is actually stored. Objects can also be enriched with metadata that can be used to tag the stored content with additional information.

S3 Key Concepts

- As the name suggests, S3 has been designed to provide a simple storage service that's accessible through a Representational State Transfer (REST) interface, which is quite similar to a distributed file

system but which presents some important differences that allow the infrastructure to be highly efficient:

- The storage is organized in a two-level hierarchy. S3 organizes its storage space into buckets that cannot be further partitioned. This means that it is not possible to create directories or other kinds of physical groupings for objects stored in a bucket. Despite this fact, there are few limitations in naming objects, and this allows users to simulate directories and create logical groupings.
- Stored objects cannot be manipulated like standard files. S3 has been designed to essentially provide storage for objects that will not change over time. Therefore, it does not allow renaming, modifying, or relocating an object. Once an object has been added to a bucket, its content and position is immutable, and the only way to change it is to remove the object from the store and add it again.
- Content is not immediately available to users. The main design goal of S3 is to provide an eventually consistent data store. As a result, because it is a large distributed storage facility, changes are not immediately reflected. For instance, S3 uses replication to provide redundancy and efficiently serve objects across the globe; this practice introduces latencies when adding objects to the store especially large ones which are not available instantly across the entire globe.
- Requests will occasionally fail. Due to the large distributed infrastructure being managed, requests for object may occasionally fail. Under certain conditions, S3 can decide to drop a request by returning an internal server error. Therefore, it is expected to have a small failure rate during day-to-day operations, which is generally not identified as a persistent failure.

- Access to S3 is provided with RESTful Web services. These express all the operations that can be performed on the storage in the form of HTTP requests (GET, PUT, DELETE, HEAD, and POST), which operate differently according to the element they address. As a rule of thumb PUT/ POST requests add new content to the store, GET/HEAD requests are used to retrieve content and information, and DELETE requests are used to remove elements or information attached to them.

Resource Naming

- Buckets, objects, and attached metadata are made accessible through a REST interface. Therefore, they are represented by uniform resource identifiers (URIs) under the s3.amazonaws.comdomain. All the operations are then performed by expressing the entity they are directed to in the form of a request for a URI.

Amazon Offers Three Different Ways of Addressing a Bucket:

1. **Canonical Form:**

   http://s3.amazonaws.com/bukect_name/. The bucket name is expressed as a path component of the domain name s3.amazonaws.com. This is the naming convention that has less restriction in terms of allowed characters, since all the characters that are allowed for a path component can be used.

2. **Subdomain Form:**

   http://bucketname.s3.amazon.com/. Alternatively, it is also possible to reference a bucket as a subdomain of s3.amazonaws.com. To express a bucket name in this form, the name has to do all of the following:

   - Be between 3 and 63 characters long
   - Contain only letters, numbers, periods, and dashes
   - Start with a letter or a number
   - Contain at least one letter
   - Have no fragments between periods that start with a dash or end with a dash or that are empty strings

   This form is equivalent to the previous one when it can be used, but it is the one to be preferred since it works more effectively for all the geographical locations serving resources stored inS3.

3. **Virtual Hosting Form:**

   http://bucket-name.com/.Amazon also allows referencing of its resources with custom URLs. This is accomplished by entering a CNAME record into the DNS that points to the subdomain form of the bucket URI.

- Since S3 is logically organized as a flat data store, all the buckets are managed under the s3.amazonaws.comdomain. Therefore, the names of buckets must be unique across all the users.
- Objects are always referred as resources local to a given bucket. Therefore, they always appear as a part of the resource component of a URI. Since a bucket

can be expressed in three different ways, objects indirectly inherit this flexibility:

- ➢ **Canonical Form:** http://s3.amazonaws.com/bukect_name/object_name
- ➢ **Subdomain Form:** http://bucketname/s3.amzonaws.com/object_name
- ➢ **Virtual Hosting Form:** http://bucket-name.com/object_name

- Except for the ?, which separates the resource path of a URI from the set of parameters passed with the request, all the characters that follow the / after the bucket reference constitute the name of the object. For instance, path separator characters expressed as part of the object name do not have corresponding physical layout within the bucket store. Despite this fact, they can still be used to create logical groupings that look like directories.
- Finally, specific information about a given object, such as its access control policy or the server logging settings defined for a bucket, can be referenced using a specific parameter. More precisely:
  - ➢ Object ACL: http://s3.amazonaws.com/bukect_name/object_name?acl
  - ➢ Bucket server logging: http://s3.amzonaws.com/bucket_name?logging
- Object metadata are not directly accessible through a specific URI, but they are manipulated by adding attributes in the request of the URL and are not part of the identifier.

Buckets

- A bucket is a container of objects. It can be thought of as a virtual drive hosted on the S3 distributed storage, which provides users with a flat store to which they can add objects. Buckets are top- level elements of the S3 storage architecture and do not support nesting. That is, it is not possible to create "subbuckets" or other kinds of physical divisions.
- A bucket is located in a specific geographic location and eventually replicated for fault tolerance and better content distribution. Users can select the location at which to create buckets, which by default are created in Amazon's U.S. datacenters. Once a bucket is created, all the objects that belong to the bucket will be stored in the same availability zone of the bucket. Users create a bucket by sending a PUT request to http://s3.amazonaws.com/ with the name of the bucket and, if they want to specify the availability zone, additional information about the preferred location. The content of a bucket can be listed by sending a GET request specifying the name of the bucket. Once created, the bucket cannot be renamed or relocated. If it is necessary to do so, the bucket needs to be deleted and recreated. The deletion of a bucket is performed by a DELETE request, which can be successful if and only if the bucket is empty.

Objects and Metadata

- Objects constitute the content elements stored in S3. Users either store files or push to the S3 text stream representing the object's content. An object is identified by a name that needs to be unique within the bucket in which the content is stored. The name cannot be longer than 1,024 bytes when encoded in UTF-8, and it allows almost any character. Since buckets do not support nesting, even characters normally used as path separators are allowed. This actually compensates for the lack of a structured file system, since directories can be emulated by properly naming objects.
- Users create an object via a PUT request that specifies the name of the object together with the bucket name, its contents, and additional properties. The maximum size of an object is 5 GB. Once an object is created, it cannot be modified, renamed, or moved into another bucket. It is possible to retrieve an object via a GET request; deleting an object is per formed via a DELETE request.
- Objects can be tagged with metadata, which are passed as properties of the PUT request. Such properties are retrieved either with a GET request or with a HEAD request, which only returns the object's metadata without the content. Metadata are both system and user defined: the first ones are used by S3 to control the interaction with the object, whereas the second ones are meaningful to the user, who can store up to 2 KB per metadata property represented by a key-value pair of strings.

Access Control and Security

- Amazon S3 allows controlling the access to buckets and objects by means of Access Control Policies (ACPs). An ACP is a set of grant permissions that are attached to a resource expressed by means of an XML configuration file. A policy allows defining up to 100 access rules, each of them granting one of the available permissions to a grantee. Currently, five different permissions can be used:
  1. READ allows the grantee to retrieve an object and its metadata and to list the content of a bucket as well as getting its metadata.
  2. WRITE allows the grantee to add an object to a bucket as well as modify and remove it.
  3. READ_ACP allows the grantee to read the ACP of are source.
  4. WRITE_ACP allows the grantee to modify the ACP of are source.
  5. FULL_CONTROL grants all of the preceding permissions.
- Grantees can be either single users or groups. Users can be identified by their canonical IDs or the email addresses they provided when they signed up for S3. For groups, only three options are available: all users, authenticated users, and log delivery user.
- Once a resource is created, S3 attaches a default ACP granting full control permissions to its owner only. Changes to the ACP can be made by using the request to the resource URI followed by ? A GET method allows retrieval of the ACP; a PUT method allows uploading of a new ACP to replace the existing one. Alternatively, it is possible to use a predefined set of permissions called canned policies to set the ACP at the time a resource is created. These policies represent the most common access patterns for S3resources.
- ACPs provide a set of powerful rules to control S3 users' access to resources, but they do not exhibit fine grain in the case of non authenticated users, who cannot be differentiated and are considered as a group. To provide a finer grain in this scenario, S3 allows defining signed URIs, which grant access to a resource for a limited amount of time to all the requests that can provide a temporary access token.

Advanced Features

- Besides the management of buckets, objects, and ACPs, S3 offers other additional features that can be helpful. These features are server access logging and integration with the BitTorrentfile-sharing network.
- Server access logging allows bucket owners to obtain detailed information about the request made for the bucket and all the objects it contains. By default, this feature is turned off; it can be activated by issuing a PUT request to the bucket URI followed by ?logging. The request should include an XML file specifying the target bucket in which to save the logging files and the file name prefix. A GET request to the same URI allows the user to retrieve the existing logging configuration for the bucket.
- The second feature of interest is represented by the capability of exposing S3 objects to the Bit Torrent network, thus allowing files stored in S3 to be downloaded using the Bit Torrent protocol. This is done by appending? torrent to the URI of the S3 object. To actually download the object, its ACP must grant read permission to everyone.

Amazon Elastic Blockstore

- The Amazon Elastic Block Store (EBS) allows AWS users to provide EC2 instances with persistent storage in the form of volumes that can be mounted at instance startup. They accommodate up to 1 TB of space and are accessed through a block device interface, thus allowing users to format them according to the needs of the instance they are connected to (raw storage, file system, or other). The content of an EBS volume survives the instance life cycle and is persisted into S3. EBS volumes can be cloned, used as boot partitions, and constitute durable storage since they rely on S3 and it is possible to take incremental snap shots of their content.
- EBS volumes normally reside within the same availability zone of the EC2 instances that will use them to maximize the I/O performance. It is also possible to connect volumes located in different availability zones. Once mounted as volumes, their content is lazily loaded in the background and according to the request made by the operating system. This reduces the number of I/O requests that go to the network. Volume images cannot be shared among instances,

but multiple (separate) active volumes can be created from them. In addition, it is possible to attach multiple volumes to a single instance or create a volume from a given snapshot and modify its size, if the formatted file system allows such an operation.

- The expense related to a volume comprises the cost generated by the amount of storage occupied in S3 and by the number of I/O requests performed against the volume. Currently, Amazon charges $0.10/GB/month of allocated storage and $0.10 per 1 million requests made to the volume.

Amazon ElastiCache

- ElastiCache is an implementation of an elastic in-memory cache based on a cluster of EC2 instances. It provides fast data access from other EC2 instances through a Memcached-compatible protocol so that existing applications based on such technology do not need to be modified and can transparently migrate to ElastiCache.
- ElastiCache is based on a cluster of EC2 instances running the caching software, which is made available through Web services. An ElastiCache cluster can be dynamically resized according to the demand of the client applications. Furthermore, automatic patch management and failure detection and recovery of cache nodes allow the cache cluster to keep running without administrative intervention from AWS users, who have only to elastically size the cluster when needed.
- ElastiCache nodes are priced according to the EC2 costing model, with a small price difference due to the use of the caching service installed on such instances. It is possible to choose between different types of instances; Table 5.3 provides an overview of the pricing options.
- The prices indicated in Table 5.3 are related to the Amazon offerings during 2011 2012, and the amount of memory specified represents the memory available after taking system software overhead into account.

Structured Storage Solutions

- Enterprise applications quite often rely on databases to store data in a structured form, index, and perform analytics against it. Traditionally, RDBMS have been the common data back-end for a wide range of applications, even though recently more scalable and lightweight solutions have been proposed. Amazon provides applications with structured storage services in three different forms: pre configured EC2 AMIs, Amazon Relational Data Storage (RDS), and Amazon Simple DB.

Preconfigured EC2 AMIs

- Preconfigured EC2 AMIs are predefined templates featuring an installation of a given database management system. EC2 instances created from these AMIs can be completed with an EBS volume for storage persistence. Available AMIs include installations of IBM DB2, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, Sybase, and Vertica. Instances are priced hourly according to the EC2 cost model. This solution poses most of the administrative burden on the EC2 user, who has to configure, maintain, and manage the relational database, but offers the greatest variety of products to choose from.

Amazon RDS

- RDS is relational database service that relies on the EC2 infrastructure and is managed by Amazon. Developers do not have to worry about configuring the storage for high availability, designing failover strategies, or keeping the servers up-to-date with patches.

Table 5.3 : Amazon EC2 (On-Demand) Cache Instances Characteristics, 2011—2012

Instance Type	ECU	Plat form	Memory	I/O Capacity	Price (U.S. East) (USD/hour)
Standard instances					
Small	1(1 3 1)	64 bit	1.3 GB	Moderate	$0.095
Large	4(2 3 2)	64 bit	7.1 GB	High	$0.380
Extra Large	8(4 3 2)	64 bit	14.6 GB	High	$0.760
High-Memory instances					
Extra Large	6.5(2 3 3.25)	64 bit	16.7 GB	High	$0.560
Double Extra Large	13(4 3 3.25)	64 bit	33.8 GB	High	$1.120
Quadruple Extra Large	26(8 3 3.25)	64 bit	68 GB	High	$2.240
High-CPU instances					
Extra Large	26(8 3 3.25)	64 bit	6.6 GB	High	$0.760

- Moreover, the service provides users with automatic backups, snapshots, point-in-time recoveries, and facilities for implementing replications. These and the common database management services are available through the AWS console or a specific Web service. Two relational engines are available: MySQL and Oracle.
- Two key advanced features of RDS are multi-AZ deployment and read replicas. The first option provides users with a failover infrastructure for their RDBMS solutions. The high-availability solution is implemented by keeping in standby synchronized copies of the services in different availability zones that are activated if the primary service goes down. The second option provides users with increased performance for applications that are heavily based on database reads. In this case, Amazon deploys copies of the primary service that are only available for database reads, thus cut- ting down the response time of the service.
- The available options and the relative pricing of the service during 2011 2012 are shown in Table 5.4. The table shows the costing details of the on-demand instances. There is also the possibility of using reserved instances for long terms (one to three years) by paying up-front at discounted hourly rates.
- With respect to the previous solution, users are not responsible for managing, configuring, and patching the database management software, but these operations are performed by the AWS. In addition, support for elastic management of servers is simplified. Therefore, this solution is optimal for applications based on the Oracle and MySQL engines, which are migrated on the AWS infrastructure and require a scalable database solution.

Amazon SimpleDB

- Amazon SimpleDB is a lightweight, highly scalable, and flexible data storage solution for applications that do not require a fully relational model for their data. SimpleDB provides support for semi structured data, the model for which is based on the concept of domains, items, and attributes. With respect to the relational model, this model provides fewer constraints on the structure of data entries, thus obtaining improved performance in querying large quantities of data. As happens for Amazon RDS, this service frees AWS users from performing configuration, management, and high-availability design for their data stores.

Table 5.4 : Amazon RDS (On-Demand) Instances Characteristics, 2011—2012

Instance Type	ECU	Plat form	Memory	I/O Capacity	Price (U.S. East) (USD/hour)
Standard instances					
Small	1(1 3 1)	64 bit	1.7 GB	Moderate	$0.11
Large	4(2 3 2)	64 bit	7.5 GB	High	$0.44
Extra Large	8(4 3 2)	64 bit	15 GB	High	$0.88
High-Memory instances					
Extra Large	6.5(2 3 3.25)	64 bit	17.1 GB	High	$0.65
Double Extra Large	13(4 3 3.25)	64 bit	34 GB	High	$1.30
Quadruple Extra Large	26(8 3 3.25)	64 bit	68 GB	High	$2.60

- Simple DB uses domains a stop-level elements to organize a data store. These domains are roughly comparable to tables in the relational model. Unlike tables, they allow items not to have all the same column structure; each item is therefore represented a collection of attributes expressed in the form of a key value pair. Each domain can grow up to 10GB of data, and by default a single user can allocate a maximum of 250 domains. Clients can create, delete, modify, and make snapshots of domains. They can insert, modify, delete, and query items and attributes. Batch insertion and deletion are also supported. The capability of querying data is one of the most relevant functions of the model, and the select clause supports the following test operators: 5,!5,,,,, ,5, .5, like, not like, between, is null, is not null, and every(). Here is a simple example on how to query data: select x from domain_name where every (attribute_name) 5 'value'
- Moreover, the select operator can extend its query beyond the boundaries of a single domain, thus allowing users to query effectively a large amount of data.

- To efficiently provide AWS users with a scalable and faulttolerant service, Simple DB implements a relaxed constraint model, which leads to eventually consistent data. The adverb eventually denotes the fact that multiple accesses on the same data might not read the same value in the very short term, but they will eventually converge over time. This is because SimpleDB does not lock all the copies of the data during an update, which is propagated in the background. Therefore, there is a transient period of time in which different clients can access different copies of the same data that have different values. This approach is very scalable with minor drawbacks, and it is also reasonable, since the application scenario for SimpleDB is mostly characterized by querying and indexing operations on data. Alternatively, it is possible to change the default behavior and ensure that all there are blocked during an update.
- Even though SimpleDB is not a transactional model, it allows clients to express conditional insertions or deletions, which are useful to prevent lost updates in multiple-writer scenarios. In this case, the operation is executed if and only if the condition is verified. This condition can be used to check preexisting values of attributes for an item.
- Table 5.5 provides an overview of the pricing options for the SimpleDB service for data transfer during 2011 2012. The service charges either for data transfer or stored data. Data transfer within the AWS network is not charged. In addition, SimpleDB also charges users for machine usage. The first 25 SimpleDB instances per month are free; after this threshold there is an hourly charge ($0.140 hour in the U.S. Eastregion).
- If we compare this cost model with the one characterizing S3, it becomes evident that S3 is a cheaper option for storing large objects. This is useful information for clarifying the different nature of SimpleDB with respect to S3: The former has been designed to provide fast access to semi structured collections of small objects and not for being a long-term storage option for large objects.

Amazon CloudFront

- CloudFront is an implementation of a content delivery network on top of the Amazon distributed storage infrastructure. It leverages a collection of edge servers strategically located around the globe to better serve requests for static and streaming Web content so that the transfer time is reduced as much as possible.

Table 5.5 : Amazon SimpleDB Data Transfer Charges, 2011—2012

Instance Type	Price (U.S. East) (USD)
Data Transfer In	
All datatransferin	$0.000
Data Transfer Out	
1stGB/month	$0.000
Up to10TB/month	$0.120
Next40TB/month	$0.090
Next100TB/month	$0.070
Next350TB/month	$0.050
Next524TB/month	Special arrangements
Next4PB/month	Special arrangements
Greater than5PB/month	Special arrangements

- AWS provides users with simple Web service APIs to manage CloudFront. To make available content through CloudFront, it is necessary to create a distribution. This identifies an origin server, which contains the original version of the content being distributed, and it is referenced by a DNS domain under the Cloudfront.net domain name (i.e., my-distribution.Cloudfront.net). It is also possible to map a given domain name to a distribution. Once the distribution is created, it is sufficient to reference the distribution name, and the CloudFront engine will redirect the request to the closest replica and eventually download the original version from the origin server if the content is not found or expired on the selected edge server.
- The content that can be delivered through CloudFront is static (HTTP and HTTPS) or streaming (Real Time Messaging Protocol, or RMTP). The origin server hosting the original copy of the distributed content can be an S3 bucket, an EC2 instance, or a server external to the Amazon network. Users can restrict access to the distribution to only one or a few of the available protocols, or they can set up access rules for finer control. It is also possible to invalidate content to remove it from the distribution or force its update before expiration.

- Table 5.6 provides a breakdown of the pricing during 2011 2012. Note that CloudFrontis cheaper than S3. This reflects its different purpose: CloudFront is designed to optimize the distribution of very popular content that is frequently downloaded, potentially from the entire globe and not only the Amazon network.

5.4.4 Communication Services

- Amazon provides facilities to structure and facilitate the communication among existing applications and services residing within the AWS infrastructure. These facilities can be organized into two major categories: virtual networking and messaging.

1. Virtual Networking

- Virtual networking comprises a collection of services that allow AWS users to control the connectivity to and between compute and storage services. Amazon Virtual Private Cloud (VPC) and

Table 5.6 : Amazon CloudFront On-Demand Pricing, 2011—2012

Pricing Item	United States	Europe	Hong Kong and Singapore	Japan	South America
Requests					
Per 10,000 HTTP requests	$0.0075	$0.0090	$0.0090	$0.0095	$0.0160
Per 10,000 HTTPS requests	$0.0100	$0.0120	$0.0120	$0.0130	$0.0220
Regional Data Transfer Out	$0.120/GB	$0.120/GB	$0.190/GB	$0.201/GB	$0.250/GB
First 10 TB/month	$0.080/GB	$0.080/GB	$0.140/GB	$0.148/GB	$0.200/GB
Next 40 TB/month	$0.060/GB	$0.060/GB	$0.120/GB	$0.127/GB	$0.180/GB
Next 100 TB/month	$0.040/GB	$0.040/GB	$0.100/GB	$0.106/GB	$0.160/GB
Next 350 TB/month	$0.030/GB	$0.030/GB	$0.080/GB	$0.085/GB	$0.140/GB
Next 524 TB/month	$0.025/GB	$0.025/GB	$0.070/GB	$0.075/GB	$0.130/GB
Next 4 PB/month (greaterthan5 PB/month	$0.020/GB	$0.020/GB	$0.060/GB	$0.065/GB	$0.125/GB

- Amazon Direct Connect provide connectivity solutions in terms of infrastructure; Route 53 facilitates connectivity in terms of naming.
- Amazon VPC provides a great degree of flexibility in creating virtual private networks within the Amazon infrastructure and beyond. The service providers prepare either templates covering most of the usual scenarios or a fully customizable network service for advanced configurations. Prepared templates include public subnets, isolated networks, private networks accessing Internet through network address translation (NAT), and hybrid networks including AWS resources and private resources. Also, it is possible to control connectivity between different services (EC2 instances and S3 buckets) by using the Identity Access Management (IAM) service. During 2011, the cost of Amazon VPC was $0.50 per connection hour.
- Amazon Direct Connect allows AWS users to create dedicated networks between the user private network and Amazon Direct Connect locations, called ports. This connection can be further partitioned in multiple logical connections and give access to the public resources hosted on the Amazon infrastructure. The advantage of using Direct Connect versus other solutions is the consistent performance of the connection between the users' premises and the Direct Connect locations. This service is compatible with other services such as EC2, S3, and Amazon VPC and can be used in scenarios requiring high bandwidth between the Amazon network and the outside world. There are only two available ports located in the United States, but users can leverage external providers that offer guaranteed high bandwidth to these ports. Two different bandwidths can be chosen: 1 Gbps, priced at $0.30 per hour, and 10 Gbps, priced at $2.25 per hour. Inbound traffic is free; outbound traffic is priced at $0.02 per GB.
- Amazon Route 53 implements dynamic DNS services that allow AWS resources to be reached through domain names different from the amazon.com domain. By leveraging the large and globally distributed network of Amazon DNS servers, AWS users can expose EC2 instances or S3 buckets as resources unde

a domain of their property, for which Amazon DNS servers become authoritative. EC2 instances are likely to be more dynamic than the physical machines, and S3 buckets might also exist for a limited time. To cope with such a volatile nature, the service provides AWS users with the capability of dynamically mapping names to resources as instances are launched on EC2 or as new buckets are created in S3. By interacting with the Route 53 Web ser- vice, users can manage a set of hosted zones, which represent the user domains controlled by the service, and edit the resources made available through it. Currently, a single user can have up to 100 zones. The costing model includes a fixed amount ($1 per zone per month) and a dynamic component that depends on the number of queries resolved by the service for the hosted zones ($0.50 per million queries for the first billion of queries a month, $0.25 per million queries over 1 billion of queries a month).

2. Messaging

- Messaging services constitute the next step in connecting applications by leveraging AWS capabilities. The three different types of messaging services offered are Amazon Simple Queue Service (SQS), Amazon Simple Notification Service (SNS), and Amazon Simple Email Service (SES).
- Amazon SQS constitutes disconnected model for exchanging messages between applications by means of message queues, hosted within the AWS infrastructure. Using the AWS console or directly the underlying Web service AWS, users can create an unlimited number of message queues and configure them to control their access. Applications can send messages to any queue they have access to. These messages are securely and redundantly stored within the AWS infrastructure for a limited period of time, and they can be accessed by other (authorized) applications. While a message is being read, it is kept locked to avoid spurious processing from other applications. Such a lock will expire after a given period.
- Amazon SNS provides a publish-subscribe method for connecting heterogeneous applications. With respect to Amazon SQS, where it is necessary to continuously poll a given queue for a new message to process, Amazon SNS allows applications to be notified when new content of interest is available. This feature is accessible through a Web service whereby AWS users can create a topic, which other applications can subscribe to. At any time, applications can publish content on a given topic and subscribers can be automatically notified. The service provides subscribers with different notification models (HTTP/HTTPS, email/email JSON, and SQS).
- Amazon SES provides AWS users with a scalable email service that leverages the AWS infra- structure. Once users are signed up for the service, they have to provide an email that SES will use to send emails on their behalf. To activate the service, SES will send an email to verify the given address and provide the users with the necessary information for the activation. Upon verification, the user is given an SES sandbox to test the service, and he can request access to the production version. Using SES, it is possible to send either SMTP-compliant emails or raw emails by specifying email headers and Multipurpose Internet Mail Extension (MIME) types. Emails are queued for delivery, and the users are notified of any failed delivery. SES also provides a wide range of statistics that help users to improve their email campaigns for effective communication with customers.
- With regard to the costing, all three services do not require a minimum commitment but are based on a pay-as-you go model. Currently, users are not charged until they reach a minimum threshold. In addition, data transfer in is not charged, but data transfer out is charged by ranges.

5.4.5 Additional Services

- Besides compute, storage, and communication services, AWS provides a collection of services that allow users to utilize services in aggregation. The two relevant services are Amazon Cloud Watch and Amazon Flexible Payment Service (FPS).

- Amazon Cloud Watch is a service that provides a comprehensive set of statistics that help developers understand and optimize the behavior of their application hosted on AWS. Cloud Watch collects information from several other AWS services: EC2, S3, SimpleDB, CloudFront, and others. Using Cloud Watch, developers can see a detailed breakdown of their usage of the service they are renting on AWS and can devise more efficient and cost-saving applications. Earlier services of Cloud Watch were offered only through subscription, but now it is made available for free to all the AWS users.
- Amazon FPS infrastructure allows AWS users to leverage Amazon's billing infrastructure to sell goods and services to other AWS users. Using Amazon FPS, developers do not have to set up alternative payment methods, and they can charge users via a billing service. The payment models available through FPS include one-time payments and delayed and periodic payments, required by subscriptions and usage based services, transactions, and aggregate multiple payments.

Summary

- Amazon provides a complete set of services for developing, deploying, and managing cloud computing systems by leveraging the large and distributed AWS infrastructure. Developers can use EC2 to control and configure the computing infrastructure hosted in the cloud. They can leverage other services, such as AWS CloudFormation, Elastic Beanstalk, or Elastic MapReduce, if they do not need complete control over the computing stack. Applications hosted in the AWS Cloud can leverage S3, SimpleDB, or other storage services to manage structured and unstructured data. These services are primarily meant for storage, but other options, such as Amazon SQS, SNS, and SES, provide solutions for dynamically connecting applications from both inside and outside the AWS Cloud. Network connectivity to AWS applications is addressed by Amazon VPC and Amazon Direct Connect.

5.5 GOOGLE APP ENGINE

- Google App Engine is a PaaS implementation that provides services for developing and hosting scalable Web applications. AppEngine is essentially a distributed and scalable runtime environment that leverages Google's distributed infrastructure to scale out applications facing a large number of requests by allocating more computing resources to them and balancing the load among them. The runtime is completed by a collection of services that allow developers to design and implement applications that naturally scale on AppEngine. 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.

5.5.1 Architecture and Core Concepts

- AppEngine is a platform for developing scalable applications accessible through the Web (Fig. 5.12). The platform is logically divided into four major components: infrastructure, the run- time environment, the underlying storage, and the set of scalable services that can be used to develop applications.

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.

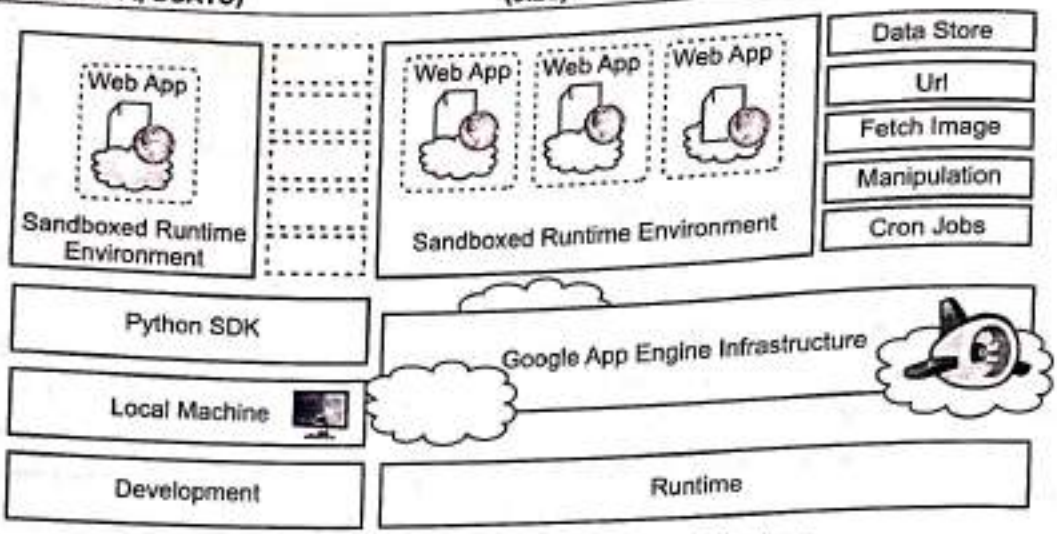


Fig. 5.12 : Google AppEngine platform architecture.

Runtime Environment

- The runtime environment represents the execution context 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.

Sand Boxing

- 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 job; and processing are quest for more than 30 seconds.

Supported Runtimes

- Currently, it is possible to develop AppEngine applications using three different languages and related technologies: Java, Python, and Go.
- AppEngine currently supports Java 6, and developers can use the common tools for Web application development in Java, such as the Java Server Pages (JSP), and the applications interact with the environment by using the Java Servlet standard. Furthermore, 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 restriction simposed by these and box.
- 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.
- 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: in memory-cache, storage for semi structured data, and long-term storage for static data. In this section, we describe Data Store and the use of static file servers. We cover Mem Cache 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 con- tent, 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 in which to store objects that can be retrieved by a specified key. Both the type of the key and the structure of the object canvary.
- With respect to the traditional Web applications backed by a relational database, DataStore imposes less constraint on the regularity of the data but, at the same time, does not implement some of the features of the relational model (such as reference constraints and join operations). These design decisions originated from a careful analysis of data usage patterns for Web applications and were taken in order to obtain a more scalable and efficient datastore. The underlying infrastructure of DataStore is based on Big table a redundant, distributed, and semi structured datastore that organizes data in the form of tables.
- DataStore provides high-level abstractions that simplify interaction with Big table. Developers define their data in terms of entity and properties, and these are persisted and maintained by the service into tables in Big table. An entity constitutes the level of granularity for the storage, and it identifies a collection of properties that define the data it stores. Properties are defined according to one of the several primitive types supported by the service. Each entity is associated with a key, which is either provided by the user or created automatically by AppEngine. An entity is associated with a named kind that AppEngine uses to optimize its retrieval from Big table. Although entities and properties seem to be similar to rows and tables in SQL, there are a few differences that have to be taken into account. Entities of the same kind might not have the same properties, and proper- ties of the same name might contain values of different types. Moreover, properties can store different versions of the same values. Finally, keys are immutable elements and, once created, they cannot be changed.
- DataStore also provides facilities for creating indexes on data and to update data within the con- text of a transaction. Indexes are used to support and speed up queries. A query can return zero or more objects of the same kind or simply the corresponding keys. It is possible to query the data store by specifying either the key or conditions on the values of the properties. 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. 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 datastore 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 pre computed indexes makes the query execution time-independent from the size of the stored data but only influenced by the size of the result set.

- 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 in to exceptions.

Application Services

- Applications hosted on AppEngine take the most from the services made available through the runtime environment. These services simplify most of the common operations that are performed in Web applications: access to data, account management, integration of external resources, messaging and communication, image manipulation, and 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 rendered from an XML document retrieved from a Web service, or they can be rendered 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. Moreover, the ability to perform such requests asynchronously allows the applications to continue with their logic while the resource is retrieved in the background. UrlFetch is not only used to integrate meshes into a Web page but also to leverage remote Web services in accordance with the SOA reference model for distributed applications.

MemCache

- AppEngine provides developers with access to fast and reliable storage, which is DataStore. Despite this, 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; 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. To facilitate the implementation of such tasks, 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 he error.

- AppEngine provides also another way to communicate with the external world: 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. Even though 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. AppEngine 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. In this case, the applications can be easily integrated with all the other services (and profile settings) included in Google Apps.

Image Manipulation

- Web applications render 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: 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 that are off-bandwidth or those that cannot be per- formed within the time frame of the Web request.

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 service. 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.

5.5.2 Application Lifecycle

- 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. As discussed earlier, AppEngine builds indexes for each of the queries performed by a given application in order to speed up access to the relevant data. This capability is enabled by a priori knowledge about all the possible queries made by the application; such knowledge is made available to AppEngine by the developer while uploading the application. 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. Moreover, 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 AppEngine with Python 2.5. It provides a standalone tool, called Google AppEngine Launcher, for managing Web applications locally and deploying them to AppEngine. 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. In addition, 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 pre- view 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 frame- work 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 webapp framework has been re implemented and made available in the Python SDK so that it can be used with AppEngine. Another Web framework that is known to work well is Django.
- 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 deploy done 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://,applicationid..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. Besides being unique, it also needs to be compliant to the rules that are enforced for domain names. 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 overtime.
- 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 AppEngine Launcher and Google AppEngine 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.

5.6 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). These quotas are generally bigger than billable quotas and are set by AppEngine to avoid applications impacting each other's performance or overloading the infrastructure. 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 depleted and will not be available to the application until the quota is replenished. Once a resource is depleted, 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; all the other resources and services will generate an exception that can be trapped in code to provide more useful feedback to users.
- Resources and services quotas are organized into free default quotas and 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.

Observations

- AppEngine, a framework for developing scalable Web applications, leverages Google's infrastructure. The core components of the service are a scalable and sandboxed runtime environment for executing applications and a collection of services that implement most of the common features required for Web development and that help developers build applications that are easy to scale. One of the characteristic elements of AppEngine is the use of simple interfaces that allow applications to perform specific operations that are optimized and designed to scale. Building on top of these blocks, developers can build applications and let AppEngine scale the mout when needed.
- With respect to the traditional approach to Web development, the implementation of rich and powerful applications requires a change of perspective and more effort. Developers have to become familiar with the capabilities of AppEngine and implement the required features in a way that conforms with the AppEngine application model.

5.7 MICROSOFT AZURE

- Microsoft Windows Azure is a cloud operating system built on top of Microsoft datacenters' infra- structure and provides developers with a collection of services for building applications with cloud technology. Services range from compute, storage, and networking to application connectivity, access control, and business intelligence. Any application that is built on the Microsoft technology can be scaled using the Azure platform, which integrates the scalability features into the common Microsoft technologies such as Microsoft Windows Server 2008, SQL Server, and ASP.NET.
- Fig 5.13 provides an overview of services provided by Azure. 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.

5.7.1 Azure Core Concepts

- 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. This scalable computing environment is hosted within Microsoft datacenters and accessible through the Windows Azure Management Portal. Alternatively, developers can recreate a Windows Azure environment (with limited capabilities) on their own machines for development and testing purposes. In this section, we provide an overview of the Azure middleware and its services.

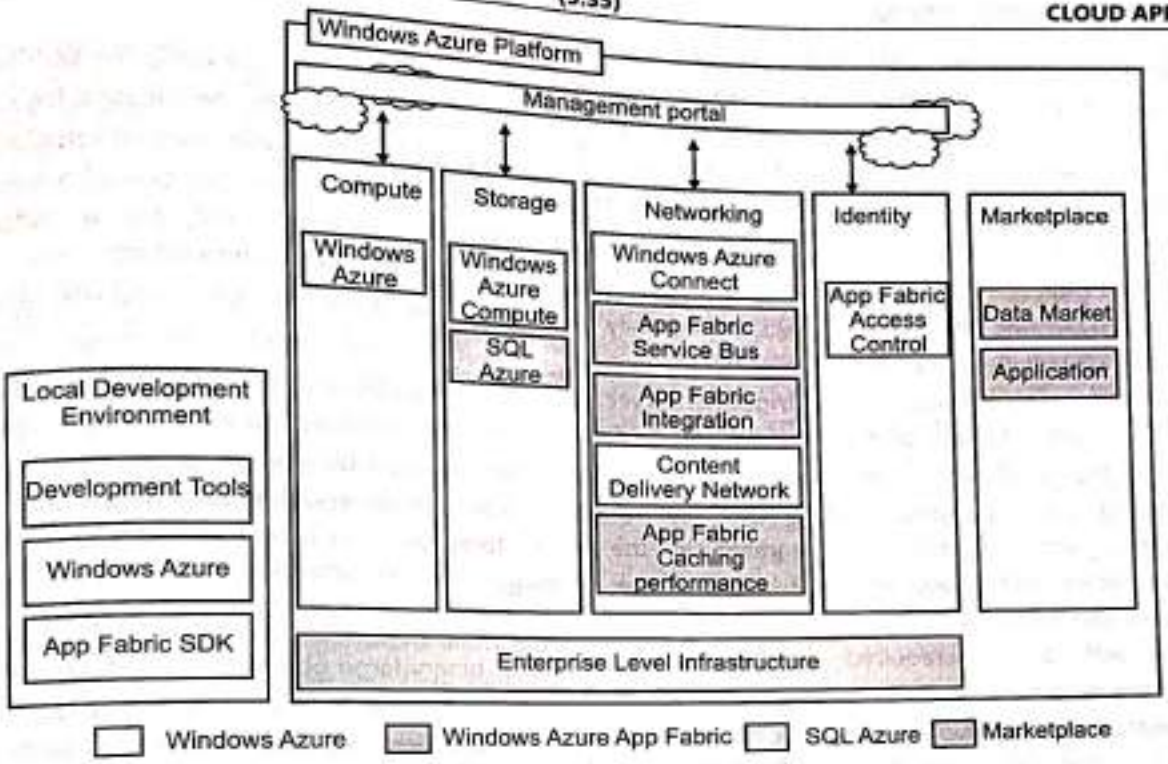


Fig. 5.13 : Microsoft Windows Azure Platform Architecture

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 runtime 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.

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 MVC 2 Web Role) and WCF (WCF Service Web Role) applications. Since IIS 7 also supports the PHP runtime environment by means of the FastCGI module, Web roles can be used to run and scale PHP Web applications on Azure (CGI Web Role). 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 pro- vided 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.

Virtual Machine Role

- The Virtual Machine role allows developers to fully control the computing stack of their compute service by defining a custom image of the Windows Server 2008 R2 operating system and all the service stack required by their applications. The Virtual Machine role is based on the Windows Hyper-V virtualization technology which is natively integrated in the Windows server technology at the base of Azure. Developers can image a Windows server installation complete with all the required applications and components, save it into a Virtual Hard Disk (VHD) file, and upload it to Windows Azure to create compute instances on demand. Different types of instances are available, and Table 5.7 provides an overview of the options offered during 2011-2012.

Table 5.7 : Windows Azure Compute Instances Characteristics, 2011—2012

Compute Instance Type	CPU	Memory	Instance Storage	I/O Performance	Hourly Cost (USD)
Extra Small	1.0 GHz	768 MB	20 GB	Low	$0.04
Small	1.6 GHz	1.75 GB	225 GB	Moderate	$0.12
Medium	2 3 1.6 GHz	3.5 GB	490 GB	High	$0.24
Large	4 3 1.6 GHz	7 GB	1,000 GB	High	$0.48
Extra Large	8 3 1.6 GHz	14 GB	2,040 GB	High	$0.96

- Compared to the Worker and Web roles, the VM role provides finer control of the compute ser- vice and resource that are deployed on the Azure Cloud. An additional administrative effort is required for configuration, installation, and management of services.

Storage Services

- Compute resources are equipped with local storage in the form of a directory on the local file system that can be used to temporarily store information that is useful for the current execution cycle of a role. If the role is restarted and activated on a different physical machine, this information is lost.
- Windows Azure provides different types of storage solutions that complement compute services with a more durable and redundant option compared to local storage. Compared to local storage, these services can be accessed by multiple clients at the same time and from everywhere, thus becoming a general solution for storage.

Blobs

- 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:
  1. **Block Blobs :** 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.
  2. **Page Blobs :** 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 be1TB.
- Blobs storage provides users with the ability to describe the data by adding metadata. It is also possible to take snapshots of a blob for backup purposes. Moreover, to optimize its distribution, blobs storage can leverage the Windows Azure CDN so that blobs are kept close to users requesting them and can be served efficiently.

Azure Drive

- Page blobs can be used to store an entire file system in the form of a single Virtual Hard Drive (VHD) file. This can then be mounted as a part of the NTFS file system by Azure compute resources, thus providing persistent and durable storage. A page blob mounted as part of an NTFS tree is called an Azure Drive.

Tables

- Tables constitute a semi structured storage solution, allowing users to store information in the form of entities with a collection of properties. Entities are stored as rows in the table and are identified by a key, which also constitutes the unique index built for the table. Users can insert, update, delete, and select a subset of the rows stored in the table. Unlike SQL tables, there are no schema enforcing constraints on the properties of entities and there is no facility for representing relation- ships among entities. For this reason, tables are more similar to spreadsheets rather than SQL tables.
- The service is designed to handle large amounts of data and queries returning huge result sets. This capability is supported by partial result sets and table partitions. A partial result set is returned together with a continuation token, allowing the client to resume the query for large result sets. Table partitions allow tables to be divided among several servers for load-balancing purposes. A partition is identified by a key, which is represented by three of the columns of the table.
- Currently, a table can contain up to 100 TB of data, and rows can have up to 255 properties, with a maximum of 1 MB for each row. The maximum dimension of a row key and partition keys is 1KB.

Queues

- Queue storage allows applications to communicate by exchanging messages through durable queues, thus avoiding lost or unprocessed messages. Applications enter messages into a queue, and other applications can read the min a first-in, first-out(FIFO)style.
- To ensure that messages get processed, when an application reads a message it is marked as invisible; hence it will not be available to other clients. Once the application has completed processing the message, it needs to explicitly delete the message from the queue. This two-phase process ensures that messages get processed before they are removed from the queue, and the client failures do not prevent messages from being processed. At the same time, this is also a reason that the queue does not enforce a strict FIFO model: Messages that are read by applications that crash during processing are made available again after a timeout, during which other messages can be read by other clients. An alternative to reading a message is peeking, which allows retrieving the message but letting it stay visible in the queue. Messages that are peeked are not considered processed. All the services described are geo-replicated three times to ensure their availability in case of major disasters. Geo-replication involves the copying of data into a different datacenter that is hundreds or thousands of miles away from the original datacenter.

Core Infrastructure : AppFabric

- AppFabric is a comprehensive middleware for developing, deploying, and managing applications on the cloud or for integrating existing applications with cloud services. AppFabric implements an optimized infrastructure supporting scaling out and high availability; sand boxing and multi tenancy; state management; and dynamic address resolution and routing. On top of this infra- structure, the middleware offers a collection of services that simplify many of the common tasks in a distributed application, such as communication, authentication and authorization, and data access. These services are available through language-agnostic interfaces, thus allowing developers to build heterogeneous applications.

Access Control

- AppFabric provides the capability of encoding access control to resources in Web applications and services into a set of rules that are expressed outside the application code base. These rules give a great degree of flexibility in terms of the ability to secure components of the application and define access control policies for users and groups.
- Access control services also integrate several authentication providers into a single coherent identity management framework. Applications can leverage Active Directory, Windows Live, Google, Facebook, and other services to authenticate users. This feature also allows easy building of hybrid systems, with some parts existing in the private premises and others deployed in the public cloud.

Service Bus

- Service Bus constitutes the messaging and connectivity infrastructure provided with AppFabric for building distributed and disconnected applications in the Azure Cloud and between the private premises and the Azure Cloud. Service Bus allows applications to interact with different protocols and patterns over a reliable communication channel that guarantees delivery.
- The service is designed to allow transparent network traversal and to simplify the development of loosely coupled applications, without renouncing security and reliability and letting developers focus on the logic of the interaction rather than the details of its implementation. Service Bus allows services to be available by simple URLs, which are untied from their deployment location. It is possible to support publish-subscribe models, full-duplex communications point to

point as well as in a peer-to-peer environment, unicast and multicast message delivery in one-way communications, and asynchronous messaging to decouple application components.

- In order to leverage these features, applications need to be connected to the bus, which provides these services. A connection is the Service Bus element that is priced by Azure on a pay-as-you-go basis. Users are billed on a connections-per-month basis, and they can buy advance "connection packs," which have a discounted price, if they can estimate their needs in advance.

Azure Cache

- Windows Azure provides a set of durable storage solutions that allow applications to persist their data. These solutions are based on disk storage, which might constitute a bottleneck for the applications that need to gracefully scale along the clients' requests and dataset size dimensions.
- Azure Cache is a service that allows developers to quickly access data persisted on Windows Azure storage or in SQL Azure. The service implements a distributed in-memory cache of which the size can be dynamically adjusted by applications according to their needs. It is possible to store any .NET managed object as well as many common data formats (table rows, XML, and binary data) and control its access by applications. Azure Cache is delivered as a service, and it can be easily integrated with applications. This is a particularly true for ASP.NET applications, which already integrate providers for session state and page output caching based on Azure Cache.
- The service is priced according the size of cache allocated by applications per month, despite their effective use of the cache. Currently, several cache sizes are available, ranging from 128 MB ($45/month) to 4 GB($325/month).

Other Services

- Compute, storage, and middleware services constitute the core components of the Windows Azure platform. Besides these, other services and components simplify the development and integration of applications with the Azure Cloud. An important area for these services is applications connectivity, including virtual networking and content delivery.

Windows Azure Virtual Network

- Networking services for applications are offered under the name Windows Azure Virtual Network, which includes Windows Azure Connect and Windows Azure Traffic Manager.
- Windows Azure Connect allows easy setup of IP-based network connectivity among machines hosted on the private premises and the roles deployed on the Azure Cloud. This service is particularly useful in the case of VM roles, where machines hosted in the Azure Cloud become part of the private network of the enterprise and can be managed with the same tools used in the private premises.
- Windows Azure Traffic Manager provides load-balancing features for services listening to the HTTP or HTTPS ports and hosted on multiple roles. It allows developers to choose from three different load-balancing strategies: Performance, Round-Robin, and Failover.
- Currently, the two services are still in beta phase and are available for free only by invitation.

Windows Azure Content Delivery Network

- Windows Azure Content Delivery Network (CDN) is the content delivery network solution that improves the content delivery capabilities of Windows Azure Storage and several other Microsoft services, such as Microsoft Windows Update and Bing maps. The service allows serving of Web objects (images, static HTML, CSS, and scripts) as well as streaming content by using a network of 24 locations distributed across the world.

5.7.2 SQL Azure

- SQL Azure is a relational database service hosted on Windows Azure and built on the SQL Server technologies. The service extends the capabilities of SQL Server to the cloud and provides developers with a scalable, highly available, and fault-tolerant relational database. SQL Azure is accessible from either the Windows Azure Cloud or any other location that has access to the Azure Cloud. It is fully compatible with the interface exposed by SQL Server, so applications built for SQL Server can transparently migrate to SQL Azure. Moreover, the service is fully manageable using REST APIs, allowing developers to control databases deployed in the Azure Cloud as well as the firewall rules set up for their accessibility.
- Fig. 5.14 shows the architecture of SQL Azure. Access to SQL Azure is based on the Tabular Data Stream (TDS) protocol, which is the communication protocol underlying all the different interfaces used by applications to connect to a SQL Server-based installation such as ODBC and ADO.NET.
- On the SQL Azure side, access to data is mediated by the service layer, which provides provisioning, billing, and connection-routing services. These services are logically part of server instances, which are managed by SQL Azure Fabric. This is the distributed database middleware that constitutes the infrastructure of SQL Azure and that is deployed on Microsoft datacenters.

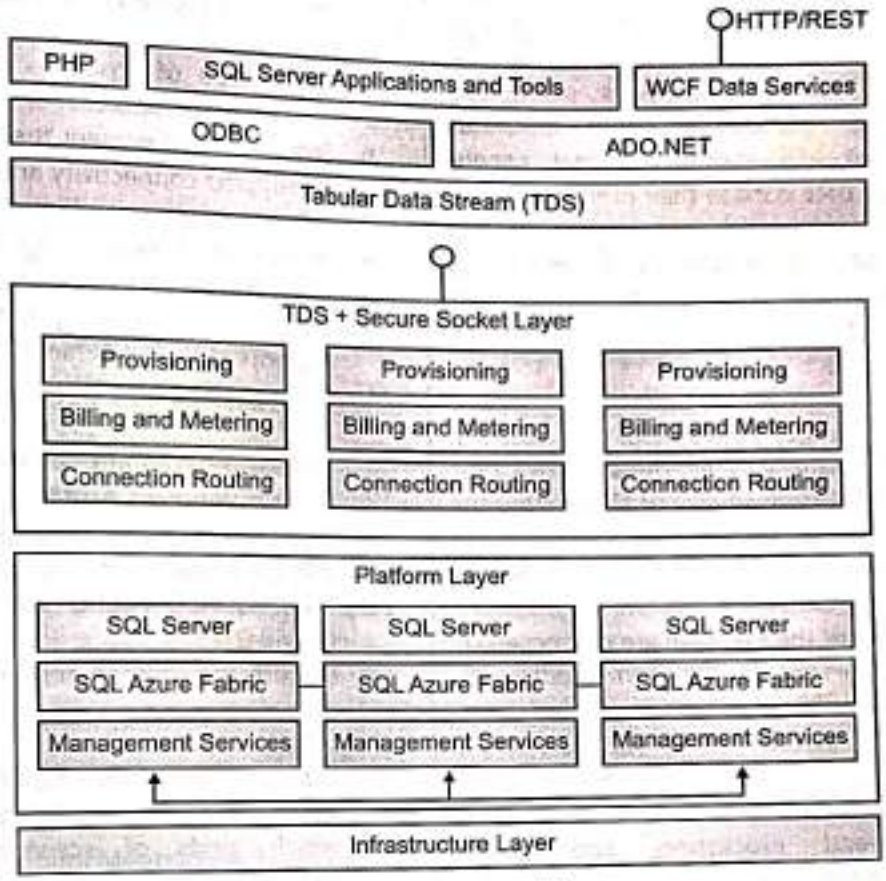


Fig. 5.14 : SQL Azure architecture

- Developers have to sign up for a Windows Azure account in order to use SQL Azure. Once the account is activated, they can either use the Windows Azure Management Portal or the REST APIs to create servers and logins and to configure access to servers. SQL Azure servers are abstractions that closely resemble physical SQL Servers: They have a fully qualified domain name under the database.windows.net (i.e., server-name.database.windows.net) domain name. This simplifies the management tasks and the interaction with SQL Azure from client applications. SQL Azure ensures that multiple copies of each server are maintained within the Azure Cloud and that these copies are kept synchronized when client applications insert, update, and delete data on them.
- Currently, the SQL Azure service is billed according to space usage and the type of edition. Currently, two different editions are available: Web Edition and Business Edition. The former is suited for small Web applications and supports databases with a maximum size of 1 GB or 5 GB. The latter is suited for independent software vendors, line-of-business applications, and enterprise applications and supports databases with a maximum size from 10 GB to 50 GB, in increments of 10 GB. Moreover, a bandwidth fee applies for any data transfer trespassing the Windows Azure Cloud or the region where the database is located. A monthly fee per user/database is also charged and is based on the peak size the database reaches during the month.

Windows Azure Platform Appliance

- The Windows Azure platform can also be deployed as an appliance on third-party data centers and constitutes the cloud infrastructure governing the physical servers of the datacenter. The Windows Azure Platform Appliance includes Windows Azure, SQL Azure, and Microsoft- specified configuration of network, storage, and server hardware. The appliance is a solution that targets governments and service providers who want to have their own cloud computing infrastructure.
- As introduced earlier, Azure already provides a development environment that allows building applications for Azure in their own premises. The local development environment is not intended to be production middleware, but it is designed for developing and testing the functionalities of applications that will eventually be deployed on Azure. The Azure appliance is instead a full-featured implementation of Windows Azure. Its goal is to replicate Azure on a third-

party infrastructure and make available its services beyond the boundaries of the Microsoft Cloud. The appliance addresses two major scenarios: institutions that have very large computing needs (such as government agencies) and institutions that cannot afford to transfer their data outside their premises.

Observations

- Windows Azure is Microsoft's solution for developing cloud computing applications. Azure is an implementation of the PaaS layer and provides the developer with a collection of services and scalable middleware hosted on Microsoft datacenters that address compute, storage, networking, and identity management needs of applications. The services Azure offers can be used either individually or all together for building both applications that integrate cloud features and elastic computing systems completely hosted in the cloud.
- The core components of the platform are composed of compute services, storage services, and middleware. Compute services are based on the abstraction of roles, which identify a sandboxed environment where developers can build their distributed and scalable components. These roles are useful for Web applications, back-end processing, and virtual computing. Storage services include solutions for static and dynamic content, which is organized in the form of tables with fewer constraints than those imposed by the relational model. These and other services are implemented and made available through AppFabric, which constitutes the distributed and scalable middleware of Azure.
- SQL Azure is another important element of Windows Azure and provides support for relational data in the cloud. SQL Azure is an extension of the capabilities of SQL Server adapted for the cloud environment and designed for dynamic scaling.
- The platform is mostly based on the .NET technology and Windows systems, even though other technologies and systems can be supported. For this reason, Azure constitutes the solution of choice for migrating to the cloud applications that are already based on the .NET technology.

EXERCISE

1. What is AWS? What types of services does it provide?
2. Describe Amazon EC2 and its basic features.
3. What is a bucket? What type of storage does it provide?
4. What are the differences between Amazon SimpleDB and AmazonRDS?
5. What type of problems does the Amazon Virtual Private Cloud address?
6. Introduce and present the services provided by AWS to support connectivity among applications.
7. What is the Amazon Cloud Watch?
8. What type of service is AppEngine?
9. Describe the core components of AppEngine.
10. What are the development technologies currently supported by AppEngine?
11. What is DataStore? What type of data can be stored in it?
12. Discuss the compute services offered by AppEngine.
13. What is Windows Azure?
14. Describe the architecture of Windows Azure.
15. What is a role? What types of roles can be used?
16. What is App Fabric, and which services does it provide?
17. Discuss the storage services provided by Windows Azure.
18. What is SQL Azure?
19. Illustrate the architecture of SQL Azure.
20. What is the Windows Azure Platform Appliance? For which kinds of scenarios was this appliance designed?
21. What are the types of applications that can benefit from cloud computing?
22. What fundamental advantages does cloud technology bring to scientific applications?
23. Describe how cloud computing technology can be applied to support remote ECG monitoring.
24. Describe an application of cloud computing technology in the field of biology.
25. What are the advantages cloud computing brings to the field of geoscience? Explain with an example.
26. Describe some examples of CRM and ERP implementations based on cloud computing technologies.
27. What is Salesforce.com?
28. What are Dropbox and iCloud? Which kinds of problems do they solve by using cloud technologies?
29. Describe the key features of Google Apps.
30. What are Web desktops? What is the ir-relationship to cloud computing?
31. What is the most important advantage of cloud technologies for social networking applications?
32. Provide some examples of media applications that use cloud technologies.
33. Describe an application of cloud technologies for online gaming.

✠ ✠ ✠

Semester Examination

PAPER I

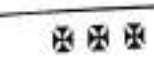
Time : 3 Hours **Max. Marks : 60**

Instructions to the candidates :

1. Each Question carries 12 Marks.
2. Attempt any five questions to the following.
3. Illustrate your answers with neat sketches, diagram etc., wherever necessary.
4. If some part or parameter is noticed to be missing, you may appropriately assume it and should mention it clearly.

Attempt Any Five Questions

1. **Attempt any Two of the Following :**
   (a) Name the role of actors in cloud computing and explain their role in the model. [6]
   (b) Mention the benefits and limitations of cloud computing. [6]
   (c) Define the term cloud computing. What are the characteristics of cloud computing? [6]
2. **Attempt any three of the following.**
   (a) Draw and explain the cloud reference model. [4]
   (b) Describe in brief cloud delivery models. [4]
   (c) Define the term cloud. Mention the types of clouds. [4]
   (d) Write short note on architecture of cloud computing. [4]
3. **Attempt the following.**
   (a) Describe in brief following approaches of XaaS.
   (i) Storage as a Service (ii) Database as a Service [6]
   (b) Explain the concept of "capacity planning" in the context of cloud scaling. [6]
4. **Attempt any two of the following.**
   (a) What do you mean by disaster recovery in the cloud? [6]
   (b) Explain the disaster recovery planning and disaster management approach for cloud. [6]
   (c) What is Aneka SDK? Describe the Aneka SDK component models. [6]
5. **Attempt any two of the following.**
   (a) Describe the Scheduling and execution services in Aneka. [6]
   (b) Describe how cloud computing technology can be applied to support remote ECG monitoring. [6]
   (c) Draw and explain infrastructural organization of Aneka Clouds. [6]
6. **Attempt the following.**
   (a) What fundamental advantages does cloud technology bring to scientific applications? [6]
   (b) What are the types of applications that can benefit from cloud computing? [6]



PAPER II

Time : 3 Hours **Max. Marks : 60**

Instructions to the candidates :

1. Each Question carries 12 Marks.
2. Attempt any five questions to the following.
3. Illustrate your answers with neat sketches, diagram etc., wherever necessary.
4. If some part or parameter is noticed to be missing, you may appropriately assume it and should mention it clearly.

Attempt Any Five Questions

1. Attempt any Two of the Following :

(a) What is cloud migration? What are the benefits of cloud migration? [6]

(b) Mention the characteristics and benefits of cloud computing. [6]

(c) What is virtualization? Mention the types of virtualization. [6]

2. Attempt any two of the following.

(a) Write short note on "Scalability and fault tolerance in cloud computing". [6]

(b) Explain the machine image design in cloud computing. [6]

(c) Explain the concept of database management in cloud computing. [6]

3. Attempt the following.

(a) What do you mean by cloud scale? Differentiate between horizontal scaling and vertical scaling. [6]

(b) Explain the disaster recovery planning and disaster management approach for cloud. [6]

4. Attempt the following.

(a) State and explain the Aneka Cloud deployment models. [6]

(b) Draw and explain logical organization of Aneka clouds. [6]

5. Attempt any two of the following.

(a) Describe the accounting, billing and resource pricing services in Aneka. [6]

(b) What are Web desktops? What is the ir-relationship to cloud computing? [6]

(c) What is the most important advantage of cloud technologies for social networking applications? [6]

6. Attempt the following.

(a) What are Dropbox and iCloud? Which kinds of problems do they solve by using cloud technologies? [6]

(b) What are the advantages cloud computing brings to the field of geoscience? Explain with an example. [6]

✠ ✠ ✠