SRI VENKATESHWARAA COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CLOUD COMPUTING

VIII SEMSTER

CS E811 CLOUD COMPUTING

UNIT – I

Introduction to Cloud Computing: Overview, Roots of Cloud Computing, Layers and Types of Cloud, Desired Features of a Cloud, Benefits and Disadvantages of Cloud Computing, Cloud Infrastructure Management, Infrastructure as a Service Providers, Platform as a Service Providers, Challenges and Risks, Assessing the role of Open Standards.

UNIT – II

Cloud Architecture, Services and Applications: Exploring the Cloud Computing Stack, Connecting to the Cloud, Infrastructure as a Service, Platform as a Service, Saas Vs. Paas, Using PaaS Application Frameworks, Software as a Service, Identity as a Service, Compliance as a Service.

UNIT – III

Abstraction and Virtualization: Introduction to Virtualization Technologies, Load Balancing and Virtualization, Understanding Hyper visors, Understanding Machine Imaging, Porting Applications, Virtual Machines Provisioning and Manageability Virtual Machine Migration Services, Virtual Machine Provisioning and Migration in Action, Provisioning in the Cloud Context

UNIT – IV

Managing & Securing the Cloud: Administrating the Clouds, Cloud Management Products, Emerging Cloud Management Standards, Securing the Cloud, Securing Data, Establishing Identity and Presence

UNIT – V

Case-Studies: Using Google Web Services, Using Amazon Web Services, Using Microsoft Cloud Services

Text Books:

1. Buyya R., Broberg J., Goscinski A., "Cloud Computing: Principles and Paradigm", First Edition, John Wiley & Sons,2011.
2. Sosinsky B., "Cloud Computing Bible", First Edition,Wiley Edition, 2011.

Reference Books:

1. Miller Michael, "Cloud Computing: Web Based Applications that Change the Way You Work and Collaborate Online", Pearson Education India
2. Smooth S., Tan N., "Private Cloud Computing", Morgan Kauffman , First Edition, 2011.
3. Linthicium D., "Cloud Computing and SOA Convergence in Enterprise", Pearson Education India.

Website:

1. www.ibm.com/cloud-computing/
2. www.microsoft.com/enterprise/it-trends/cloud-computing/

UNIT – I

Introduction to Cloud Computing: Overview, Roots of Cloud Computing, Layers and Types of Cloud, Desired Features of a Cloud, Benefits and Disadvantages of Cloud Computing, Cloud Infrastructure Management, Infrastructure as a Service Providers, Platform as a Service Providers, Challenges and Risks, Assessing the role of Open Standards.

1. Introduction to Cloud Computing: Overview:

When plugging an electric appliance into an outlet, we care neither how electric power is generated nor how it gets to that outlet. This is possible because electricity is virtualized; that is, it is readily available from a wall socket that hides power generation stations and a huge distribution grid. When extended to information technologies, this concept means delivering useful functions while hiding how their internals work. Computing itself, to be considered fully virtualized, must allow computers to be built from distributed components such as processing, storage, data, and software resources .

Technologies such as cluster, grid, and now, cloud computing, have all aimed at allowing access to large amounts of computing power in a fully virtualized manner, by aggregating resources and offering a single system view. In addition, an important aim of these technologies has been delivering computing as a utility. Utility computing describes a business model for on-demand delivery of computing power; consumers pay providers based on usage ("payas- you-go"), similar to the way in which we currently obtain services from traditional public utility services such as water, electricity, gas, and telephony. Cloud computing has been coined as an umbrella term to describe a category of sophisticated on-demand computing services initially offered by commercial providers, such as Amazon, Google, and Microsoft. It denotes a model on which a computing infrastructure is viewed as a "cloud," from which businesses and individuals access applications from anywhere in the world on demand .

The main principle behind this model is offering computing, storage, and software "as a service."

Many practitioners in the commercial and academic spheres have attempted to define exactly what "cloud computing" is and what unique characteristics it presents. Buyya et al. have defined it as follows: "Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers." Vaquero et al. have stated "clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized Service Level Agreements." A recent McKinsey and Co. report claims that "Clouds are hardware based services

offering compute, network, and storage capacity where: Hardware management is highly abstracted from the buyer, buyers incur infrastructure costs as variable OPEX, and infrastructure capacity is highly elastic."

A report from the University of California Berkeley summarized the key characteristics of cloud computing as: "(1) the illusion of infinite computing resources; (2) the elimination of an up-front commitment by cloud users; and (3) the ability to pay for use . . . as needed . . ."

The National Institute of Standards and Technology (NIST) characterizes cloud computing as ". . . a pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

In a more generic definition, Armbrust et al. define cloud as the "data center hardware and software that provide services." Similarly, Sotomayor et al. point out that "cloud" is more often used to refer to the IT infrastructure deployed on an Infrastructure as a Service provider data center. While there are countless other definitions, there seems to be common characteristics between the most notable ones listed above, which a cloud should have: (i) pay-per-use (no ongoing commitment, utility prices); (ii) elastic capacity and the illusion of infinite resources; (iii) self-service interface; and (iv) resources that are abstracted or virtualised. In addition to raw computing and storage, cloud computing providers usually offer a broad range of software services. They also include APIs and development tools that allow developers to build seamlessly scalable applications upon their services. The ultimate goal is allowing customers to run their everyday IT infrastructure "in the cloud." A lot of hype has surrounded the cloud computing area in its infancy, often considered the most significant switch in the IT world since the advent of the Internet. In midst of such hype, a great deal of confusion arises when trying to define what cloud computing is and which computing infrastructures can be termed as "clouds."

Indeed, the long-held dream of delivering computing as a utility has been realized with the advent of cloud computing. However, over the years, several technologies have matured and significantly contributed to make cloud computing viable. In this direction, this introduction tracks the roots of cloud computing by surveying the main technological advancements that significantly contributed to the advent of this emerging field. It also explains concepts and developments by categorizing and comparing the most relevant R&D efforts in cloud computing, especially public clouds, management tools, and development frameworks. The most significant practical cloud computing realizations are listed, with special focus on architectural aspects and innovative technical features.

ROOTS OF CLOUD COMPUTING

We can track the roots of clouds computing by observing the advancement of several technologies, especially in hardware (virtualization, multi-core chips), Internet technologies (Web services, service-oriented architectures, Web 2.0), distributed computing (clusters, grids), and systems management (autonomic computing, data center automation). Figure 1.1 shows the convergence of technology fields that significantly advanced and contributed to the advent of cloud computing. Some of these technologies have been tagged as hype in their early stages of development; however, they later received significant attention from academia and were sanctioned by major industry players. Consequently, a specification and standardization process followed, leading to maturity and wide adoption. The emergence of cloud computing itself is closely linked to the maturity of such technologies. We present a closer look at the technologies that form the base of cloud computing, with the aim of providing a clearer picture of the cloud ecosystem as a whole.

From Mainframes to Clouds:

We are currently experiencing a switch in the IT world, from in-house generated computing power into utility-supplied computing resources delivered over the Internet as Web services. This trend is similar to what occurred about a century ago when factories, which used to generate their own electric power, realized that it is was cheaper just plugging their machines into the newly formed electric power grid . Computing delivered as a utility can be defined as "on demand delivery of infrastructure, applications, and business processes in a security-rich, shared, scalable, and based computer environment over the Internet for a fee" .

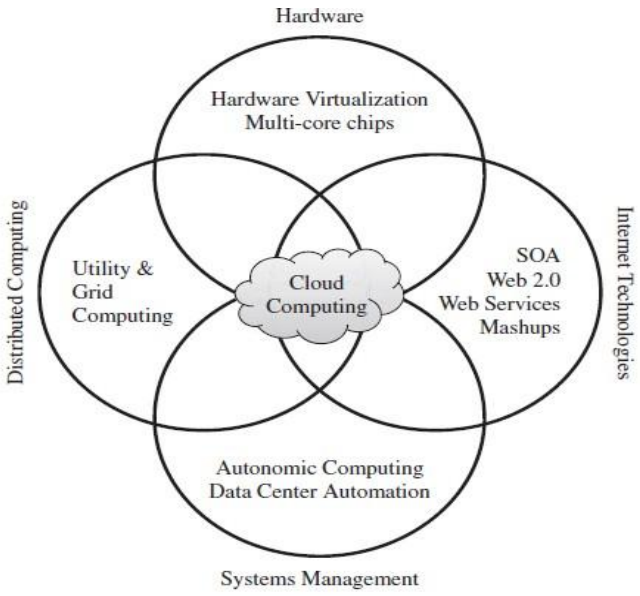


FIGURE 1.1. Convergence of various advances leading to the advent of cloud computing.

This model brings benefits to both consumers and providers of IT services. Consumers can attain reduction on IT-related costs by choosing to obtain cheaper services from external providers as opposed to heavily investing on IT infrastructure and personnel hiring. The "on-demand" component of this model allows consumers to adapt their IT usage to rapidly increasing or unpredictable computing needs. Providers of IT services achieve better operational costs; hardware and software infrastructures are built to provide multiple solutions and serve many users, thus increasing efficiency and ultimately leading to faster return on investment (ROI) as well as lower total cost of ownership (TCO) .

Several technologies have in some way aimed at turning the utility computing concept into reality. In the 1970s, companies who offered common data processing tasks, such as payroll automation, operated time-shared mainframes as utilities, which could serve dozens of applications and often operated close to 100% of their capacity. In fact, mainframes had to operate at very high utilization rates simply because they were very expensive and costs should be justified by efficient usage .

The mainframe era collapsed with the advent of fast and inexpensive microprocessors and IT data centers moved to collections of commodity servers. Apart from its clear advantages, this new model inevitably led to isolation of workload into dedicated servers, mainly due to incompatibilities between software stacks and operating systems. In addition, the unavailability of efficient computer networks meant that IT infrastructure should be hosted in proximity to where it would be consumed. Altogether, these facts have prevented the utility computing reality of taking place on modern computer systems. Similar to old electricity generation stations, which used to power individual factories, computing servers and desktop computers in a modern organization are often underutilized, since IT infrastructure is configured to handle theoretical demand peaks.

In addition, in the early stages of electricity generation, electric current could not travel long distances without significant voltage losses. However, new paradigms emerged culminating on transmission systems able to make electricity available hundreds of kilometers far off from where it is generated. Likewise, the advent of increasingly fast fiber-optics networks has relit the fire, and new technologies for enabling sharing of computing power over great distances have appeared. These facts reveal the potential of delivering computing services with the speed and reliability that businesses enjoy with their local machines. The benefits of economies of scale and high utilization allow providers to offer computing services for a fraction of what it costs for a typical company that generates its own computing power .

SOA, Web Services, Web 2.0, and Mashups

The emergence of Web services (WS) open standards has significantly contributed to advances in the domain of software integration. Web services can glue together applications running on different

messaging product platforms, enabling information from one application to be made available to others, and enabling internal applications to be made available over the Internet. Over the years a rich WS software stack has been specified and standardized, resulting in a multitude of technologies to describe, compose, and orchestrate services, package and transport messages between services, publish and discover services, represent quality of service (QoS) parameters, and ensure security in service access . WS standards have been created on top of existing ubiquitous technologies such as HTTP and XML, thus providing a common mechanism for delivering services, making them ideal for implementing a service-oriented architecture (SOA). The purpose of a SOA is to address requirements of looselycoupled, standards-based, and protocol-independent distributed computing. In a SOA, software resources are packaged as "services," which are well-defined, self contained modules that provide standard business functionality and are independent of the state or context of other services. Services are described in a standard definition language and have a published interface . The maturity of WS has enabled the creation of powerful services that can be accessed on-demand, in a uniform way. While some WS are published with the intent of serving end-user applications, their true power resides in its interface being accessible by other services. An enterprise application that follows the SOA paradigm isa collection of services that together perform complex business logic.

This concept of gluing services initially focused on the enterprise Web, but gained space in the consumer realm as well, especially with the advent of Web 2.0. In the consumer Web, information and services may be programmatically aggregated, acting as building blocks of complex compositions, called service mashups. Many service providers, such as Amazon, del.icio.us, Facebook, and Google, make their service APIs publicly accessible using standard protocols such as SOAP and REST. Consequently, one can put an idea of a fully functional Web application into practice just by gluing pieces with few lines of code. In the Software as a Service (SaaS) domain, cloud applications can be built as compositions of other services from the same or different providers. Services such user authentication, e-mail, payroll management, and calendars are examples of building blocks that can be reused and combined in a business solution in case a single, ready-made system does not provide all those features. Many building blocks and solutions are now available in public marketplaces.

For example, Programmable Web1 is a public repository of service APIs and mashups currently listing thousands of APIs and mashups. Popular APIs such as Google Maps, Flickr, YouTube, Amazon eCommerce, and Twitter, when combined, produce a variety of interesting solutions, from finding video game retailers to weather maps. Similarly, Salesforce.com's offers AppExchange,2 which enables the sharing of solutions developed by third-party developers on top of Salesforce.com components.

Grid Computing

Grid computing enables aggregation of distributed resources and transparently access to them. Most production grids such as TeraGrid and EGEE seek to share compute and storage resources distributed across different administrative domains, with their main focus being speeding up a broad range of scientific applications, such as climate modeling, drug design, and protein analysis. A key aspect of the grid vision realization has been building standard Web services-based protocols that allow distributed resources to be "discovered, accessed, allocated, monitored, accounted for, and billed for, etc., and in general managed as a single virtual system." The Open Grid Services Architecture (OGSA) addresses this need for standardization by defining a set of core capabilities and behaviors that address key concerns in grid systems.

Globus Toolkit is a middleware that implements several standard Grid services and over the years has aided the deployment of several service-oriented Grid infrastructures and applications. An ecosystem of tools is available to interact with service grids, including grid brokers, which facilitate user interaction with multiple middleware and implement policies to meet QoS needs. The development of standardized protocols for several grid computing activities has contributed—theoretically—to allow delivery of on-demand computing services over the Internet. However, ensuring QoS in grids has been perceived as a difficult endeavor. Lack of performance isolation has prevented grids adoption in a variety of scenarios, especially on environments where resources are oversubscribed or users are uncooperative. Activities associated with one user or virtual organization (VO) can influence, in an uncontrollable way, the performance perceived by other users using the same platform. Therefore, the impossibility of enforcing QoS and guaranteeing execution time became a problem, especially for time-critical applications . Another issue that has lead to frustration when using grids is the availability of resources with diverse software configurations, including disparate operating systems, libraries, compilers, runtime environments, and so forth. At the same time, user applications would often run only on specially customized environments. Consequently, a portability barrier has often been present on most grid infrastructures, inhibiting users of adopting grids as utility computing environments.

Virtualization technology has been identified as the perfect fit to issues that have caused frustration when using grids, such as hosting many dissimilar software applications on a single physical platform. In this direction, some research projects (e.g., Globus VirtualWorkspaces) aimed at evolving grids to support an additional layer to virtualize computation, storage, and network resources.

Utility Computing

With increasing popularity and usage, large grid installations have faced new problems, such as excessive spikes in demand for resources coupled with strategic and adversarial behavior by users.

Initially, grid resource management techniques did not ensure fair and equitable access to resources in many systems. Traditional metrics (throughput, waiting time, and slowdown) failed to capture the more subtle requirements of users. There were no real incentives for users to be flexible about resource requirements or job deadlines, nor provisions to accommodate users with urgent work. In utility computing environments, users assign a "utility" value to their jobs, where utility is a fixed or time-varying valuation that captures various QoS constraints (deadline, importance, satisfaction). The valuation is the amount they are willing to pay a service provider to satisfy their demands. The service providers then attempt to maximize their own utility, where said utility may directly correlate with their profit. Providers can choose to prioritize high yield (i.e., profit per unit of resource) user jobs, leading to a scenario where shared systems are viewed as a marketplace, where users compete for resources based on the perceived utility or value of their jobs. Further information and comparison of these utility computing environments are available in an extensive survey of these platforms .

Hardware Virtualization

Cloud computing services are usually backed by large-scale data centers composed of thousands of computers. Such data centers are built to serve many users and host many disparate applications. For this purpose, hardware virtualization can be considered as a perfect fit to overcome most operational issues of data center building and maintenance.

The idea of virtualizing a computer system's resources, including processors, memory, and I/O devices, has been well established for decades, aiming at improving sharing and utilization of computer systems . Hardware virtualization allows running multiple operating systems and software stacks on a single physical platform. As depicted in Figure 1.2, a software layer, the virtual machine monitor (VMM), also called a hypervisor, mediates access to the physical hardware presenting to each guest operating system a virtual machine (VM), which is a set of virtual platform interfaces . The advent of several innovative technologies—multi-core chips, para virtualization, hardware-assisted virtualization, and live migration of VMs—has contributed to an increasing adoption of virtualization on serversystems. Traditionally, perceived benefits were improvements on sharing and utilization, better manageability, and higher reliability. More recently, with the adoption of virtualization on a broad range of server and client systems, researchers and practitioners have been emphasizing three basic capabilities regarding management of workload in a virtualized system, namely isolation, consolidation, and migration . Workload isolation is achieved since all program instructions are fully confined inside a VM, which leads to improvements in security. Better reliability is also achieved because software failures inside one VM do not affect others. Moreover, better performance control is attained since execution of one VM should not affect the performance of another VM . The consolidation of several individual and

heterogeneous workloads onto a single physical platform leads to better system utilization. This practice is also employed for overcoming potential software and hardware incompatibilities in case of upgrades, given that it is possible to run legacy and new operation systems concurrently. Workload migration, also referred to as application mobility, targets at facilitating hardware maintenance, load balancing, and disaster recovery. It is done by encapsulating a guest OS state within a VM and allowing it to be suspended, fully serialized, migrated to a different platform, and resumed immediately or preserved to be restored at a later date. A VM's state includes a full disk or partition image, configuration files, and an image of its RAM.

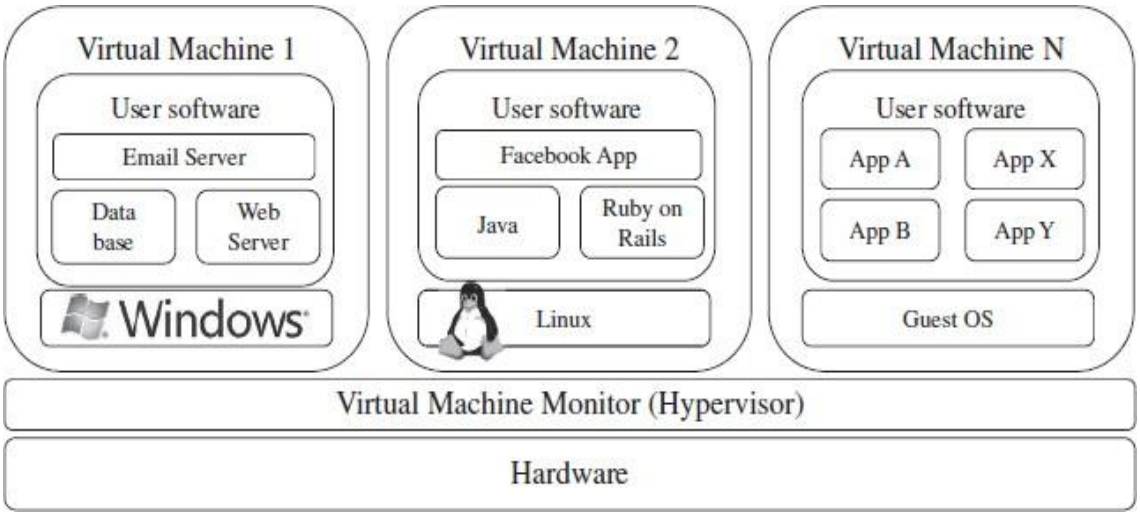


FIGURE 1.2. A hardware virtualized server hosting three virtual machines, each one running distinct operating system and user level software stack.

A number of VMM platforms exist that are the basis of many utility or cloud computing environments. The most notable ones, VMWare, Xen, and KVM, are outlined in the following sections.

VMWare ESXi. VMware is a pioneer in the virtualization market. Its ecosystem of tools ranges from server and desktop virtualization to high-level management tools. ESXi is a VMM from VMWare. It is a bare-metal hypervisor, meaning that it installs directly on the physical server, whereas others may require a host operating system. It provides advanced virtualization techniques of processor, memory, and I/O. Especially, through memory ballooning and page sharing, it can overcommit memory, thus increasing the density of VMs inside a single physical server.

Xen. The Xen hypervisor started as an open-source project and has served as a base to other virtualization products, both commercial and open-source. It has pioneered the para-virtualization concept, on which the guest operating system, by means of a specialized kernel, can interact with the

hypervisor, thus significantly improving performance. In addition to an open-source distribution, Xen currently forms the base of commercial hypervisors of a number of vendors, most notably Citrix XenServer and Oracle VM.

KVM. The kernel-based virtual machine (KVM) is a Linux virtualization subsystem. Is has been part of the mainline Linux kernel since version 2.6.20, thus being natively supported by several distributions. In addition, activities such as memory management and scheduling are carried out by existing kernel features, thus making KVM simpler and smaller than hypervisors that take control of the entire machine. KVM leverages hardware-assisted virtualization, which improves performance and allows it to support unmodified guest operating systems; currently, it supports several versions of Windows, Linux, and UNIX.

Virtual Appliances and the Open Virtualization Format

An application combined with the environment needed to run it (operating system, libraries, compilers, databases, application containers, and so forth) is referred to as a "virtual appliance." Packaging application environments in the shape of virtual appliances eases software customization, configuration, and patching and improves portability. Most commonly, an appliance is shaped as a VM disk image associated with hardware requirements, and it can be readily deployed in a hypervisor.

On-line marketplaces have been set up to allow the exchange of ready-made appliances containing popular operating systems and useful software combinations, both commercial and open-source. Most notably, the VMWare virtual appliance marketplace allows users to deploy appliances on VMWare hypervisors or on partners public clouds, and Amazon allows developers to share specialized Amazon Machine Images (AMI) and monetize their usage on Amazon EC2. In a multitude of hypervisors, where each one supports a different VM image format and the formats are incompatible with one another, a great deal of interoperability issues arises.

For instance, Amazon has its Amazon machine image (AMI) format, made popular on the Amazon EC2 public cloud. Other formats are used by Citrix XenServer, several Linux distributions that ship with KVM, Microsoft Hyper-V, and VMware ESX. In order to facilitate packing and distribution of software to be run on VMs several vendors, including VMware, IBM, Citrix, Cisco, Microsoft, Dell, and HP, have devised the Open Virtualization Format (OVF). It aims at being "open, secure, portable, efficient and extensible". An OVF package consists of a file, or set of files, describing the VM hardware characteristics(e.g., memory, network cards, and disks), operating system details, startup, and shutdown actions, the virtual disks themselves, and other metadata containing product and licensing information. OVF also supports complex packages composed of multiple VMs (e.g., multi-tier applications) . OVF's extensibility has encouraged additions relevant to management of data centers and clouds. Mathews et

al.have devised virtual machine contracts (VMC) as an extension to OVF. A VMC aids in communicating and managing the complex expectations that VMs have of their runtime environment and vice versa. A simple example of a VMC is when a cloud consumer wants to specify minimum and maximum amounts of a resource that a VM needs to function; similarly the cloud provider could express resource limits as a way to bound resource consumption and costs.

Autonomic Computing

The increasing complexity of computing systems has motivated research on autonomic computing, which seeks to improve systems by decreasing human involvement in their operation. In other words, systems should manage themselves, with high-level guidance from humans .

Autonomic, or self-managing, systems rely on monitoring probes and gauges (sensors), on an adaptation engine (autonomic manager) for computing optimizations based on monitoring data, and on effectors to carry out changes on the system. IBM's Autonomic Computing Initiative has contributed to define the four properties of autonomic systems: self-configuration, self optimization, self-healing, and self-protection. IBM has also suggested a reference model for autonomic control loops of autonomic managers, called MAPE-K (Monitor Analyze Plan Execute—Knowledge).

The large data centers of cloud computing providers must be managed in an efficient way. In this sense, the concepts of autonomic computing inspire software technologies for data center automation, which may perform tasks such as: management of service levels of running applications; management of data center capacity; proactive disaster recovery; and automation of VM provisioning .

LAYERS AND TYPES OF CLOUDS

Cloud computing services are divided into three classes, according to the abstraction level of the capability provided and the service model of providers, namely:

(1) Infrastructure as a Service,

(2) Platform as a Service, and

(3) Software as a Service.

Figure 1.3 depicts the layered organization of the cloud stack from physical infrastructure to applications. These abstraction levels can also be viewed as a layered architecture where services of a higher layer can be composed from services of the underlying layer. The reference model of Buyya et al. explains the role of each layer in an integrated architecture. A core middleware manages physical resources and the VMs deployed on top of them; in addition, it provides the required features (e.g., accounting and billing) to offer multi-tenant pay-as-you-go services. Cloud development environments are built on top of infrastructure services to offer application development and deployment capabilities; in this level, various programming models, libraries, APIs, and mashup editors enable the creation of a

range of business, Web, and scientific applications. Once deployed in the cloud, these applications can be consumed by end users.

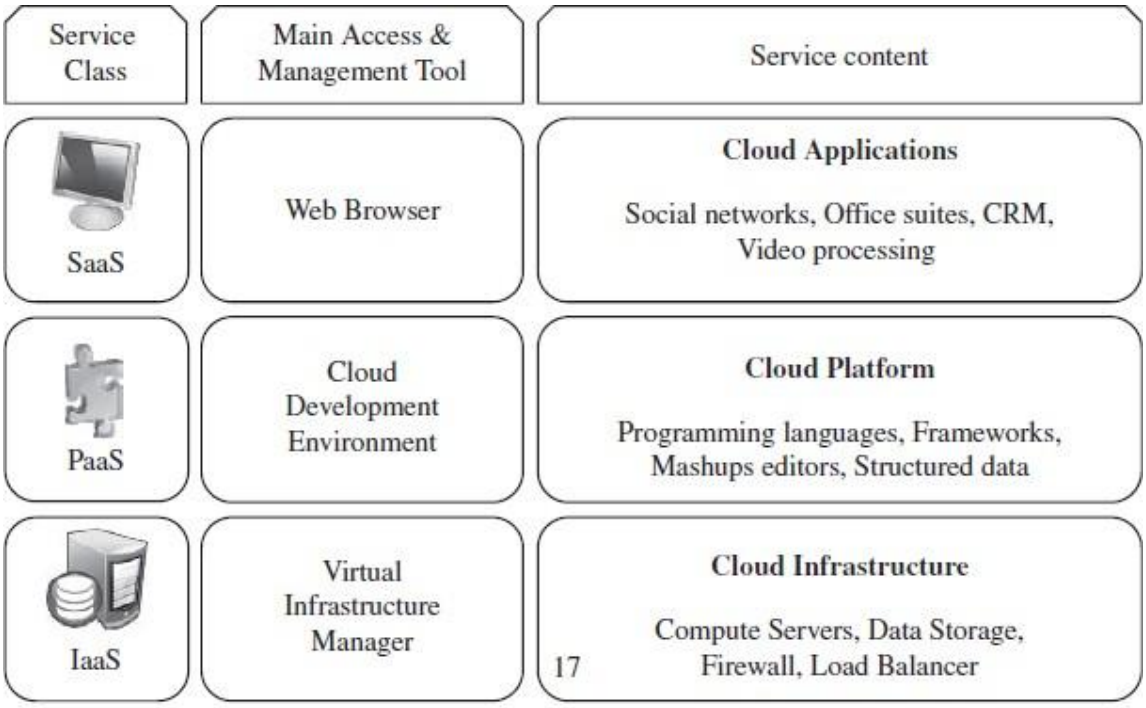


FIGURE 1.3. The cloud computing stack.

Infrastructure as a Service:

Infrastructure as a Service Offering virtualized resources (computation, storage, and communication) on demand is known as Infrastructure as a Service (IaaS). A cloud infrastructure enables on-demand provisioning of servers running several choices of operating systems and a customized software stack. Infrastructure services are considered to be the bottom layer of cloud computing systems.

Amazon Web Services mainly offers IaaS, which in the case of its EC2 service means offering VMs with a software stack that can be customized similar to how an ordinary physical server would be customized. Users are given privileges to perform numerous activities to the server, such as: starting and stopping it, customizing it by installing software packages, attaching virtual disks to it, and configuring access permissions and firewalls rules.

Platform as a Service

In addition to infrastructure-oriented clouds that provide raw computing and storage services, another approach is to offer a higher level of abstraction to make a cloud easily programmable, known as Platform as a Service (PaaS). A cloud platform offers an environment on which developers create and

deploy applications and do not necessarily need to know how many processors or how much memory that applications will be using. In addition, multiple programming models and specialized services (e.g., data access, authentication, and payments) are offered as building blocks to new applications.

Google AppEngine, an example of Platform as a Service, offers a scalable environment for developing and hosting Web applications, which should be written in specific programming languages such as Python or Java, and use the services' own proprietary structured object data store. Building blocks include an in-memory object cache (memcache), mail service, instant messaging service (XMPP), an image manipulation service, and integration with Google Accounts authentication service.

Software as a Service

Applications reside on the top of the cloud stack. Services provided by this layer can be accessed by end users through Web portals. Therefore, consumers are increasingly shifting from locally installed computer programs to on-line software services that offer the same functionally. Traditional desktop applications such as word processing and spreadsheet can now be accessed as a service in the Web. This model of delivering applications, known as Software as a Service (SaaS), alleviates the burden of software maintenance for customers and simplifies development and testing for providers.

Salesforce.com, which relies on the SaaS model, offers business productivity applications (CRM) that reside completely on their servers, allowing costumers to customize and access applications on demand.

Deployment Models

Although cloud computing has emerged mainly from the appearance of public computing utilities, other deployment models, with variations in physical location and distribution, have been adopted. In this sense, regardless of its service class, a cloud can be classified as public, private, community, or hybrid based on model of deployment as shown in Figure 1.4.

Armbrust et al. propose definitions for public cloud as a "cloud made available in a pay-as-you-go manner to the general public" and private cloud as "internal data center of a business or other organization, not made available to the general public."

In most cases, establishing a private cloud means restructuring an existing infrastructure by adding virtualization and cloud-like interfaces. This allows users to interact with the local data center while experiencing the same advantages of public clouds, most notably self-service interface, privileged access to virtual servers, and per-usage metering and billing.

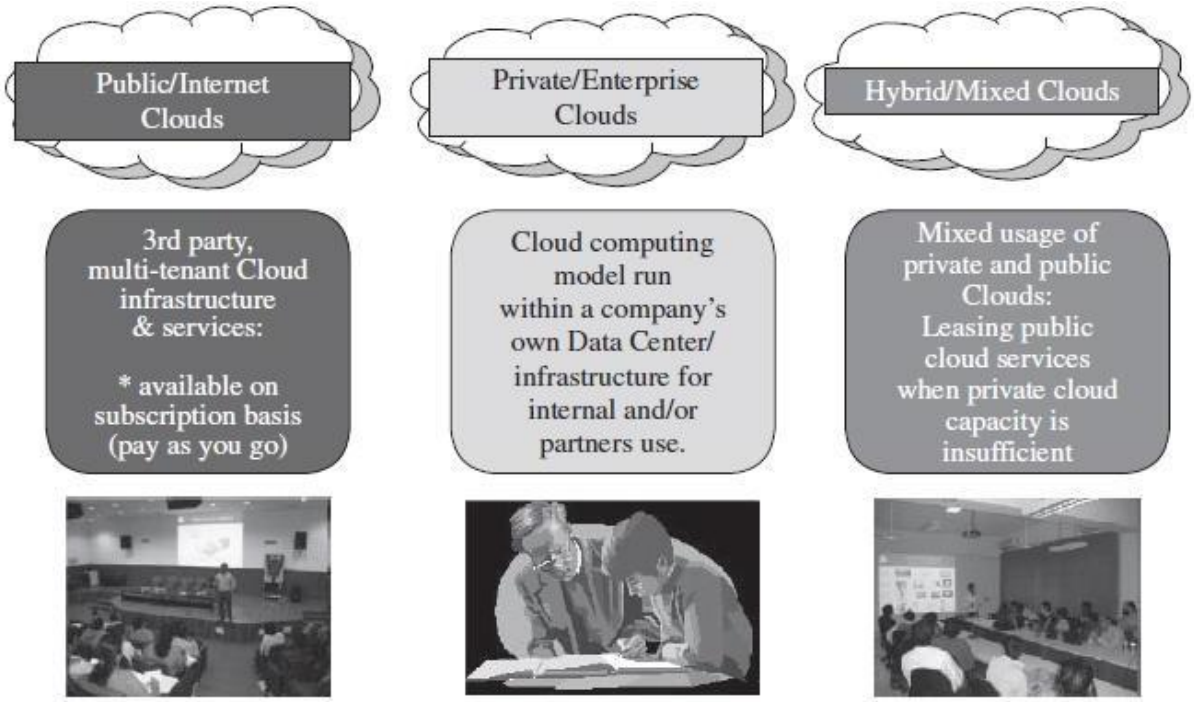


FIGURE 1.4. Types of clouds based on deployment models.

A community cloud is "shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations) ."

A hybrid cloud takes shape when a private cloud is supplemented with computing capacity from public clouds. The approach of temporarily renting capacity to handle spikes in load is known as "cloud-bursting".

DESIRED FEATURES OF A CLOUD

Certain features of a cloud are essential to enable services that truly represent the cloud computing model and satisfy expectations of consumers, and cloud offerings must be (i) self-service, (ii) per-usage metered and billed, (iii) elastic, and (iv)customizable.

Self-Service

Consumers of cloud computing services expect on-demand, nearly instant access to resources. To support this expectation, clouds must allow self-service access so that customers can request, customize, pay, and use services without intervention of human operators .

Per-Usage Metering and Billing

Cloud computing eliminates up-front commitment by users, allowing them to request and use only the necessary amount. Services must be priced on a short term basis (e.g., by the hour), allowing users to release (and not pay for) resources as soon as they are not needed. For these reasons, clouds

must implement features to allow efficient trading of service such as pricing, accounting, and billing. Metering should be done accordingly for different types of service (e.g., storage, processing, and bandwidth) and usage promptly reported, thus providing greater transparency.

Elasticity

Cloud computing gives the illusion of infinite computing resources available on demand. Therefore users expect clouds to rapidly provide resources in any quantity at any time. In particular, it is expected that the additional resources can be (a) provisioned, possibly automatically, when an application load increases and (b) released when load decreases (scale up and down).

Customization

In a multi-tenant cloud a great disparity between user needs is often the case. Thus, resources rented from the cloud must be highly customizable. In the case of infrastructure services, customization means allowing users to deploy specialized virtual appliances and to be given privileged (root) access to the virtual servers. Other service classes (PaaS and SaaS) offer less flexibility and are not suitable for general-purpose computing , but still are expected to provide a certain level of customization.

(a) Benefits of cloud computing

"The NIST Definition of Cloud Computing" by Peter Mell and Tim Grance (version 14, 10/7/2009) (refer to Figure 1.5) that classified cloud computing into the three SPI service models (SaaS, IaaS, and PaaS) and four cloud types (public, private, community, and hybrid), also assigns five essential characteristics that cloud computing systems must offer:

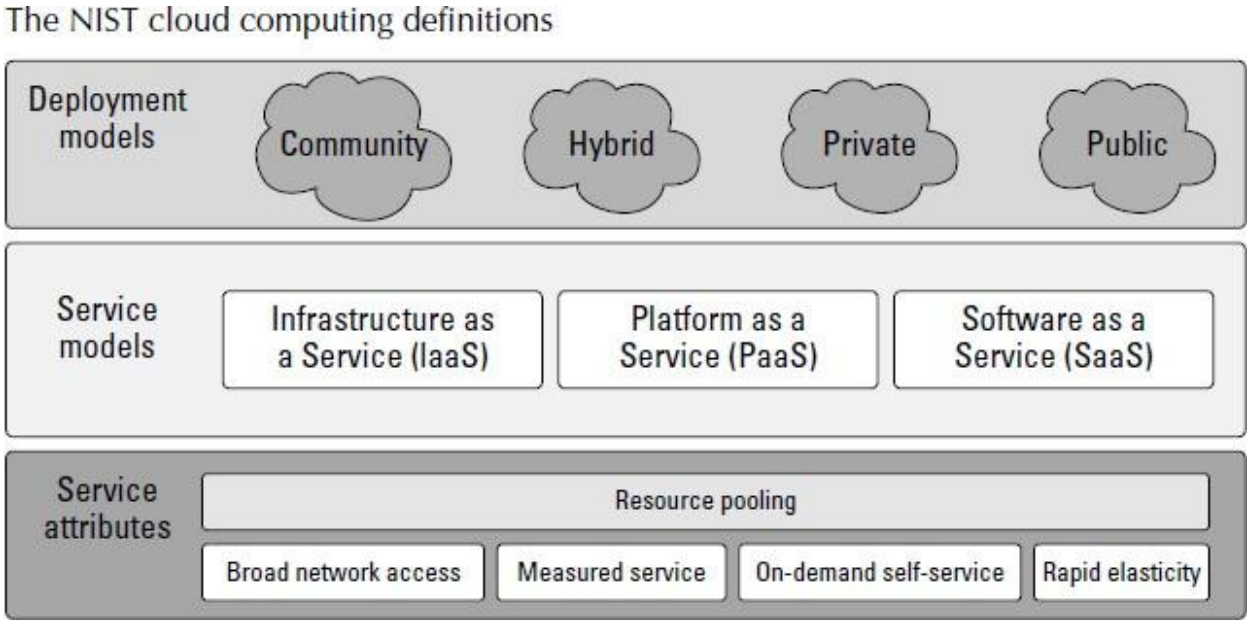


Fig 1.5 The NIST cloud computing definitions

- **On-demand self-service:** A client can provision computer resources without the need for interaction with cloud service provider personnel.
- **Broad network access:** Access to resources in the cloud is available over the network using standard methods in a manner that provides platform-independent access to clients of all types. This includes a mixture of heterogeneous operating systems, and thick and thin platforms such as laptops, mobile phones, and PDA.
- **Resource pooling:** A cloud service provider creates resources that are pooled together in a system that supports multi-tenant usage. Physical and virtual systems are dynamically allocated or reallocated as needed. Intrinsic in this concept of pooling is the idea of abstraction that hides the location of resources such as virtual machines, processing, memory, storage, and network bandwidth and connectivity.
- **Rapid elasticity:** Resources can be rapidly and elastically provisioned. The system can add resources by either scaling up systems (more powerful computers) or scaling out systems (more computers of the same kind), and scaling may be automatic or manual. From the standpoint of the client, cloud computing resources should look limitless and can be purchased at any time and in any quantity.
- **Measured service:** The use of cloud system resources is measured, audited, and reported to the customer based on a metered system. A client can be charged based on a known metric such as amount of storage used, number of transactions, network I/O (Input/Output) or bandwidth, amount of processing power used, and so forth. A client is charged based on the level of services provided.

While these five core features of cloud computing are on almost anybody's list, you also should consider these additional advantages:

- ✓ **Lower costs:** Because cloud networks operate at higher efficiencies and with greater utilization, significant cost reductions are often encountered.
- ✓ **Ease of utilization:** Depending upon the type of service being offered, you may find that you do not require hardware or software licenses to implement your service.
- ✓ **Quality of Service:** The Quality of Service (QoS) is something that you can obtain under contract from your vendor.
- ✓ **Reliability:** The scale of cloud computing networks and their ability to provide load balancing and failover makes them highly reliable, often much more reliable than what you can achieve in a single organization.
- ✓ **Outsourced IT management:** A cloud computing deployment lets someone else manage your computing infrastructure while you manage your business. In most instances, you achieve considerable reductions in IT staffing costs.
- ✓ **Simplified maintenance and upgrade:** Because the system is centralized, you can easily apply patches and upgrades. This means your users always have access to the latest software versions.
- ✓ **Low Barrier to Entry:** In particular, upfront capital expenditures are dramatically reduced. In cloud computing, anyone can be a giant at any time.

(b) DISADVANTAGES OF CLOUD COMPUTING

While the benefits of cloud computing are many, the disadvantages are just as numerous. As a general rule, the advantages of cloud computing present a more compelling case for small organizations than for larger ones. Larger organizations can support IT staff and development efforts that put in place custom software solutions that are crafted with their particular needs in mind.

When you use an application or service in the cloud, you are using something that isn't necessarily as customizable as you might want. Additionally, although many cloud computing applications are very capable, applications deployed on-premises still have many more features than their cloud counterparts.

All cloud computing applications suffer from the inherent latency that is essential in their WAN connectivity. While cloud computing applications excel at large-scale processing tasks, if your application needs large amounts of data transfer, cloud computing may not be the best model for you.

Additionally, cloud computing is a stateless system, as is the Internet in general. In order for communication to survive on a distributed system, it is necessarily unidirectional in nature. All the requests you use in HTTP: PUTs, GETs, and so on are requests to a service provider.

The service provider then sends a response. Although it may seem that you are carrying on a conversation between client and provider, there is an architectural disconnect between the two. That lack of state allows messages to travel over different routes and for data to arrive out of sequence, and many other characteristics allow the communication to succeed even when the medium is faulty. Therefore, to impose transactional coherency upon the system, additional overhead in the form of service brokers, transaction managers, and other middleware must be added to the system. This can introduce a very large performance hit into some applications. If you had to pick a single area of concern in cloud computing, that area would undoubtedly be privacy and security. When your data travels over and rests on systems that are no longer under your control, you have increased risk due to the interception and malfeasance of others. You can't count on a cloud provider maintaining your privacy in the face of government actions.

These days most organizations are faced with regulatory compliance issues of various kinds. In the United States, companies must comply with the accounting requirements of the Sarbanes-Oxley Act; health care providers comply with the data privacy rules of HIPAA, and so on. In Europe, the European Common Market has a raft of its own legislation for companies to deal with. Rules apply to data at rest, and different rules may apply to data in transit. If you stage your cloud computing deployment across states and countries, the bad news is that you may end up having to comply with multiple jurisdictions. Don't expect much support from the cloud system provider or from the governments involved. The laws of most regulatory agencies place the entire burden on the client. So when it comes to compliance, cloud computing is still the "Wild West" of computing.

CLOUD INFRASTRUCTURE MANAGEMENT

A key challenge IaaS providers face when building a cloud infrastructure is managing physical and virtual resources, namely servers, storage, and networks, in a holistic fashion. The orchestration of resources must be performed in a way to rapidly and dynamically provision resources to applications .

The software toolkit responsible for this orchestration is called a virtual infrastructure manager (VIM). This type of software resembles a traditional operating system—but instead of dealing with a single computer, it aggregates resources from multiple computers, presenting a uniform view to user and applications. The term "cloud operating system" is also used to refer to it. Other terms include "infrastructure sharing software " and "virtual infrastructure engine ."

Sotomayor et al. , in their description of the cloud ecosystem of software tools, propose a differentiation between two categories of tools used to manage clouds. The first category—cloud toolkits—includes those that "expose a remote and secure interface for creating, controlling and monitoring virtualize resources," but do not specialize in VI management. Tools in the second category—the virtual infrastructure managers—provide advanced features such as automatic load balancing and server consolidation, but do not expose remote cloud-like interfaces. However, the authors point out that there is a superposition between the categories; cloud toolkits can also manage virtual infrastructures, although they usually provide less sophisticated features than specialized VI managers do.

The availability of a remote cloud-like interface and the ability of managing many users and their permissions are the primary features that would distinguish "cloud toolkits" from "VIMs." However, in this chapter, we place both categories of tools under the same group (of the VIMs) and, when applicable, we highlight the availability of a remote interface as a feature.

Virtually all VIMs we investigated present a set of basic features related to managing the life cycle of VMs, including networking groups of VMs together and setting up virtual disks for VMs. These basic features pretty much define whether a tool can be used in practical cloud deployments or not. On the other hand, only a handful of software present advanced features (e.g., high availability) which allow them to be used in large-scale production clouds.

Features

We now present a list of both basic and advanced features that are usually available in VIMs.

Virtualization Support. The multi-tenancy aspect of clouds requires multiple customers with disparate requirements to be served by a single hardware infrastructure. Virtualized resources (CPUs, memory, etc.) can be sized and resized with certain flexibility. These features make hardware virtualization, the ideal technology to create a virtual infrastructure that partitions a data center among multiple tenants.

Self-Service, On-Demand Resource Provisioning: Self-service access to resources has been perceived as one the most attractive features of clouds. This feature enables users to directly obtain services from clouds, such as spawning the creation of a server and tailoring its software, configurations, and security policies, without interacting with a human system administrator. This capability "eliminates the need for more time-consuming, labor-intensive, human drivenprocurement processes familiar to many in IT".

Therefore, exposing a self-service interface, through which users can easily interact with the system, is a highly desirable feature of a VI manager.

Multiple Backend Hypervisors: Different virtualization models and tools offer different benefits, drawbacks, and limitations. Thus, some VI managers provide a uniform management layer regardless of the virtualization technology used. This characteristic is more visible in open-source VI managers, which usually provide pluggable drivers to interact with multiple hypervisors. In this direction, the aim of libvirt is to provide a uniform API that VI managers can use to manage domains (a VM or container running an instance of an operating system) in virtualized nodes using standard operations that abstract hypervisor specific calls.

Storage Virtualization: Virtualizing storage means abstracting logical storage from physical storage. By consolidating all available storage devices in a data center, it allows creating virtual disks independent from device and location. Storage devices are commonly organized in a storage area network (SAN) and attached to servers via protocols such as Fibre Channel, iSCSI, and NFS; a storage controller provides the layer of abstraction between virtual and physical storage.

In the VI management sphere, storage virtualization support is often restricted to commercial products of companies such as VMWare and Citrix. Other products feature ways of pooling and managing storage devices, but administrators are still aware of each individual device.

Interface to Public Clouds: Researchers have perceived that extending the capacity of a local in-house computing infrastructure by borrowing resources from public clouds is advantageous. In this fashion, institutions can make good use of their available resources and, in case of spikes in demand, extra load can be offloaded to rented resources.

A VI manager can be used in a hybrid cloud setup if it offers a driver to manage the life cycle of virtualized resources obtained from external cloud providers. To the applications, the use of leased resources must ideally be transparent.

Virtual Networking: Virtual networks allow creating an isolated network on top of a physical infrastructure independently from physical topology and locations. A virtual LAN (VLAN) allows isolating traffic that shares a switched network, allowing VMs to be grouped into the same broadcast domain. Additionally, a VLAN can be configured to block traffic originated from VMs from other networks. Similarly, the VPN (virtual private network) concept is used to describe a secure and private overlay network on top of a public network (most commonly the public Internet).

Support for creating and configuring virtual networks to group VMs placed throughout a data center is provided by most VI managers. Additionally, VI managers that interface with public clouds often support secure VPNs connecting local and remote VMs.

Dynamic Resource Allocation: Increased awareness of energy consumption in data centers has encouraged the practice of dynamic consolidating VMs in a fewer number of servers. In cloud infrastructures, where applications have variable and dynamic needs, capacity management and demand prediction are especially complicated. This fact triggers the need for dynamic resource allocation aiming at obtaining a timely match of supply and demand.

Energy consumption reduction and better management of SLAs can be achieved by dynamically remapping VMs to physical machines at regular intervals. Machines that are not assigned any VM can be turned off or put on a low power state. In the same fashion, overheating can be avoided by moving load away from hotspots .

A number of VI managers include a dynamic resource allocation feature that continuously monitors utilization across resource pools and reallocates available resources among VMs according to application needs.

Virtual Clusters: Several VI managers can holistically manage groups of VMs. This feature is useful for provisioning computing virtual clusters on demand, and interconnected VMs for multi-tier Internet applications .

INFRASTRUCTURE AS A SERVICE PROVIDERS

Public Infrastructure as a Service providers commonly offer virtual servers containing one or more CPUs, running several choices of operating systems and a customized software stack. In addition, storage space and communication facilities are often provided.

Features

In spite of being based on a common set of features, IaaS offerings can be distinguished by the availability of specialized features that influence the cost_benefit ratio to be experienced by user applications when moved to the cloud. The most relevant features are:

(i) geographic distribution of data centers;

(ii) variety of user interfaces and APIs to access the system;

(iii) specialized components and services that aid particular applications (e.g., loadbalancers, firewalls);
(iv) choice of virtualization platform and operating systems; and

(v) different billing methods and period (e.g., prepaid vs. post-paid, hourly vs. monthly).

Geographic Presence: To improve availability and responsiveness, a provider of worldwide services would typically build several data centers distributed around the world. For example, Amazon Web Services presents the concept of “availability zones” and “regions” for its EC2 service. Availability zones are “distinct locations that are engineered to be insulated from failures in other availability zones and provide inexpensive, low-latency network connectivity to other availability zones in the same region.” Regions, in turn, “are geographically dispersed and will be in separate geographic areas or countries.”

User Interfaces and Access to Servers: Ideally, a public IaaS provider must provide multiple access means to its cloud, thus catering for various users and their preferences. Different types of user interfaces (UI) provide different levels of abstraction, the most common being graphical user interfaces (GUI), command-line tools (CLI), and Web service (WS) APIs.

GUIs are preferred by end users who need to launch, customize, and monitor a few virtual servers and do not necessary need to repeat the process several times. On the other hand, CLIs offer more flexibility

and the possibility of automating repetitive tasks via scripts (e.g., start and shutdown a number of virtual servers at regular intervals). WS APIs offer programmatic access to a cloud using standard HTTP requests, thus allowing complex services to be built on top of IaaS clouds.

Advance Reservation of Capacity: Advance reservations allow users to request for an IaaS provider to reserve resources for a specific time frame in the future, thus ensuring that cloud resources will be available at that time. However, most clouds only support best-effort requests; that is, users requests are server whenever resources are available.

Amazon Reserved Instances is a form of advance reservation of capacity, allowing users to pay a fixed amount of money in advance to guarantee resource availability at anytime during an agreed period and then paying a discounted hourly rate when resources are in use. However, only long periods of 1 to 3 years are offered; therefore, users cannot express their reservations in finer granularities—for example, hours or days.

Automatic Scaling and Load Balancing: As mentioned earlier in this chapter, elasticity is a key characteristic of the cloud computing model. Applications often need to scale up and down to meet varying load conditions. Automatic scaling is a highly desirable feature of IaaS clouds. It allow users to set conditions for when they want their applications to scale up and down, based on application-specific metrics such as transactions per second, number of simultaneous users, request latency, and so forth.

When the number of virtual servers is increased by automatic scaling, incoming traffic must be automatically distributed among the available servers. This activity enables applications to promptly respond to traffic increase while also achieving greater fault tolerance.

Service-Level Agreement: Service-level agreements (SLAs) are offered by IaaS providers to express their commitment to delivery of a certain QoS. To customers it serves as a warranty. An SLA usually include availability and performance guarantees. Additionally, metrics must be agreed upon by all parties as well as penalties for violating these expectations. Most IaaS providers focus their SLA terms on availability guarantees, specifying the minimum percentage of time the system will be available during a certain period.

For instance, Amazon EC2 states that “if the annual uptime Percentage for a customer drops below 99.95% for the service year, that customer is eligible to receive a service credit equal to 10% of their bill.3” 3 http://aws.amazon.com/ec2-sla

Hypervisor and Operating System Choice: Traditionally, IaaS offerings have been based on heavily customized open-source Xen deployments. IaaS providers needed expertise in Linux, networking, virtualization, metering, resource management, and many other low-level aspects to successfully deploy and maintain their cloud offerings. More recently, there has been an emergence of turnkey IaaS platforms such as VMWare vCloud and Citrix Cloud Center (C3) which have lowered the barrier of entry for IaaS competitors, leading to a rapid expansion in the IaaS marketplace.

Case Studies:

Amazon Web Services. Amazon WS4 (AWS) is one of the major players in the cloud computing market. It pioneered the introduction of IaaS clouds in 2006. It offers a variety cloud services, most notably: S3 (storage), EC2 (virtual servers), Cloudfront (content delivery), Cloudfront Streaming (video streaming), SimpleDB (structured datastore), RDS (Relational Database), SQS (reliable messaging), and Elastic MapReduce (data processing). The ElasticCompute Cloud (EC2) offers Xen-based virtual servers (instances) that can be instantiated from Amazon Machine Images (AMIs). Instances are available in a variety of sizes, operating systems, architectures, and price. CPU capacity of instances is measured in Amazon Compute Units and, although fixed for each instance, vary among instance types from 1 (small instance) to 20 (high CPU instance). Each instance provides a certain amount of nonpersistent disk space; a persistence disk service (Elastic Block Storage) allows attaching virtual disks to instances with space up to 1TB. Elasticity can be achieved by combining the CloudWatch, Auto Scaling, and Elastic Load Balancing features, which allow the number of instances to scale up and down automatically based on a set of customizable rules, and traffic to be distributed across available instances. Fixed IP address (Elastic IPs) are not available by default, but can be obtained at an additional cost.

In summary, Amazon EC2 provides the following features: multiple data centers available in the United States (East and West) and Europe; CLI, Web services (SOAP and Query), Web-based console user interfaces; access to instance mainly via SSH (Linux) and Remote Desktop (Windows); advanced reservation of capacity (aka reserved instances) that guarantees availability for periods of 1 and 3 years; 99.5% availability SLA; per hour pricing; Linux and Windows operating systems; automatic scaling; load balancing.

GoGrid: GoGrid, like many other IaaS providers, allows its customers to utilize a range of pre-made Windows and Linux images, in a range of fixed instance sizes. GoGrid also offers “value-added” stacks on top for applications such as high-volume Web serving, e-Commerce, and database stores. It offers some notable features, such as a “hybrid hosting” facility, which combines traditional dedicated hosts with auto-scaling cloud server infrastructure.

In this approach, users can take advantage of dedicated hosting (which may be required due to specific performance, security or legal compliance reasons) and combine it with on-demand cloud infrastructure as appropriate, taking the benefits of each style of computing.

As part of its core IaaS offerings, GoGrid also provides free hardware load balancing, auto-scaling capabilities, and persistent storage, features that typically add an additional cost for most other IaaS providers.

PLATFORM AS A SERVICE PROVIDERS

Public Platform as a Service providers commonly offer a development and deployment environment that allow users to create and run their applications with little or no concern to low-level details of the platform. In addition, specific programming languages and frameworks are made available in the platform, as well as other services such as persistent data storage and in memory caches.

Features

Programming Models, Languages, and Frameworks. Programming models made available by IaaS providers define how users can express their applications using higher levels of abstraction and efficiently run them on the cloud platform. Each model aims at efficiently solving a particular problem. In the cloud computing domain, the most common activities that require specialized models are: processing of large dataset in clusters of computers (MapReduce model), development of request-based Web services and applications; definition and orchestration of business processes in the form of workflows (Workflow model); and high-performance distributed execution of various computational tasks.

For user convenience, PaaS providers usually support multiple programming languages. Most commonly used languages in platforms include Python and Java (e.g., Google AppEngine), .NET languages (e.g., Microsoft Azure), and Ruby (e.g., Heroku). Force.com has devised its own programming language (Apex) and an Excel-like query language, which provide higher levels of abstraction to key platform functionalities.

A variety of software frameworks are usually made available to PaaS developers, depending on application focus. Providers that focus on Web and enterprise application hosting offer popular frameworks such as Ruby on Rails, Spring, Java EE, and .NET.

Persistence Options: A persistence layer is essential to allow applications to record their state and recover it in case of crashes, as well as to store user data. Traditionally, Web and enterprise application developers have chosen relational databases as the preferred persistence method. These databases offer fast and reliable structured data storage and transaction processing, but may lack scalability to handle several petabytes of data stored in commodity computers.

In the cloud computing domain, distributed storage technologies have emerged, which seek to be robust and highly scalable, at the expense of relational structure and convenient query languages. For example, Amazon SimpleDB and Google AppEngine datastore offer schema-less, automatically indexed database services. Data queries can be performed only on individual tables; that is, join operations are unsupported for the sake of scalability.

Case Studies:

Microsoft Azure: Microsoft Azure Cloud Services offers developers a hosted . NET Stack (C#, VB.Net, ASP.NET). In addition, a Java & Ruby SDK for .NET Services is also available. The Azure system consists of a number of elements.

The Windows Azure Fabric Controller provides auto-scaling and reliability, and it manages memory resources and load balancing. The .NET Service Bus registers and connects applications together.

The .NET Access Control identity providers include enterprise directories and Windows LiveID. Finally, the .NET Workflow allows construction and execution of workflow instances.

Force.com: In conjunction with the Salesforce.com service, the Force.com PaaS allows developers to create add-on functionality that integrates into main Salesforce CRM SaaS application.

Force.com offers developers two approaches to create applications that can be deployed on its SaaS plaform: a hosted Apex or Visual force application.

Apex is a proprietary Java-like language that can be used to create Salesforce applications. Visualforce is an XML-like syntax for building UIs in HTML, AJAX, or Flex to overlay over the Salesforce hosted CRM system. An application store called AppExchange is also provided, which offers a paid & free application directory.

CHALLENGES AND RISKS

Despite the initial success and popularity of the cloud computing paradigm and the extensive availability of providers and tools, a significant number of challenges and risks are inherent to this new model of computing. Providers, developers, and end users must consider these challenges and risks to take good advantage of cloud computing.

Issues to be faced include

- User privacy, data security,
- data lock-in,
- availability of service,
- disaster recovery,
- performance,
- scalability,
- energy-efficiency, and
- programmability.

Security, Privacy, and Trust

Ambrust et al. cite information security as a main issue: "current cloud offerings are essentially public . . . exposing the system to more attacks." For this reason there are potentially additional challenges to make cloud computing environments as secure as in-house IT systems. At the same time, existing, well understood technologies can be leveraged, such as data encryption, VLANs, and firewalls.

Security and privacy affect the entire cloud computing stack, since there is a massive use of third-party services and infrastructures that are used to host important data or to perform critical operations. In this scenario, the trust toward providers is fundamental to ensure the desired level of privacy for applications hosted in the cloud.

Legal and regulatory issues also need attention. When data are moved into the Cloud, providers may choose to locate them anywhere on the planet. The physical location of data centers determines the set of laws that can be applied to the management of data. For example, specific cryptography techniques could not be used because they are not allowed in some countries. Similarly, country laws can impose that sensitive data, such as patient health records, are to be stored within national borders.

Data Lock-In and Standardization

A major concern of cloud computing users is about having their data locked-in by a certain provider. Users may want to move data and applications out from a provider that does not meet their

requirements. However, in their current form, cloud computing infrastructures and platforms do not employ standard methods of storing user data and applications. Consequently, they do not interoperate and user data are not portable.

The answer to this concern is standardization. In this direction, there are efforts to create open standards for cloud computing. The Cloud Computing Interoperability Forum (CCIF) was formed by organizations such as Intel, Sun, and Cisco in order to "enable a global cloud computing ecosystem whereby organizations are able to seamlessly work together for the purposes for wider industry adoption of cloud computing technology." The development of the Unified Cloud Interface (UCI) by CCIF aims at creating a standard programmatic point of access to an entire cloud infrastructure.

Availability, Fault-Tolerance, and Disaster Recovery

It is expected that users will have certain expectations about the service level to be provided once their applications are moved to the cloud. These expectations include availability of the service, its overall performance, and what measures are to be taken when something goes wrong in the system or its components.

In summary, users seek for a warranty before they can comfortably move their business to the cloud. SLAs, which include QoS requirements, must be ideally set up between customers and cloud computing providers to act as warranty. An SLA specifies the details of the service to be provided, including availability and performance guarantees. Additionally, metrics must be agreed upon by all parties, and penalties for violating the expectations must also be approved.

Resource Management and Energy-Efficiency

One important challenge faced by providers of cloud computing services is the efficient management of virtualized resource pools. Physical resources such as CPU cores, disk space, and network bandwidth must be sliced and shared among virtual machines running potentially heterogeneous workloads. The multi-dimensional nature of virtual machines complicates the activity of finding a good mapping of VMs onto available physical hosts while maximizing user utility. Dimensions to be considered include: number of CPUs, amount of memory, size of virtual disks, and network bandwidth.

Dynamic VM mapping policies may leverage the ability to suspend, migrate, and resume VMs as an easy way of preempting low-priority allocations in favor of higher-priority ones. Migration of VMs also brings additional challenges such as detecting when to initiate a migration, which VM to migrate, and where to migrate. In addition, policies may take advantage of live migration of virtual machines to relocate data center load without significantly disrupting running services.

In this case, an additional concern is the trade-off between the negative impact of a live migration on the performance and stability of a service and the benefits to be achieved with that migration.

Another challenge concerns the outstanding amount of data to be managed in various VM management activities. Such data amount is a result of particular abilities of virtual machines, including the ability of traveling through space (i.e., migration) and time (i.e., check pointing and rewinding),

operations that may be required in load balancing, backup, and recovery scenarios. In addition, dynamic provisioning of new VMs and replicating existing VMs require efficient mechanisms to make VM block storage devices (e.g., image files) quickly available at selected hosts.

Data centers consumer large amounts of electricity. According to a data published by HP[4], 100 server racks can consume 1.3MWof power and another 1.3 MW are required by the cooling system, thus costing USD 2.6 million per year.

Besides the economic cost, data centers significantly impact the environment in terms of CO2 emissions from the cooling systems. In addition to optimize application performance, dynamic resource management can also improve utilization and consequently minimize energy consumption in datacenters. This can be done by judiciously consolidating workload onto smaller number of servers and turning off idle resources.

Assessing the Role of Open Standards

When you consider the development of cloud computing to date, it is clear that the technology is the result of the convergence of many different standards. Cloud computing promise of scalability completely changes the manner in which services and applications are deployed. Without standards, the industry creates proprietary systems with vendor lock-in, sometimes referred to as “stovepipe” clouds. Because clients do not want to be locked into any single system, there is a strong industry push to create standards-based clouds.

The cloud computing industry is working with these architectural standards:

- Platform virtualization of resources
- Service-oriented architecture
- Web-application frameworks
- Deployment of open-source software
- Standardized Web services
- Autonomic systems
- Grid computing

These standards help to enable different business models that cloud computing vendors can support, most notably Software as a Service (SaaS), Web 2.0 applications, and utility computing. These businesses require open standards so that data is both portable and universally accessible.

The race to create the first generation of open cloud platform technologies that will compete with proprietary technologies offered by companies such as Microsoft (Azure Platform) and VMware (vSphere) is already underway.

Rackspace.com, one of the large IaaS cloud service providers, announced in July 2010 that it is initiating an open-source project called OpenStack that will begin with the code used to run its Cloud Files and Cloud Servers technologies. NASA has also donated some of the Nebula Cloud Platform technology that it developed. The software developed will be released under the Apache 2.0 license. Founding members of this project include AMD, Citrix, Dell, Intel, NTT Data, and several other cloud service providers. OpenStack.org’s home page (http://www.openstack.org/)

UNIT – II

Cloud Architecture, Services and Applications: Exploring the Cloud Computing Stack, Connecting to the Cloud, Infrastructure as a Service, Platform as a Service, Saas Vs. Paas, Using PaaS Application Frameworks, Software as a Service, Identity as a Service, Compliance as a Service.

Exploring the Cloud Computing Stack

Cloud computing builds on the architecture developed for staging large distributed network applications on the Internet over the last 20 years. To these standard networking protocols, cloud computing adds the advances in system virtualization that became available over the last decade. The cloud creates a system where resources can be pooled and partitioned as needed. Cloud architecture can couple software running on virtualized hardware in multiple locations to provide an on demand service to user-facing hardware and software. It is this unique combination of abstraction and metered service that separates the architectural requirements of cloud computing systems from the general description given for an n-tiered Internet application.

Many descriptions of cloud computing describe it in terms of two architectural layers:

- ✓ **A client as a front end**
- ✓ **The "cloud" as a backend**

This is a very simplistic description because each of these two components is composed of several component layers, complementary functionalities, and a mixture of standard and proprietary protocols. Cloud computing may be differentiated from older models by describing an encapsulated information technology service that is often controlled through an Application Programming Interface (API), thus modifying the services that are delivered over the network.

A cloud can be created within an organization's own infrastructure or outsourced to another datacenter. While resources in a cloud can be real physical resources, more often they are virtualized resources because virtualized resources are easier to modify and optimize. A compute cloud requires virtualized storage to support the staging and storage of data. From a user's perspective, it is important that the resources appear to be infinitely scalable, that the service be measurable, and that the pricing be metered.

a. Composability

Applications built in the cloud often have the property of being built from a collection of components, a feature referred to as composability. A composable system uses components to assemble services that can be tailored for a specific purpose using standard parts. A composable component must be:

- **Modular:** It is a self-contained and independent unit that is cooperative, reusable, and replaceable.
- **Stateless:** A transaction is executed without regard to other transactions or requests.

It isn't an absolute requirement that transactions be stateless, some cloud computing applications provide managed states through brokers, transaction monitors, and service buses. In rare cases, full transactional systems are deployed in the clouds, but these systems are harder to architect in a distributed architecture. Although cloud computing doesn't require that hardware and software be composable, it is a highly desirable characteristic from a developer or user's standpoint, because it makes system design easier to implement and solutions more portable and interoperable.

There is a tendency for cloud computing systems to become less composable for users as the services incorporate more of the cloud computing stack. From the standpoint of an IaaS (Infrastructure as a Service) vendor such as Amazon Web Services, GoGrid, or Rackspace, it makes no sense to offer non-standard machine instances to customers, because those customers are almost certainly deploying applications built on standard operating systems such as Linux, Windows, Solaris, or some other well-known operating system.

In the next step up the cloud computing stack, PaaS (Platform as a Service) vendors such as Windows Azure or Google AppEngine may narrow the definition of standard parts to standard parts that work with their own platforms, but at least from the standpoint of the individual platform service provider, the intent is to be modular for their own developers.

When you move to the highest degree of integration in cloud computing, which is SaaS (Software as a Service), the notion of composability for users may completely disappear. An SaaS vendor such as Quicken.com or Salesforce.com is delivering an application as a service to a customer, and

there's no particular benefit from the standpoint of the service provider that the customer be able to compose its own custom applications. A service provider reselling an SaaS may have the option to offer one module or another, to customize the information contained in the module for a client, to sell the service under their own brand, or to perform some other limited kind of customization, but modifications are generally severely limited.

This idea that composability diminishes going up the cloud computing stack is from the user'spoint of view. If you are a PaaS or SaaS service provider and your task is to create the platform or service presented to the developer, reseller, or user, the notion of working with a composable system is still a very powerful one. A PaaS or SaaS service provider gets the same benefits from a composable system that a user does—these things, among others:

- Easier to assemble systems
- Cheaper system development
- More reliable operation
- A larger pool of qualified developers
- A logical design methodology

You encounter the trend toward designing composable systems in cloud computing in the widespread adoption of what has come to be called the Service Oriented Architecture (SOA). The essence of a service oriented design is that services are constructed from a set of modules using standard communications and service interfaces. An example of a set of widely used standards describes the services themselves in terms of the Web Services Description Language (WSDL), data exchange between services using some form of XML, and the communications between the services using the SOAP protocol. There are, of course, alternative sets of standards.

b. Infrastructure

Most large Infrastructure as a Service (IaaS) providers rely on virtual machine technology to deliver servers that can run applications. Virtual servers described in terms of a machine image or instance have characteristics that often can be described in terms of real servers delivering a certain number of microprocessor (CPU) cycles, memory access, and network bandwidth to customers. Virtual machines are containers that are assigned specific resources. The software that runs in the virtual machines is what defines the utility of the cloud computing system.

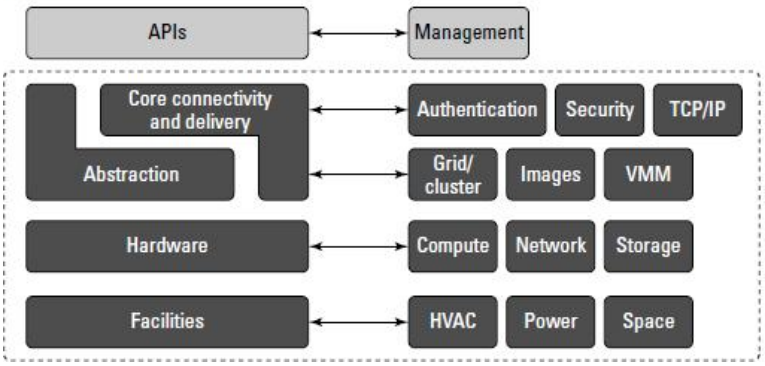
Figure 3.1 shows the portion of the cloud computing stack that is defined as the "server." In the diagram, the API is shown shaded in gray because it is an optional component that isn't always delivered with the server. The VMM component is the Virtual Machine Monitor, also called a hypervisor. This is the low-level software that allows different operating systems to run in their own memory space and manages I/O for the virtual machines.

The notion of a virtual server presents to an application developer a new way of thinking about and programming applications. For example, when a programmer is creating software that requires several different tasks to be performed in parallel, he might write an application that creates additional threads of execution that must be managed by the application. When a developer creates an application that uses a cloud service, the developer can attach to the appropriate service(s) and allow the application itself to scale the program execution. Thus, an application such as a threedimensional rendering that might take a long time for a single server to accomplish can be scaled in the cloud to many servers at once for a short period of time, accomplishing the task at a similar or lower price but at a much faster rate.

In future applications, developers will need to balance the architectural needs of their programs so their applications create new threads when it is appropriate or create new virtual machines. Applications will also need to be mindful of how they use cloud resources, when it is appropriate to scale execution to the cloud, how to monitor the instances they are running, and when not to expand their application's usage of the cloud. This will require a new way of thinking about application development, and the ability to scale correctly is something that will have to be architected into applications from the ground up.

FIGURE 3.1

This architectural diagram illustrates the portion of the cloud computing stack that is designated as the server.



c. Platforms

A platform in the cloud is a software layer that is used to create higher levels of service. There are many different Platform as a Service (PaaS) providers offer services meant to provide developers with different capabilities and three of the major examples that are provided as follows:

- Salesforce.com's Force.com Platform
- Windows Azure Platform
- Google Apps and the Google App Engine

These three services offer all the hosted hardware and software needed to build and deploy Web applications or services that are custom built by the developer within the context and range of capabilities that the platform allows. Platforms represent nearly the full cloud software stack, missing only the presentation layer that represents the user interface. This is the same portion of the cloud computing stack that is a virtual appliance and is shown in Figure 3.2. What separates a platform from a virtual appliance is that the software that is installed is constructed from components and services and controlled through the API that the platform provider publishes.

It makes sense for operating system vendors to move their development environments into them cloud with the same technologies that have been successfully used to create Web applications.Thus, you might find a platform based on a Sun xVM hypervisor virtual machine that includes a NetBeans Integrated Development Environment (IDE) and that supports the Sun GlassFish,Web stack programmable using Perl or Ruby. For Windows, Microsoft would be similarly interested in providing a platform that allowed Windows developers to run on a Hyper-V VM, use the ASP.NET application framework, support one of its enterprise applications such as SQL Server, and be programmable within Visual Studio—which is essentially what the Azure Platform does.This approach allows someone to develop a program in the cloud that can be used by others.

Platforms often come replete with tools and utilities to aid in application design and deployment. Depending upon the vendor, you may find developer tools for team collaboration, testing tools, instrumentation for measuring program performance and attributes, versioning, database and Web service integration, and storage tools. Most platforms begin by establishing a developer community to support the work done in the environment.

Just as a virtual appliance may expose itself to users through an API, so too an application built in the cloud using a platform service would encapsulate the service through its own API. Users would then interact with the platform, consuming services through that API, leaving the platform to manage and scale the service appropriately. Many platforms offer user interface development tools based on HTML, JavaScript, or some other technology. As the Web becomes more media-oriented, many developers have chosen to work with rich Internet environments such as Adobe Flash, Flex, or Air, or alternatives such as Windows Silverlight. A user interface abstracts away the platform API, making those services managed through the UI. Figure 3.3 shows the top portion of the cloud computing stack, which includes the API and the presentation functionality.

FIGURE 3.2

A virtual appliance is software that installs as middleware onto a virtual machine.

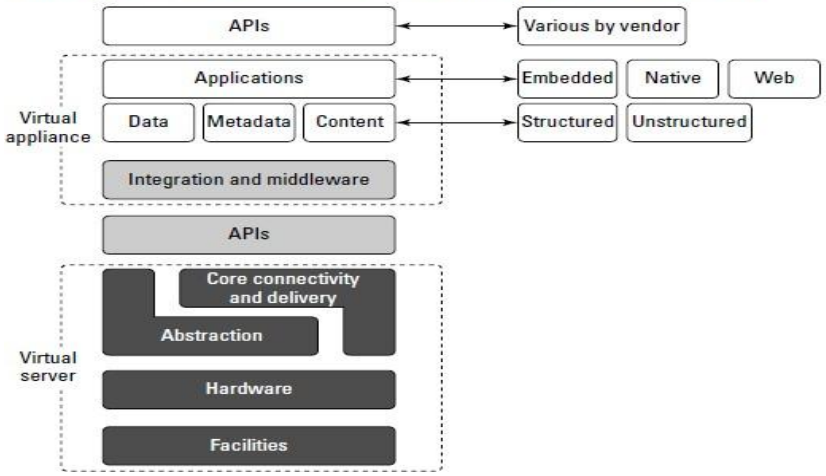
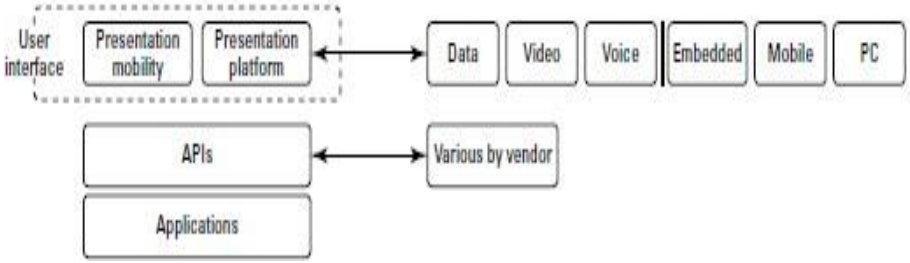


FIGURE 3.3

The top of the cloud computing interface includes the user interface and the API for the application layer.



The Application Programming Interface is one of the key differentiators separating cloud computing from the older models of Internet applications, because it is the means for instantiating resources needed to support applications. An API can control data flow, communications, and other important aspects of the cloud application. Unfortunately, each cloud vendor has their own cloud API, none of them are standard, and the best you can hope for is that eventually the major cloud vendor's APIs will interoperate and exchange data. For now, the use of proprietary APIs results in vendor lock-in, which is why you are advised to choose systems that implement APIs based on open standards.

d. Virtual Appliances

Applications such as a Web server or database server that can run on a virtual machine image are referred to as virtual appliances. The name *virtual appliance* is a little misleading because it conjures up the image of a machine that serves a narrow purpose. Virtual appliances are software installed on virtual servers—application modules that are meant to run a particular machine instance or, image type. A virtual appliance is a platform instance. Therefore, virtual appliances occupy the middle of the cloud computing stack (refer to Figure 3.2).

A virtual appliance is a common deployment object in the cloud, and it is one area where there is considerable activity and innovation. One of the major advantages of a virtual appliance is that you can use the appliances as the basis for assembling more complex services, the appliance being one of your standardized components. Virtual appliances remove the need for application configuration and maintenance from your list of system management chores.

FIGURE 3.4

Amazon Machine Images are a collection of virtual appliances that you can install on their Xen hypervisor servers.

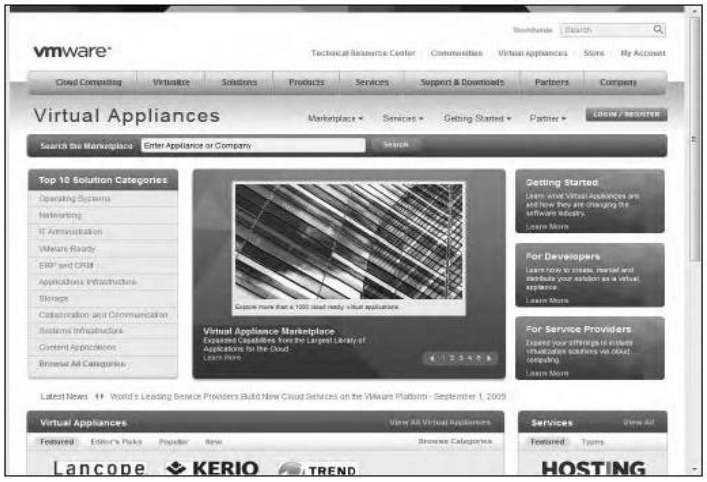
Virtual appliances have begun to affect the PC industry in much the same way that application stores have affected the cell phone industry. You can find various Web sites that either sell or distribute ready-to-use virtual appliances in various forms. Perhaps the best developed of these marketplaces is VMware's Virtual Appliances site (http://www.vmware.com/appliances/) shown in Figure 3.5. These appliances are certified by VMware to be ready to use in the enterprise. Among the other places you can find virtual appliances are at the Web sites of the various operating system vendors, such as Ubuntu, Xen (http://www.xen.org/), and others, including these:

- **Bagvapp** (http://bagside.com/bagvapp/) offers virtual appliances, including ones based on Windows, all of which run on VMware Player.
- **HelpdeskLive**(http://helpdesklive.info/download/VirtualBox%20 VDI%20free%20images.html) offers various Linux distributions upon which you can build a virtual machine.
- **Jcinacio** (http://www.jcinacio.com/) has Ubuntu appliances.
- **Jumpbox** (http://www.jumpbox.com) offers open source virtual appliances installed by them as a managed service. Jumpbox offers virtual appliances for many applications including Bugzilla, DokuWiki, Drupal, Joomla!, Nagios, OpenVPN, PostgreSQL, Redmine,WordPress, and many others. Figure 3.6 shows the Jumpbox home page.
- **QEMU** (http://www.qemu.org/) is a CPU emulator and virtual machine monitor.

- **Parallels** (http://ptn.parallels.com/ptn) hosts a variety of appliances that includes Linux distros, server software, and other products.

FIGURE 3.5

VMware's Virtual Appliance marketplace (`http://www.vmware.com/appliances/`) sells virtual appliances that run on VMware's hypervisor in cloud computing applications.



- **ThoughtPolice** (http://www.thoughtpolice.co.uk/vmware/) offers appliances based on a variety of Linux distributions.
- **VirtualBox** (http://www.virtualbox.org/) is a virtual machine technology now owned by Oracle that can run various operating systems and serves as a host for a variety of virtual appliances
- **Vmachines** (http://www.vmachines.net/) is a site with desktop, server, and securityrelated operating systems that run on VMware.

FIGURE 3.6

Jumpbox (`http://www.jumpbox.com/`) is an open-source virtual appliance installation and management service.

Converting a virtual appliance from one platform to another isn't an easy proposition. Efforts are underway to create file format standards for these types of objects that make this task easier. The best known of these file formats is the Open Virtualization Format (OVF), the work of the Distributed Management Task Force (DMTF) group. Nearly all major virtualization platform vendors support OVF, notably VMware, Microsoft, Oracle, and Citrix.

e. Communication Protocols

Cloud computing arises from services available over the Internet communicating using the standard Internet protocol suite underpinned by the HTTP and HTTPS transfer protocols. The other protocols and standards that expose compute and data resources in the cloud either format data or communications in packets that are sent over these two transport protocols.

In order to engage in interprocess communication (IPC) processes, many client/server protocols have been applied to distributed networking over the years. Various forms of RPC (Remote Procedure Call) implementations (including DCOM, Java RMI, and CORBA) attempt to solve the problem of engaging services and managing transactions over what is essentially a stateless network.The first of the truly Web-centric RPC technologies was XML-RPC, which uses platformindependent XML data to encode program calls that are transported over HTTP, the networking transport to which nearly everyone is connected.

Connecting to the Cloud

Clients can connect to a cloud service in a number of different ways. These are the two most common means:

- A Web browser
- A proprietary application

These applications can be running on a server, a PC, a mobile device, or a cell phone. What these devices have in common with either of these application types is that they are exchanging data over an inherently insecure and transient medium. There are three basic methods for securely connecting over a connection:

- Use a secure protocol to transfer data such as SSL (HTTPS), FTPS, or IPsec, or connect using a secure shell such as SSH to connect a client to the cloud.
- Create a virtual connection using a virtual private network (VPN), or with a remote data transfer protocol such as Microsoft RDP or Citrix ICA, where the data is protected by a tunneling mechanism.
- Encrypt the data so that even if the data is intercepted or sniffed, the data will not be meaningful.

The best client connections use two or more of these techniques to communicate with the cloud. In current browser technology, clients rely on the Web service to make available secure connections, but in the future, it is likely that cloud clients will be hardened so the client itself enforces a secure connection.

If you've ever logged into a hotel connection and browsed the network, you may find that often you can access systems on the network that haven't been protected with a firewall; an improperly configured firewall connection to the cloud is even worse. That has led people to drag portable routers with them, which provide a personal hardware firewall; many of these devices have VPN built directly into them.

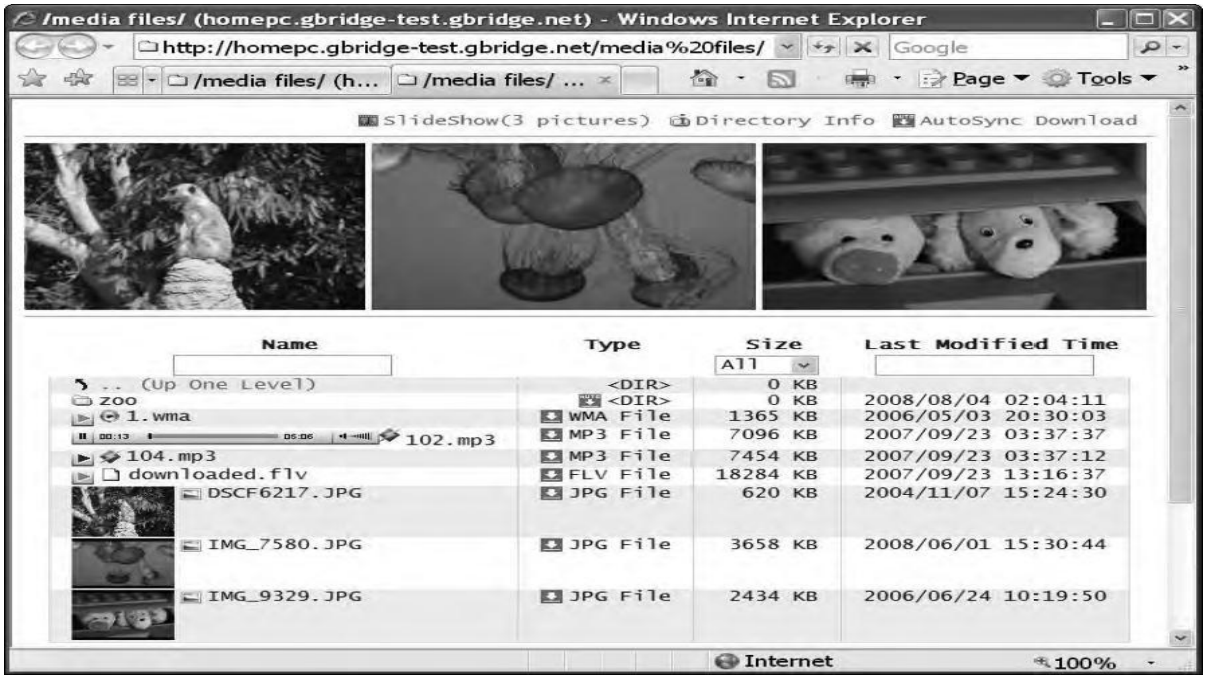
Other solutions include using VPN software; here are three recommended solutions:

- Hotspot VPN (http://www.hotspotvpn.com/)
- AnchorFree Hotspot Shield (http://hotspotshield.com/)
- Gbridge (http://www.gbridge.com/), a third-party VPN based on Google's GoogleTalk infrastructure

Gbridge is an interesting solution that illustrates the use of VPN over a cloud connection. To use this product, you need to log into the GoogleTalk (or Gtalk) network and connect to another computer using your Google account. Gbridge allows additional people to join a connection when invited and supports collaborative features such as desktop sharing using the Virtual Network Computing (VNC) software, chat, live folder browsing, folder synchronization, and automated backup. Gbridge also works with applications deployed using Google Apps, allowing you to securely connect to these applications using a VPN. Figure 3.7 shows browsing a folder over a VPN connection using Gbridge's SecureShares feature.

Figure 3.7 Gbridge provides a means for securely connecting one computer to another using Gtalk.

Shown here is the Secure Shares folder-browsing feature.



The Jolicloud Netbook OS

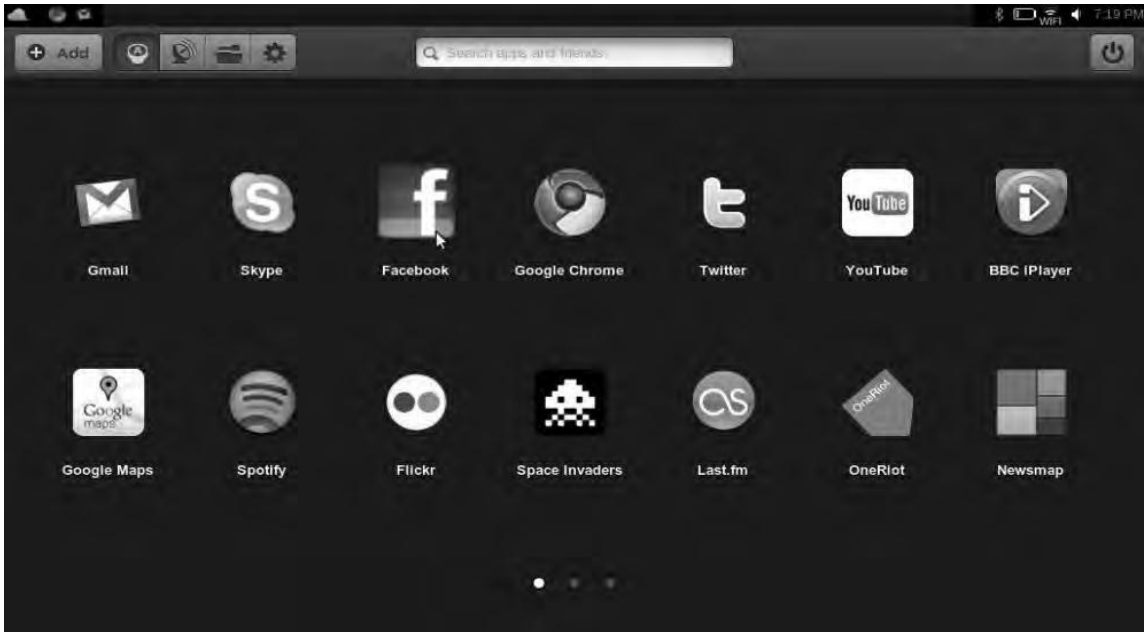
The popularity of ultralight netbooks and mobile phones has greatly expanded the potential audience of dedicated cloud computing devices, but until recently these devices ran standard operating systems such as Windows, Linux, and Macintosh on the PC and Android, IOS, and Windows Mobile (among others) on cell phones. The primary differentiation between these devices is whether or not they are capable of running video and animation (particularly Adobe Flash).

None of these portable devices has been optimized to connect securely to the cloud by narrowing the operating system functions to harden these devices. The French firm Jolicloud (http://www. jolicloud.com/) has recently released a lightweight version of Linux designed specifically to run connected to the cloud as a dedicated cloud client. Jolicloud 1.0 ("The Anywhere OS") can be loaded onto a netbook as the only operating system, or it can be set up as a dual boot system that shares files with a Windows partition.

Jolicloud concentrates on building a social platform with automatic software updates and installs.The application launcher is built in HTML 5 and comes preinstalled with Gmail, Skype, Twitter, Firefox, and other applications. Any HTML 5 browser can be used to work with the Jolicloud interface. Jolicloud maintains a library or App Directory of over 700 applications as part of an app

store. When you click to select an application, the company both installs and updates the application going forward, just as the iPhone manages applications on that device. Figure 3.8 shows the Jolicloud interface.

Figure 3.8 The Jolicloud cloud client operating system is a social networking platform for netbooks with a dedicated application store.



When you install Jolicloud on multiple devices, the system automatically synchronizes your applications so you are working with the same content on all your devices. You can manage your devices from any cloud-connected device. Your files are also unified in a single location, and the operating system provides access to shared storage cloud services such as box.net, Dropbox, and drop.io, among other services.

Chromium OS: The Browser as an Operating System

The Google Chrome OS is a Linux open-source operating system designed to be a robust cloud client.Unlike many other Linux distributions, Google's Chrome is not a software installation, but is shipped installed on validated hardware from Google-approved OEMs (Original Equipment Manufacturers), just as the Android operating system is shipped on a variety of phones. The intent is to

have a tightly coupled hardware offering that supports features in the Chrome OS and that would be highly efficient. Early designs have shown Chrome running on tablet designs that would position it as an Apple iPad competitor running on netbook-type devices in the $300-400 range.

Note: An OEM or original equipment manufacturer builds systems from components and sells them under a brand name.

The expectation is that the first versions of Chrome systems will appear in late 2010, perhaps in both consumer and enterprise offerings. There is also an open-source version of this cloud client called Chromium (http://www.chromium.org/chromium-os), which shares the same code base. The Chromium architecture is built as a three-tier system with a hardware layer, the browser and window manager, and a set of system software and utilities.

The Chrome OS has been described as a hardened operating system because it incorporates a sandbox architecture for running applications and also performs automatic updates. Also included in the system is a version of remote desktop connection software that creates an encrypted connection like Microsoft's RDP, Citrix's ICA, or a VNC client. The Chrome OS hardware specification includes a Trusted Platform Module, which provides for a "trusted bootpath" along with a hardware switch that can be used to boot the system into a developer model. In that mode, some of the security features are turned off, allowing the user to reset the system.

Demonstrations of early prototypes of Chrome and Chromium OS systems have shown that they are capable of nearly instantaneous startup. Chromium Linux kernel has adopted the Upstart (http://upstart.ubuntu.com/) event-based replacement for the init daemon, which is used to launch services concurrently, restore stalled jobs, and perform delayed system startup.

The fast boot time is possible because the device is devoid of most of the devices in modern PCs. Chromium has also adopted a set of security routines in firmware that run during startup and store the information necessary to perform verified system restoration.

Essentially, the Chrome OS looks like the Chrome browser. Chrome is interesting because Google has essentially stripped down the operating system to run one specific application that connects to the Internet. The user interface is similar to the Chrome Web browser and includes a media player that plays MP3, views JPEGs, and plays media content both online and offline. Adobe Flash is integrated directly into the Chrome OS, just as it is in the Chrome browser. When you launch Chrome, you see links to the major Google cloud applications such as Gmail, Google Apps, and YouTube, along with other major sites such as Facebook, Hulu, Pandora, Twitter, and others.Figure 3.9 shows an early demonstration of the Chrome OS, its multi-tab interface, and its application launcher utility. From this same demonstration is shown the Google Reader application with a page from *Alice in Wonderland,* displayed in Figure 3.10.

Google will include on Chromium the Google Cloud Print service that allows an application to print to any connected printer without accessing a printer driver. This system frees Chromium from having to develop hardware- and OS-specific print subsystems. Instead, a proxy is installed in

Chrome that registers a printer with the service, and this proxy manages print jobs for the user. The Chrome OS devices that appear, as well as the competitors such as the Apple iPad and a host of other

similar devices from other system vendors, signal a sea change in the manner in which users access the cloud, and they represent the cloud's impact on the manner in which many users perform their daily work. It's anyone's guess how impactful these introductions will be, but it is clear that they are not simply another competitor to Windows and Macintosh desktop-oriented systems. They represent the move into a cloud-based future where applications run and data is stored remotely. Their success is likely to be contingent upon how fast consumers and businesses become comfortable with the idea of outsourcing these functions. In time, the transition is probably inevitable because the economies of scale and efficiency that cloud computing offers is too compelling to ignore.

FIGURE 3.9 The Chrome OS operating system's application launcher from an early demo of the product found at http://www.youtube.com/watch?v=ANMrzw7JFzA&feature=channel

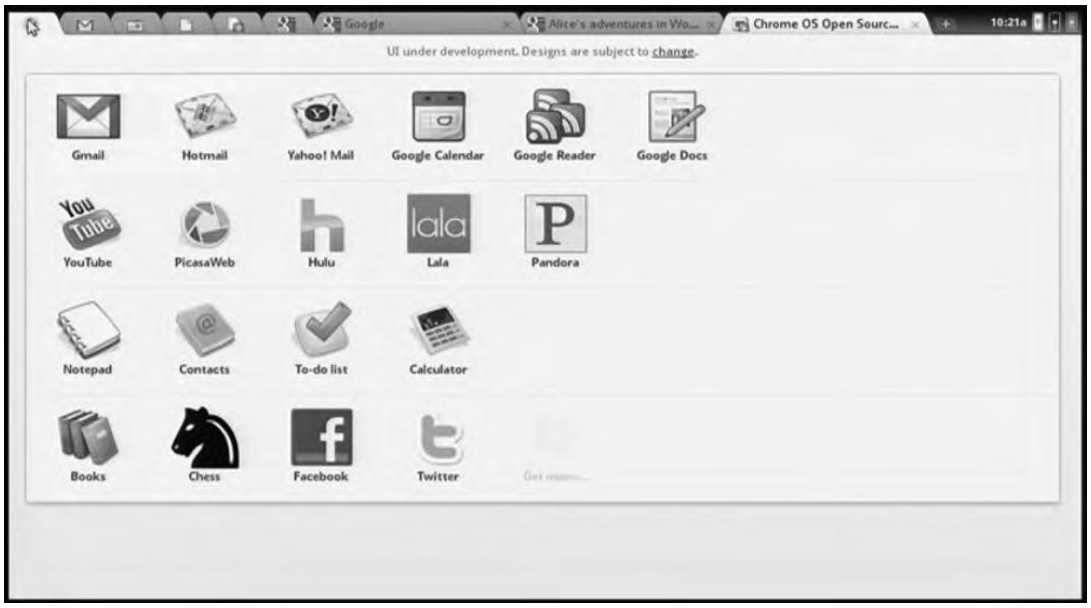
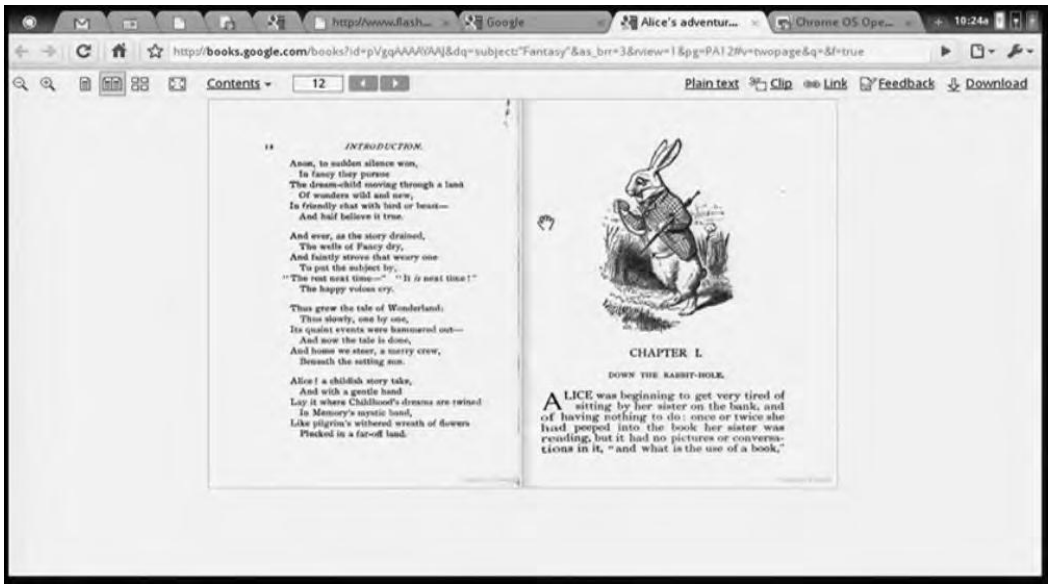


Figure 3.10. From the same source as Figure 3.9, this figure illustrates how the Google Reader appears within the Chrome OS as a tab.



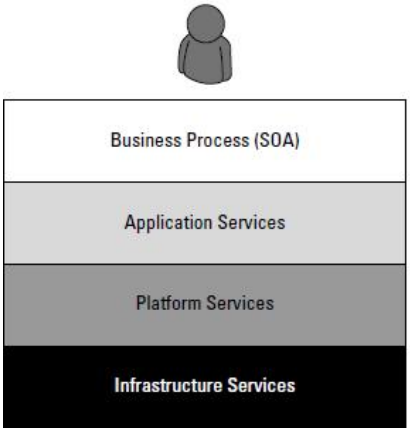
Defining Infrastructure as a Service (IaaS)

You can broadly partition cloud computing into four layers that form a cloud computing ecosystem, as shown in Figure 4.1. The Application layer forms the basis for Software as a Service (SaaS), while the Platform layer forms the basis for Platform as a Service (PaaS) models that are described in the next two sections. Infrastructure as a Service (IaaS) creates what may be determined to be a utility computing model, something that you can tap into and draw from as you need it without significant limits on the scalability of your deployment. You pay only for what you need when you need it. IaaS may be seen to be an incredibly disruptive technology, one that can help turn a small business into a large business nearly overnight. This is a most exciting prospect; one that is fueling a number of IaaS startups during one of the most difficult recessions of recent memory.

Infrastructure as a Service (IaaS) is a cloud computing service model in which hardware is virtualized in the cloud. In this particular model, the service vendor owns the equipment: servers, storage, network infrastructure, and so forth. The developer creates virtual hardware on which to develop applications and services. Essentially, an IaaS vendor has created a hardware utility service where the user provisions virtual resources as required.

FIGURE 4.1

The cloud computing ecosystem



The developer interacts with the IaaS model to create virtual private servers, virtual private storage, virtual private networks, and so on, and then populates these virtual systems with the applications and services it needs to complete its solution. In IaaS, the virtualized resources are mapped to real systems. When the client interacts with an IaaS service and requests resources from the virtual systems, those requests are redirected to the real servers that do the actual work.

IaaS workloads

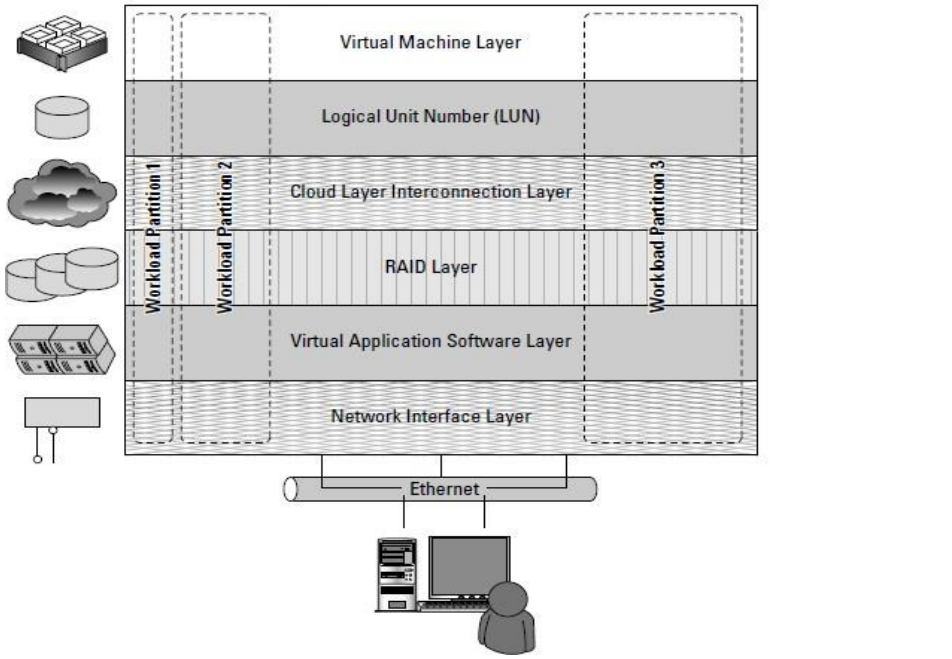
The fundamental unit of virtualized client in an IaaS deployment is called a *workload.* A workload simulates the ability of a certain type of real or physical server to do an amount of work. The work done can be measured by the number of Transactions Per Minute (TPM) or a similar metric against a certain type of system. In addition to throughput, a workload has certain other attributes such as Disk I/Os measured in Input/Output Per Second IOPS, the amount of RAM consumed under load in MB, network throughput and latency, and so forth. In a hosted application environment, a client's

application runs on a dedicated server inside a server rack or perhaps as a standalone server in a room full of servers. In cloud computing, a provisioned server called an instance is reserved by a customer, and the necessary amount of computing resources needed to achieve that type of physical server is allocated to the client's needs.

Figure 4.2 shows how three virtual private server instances are partitioned in an IaaS stack.

FIGURE 4.2

A virtual private server partition in an IaaS cloud



The three workloads require three different sizes of computers: small, medium, and large. A client would reserve a machine equivalent required to run each of these workloads. The IaaS infrastructure runs these server instances in the data center that the service offers, drawing from a pool of virtualized machines, RAID storage, and network interface capacity. These three layers are expressions of physical systems that are partitioned as logical units.

LUNs (Logical Unit Number), the cloud interconnect layer, and the virtual application software layer are logical constructs. LUNs are logical storage containers, the cloud interconnect layer is a virtual network layer that is assigned IP addresses from the IaaS network pool, and the virtual application software layer contains software that runs on the physical VM instance(s) that have been partitioned from physical assets on the IaaS' private cloud.

From an architectural standpoint, the client in an IaaS infrastructure is assigned its own private network. The Amazon Elastic Computer Cloud (EC2), described in detail in Chapter 8, behaves as if each server is its own separate network—unless you create your own Virtual Private Cloud (an EC2 add-on feature), which provides a workaround to this problem. When you scale your EC2 deployment, you are adding additional networks to your infrastructure, which makes it easy to logically scale an EC2 deployment, but imposes additional network overhead because traffic must be routed between logical networks. Amazon Web Service's routing limits broadcast and multicast traffic because Layer-2 (Data

Link) networking is not supported. Rackspace Cloud (http://www. rackspacecloud.com/) follows the AWS IP assignment model.

Other IaaS infrastructures such as the one Cloudscaling.com (http://www.cloudscaling.com) offers or a traditional VMWare cloud-assigned networks on a per-user basis, which allows for Level 2 networking options. The most prominent Level 2 protocols that you might use are tunnelling options, because they enable VLANs.

Consider a transactional eCommerce system, for which a typical stack contains the following components:

- Web server
- Application server
- File server
- Database
- Transaction engine

This eCommerce system has several different workloads that are operating: queries against the database, processing of business logic, and serving up clients' Web pages.

The classic example of an IaaS service model is Amazon.com's Amazon Web Services (AWS). AWS has several data centers in which servers run on top of a virtualization platform (Xen) and may be partitioned into logical compute units of various sizes. Developers can then apply system images containing different operating systems and applications or create their own system images. Storage may be partitions, databases may be created, and a range of services such a messaging and notification can be called upon to make distributed application work correctly.

Pods, aggregation, and silos

Workloads support a certain number of users, at which point you exceed the load that the instance sizing allows. When you reach the limit of the largest virtual machine instance possible, you must make a copy or clone of the instance to support additional users. A group of users within a particular instance is called a *pod.* Pods are managed by a Cloud Control System (CCS). In AWS, the CCS is the AWS Management Console.

Sizing limitations for pods need to be accounted for if you are building a large cloud-based application. Pods are aggregated into pools within an IaaS region or site called an *availability zone.*

In very large cloud computing networks, when systems fail, they fail on a pod-by-pod basis, and often on a zone-by-zone basis. For AWS' IaaS infrastructure, the availability zones are organized around the company's data centers in Northern California, Northern Virginia, Ireland, and Singapore. A failover system between zones gives IaaS private clouds a very high degree of availability.

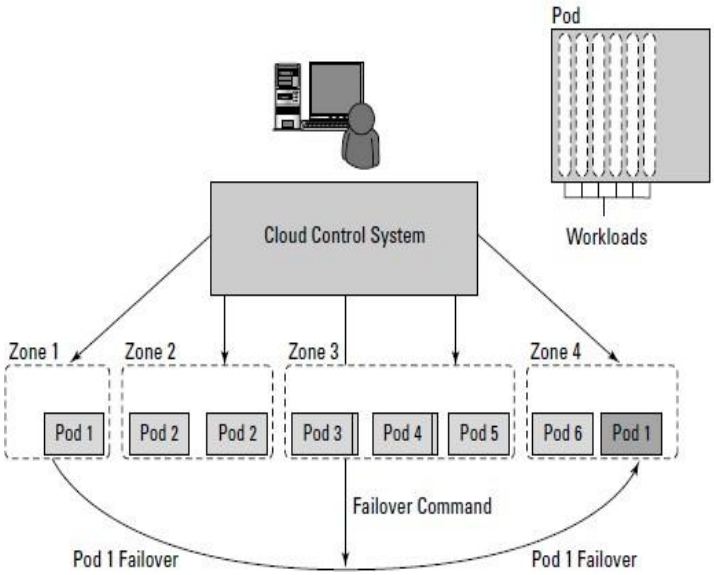
When a cloud computing infrastructure isolates user clouds from each other so the management system is incapable of interoperating with other private clouds, it creates an information silo, or simply a silo. Most often, the term *silo* is applied to PaaS offerings such as Force.com or QuickBase, but silos often are an expression of the manner in which a cloud computing infrastructure is architected.

Silos are the cloud computing equivalent of compute islands: They are processing domains that are sealed off from the outside.

Figure 4.3 shows how pods are aggregated and virtualized in IaaS across zones.

FIGURE 4.3

Pods, aggregation, and failover in IaaS



When you create a private virtual network within an IaaS framework, the chances are high that you are creating a silo. Silos impose restrictions on interoperability that runs counter to the open nature of build-componentized service-oriented applications. However, that is not always a bad thing. A silo can be its own ecosystem; it can be protected and secured in ways that an open system can't be. Silos just aren't as flexible as open systems and are subject to vendor lock-in.

Defining Platform as a Service (PaaS)

The Platform as a Service model describes a software environment in which a developer can create customized solutions within the context of the development tools that the platform provides. Platforms can be based on specific types of development languages, application frameworks, or other constructs. A PaaS offering provides the tools and development environment to deploy applications on another vendor's application. Often a PaaS tool is a fully integrated development environment; that is, all the tools and services are part of the PaaS service. To be useful as a cloud computing offering, PaaS systems must offer a way to create user interfaces, and thus support standards such as HTLM, JavaScript, or other rich media technologies.

In a PaaS model, customers may interact with the software to enter and retrieve data, perform actions, get results, and to the degree that the vendor allows it, customize the platform involved.The customer takes no responsibility for maintaining the hardware, the software, or the development of the applications and is responsible only for his interaction with the platform. The vendor is responsible for all the operational aspects of the service, for maintenance, and for managing the product(s) lifecycle.

The one example that is most quoted as a PaaS offering is Google's App Engine platform, which is described in more detail in Chapter 8. Developers program against the App Engine using Google's published APIs. The tools for working within the development framework, as well as the structure of the file system and data stores, are defined by Google. Another example of a PaaS offering is Force.com, Salesforce.com's developer platform for its SaaS offerings, described in the next section.Force.com is an example of an add-on development environment.

A developer might write an application in a programming language like Python using the Google API. The vendor of the PaaS solution is in most cases the developer, who is offering a complete solution to the customer. Google itself also serves as a PaaS vendor within this system, because it offers many of its Web service applications to customers as part of this service model. You can think of Google Maps, Google Earth, Gmail, and the myriad of other PaaS offerings as conforming to the PaaS service model, although these applications themselves are offered to customers under what is more aptly described as the Software as a Service (SaaS) model that is described below.

The difficulty with PaaS is that it locks the developer (and the customer) into a solution that is dependent upon the platform vendor. An application written in Python against Google's API using the Google App Engine is likely to work only in that environment. There is considerable vendor lock-in associated with a PaaS solution.

Saas Vs. Paas

Salesforce.com versus Force.com: SaaS versus PaaS

There can be no better example illustrating the difference between a SaaS and PaaS system than that of Salesforce.com and Force.com. Salesforce.com is a Web application suite that is an SaaS.

Force.com is Salesforce.com's PaaS platform for building your own services. Salesforce.com was formed by several Oracle employees in 1999 to create a hosted Customer Relationship Management (CRM) system. CRM has long been one of Oracle's core database services. The Salesforce.com team created hosted software based on a cloud computing model: pay as you go, simple to use, and multifunctional. The Salesforce.com platform looks like a typical Web site such as Amazon.com, with a multi-tabbed interface—each tab being an individual application.

Some of the applications included in the site are:

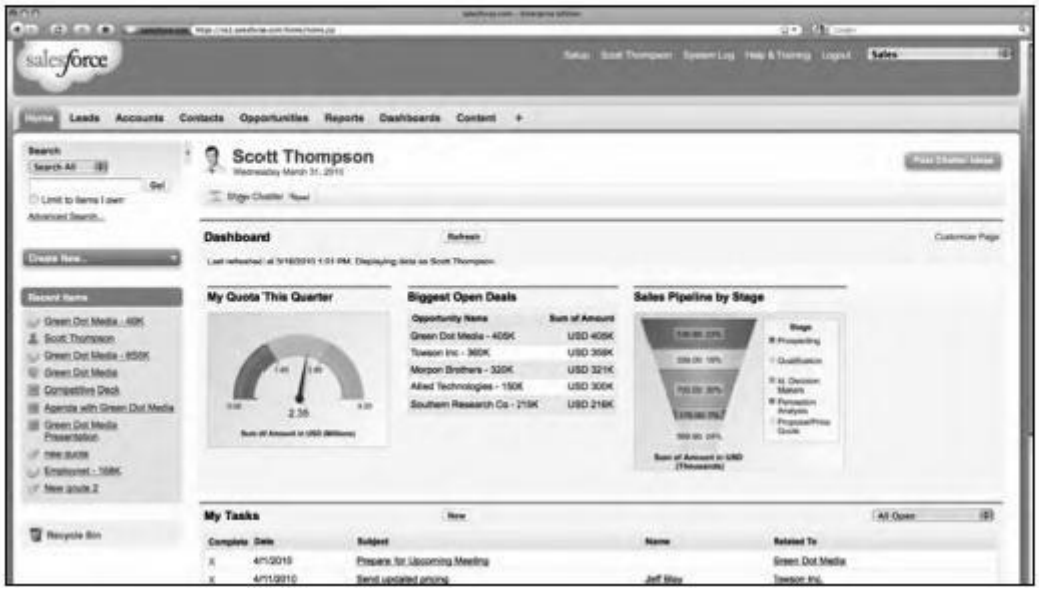
- Accounts and Contact
- Analytics and Forecasting
- Approvals and Workflow
- Chatter (Instant Messaging/Collaboration)
- Content Library
- E-mail and Productivity
- Jigsaw Business Data
- Marketing and Leads
- Opportunities and Quotes
- Partner Relationship
- Sales, Service and Support

Which tabs you see, and how capable each hosted application is, depends on the level of service you purchase from Salesforce.com, as well as the particular type of bundle you buy. Salesforce.com tailors its SaaS for individual industries.

Shown in Figure 7.1 is a Salesforce.com portal with the multi-tabbed interface exposing the different applications.

FIGURE 7.1

In Salesforce.com, each tab is an application, and data is shared. Shown here is a dashboard view.



As Salesforce.com developed its SaaS production, it became obvious that many customers wanted to extend their Salesforce.com applications beyond what an SaaS offering would allow. Salesforce.com developed a PaaS platform known as Force.com, which allows developers to create applications that could be added to Salesforce.com's offerings and hosted on Salesforce.com's infrastructure.

Force.com uses a Java-based programming language called Apex for its application building, and it has an interface builder called Visualforce that allows a developer to create interfaces using HTML, Flex, and AJAX. Visualforce uses an XML-type language in its visual interface builder. Using the Force.com platform, more than 1,000 applications have been created and are offered for sale on Salesforce.com's AppExchange, which has greatly enhanced its PaaS offerings. These applications can show up as customizable tabs for different functions in customer applications or as a set of S-controls that are JavaScript widgets. Because Salesforce.com is browser-based, it is platform-independent. However, the company has extended its audience to mobile devices, such as the Android, Blackberry, iPhone, and Windows Mobile Devices. It also has a server product that supports Salesforce.com applications in-house called the Resin Application Server.

Force.com has been a major hit and has served as the model from many of the PaaS systems of today. The company Salesforce.com is a recognized thought leader in the field of cloud computing. It is a $1.3 billion company as of 2009, with over 2 million subscribers.

Figure 7.2 shows the Force.com platform page at Salesforce.com.

FIGURE 7.2

Force.com's Web site (`http://www.salesforce.com/platform/`) leads to a set of developer tools as well as a gallery of sites built on this PaaS.

Using PaaS Application Frameworks

Application frameworks provide a means for creating SaaS hosted applications using a unified development environment or an *integrated development environment* (IDE). PaaS IDEs run the gamut from a tool that requires a dedicated programming staff to create and run to point-and-click graphical interfaces that any knowledgeable computer user can navigate and create something useful with.

In selecting the six different examples of Web sites and application building PaaS systems, a full range of user experience is considered. Many Web sites are based on the notion of information management and organization; they are referred to as *content management systems* (CMS). A database is a content management system, but the notion of a Web site as a CMS adds a number of special features to the concept that includes rich user interaction, multiple data sources, and extensive customization and extensibility. The Drupal CMS was chosen as an example of this type of PaaS because it is so extensively used and has broad industry impact, and it is a full-strength developer tool.

Whereas Drupal is used in major Web sites and organizes vast amounts of information, the site Squarespace.com was chosen to illustrate a point-and-click CMS system aimed at supporting individuals, small businesses, and other small organizations. Squarespace is often associated with blogging tools (as is Drupal), but it is more than that. Squarespace works with photos, imports information from other social tools, and allows very attractive Web sites to be created by average users.

Drupal Frameworks

Drupal (http://drupal.org/) is a content management system (CMS) that is used as the backend to a large number of Web sites worldwide. The software is an open-source project that was created in the PHP programming language. Drupal is really a programming environment for managing content, and it has elements of blogging and collaboration software as part of its distribution.

Drupal is offered to the public under the GNU General Public License version 2 and is used by many prominent Web sites. The Drupal core is the standard distribution, with the current version being 6.19; version 7.0 is in preview. Drupal is in this section because it is a highly extensible way to create Web sites with rich features.

Druplas has a large developer community that has created nearly 6,000 third-party add-ons called *contrib modules.* Several thousand Drupal developers worldwide come together twice a year at the DrupalCon convention. It's a vibrant community of users and developers.

The number of Web sites that use Drupal is really quite remarkable, and many of them are very well known. Drupal is very popular with government agencies and with media companies, but its reach extends into nearly any industry, organization, and business type you can think of. Some of these sites are beautifully constructed. A short list of sites includes att.com, data.gov.uk, government.fr, intel.com, lucasfilms.com, mattel.com, thenation.com, whitehouse.gov, and ubuntu.com.

Drupal has a gallery of screenshots of sites and features on its Web site, but for a better look at some of the more attractive sites, go to the Showcase of Popular Web sites Developed Using Drupal CMS (http://artatm.com/2010/02/showcase-of-popular-website-developedusing- drupal/), shown in Figure 7.3.

You find Drupal applications running on any Web server that can run PHP 4.4.0 and later. The most common deployments are on Apache, but you also can find Drupal on Microsoft IIS and other Unix Web servers. To store content, Drupal must be used with a database. Because LAMP installations are a standard Web deployment platform, the database most often used is MySQL. Other SQL databases work equally well.

The Drupal core by itself contains a number of modules that provide for the following:

- Auto-updates
- Blogs, forums, polls, and RSS feeds
- Multiple site management
- OpenID authentication
- Performance optimization through caching and throttling
- Search
- User interface creation tools
- User-level access controls and profiles
- Themes
- Traffic management
- Workflow control with events and triggers

FIGURE 7.3

Artatm.com has a gallery of some of the more attractive and well-known sites built with Drupal.

Drupal is modular and exposes its functionality through a set of published APIs. The contrib modules can be added to Drupal to replace other modules, enhance capabilities, or provide entirely new features. Third-party modules include messaging systems, visual editors, a content construction kit (CCK) for database schema extension, views, and panels. CCK Fields API is in the latest version of Drupal, version 7.0.

Drupal is reputed to be somewhat difficult to learn, and new versions often break old features. It is much more widely used than its competitor Joomla! (http://www.joomla.org/), and Drupal seems to have better performance than Joomla! as well. Another open source competitor in the content management space is eZ Publish (http://ez.no/).

Eccentex AppBase 3.0

Eccentex is a Culver City, California, company founded in 2005 that has a PaaS development platform for Web applications based on SOA component architecture to create what it calls Cloudware applications using its AppBase architecture. Figure 7.4 shows the AppBase platform page.

AppBase includes a set of different tools for building these applications, including the following:

- **Business Objects Build:** This object database has the ability to create rich data objects and create relationships between them.

- **Presentation Builder:** This user interface (UI) builder allows you to drag and drop visual controls for creating Web forms and data entry screens and to include the logic necessary to automate what the user sees.
- **Business Process Designer:** This tool is used to create business logic for your application. With it, you can manage workflow, integrate modules, create rules, and validate data.
- **Dashboard Designer:** This instrumentation tool displays the real-time parameters of your application in a visual form.
- **Report Builder:** This output design tool lets you sort, aggregate, display, and format report information based on the data in your application.
- **Security Roles Management:** This allows you to assign access rights to different objects in the system, to data sets, fields, desktop tabs, and reports. Security roles can be assigned in groups without users, and users can be added later as the application is deployed.

FIGURE 7.4

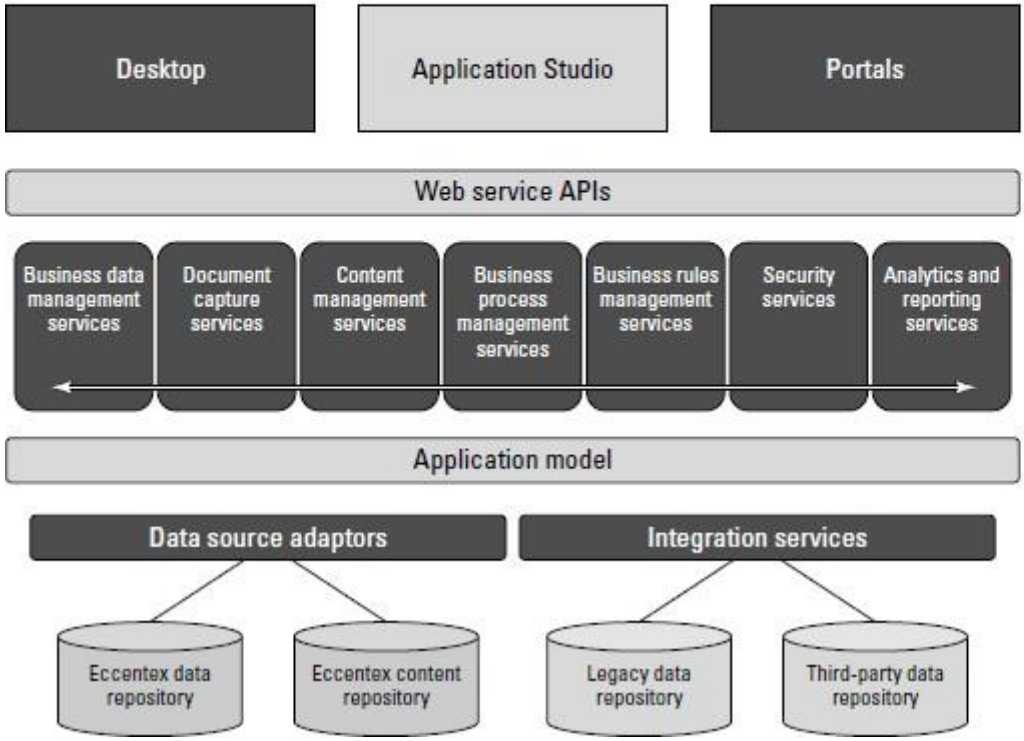
The Eccentex AppBase (`http://www.eccentex.com/platform/platform.html`) PaaS application delivery platform creates SOA applications that work on several different IaaS vendors.

Applications that you create are deployed with the AppBase Application Revision Management console. The applications you create in AppBase, according to the company, may be integrated with Amazon S3 Web Services (storage), Google AppEngine (PaaS), Microsoft Windows Azure (PaaS), Facebook, and Twitter.

Figure 7.5 shows the AppBase architecture with the various tools identified. You can view a set of screenshots that illustrate the different tools and some features in the build process at `http://www.eccentex.com/platform/screenshots.html`.

FIGURE 7.5

AppBase's architecture with the different tools and modules shown



LongJump

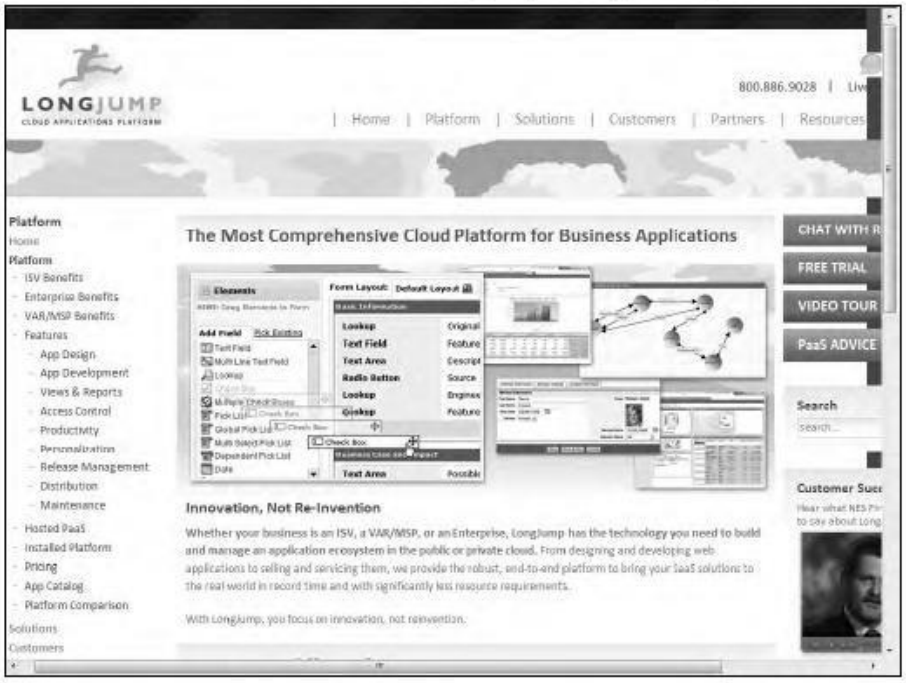
LongJump (http://www.longjump.com/) is a Sunnyvale, California, company hosting service created in 2003 with a PaaS application development suite. Its development environment is based on Java and uses REST/SOAP APIs. Figure 7.6 shows the LongJump platform page.

LongJump creates browser-based Web applications that are database-enabled. Like other products mentioned in this section, LongJump comes with an Object Model Viewer, forms, reports, layout tools, dashboards, and site management tools. Access control is based on role- and rule-based access, and it allows for data-sharing between teams and between tenants. LongJump comes with a security policy engine that has user and group privileges, authentication, IP range blocking, SSO, and LDAP interoperability. Applications are packaged using a packaging framework that can support a catalog system, XML package file descriptions, and a distribution engine.

LongJump extends Java and uses a Model-View-Controller architecture (MVC) for its framework in the Developer Suite. The platform uses Java Server Pages (JSP), Java, and JavaScript for its various components and its actions with objects built with Java classes. Objects created in custom classes are referenced using POJO (Plain Old Java Object).

FIGURE 7.6

LongJump's PaaS (http://www.longjump.com/index.php?option=com_content&view=article&id=8&Itemid=57) is based on standard Java/JavaScript, SOAP, and REST.



Localization is supported using a module called the Translation Workbench that includes specified labels, errors, text, controls, and messaging text files (and header files) that allow them to be modified by a translation service to support additional languages. The development environment supports the Eclipse (http://www.eclipse.org/) plug-in for creating widgets using Java standard edition.

Squarespace

Squarespace (http://www.squarespace.com/), shown in Figure 7.7, is an example of a next-generation Web site builder and deployment tool that has elements of a PaaS development environment. The applications are built using visual tools and deployed on hosted infrastructure.

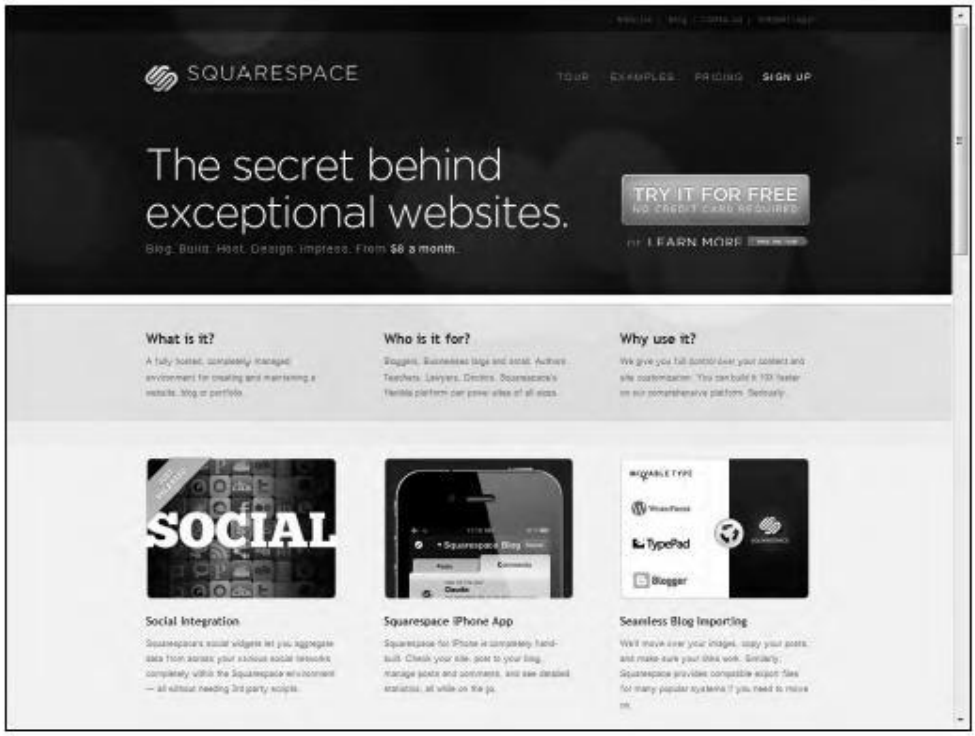
Squarespace presents itself, among other things, as:

- A blogging tool
- A social media integration tool
- A photo gallery
- A form builder and data collector
- An item list manager
- A traffic and site management and analysis tool

The platform has more than 20 core modules that you can add to your Web site. Squarespace sites can be managed on the company's iPhone app.

FIGURE 7.7

Squarespace lets you create beautiful hosted Web sites with a variety of capabilities with visual tools alone.



With Squarespace, users have created some very visually beautiful sites. Users tend to fall into these categories: personal Web sites, portfolios, and business brand identification. Although Squarespace positions itself as a competitor to blogging sites such as Wordpress (http:// wordpress.org/), Tumblr (http://www.tumblr.com/), Posterous (https:// posterous.com/), and other sites of their ilk, the site borders on a full content management system with a variety of useful and eclectic features.

WaveMaker

WaveMaker (http://www.wavemaker.com/) is a visual rapid application development environment for creating Java-based Web and cloud Ajax applications. The software is open-source and offered under the Apache license. WaveMaker is a WYSIWYG (What You See is What You Get) drag-and-drop environment that runs inside a browser. The metaphor used to build applications is described as the Model-View-Controller system of application architecture. In this regard, WaveMaker has some similarities to PowerBuilder (http://www.sybase.com/products/internetappdevttools/powerbuilder).

Figure 7.8 shows the WaveMaker home page. A gallery of features is accessible from that page.

WaveMaker is a framework that creates applications that can interoperate with other Java frameworks and LDAP systems, including the following:

- Dojo Toolkit 1.0 (http://dojotoolkit.org/), a JavaScript library or toolbox
- LDAP directories
- Microsoft Active Directory
- POJO (Plain Old Java Object)

- Spring Framework (http://www.springsource.org/), an open-source application framework for Java that now also includes ACEGI

FIGURE 7.8

WaveMaker is a visual development environment for creating Java-based cloud applications.

The visual builder tool is called Visual Ajax Studio, and the development server is called the WaveMaker Rapid Deployment Server for Java applications. When you develop within the Visual Ajax Studio, a feature called LiveLayout allows you to create applications while viewing live data. The data schema is prepared within a part of the tool called LiveForms. Mashups can be created using the Mashup Tool, which integrates applications using Java Services, SOAP, REST, and RSS to access databases.

Applications developed in WaveMaker run on standard Java servers such as Tomcat, DojoToolkit, Spring, and Hibernate. A 4GL version of WaveMaker also runs on Amazon EC2, and the development environment can be loaded on an EC2 instance as one of its machine images.

Wolf Frameworks

Many application frameworks like Google AppEngine and the Windows Azure Platform are tied to the platform on which they run. You can't build an AppEngine application and port it to Windows Azure without completely rewriting the application. There isn't any particular necessity to build an application framework in this way, but it suits the purpose of these particular vendors: for Google to have a universe

of Google applications that build on the Google infrastructure, and for Microsoft to provide another platform on which to extend .NET Framework applications for their developers.

If you are building an application on top of an IaaS vendor such as AWS, GoGrid, or RackSpace, what you really want are application development frameworks that are open, standards-based, and portable. Wolf Frameworks is an example of a PaaS vendor offering a platform on which you can build an SaaS solution that is open and cross-platform. Wolf Frameworks (http://www.wolfframeworks.com/) was founded in Bangalore, India, in 2006, and it has offices in the United States.

Wolf Frameworks is based on the three core Windows SOA standard technologies of cloud computing:

- AJAX, asynchronous Java
- XML
- .NET Framework

Wolf Frameworks uses a C# engine and supports both Microsoft SQL Server and MySQL database. Applications that you build in Wolf are 100-percent browser-based and support mashable and multisource overlaid content. Figure 7.9 shows the Wolf Frameworks home page.

The Wolf platform is interesting in a number of ways. Wolf has architected its platform so applications can be built without the need to write technical code. It also allows application data to be written to the client's database server of choice, and data can be imported or exported from a variety of data formats. In Wolf, you can view your Business Design of the software application that you build in XML.

Wolf supports forms, search, business logic and rules, charts, reports, dashboards, and both custom and external Web pages. After you create entities and assign their properties, you create business rules with a rules designer. You can automate tasks via business rules. There are tools for building the various site features such as forms, reports, dashboards, and so on. Connections to the datacenter are over a 128-bit encrypted SSL connection, with authentication, access control, and a transaction history and audit trail. Security to multiple modules can be made available through a Single Sign-On (SSO) mechanism.

In Wolf, the data and transaction management conforms to the business rules you create. The data and UI rendering are separate systems. Thus, you can change the UI as you need to without affecting your stored data. Wolf lets you work with Adobe Flash or Flex or with Microsoft Silverlight. You can also use third-party on- or off-premises applications with your SaaS application.

A backup system lets you back up data with a single click. Figure 7.10 shows the WOLF platform architecture. These features enable Wolf developers to create a classic multitenant SOA application without the need for high-level developer skills. These applications are interoperable, portable from one Windows virtual machine to another, and support embedded business applications. You can store your Wolf applications on a private server or in the Wolf cloud.

FIGURE 7.9

Wolf Frameworks offers an open platform based on SOA standards for building portable SaaS solutions.

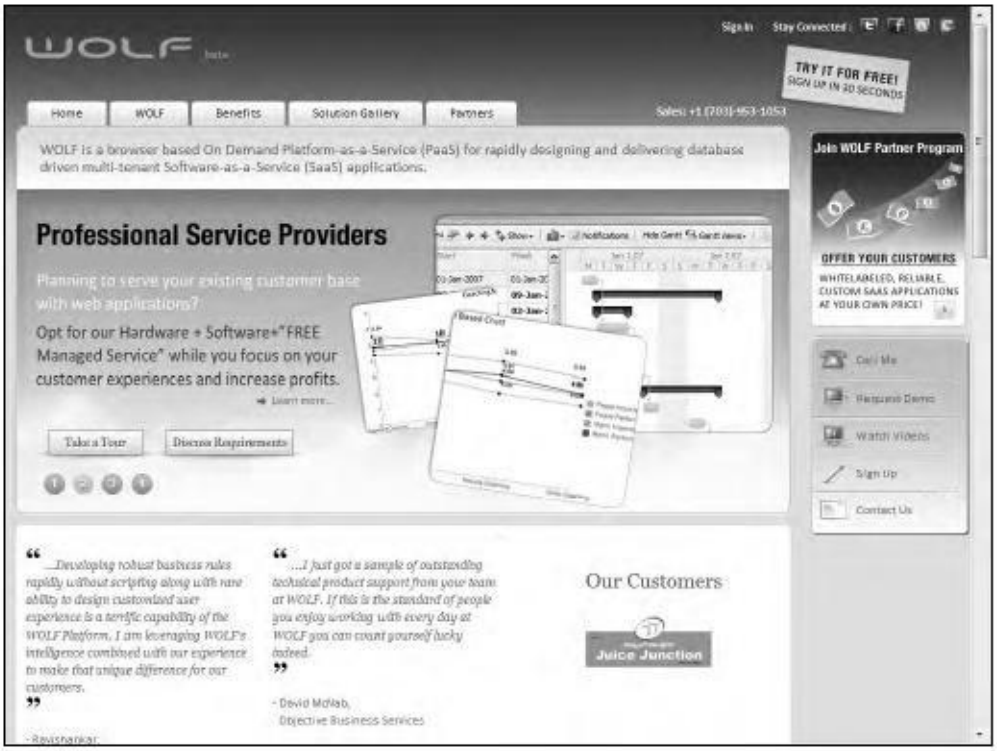
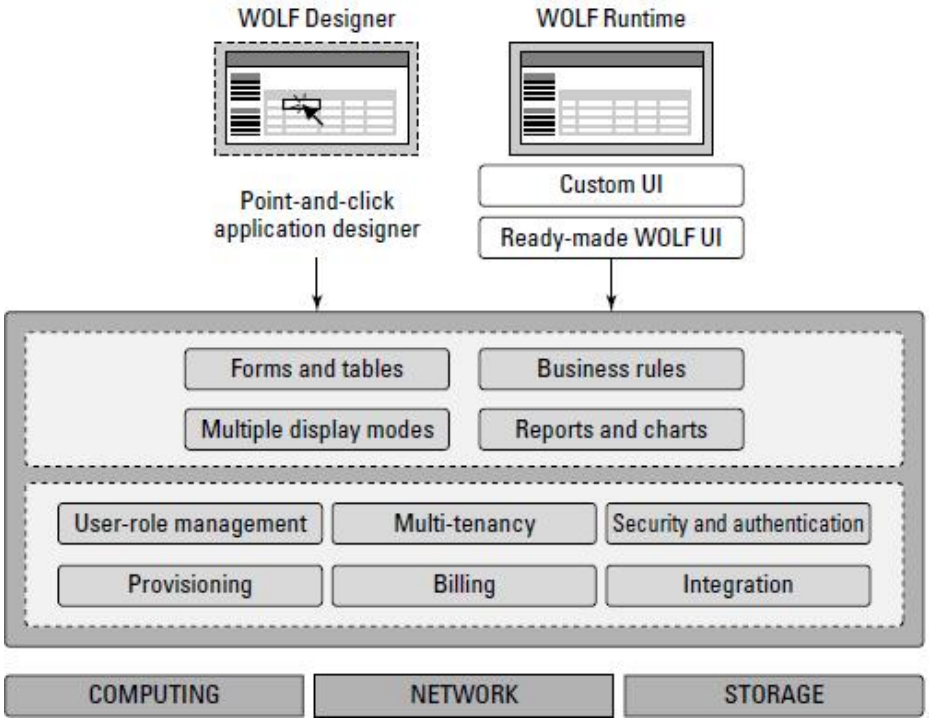


FIGURE 7.10

The Wolf platform architecture; source: `http://www.wolfframeworks.com/platform.asp`.



Defining Software as a Service (SaaS)

The most complete cloud computing service model is one in which the computing hardware and software, as well as the solution itself, are provided by a vendor as a complete service offering. It is referred to as the Software as a Service (SaaS) model. SaaS provides the complete infrastructure, software, and solution stack as the service offering. A good way to think about SaaS is that it is the cloud-based equivalent of shrink-wrapped software.

Software as a Service (SaaS) may be succinctly described as software that is deployed on a hosted service and can be accessed globally over the Internet, most often in a browser. With the exception of the user interaction with the software, all other aspects of the service are abstracted away.

Every computer user is familiar with SaaS systems, which are either replacements or substitutes for locally installed software. Examples of SaaS software for end-users are Google Gmail and Calendar, QuickBooks online, Zoho Office Suite, and others that are equally well known. SaaS applications come in all shapes and sizes, and include custom software such as billing and invoicing systems, Customer Relationship Management (CRM) applications, Help Desk applications, Human Resource (HR) solutions, as well as myriad online versions of familiar applications.

Many people believe that SaaS software is not customizable, and in many SaaS applications this is indeed the case. For user-centric applications such as an office suite, that is mostly true; those suites allow you to set only options or preferences. However, many other SaaS solutions expose Application Programming Interfaces (API) to developers to allow them to create custom composite applications. These APIs may alter the security model used, the data schema, workflow characteristics, and other fundamental features of the service's expression as experienced by the user. Examples of an SaaS platform with an exposed API are Salesforce.com and Quicken.com. So SaaS does not necessarily mean that the software is static or monolith.

2.7.1 SaaS characteristics:

All Software as a Service (SaaS) applications share the following characteristics:

1. The software is available over the Internet globally through a browser on demand.

2. The typical license is subscription-based or usage-based and is billed on a recurring basis. In a small number of cases a flat fee may be changed, often coupled with a maintenance fee. Table 4.1 shows how different licensing models compare.

3. The software and the service are monitored and maintained by the vendor, regardless of where all the different software components are running. There may be executable client-side code, but the user isn't responsible for maintaining that code or its interaction with the service.

4. Reduced distribution and maintenance costs and minimal end-user system costs generally make SaaS applications cheaper to use than their shrink-wrapped versions.

5. Such applications feature automated upgrades, updates, and patch management and much faster rollout of changes.

6. SaaS applications often have a much lower barrier to entry than their locally installed competitors, a known recurring cost, and they scale on demand (a property of cloud computing in general).

7. All users have the same version of the software so each user's software is compatible with another's.

8. SaaS supports multiple users and provides a shared data model through a single-instance, multi-tenancy model.

The alternative of software virtualization of individual instances also exists, but is less common.

TABLE 4.1

Shrink-Wrapped versus SaaS Licensing

	Shrink-Wrapped Software	Hybrid Model	SaaS
Licensing	Owned	Subscription (flat fee)	Metered subscription
Location	Locally installed	Available through an application	Cloud based
Management	Local IT staff	Application Service Provider (ASP)	Cloud vendor through a Service Level Agreement (SLA)

Defining Identity as a Service (IDaaS)

The establishment and proof of an identity is a central network function. An identity service is one that stores the information associated with a digital entity in a form that can be queried and managed for use in electronic transactions. Identity services have as their core functions: a data store, a query engine, and a policy engine that maintains data integrity.

Distributed transaction systems such as internetworks or cloud computing systems magnify the difficulties faced by identity management systems by exposing a much larger attack surface to an intruder than a private network does. Whether it is network traffic protection, privileged resource access, or some other defined right or privilege, the validated authorization of an object based on its identity is the central tenet of secure network design. In this regard, establishing identity may be seen as the key to obtaining trust and to anything that an object or entity wants to claim ownership of.

Services that provide digital identity management as a service have been part of internetworked systems from Day One. Like so many concepts in cloud computing, IDentity as a Service is a FLAVor (Four Letter Acronym) of the month, applied to services that already exist. The Domain Name Service can run on a private network, but is at the heart of the Internet as a service that provides identity authorization and lookup. The name servers that run the various Internet domains (.COM, .ORG, .EDU, .MIL, .TV, .RU, and so on) *are* IDaaS servers. DNS establishes the identity of a domain as belonging to a set of assigned addresses, associated with an owner and that owner's information, and so forth. If the identification is the assigned IP number, the other properties are its metadata.

What is an identity?

An identity is a set of characteristics or traits that make something recognizable or known. In computer network systems, it is one's digital identity that most concerns us. A digital identity is those attributes and metadata of an object along with a set of relationships with other objects that makes an object identifiable. Not all objects are unique, but by definition a digital identity must be unique, if only trivially so, through the assignment of a unique identification attribute. An identity must therefore have a context in which it exists.

This description of an identity as an object with attributes and relationships is one that programmer's would recognize. Databases store information and relationships in tables, rows, and columns ,and the identity of information stored in this way conforms to the notion of an entity and a relationship—or lternatively under the notion of an object role model (ORM)—and database architects are always wrestling with the best way of reducing their data set to a basic set of identities. You can extend this notion to the idea of an identity having a profile and profiling services such as Facebook as being an extension of the notion of Identity as a Service in cloud computing.

An identity can belong to a person and may include the following:

- Things you are: Biological characteristics such as age, race, gender, appearance, and so forth
- Things you know: Biography, personal data such as social security numbers, PINs, where you went to school, and so on
- Things you have: A pattern of blood vessels in your eye, your fingerprints, a bank account you can access, a security key you were given, objects and possessions, and more
- Things you relate to: Your family and friends, a software license, beliefs and values, activities and endeavors, personal selections and choices, habits and practices, an iGoogle account, and more

To establish your identity on a network, you might be asked to provide a name and password, which is called a single-factor authentication method. More secure authentication requires the use of at least two-factor authentication; for example, not only name and password (things you know) but also a transient token number provided by a hardware key (something you have). To get to multifactor authentication, you might have a system that examines a biometric factor such as a fingerprint or retinal blood vessel pattern—both of which are essentially unique things you are. Multifactor authentication requires the outside use of a network security or trust service, and it is in the deployment of trust services that our first and most common IDaaS applications are employed in the cloud.

Of course, many things have digital identities. User and machine accounts, devices, and other objects establish their identities in a number of ways. For user and machine accounts, identities are created and stored in domain security databases that are the basis for any network domain, in directory services, and in data stores in federated systems. Network interfaces are identified uniquely by Media Access Control (MAC) addresses, which alternatively are referred to as Ethernet Hardware Addresses (EHAs). It is the assignment of a network identity to a specific MAC address that allows systems to be found on networks.

The manner in which Microsoft validates your installation of Windows and Office is called Windows Product Activation and creates an identification index or profile of your system, which is instructive. During activation, the following unique data items are retrieved:

- A 25-character software product key and product ID
- The uniquely assigned Global Unique Identifier or GUID
- PC manufacturer
- CPU type and serial number
- BIOS checksum
- Network adapter and its MAC address
- Display adapter
- SCSCI and IDE adapters
- RAM amount
- Hard drive and volume serial number
- Optical drive
- Region and language settings and user locale

From this information, a code is calculated, checked, and entered into the registration database. Each of these uniquely identified hardware attributes is assigned a weighting factor such that an overall sum may be calculated. If you change enough factors—NIC and CPU, display adapter, RAM amount, and hard drive—you trigger a request for a reactivation based on system changes. This activation profile is also required when you register for the Windows Genuine Advantage program. Windows Product Activation and Windows Genuine Advantage are cloud computing applications, albeit proprietary ones. Whether people consider these applications to be services is a point of contention.

Networked identity service classes

To validate Web sites, transactions, transaction participants, clients, and network services—various forms of identity services—have been deployed on networks. Ticket or token providing services, certificate servers, and other trust mechanisms all provide identity services that can be pushed out of private networks and into the cloud.

Identity protection is one of the more expensive and complex areas of network computing. If you think about it, requests for information on identity by personnel such as HR, managers, and others; by systems and resources for access requests; as identification for network traffic; and the myriad other requirements mean that a significant percentage of all network traffic is supporting an identification service. Literally hundreds of messages on a network every minute are checking identity, and every Ethernet packet contains header fields that are used to identify the information it contains.

As systems become even more specialized, it has become increasingly difficult to find the security experts needed to run an ID service. So Identity as a Service or the related hosted (managed) identity services may be the most valuable and cost effective distributed service types you can subscribe to.

Identity as a Service (IDaaS) may include any of the following:

- Authentication services (identity verification)
- Directory services
- Federated identity
- Identity governance
- Identity and profile management
- Policies, roles, and enforcement
- Provisioning (external policy administration)
- Registration
- Risk and event monitoring, including audits
- Single sign-on services (pass-through authentication)

The sharing of any or all of these attributes over a network may be the subject of different government regulations and in many cases must be protected so that only justifiable parties may have access to the minimal amount that may be disclosed. This level of access defines what may be called an identity relationship.

Identity system codes of conduct

Certain codes of conduct must be observed legally, and if not legally at the moment, then certainly on a moral basis. Cloud computing services that don't observe these codes do so at their peril. In working with IDaaS software, evaluate IDaaS applications on the following basis:

- **User control for consent:** Users control their identity and must consent to the use of their information.
- **Minimal Disclosure:** The minimal amount of information should be disclosed for an intended use.
- **Justifiable access:** Only parties who have a justified use of the information contained in a digital identity and have a trusted identity relationship with the owner of the information may be given access to that information.
- **Directional Exposure:** An ID system must support bidirectional identification for a public entity so that it is discoverable and a unidirectional identifier for private entities, thus protecting the private ID.

- **Interoperability:** A cloud computing ID system must interoperate with other identity services from other identity providers.
- **Unambiguous human identification:** An IDaaS application must provide an unambiguous mechanism for allowing a human to interact with a system while protecting that user against an identity attack.
- **Consistency of Service:** An IDaaS service must be simple to use, consistent across all its uses, and able to operate in different contexts using different technologies.

IDaaS interoperability

Identity as a Service provides an easy mechanism for integrating identity services into individual applications with minimal development effort, by allowing the identification logic and storage of an identity's attributes to be maintained externally. IDaaS applications may be separated from other distributed security systems by their compliance with SOA standards (as described in Chapter 13, "Understanding Service Oriented Architecture"), particularly if you want to have these services interoperate and be federated.

Therefore, cloud computing IDaaS applications must rely on a set of developing industry standards to provide interoperability. The following are among the more important of these services:

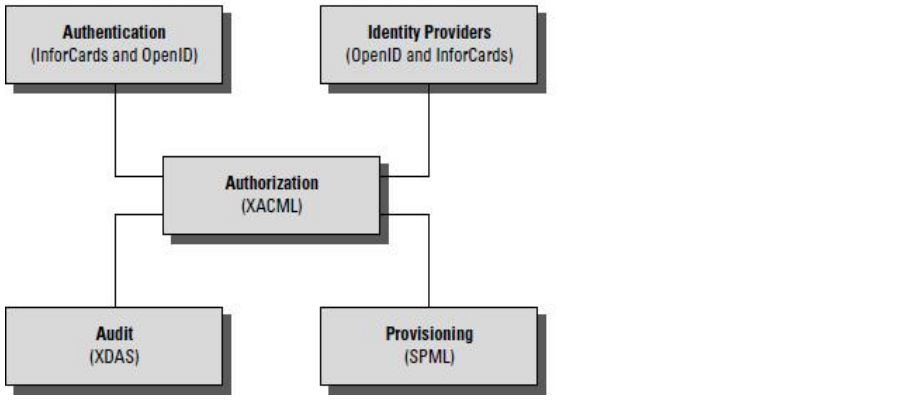
- **User centric authentication (usually in the form of information cards):** The OpenID and CardSpace specifications support this type of data object.
- **The XACML Policy Language:** This is a general-purpose authorization policy language that allows a distributed ID system to write and enforce custom policy expressions. XACML can work with SAML; when SAML presents a request for ID authorization, XACML checks the ID request against its policies and either allows or denies the request.
- **The SPML Provisioning Language:** This is an XML request/response language that is used to integrate and interoperate service provisioning requests. SPML is a standard of OASIS's Provision Services Technical Committee (PSTC) that conforms to the SOA architecture.
- **The XDAS Audit System:** The Distributed Audit Service provides accountability for users accessing a system, and the detection of security policy violations when attempts are made to access the system by unauthorized users or by users accessing the system in an unauthorized way.

The Identity Governance Framework (IGF) is a standards initiative of the Liberty Alliance (http://www.projectliberty.org/) that is concerned with the exchange and control of identity information using standards such as WS-Trust, ID-WSF, SAML, and LDAP directory services. The Liberty Alliance was established by an industry group in 2001 with the purpose of promoting open identity interchanges through policy standards that applications can use to enforce privacy as well as to allow privacy auditing. In 2009, this group released its Client Attribute Requirements Markup Language (CARML) and a set of IGF Privacy Constraints that forms the basis of the open source project called Aristotle

(http://www.openliberty.org/wiki/index.php/ ProjectAris), which has as its goal the creation of an API for identity interchange.

FIGURE 4.6

Open standards that support an IDaaS infrastructure for cloud computing



User authentication

OpenID is a developing industry standard for authenticating "end users" by storing their digital identity in a common format. When an identity is created in an OpenID system, that information is stored in the system of any OpenID service provider and translated into a unique identifier.Identifiers take the form of a Uniform Resource Locator (URL) or as an Extensible Resource Identifier (XRI) that is authenticated by that OpenID service provider. Any software application that complies with the standard accepts an OpenID that is authenticated by a trusted provider.

A very impressive group of cloud computing vendors serve as identity providers (or OpenID providers), including AOL, Facebook, Google, IBM, Microsoft, MySpace, Orange, PayPal, VeriSign, LiveJournal, Ustream, Yahoo!, and others.

The OpenID standard applies to the unique identity of the URL; it is up to the service provider to store the information and specify the forms of authentication required to successfully log onto the system. Thus an OpenID authorization can include not only passwords, but smart cards, hardware keys, tokens, and biometrics as well. OpenID is supported by the OpenID Foundation (http:// openid.net/foundation/), a not-for-profit organization that promotes the technology.

These are samples of trusted providers and their URL formats:

- **Blogger:** <*username*>.blogger.com or <*blogid*>.blogspot.com
- **MySpace:** myspace.com/<*username*>
- **Google:** https://www.google.com/accounts/o8/id
- **Google Profile:** google.com/profiles/<*username*>
- **Microsoft:** accountservices.passport.net/

- **MyOpenID:** <*username*>.myopenid.com
- **Orange:** openid.orange.fr/username or simply orange.fr/
- **Verisign:** <*username*>.pip.verisinglabs.com
- **WordPress:** <*username*>.wordpress.com
- **Yahoo!:** openid.yahoo.com

After you have logged onto a trusted provider, that logon may provide you access to other Web sites that support OpenID. When you request access to a site through your browser (or another application that is referred to as a user-agent), that site serves as the "relying party" and requests of the server or server-agent that it verify the end-user's identifier. You won't need to log onto these other Web sites, if your OpenID is provided. Most trusted providers require that you indicate which Web sites you want to share your OpenID identifier with and the information is submitted automatically to the next site.

CardSpace is a Microsoft software client that is part of the company's Identity Metasystem and built into the Web Services Protocol Stack. This stack is built on the OASIS standards (WS-Trust, WS-Security, WS-SecurityPolicy, and WS-MetadataExchange), so any application that conforms with the OASIS WS- standards can interoperate with CardSpace. CardSpace was introduced with .NET Frameworks 3.0 and can be installed on Windows XP, Server 2003, and later. It is installed by default on Windows Vista and Windows 7.

CardSpace offers another way of authenticating users in the cloud. An Information Card may be requested with an HTML <OBJECT> tag, and the trusted Identity Provider then creates an encrypted and digitally signed token using the Security Token Service (STS) that is part of a WS-Trust request/ reply mechanism. CardSpace may be seen as an alternative mechanism to the use of OpenID and SAML and is used to sign into those services as well as Windows Live ID accounts.

Defining Compliance as a Service (CaaS)

Cloud computing by its very nature spans different jurisdictions. The laws of the country of a request's origin may not match the laws of the country where the request is processed, and it's possible that neither location's laws match the laws of the country where the service is provided. Compliance is much more than simply providing an anonymous service token to an identity so they can obtain access to a resource. Compliance is a complex issue that requires considerable expertise.

While Compliance as a Service (CaaS) appears in discussions, few examples of this kind of service exist as a general product for a cloud computing architecture. A Compliance as a Service application would need to serve as a trusted third party, because this is a man-in-the-middle type of service. CaaS may need to be architected as its own layer of a SOA architecture in order to be trusted. A CaaS would need to be able to manage cloud relationships, understand security policies and procedures, know how to handle information and administer privacy, be aware of geography, provide an incidence response, archive, and allow for the system to be queried, all to a level that can be captured in a Service Level Agreement. That's a tall order, but CaaS has the potential to be a great value-added service.

In order to implement CaaS, some companies are organizing what might be referred to as "vertical clouds," clouds that specialize in a vertical market. Examples of vertical clouds that advertise CaaS capabilities include the following:

- **athenahealth** (http://www.athenahealth.com/) for the medical industry
- **bankserv** (http://www.bankserv.com/) for the banking industry
- **ClearPoint PCI** Compliance-as-a-Service for merchant transactions under the Payment Card Industry Data Security Standard
- **FedCloud** (http://www.fedcloud.com/) for government
- **Rackserve PCI** Compliant Cloud (http://www.rackspace.com/; another PCI CaaS service)

It's much easier to envisage a CaaS system built inside a private cloud where the data is under the control of a single entity, thus ensuring that the data is under that entity's secure control and that transactions can be audited. Indeed, most of the cloud computing compliance systems to date have been built using private clouds. It is easy to see how CaaS could be an incredibly valuable service. A well-implemented CaaS service could measure the risks involved in servicing compliance and ensure or indemnify customers against that risk. CaaS could be brought to bear as a mechanism to guarantee that an e-mail conformed to certain standards, something that could be a new electronic service of a network of national postal systems—and something that could help bring an end to the scourge of spam.

UNIT – III

Abstraction and Virtualization: Introduction to Virtualization Technologies, Load Balancing and Virtualization, Understanding Hyper visors, Understanding Machine Imaging, Porting Applications, Virtual Machines Provisioning and Manageability Virtual Machine Migration Services, Virtual Machine Provisioning and Migration in Action, Provisioning in the Cloud Context.

3.1 Using Virtualization Technologies

The dictionary includes many definitions for the word "cloud." A cloud can be a mass of water droplets, gloom, an obscure area, or a mass of similar particles such as dust or smoke. When it comes to cloud computing, the definition that best fits the context is "a collection of objects that are grouped together." It is that act of grouping or creating a resource pool that is what succinctly differentiates cloud computing from all other types of networked systems.

Not all cloud computing applications combine their resources into pools that can be assigned on demand to users, but the vast majority of cloud-based systems do. The benefits of pooling resources to allocate them on demand are so compelling as to make the adoption of these technologies a priority. Without resource pooling, it is impossible to attain efficient utilization, provide reasonable costs to users, and proactively react to demand. In this chapter, you learn about the technologies that abstract physical resources such as processors, memory, disk, and network capacity into virtual resources.

When you use cloud computing, you are accessing pooled resources using a technique called virtualization. Virtualization assigns a logical name for a physical resource and then provides a pointer to that physical resource when a request is made. Virtualization provides a means to manage resources efficiently because the mapping of virtual resources to physical resources can be both dynamic and facile. Virtualization is dynamic in that the mapping can be assigned based on rapidly changing conditions, and it is facile because changes to a mapping assignment can be nearly instantaneous.

These are among the different types of virtualization that are characteristic of cloud computing:

- **Access:** A client can request access to a cloud service from any location.
- **Application:** A cloud has multiple application instances and directs requests to an instance based on conditions.
- **CPU:** Computers can be partitioned into a set of virtual machines with each machine being assigned a workload. Alternatively, systems can be virtualized through load-balancing technologies.
- **Storage:** Data is stored across storage devices and often replicated for redundancy.

To enable these characteristics, resources must be highly configurable and flexible. You can define the features in software and hardware that enable this flexibility as conforming to one or more of the following mobility patterns:

- **P2V:** Physical to Virtual
- **V2V:** Virtual to Virtual
- **V2P:** Virtual to Physical
- **P2P:** Physical to Physical
- **D2C:** Datacenter to Cloud
- **C2C:** Cloud to Cloud
- **C2D:** Cloud to Datacenter
- **D2D:** Datacenter to Datacenter

The techniques used to achieve these different types of virtualization are the subject of this chapter. According to Gartner ("Server Virtualization: One Path that Leads to Cloud Computing," by Thomas J. Bittman, 10/29/2009, Research Note G00171730), virtualization is a key enabler of the first four of five key attributes of cloud computing:

- **Service-based:** A service-based architecture is where clients are abstracted from service providers through service interfaces.
- **Scalable and elastic:** Services can be altered to affect capacity and performance on demand.
- **Shared services:** Resources are pooled in order to create greater efficiencies.
- **Metered usage:** Services are billed on a usage basis.
- **Internet delivery:** The services provided by cloud computing are based on Internet protocols and formats.

3.2 Load Balancing and Virtualization

One characteristic of cloud computing is virtualized network access to a service. No matter where you access the service, you are directed to the available resources. The technology used to distribute service requests to resources is referred to as *load balancing.* Load balancing can be implemented in hardware, as is the case with F5's BigIP servers, or in software, such as the Apache mod_proxy_balancer extension, the Pound load balancer and reverse proxy software, and the Squid proxy and cache daemon. Load balancing is an optimization technique; it can be used to increase utilization and throughput, lower latency, reduce response time, and avoid system overload.

The following network resources can be load balanced:

- Network interfaces and services such as DNS, FTP, and HTTP
- Connections through intelligent switches
- Processing through computer system assignment
- Storage resources
- Access to application instances

Without load balancing, cloud computing would very difficult to manage. Load balancing provides the necessary redundancy to make an intrinsically unreliable system reliable through managed redirection. It also provides fault tolerance when coupled with a failover mechanism. Load balancing is nearly always a feature of server farms and computer clusters and for high availability applications.

A load-balancing system can use different mechanisms to assign service direction. In the simplest load-balancing mechanisms, the load balancer listens to a network port for service requests. When a request from a client or service requester arrives, the load balancer uses a scheduling algorithm to assign where the request is sent. Typical scheduling algorithms in use today are round robin and weighted round robin, fastest response time, least connections and weighted least connections, and custom assignments based on other factors.

A session ticket is created by the load balancer so that subsequent related traffic from the client that is part of that session can be properly routed to the same resource. Without this session record or persistence, a load balancer would not be able to correctly failover a request from one resource to another. Persistence can be enforced using session data stored in a database and replicated across

multiple load balancers. Other methods can use the client's browser to store a client-side cookie or through the use of a rewrite engine that modifies the URL. Of all these methods, a session cookie stored on the client has the least amount of overhead for a load balancer because it allows the load balancer an independent selection of resources.

The algorithm can be based on a simple round robin system where the next system in a list of systems gets the request. Round robin DNS is a common application, where IP addresses are assigned out of a pool of available IP addresses. Google uses round robin DNS.

3.2.1 Advanced load balancing

The more sophisticated load balancers are workload managers. They determine the current utilization of the resources in their pool, the response time, the work queue length, connection latency and capacity, and other factors in order to assign tasks to each resource. Among the features you find in load balancers are polling resources for their health, the ability to bring standby servers online (priority activation), workload weighting based on a resource's capacity (asymmetric loading), HTTP traffic compression, TCP offload and buffering, security and authentication, and packet shaping using content filtering and priority queuing.

An Application Delivery Controller (ADC) is a combination load balancer and application server that is a server placed between a firewall or router and a server farm providing Web services. An Application Delivery Controller is assigned a virtual IP address (VIP) that it maps to a pool of servers based on application specific criteria. An ADC is a combination network and application layer device.

You also may come across ADCs referred to as a content switch, multilayer switch, or Web switch.

These vendors, among others, sell ADC systems:

- A10 Networks (http://www.a10networks.com/)
- Barracuda Networks (http://www.barracudanetworks.com/)
- Brocade Communication Systems (http://www.brocade.com/)
- Cisco Systems (http://www.cisco.com/)
- Citrix Systems (http://www.citrix.com/)
- F5 Networks (http://www.f5.com/)
- Nortel Networks (http://www.nortel.com/)
- Coyote Point Systems (http://www.coyotepoint.com/)
- Radware (http://www.radware.com/)

An ADC is considered to be an advanced version of a load balancer as it not only can provide the features described in the previous paragraph, but it conditions content in order to lower the workload of the Web servers. Services provided by an ADC include data compression, content caching, server health monitoring, security, SSL offload and advanced routing based on current conditions. An ADC is considered to be an application accelerator, and the current products in this area are usually focused on two areas of technology: network optimization, and an application or framework optimization. For example, you may find ADC's that are tuned to accelerate ASP.NET or AJAX applications.

An architectural layer containing ADCs is described as an Application Delivery Network (ADN), and is considered to provide WAN optimization services. Often an ADN is comprised of a pair of redundant ADCs. The purpose of an ADN is to distribute content to resources based on application specific criteria. ADN provide a caching mechanism to reduce traffic, traffic prioritization and optimization, and other techniques. ADN began to be deployed on Content Delivery Networks (CDN) in the late 1990s, where it added the ability to optimize applications (application fluency) to those networks. Most of the ADC vendors offer commercial ADN solutions.

In addition to the ADC vendors in the list above, these are additional ADN vendors, among others:

- Akamai Technologies (http://www.akamai.com/)
- Blue Coat Systems (http://www.bluecoat.com/)
- CDNetworks (http://www.cdnetworks.com/)
- Crescendo Networks (http://www.crescendonetworks.com/)
- Expand Networks (http://www.expand.com/)
- Juniper Networks (http://www.juniper.net/)
- Google's cloud is a good example of the use of load balancing, so in the next section let's consider how Google handles the many requests that they get on a daily basis.

3.2.2 The Google cloud

According to the Web site tracking firm Alexa (http://www.alexa.com/topsites), Google is the single most heavily visited site on the Internet; that is, Google gets the most hits. The investment Google has made in infrastructure is enormous, and the Google cloud is one of the largest in use today. It is estimated that Google runs over a million servers worldwide, processes a billion search requests, and generates twenty petabytes of data per day.

Google is understandably reticent to disclose much about its network, because it believes that its infrastructure, system response, and low latency are key to the company's success. Google never gives datacenter tours to journalists, doesn't disclose where its datacenters are located, and obfuscates the locations of its datacenters by wrapping them in a corporate veil. Thus, the discretely named Tetra LLC (limited liability company) owns the land for the Council Bluffs, Iowa, site, and Lapis LLC owns the land for the Lenoir, North Carolina, site. This makes Google infrastructure watching something akin to a sport to many people. So what follows is what we think we know about Google's infrastructure and the basic idea behind how Google distributes its traffic by pooling IP addresses and performing several layers of load balancing.

Google has many datacenters around the world. As of March 2008, Rich Miller of DataCenterKnowledge.com wrote that Google had at least 12 major installations in the United States and many more around the world. Google supports over 30 country specific versions of the Google index, and each localization is supported by one or more datacenters. For example, Paris, London, Moscow, Sao Paolo, Tokyo, Toronto, Hong Kong, Beijing and others support their countries' locale. Germany has three centers in Berlin, Frankfurt, and Munich; the Netherlands has two at Groningen and Eemshaven. The countries with multiple datacenters store index replicas and support network peering relationships. Network peering helps Google have low latency connections to large Internet hubs run by different network providers.

You can find a list of sites as of 2008 from Miller's FAQ at http://www.datacenter knowledge.com/archives/2008/03/27/google-data-center-faq/.

Based on current locations and the company's statements, Google's datacenters are sited based on the following factors (roughly in order of importance):

1. Availability of cheap and, if possible, renewable energy

2. The relative locations of other Google datacenters such that the site provides the lowest latency response between sites

3. Location of nearby Internet hubs and peering sites

4. A source of cooling water

5. The ability to purchase a large area of land surrounding the site Speculation on why Google purchases large parcels of land ranges from creating a buffer zone between the datacenter and surrounding roads and towns or possibly to allow for building wind farms when practical.

6. Tax concessions from municipalities that lower Google's overhead.

Google maintains a pool of hundreds of IP addresses, all of which eventually resolve to its Mountain View, California, headquarters. When you initiate a Google search, your query is sent to a DNS server, which then queries Google's DNS servers. The Google DNS servers examine the pool of addresses to determine which addresses are geographically closest to the query origin and uses a round robin policy to assign an IP address to that request. The request usually goes to the nearest Google doesn't use hardware virtualization; it performs server load balancing to distribute the processing load and to get high utilization rates. The workload management software transfers the workload from a failed server over to a redundant server, and the failed server is taken offline. Multiple instances of various Google applications are running on different hosts, and data is stored on redundant storage systems.

3.3 Understanding Hypervisors

Load balancing virtualizes systems and resources by mapping a logical address to a physical address. Another fundamental technology for abstraction creates virtual systems out of physical systems. If load balancing is like playing a game of hot potato, then virtual machine technologies is akin to playing slice and dice with the potato.

Given a computer system with a certain set of resources, you can set aside portions of those resources to create a virtual machine. From the standpoint of applications or users, a virtual machine has all the attributes and characteristics of a physical system but is strictly software that emulates a physical machine. A system virtual machine (or a hardware virtual machine) has its own address space in memory, its own processor resource allocation, and its own device I/O using its own virtual device drivers. Some virtual machines are designed to run only a single application or process and are referred to as process virtual machines.

A virtual machine is a computer that is walled off from the physical computer that the virtual machine is running on. This makes virtual machine technology very useful for running old versions of operating systems, testing applications in what amounts to a sandbox, or in the case of cloud computing, creating

virtual machine instances that can be assigned a workload. Virtual machines provide the capability of running multiple machine instances, each with their own operating system.

From the standpoint of cloud computing, these features enable VMMs to manage application provisioning, provide for machine instance cloning and replication, allow for graceful system failover, and provide several other desirable features. The downside of virtual machine technologies is that having resources indirectly addressed means there is some level of overhead.

3.3.1 Virtual machine types

A low-level program is required to provide system resource access to virtual machines, and this program is referred to as the hypervisor or Virtual Machine Monitor (VMM). A hypervisor running on bare metal is a Type 1 VM or native VM. Examples of Type 1 Virtual Machine Monitors are LynxSecure, RTS Hypervisor, Oracle VM, Sun xVM Server, VirtualLogix VLX, VMware ESX and ESXi, and Wind River VxWorks, among others. The operating system loaded into a virtual machine is referred to as the guest operating system, and there is no constraint on running the same guest on multiple VMs on a physical system. Type 1 VMs have no host operating system because they are installed on a bare system.

An operating system running on a Type 1 VM is a full virtualization because it is a complete simulation of the hardware that it is running on. Some hypervisors are installed over an operating system and are referred to as Type 2 or hosted VM. Examples of Type 2 Virtual Machine Monitors are Containers, KVM, Microsoft Hyper V, Parallels Desktop for Mac, Wind River Simics, VMWare Fusion, Virtual Server 2005 R2, Xen, Windows Virtual PC, and VMware Workstation 6.0 and Server, among others. This is a very rich product category.

Type 2 virtual machines are installed over a host operating system; for Microsoft Hyper-V, that operating system would be Windows Server. In the section that follows, the Xen hypervisor (which runs on top of a Linux host OS) is more fully described. Xen is used by Amazon Web Services to provide Amazon Machine Instances (AMIs).

Figure 5.1 shows a diagram of Type 1 and Type 2 hypervisors.

On a Type 2 VM, a software interface is created that emulates the devices with which a system would normally interact. This abstraction is meant to place many I/O operations outside the virtual environment, which makes it both programmatically easier and more efficient to execute device I/O than it would be inside a virtual environment. This type of virtualization is sometimes referred to as *paravirtualization,* and it is found in hypervisors such as Microsoft's Hyper-V and Xen. It is the host operating system that is performing the I/O through a para-API.

FIGURE 5.1

VMware's vSphere cloud computing infrastructure model

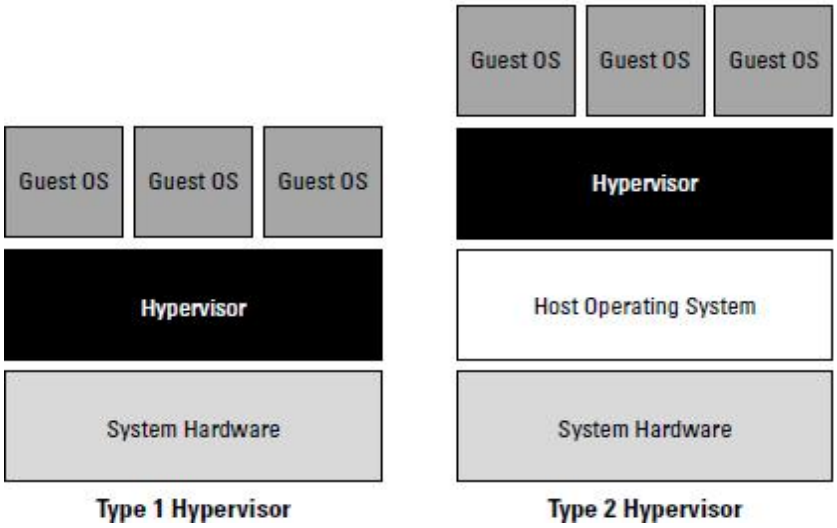
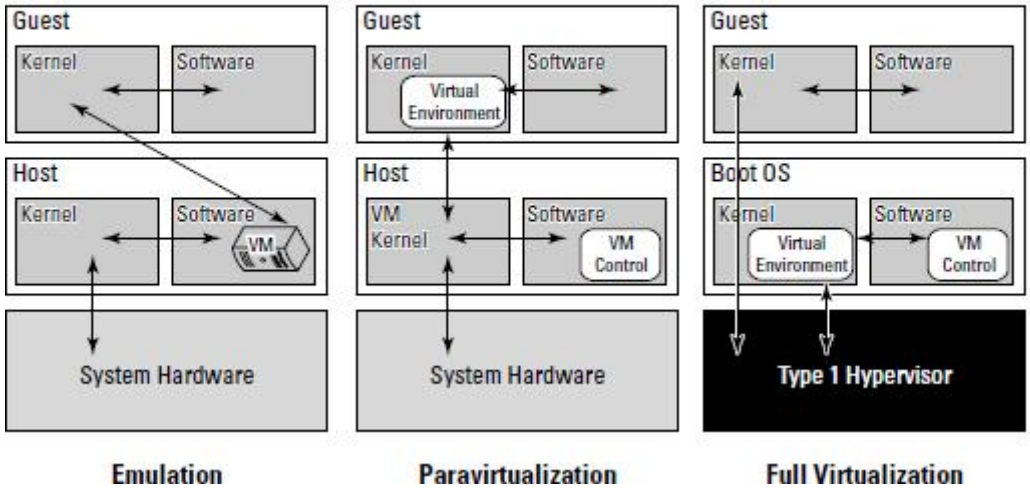


Figure 5.2 shows the difference between emulation, paravirtualization, and full virtualization. In emulation, the virtual machine simulates hardware, so it can be independent of the underlying system hardware. A guest operating system using emulation does not need to be modified in any way. Paravirtualization requires that the host operating system provide a virtual machine interface for the guest operating system and that the guest access hardware through that host VM. An operating system running as a guest on a paravirtualization system must be ported to work with the host interface. Finally, in a full virtualization scheme, the VM is installed as a Type 1 Hypervisor directly onto the hardware. All operating systems in full virtualization communicate directly with the VM hypervisor, so guest operating systems do not require any modification. Guest operating systems in full virtualization systems are generally faster than other virtualization schemes.

FIGURE 5.2

Emulation, paravirtualization, and full virtualization types



The Virtual Machine Interface (VMI) open standard (http://vmi.ncsa.uiuc.edu/) that VMware has proposed is an example of a paravirtualization API. The latest version of VMI is 2.1, and it ships as a default installation with many versions of the Linux operating system.

You are probably familiar with process or application virtual machines. Most folks run the Java Virtual Machine or Microsoft's .NET Framework VM (called the Common Language Runtime or CLR) on their computers. A process virtual machine instantiates when a command begins a process, the VM is created by an interpreter, the VM then executes the process, and finally the VM exits the system and is destroyed. During the time the VM exists, it runs as a high-level abstraction. Applications running inside an application virtual machine are generally slow, but these programs are very popular because they provide portability, offer rich programming languages, come with many advanced features, and allow platform independence for their programs. Although many cloud computing applications provide process virtual machine applications, this type of abstraction isn't really suitable for building a large or high-performing cloud network, with one exception.

The exception is the process VMs that enable a class of parallel cluster computing applications. These applications are high-performance systems where the virtual machine is operating one process per cluster node, and the system maintains the necessary intra-application communications over the network interconnect. Examples of this type of system are the Parallel Virtual Machine (PVM; see http://www.csm.ornl.gov/pvm/pvm_home.html) and the Message Passing Interface (MPI; see http://www.mpi-forum.org/). Some people do not consider these application VMs to be true virtual machines, noting that these applications can still access the host operating system services on the specific system on which they are running. The emphasis on using these process VMs is in creating a high-performance networked supercomputer often out of heterogeneous systems, rather than on creating a ubiquitous utility resource that characterizes a cloud network.

Some operating systems such as Sun Solaris and IBM AIX 6.1 support a feature known as *operating system virtualization.* This type of virtualization creates virtual servers at the operating system or kernel level. Each virtual server is running in its own virtual environment (VE) as a virtual private server (VPS). Different operating systems use different names to describe these machine instances, each of which can support its own guest OS. However, unlike true virtual machines, VPS must all be running the same OS and the same version of that OS. Sun Solaris 10 uses VPS to create what is called Solaris Zones. With IBM AIX, the VPS is called a System Workload Partition (WPAR). This type of virtualization allows for a dense collection of virtual machines with relatively low overhead.

3.3.2 VMware vSphere

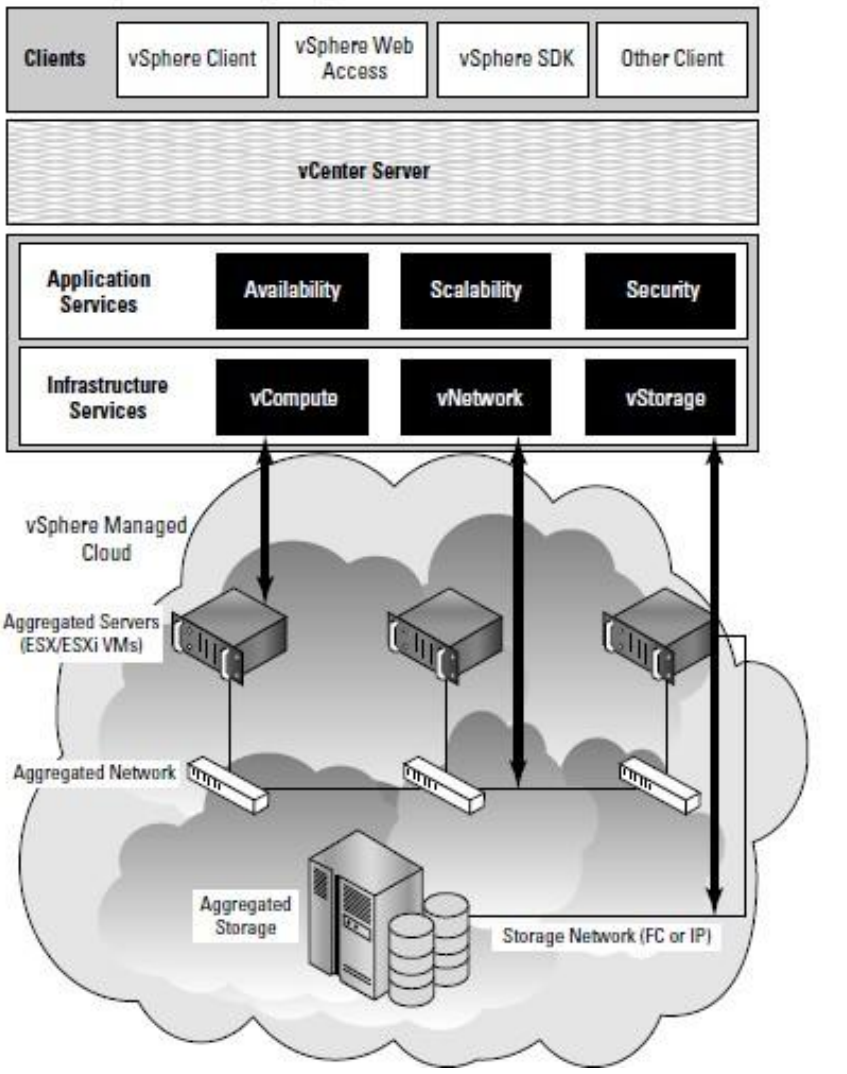
VMware vSphere is a management infrastructure framework that virtualizes system, storage, and networking hardware to create cloud computing infrastructures. vSphere is the branding for a set of management tools and a set of products previously labeled VMware Infrastructure. vSphere provides a set of services that applications can use to access cloud resources, including these:

- **VMware vCompute:** A service that aggregates servers into an assignable pool
- **VMware vStorage:** A service that aggregates storage resources into an assignable pool.
- **VMware vNetwork:** A service that creates and manages virtual network interfaces.
- **Application services:** Such as HA (High Availability) and Fault Tolerance.
- **vCenter Server:** A provisioning, management, and monitoring console for VMware cloud infrastructures

Figure 5.3 shows an architectural diagram of a vSphere cloud infrastructure.

FIGURE 5.3

VMware's vSphere cloud computing infrastructure model



A vSphere cloud is a pure infrastructure play. The virtualization layer that abstracts processing, memory, and storage uses the VMware ESX or ESXi virtualization server. ESX is a Type 1 hypervisor; it installs over bare metal (a clean system) using a Linux kernel to boot and installs the vmkernel hypervisor (virtualization kernel and support files). When the system is rebooted, the vmkernel loads first, and then the Linux kernel becomes the first guest operating system to run as a virtual machine on the system and contains the service console.

VMware is a very highly developed infrastructure and the current leader in this industry. A number of important add-on products are available for cloud computing applications. These are among the more notable products:

- **Virtual Machine File System (VMFS):** A high-performance cluster file system for an ESX/ESXi cluster.

- **VMotion:** A service that allows for the migration of a virtual machine from one physical server to another physical server while the virtual server runs continuously and without any interruption of ongoing transactions.

The ability to live migrate virtual machines is considered to be a technological tour de force and a differentiator from other virtual machine system vendors.

- **Storage VMotion:** A product that can migrate files from one datastore to another datastore while the virtual machine that uses the datastore continues to run.
- **Virtual SMP:** A feature that allows a virtual machine to run on two or more physical processors at the same time.
- **Distributed Resource Scheduler (DRS):** A system for provisioning virtual machines and load balancing processing resources dynamically across the different physical systems that are in use. A part of the DRS called the distributed power management (DPM) module can manage the power consumption of systems.
- **vNetwork Distributed Switch (DVS):** A capability to maintain a network runtime state for virtual machines as they are migrated from one physical system to another. DVS also monitors network connections, provides firewall services, and enables the use of third party switches such as the Cisco Nexus 1000V to manage virtual networks.

You can get a better sense of how the different resources are allocated by vSphere into a virtual set of components by examining Figure 5.4. Physical computers can be standalone hosts or a set of clustered systems. In either case, a set of virtual machines can be created that is part of a single physical system or spans two or more physical systems.

You can define a group of VMs as a Resource Pool (RP) and, by doing so, manage those virtual machines as a single object with a single policy. Resource Pools can be placed into a hierarchy or nested and can inherit properties of their parent RP. As more hosts or cluster nodes are added or removed, vSphere can dynamically adjust the provisioning of VMs to accommodate the policy in place. This fine tuning of pooled resources is required to accommodate the needs of cloud computing networks.

The datastore shown at the center of Figure 5.4 is a shared storage resource. These storage resources can be either Direct Attached Storage (DAS) of a server using SCSI, SAS, or SATA connections, Fibre Channel disk arrays/SANs, iSCSI disk arrays/SANs, or Network Attached Storage (NAS) disk arrays. Although the lines drawn between the datastore and different VMs indicate a direct connection, with the exception of DAS, the other storage types are shared storage solutions.

Storage virtualization is most commonly achieved through a mapping mechanism where a logical storage address is translated into a physical storage address. Block-based storage such as those used in SANs use a feature called a Logical Unit Identifier (LUN**)** with specific addresses stored in the form of an offset called the Logical Block Address (LBA). The address space mapping then maps the address of the logical or virtual disk (vdisk) to the logical unit on a storage controller. Storage virtualization may be done in software or in hardware, and it allows requests for virtualized storage to be redirected as needed.

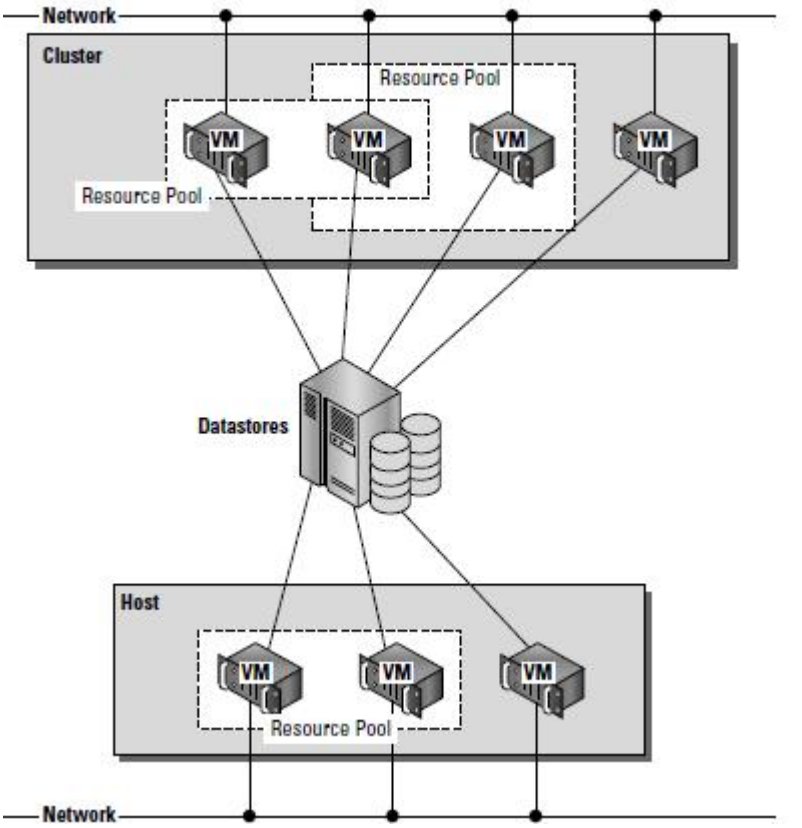
Similarly, network virtualization abstracts networking hardware and software into a virtual network that can be managed. A virtual network can create virtual network interfaces (VNICs) or virtual LANs

(VLANS) and can be managed by a hypervisor, operating system, or external management console. In a virtualized infrastructure such as the one presented in this section, internal network virtualization is occurring and the hypervisor interacts with networking hardware to create a pseudo-network interface. External network virtualization can be done using network switches and VLAN software.

The key feature that makes virtual infrastructure so appealing for organizations implementing a cloud computing solution is flexibility. Instantiating a virtual machine is a very fast process, typically only a few seconds in length. You can make machine images of systems in the configuration that you want to deploy or take snapshots of working virtual machines. These images can be brought on-line as needed.

FIGURE 5.4

Virtual infrastructure elements



3.4 Understanding Machine Imaging

In the preceding sections, you have seen how the abstractions that cloud computing needs can be achieved through redirection and virtualization. A third mechanism is commonly used to provide system portability, instantiate applications, and provision and deploy systems in the cloud. This third mechanism is through storing the state of a systems using a system image. A system image makes a copy or a clone of the entire computer system inside a single container such as a file. The system imaging program is used to make this image and can be used later to restore a system image. Some imaging programs can take snapshots of systems, and most allow you to view the files contained in the image and do partial restores.

A prominent example of a system image and how it can be used in cloud computing architectures is the Amazon Machine Image (AMI) used by Amazon Web Services to store copies of a virtual

machine. Because this is a key feature of Amazon's Elastic Compute Cloud and is discussed in detail in Chapter 9, I briefly mention it here. An AMI is a file system image that contains an operating system, all appropriate device drivers, and any applications and state information that the working virtual machine would have.

When you subscribe to AWS, you can choose to use one of its hundreds of canned AMIs or to create a custom system and capture that system's image to an AMI. An AMI can be for public use under a free distribution license, for pay-per-use with operating systems such as Windows, or shared by an EC2 user with other users who are given the privilege of access.

The AMI file system is not a standard bit-for-bit image of a system that is common to many disk imaging programs. AMI omits the kernel image and stores a pointer to a particular kernel that is part of the AWS kernel library. Among the choices are Red Hat Linux, Ubuntu, Microsoft Windows, Solaris, and others. Files in AMI are compressed and encrypted, and an XML file is written that describes the AMI archive. AMIs are typically stored in your Amazon S3 (Simple Storage System) buckets as a set of 10MB chunks.

Machine images are sometimes referred to as "virtual appliances"—systems that are meant to run on virtualization platforms. AWS EC2 runs on the Xen hypervisor, for example. The term *virtual appliance* is meant to differentiate the software image from an operating virtual machine. The system image contains the operating system and applications that create an environment. Most virtual appliances are used to run a single application and are configurable from a Web page. Virtual appliances are a relatively new paradigm for application deployment, and cloud computing is the major reason for the interest in them and for their adoption. This area of WAN application portability and deployment, and of WAN optimization of an application based on demand, is one with many new participants. Certeon (http://www.certeon.com/), Expand Networks (http://www.expand.com/), and Replify (http://www.replify.com/) are three vendors offering optimization appliances for VMware's infrastructure.

3.5 Porting Applications

Cloud computing applications have the ability to run on virtual systems and for these systems to be moved as needed to respond to demand. Systems (VMs running applications), storage, and network assets can all be virtualized and have sufficient flexibility to give acceptable distributed WAN application performance. Developers who write software to run in the cloud will undoubtedly want the ability to port their applications from one cloud vendor to another, but that is a much more difficult proposition. Cloud computing is a relatively new area of technology, and the major vendors have technologies that don't interoperate with one another.

3.5.1 The Simple Cloud API

If you build an application on a platform such as Microsoft Azure, porting that application to Amazon Web Services or GoogleApps may be difficult, if not impossible. In an effort to create an interoperability standard, Zend Technologies has started an open source initiative to create a common application program interface that will allow applications to be portable. The initiative is called the Simple API for Cloud Application Services (http://www.simplecloud.org/), and the effort has drawn

interest from several major cloud computing companies. Among the founding supporters are IBM, Microsoft, Nivanix, Rackspace, and GoGrid.

Simple Cloud API has as its goal a set of common interfaces for:

- **File Storage Services:** Currently Amazon S3, Windows Azure Blob Storage, Nirvanix, and Local storage is supported by the Storage API. There are plans to extend this API to Rackspace Cloud Files and GoGrid Cloud Storage.
- **Document Storage Services:** Amazon SimpleDB and Windows Azure Table Storage are currently supported. Local document storage is planned.
- **Simple Queue Services:** Amazon SQS, Windows Azure Queue Storage, and Local queue services are supported.

Zend intends to add the interface to their open source PHP Framework (http://www.framework.zend.com) as the Zend_Cloud framework component. Vendors such as Microsoft and IBM are supplying adapters that will use part of the Simple Cloud API for their cloud application services.

3.5.2 AppZero Virtual Application Appliance

Applications that run in datacenters are captive to the operating systems and hardware platforms that they run on. Many datacenters are a veritable Noah's Ark of computing. So moving an application from one platform to another isn't nearly as simple as moving a machine image from one system to another.

The situation is further complicated by the fact that applications are tightly coupled with the operating systems on which they run. An application running on Windows, for example, isn't isolated from other applications. When the application loads, it often loads or uses different Dynamic Link Libraries (DLL), and it is through the sharing or modification of DLLs that Windows applications get themselves in trouble. Further modifications include modifying the registry during installation. These factors make it difficult to port applications from one platform to another without lots of careful work. If you are a Platform as a Service (PaaS) application developer, you are packaging a complete software stack that includes not only your application, but the operating system and application logic and rules as well. Vendor lock-in for you application is assured.

The ability to run an application from whatever platform you want is not one of the characteristics of cloud computing, but you can imagine that it is a very attractive proposition. While the Simple Cloud API is useful for applications written in PHP, other methods may be needed to make applications easily portable. One company working on this problem is AppZero (http://www.appzero.com/), and its solution is called the Virtual Application Appliance (VAA).

The AppZero solution creates a virtual application appliance as an architectural layer between the Windows or the UNIX operating system and applications. The virtualization layer serves as the mediator for file I/O, memory I/O, and application calls and response to DLLs, which has the effect of sandboxing the application. The running application in AppZero changes none of the registry entries or any of the files on the Windows Server.

VAA creates a container that encapsulates the application and all the application's dependencies within a set of files; it is essentially an Application Image for a specific OS. Dependencies include DLL, service settings, necessary configuration files, registry entries, and machine and network settings. This container forms an installable server-side application stack that can be run after installation, but has no impact on the underlying operating system. VAAs are created using the AppZero Creator wizard, managed with the AppZero Admin tool, and may be installed using the AppZero Director, which creates a VAA runtime application. If desired, an application called AppZero Dissolve removes the VAA virtualization layer from the encapsulated application and installs that application directly into the operating system.

Installations can be done over the network after the AppZero application appliance is installed. Therefore, with this system, you could run applications on the same Windows Server and eliminate one application from interfering with another; applications would be much more easily ported from one Windows system to another. AppZero's approach provides the necessary abstraction layer that frees an application from its platform dependence.

An interesting use of VAAs involves segmenting an application into several VAAs, some of which are read-only runtime components, while others can be modified. When backing up or replicating VAAs in a cloud, you would need to synchronize only those VAAs that are modified. In many instances, the portion of an application that changes is only a very small component of large applications, which means that this technique can greatly reduce the amount of data required to replicate a VM in the cloud.

AppZero envisages using VAAs to create what it calls a *stateless cloud.* In a stateless cloud, the application's state information is stored on a network share where it is available to run on different cloud systems as needed. This approach allows the cloud system to run with a VM containing a clean operating system (like AWS does now) and provisioned by the VAA. This approach should greatly reduce the number of complete system images that cloud vendors and cloud users should need to store to support their work; it also should make the running of applications on secure, well-performing VEs easier to achieve.

3.6 VIRTUAL MACHINES PROVISIONING AND MANAGEABILITY

In this section, we will have an overview on the typical life cycle of VM and its major possible states of operation, which make the management and automation of VMs in virtual and cloud environments easier than in traditional computing environments.

As shown in Figure 3.1, the cycle starts by a request delivered to the IT department, stating the requirement for creating a new server for a particular service. This request is being processed by the IT administration to start seeing the servers' resource pool, matching these resources with the requirements, and starting the provision of the needed virtual machine. Once it is provisioned and started, it is ready to provide the required service according to an SLA, or a time period after which the virtual is being released; and free resources, in this case, won't be needed.

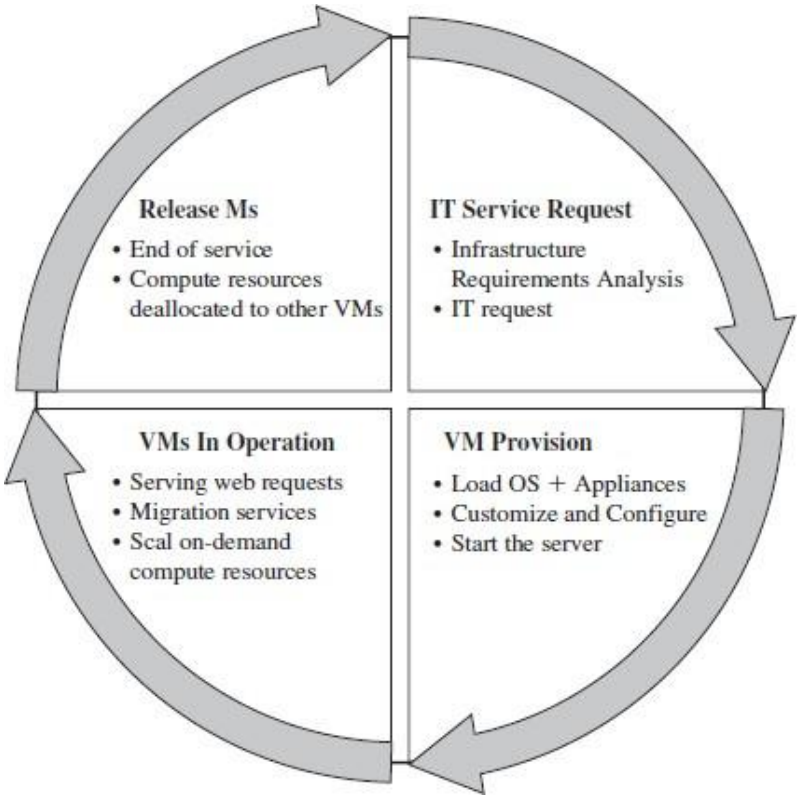


FIGURE 3.1. Virtual machine life cycle.

3.6.1 VM Provisioning Process

Provisioning a virtual machine or server can be explained and illustrated as in Figure 3.2.:

Steps to Provision VM. Here, we describe the common and normal steps of provisioning a virtual server:

- Firstly, you need to select a server from a pool of available servers(physical servers with enough capacity) along with the appropriate OS template you need to provision the virtual machine.
- Secondly, you need to load the appropriate software (operating system you selected in the previous step, device drivers, middleware, and the needed applications for the service required).
- Thirdly, you need to customize and configure the machine (e.g., IP address, Gateway) to configure an associated network and storage resources.
- Finally, the virtual server is ready to start with its newly loaded software.

Typically, these are the tasks required or being performed by an IT or a datacenter's specialist to provision a particular virtual machine.

To summarize, server provisioning is defining server's configuration based on the organization requirements, a hardware, and software component (processor, RAM, storage, networking, operating system, applications, etc.). Normally, virtual machines can be provisioned by manually installing an operating system, by using a preconfigured VM template, by cloning an existing VM, or by importing a physical server or a virtual server from another hosting platform. Physical servers can also be virtualized and provisioned using P2V(physical to virtual) tools and techniques (e.g., virt-p2v).

After creating a virtual machine by virtualizing a physical server, or by building a new virtual server in the virtual environment, a template can be created out of it. Most virtualization management vendors (VMware, Xen Server,etc.) provide the data center's administration with the ability to do such tasks in an easy way. Provisioning from a template is an invaluable feature, because it reduces the time required to create a new virtual machine.

FIGURE 3.2 Virtual machine provision process

Administrators can create different templates for different purposes. For example, you can create a Windows 2003 Server template for the finance department, or a Red Hat Linux template for the engineering department. This enables the administrator to quickly provision a correctly configured virtual server on demand.

This ease and flexibility bring with them the problem of virtual machine's sprawl, where virtual machines are provisioned so rapidly that documenting and managing the virtual machine's life cycle become a challenge.

3.7 VIRTUAL MACHINE MIGRATION SERVICES

Migration service, in the context of virtual machines, is the process of moving a virtual machine from one host server or storage location to another; there are different techniques of VM migration, hot/life migration, cold/regular migration, and live storage migration of a virtual machine . In this process, all key machines' components, such as CPU, storage disks, networking, and memory, are completely virtualized, thereby facilitating the entire state of a virtual machine to be captured by a set of easily moved data files. We will cover some of the migration's techniques that most virtualization tools provide as a feature.

3.7.1 Migrations Techniques

Live Migration and High Availability. Live migration (which is also called hot or real-time migration) can be defined as the movement of a virtual machine from one physical host to another while being powered on. When it is properly carried out, this process takes place without any noticeable effect from the end user's point of view (a matter of milliseconds). One of the most significant advantages of live migration is the fact that it facilitates proactive maintenance in case of failure, because the potential problem can be resolved before the disruption of service occurs. Live migration can also be used for load balancing in which work is shared among computers in order to optimize the utilization of available CPU resources.

Live Migration Anatomy, Xen Hypervisor Algorithm. In this section we will explain live migration's mechanism and how memory and virtual machine states are being transferred, through the network, from one host A to another host B; the Xen hypervisor is an example for this mechanism. The logical steps that are executed when migrating an OS are summarized in Figure 3.3. In this research, the migration process has been viewed as a transactional interaction between the two hosts involved:

Stage 0: Pre-Migration. An active virtual machine exists on the physical host A.

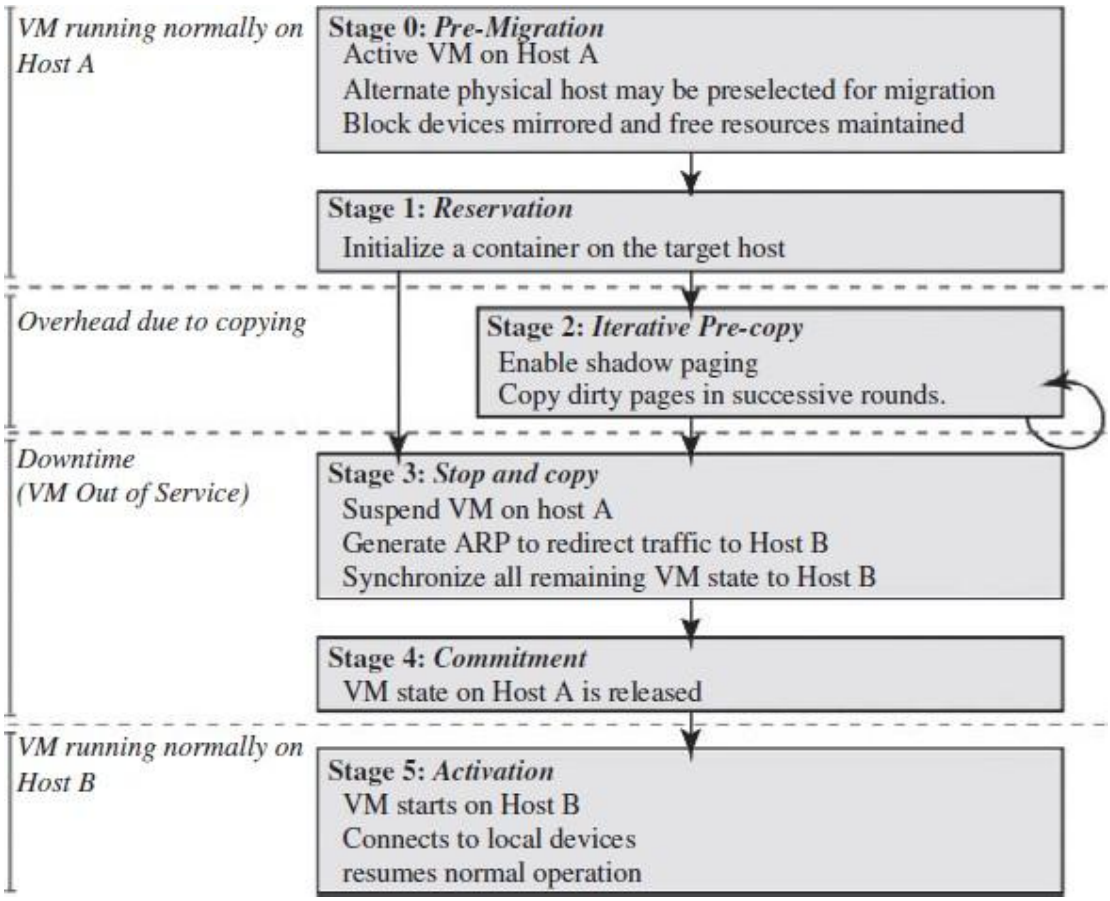


FIGURE 3.3 Live Migration Timeline

Stage 1: Reservation. A request is issued to migrate an OS from host A to host B (a precondition is that the necessary resources exist on B and on a VM container of that size).

Stage 2: Iterative Pre-Copy. During the first iteration, all pages are transferred from A to B. Subsequent iterations copy only those pages dirtied during the previous transfer phase.

Stage 3: Stop-and-Copy. Running OS instance at A is suspended, and its network traffic is redirected to B. CPU state and any remaining inconsistent memory pages are then transferred. At the end of this stage, there is a consistent suspended copy of the VM at both A and B. The copy at A is considered primary and is resumed in case of failure.

Stage 4: Commitment. Host B indicates to A that it has successfully received a consistent OS image. Host A acknowledges this message as a commitment of the migration transaction. Host A may now discard the original VM, and host B becomes the primary host.

Stage 5: Activation. The migrated VM on B is now activated. Post-migration code runs to reattach the device's drivers to the new machine and advertise moved IP addresses.

This approach to failure management ensures that at least one host has a consistent VM image at all times during migration. It depends on the assumption that the original host remains stable until the migration commits and that the VM may be suspended and resumed on that host with no risk of failure. Based on these assumptions, a migration request essentially attempts to move the VM to a new host and on any sort of failure, execution is resumed locally, aborting the migration.

Live Migration Effect on a Running Web Server. Clark et al. did evaluate the above migration on an Apache 1.3 Web server; this served static content at a high rate, as illustrated in Figure 3.4. The throughput is achieved when continuously serving a single 512-kB file to a set of one hundred concurrent clients. The Web server virtual machine has a memory allocation of 800 MB. At the start of the trace, the server achieves a consistent throughput of approximately 870 Mbit/sec. Migration starts 27 sec into the trace, but is initially rate-limited to 100 Mbit/sec (12% CPU), resulting in server's throughput drop to 765 Mbit/sec. This initial low-rate pass transfers 776 MB and lasts for 62 sec. At this point, the migration's algorithm, increases its rate over several iterations and finally suspends the VM after a further 9.8 sec. The final stop-and-copy phase then transfers the remaining pages, and the Web server resumes at full rate after a 165-msec outage.

This simple example demonstrates that a highly loaded server can be migrated with both controlled impact on live services and a short downtime. However, the working set of the server, in this case, is rather small. So, this should be expected as a relatively easy case of live migration.

Live Migration Vendor Implementations Examples. There are lots of VM management and provisioning tools that provide the live migration of VM facility, two of which are VMware VMotion and Citrix XenServer"XenMotion."

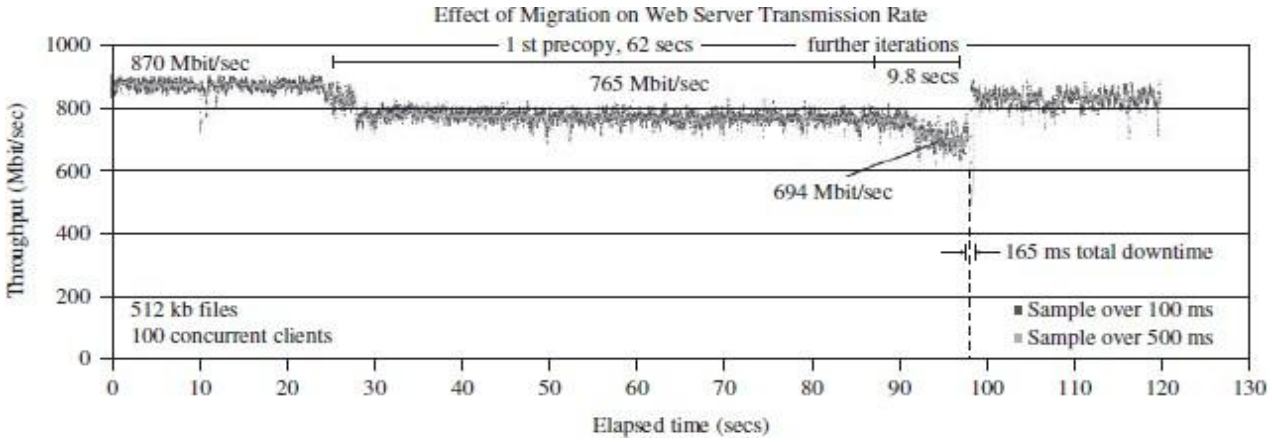


FIGURE 3.4 Result of migrating a running web server VM

VMware Vmotion. This allows users to (a) automatically optimize and allocate an entire pool of resources for maximum hardware utilization, flexibility, and availability and (b) perform hardware's maintenance without scheduled downtime along with migrating virtual machines away from failing or underperforming servers.

Citrix Xen Server Xen Motion. This is a nice feature of the Citrix Xen Server product, inherited from the Xen live migrate utility, which provides the IT administrator with the facility to move a running VM from one Xen Server to another in the same pool without interrupting the service (hypothetically for zero-downtime server maintenance, which actually takes minutes), making it a highly available service. This also can be a good feature to balance the workloads on the virtualized environment.

Regular/Cold Migration. Cold migration is the migration of a powered-off virtual machine. With cold migration, you have the option of moving the associated disks from one data store to another. The virtual machines are not required to be on a shared storage. It's important to highlight that the two main differences between live migration and cold migration are that live migration needs a shared storage for virtual machines in the server's pool, but cold migration does not; also, in live migration for a virtual machine between two hosts, there would be certain CPU compatibility checks to be applied; while in cold migration this checks do not apply. The cold migration process is simple to implement (as the case for the VMware product), and it can be summarized as follows [24]:

- The configuration files, including the NVRAM file (BIOS settings), logfiles, as well as the disks of the virtual machine, are moved from the sourcehost to the destination host's associated storage area.
- The virtual machine is registered with the new host.
- After the migration is completed, the old version of the virtual machine isdeleted from the source host.

Live Storage Migration of Virtual Machine. This kind of migration constitutes moving the virtual disks or configuration file of a running virtual machine to a new data store without any interruption in the availability of the virtual machine's service. For more details about how this option is working in a VMware product, see reference 20.

3.7.2 VM Migration, SLA and On-Demand Computing

As we discussed, virtual machines' migration plays an important role in data centers by making it easy to adjust resource's priorities to match resource's demand conditions.

This role is completely going in the direction of meeting SLAs; once it has been detected that a particular VM is consuming more than its fair share of resources at the expense of other VMs on the same host, it will be eligible, for this machine, to either be moved to another underutilized host or assign more resources for it, in case that the host machine still has resources; this in turn will highly avoid the violations of the SLA and will also, fulfill the requirements of on-demand computing resources. In order to achieve such goals, there should be an integration between virtualization's management tools (with its migrations and performance's monitoring capabilities), and SLA's management tools to achieve balance in resources by migrating and monitoring the workloads, and accordingly, meeting the SLA.

3.7.3 Migration of Virtual Machines to Alternate Platforms

One of the nicest advantages of having facility in data center's technologies is to have the ability to migrate virtual machines from one platform to another; there are a number of ways for achieving this, such as depending on the source and target virtualization's platforms and on the vendor's tools that manage this facility—for example, the VMware converter that handles migrations between ESX hosts; the VMware server; and the VMware workstation. The VMware converter can also import from other virtualization platforms, such as Microsoft virtual server machines.

3.8 VM PROVISIONING AND MIGRATION IN ACTION

Now, it is time to get into business with a real example of how we can manage the life cycle, provision, and migrate a virtual machine by the help of one of the open source frameworks used to manage virtualized infrastructure. Here, we will use ConVirt (open source framework for the management of open source virtualization like Xen and KVM , known previously as Xen Man).

Deployment Scenario. ConVirt deployment consists of at least one ConVirt workstation, where ConVirt is installed and ran, which provides the main console for managing the VM life cycle, managing images, provisioning new VMs, monitoring machine resources, and so on.

There are two essential deployment scenarios for ConVirt: A, basic configuration in which the Xenor KVM virtualization platform is on the local machine, where ConVirt is already installed; B, an advanced configuration in which the Xen or KVM is on one or more remote servers. The scenario in use here is the advanced one. In data centers, it is very common to install centralized management software (ConVirt here) on a dedicated machine for use in managing remote servers in the data center.

In our example, we will use this dedicated machine where ConVirt is installed and used to manage a pool of remote servers (two machines). In order to use advanced features of ConVirt (e.g., live migration), you should set up a shared storage for the server pool in use on which the disks of the provisioned virtual machines are stored. Figure 3.5 illustrates the scenario.

Installation. The installation process involves the following:

- Installing ConVirt on at least one computer. See reference 28 for installation details.
- Preparing each managed server to be managed by ConVirt. See reference 28 for managed servers' installation details. We have two managing servers with the following Ips (managed server 1, IP:172.16.2.22; and managed server 2, IP:172.16.2.25) as shown in the deployment diagram (Figure 3.5).
- Starting ConVirt and discovering the managed servers you have prepared.

Notes

- Try to follow the installation steps existing in reference 28 according to the distribution of the operating system in use. In our experiment, we use Ubuntu 8.10 in our setup.
- Make sure that the managed servers include Xen or KVM hypervisors installed.
- Make sure that you can access managed servers from your ConVirt management console through SSH.

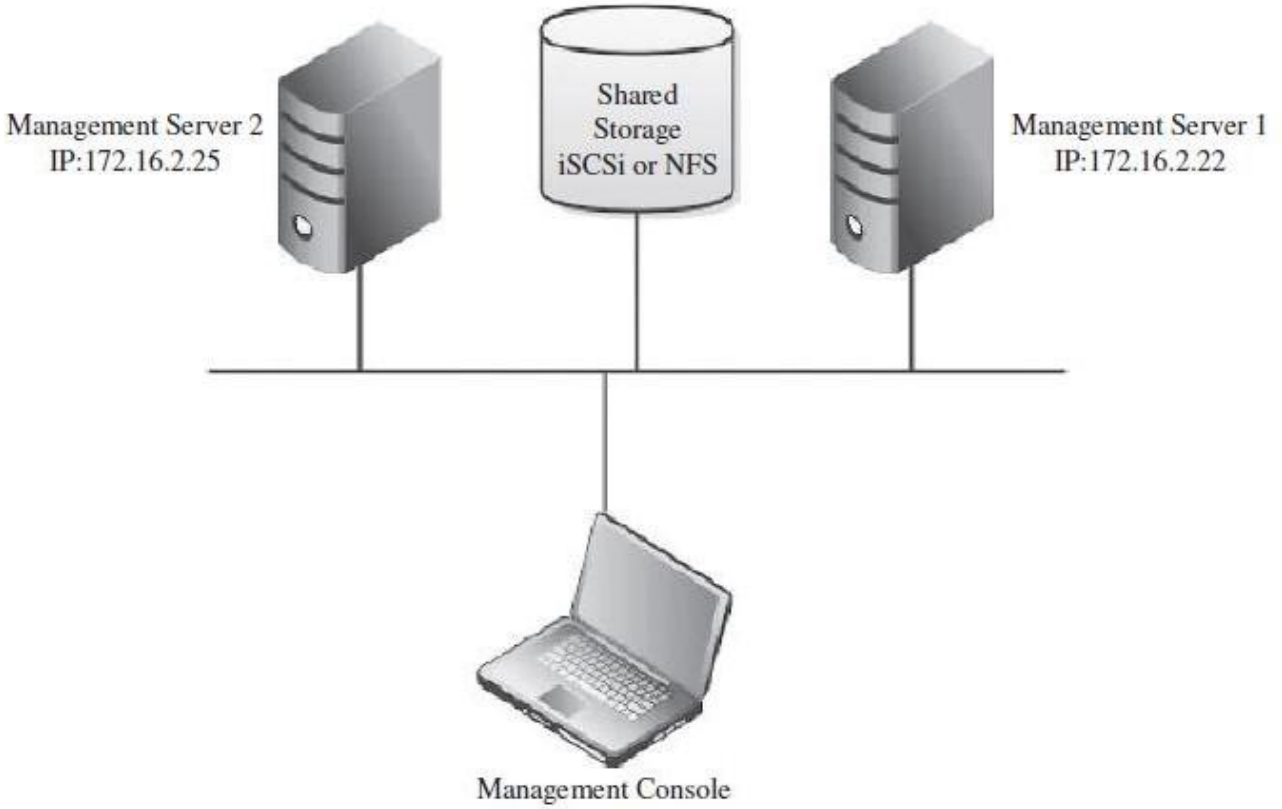


FIGURE 3.5 A deployment scenario network diagram

Environment, Software, and Hardware. ConVirt 1.1, Linux Ubuntu 8.10, three machines, Dell core 2 due processor, 4G RAM.

Adding Managed Servers and Provisioning VM. Once the installation is done and you are ready to manage your virtual infrastructure, then you can start the ConVirt management console (see Figure 3.6): Select any of servers' pools existing (QA Lab in our scenario) and on its context menu, select "Add Server."

- You will be faced with a message asking about the virtualization platform you want to manage (Xen or KVM), as shown in Figure 3.7:
- Choose KVM, and then enter the managed server information and credentials (IP, username, and password) as shown in Figure 3.8.
- Once the server is synchronized and authenticated with the management console, it will appear in the left pane/of the ConVirt, as shown in Figure 3.9.
- Select this server, and start provisioning your virtual machine as in Figure 3.10:

- Fill in the virtual machine's information (name, storage, OS template, etc.; Figure 3.11); then you will find it created on the managed server tree powered-off.

Note: While provisioning your virtual machine, make sure that you create disks on the shared storage (NFS or iSCSi). You can do so by selecting the "provisioning" tab, and changing the VM_DISKS_DIR to point to the location of your shared NFS.

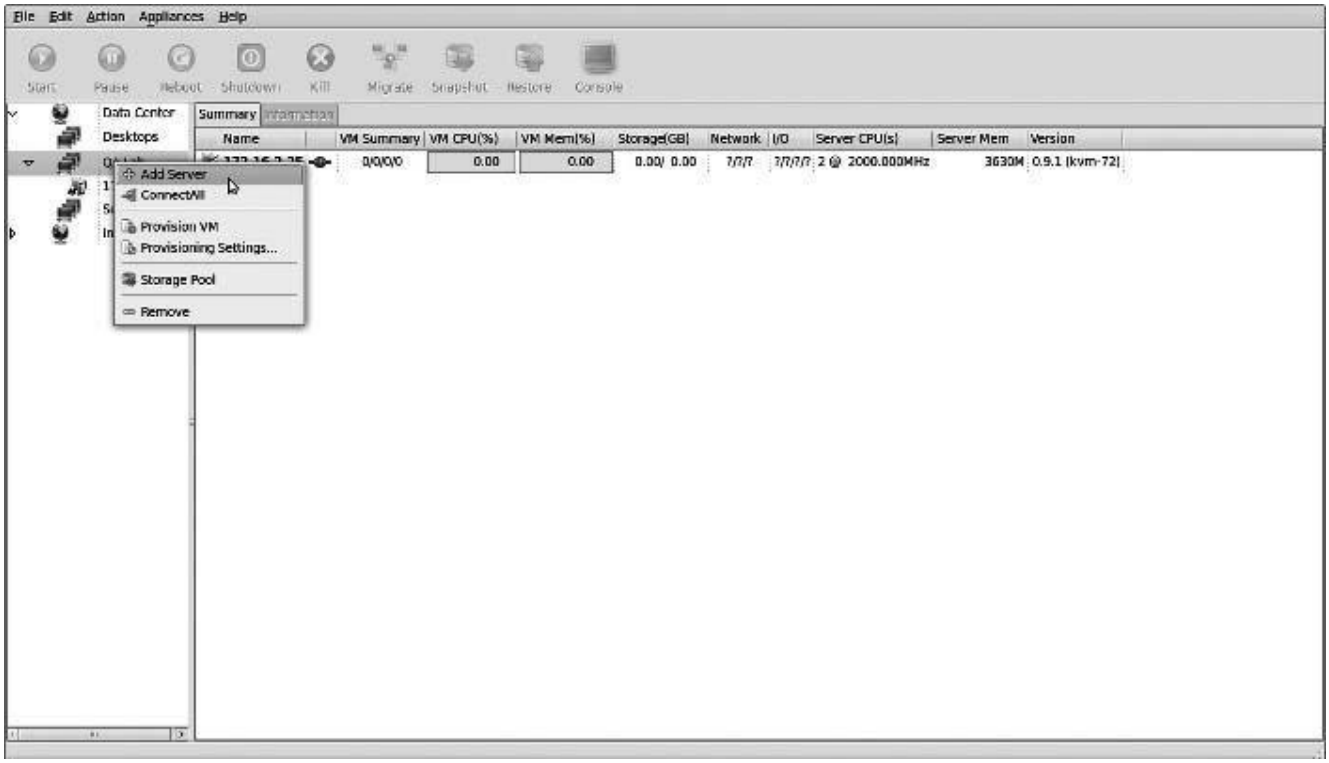


FIGURE 3.6. Adding managed server on the data centre's management console.

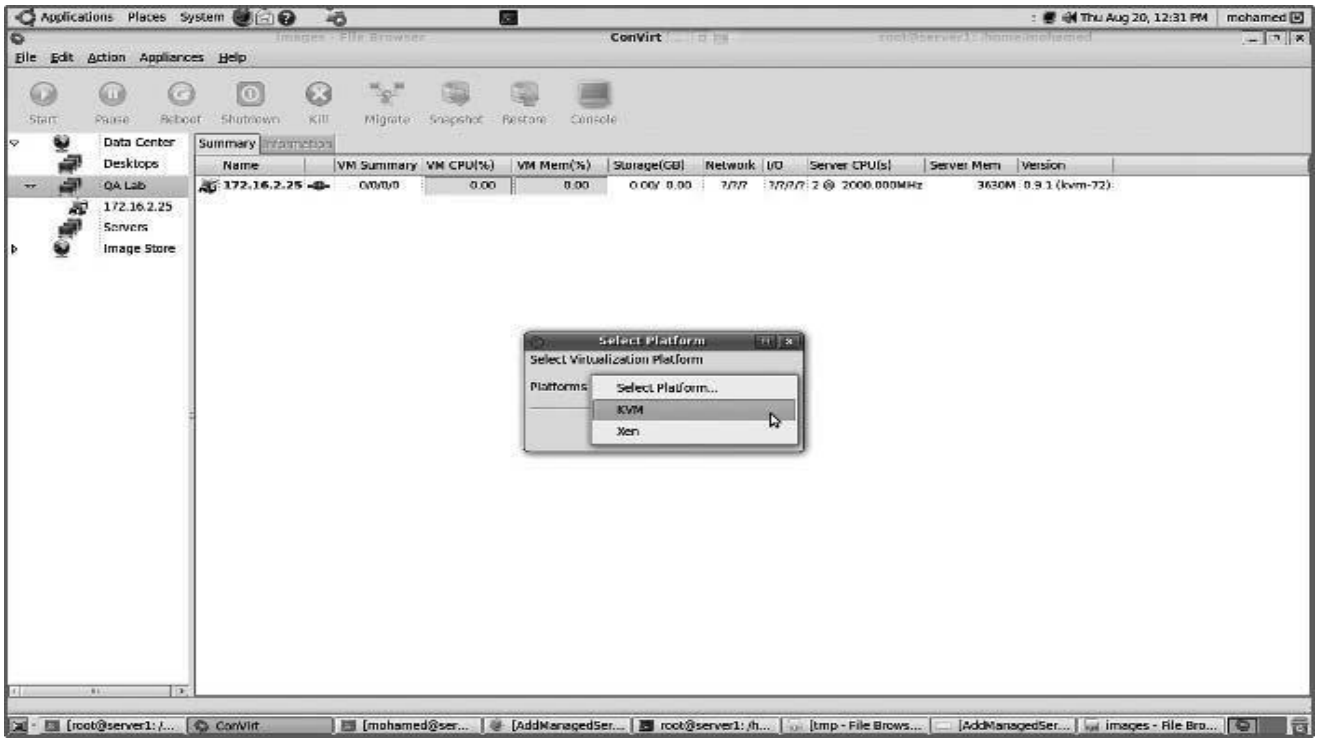


FIGURE 3.7. Select virtualization platform.

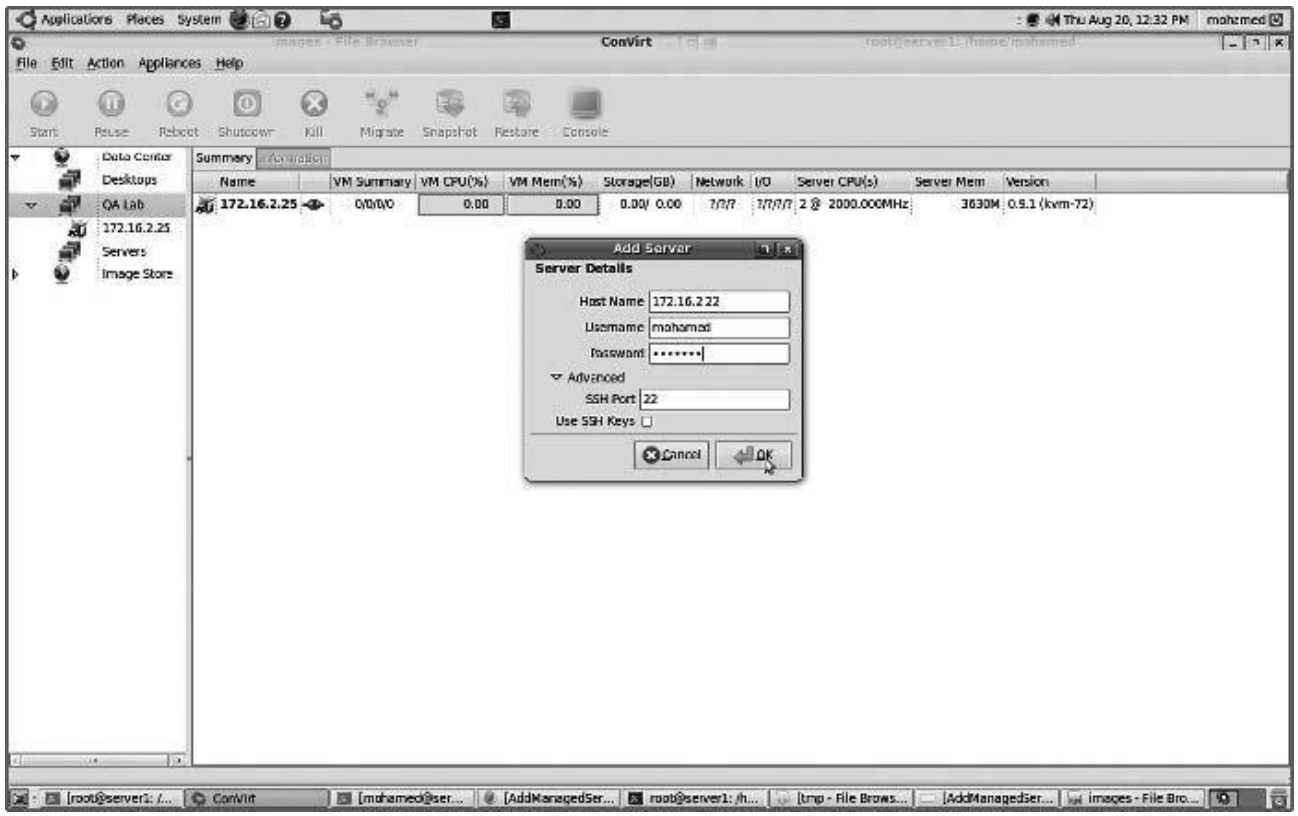


FIGURE 3.8 Managed server info and credentials.

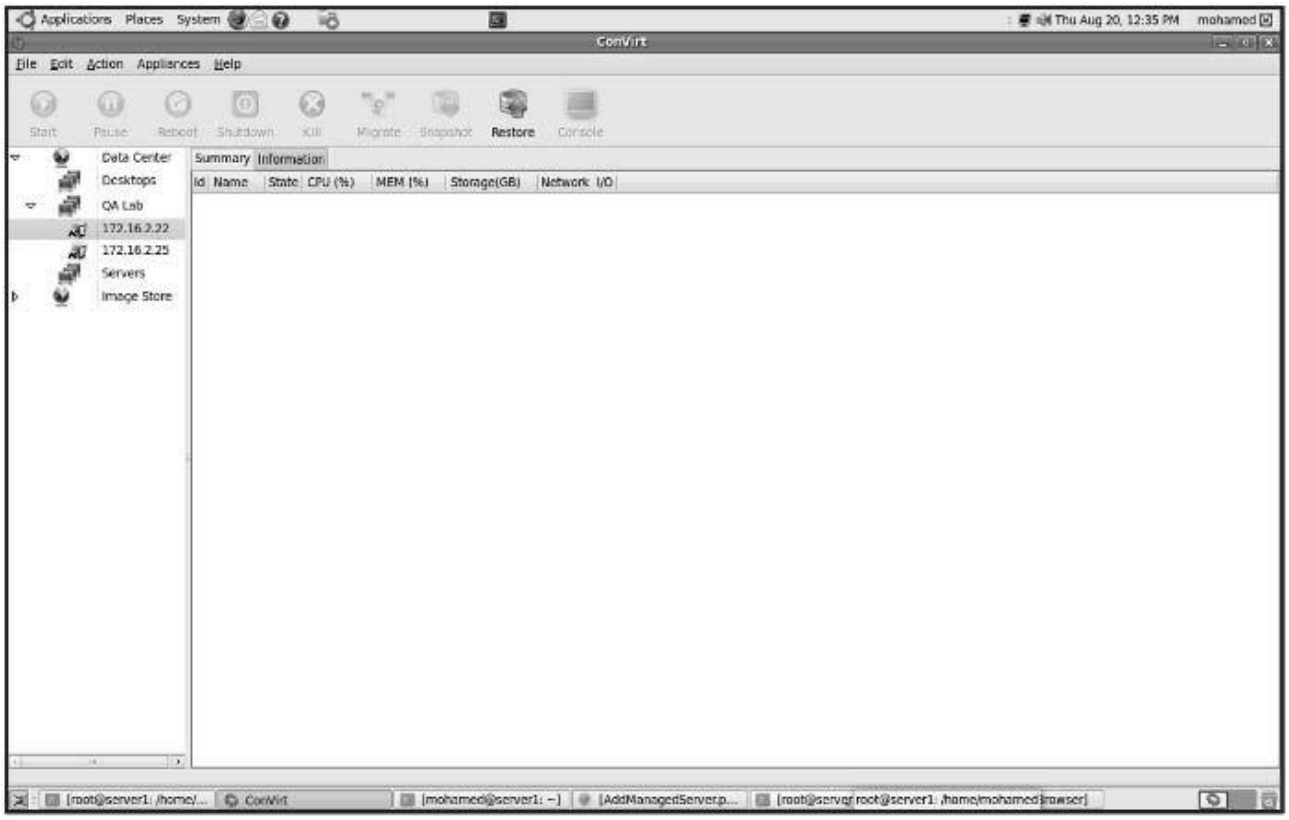


FIGURE 3.9 Managed server has been added.

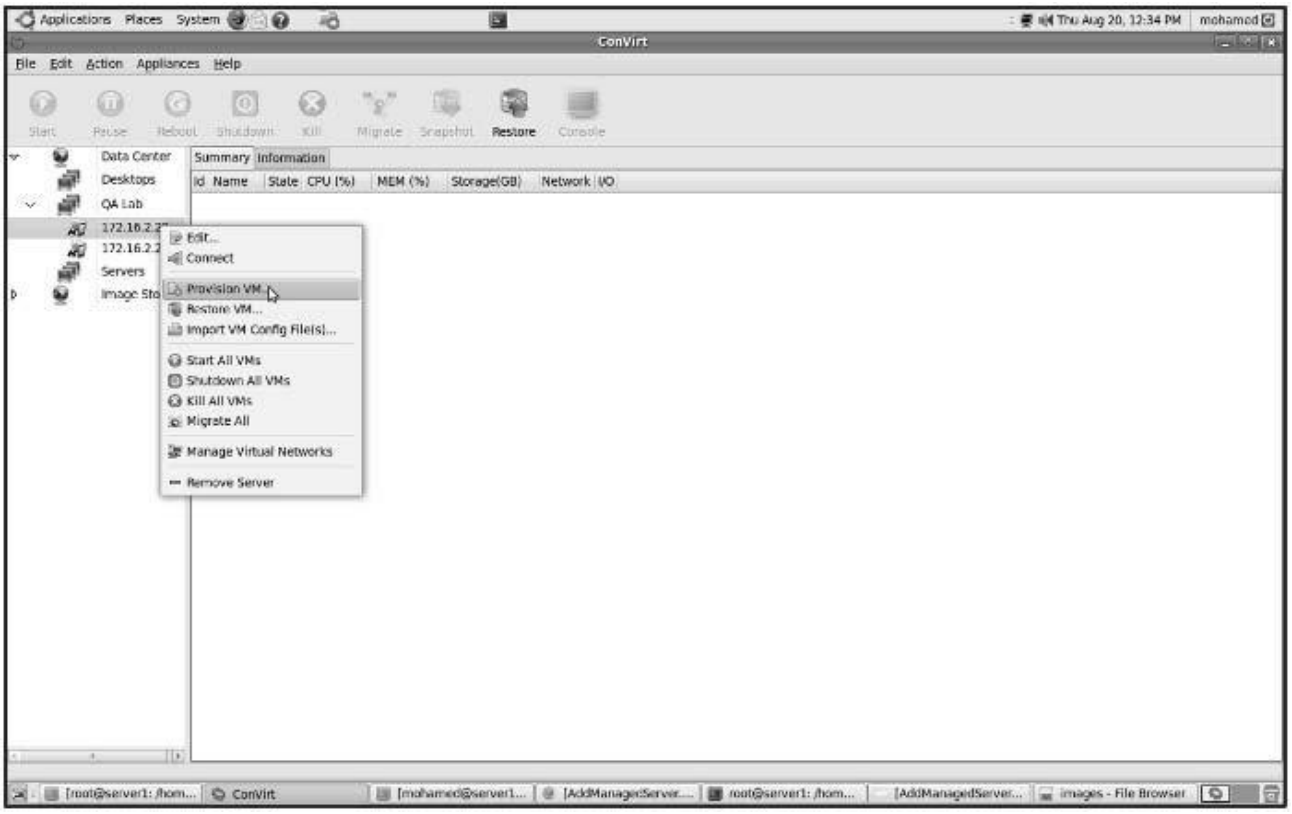


FIGURE 3.10. Provision a virtual machine.

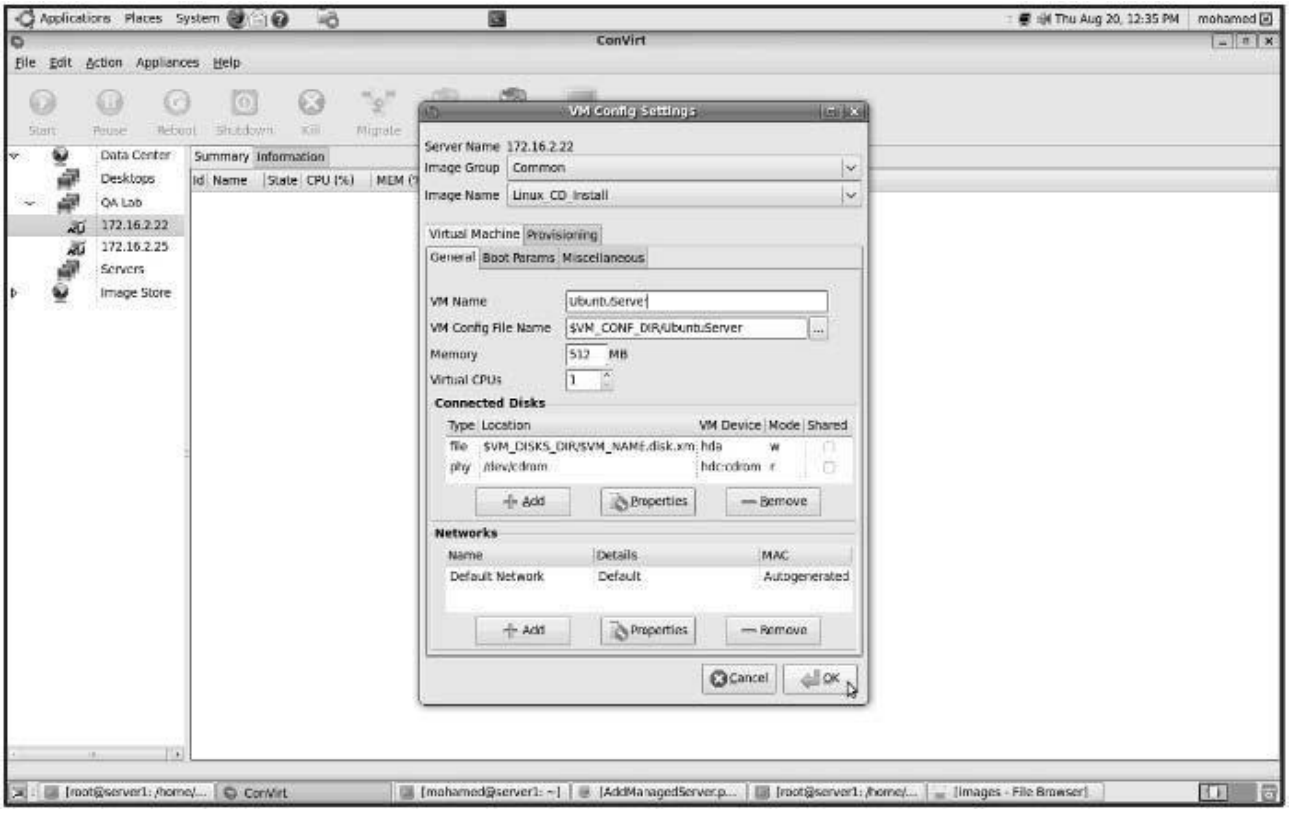


FIGURE 3.11 Configuring virtual machine.

- Start your VM (Figures 3.12 and 3.13), and make sure the installation media of the operating system you need is placed in drive, in order to use itfor booting the new VM and proceed in the installation process; then start the installation process as shown in Figure 3.14.
- Once the installation finishes, you can access your provisioned virtual machine from the consol icon on the top of your ConVirt management console.
- Reaching this step, you have created your first managed server and provisioned virtual machine. You can repeat the same procedure to add the second managed server in your pool to be ready for the next step of migrating one virtual machine from one server to the other.

3.8.1 VM Life Cycle and VM Monitoring

You can notice through working with ConVirt that you are able to manage the whole life cycle of the virtual machine; start, stop, reboot, migrate, clone, and so on. Also, you noticed how easy it is to monitor the resources of the managed server and to monitor the virtual machine's guests that help you balance and control the load on these managed servers once needed. In the next section, we are going to discuss how easy it is to migrate a virtual machine from host to host.

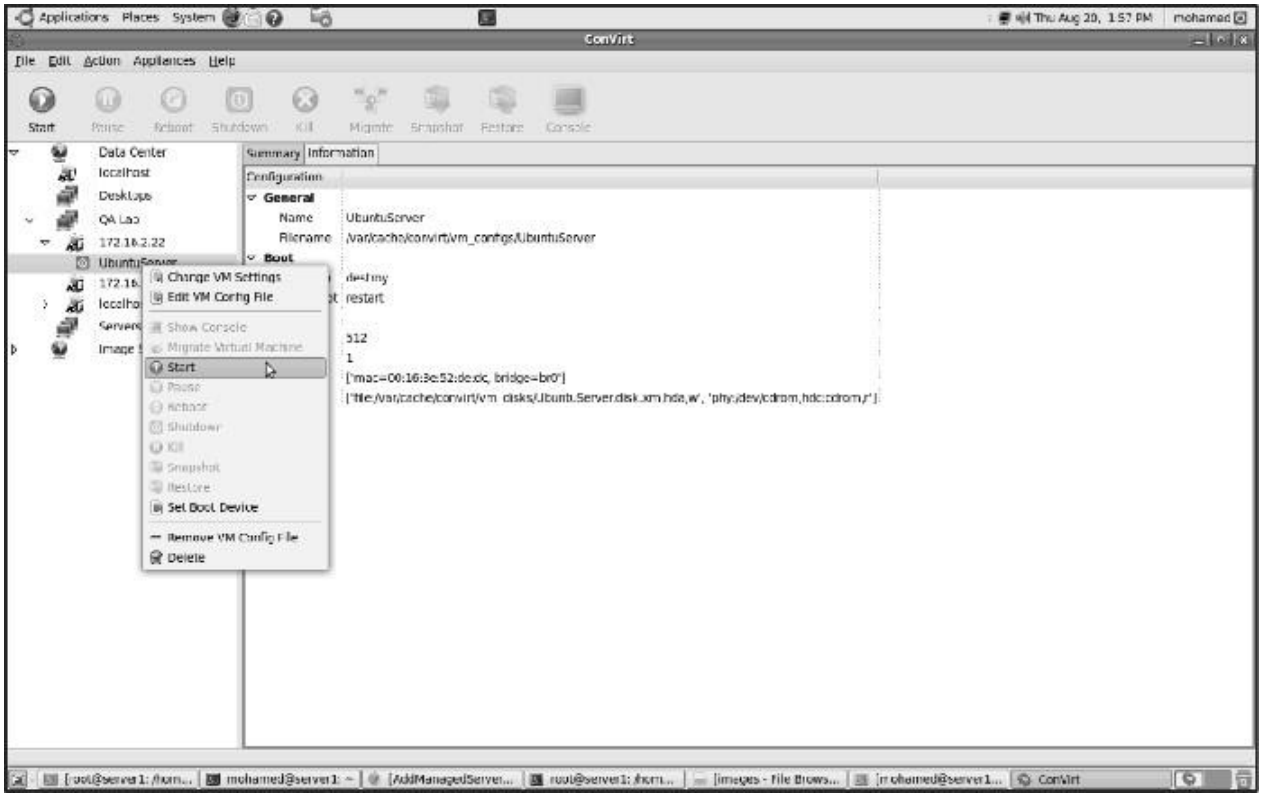


FIGURE 3.12. Provisioned VM ready to be started.

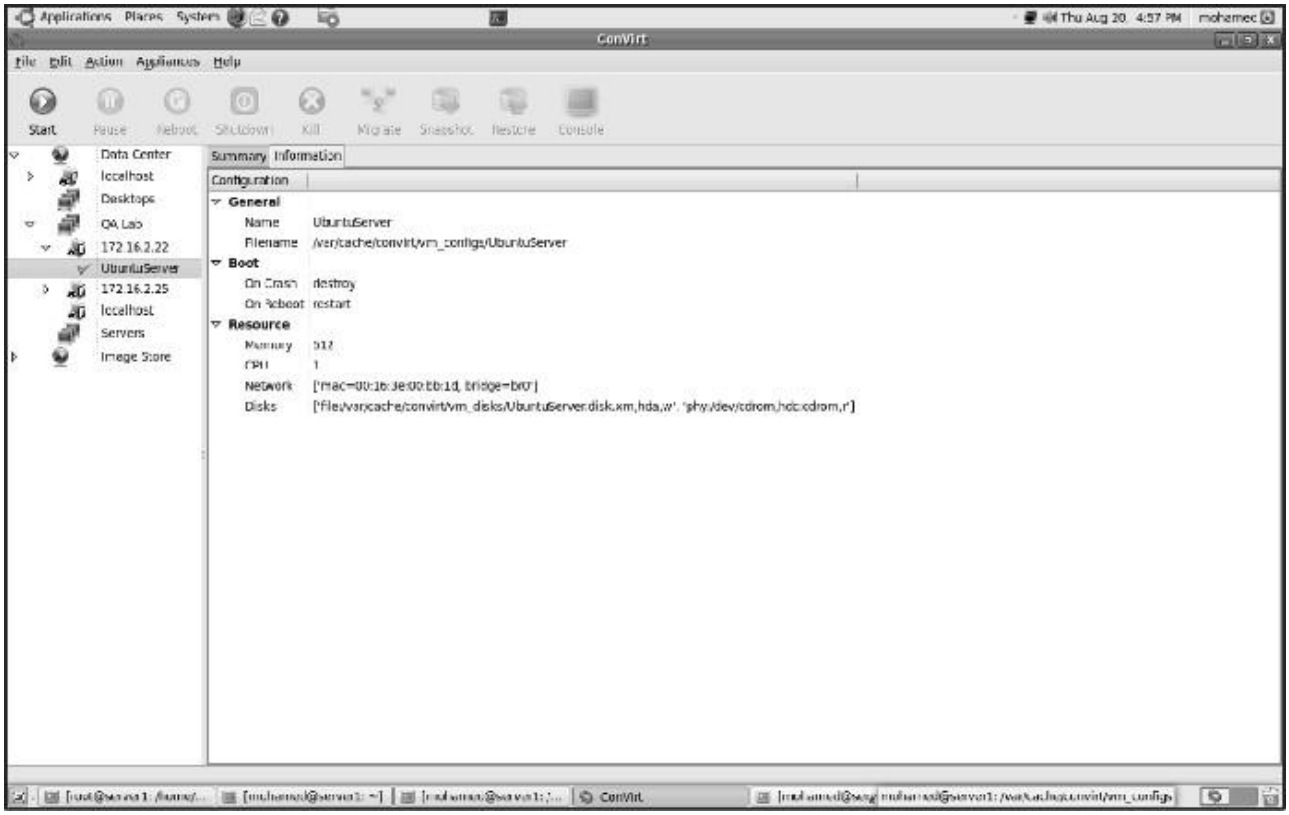


FIGURE 3.13. Provisioned VM started.

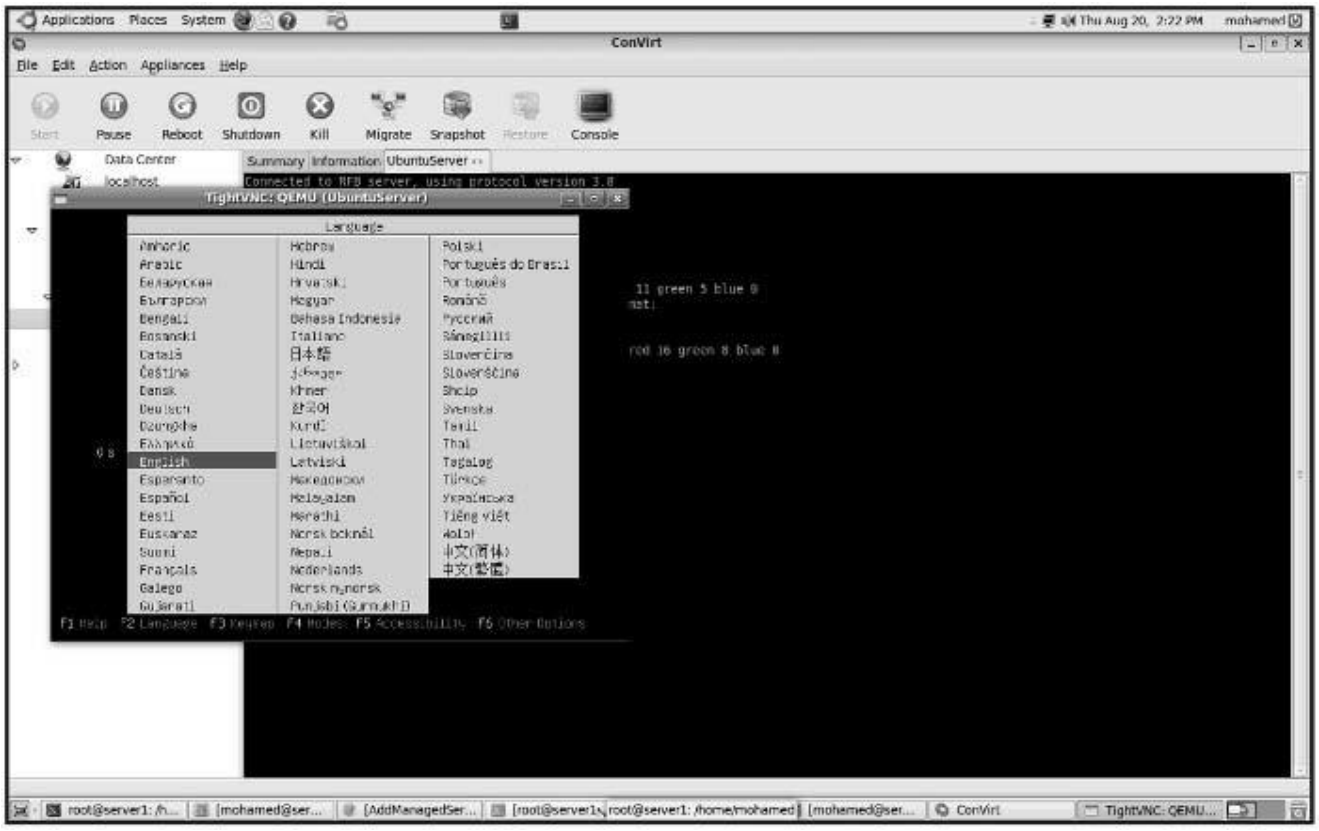


FIGURE 3.14. VM booting from the installation CD to start the installation process.

3.8.2 Live Migration

ConVirt tool allows running virtual machines to be migrated from one server to another .This feature makes it possible to organize the virtual machine to physical machine relationship to balance the workload; for example, a VM needing more CPU can be moved to a machine having available CPU cycles, or, in other cases, like taking the host machine for maintenance. For proper VM migration the following points must be considered:

- Shared storage for all Guest OS disks (e.g., NFS, or iSCSI).
- Identical mount points on all servers (hosts).
- The kernel and ram disk when using para-virtualized virtual machines should, also, be shared. (This is not required, if pygrub is used.)
- Centrally accessible installation media (iso).
- It is preferable to use identical machines with the same version of virtualization platform.
- Migration needs to be done within the same subnet.

Migration Process in ConVirt

- To start the migration of a virtual machine from one host to the other, select it and choose a migrating virtual machine, as shown in Figure 3.15.
- You will have a window containing all the managed servers in your data center (as shown in Figure 3.16). Choose one as a destination and start migration, or drag the VM and drop it on to another managed server to initiate migration.
- Once the virtual machine has been successfully placed and migrated to the destination host, you can see it still living and working (as shown in Figure 3.17).

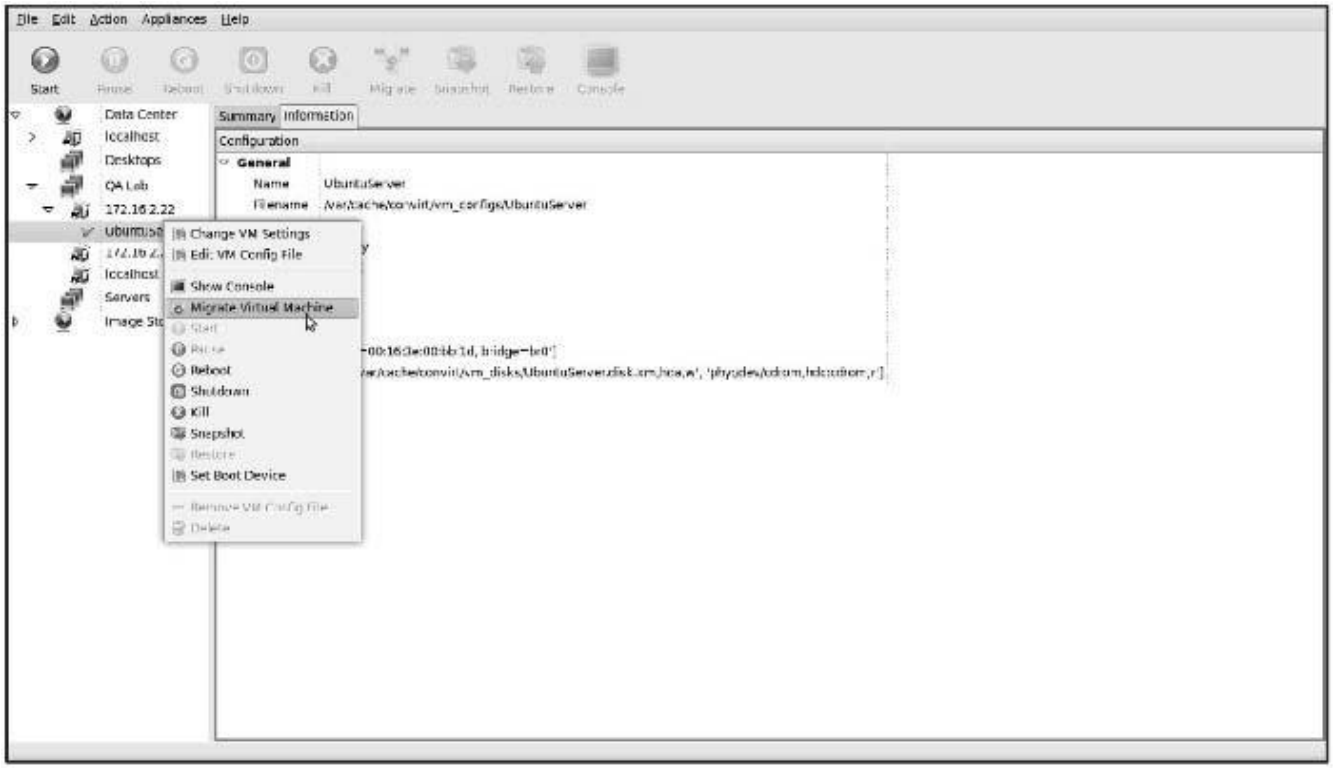


FIGURE 3.15. VM migration.

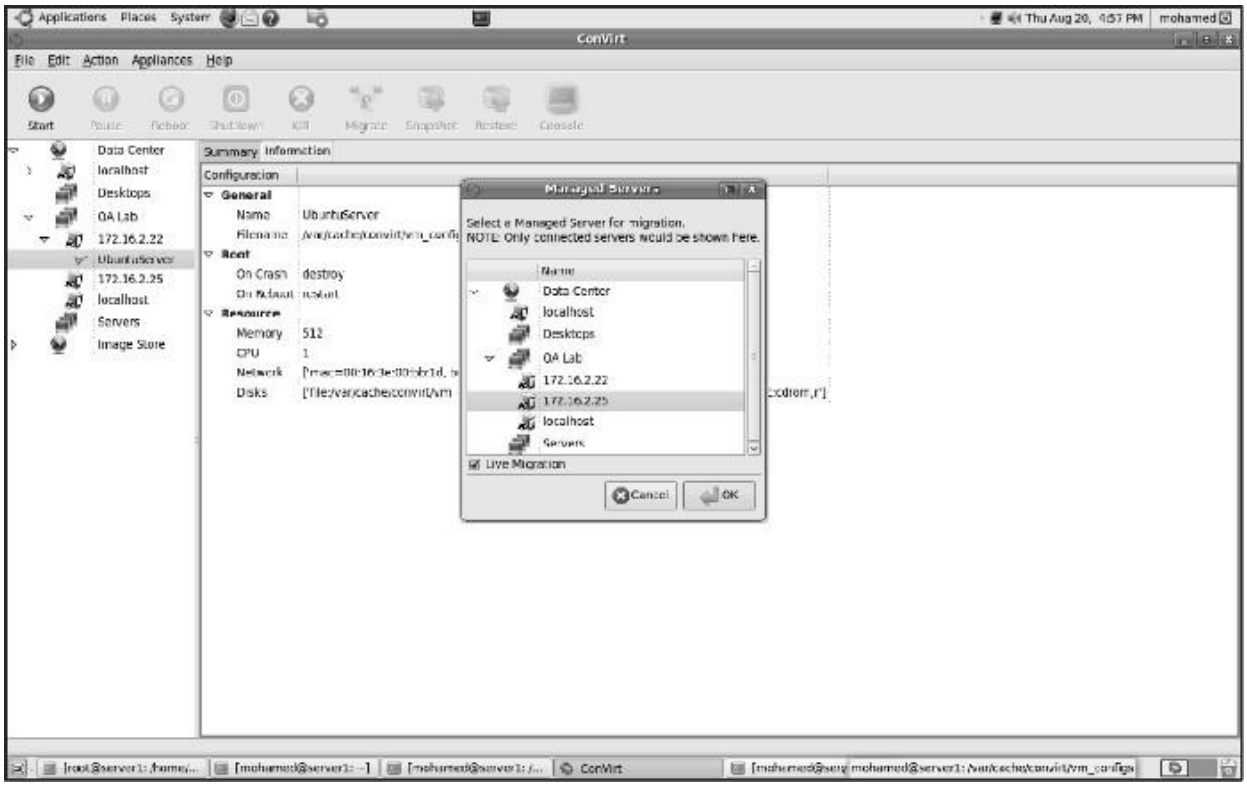


FIGURE 3.16. Select the destination managed server candidate for migration.

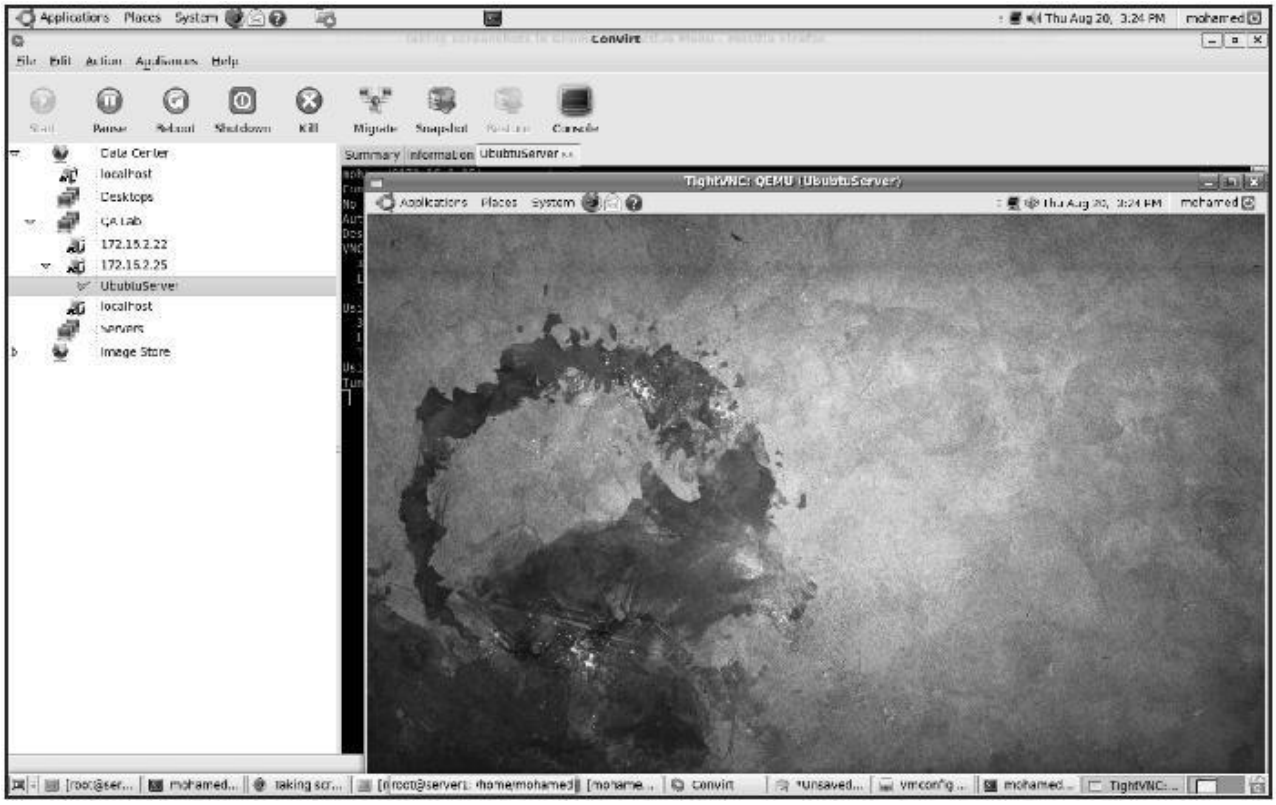


FIGURE 3.17. VM started on the destination server after migration.

3.8.3 Final Thoughts about the Example

This is just a demonstrating example of how to provision and migrate virtual machines; however, there are more tools and vendors that offer virtual infrastructure's management like Citrix Xen Server, VMware vSphere, and so on.

3.9 PROVISIONING IN THE CLOUD CONTEXT

In the cloud context, we shall discuss systems that provide the virtual machine provisioning and migration services; Amazon EC2 is a widely known example for vendors that provide public cloud services. Also, Eucalyptus and Open- Nebula are two complementary and enabling technologies for open source cloud tools, which play an invaluable role in infrastructure as a service and in building private, public, and hybrid cloud architecture.

Eucalyptus is a system for implementing on-premise private and hybrid clouds using the hardware and software's infrastructure, which is in place without modification. The current interface to Eucalyptus is compatible with Amazon's EC2, S3, and EBS interfaces, but the infrastructure is designed to support multiple client-side interfaces. Eucalyptus is implemented using commonly available Linux tools and basic Web service's technologies. Eucalyptus adds capabilities, such as end-user customization, self-service provisioning, and legacy application support to data center's virtualization's features, making the IT customer's service easier.

On the other hand, OpenNebula is a virtual infrastructure manager that orchestrates storage, network, and virtualization technologies to enable the dynamic placement of multi-tier services on distributed infrastructures, combining both data center's resources and remote cloud's resources according to allocation's policies. OpenNebula provides internal cloud administration and user's interfaces for the full management of the cloud's platform.

3.9.1 Amazon Elastic Compute Cloud

The Amazon EC2 (Elastic Compute Cloud) is a Web service that allows users to provision new machines into Amazon's virtualized infrastructure in a matter of minutes; using a publicly available API (application programming interface), it reduces the time required to obtain and boot a new server. Users get full root access and can install almost any OS or application in their AMIs (Amazon Machine Images). Web services APIs allow users to reboot their instances remotely, scale capacity quickly, and use on-demand service when needed; by adding tens, or even hundreds, of machines. It is very important to mention that there is no up-front hardware setup and there are no installation costs, because Amazon charges only for the capacity you actually use. EC2 instance is typically a virtual machine with a certain amount of RAM, CPU, and storage capacity.

Setting up an EC2 instance is quite easy. Once you create your AWS (Amazon Web service) account, you can use the on-line AWS console, or simply download the offline command line's tools to start provisioning your instances.
Amazon EC2 provides its customers with three flexible purchasing models to make it easy for the cost optimization:

- On-Demand instances, which allow you to pay a fixed rate by the hour with no commitment.
- Reserved instances, which allow you to pay a low, one-time fee and in turn receive a significant discount on the hourly usage charge for that instance. It ensures that any reserved instance you launch is guaranteed to succeed(provided that you have

booked them in advance). This means that users of these instances should not be affected by any transient limitations inEC2 capacity.

- Spot instances, which enable you to bid whatever price you want for instance capacity, providing for even greater savings, if your applications have flexible start and end times.

Amazon and Provisioning Services. Amazon provides an excellent set of tools that help in provisioning service; Amazon Auto Scaling is a set of command line tools that allows scaling Amazon EC2 capacity up or down automatically and according to the conditions the end user defines. This feature ensures that the number of Amazon EC2 instances can scale up seamlessly during demand spikes to maintain performance and can scale down automatically when loads diminish and become less intensive to minimize the costs. Auto Scaling service and CloudWatch (a monitoring service for AWS cloud resources and their utilization) help in exposing functionalities required for provisioning application services on Amazon EC2.

Amazon Elastic Load Balancer is another service that helps in building fault-tolerant applications by automatically provisioning incoming application workload across available Amazon EC2 instances and in multiple availability zones.

3.9.2 Infrastructure Enabling Technology

Offering infrastructure as a service requires software and platforms that can manage the Infrastructure that is being shared and dynamically provisioned. For this, there are three noteworthy technologies to be considered: Eucalyptus, OpenNebula, and Aneka.

3.9.3 Eucalyptus

Eucalyptus is an open-source infrastructure for the implementation of cloud computing on computer clusters. It is considered one of the earliest tools developed as a surge computing (in which data center's private cloud could augment its ability to handle workload's spikes by a design that allows it to send overflow work to a public cloud) tool. Its name is an acronym for "elastic utility computing architecture for linking your programs to useful systems." Here are some of the Eucalyptus features:

- Interface compatibility with EC2, and S3 (both Web service and Query/ REST interfaces).
- Simple installation and deployment.
- Support for most Linux distributions (source and binary packages).
- Support for running VMs that run atop the Xen hypervisor or KVM. Support for other kinds of VMs, such as VMware, is targeted for future releases.
- Secure internal communication using SOAP with WS security.
- Cloud administrator's tool for system's management and user's accounting.
- The ability to configure multiple clusters each with private internal network addresses into a single cloud.

Eucalyptus aims at fostering the research in models for service's provisioning, scheduling, SLA formulation, and hypervisors' portability.

Eucalyptus Architecture. Eucalyptus architecture, as illustrated in Figure 3.18, constitutes each high-level system's component as a stand-alone Web service with the following high-level components.

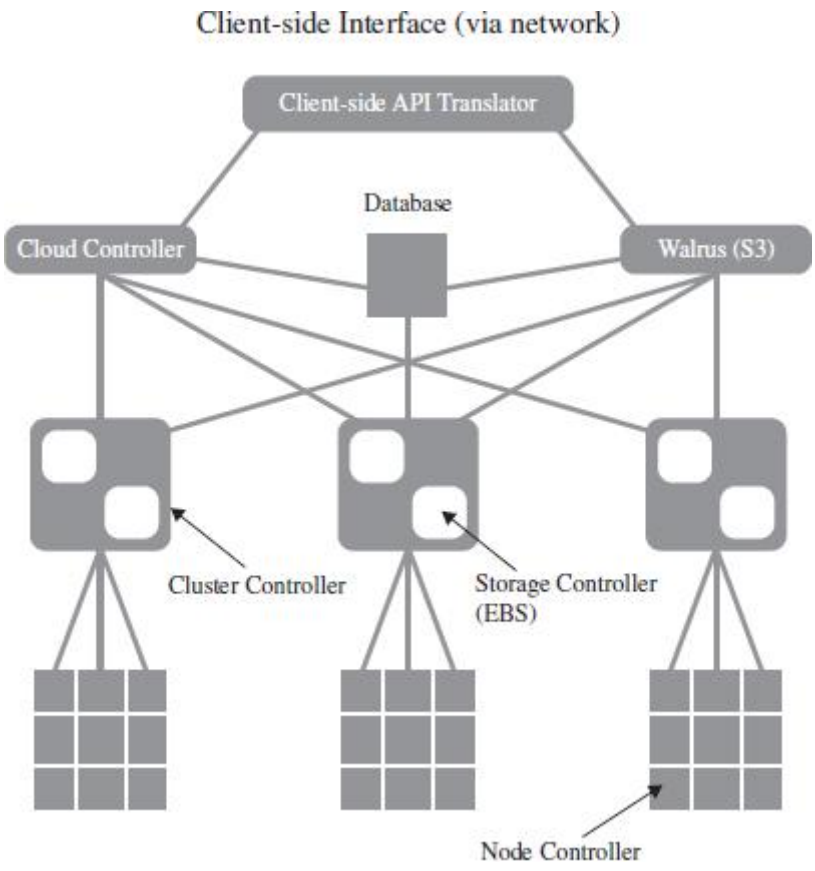


FIGURE 3.18 Eucalyptus high-level Architecture

- **Node controller (NC)** controls the execution, inspection, and termination of VM instances on the host where it runs.
- **Cluster controller (CC)** gathers information about and schedules VM execution on specific node controllers, as well as manages virtual instance network.
- **Storage controller (SC)** is a put/get storage service that implements Amazon's S3 interface and provides a way for storing and accessing VM images and user data.
- **Cloud controller (CLC)** is the entry point into the cloud for users and administrators. It queries node managers for information about resources, makes high-level scheduling decisions, and implements them by making requests to cluster controllers.
- **Walrus (W)** is the controller component that manages access to the storage services within Eucalyptus. Requests are communicated to Walrus using the SOAP or REST-based interface.

Its design is an open and elegant one. It can be very beneficial in testing and debugging purposes before deploying it on a real cloud. For more details about Eucalyptus architecture and design.

Ubuntu Enterprise Cloud and Eucalyptus. Ubuntu Enterprise Cloud (UEC) is a new initiative by Ubuntu to make it easier to provision, deploy, configure, and use cloud infrastructures based on Eucalyptus. UEC brings Amazon EC2-like infrastructure's capabilities inside the firewall.

This is by far the simplest way to install and try Eucalyptus. Just download the Ubuntu server version and install it wherever you want. UEC is also the first open source project that lets you create cloud services in your local environment easily and leverage the power of cloud computing.

3.9.4 VM Dynamic Management Using OpenNebula

OpenNebula is an open and flexible tool that fits into existing data center's environments to build any type of cloud deployment. OpenNebula can be primarily used as a virtualization tool to manage your virtual infrastructure, which is usually referred to as private cloud. OpenNebula supports a hybrid cloud to combine local infrastructure with public cloud-based infrastructure, enabling highly scalable hosting environments. OpenNebula also supports public clouds by providing cloud's interfaces to expose its functionality for virtual machine, storage, and network management. OpenNebula is one of the technologies being enhanced in the Reservoir Project, European research initiatives in virtualized infrastructures, and cloud computing.

OpenNebula architecture is shown in Figure 3.19, which illustrates the existence of public and private clouds and also the resources being managed by its virtual manager.

OpenNebula is an open-source alternative to these commercial tools for the dynamic management of VMs on distributed resources. This tool is supporting several research lines in advance reservation of capacity, probabilistic admission control, placement optimization, resource models for the efficient management of groups of virtual machines, elasticity support, and so on. These research lines address the requirements from both types of clouds namely, private and public.

OpenNebula and Haizea. Haizea is an open-source virtual machine-based lease management architecture developed by Sotomayor et al.; it can be used as a scheduling backend for OpenNebula. Haizea uses leases as a fundamental resource provisioning abstraction and implements those leases as virtual machines, taking into account the overhead of using virtual machines when scheduling leases. Haizea also provides advanced functionality such as:

- Advance reservation of capacity.
- Best-effort scheduling with backfilling.
- Resource preemption (using VM suspend/resume/migrate).
- Policy engine, allowing developers to write pluggable scheduling policies in Python.

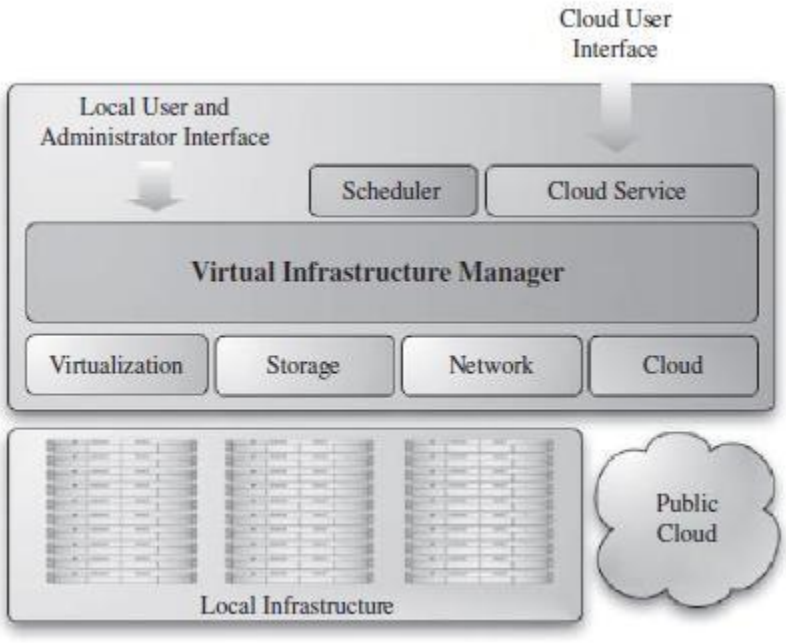


FIGURE 3.19 OpenNebula high-level architecture

3.9.5 Aneka

Manjrasoft Aneka is a .NET-based platform and framework designed for building and deploying distributed applications on clouds. It provides a set of APIs for transparently exploiting distributed resources and expressing the business logic of applications by using the preferred programming abstractions. Aneka is also a market-oriented cloud platform since it allows users to build and schedule applications, provision resources, and monitor results using pricing, accounting, and QoS/SLA services in private and/or public cloud environments.

It allows end users to build an enterprise/private cloud setup by exploiting the power of computing resources in the enterprise data centers, public clouds such as Amazon EC2, and hybrid clouds by combining enterprise private clouds managed by Aneka with resources from Amazon EC2 or other enterprise clouds built and managed using technologies such as Xen Server.

Aneka also provides support for deploying and managing clouds. By using its Management Studio and a set of Web interfaces, it is possible to set up either public or private clouds, monitor their status, update their configuration, and perform the basic management operations.

Aneka Architecture. Aneka platform architecture, as illustrated in Figure 3.20, consists of a collection of physical and virtualized resources connected through a network. 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.

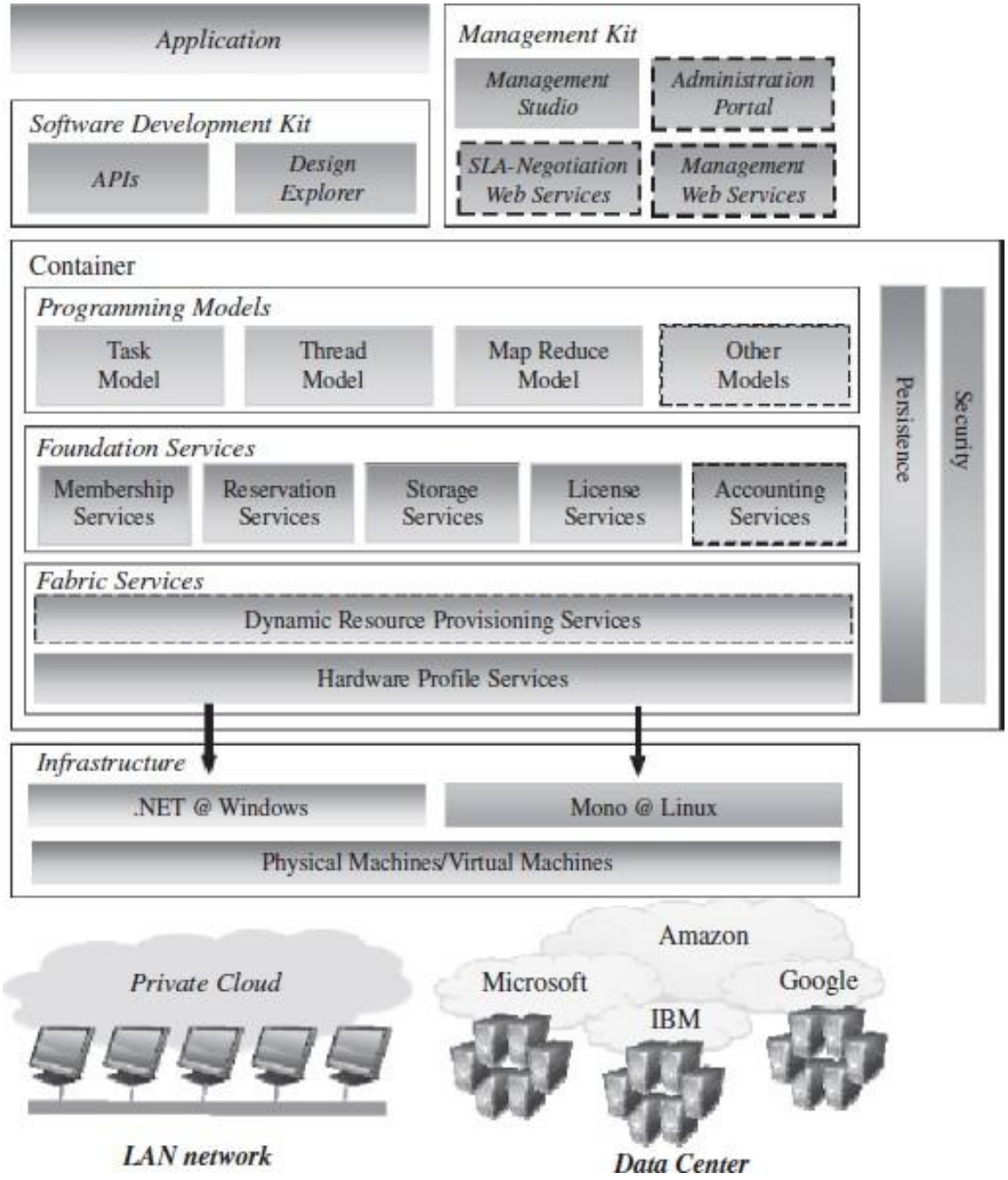


FIGURE 3.20 Manjrasoft Aneka layered Architecture

UNIT – IV

Managing & Securing the Cloud: Administrating the Clouds, Cloud Management Products, Emerging Cloud Management Standards, Securing the Cloud, Securing Data, Establishing Identity and Presence

4.1 MANAGING & SECURING THE CLOUD:

Managing the Cloud:

Cloud computing deployments must be monitored and managed in order to be optimized for best performance. To the problems associated with analyzing distributed network applications, the cloud adds the complexity of virtual infrastructure. This is one of the most active areas of product development in the entire cloud computing industry, and this chapter introduces you to the different products in this nascent area. Cloud management software provides capabilities for managing faults, configuration, accounting, performance, and security; this is referred to as FCAPS. Many products address one or more of these areas, and through network frameworks, you can access all five areas. Framework products are being repositioned to work with cloud systems. Your management responsibilities depend on the particular service model for your cloud deployment. Cloud management includes not only managing resources in the cloud, but managing resources on-premises. The management of resources in the cloud requires new technology, but management of resources on-premises allows vendors to use well-established network management technologies.

The lifecycle of a cloud application includes six defined parts, and each must be managed. In this chapter, the tasks associated with each stage are described. Efforts are underway to develop cloud management interoperability standards. One effort you learn about in this chapter is the DMTF's (Distributed Management Task Force) Open Cloud Standards Incubator. The goal of these efforts is to develop management tools that work with any cloud type. Another group called the Cloud Commons is developing a technology called the Service Measurement Index (SMI). SMI aims to deploy methods for measuring various aspects of cloud performance in a standard way.

4.2 ADMINISTRATING THE CLOUDS:

The explosive growth in cloud computing services has led many vendors to rename their products and reposition them to get in on the gold rush in the clouds. What was once a network management product is now a cloud management product. Nevertheless, this is one area of technology that is very actively funded, comes replete with interesting startups, has been the focus of several recent strategic acquisitions, and has resulted in some interesting product alliances. Let's join the party and see what all the fuss is about.

These fundamental features are offered by traditional network management systems:

- Administration of resources
- Configuring resources
- Enforcing security
- Monitoring operations
- Optimizing performance
- Policy management
- Performing maintenance
- Provisioning of resources

Network management systems are often described in terms of the acronym FCAPS, which stands for these features:

- Fault
- Configuration
- Accounting
- Performance
- Security

Most network management packages have one or more of these characteristics; no single package provides all five elements of FCAPS.

To get the complete set of all five of these management areas from a single vendor, you would need to adopt a network management framework. These large network management frameworks were industry leaders several years back: BMC PATROL, CA Unicenter, IBM Tivoli, HP OpenView, and Microsoft System Center. Network framework products have been sliced and diced in many different ways over the years, and they are rebranded from time to time. Today, for example, BMC PATROL is now part of BMC ProactiveNet Performance Management (http://www.bmc.com/products/product-listing/ProactiveNet-Performance-Management.html), HP OpenView has been split (https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-10^36657_4000_100) into a set of HP Manager products.

The impact that cloud computing is having on network frameworks is profound. These five vendors have (or soon will have) products for cloud management. Computer Associates, for example, has completely repositioned its network management portfolio as an IT Management Software as a Service. Find the cloud products for these five large cloud vendors at the following URLs:

- BMC Cloud Computing (http://www.bmc.com/solutions/esm-initiative/ cloud-computing.html)
- Computer Associates Cloud Solutions (http://www.ca.com/us/cloudcomputing. aspx)
- HP Cloud Computing (http://h20338.www2.hp.com/enterprise/w1/en/ technologies/cloud-computing-overview.html)
- IBM Cloud Computing (http://www.ibm.com/ibm/cloud/)
- Microsoft Cloud Services (http://www.microsoft.com/cloud/)

Figure 4.1 Tivoli Service Automation Manager lets you create and stage cloud-based servers

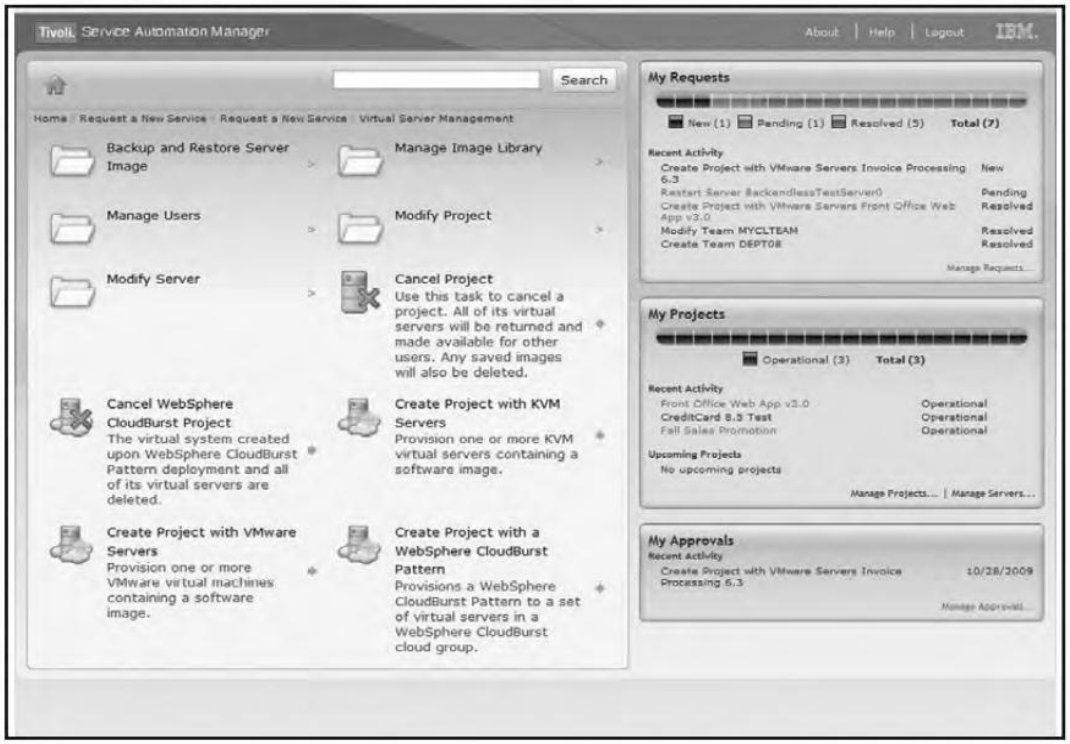


Figure 4.1 shows IBM Tivoli Service Automation Manager, a framework tool for managing cloud infrastructure.

4.2.1 Management responsibilities:

What separates a network management package from a cloud computing management package is the ―cloudly‖ characteristics that cloud management service must have:

- Billing is on a pay-as-you-go basis.
- The management service is extremely scalable.
- The management service is ubiquitous.
- Communication between the cloud and other systems uses cloud networking standards.

To monitor an entire cloud computing deployment stack, you monitor six different categories:
1. End-user services such as HTTP, TCP, POP3/SMTP, and others
2. Browser performance on the client
3. Application monitoring in the cloud, such as Apache, MySQL, and so on
4. Cloud infrastructure monitoring of services such as Amazon Web Services, GoGrid, Rackspace, and others
5. Machine instance monitoring where the service measures processor utilization, memory usage, disk consumption, queue lengths, and other important parameters
6. Network monitoring and discovery using standard protocols like the Simple Network Management Protocol (SNMP), Configuration Management Database (CMDB), Windows Management Instrumentation (WMI), and the like

It's important to note that there are really two aspects to cloud management:

- ✓ Managing resources ***in the cloud***
- ✓ Using the cloud to manage resources ***on-premises***

When you move to a cloud computing architecture from a traditional networked model like client/ server or a three-tier architecture, many of the old management tasks for processes going on in the cloud become irrelevant or nearly impossible to manage because the tools to effectively manage resources of various kinds fall outside of your own purview. In the cloud, the particular service model you are using directly affects the type of monitoring you are responsible for.

Consider the case of an Infrastructure as a Service vendor such as Amazon Web Services or Rackspace. You can monitor your usage of resources either through their native monitoring tools like Amazon CloudWatch or Rackspace Control Panel or through the numerous third-party tools that work with these sites' APIs. In IaaS, you can alter aspects of your deployment, such as the number of machine instances you are running or the amount of storage you have, but you have very limited control over many important aspects of the operation. For example, your network bandwidth is locked into the type of instance you deploy. Even if you can provision more bandwidth, you likely have no control over how network traffic flows into and out of the system, whether there is packet prioritization, how routing is done, and other important characteristics.

The situation—as you move first to Platform as a Service (PaaS) like Windows Azure or Google App Engine and then onto Software as a Service (SaaS) for which Salesforce.com is a prime example—becomes even more restrictive. When you deploy an application on Google's PaaS App Engine cloud service, the Administration Console provides you with the following monitoring capabilities:

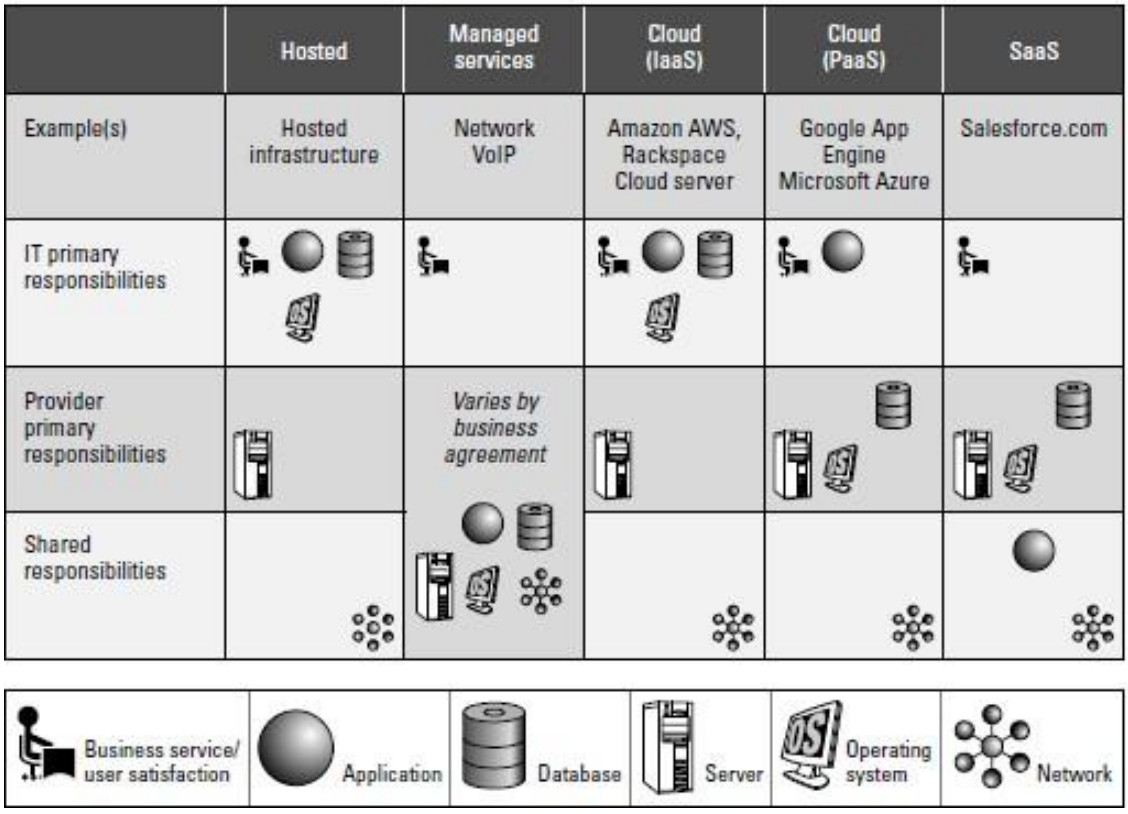
- Create a new application, and set it up in your domain.
- Invite other people to be part of developing your application.
- View data and error logs.
- Analyze your network traffic.
- Browse the application datastore, and manage its indexes.
- View the application's scheduled tasks.
- Test the application, and swap out versions.

However, you have almost no operational control. Essentially, Google App Engine lets you deploy the application and monitor it, and that's about it. All the management of devices, networks, and other aspects of the platform are managed by Google. You have even less control when you are selling software in the cloud, as you would with Salesforce.com.

Figure 11.2 graphically summarizes the management responsibilities by service model type.

FIGURE 11.2

Management responsibilities by service model type



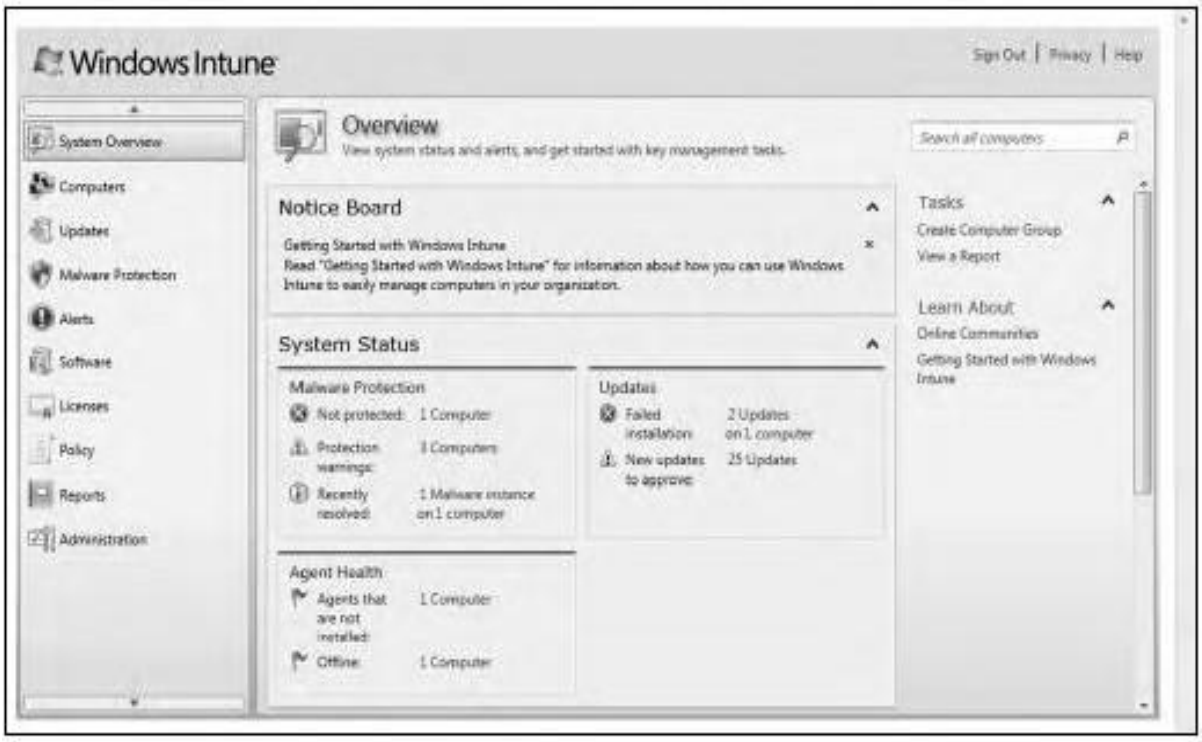
The second aspect of cloud management is the role that cloud-based services can play in managing on-premises resources. From the standpoint of the client, a cloud service provider is no different than any other networked service. The full range of network management capabilities may be brought to bear to solve mobile, desktop, and local server issues, and the same sets of tools can be used for measurement. Microsoft System Center is an example of how management products are being adapted for the cloud. System Center provides tools for managing Windows servers and desktops. The management services include an Operations Manager, the Windows Service Update Service (WSUS), a Configuration Manager for asset management, a Data Protection Manager, and a Virtual Machine Manager, among other components.

One of these service sets was called the System Center Online Desktop Manager (SCODM). Microsoft has taken SCODM and repositioned it as a cloud-based service for managing updates, monitoring PCs for license compliance and health, enforcing security policies, and using Forefront protect systems from malware, and the company has branded it as Windows Intune (http://www.microsoft.com/windows/windowsintune/default.aspx). From the client's standpoint, it makes little difference whether the service is in the cloud or on a set of servers in a datacenter. The benefit of a cloud management service accrues to the organization responsible for managing the

desktops or mobile devices. **Figure 11.3** shows an Overview screen from the beta version of Windows Intune. The product is due to be released in the first or second quarter of 2011.

FIGURE 11.3

Intune is Microsoft's cloud-based management service for Windows systems.



4.2.2 Lifecycle management:

Cloud services have a defined lifecycle, just like any other system deployment. A management program has to touch on each of the six different stages in that lifecycle:

1. The definition of the service as a template for creating instances
   Tasks performed in Phase 1 include the creation, updating, and deletion of service templates.
2. Client interactions with the service, usually through an SLA (Service Level Agreement) contract
   This phase manages client relationships and creates and manages service contracts.
3. The deployment of an instance to the cloud and the runtime management of instances
   Tasks performed in Phase 3 include the creation, updating, and deletion of service offerings.
4. The definition of the attributes of the service while in operation and performance of modifications of its properties

   The chief task during this management phase is to perform service optimization and customization.
5. Management of the operation of instances and routine maintenance

   During Phase 5, you must monitor resources, track and respond to events, and perform reporting and billing functions.
6. Retirement of the service

   End of life tasks include data protection and system migration, archiving, and service contract termination.

4.3 CLOUD MANAGEMENT PRODUCTS:

Cloud management software and services is a very young industry, and as such, it has a very large number of companies, some with new products and others with older products competing in this area. Table 11.1 shows some of the current players in this market, along with the products they either are offering or are promising in the very near future. When considering products in cloud management, you should be aware that—as in all new areas of technology—there is considerable churn as companies grow, get acquired, or fail along the way. It is entirely possible that if you return to this list a year or two after this book is published, half of these products or services will no longer exist as listed; you should keep this in mind.

TABLE 11.1

Cloud and Web Monitoring Solutions

Product	URL	Description
AbiCloud	http://www.abiquo.com/	Virtual machine conversion and management
Amazon CloudWatch	http://aws.amazon.com/cloudwatch/	AWS dashboard
BMC Cloud Computing Initiative	http://www.bmc.com/solutions/esm-initiative/cloud-computing.html	Cloud planning, lifecycle management, optimization, and guidance
CA Cloud Connected Management Suite	http://www.ca.com/us/cloud-solutions.aspx	CA Cloud Insight, CA Cloud Compose, CA Cloud Optimize, and CA Cloud Orchestrate are described below
Cacti	http://www.cacti.net/	Network performance graphing solution
CloudKick	https://www.cloudkick.com/	Cloud server monitoring
Dell Scalent	http://www.scalent.com/index.php	Virtualization provisioning system that will be rolled into Dell's Advanced Infrastructure Manager (AIM)
Elastra	http://www.elastra.com/	Federated hybrid cloud management software
Ganglia	http://ganglia.info/	Distributed network monitoring software
Gomez	http://www.gomez.com/	Web site monitoring and analytics
HP Cloud Computing	http://h20338.www2.hp.com/enterprise/w1/en/technologies/cloud-computing-overview.html	A variety of management products and services, both released and under development
Hyperic	http://www.hyperic.com/	Performance management for virtualized Java Apps with VMware integration
IBM Service Management and Cloud Computing	http://www-01.ibm.com/software/tivoli/solutions/cloudcomputing/	Various IBM Tivoli managers and monitors
Internetseer	http://www.internetseer.com/home/index.xtp	Web site monitoring service
Intune	http://www.microsoft.com/windows/windowsintune/default.aspx	Cloud-based Windows system management
Keynote	http://www.keynote.com/	Web, mobile, streaming, and customer test and measurement products

Product	URL	Description
ManageEngine OpManager	http://www.manageengine.com/network-performance-management.html	Network and server monitoring, server desk, event and security management
ManageIQ	http://www.manageiq.com/	Enterprise Virtualization Management Suite (EVM) that provides monitoring, provisioning, and cloud integration services
Managed Methods JaxView	http://managedmethods.com/	SOA management tool
Monit	http://mmonit.com/monit/	Unix system monitoring and management
Montis	http://portal.monitis.com/index.php/home	Cloud-based monitoring service
Morph	http://mor.ph/	Infrastructure management, provisioning, deployment, and monitoring tools
Nagios	http://www.nagios.org/	Network monitoring system
NetIQ	http://www.netiq.com/	Network management, monitoring, deployment, and security software
New Relic RPM	http://www.newrelic.com/	Java and Ruby application monitor and troubleshooting
Nimsoft	http://www.ca.com/us/products/detail/CA-Nimsoft-Monitoring-Solution.aspx	Cloud monitoring software
OpenQRM	http://www.openqrm.com/	Data center management platform
Pareto Networks	http://www.paretonetworks.com/	Cloud provisioning and deployment
Pingdom	http://www.pingdom.com/	Web site and server uptime and performing monitoring
RightScale	http://www.rightscale.com/	Automated virtual server scaling
ScienceLogic	http://www.sciencelogic.com/	Datacenter and cloud management solutions and appliances
Scout	http://scoutapp.com/	Hosted server management service
ServiceUptime	http://www.serviceuptime.com/	Web site monitoring service
Site24X7	http://site24x7.com/	Web site monitoring service
Solarwinds	http://www.solarwinds.com/	Network monitoring and management software

continued

TABLE 11.1 *(continued)*

Product	URL	Description
Tapinsystems	http://www.tapinsystems.com/home	Provisioning and management service
Univa UD	http://univaud.com/index.php	Application and infrastructure management software for hybrid multi-clouds
VMware Hyperic	http://www.springsource.com/	Performance management for VMware deployed Java applications
Webmetrics	http://www.webmetrics.com/	Web performance management, load testing, and application monitor for cloud services
WebSitePulse	http://www.websitepulse.com/	Server, Web site, and application monitoring service
Whatsup Gold	http://www.whatsupgold.com/	Network monitoring and management software
Zenoss	http://www.zenoss.com/	IT operations monitoring
Zeus	http://www.zeus.com/	Web-based application traffic manager

The core management features offered by most cloud management service products include the following:

- Support of different cloud types
- Creation and provisioning of different types of cloud resources, such as machine instances, storage, or staged applications
- Performance reporting including availability and uptime, response time, resource quota usage, and other characteristics
- The creation of dashboards that can be customized for a particular client's needs

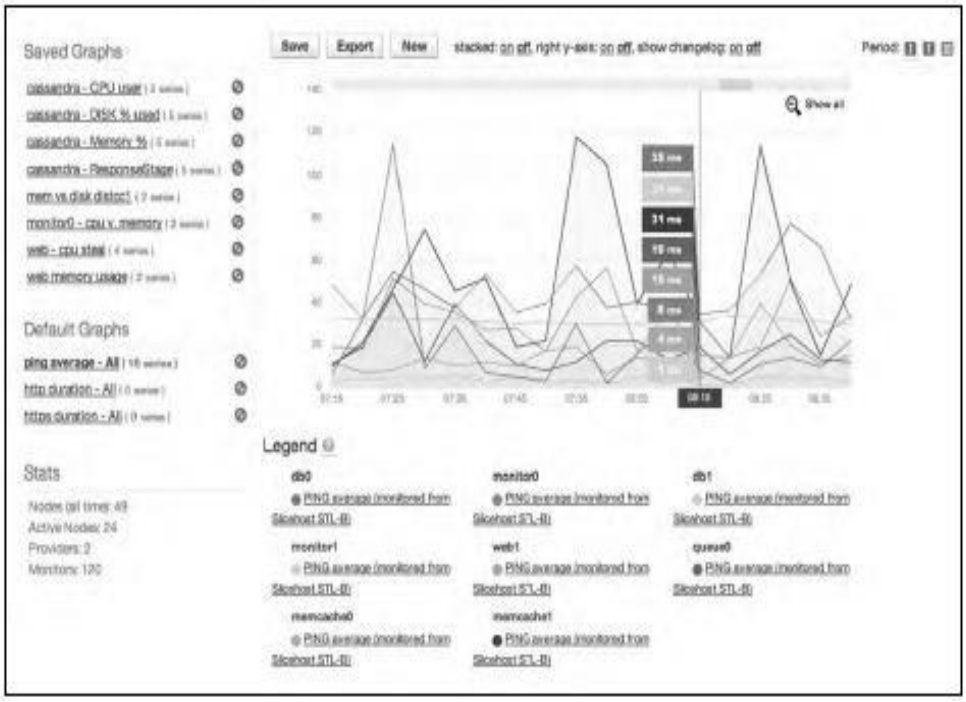
Automated deployment on IaaS systems represents one class of cloud management services. One of the more interesting and successful vendors in this area is Rightscale (http://www.rightscale.com/) whose software allows clients to stage and manage applications on AWS (Amazon Web Service), Eucalyptus, Rackspace, and the Chef Multicloud framework or a combination of these cloud types. Rightscale creates cloud-ready server templates and provides the automation and orchestration necessary to deploy them. Eucalyptus and Rackspace both use Amazon EC2 and S3 services, although Eucalyptus is open source and portable. RightScale server templates and the Rightscript technology are highly configurable and can be run under batch control. The RightScale user interface also provides real-time measurements of individual server instances.

Cloudkick (https://www.cloudkick.com/) is another infrastructure monitoring solution that is well regarded. Its service is noted for being agnostic and working with multiple vendor cloud platforms. The Cloudkick user interface is designed for rapid deployment assessment, and its at-a-glance-monitoring Insight module is particularly easy to use. Figure 11.4 shows the Insight module, and Figure 11.5 shows Cloudkick's real-time server visualization tool, which is one of the more interesting presentation tools we've seen. In Figure 11.5, the circles are servers with their location on the different axes based on observed metrics. Powerful servers are larger circles, and the colors indicate the current state of the server.

Users have commented on Cloudkick's instant launching being difficult, and both Cloudkick and RightScale are known to be easy to use with Linux virtual servers and less so with Windows instances.

FIGURE 11.4

Cloudkick's Insight module (https://www.cloudkick.com/site_media/images/graphs2.png) is powerful and particularly easy to use.

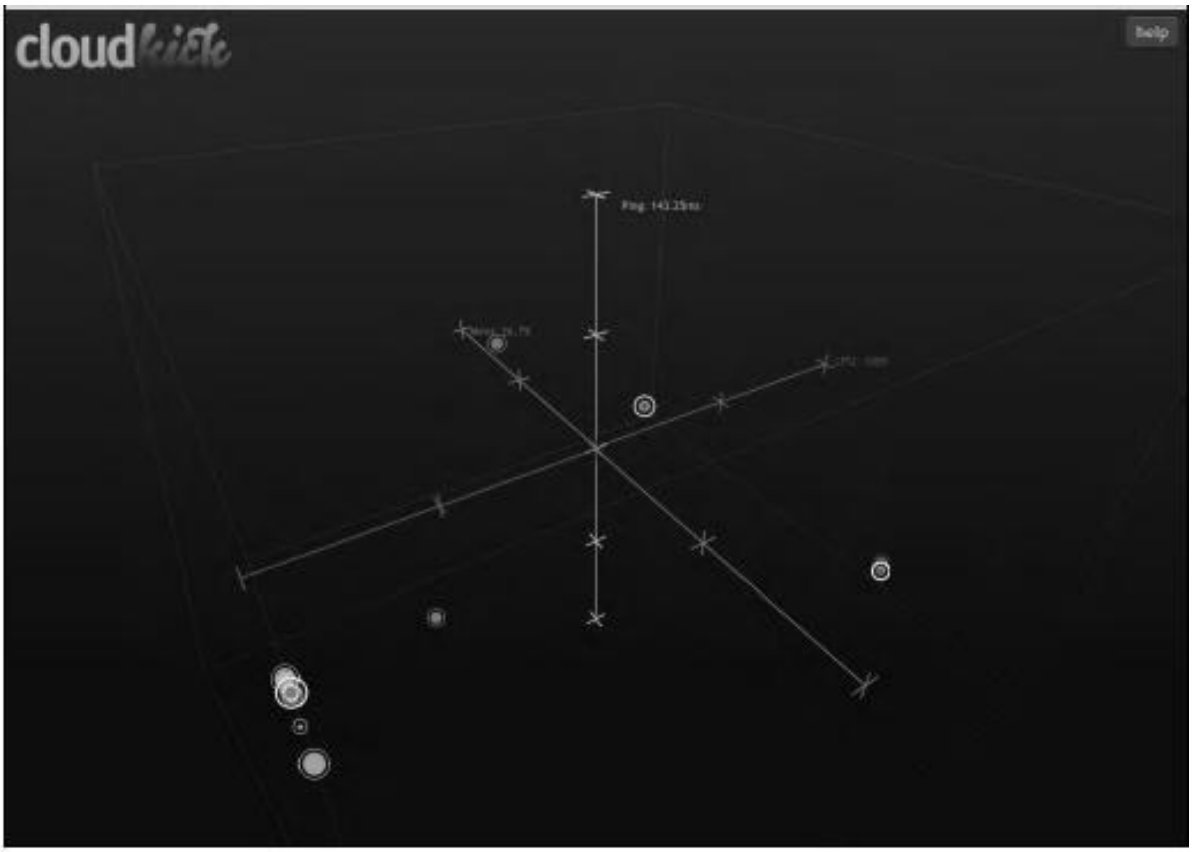


All of the service models support monitoring solutions, most often through interaction with the service API. Tapping into a service API allows management software to perform command actions that a user would normally perform. Some of these APIs are themselves scriptable, while in some cases, scripting is supported in the management software. One key differentiator in monitoring and management software is whether the service needs to install an agent or it performs its service without an agent. The monitoring function normally can be performed through direct interaction with a cloud service or client using processes such as an HTTP GET or a network command like PING. For management functions, an agent is helpful in that it can provide needed hooks to manipulate a cloud resource. Agents also, as a general rule, are useful in helping to solve problems associated with firewall NAT traversal.

ManageIQ (http://www.manageiq.com/) and Service-now.com offer an integrated cloud stack that combines the ManageIQ Enterprise Virtualization Management Suite with Service-Now. com's ITSM SaaS service. The system has offers management, discovery, CMDB synchronization, and automated provisioning services. You can integrate these services into your Web applications using an open API that these companies offer.

FIGURE 11.5

The Cloudkick visualization demo (`https://www.cloudkick.com/viz/demo/`) provides a real-time graphical illustration of the state of monitored servers.



Distributed network applications often benefit from the deployment of a management appliance. Because cloud services tend to distribute applications across multiple sites, physical appliances need to be deployed in different locations—something that only cloud service providers can do. However, there has been a tendency to create virtual appliances, and those can be deployed as server instances wherever an application is deployed. Pareto Networks (http://www.pareto/networks.com/) has a cloud computing service that can monitor and manage distributed network services using a physical or virtual appliance. The system can be used to control and provision network services. Pareto Networks plans to add an API to this service.

4.4 EMERGING CLOUD MANAGEMENT STANDARDS:

As it stands now, different cloud service providers use different technologies for creating and managing cloud resources. As the area matures, cloud providers are going to be under considerable pressure from large cloud users like the federal government to conform to standards and make their systems interoperable with one another. No entity is likely to want to make a major investment in a service that is a silo or from which data is difficult to stage or to extract. To this end, a number of large industry players such as VMware, IBM, Microsoft, Citrix, and HP have gotten together to create standards that can be used to promote cloud interoperability. In the section that follows, you learn about the work of the DMTF in this area.

Another effort just getting underway has been started by CA (the company formerly known as Computer Associates) in association with Carnegie Mellon called the Cloud Commons. This effort is aimed at creating an industry community and working group, and promoting a set of monitoring standards that were part of CA's cloud technology portfolio but are now open sourced.

4.4.1 DMTF cloud management standards:

The Distributed Management Task Force (DMTF; see http://www.dmtf.org/) is an industry organization that develops industry system management standards for platform interoperability. Its membership is a –who's who‖ in computing, and since its founding in 1992, the group has been responsible for several industry standards, most notably the Common Information Model (CIM). The DMTF organizes itself into a set of working groups that are tasked with specifying standards for different areas of technology. A recent standard called the Virtualization Management Initiative (VMAN) was developed to extend CIM to virtual computer system management. VMAN has resulted in the creation of the Open Virtualization Format (OVF), which describes a standard method for creating, packaging, and provisioning virtual appliances. OVF is essentially a container and a file format that is open and both hypervisor- and processor-architecture-agnostic. Since OVF was announced in 2009, vendors such as VirtualBox, AbiCloud, IBM, Red Hat, and VMWare have announced or introduced products that use OVF.

It was, therefore, a natural extension of the work that DMTF does in virtualization to solve management issues in cloud computing. DMTF has created a working group called the Open Cloud Standards Incubator (OCSI) to help develop interoperability standards for managing interactions between and in public, private, and hybrid cloud systems. The group is focused on describing resource management and security protocols, packaging methods, and network management technologies. The Web site of the Cloud Management group (http://dmtf.org/standards/cloud) is shown in Figure 11.6.

DMTF's cloud management efforts are really in their initial stages, but the group has broad industry support. Part of the group's task is to provide industry education, so you can find a number of white papers and technology briefs published on this site. It's an effort that's worth checking back with over time. Although the OCSI's work has not yet been joined by Amazon or Salesforce.com, a set of open standards that extend the use of industry standard protocols—such as the Common Information Model (CIM), the Open Virtualization Format (OVF), and WBEM—to the cloud aregoing to be hard for vendors to resist.

FIGURE 11.6

DMTF (http://dmtf.org/standards/cloud) has a large and important effort underway for developing cloud interoperability management standards.

Cloud Commons and SMI:

CA Technologies (http://www.ca.com), the company once known as Computer Associates, has taken some of its technologies in measuring distributed network performance metrics and repositioned its products as the following:

- CA Cloud Insight, a cloud metrics measurement service
- CA Cloud Compose, a deployment service
- CA Cloud Optimize, a cloud optimization service
- CA Cloud Orchestrate, a workflow control and policy based automation service

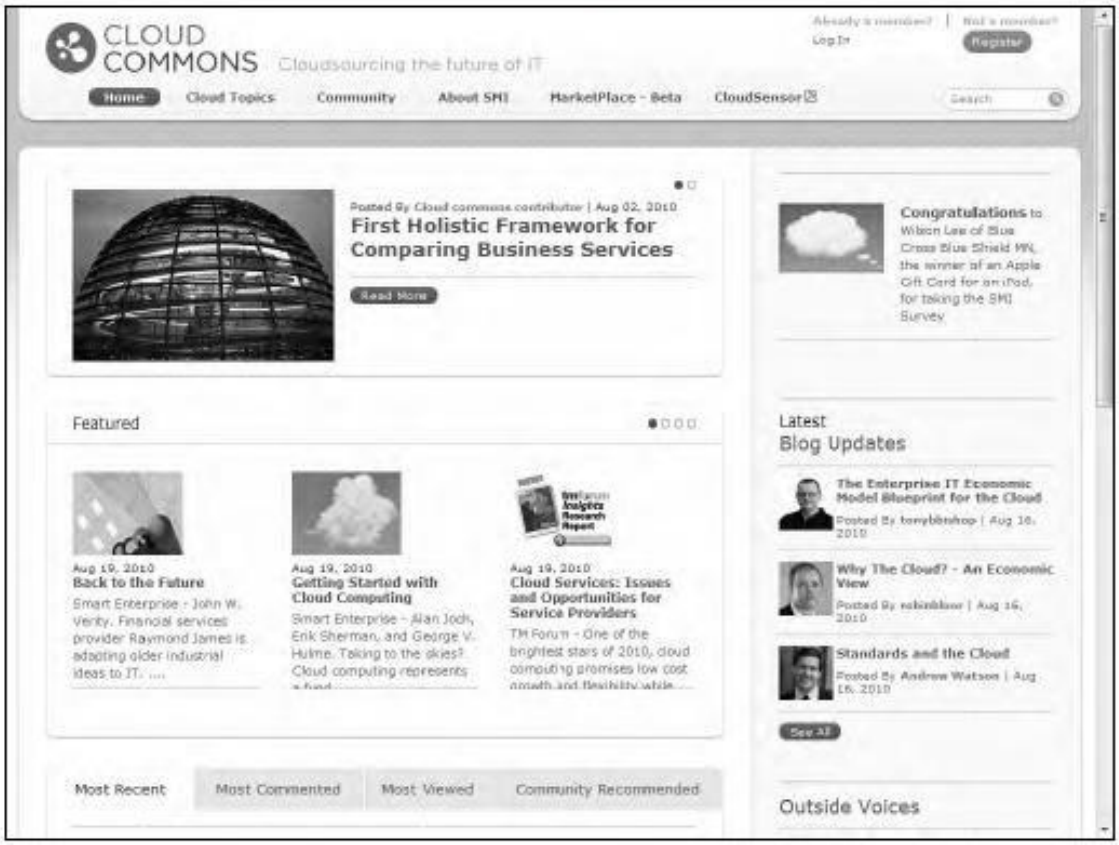
Taken together, these products form the basis for CA's Cloud Connected Management Suite (http://www.ca.com/us/cloud-solutions.aspx). CA has lots of experience in this area through its Unicenter management suite and the products that were spawned from it. The company also has invested in cloud vendors such as 3Tera, Oblicore, and Cassatt to create their cloud services. CA acquired Nimsoft in March 2010. Nimsoft has a monitoring and management package called Nimsoft United Monitoring that creates a monitoring portal with customizable dashboards. The system can gather information from up to 100 types of data points and can work with both Google and Rackspace cloud deployments. Among the data points that can be monitored are resource usage and UPS status.

At the heart of CA Cloud Insight is a method for measuring different cloud metrics that creates what CA calls a Service Measurement Index or SMI. The SMI measures things like SLA compliance, cost, and other values and rolls them up into a score. To help allow SMI to gain traction in the industry, CA has donated the core technology to the Software Engineering Institute at Carnegie Mellon as part of what is called the SMI Consortium. This same group is responsible for the Capability Maturity Model Integration (CMMI) process optimization technology and other efforts. The second CA initiative is the

funding of an industry online community called the Cloud Commons(http://www.cloudcommons.com/) , the home page of which is shown in Figure 11.7.

FIGURE 11.7

The Cloud Commons (http://www.cloudcommons.com/web/guest) is a new online community founded by CA to promote information exchange on cloud services and the SMI standard.



Because the Cloud Commons is brand new, it is hard to tell whether this group will have impact in the cloud community, but it is an interesting effort. The hope is that not only will this site establish CA's performance metrics, but that community users will eventually provide detailed information and ratings on particular services. To demonstrate the potential of cloud-based metrics, the Cloud Commons has built a dashboard called the CloudSensor that monitors the performance of the major cloud-based services in real time.

This tool measures the performance of the following:

- RackSpace file creation and deletion
- E-mail availability (system uptime) based on Google Gmail, Windows Live Hotmail, and Yahoo! Mail
- Amazon Web Services server creation/destruction times at four AWS sites
- Dashboard Response Times for the consoles of AWS.Amazon, Google App Status, RackSpace Cloud, and Saleforce
- Windows Azure storage benchmarks
- Windows Azure SQL benchmarks

It is meant to demonstrate the value of cloud performance measurements. These metrics are based on real-time data derived from real transactions. Each chart shows the last two hours of activity. Figure 11.8 shows the CloudSensor performance dashboard. The Service Measurement Index (SMI) is based on

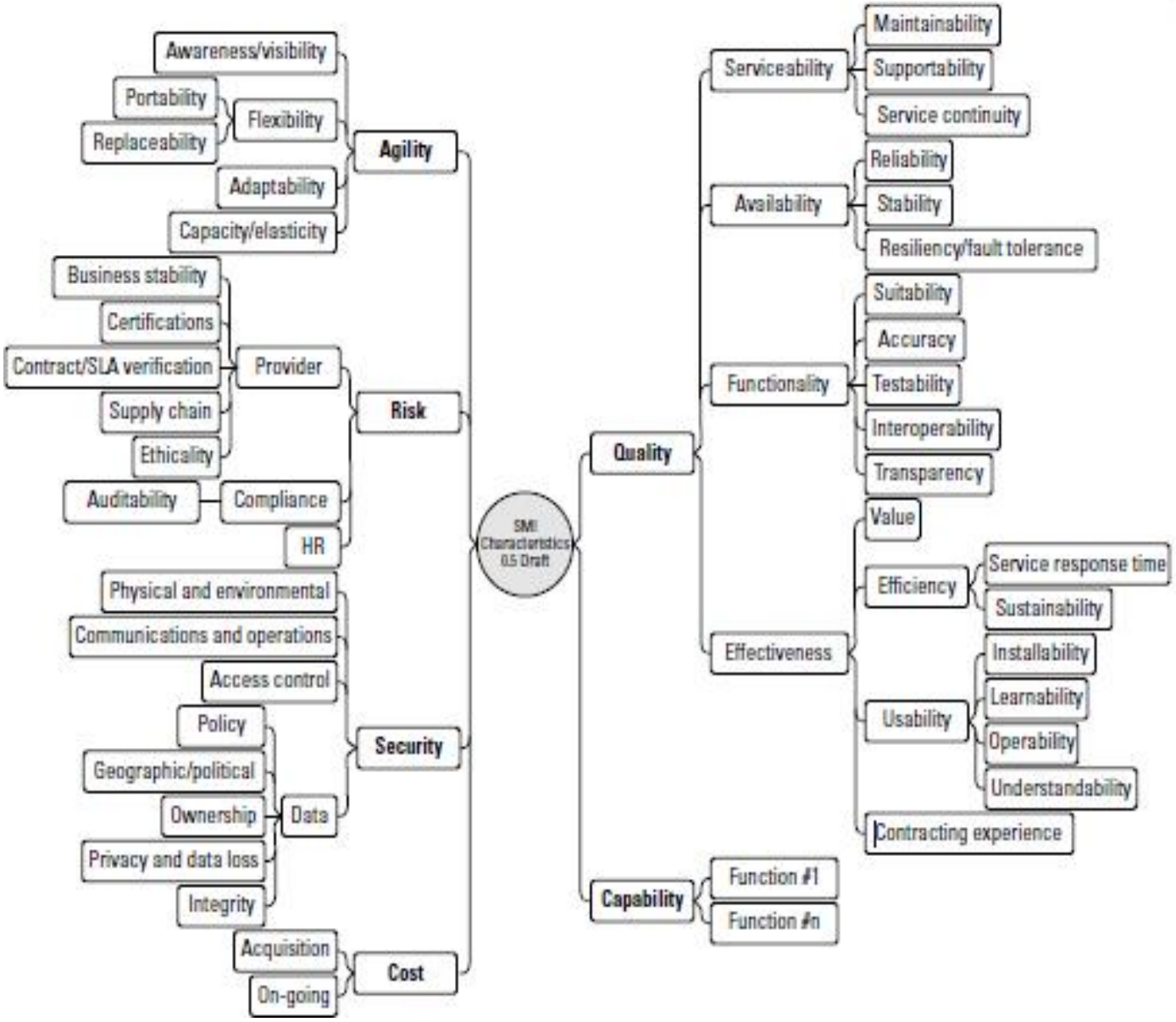
a set of measurement technologies forming the SMI Framework that CA donated to the SMI Consortium. It measures cloud-based services in six areas:

- Agility
- Capability
- Cost
- Quality
- Risk
- Security

These form a set of Key Performance Indicators (KPI) that can be used to compare one service to another. Figure 11.9 shows the different characteristics that make up each of the KPIs of the Service Measurement Index. It's too early to determine whether SMI will gain traction, but the positioning of the technology as an open industry working group makes the project very interesting and worthy of note.

FIGURE 11.9

SMI defined characteristics (Source: "The Details behind the Service Measurement Index" by Keith Allen, 2010)



4.5 SECURING THE CLOUD:

4.5.1 Understanding Cloud Security:

Cloud computing has lots of unique properties that make it very valuable. Unfortunately, many of those properties make security a singular concern. Many of the tools and techniques that you would use to protect your data, comply with regulations, and maintain the integrity of your systems are complicated by the fact that you are sharing your systems with others and many times outsourcing their operations as well. Cloud computing service providers are well aware of these concerns and have developed new technologies to address them.

Different types of cloud computing service models provide different levels of security services. You get the least amount of built in security with an Infrastructure as a Service provider, and the most with a Software as a Service provider. This chapter presents the concept of a security boundary separating the client's and vendor's responsibilities. Adapting your on-premises systems to a cloud model requires that you determine what security mechanisms are required and mapping those to controls that exist in your chosen cloud service provider. When you identify missing security elements in the cloud, you can use that mapping to work to close the gap. Storing data in the cloud is of particular concern. Data should be transferred and stored in an encrypted format. You can use proxy and brokerage services to separate clients from direct access to shared cloud storage.

Logging, auditing, and regulatory compliance are all features that require planning in cloud computing systems. They are among the services that need to be negotiated in Service Level Agreements. Also in this chapter, you learn about identity and related protocols from a security standpoint. The concept of presence as it relates to identity is also introduced.

4.5.2 Securing the Cloud:

The Internet was designed primarily to be resilient; it was not designed to be secure. Any distributed application has a much greater attack surface than an application that is closely held on a Local Area Network. Cloud computing has all the vulnerabilities associated with Internet applications, and additional vulnerabilities arise from pooled, virtualized, and outsourced resources. In the report ―Assessing the Security Risks of Cloud Computing,‖ Jay Heiser and Mark Nicolett of the Gartner Group (http://www.gartner.com/DisplayDocument?id=685308) highlighted the following areas of cloud computing that they felt were uniquely troublesome:

- Auditing
- Data integrity
- e-Discovery for legal compliance
- Privacy
- Recovery
- Regulatory compliance

Your risks in any cloud deployment are dependent upon the particular cloud service model chosen and the type of cloud on which you deploy your applications. In order to evaluate your risks, you need to perform the following analysis:

1. Determine which resources (data, services, or applications) you are planning to move to the cloud.

2. Determine the sensitivity of the resource to risk. Risks that need to be evaluated are loss of privacy, unauthorized access by others, loss of data, and interruptions in availability.

3. Determine the risk associated with the particular cloud type for a resource. Cloud types include public, private (both external and internal), hybrid, and shared community types. With each type, you need to consider where data and functionality will be maintained.

4. Take into account the particular cloud service model that you will be using. Different models such as IaaS, SaaS, and PaaS require their customers to be responsible for security at different levels of the service stack.

5. If you have selected a particular cloud service provider, you need to evaluate its system to understand how data is transferred, where it is stored, and how to move data both in and out of the cloud.

You may want to consider building a flowchart that shows the overall mechanism of the system you are intending to use or are currently using. One technique for maintaining security is to have ―golden‖ system image references that you can return to when needed. The ability to take a system image off-line and analyze the image for vulnerabilities or compromise is invaluable. The compromised image is a primary forensics tool. Many cloud providers offer a snapshot feature that can create a copy of the client's entire environment; this includes not only machine images, but applications and data, network interfaces, firewalls, and switch access. If you feel that a system has been compromised, you can replace that image with a known good version and contain the problem.

Many vendors maintain a security page where they list their various resources, certifications, and credentials. One of the more developed offerings is the AWS Security Center, shown in Figure 12.1, where you can download some backgrounders, white papers, and case studies related to the Amazon Web Service's security controls and mechanisms.

FIGURE 12.1

The AWS Security Center (`http://aws.amazon.com/security/`) is a good place to start learning about how Amazon Web Services protects users of its IaaS service.

4.5.3 The security boundary:

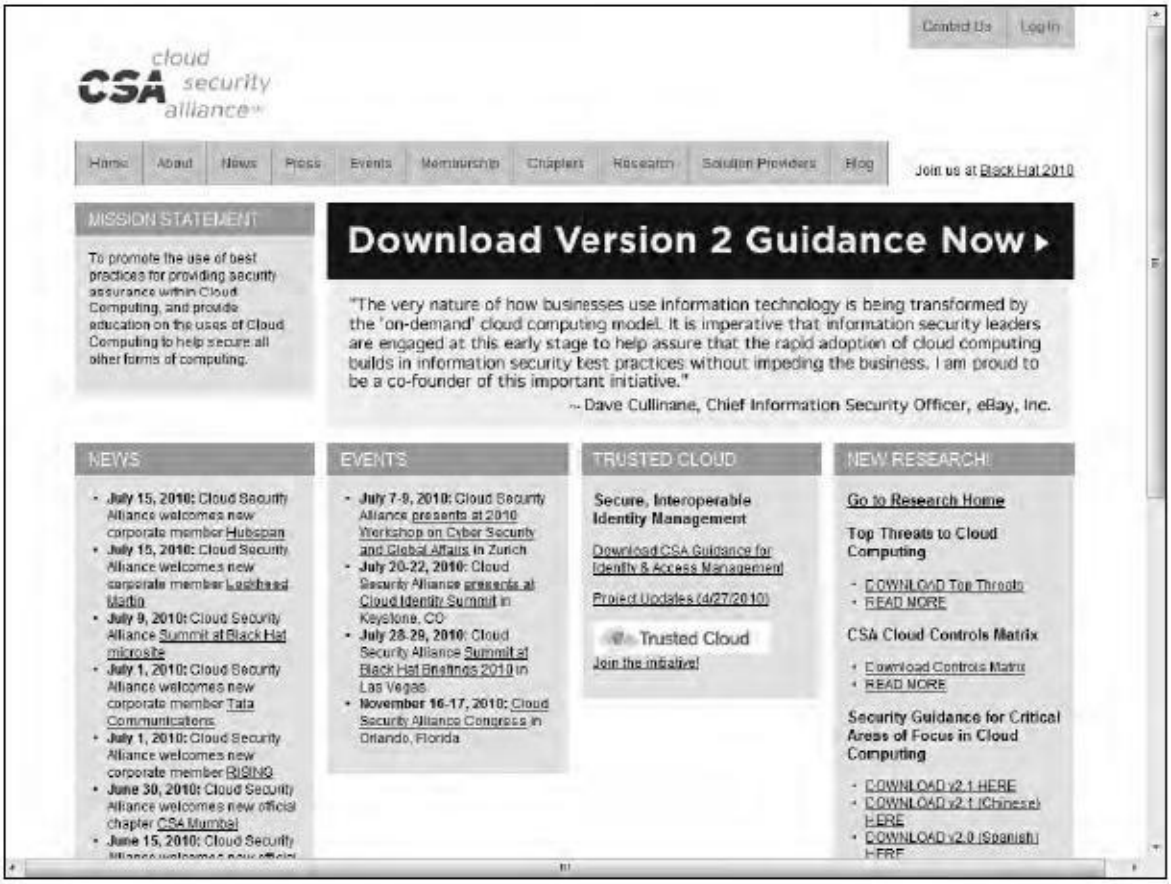
In order to concisely discuss security in cloud computing, you need to define the particular model of cloud computing that applies. This nomenclature provides a framework for understanding what security is already built into the system, who has responsibility for a particular security mechanism, and where the boundary between the responsibility of the service provider is separate from the responsibility of the customer.

All of Chapter I was concerned with defining what cloud computing is and defining the lexicon of cloud computing. There are many definitions and acronyms in the area of cloud computing that will probably not survive long. The most commonly used model based on U.S. National Institute of Standards and Technology (NIST; http://www.csrc.nist.gov/groups/SNS/cloudcomputing/index.html) separates deployment models from service models and assigns those models a set of service attributes. Deployment models are cloud types: community, hybrid, private, and public clouds. Service models follow the SPI Model for three forms of service delivery: Software, Platform, and Infrastructure as a Service. In the NIST model, as you may recall, it was not required that a cloud use virtualization to pool resources, nor did that model require that a cloud support multi-tenancy. It is just these factors that make security such a complicated proposition in cloud computing.

Also presented the Cloud Security Alliance (CSA; http://www.cloudsecurityalliance.org/) cloud computing stack model, which shows how different functional units in a network stack relate to one another. As you may recall from Chapter 1, this model can be used to separate the different service models from one another. CSA is an industry working group that studies security issues in cloud computing and offers recommendations to its members. The work of the group is open and available, and you can download its guidance from its home page, shown in Figure 12.2.

FIGURE 12.2

The Cloud Security Alliance (CSA) home page at `http://www.cloudsecurityalliance.org/` offers a number of resources to anyone concerned with securing his cloud deployment.



The CSA partitions its guidance into a set of operational domains:

- Governance and enterprise risk management
- Legal and electronic discovery
- Compliance and audit
- Information lifecycle management
- Portability and interoperability
- Traditional security, business continuity, and disaster recovery
- Datacenter operations
- Incidence response, notification, and remediation
- Application security
- Encryption and key management
- Identity and access management
- Virtualization

You can download the group's current work in these areas from the different sections of its Web site. One key difference between the NIST model and the CSA is that the CSA considers multi-tenancy to be an essential element in cloud computing. Multi-tenancy adds a number of additional security concerns to cloud computing that need to be accounted for. In multi-tenancy, different customers must be isolated, their data segmented, and their service accounted for. To provide these features, the cloud service provider must provide a policy-based environment that is capable of supporting different levels and quality of service, usually using different pricing models. Multi-tenancy expresses itself in different ways in the different cloud deployment models and imposes security concerns in different places.

4.5.4 Security service boundary:

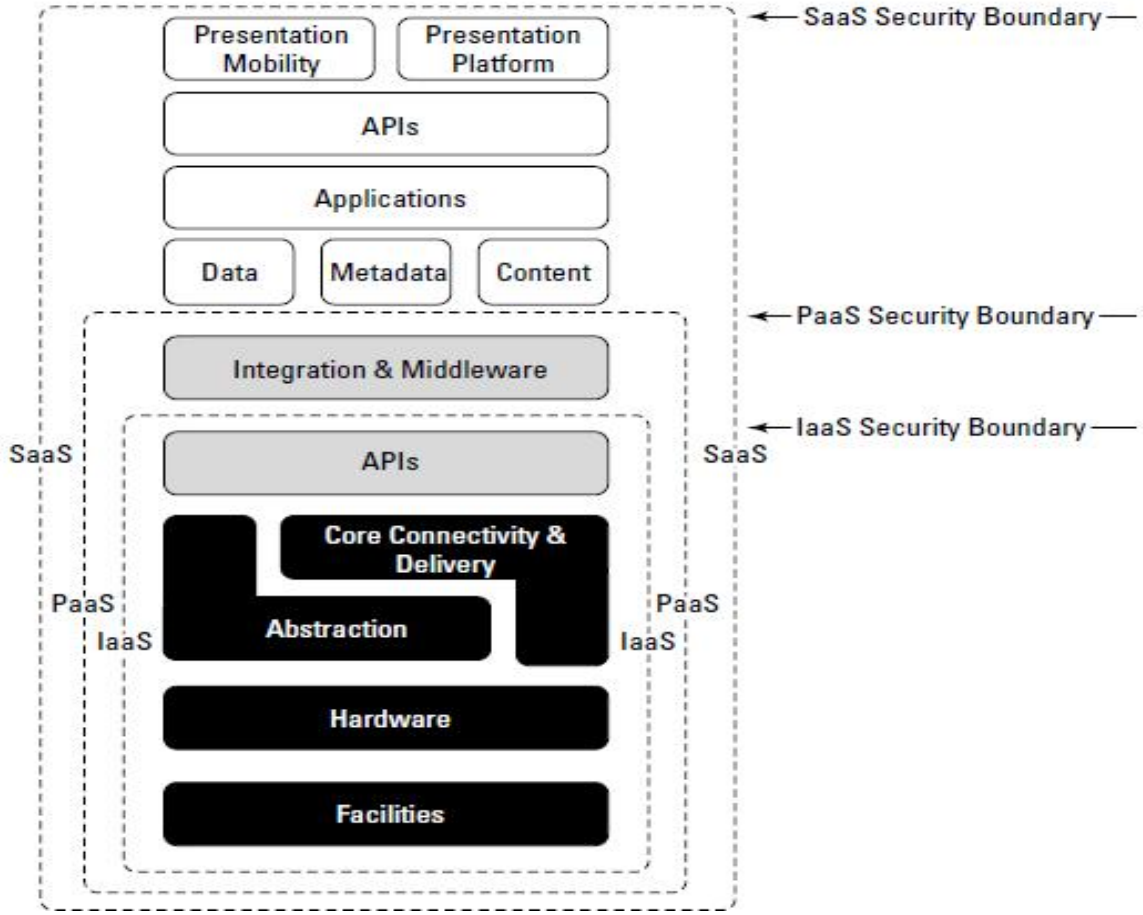
The CSA functional cloud computing hardware/software stack is the Cloud Reference Model. This model, which was discussed in Chapter 1, is reproduced in Figure 12.3. IaaS is the lowest level service, with PaaS and SaaS the next two services above. As you move upward in the stack, each service model inherits the capabilities of the model beneath it, as well as all the inherent security concerns and risk factors. IaaS supplies the infrastructure; PaaS adds application development frameworks, transactions, and control structures; and SaaS is an operating environment with applications, management, and the user interface. As you ascend the stack, IaaS has the least levels of integrated functionality and the lowest levels of integrated security, and SaaS has the most.

The most important lesson from this discussion of architecture is that each different type of cloud service delivery model creates a security boundary at which the cloud service provider's responsibilities end and the customer's responsibilities begin. Any security mechanism below the security boundary must be built into the system, and any security mechanism above must be maintained by the customer. As you move up the stack, it becomes more important to make sure that the type and level of security is part of your Service Level Agreement.

In the SaaS model, the vendor provides security as part of the Service Level Agreement, with the compliance, governance, and liability levels stipulated under the contract for the entire stack. For the PaaS model, the security boundary may be defined for the vendor to include the software framework and middleware layer. In the PaaS model, the customer would be responsible for the security of the application and UI at the top of the stack. The model with the least built-in security is IaaS, where everything that involves software of any kind is the customer's problem. Numerous definitions of services tend to muddy this picture by adding or removing elements of the various functions from any particular offering, thus blurring which party has responsibility for which features, but the overall analysis is still useful.

FIGURE 12.3

The CSA Cloud Reference Model with security boundaries shown



In thinking about the Cloud Security Reference Model in relationship to security needs, a fundamental distinction may be made between the nature of how services are provided versus where those services are located. A private cloud may be internal or external to an organization, and although a public cloud is most often external, there is no requirement that this mapping be made so. Cloud computing has a tendency to blur the location of the defined security perimeter in such a way that the previous notions of network firewalls and edge defenses often no longer apply. This makes the location of trust boundaries in cloud computing rather ill defined, dynamic, and subject to change depending upon a number of factors. Establishing trust boundaries and creating a new perimeter defense that is consistent with your cloud computing network is an important consideration.

The key to understanding where to place security mechanisms is to understand where physically in the cloud resources are deployed and consumed, what those resources are, who manages the resources, and what mechanisms are used to control them. Those factors help you gauge where systems are located and what areas of compliance you need to build into your system. Table 12.1 lists some of the different service models and lists the parties responsible for security in the different instances.

TABLE 12.1

Security Responsibilities by Service Model

Model Type	Infrastructure Security Management	Infrastructure Owner	Infrastructure Location	Trust Condition
Hybrid	Both vendor and customer	Both vendor and customer	Both on- and off-premises	Both trusted and untrusted
Private/Community	Customer	Customer	On- or off-premises	Trusted
Private/Community	Customer	Vendor	Off- or on-premises	Trusted
Private/Community	Vendor	Customer	On- or off-premises	Trusted
Private/Community	Vendor	Vendor	Off- or on-premises	Trusted
Public	Vendor	Vendor	Off-premises	Untrusted

Security mapping:

The cloud service model you choose determines where in the proposed deployment the variety of security features, compliance auditing, and other requirements must be placed. To determine the particular security mechanisms you need, you must perform a mapping of the particular cloud service model to the particular application you are deploying. These mechanisms must be supported by the various controls that are provided by your service provider, your organization, or a third party. It's unlikely that you will be able to duplicate security routines that are possible on-premises, but this analysis allows you to determine what coverage you need.

A security control model includes the security that you normally use for your applications, data, management, network, and physical hardware. You may also need to account for any compliance standards that are required for your industry. A compliance standard can be any government regulatory framework such as Payment Card Industry Data Security Standards (PCI-DSS), Health Insurance Portability and Accountability Act (HIPPA), Gramm–Leach–Bliley Act (GLBA), or the Sarbanes–Oxley Act (SOX) that requires you operate in a certain way and keep records.

Essentially, you are looking to identify the missing features that would be required for an on-premises deployment and seek to find their replacements in the cloud computing model. As you assign accountability for different aspects of security and contract away the operational responsibility to others, you want to make sure they remain accountable for the security you need.

4.6 SECURING DATA:

Securing data sent to, received from, and stored in the cloud is the single largest security concern that most organizations should have with cloud computing. As with any WAN traffic, you must assume that any data can be intercepted and modified. That's why, as a matter of course, traffic to a cloud service provider and stored off-premises is encrypted. This is as true for general data as it is for any passwords or account IDs.

These are the key mechanisms for protecting data mechanisms:

- Access control
- Auditing
- Authentication
- Authorization

Whatever service model you choose should have mechanisms operating in all four areas that meet your security requirements, whether they are operating through the cloud service provider or your own local infrastructure.

Brokered cloud storage access:

The problem with the data you store in the cloud is that it can be located anywhere in the cloud service provider's system: in another datacenter, another state or province, and in many cases even in another country. With other types of system architectures, such as client/server, you could count on a firewall to serve as your network's security perimeter; cloud computing has no physical system that serves this purpose. Therefore, to protect your cloud storage assets, you want to find a way to isolate data from direct client access.

One approach to isolating storage in the cloud from direct client access is to create layered access to the data. In one scheme, two services are created: a broker with full access to storage but no access to the client, and a proxy with no access to storage but access to both the client and broker. The location of the proxy and the broker is not important (they can be local or in the cloud); what is important is that these two services are in the direct data path between the client and data stored in the cloud. Under this system, when a client makes a request for data, here's what happens:

1. The request goes to the external service interface (or endpoint) of the proxy, which has only a partial trust.

2. The proxy, using its internal interface, forwards the request to the broker.

3. The broker requests the data from the cloud storage system.

4. The storage system returns the results to the broker.
5. The broker returns the results to the proxy.

6. The proxy completes the response by sending the data requested to the client.

Figure 12.4 shows this storage ‒proxy‖ system graphically.

NOTE:

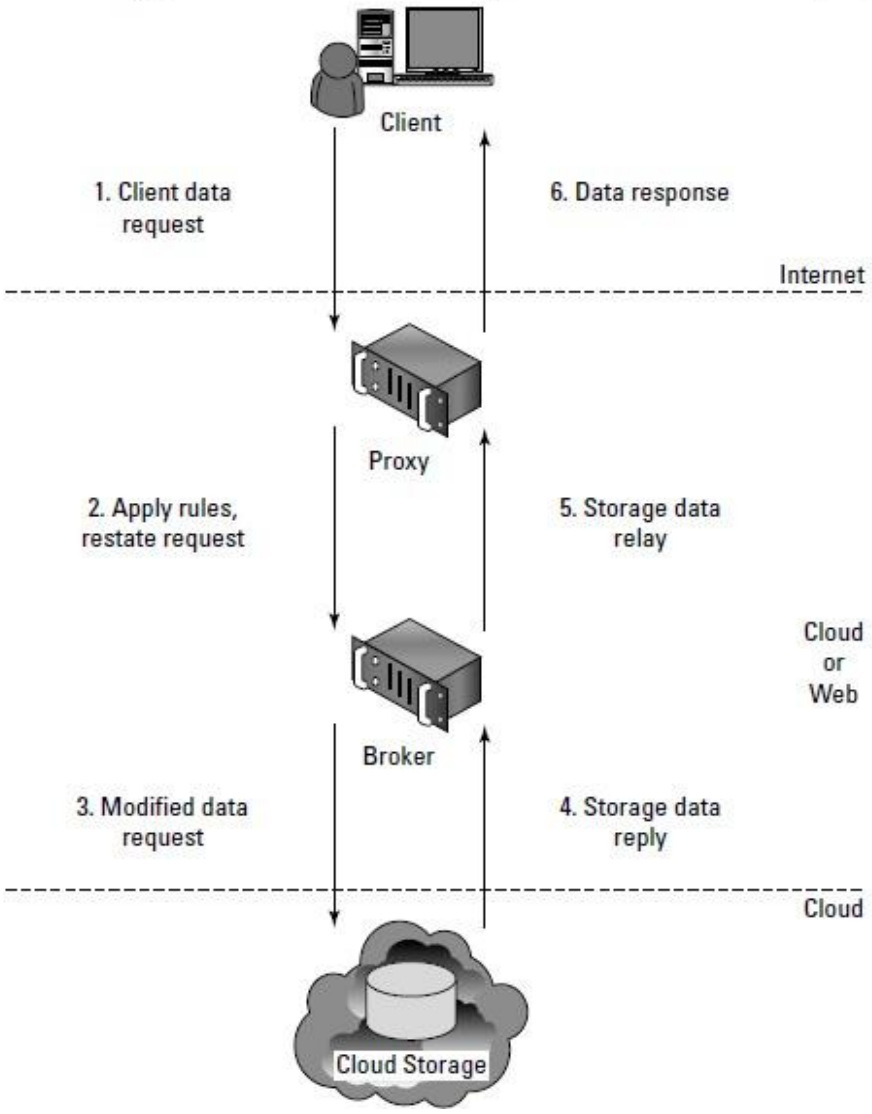
This discussion is based on a white paper called "Security Best Practices For Developing Windows Azure Applications," by Andrew Marshall, Michael Howard, Grant Bugher, and Brian Harden that you can find at http://download.microsoft.com/download/7/3/E/73E4EE93-559F-4D0F-A6FC-7FEC5F1542D1/SecurityBestPracticesWindowsAzureApps.docx**. In their presentation, the proxy service is called the Gatekeeper and assigned a Windows Server Web Role, and the broker is called the KeyMaster and assigned a Worker Role.**

This design relies on the proxy service to impose some rules that allow it to safely request data that is appropriate to that particular client based on the client's identity and relay that request to the broker. The broker does not need full access to the cloud storage, but it may be configured to grant READ and QUERY operations, while not allowing APPEND or DELETE. The proxy has a limited trust role, while the broker can run with higher privileges or even as native code.

The use of multiple encryption keys can further separate the proxy service from the storage account. If you use two separate keys to create two different data zones—one for the untrusted communication between the proxy and broker services, and another a trusted zone between the broker and the cloud storage—you create a situation where there is further separation between the different service roles.

FIGURE 12.4

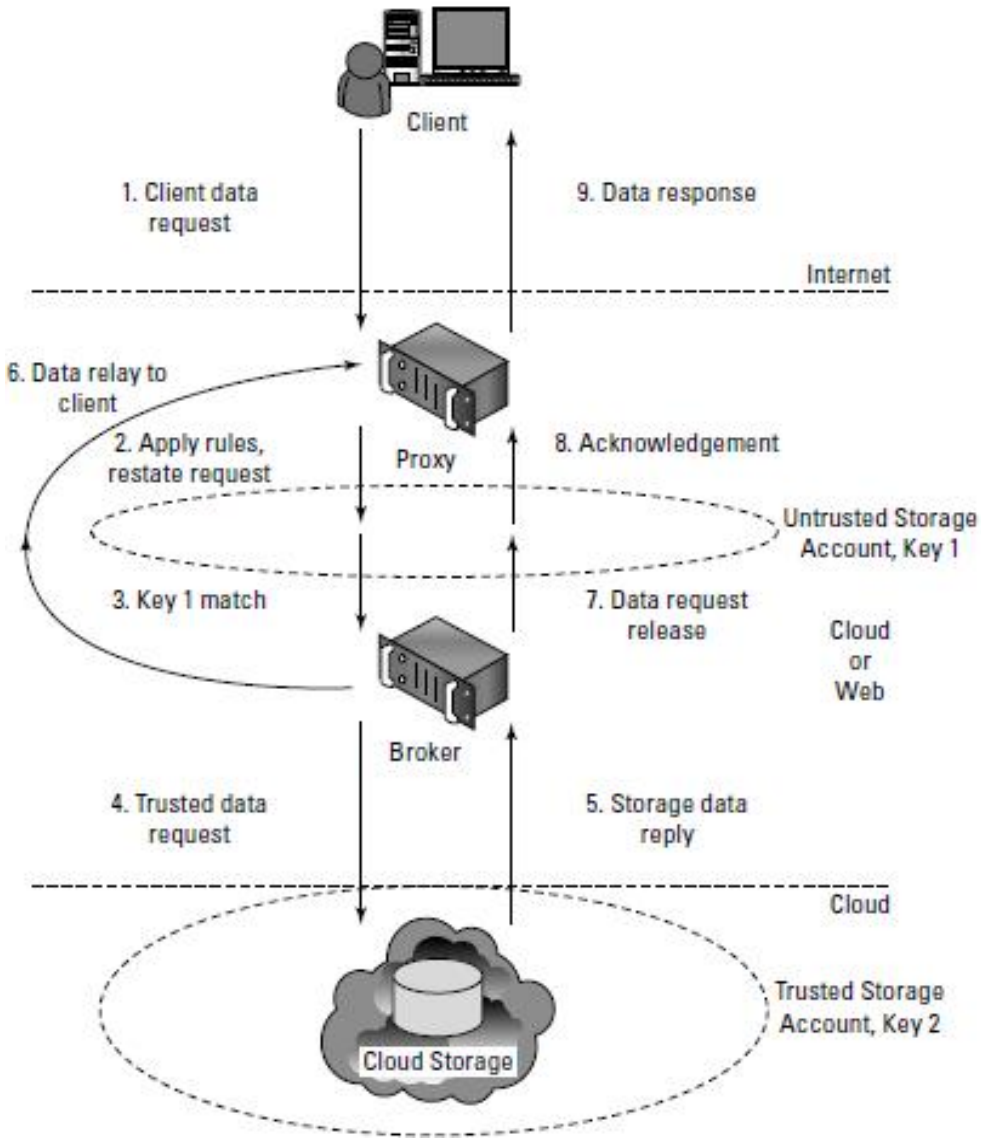
In this design, direct access to cloud storage is eliminated in favor of a proxy/broker service.



Even if the proxy service is compromised, that service does not have access to the trusted key necessary to access the cloud storage account. In the multi-key solution, shown in Figure 12.5, you have not only eliminated all internal service endpoints, but you also have eliminated the need to have the proxy service run at a reduced trust level.

FIGURE 12.5

The creation of storage zones with associated encryption keys can further protect cloud storage from unauthorized access.



Storage location and tenancy:

Some cloud service providers negotiate as part of their Service Level Agreements to contractually store and process data in locations that are predetermined by their contract. Not all do. If you can get the commitment for specific data site storage, then you also should make sure the cloud vendor is under contract to conform to local privacy laws. Because data stored in the cloud is usually stored from multiple tenants, each vendor has its own unique method for segregating one customer's data from another. It's important to have some understanding of how your specific service provider maintains data segregation.

Another question to ask a cloud storage provider is who is provided privileged access to storage. The more you know about how the vendor hires its IT staff and the security mechanism put into place to protect storage, the better. Most cloud service providers store data in an encrypted form. While encryption is important and effective, it does present its own set of problems. When there is a problem

with encrypted data, the result is that the data may not be recoverable. It is worth considering what type of encryption the cloud provider uses and to check that the system has been planned and tested by security experts.

Regardless of where your data is located, you should know what impact a disaster or interruption will have on your service and your data. Any cloud provider that doesn't offer the ability to replicate data and application infrastructure across multiple sites cannot recover your information in a timely manner. You should know how disaster recovery affects your data and how long it takes to do a complete restoration.

Encryption:

Strong encryption technology is a core technology for protecting data in transit to and from the cloud as well as data stored in the cloud. It is or will be required by law. The goal of encrypted cloud storage is to create a virtual private storage system that maintains confidentiality and data integrity while maintaining the benefits of cloud storage: ubiquitous, reliable, shared data storage. Encryption should separate stored data (data at rest) from data in transit. Depending upon the particular cloud provider, you can create multiple accounts with different keys as you saw in the example with Windows Azure Platform in the previous section. Microsoft allows up to five security accounts per client, and you can use these different accounts to create different zones. On Amazon Web Service, you can create multiple keys and rotate those keys during different sessions.

Although encryption protects your data from unauthorized access, it does nothing to prevent data loss. Indeed, a common means for losing encrypted data is to lose the keys that provide access to the data. Therefore, you need to approach key management seriously. Keys should have a defined lifecycle. Among the schemes used to protect keys are the creation of secure key stores that have restricted role-based access, automated key stores backup, and recovery techniques. It's a good idea to separate key management from the cloud provider that hosts your data.

One standard for interoperable cloud-based key management is the OASIS Key Management Interoperability Protocol (KMIP; http://www.oasis-open.org/committees/kmip/). IEEE 1619.3 (https://siswg.net/index.php?option=com_docman) also covers both storage encryption and key management for shared storage.

Auditing and compliance:

Logging is the recording of events into a repository; auditing is the ability to monitor the events to understand performance. Logging and auditing is an important function because it is not only necessary for evaluation performance, but it is also used to investigate security and when illegal activity has been perpetrated. Logs should record system, application, and security events, at the very minimum. Logging and auditing are unfortunately one of the weaker aspects of early cloud computing service offerings.

Cloud service providers often have proprietary log formats that you need to be aware of. Whatever monitoring and analysis tools you use need to be aware of these logs and able to work with them. Often, providers offer monitoring tools of their own, many in the form of a dashboard with the potential to customize the information you see through either the interface or programmatically using the vendor's API. You want to make full use of those built-in services.

Because cloud services are both multitenant and multisite operations, the logging activity and data for different clients may not only be co-located, they may also be moving across a landscape of different hosts and sites. You can't simply expect that an investigation will be provided with the necessary information at the time of discovery unless it is part of your Service Level Agreement. Even

an SLA with the appropriate obligations contained in it may not be enough to guarantee you will get the information you need when the time comes. It is wise to determine whether the cloud service provider has been able to successfully support investigations in the past.

As it stands now, nearly all regulations were written without keeping cloud computing in mind. A regulator or auditor isn't likely to be familiar with the nature of running applications and storing data in the cloud. Even so, laws are written to ensure compliance, and the client is held responsible for compliance under the laws of the governing bodies that apply to the location where the processing or storage takes place.

Therefore, you must understand the following:

- Which regulations apply to your use of a particular cloud computing service
- Which regulations apply to the cloud service provider and where the demarcation line falls for responsibilities
- How your cloud service provider will support your need for information associated with regulation
- How to work with the regulator to provide the information necessary regardless of who had the responsibility to collect the data

Traditional service providers are much more likely to be the subject of security certifications and external audits of their facilities and procedures than cloud service providers. That makes the willingness for a cloud service provider to subject its service to regulatory compliance scrutiny an important factor in your selection of that provider over another. In the case of a cloud service provider who shows reluctance to or limits the scrutiny of its operations, it is probably wise to use the service in ways that limit your exposure to risk.

For example, although encrypting stored data is always a good policy, you also might want to consider not storing any sensitive information on that provider's system. As it stands now, clients must guarantee their own regulatory compliance, even when their data is in the care of the service provider. You must ensure that your data is secure and that its integrity has not been compromised. When multiple regulatory entities are involved, as there surely are between site locations and different countries, then that burden to satisfy the laws of those governments is also your responsibility.

For any company with clients in multiple countries, the burden of regulatory compliance is onerous. While organizations such as the EEC (European Economic Community) or Common Market provide some relief for European regulation, countries such as the United States, Japan, China, and others each have their own sets of requirements. This makes regulatory compliance one of the most actively developing and important areas of cloud computing technology.

This situation is likely to change. On March 1, 2010, Massachusetts passed a law that requires companies that provide sensitive personal information on Massachusetts residents to encrypt data transmitted and stored on their systems. Businesses are required to limit the amount of personal data collected, monitor data usage, keep a data inventory, and be able to present a security plan on how they will keep the data safe. The steps require that companies verify that any third-party services they use conform to these requirements and that there be language in all SLAs that enforce these protections. The law takes full effect in March 2012.

Going forward, you want to ensure the following:

- You have contracts reviewed by your legal staff.
- You have a right-to-audit clause in your SLA.
- You review any third parties who are service providers and assess their impact on security and regulatory compliance.

- You understand the scope of the regulations that apply to your cloud computing applications and services.
- You consider what steps you must take to comply with the demands of regulations that apply.
- You consider adjusting your procedures to comply with regulations.
- You collect and maintain the evidence of your compliance with regulations.
- You determine whether your cloud service provider can provide an audit statement that is SAS 70 Type II-compliant.

The ISO/IEC 27001/27002 standard for information security management systems has a roadmap for mission-critical services that you may want to discuss with your cloud service provider. Amazon Web Services supports SAS70 Type II Audits.

Becoming a cloud service provider requires a large investment, but as we all know, even large companies can fail. When a cloud service provider fails, it may close or more likely be acquired by another company. You likely wouldn't use a service provider that you suspected of being in difficulty, but problems develop over years and cloud computing has a certain degree of vendor lockin to it. That is, when you have created a cloud-based service, it can be difficult or often impossible to move it to another service provider. You should be aware of what happens to your data if the cloud service provider fails. At the very least, you would want to make sure your data could be obtained in a format that could be accessed by on-premise applications.

The various attributes of cloud computing make it difficult to respond to incidents, but that doesn't mean you should consider drawing up security incidence response policies. Although cloud computing creates shared responsibilities, it is often up to the client to initiate the inquiry that gets the ball rolling. You should be prepared to provide clear information to your cloud service provider about what you consider to be an incident or a breach in security and what are simply suspicious events.

4.7 ESTABLISHING IDENTITY AND PRESENCE:

Introduced the concept of identities, some of the protocols that support them, and some of the services that can work with them. Identities also are tied to the concept of accounts and can be used for contacts or ―ID cards.‖ Identities also are important from a security standpoint because they can be used to authenticate client requests for services in a distributed network system such as the Internet or, in this case, for cloud computing services.

Identity management is a primary mechanism for controlling access to data in the cloud, preventing unauthorized uses, maintaining user roles, and complying with regulations. The sections that follow describe some of the different security aspects of identity and the related concept of ―presence.‖ For this conversation, you can consider presence to be the mapping of an authenticated identity to a known location. Presence is important in cloud computing because it adds context that can modify services and service delivery.

Cloud computing requires the following:

- That you establish an identity
- That the identity be authenticated
- That the authentication be portable
- That authentication provide access to cloud resources

When applied to a number of users in a cloud computing system, these requirements describe systems that must provision identities, provide mechanisms that manage credentials and authentication, allow identities to be federated, and support a variety of user profiles and access policies. Automating these processes can be a major management task, just as they are for on-premises operations.

Identity protocol standards:

The protocols that provide identity services have been and are under active development, and several form the basis for efforts to create interoperability among services. OpenID 2.0 (http://openid.net/) is the standard associated with creating an identity and having a third-party service authenticate the use of that digital identity. It is the key to creating

Single Sign-On (SSO) systems. Some cloud service providers have adopted OpenID as a service, and its use is growing. In Chapter 4, you learned how OpenID is associated with contact cards such as vCards and InfoCards. In that chapter, I briefly discussed how OpenID provides access to important Web sites and how some Web sites allow you to use your logins based on OpenID from another site to gain access to their site. OpenID doesn't specify the means for authentication of an identity, and it is up to the particular
system how the authentication process is executed. Authentication can be by a Challenge and Response Protocol (CHAP), through a physical smart card, or using a flying finger or evil eye through a biometric measurement.

In OpenIDL, the authentication procedure has the following steps:

1. The end-user uses a program like a browser that is called a user agent to enter an OpenID identifier, which is in the form of a URL or XRI. An OpenID might take the form of *name*.openid.provider.org.

2. The OpenID is presented to a service that provides access to the resource that is desired.

3. An entity called a relaying party queries the OpenID identity provider to authenticate the veracity of the OpenID credentials.

4. The authentication is sent back to the relaying party from the identity provider and access
is either provided or denied.

According to a report by one of OpenID's directors called ―OpenID 2009 Year in Review‖ by Brian Kissel (http://openid.net/2009/12/16/openid-2009-year-in-review/) , there were over 1 billion OpenID accounts accepted by 9 million sites on the Internet.

The second protocol used to present identity-based claims in cloud computing is a set of authorization markup languages that create files in the form of being XACML and SAML. These protocols were described in Chapter 4 in detail, so I only mention them in passing here. SAML (Security Assertion Markup Language; http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security) is gaining growing acceptance among cloud service providers. It is a standard of OASIS and an XML standard for passing authentication and authorization between an identity provider and the service provider. SAML is a complimentary mechanism to OpenID and is used to create SSO systems.

Taken as a unit, OpenID and SAML are being positioned to be the standard authentication mechanism for clients accessing cloud services. It is particularly important for services such as mashups that draw information from two or more data services.

An open standard called OAuth (http://oauth.net/) provides a token service that can be used to present validated access to resources. OAuth is similar to OpenID, but provides a different mechanism for shared access. The use of OAuth tokens allows clients to present credentials that contain no account information (userID or password) to a cloud service. The token comes with a defined period after which it can no longer be used. Several important cloud service providers have begun to make OAuth APIs

available based on the OAuth 2.0 standard, most notably Facebook's Graph API and the Google Data API. The DataPortability Project (http://dataportability.org/) is an industry working group that promotes data interoperability between applications, and the group's work touches on a number of the emerging standards mentioned in this section. The group's Web site is shown in Figure 12.6.

FIGURE 12.6

The home page of the DataPortability Project, an industry working group that promotes open identity standards

A number of vendors have created server products, such as Identity and Access Managers (IAMs), to support these various standards.

Windows Azure identity standards:

The Windows Azure Platform uses a claims-based identity based on open authentication and access protocols and is a good example of a service implementing the standards described in the previous section. These standards may be used without modification on a system that is running in the cloud or on-premises, in keeping with Microsoft's S+S (software plus services) approach to cloud computing.

Windows Azure security draws on the following three services:

- Active Directory Federation Services 2.0
- Windows Azure AppFabric Access Control Service
- Windows Identity Foundation (WIF)

The Windows Identity Foundation offers .NET developers Visual Studio integration of WS-Federation and WS-Trust open standards. ASP.NET Web applications created with WIF integrate the Windows

Communication Foundation SOAP service (WCF-SOAP) into a unified object model. This allows WIF to have full access to the features of WS-Security and to work with tokens in the SAML format. WIF relies on third-party authentication and accepts authentication requests from these services in the form of a set of claims. Claims are independent of where a user account or application is located, thus allowing claims to be used in single sign-on systems (SSO). Claims support both simple resource access and the Role Based Access Control (RBAC) policies that can be enforced by Windows group policies.

Active Directory Federation Services 2.0 (AD FS) is a Security Token Service (STS) that allows users to authenticate their access to applications both locally and in the cloud with a claims-based identity. Anyone who has an account in the local Windows directory can access an application; AD FS creates and retains trust relationships with federated systems. AD FS uses WS-Federation, WS-Trust, and SAML, which allows users to access a system based on IBM, Novell, SAP, and many other vendors.

The final piece of the Windows Azure Platform claims-based identity system is built directly into the AppFabric Access Control (AC) service. You may recall from Chapter 10 that AppFabric is a service bus for Azure components that supports REST Web services. Included in AppFabric are authentication and claims-based authorization access. These can be simple logons or more complex schemes supported by AD FS. AC allows authorization to be located anywhere and allows developers to separate identity from their application.

The claims-based identity in AC is based on the OAuth Web Resource Authorization Protocol (OAuth WRAP), which works with various REST APIs. The Oauth 2.0 protocol seems to be gaining acceptance in the cloud computing industry, because SAML tokens can be accepted by many vendors.

Presence:

Presence is a fundamental concept in computer science. It is used on networks to indicate the status of available parties and their location. Commands like the WHO command in Linux that list users logged into the network go all the way back to the first network operating systems. Presence provides not only identity, but status and, as part of status, location. The status is referred to as the presence state, the identity is the presentity, and the service that manages presence is called the presence service. Many presence services rely on agents called watchers, which are small programs that relay a client's ability to connect. Among the cloud computing services that rely on presence information are telephony systems such as VoIP, instant messaging services (IM), and geo-location-based systems such as GPS. Presence is playing an important role in cell phones, particularly smart phones.

When you access an application such as AroundMe on the Apple iPhone, which lists businesses, services, and restaurants in your vicinity, you are using an example of a presence service. Figure 12.7 shows the AroundMe app with some sample results. The presence service is provided by the GPS locator inside the phone, which provides a location through AT&T (the service provider) to the application. Presence is an essential and growing component of cloud-based services, and it adds a tremendous amount of value to the ubiquity that a cloud network offers.

As cloud computing becomes more pervasive and vendors attempt to create federated systems, emerging presence services will become more important. Microsoft's Windows Identity Foundation (described in the previous section) created under the Geneva Framework project is one example of an attempt to create a claims-based presence system. WIF allows different systems to interoperate using a variety of authentication methods, including LDAP and Active Directory, OpenID, LiveID, Microsoft CardSpace, and Novell Digital Me.

FIGURE 12.7

The AroundMe iPhone app is an example of an application that makes use of a presence service.

The Internet Engineering Task Force (IETF) has developed a standard called the Extensible Messaging and Presence Protocol (XMPP) that can be used with a federation system called the Jabber Extensible Communications Platform (Jabber XCP) to provide presence information. Among the services that use Jabber XCP are the Defense Information Systems Agency (DISA), Google Talk, Earthlink, Facebook, the National Weather Service, Twitter, the U.S. Marine Corps, and the U.S. Joint Forces Command (USJFCOM). Jabber XCP is popular because it is an extensible development platform that is platform-independent and supports several communications protocols, such as the Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) and Instant Messaging and Presence Service (IMPS).

The notion of applying presence services over the standard Service Oriented Architecture (SOA) protocols such as SOAP/REST/HTTP is that in SOA all these protocols support unidirectional data exchange. You request a service/data, and a response is supplied. SOA architectures don't scale well and can't supply high-speed data transfers required by the collaboration services that are based on presence service technologies. SOA also has the problem of services that have trouble penetrating firewalls. It was these barriers that Jabber and XMPP were created to solve, and you will find that protocol incorporated into a number of cloud computing SaaS services. AOL, Apple, Google, IBM, and others are using this technology in some of their applications today.

UNIT – V

Case-Studies: Using Google Web Services, Using Amazon Web Services, Using Microsoft Cloud Services

5.1 Using Google Web Services

Google is the prototypical cloud computing Services Company, and it supports some of the largest Web sites and services in the world. In this chapter, you learn about Google's applications and services forusers and the various developer tools that Google makes available.

At the center of Google's core business is the company's search technology. Google uses automated technology to index the Web. It makes its search service available to users as a standard search engine and to developers as a collection of special search tools limited to various areas of content. The application of Google's searches to content aggregation has led to enormous societal changes and to a growing trend of disintermediation.

The most important commercial part of Google's activities is its targeting advertising business: AdWords and AdSense. Google has developed a range of services including Google Analytics that supports its targeted advertising business.

Google applications are cloud-based applications. The range of application types offered by Google spans a variety of types: productivity applications, mobile applications, media delivery, social interactions, and many more. The different applications are listed in this chapter. Google has begun to commercialize some of these applications as cloud-based enterprise application suites that are being widely adopted.

Google has a very large program for developers that spans its entire range of applications and services. Among the services highlighted are Google's AJAX APIs, the Google Web Toolkit, and in particular Google's relatively new Google Apps Engine hosting service. Using Google App Engine, you Wecan create Web applications in Java and Python that can be deployed on Google's infrastructure and scaled to a large size.

Exploring Google Applications

Few companies have had as much impact on their industries as Google has had on the computer industry and on the Internet in particular. Some companies may have more Internet users (Microsoft comes to mind) or have a stock valuation higher than Google (Apple currently fits that description), but Google remains both a technology and thought leader for all things Internet. For a company whose motto is –Don't be evil,‖ the impact of consumer tracking and targeted advertising, free sourcing applications, and the relentless assault on one knowledge domain after another has had a profound impact on the lives of many people. I call it the Google Effect.

The bulk of Google's income comes from the sales of target advertising based on information that Google gathers from your activities associated with your Google account or through cookies placed on your system using its AdWords system. In 2009, Google's revenue was $23.6 billion, and it controlled roughly 65 percent of the search market through its various sites and services. The company is highly profitable, and that has allowed Google to create a huge infrastructure as well as launch many free cloud-based applications and services that this chapter details. These applications are offered mostly on a free usage model that represents Google's Software as a Service portfolio. A business model that offers cloud-based services for free that are ―good enough‖ is very compelling. While Google is slowly growing a subscription business selling these applications to enterprises, its revenue represents only a small but growing part of Google's current income.

Google's cloud computing services falls under two umbrellas. The first and best-known offerings are an extensive set of very popular applications that Google offers to the general public. These applications include Google Docs, Google Health, Picasa, Google Mail, Google Earth, and many more. You can access a jump table of Google's cloud-based user applications by following the ―More‖ and ―Even More‖ links on Google's home page to the More Google Products page at http://www.google.com/intl/en/options/ shown in Figure 8.1; these features are described in Table 8.1.

Table 8.1 lists the current Google ―products‖ listed on its Even More page.

TABLE 8.1

Google Products

Product Name	URL	Google Description
Alerts	http://www.google.com/alerts?hl=en	Sends a periodic e-mail alert to you based on your search term. Search news, blogs, discussions, video, or everything.
Blog Search	http://www.google.com/blogsearch?hl=en	Displays an aggregation page from blogs.
Blogger	http://www.blogger.com/start?hl=en	A blogging site for personal blogs. See Chapter 18 for a description of blogging services.
Books	http://books.google.com/books?hl=en	A vast library of book content in the public domain and previews of copy-righted material.
Calendar	http://www.google.com/calendar/render?hl=en	Calendar service for managing sched-ules and events and sharing them with others.
Chrome	http://www.google.com/chrome?hl=en&brand=CHMI	Google's browser and operating system wannabe.
Checkout	http://checkout.google.com/	A payment processing system.
Code	http://code.google.com/intl/en/	Developer tools and resources. Described more fully later in this chapter.
Custom Search	http://www.google.com/coop/cse/?hl=en	Creates a custom search utility for a particular Web site.
Desktop	http://desktop.google.com/en/?ignua=1	Indexes content on your local drive for fast searches. Adds a sidebar with gadgets.
Directory	http://www.google.com/dirhp?hl=en	Search the Web by topics, a la Yahoo!
Docs	http://docs.google.com/	Online productivity applications. Described in Chapter 16.
Earth	http://earth.google.com/intl/en/	An online atlas and mapping service with mashups.
Finance	http://www.google.com/finance	A financial news aggregation service and site.
GOOG-411	http://www.google.com/goog-411/	Mobile phone search.
Google Health	http://www.google.com/health/	Health information management system.

(continued)

TABLE 8.1 *(continued)*

Product Name	URL	Google Description
Groups	http://www.google.com/grphp?hl=en	Discussion groups on specific topics.
iGoogle	http://www.google.com/ig?hl=en&source=mpes	AJAX customized home page.
Images	http://images.google.com/imghp?hl=en	Web image search.
Knol	http://knol.google.com/k?hl=en	Short articles submitted by users.
Labs	http://labs.google.com/	A collection of applications and utilities under development and testing.
Orkut	https://www.orkut.com/	Social media service with instant messaging. Described in Chapter 18.
Maps	http://maps.google.com/?hl=en	Mapping and direction service.
Maps for Mobile	http://www.google.com/mobile/default/maps.html	Mapping and direction service. Works with GPS on mobile devices.
Mobile	http://www.google.com/mobile/	Mobile search using voice and location.
News	http://news.google.com/news?ned=en	News aggregation service and Web site.
Pack	http://pack.google.com/?hl=en	Free Windows-based software selected by Google, including Chrome, apps, Desktop, Earth, Picasa, Adobe Reader, Talk, RealPlayer, Skype, and others.
Patent Search	http://www.google.com/patents?hl=en	Patent and trademark search of the United States Patents and Trademark Office.
Picasa	http://picasa.google.com/intl/en/	Photo-editing and management software.
Product Search	http://www.google.com/products	Shopping search function.
Reader	http://www.google.com/reader/view/?hl=en&source=mmm-en	An RSS reader.
Scholar	http://www.google.com/schhp?hl=en	Search site for research and scholarly work from many disciplines.
Search for Mobile	http://www.google.com/mobile/default/search.html	Google's search application optimized for mobile devices.
Sites	http://sites.google.com/	Web site and wiki creation and staging tool.
SketchUp	http://sketchup.google.com/intl/en/	Allows users to create 3D models and share them with others.

Product Name	URL	Google Description
Talk	http://www.google.com/talk/	Instant messaging and chat utility. Can be integrated in Gmail.
Toolbar	http://toolbar.google.com/intl/en/	Provides search features inside different browsers.
Translate	http://translate.google.com/?hl=en	Language translation utility.
Trends	http://www.google.com/trends	Statistical information on different search terms.
Videos	http://video.google.com/?hl=en	Searches for videos on the Web.
Voice	http://voice.google.com/	Free phone service, formerly called Grand Central. Described in Chapter 19.
Web Search	http://www.google.com/webhp?hl=en	Google's core Web search engine of indexed pages sorted with page rank.
Web Search Features	http://www.google.com/intl/en/help/features.html	A help page for special Web searches in Google.
YouTube	http://www.youtube.com/	Flash video sharing site. Described in Chapter 19.

Source: http://www.google.com/intl/en/options/.

Because I cover many of these products in other chapters in this book, the focus in this chapter is to survey the applications that Google offers, to understand why Google offers them as services, and to gain some insight into their potential future role. Google's cloud-based applications have put many other vendors' products—such as office suites, mapping applications, image-management programs, and many other categories of traditional shrink-wrapped software—under considerable pressure.

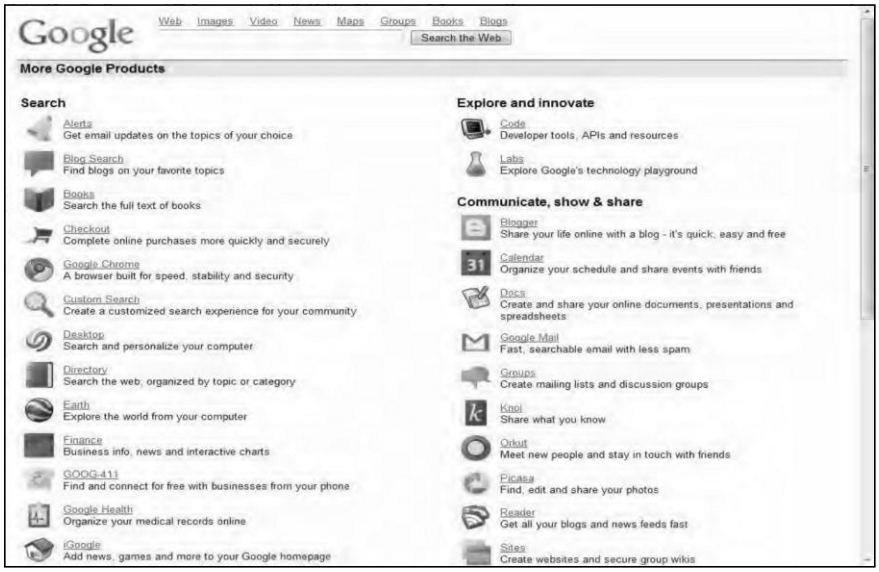
The second of Google's cloud offerings is its Platform as a Service developer tools. In April 2008, Google introduced a development platform for hosted Web applications using Google's infrastructure called the Google App Engine (GAE). The goal of GAE is to allow developers to create and deploy Web applications without worrying about managing the infrastructure necessary to have their applications run. GAE applications may be written using many high-level programming languages (most prominently Java and Python) and the Google App Engine Framework, which lowers the amount of development effort required to get an application up and running. Google also allows a certain free level of service so that the application must exceed a certain level of processor load, storage usage, and network bandwidth (Input/Output) before charges are assessed.

Google App Engine applications must be written to comply with Google's infrastructure. This narrows the range of application types that can be run on GAE; it also makes it very hard to port applications to GAE. After an application is deployed on

GAE, it is also difficult to port that application to another platform. Even with all these limitations, the Google App Engine provides developers a low-cost option on which to create an application that can run on a world-class cloud infrastructure—with all the attendant benefits that this type of deployment can bestow.

FIGURE 8.1

More Google Products equals fewer commercial products.

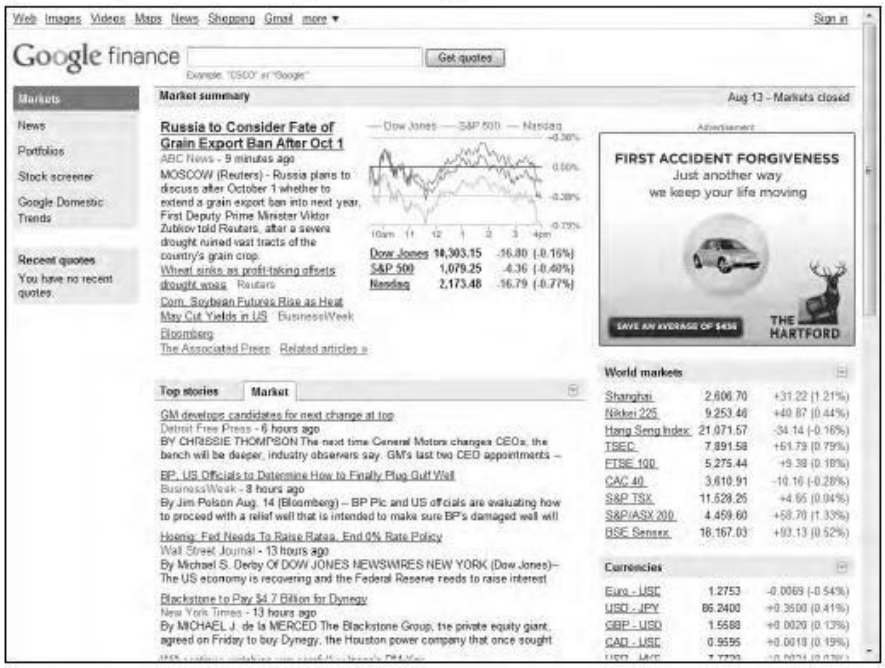


Surveying the Google Application Portfolio

It is fair to say that nearly all the products in Google's application and service portfolio are cloud computing services in that they all rely on systems staged worldwide on Google's one million plus servers in nearly 30 datacenters. Roughly 17 of the 48 services listed leverage Google's search engine in some specific way. Some of these search-related sites search through selected content such as Books, Images, Scholar, Trends, and more. Other sites such as Blog Search, Finance, News, and some others take the search results and format them into an Aggregation page. Google's Finance page at http://www.google.com/finance/ is an example of an Aggregation page. Figure shows one of these aggregation pages: Google Finance.

FIGURE 8.2

Google's Finance page at `http://www.google.com/finance/` is an example of an aggregation page provided by results from Google's search engine.



Indexed search

Google's search technology is based on automated page indexing and information retrieval by Web crawlers, also called spiders or robots. Content on pages is scanned up to a certain number of words and placed into an index. Google also caches copies of certain Web pages and stores copies of documents it finds such as DOC or PDF files in its cache.

Google uses a patented algorithm to determine the importance of a particular page based on the number of quality links to that page from other sites, along with other factors such as the use of keywords, how long the site has been available, and traffic to the site or page. That factor is called the PageRank, and the algorithm used to determine PageRank is a trade secret. Google is always tweaking the algorithm to prevent Search Engine Optimization (SEO) strategies from gaming the system. Based on this algorithm, Google returns what is called a Search Engine Results Page (SERP) for a query that is parsed for its keywords.

It is really important to understand what Google (and other search engines) offers and what it doesn't offer. Google does not search all sites. If a site doesn't register with the search engine or isn't the target of a prominent link at another site, that site may remain undiscovered. Any site can place directions in their ROBOTS.TXT file indicating whether the site can be searched or not, and if so what pages can be searched. Google developed something called the Sitemaps protocol, which lets a Web site list in an XML file information about how the Google robot can work with the site. Sitemaps can be useful in allowing content that isn't browse able to be crawled; they also can be useful as guides to finding media information that isn't normally considered, such as AJAX, Flash, or Silverlight media. The Sitemaps protocol has been widely adopted in the industry.

The dark Web

Online content that isn't indexed by search engines belongs to what has come to be called the ‐Deep Web‖—that is, content on the World Wide Web that is hidden. Any site that suppresses Web crawlers from indexing it is part of the Deep Web. You need go no further than the world's number two Web site, Facebook, for a prominent example of a site that isn't indexed in search engines.

Entire networks exist that aren't searchable, particularly peer-to-peer networks. Ian Clarke's Freenet, which is a P2P network, supports both ‐darknet‖ and ‐opennet‖ connections. Freenet (http://freenetproject.org/) has been downloaded by millions of people.

The Deep Web includes:

- Database generated Web pages or dynamic content
- Pages without links
- Private or limited access Web pages and sites
- Information contained in sources available through executable code such as JavaScript.
- Documents and files that aren't in a form that can be searched, which includes not only media files, but information in non-standard file formats

Although efforts are underway to enable information on the Deep Web to be searchable, the amount of information stored that is not accessible is many times larger than the amount of information that can currently be accessed. Some estimates at the size of the Dark Web suggest that it could be an order of magnitude larger than the content contained in the world's search engines.

It is always a good idea to keep these search engine limitations in mind when you work with this technology.

Aggregation and disintermediation

Aggregation pages are a great user service, but they are very controversial—as are a number of Google's search applications and services. It has long been argued that Google's display of information from various sites violates copyright laws and damages content providers. In several lawsuits, Google successfully defended its right to display capsule information under the Digital Millennium Copyright Act, while in other instances Google responds to requests from interested parties to remove information from its site.

The Authors Guild's filed a class action suit in 2005 regarding unauthorized scanning and copying of books for the creation of the Google Books feature. Google reached a negotiated agreement with the Authors Guild that specified Google's obligations under the fair use exemption. Google argues that the publicity associated with searchable content adds value to that content, and it is clear that this is an argument that will continue into the future.

What is clear is that Google has been a major factor in a trend referred to as disintermediation. Disintermediation is the removal of intermediaries such as a distributor, agent, broker, or some similar functionary from a supply chain. This connects producers directly with consumers, which in many cases is a very good thing. However, disintermediation also has the unfortunate side effect of impacting organizations such as news collection agencies (newspapers, for example), publishers, many different types of retail outlets, and many other businesses, some of which played a positive role in the transactions they were involved in.

Google began to introduce productivity applications starting in 2004 with Gmail. The expansion of these services has continued unabated ever since. Some of these applications are homegrown, but many of them were acquired by acquisition. An example of an acquired product is Writely, the online word processor that is now at the heart of Google Docs.

Productivity applications and services

These products store your information online in a form that Google can use to build a profile of your activities, and it is unclear how the company uses the information it stores. Google states that your information is never viewed individually by humans, and the company lists its policies in the Privacy Center, which you can find

at http://www.google.com/privacypolicy.html. Google has been vigilant in protecting its privacy reputation, but the collection of such a large amount of personal data must give any thoughtful person reason for pause.

Enterprise offerings

As Google has built out its portfolio, it has released special versions of its products for the enterprise. The following are among Google's products aimed at the enterprise market:

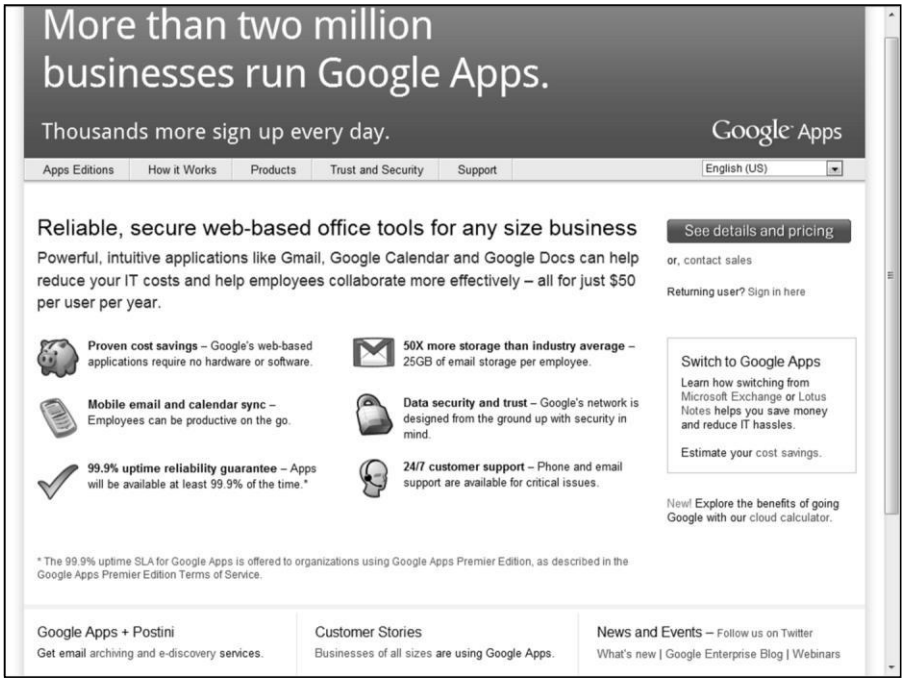
- **Google Commerce Search** (http://www.google.com/commercesearch/): This is a search service for online retailers that markets their products in their site searches with a number of navigation, filtering, promotion, and analytical functions.
- **Google Site Search** (http://www.google.com/sitesearch/): Google sells its search engine customized for enterprises under the Google Site Search service banner. The user enters a search string in the site's search, and Google returns the results from that site.
- **Google Search Appliance** (http://www.google.com/enterprise/gsa): This server can be deployed within an organization to speed up both local (Intranet) and Internet searching. The three versions of the Google Search Appliance can store an index of up to 300,000 (GB-1001), 10 million (GB-5005), or 30 million (GB-8008) documents. Beyond indexing, these appliances have document management features, perform custom searches, cache content, and give local support to Google Analytics and Google Sitemaps.
- **Google Mini** (http://www.google.com/enterprise/mini/): The Mini is the smaller version of the GSA that stores 300,000 indexed documents. Google also has some success in marketing its productivity applications as office suites to organizations. Google uses different names for the different bundles under a branded program called Google Apps for Business (http://www.google.com/apps/intl/en/business/index. html). Figure 8.3 shows the home page for Google's various office suite bundles. The company has packages for governments, schools, non-profits, and ISPs (a reseller program). Google claims that some 8 million students now use Google Apps, and Google Apps has had some large government purchases, such as the City of Los Angeles.

For business and other organizations such as governmental agencies, the company has a branded Google Apps Premier Edition, which is a paid service. The different versions offer Gmail, Docs, and Calendar as core applications. The Premier Edition adds 25GB of Gmail storage, e-mail server synchronization, Groups, Sites, Talk, Video, enhanced security, directory services, authentication and authorization services, and the customer's own supported domain—all hosted in the cloud. Premium Edition also adds access to Google APIs and a 24/7 support service with a 99.9-percent

uptime guarantee Service Level Agreement. The cost per use is $50 per user account/per year.

FIGURE 8.3

Google Apps for Business is the commercial versions of the company's productivity suites.



To support Google's Premier and Education Editions' Gmail, Google purchased the Postini archiving and discovery service. Google Postini Services (http://www.google.com/ postini/) provides security services such as threat assessment, proactive link blocking and Web policy enforcement, e-mail message encryption, message archiving, and message discovery services. These are paid services that add from $12 to $45 per user/per year, based on the options chosen. Postini allows e-mail to be retained for up to 10 years and can be used to demonstrate regulatory compliance.

Many of Google's productivity applications are quite capable, but none is a state-of-the-art client you might expect to find in a locally installed office suite. When compared one-on-one to Microsoft Office applications, Google's online offerings give users the essential features for a fraction of the Microsoft Office price.

Most sophisticated users prefer Microsoft Office, but for the average user (that is most people) Google App bundles are good enough. When that low price is coupled with the collaborative tools and features Google offers, the value of Google Apps will be increasingly more appealing. We can reasonably expect that cloud-based

productivity apps will put their shrink-wrapped competitors under great pressure. Microsoft's current strategy of putting crippled Office applications on the Web in Windows Live isn't going to be competitive.

AdWords

AdWords (http://www.google.com/AdWords) is a targeted ad service based on matching advertisers and their keywords to users and their search profiles. This service transformed Google from a competent search engine into an industry giant and is responsible for the majority of Google's revenue stream. AdWords' two largest competitors are Microsoft adcenter (http:// adcenter.microsoft.com/) and Yahoo! Search Marketing (http://searchmarketing. yahoo.com/).

Ads are displayed as text, banners, or media and can be tailored based on geographical location, frequency, IP addresses, and other factors. AdWords ads can appear not only on Google.com, but on AOL search, Ask.com, and Netscape, along with other partners. Other partners belonging to the Google Display Network can also display AdSense ads. In all these cases, the AdWords system determines which ads to match to the user searches.

Here's how the system works: Advertisers bid on keywords that are used to match a user to their product or service. If a user searches for a term such as ‒develop abdominal muscles,‖ Google returns products based on those terms. You might see an ad with Chuck Norris selling a modernday version of a torture rack that, if it doesn't give you a six-pack, at least makes your wallet lighter. Up to 12 ads per search can be returned.

Google gets paid for the ad whenever a user clicks it. The system is referred to as pay-per-click advertising, and the success of the ad is measured by what is called the click-through rate (CTR). Google calculates a *quality score* for ads based on the CTR, the strength of the connection between the ad and the keywords, and the advertiser's history with Google. This quality score is a Google trade secret and is used to price the minimum bid of a keyword.

In 2007, Google purchased DoubleClick, an Internet advertising services company. DoubleClick helps clients create ads, provides hosting services, and tracks results for analysis. DoubleClick ads leave browser cookies on systems that collect information from users that determine the number of times a user has been exposed to a particular ad, as well as various system characteristics. Some spyware trackers flag DoubleClick cookies as spyware. Both AdWords and DoubleClick are sold as packages to large clients.

Google Analytics

Google Analytics (GA; http://google.com/analytics) is a statistical tool that measures the number and types of visitors to a Web site and how the Web site is used. It is offered as a free service and has been adopted by many Web sites. GA is built on the Urchin 5 analytical package that Google acquired in 2006. Figure 8.4 shows the Google Analytics home page.

According to Builtwith.com (http://trends.builtwith.com/analytics/Google-Analytics), Google Analytics was in use on 54 percent of the top 10,000 and 100,000, and 35 percent of the top one million of the world's Web sites. Builtwith.com speculates that Google Analytics JavaScript tag is the most widely used URL in the world today. The service BackendBattles.com (http://www.backendbattles.com/backend/Google_Analytics) sets GA's market share at 57 percent for the top 10,000 sites.

FIGURE 8.4

Google Analytics is the most widely used Web traffic analysis tool on the Internet.

Analytics works by using a JavaScript snippet called the Google Analytics Tracking Code (GATC) on individual pages to implement a *page tag.* When the page loads, the JavaScript runs and creates a first-party browser cookie that can be used to

manage return visitors, perform tracking, test browser characteristics, and request tracking code that identifies the location of the visitor. GATC requests and stores information from the user's account. The code stored on the user's system acts like a beacon and collects visitor data that it sends back to GA servers for processing.

Among the visitors that can be tracked are those that land from search engines; referral links in e-mail, documents, and Web pages; display ads; PPC networks; and some other sources. GA aggregates the data and presents the information in a visual form. GA also is connected to the AdWords system so it can track the performance of particular ads in different contexts. You can view referral location statistics and time spent on a page, and you can filter by visitor site. GA lets you save and store up to 50 individual site profiles, provided the site has less than 5 million pageviews per month. This restriction is lifted for an AdWords subscription.

GA cookies are blocked by a number of technologies, such as Firefox Adblock and NoScript or by turning off JavaScript execution in other browsers. You also can delete GA cookies manually or block them, which also defeats the system.

Google Translate

Of all the Google applications, the one that might have significant immediate impact is Google Translate. Computer technology is very close to having the necessary hardware and software to realize the dream of a ‒universal translator‖ that the TV show *Star Trek* proposed some 45 years ago. The current version of Google Translate performs machine translation as a cloud service between two of your choice of 35 different languages. That's not truly universal, but until aliens appear, it will do for most people.

Google Translate was introduced in 2007 and replaced the SYSTRAN system that many other computer services utilize. The translation method uses a statistical approach that was first developed by Franz-Joseph Och in 2003. Och now heads the Translate effort at Google.

Translate uses what is referred to as a corpus linguistics approach to translation. You start off building a translation system for a language pair by collecting a database of words and then matching that database to two bilingual text corpuses. A text corpus or parallel collection is a database of word- and phrase-usage taken from the language in everyday use obtained by examining documents translated by professionals to software analysis. Among the documents that are analyzed are the translations of the United Nations and European Parliament, among others.

Google Translate can be accessed directly at http://translate.google.com/translate_t?hl=en#, where you can select the language pair to be translated. You can do the following:

- Enter text directly into the text box, and click the Translate button to have the text translated.

If you select the Detect Language option, Translate tries to determine the language automatically and translate it into English.

- Enter a URL for a Web page to have Google display a copy of the translated Web page.
- Enter a phonetic equivalent for script languages. Upload a document to the page to have it translated.

Translate parses the document into words and phrases and applies its statistical algorithm to make the translation. As the service ages, the translations are getting more accurate, and the engine is being added to browsers such as Google Chrome and through extension into Mozilla Firefox. The Google Toolbar offers page translation as one of its options, selectable in the Tools settings.

The Google Translator Toolkit (http://translate.google.com/toolkit) shown in Figure 8.5 provides a means for using the Translate to perform translations that you can edit. Shown in the figure is the translation of an article from the English version of Wikipedia into Spanish. The toolkit provides access to tools to aid you in editing the translation.

Translation services have been in development for many years. IBM has had a large effort in this area, and the Microsoft Bing search engine also has a translation engine. There are many other translation engines, and some of them are even cloud-based like Google Translate. What makes Google's efforts potentially unique is the company's work in language transcription—that is, the conversion of voice to text. As part of Google Voice and its work with Android-based cell phones, Google is sampling and converting millions and millions of conversations. Combining these two Web services together could create a translation device based on a cloud service that would have great utility.

FIGURE 8.5

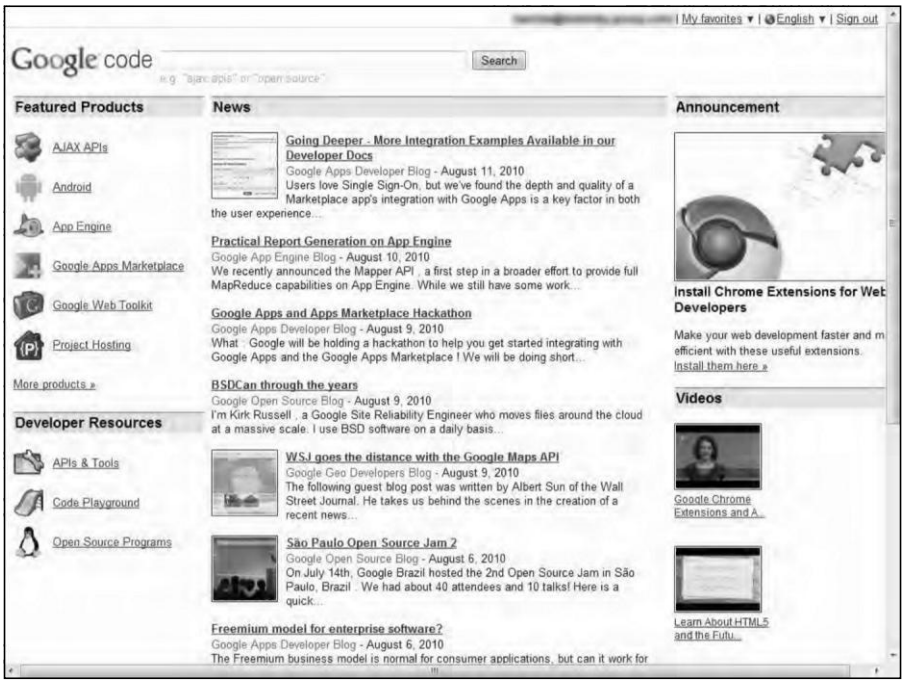
The Google Translator Toolkit lets you translate documents, Web pages, and other material from one language to another and provides tools to improve on the translation.

Exploring the Google Toolkit

Google has an extensive program that supports developers who want to leverage Google's cloudbased applications and services. These APIs reach into every corner of Google's business. Google's Code Home page for developers may be found at http://code.google.com and is shown in Figure 8.6. From this site, you can access developer tools, information on how to use its various APIs to include Google services in your own work, and technical resources.

FIGURE 8.6

Google's Code page at http://code.google.com/intl/en/



Google has a number of areas in which it offers development services, including the following:

- **AJAX APIs** (http://code.google.com/intl/en/apis/ajax/) are used to build widgets and other applets commonly found in places like iGoogle. AJAX provides access to dynamic information using JavaScript and HTML.
- **Android** (http://developer.android.com/index.html) is a phone operating system development.
- **Google App Engine** (http://appengine.google.com/) is Google's Platform as a Service (PaaS) development and deployment system for cloud computing applications.
- **Google Apps Marketplace** (http://code.google.com/intl/en/googleapps/ marketplace/) offers application development tools and a distribution channel for cloud-based applications.
- **Google Gears** (http://gears.google.com/) is a service that provides offline access to online data.

Google Gears includes a database engine installed on the client that caches data and synchronizes it. Gears allows cloud-based applications to be available to a client

even when a network connection to the Internet isn't available. Using Gears, you could work on your mail in Gmail offline, for example.

- **Google Web Toolkit** (GWT; http://code.google.com/webtoolkit) is a set of development tools for browser-based applications.

GWT is an open-source platform that has been used to create Google Wave and Google AdWords. GWT allows developers to create AJAX applications using Java or with the GWT compiler using JavaScript.

- **Project Hosting** (http://code.google.com/intl/en/projecthosting/) is a project management tool for managing source code.

The Google APIs

Most Google services are exposed by an API, which is why you find a version of Google's search engine, Google Maps, YouTube videos, Google Earth, AdWords, AdSense, and even elements of Google Apps exposed in many other Web sites. You can get to the listing of the Google APIs by clicking the More Products link on the Code page (refer to Figure 8.6). The page you see is http://code.google.com/intl/en/more/, which is shown in Figure 8.7.

Google's APIs can be categorized as belonging to the following categories:

- **Ads and AdSense:** These APIs allow Google's advertising services to be integrated into Web applications. The most commonly used services in this category are AdWords, AdSense, and Google Analytics.
- **AJAX:** The Google AJAX APIs provide a means to add content such as RSS feeds, maps, search boxes, and other information sources by including a snippet of JavaScript into your code.
- **Browser:** Google has several APIs related to building browser-based applications, including four for the Chrome browser. This category includes the Google Cloud Print API, the Installable Web Apps API for creating installation packages, the Google Web Toolkit for building AJAX applications using Java, and V8, which is a high-performance JavaScript engine.
- **Data:** The Data APIs are those that exchange data with a variety of Google services. The list of Google Data APIs includes Google Apps, Google Analytics, Blogger, Base, Book, Calendar, Code Search, Google Earth, Google Spreadsheets, Google Notebook, and Picasa Web Albums.
- **Geo:** A number of APIs exist to give location-specific information hooking into maps and geo-specific databases. Some of the more popular APIs in this category include Google Earth, Directions, JavaScripts Maps, Maps API for Flash, and Static Maps.

- **Search:** The search APIs leverage Google's core competency and its central service. APIs such as Google AJAX Search, Book Search, Code Search, Custom Search, and Webmaster Tools Data APIs allow developers to include Google searches in their applications and web sites.

Social: Many Google APIs are used for information exchange and communication tools. They support applications such as Gmail, Calendar, and others, and they provide a set of foundation services. The popular social APIs are Blogger Data, Calendar, Contacts, OpenSocial, Picasa, and YouTube.

FIGURE 8.7

Google's More Code page exposes the extensive set of APIs offered by Google for its various products.

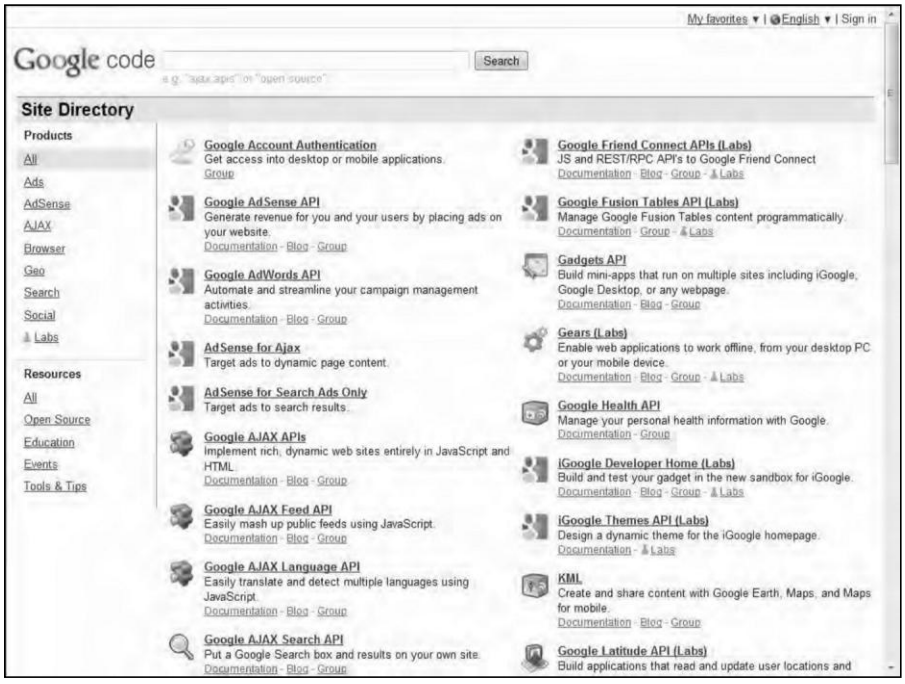


Table 8.2 summarizes the many different Google APIs.

TABLE 8.2

Google APIs

API Name	URL	Category	Google Description
Google Accounts Authentication	http://code.google.com/apis/accounts/	Infrastructure	Get access into desktop or mobile applications.
Google AdWords API	http://code.google.com/apis/adwords/	Ads	Automate and streamline your campaign management activities.
AdSense for AJAX	http://code.google.com/apis/afa/	Ads, AJAX	Target ads to dynamic page content.
AdSense for Search Ads Only	http://code.google.com/apis/afs-ads-only/	Ads	Target ads to search results.
Google AJAX APIs	http://code.google.com/apis/ajax/	AJAX	Implement rich, dynamic Web sites entirely in JavaScript and HTML.
Google AJAX Feed API	http://code.google.com/apis/ajaxfeeds/	AJAX	Easily mash up public feeds using JavaScript.
Google AJAX Language API	http://code.google.com/apis/ajaxlanguage/	AJAX	Easily translate and detect multiple languages using JavaScript.
Google AJAX Search API	http://code.google.com/apis/ajaxsearch/	AJAX, Search	Put a Google Search box and results on your own site.
Google Analytics	http://code.google.com/apis/analytics/	Ads	Track your site traffic, and write your own client applications that use Analytics data in the form of Google Data API feeds.
Android	http://code.google.com/android/	Infrastructure	Build mobile apps for Android, a software stack for mobile devices.
Google App Engine	http://code.google.com/appengine/	Infrastructure	Run your Web applications on Google's infrastructure.
Google Apps Script	http://code.google.com/googleapps/appsscript/	Productivity	Automate tasks across Google products.
BigQuery (Labs)	http://code.google.com/apis/bigquery/	Labs	Interactively analyze large datasets.
Google Apps	http://code.google.com/googleapps/	Productivity	Extend Google Apps, integrate with other systems, or build new apps.

API Name	URL	Category	Google Description
Google Apps Marketplace	http://code.google.com/googleapps/marketplace/	Productivity	Sell integrated applications to millions of Google Apps users.
Gmail APIs and Tools	http://code.google.com/apis/gmail/	Labs	Create gadgets for Gmail, and interact with the inbox.
Google Base Data API (Labs)	http://code.google.com/apis/base/	Labs	Manage Google Base content programmatically.
Blogger Data API (Labs)	http://code.google.com/apis/blogger/	Labs, Social	Enable your apps to view and update Blogger content.
Google Books Search APIs (Labs)	http://code.google.com/apis/books/	Labs, Search	Search the complete index of Book Search, and integrate with its social features.
Google Buzz (Labs)	http://code.google.com/apis/buzz/	Labs, Social	Share updates, photos, videos, and more, and start conversations about the things you find interesting.
Google Calendar APIs and Tools	http://code.google.com/apis/calendar/	Social	Create and manage events, calendars, and gadgets for Google Calendar.
Chart Tools	http://code.google.com/apis/charttools/	Productivity	Add charts and graphs to your Web page.
Google Checkout	http://code.google.com/apis/checkout/	Infrastructure	Start selling on your Web site.
Chromium	http://code.google.com/chromium/	Browser	Contribute to the open-source project behind Google Chrome.
Google Chrome Frame	http://code.google.com/chrome/chromeframe/	Browser	Enable open Web technologies and Google Chrome's fast JavaScript implementation within Internet Explorer.
Google Chrome Extensions (Labs)	http://code.google.com/chrome/extensions/	Browser, Labs	Modify and enhance the functionality of Google Chrome.
Installable Web Apps (Labs)	http://code.google.com/chrome/apps/	Browser, Labs	Package your Web apps for installation in Google Chrome.
Closure Tools	http://code.google.com/closure/	Labs	Create powerful and efficient JavaScript.
Google Cloud Print (Labs)	http://code.google.com/apis/cloudprint/	Browser, Labs	Enable any app (Web, desktop, mobile) on any device to print to any printer.

(continued)

TABLE 8.2 *(continued)*

	URL	Category	Google Description
Google Code Search Data API (Labs)	`http://code.google.com/apis/codesearch/`	Labs, Search	Enable your apps to view data from Code Search.
Google Contacts API	`http://code.google.com/apis/contacts/`	Social	Allow your apps to view and update user contacts.
Google Coupon Feeds (Labs)	`http://code.google.com/apis/coupons/`	Labs	Provide coupon listings that are included in Google search results.
Google Custom Search API	`http://code.google.com/apis/customsearch/`	Ads, Search	Create a custom search engine for your Web site or a collection of Web sites.
Google DoubleClick for Publishers (Labs)	`http://code.google.com/apis/dfp/`	Ads, Labs	Build applications that interact directly with Google's next-generation display advertising platform.
Google Data Protocol	`http://code.google.com/apis/gdata/`	Infrastructure	A simple, standard protocol for reading and writing data on the Web.
Google Desktop APIs (Labs)	`http://code.google.com/apis/desktop/`	Labs, Search	Create gadgets and indexing plugins for Google Desktop.
Google Documents List Data API	`http://code.google.com/apis/documents/`	Infrastructure	Enable your apps to view and update your list of Google Documents.
Google Interactive Media Ads (Labs	`http://code.google.com/apis/ima/`	Ads, Labs	Google Interactive Media Ads enable publishers to request and display ads into video, audio, and game content.
Google Earth API	`http://code.google.com/apis/earth/`	AJAX, Geo	Embed Google Earth into your Web page.
Google Plugin for Eclipse	`http://code.google.com/eclipse/`	Infrastructure	Enjoy simplified development of GWT and App Engine projects in the Eclipse IDE.
Feedburner API (Labs)	`http://code.google.com/apis/feedburner/`	Labs	Interact with FeedBurner's feed management and awareness-generating capabilities.
Google Finance Data API (Labs)	`http://code.google.com/apis/finance/`	Labs	View and update Finance content in the form of Google Data API feeds.

	URL	Category	Google Description
Google Friend Connect APIs (Labs)	http://code.google.com/apis/friendconnect/	Labs, Social	JS and REST/RPC API's to Google Friend Connect.
Google Fusion Tables API (Labs)	http://code.google.com/apis/fusiontables/	Labs	Manage Google Fusion Tables content programmatically.
Gadgets API	http://code.google.com/apis/fusiontables/	Social	Build mini-apps that run on multiple sites, including iGoogle, Google Desktop, or any Web page.
Gears (Labs)	http://code.google.com/apis/gears/	AJAX, Labs	Enable Web applications to work offline, from your desktop PC, or your mobile device.
Google Health API	http://code.google.com/apis/health/	Productivity	Manage your personal health information with Google.
iGoogle Developer Home (Labs)	http://code.google.com/apis/igoogle/	Labs, Social	Build and test gadgets for iGoogle.
iGoogle Themes API (Labs)	http://code.google.com/apis/themes/	Labs	Design a dynamic theme for the iGoogle home page.
KML	http://code.google.com/apis/kml/	Geo	Create and share content with Google Earth, Maps, and Maps for mobile.
Google Latitude API (Labs)	http://code.google.com/apis/latitude/	Geo, Labs	Build applications that read and update user locations and location histories.
Google Libraries API	http://code.google.com/apis/libraries/	AJAX	Load open-source JavaScript libraries.
Google Moderator API (Labs)	http://code.google.com/apis/moderator/	Labs	Collect ideas, questions, and recommendations from audiences of any size.
Google Geocoding API	http://code.google.com/apis/maps/documentation/geocoding/	AJAX, Geo	Convert addresses from geographic coordinates.
Google Directions API	http://code.google.com/apis/maps/documentation/directions/	AJAX, Geo	Plot directions using a variety of transportation options.
Google JavaScript Maps API	http://code.google.com/apis/maps/documentation/javascript/	AJAX, Geo	Integrate Google's interactive maps with data on your site.

(continued)

TABLE 8.2 *(continued)*

	URL	Category	Google Description
Google Maps API for Flash	`http://code.google.com/apis/maps/documentation/flash/`	Geo	Integrate Google Maps in Flash applications.
OpenSocial	`http://code.google.com/apis/opensocial/`	AJAX, Social	Build social applications that work across many Web sites.
Orkut Developer Home	`http://code.google.com/apis/orkut/`	Social	Create social applications for the millions of global Orkut users.
Google Project Hosting	`http://code.google.com/projecthosting/`	Infrastructure	Host your open-source project on Google Code.
Picasa APIs (Labs)	`http://code.google.com/apis/picasa/`	Labs, Social	Create custom buttons and upload files to third-party services.
Picasa Web Albums Data API	`http://code.google.com/apis/picasaweb/`	Social	Include Picasa Web Albums in your application or Web site.
Google PowerMeter API (Labs)	`http://code.google.com/apis/powermeter/`	Labs	Integrate with Google PowerMeter.
Google Prediction API (Labs)	`http://code.google.com/apis/predict/`	Labs	Add predictions to your applications.
PubSubHubbub	`http://code.google.com/apis/pubsubhubbub/`	Labs, Social	Turn your Atom and RSS feeds into real-time streams.
reCAPTCHA (Labs)	`http://code.google.com/apis/recaptcha/`	AJAX, Labs	Digitize books with this anti-bot service.
Google Safe Browsing APIs (Labs)	`http://code.google.com/apis/safebrowsing/`	Labs	Download lists of suspected phishing and malware URLs.
Google Secure Data Connector	`http://code.google.com/securedataconnector/`	Infrastructure	Connect data from behind the firewall to Google Apps.
Google Sidewiki API	`http://code.google.com/apis/sidewiki/`	Labs, Social	Enable your apps to view data from Google Sidewiki.
Google Sites Data API	`http://code.google.com/apis/sites/`	Labs	Enable your apps to modify content within a Google Site.
Google SketchUp Ruby API	`http://code.google.com/apis/sketchup/`	Geo	Extend Google SketchUp with Ruby.
Social Graph API (Labs)	`http://code.google.com/apis/socialgraph/`	Labs, Social	Enable users to quickly add their public social connections to your site.

	URL	Category	Google Description
Google Static Maps API	http://code.google.com/apis/maps/documentation/staticmaps/	Geo	Embed a Google Maps image on your Web site without requiring JavaScript or any dynamic page loading.
Google Storage for Developers (Labs)	http://code.google.com/apis/storage/	Labs	Store and share your data in the Google cloud.
Google Talk for Developers (Labs)	http://code.google.com/apis/talk/	Labs, Social	Connect your client or network to the Google Talk network, add chatback, or customize the Google Talk gadget.
Google Transit Feed Specification	http://code.google.com/transit/spec/transit_feed_specification.html	Geo	Provide public transit route and schedule information for Google Maps and more.
Google Translator Toolkit Data API	http://code.google.com/apis/gtt/	Labs	Build applications that can access and update translation-related data.
V8	http://code.google.com/apis/v8/	Browser	Google's high-performance, open-source, JavaScript engine.
Google Wave API	http://code.google.com/apis/wave	Labs, Social	Build extensions for Google Wave or embed Google Waves in your site.
Google Web Elements	http://www.google.com/webelements/	Infrastructure	Add your favorite Google products to your own Web site.
Google Web Toolkit	http://code.google.com/webtoolkit/	AJAX, Browser	Build AJAX apps in the Java language.
Google Webmaster Tools Data API (Labs)	http://code.google.com/apis/webmastertools/	Labs, Search	View and update site information and Sitemaps in the form of feeds.
YouTube API	http://code.google.com/apis/youtube/	Social	Integrate YouTube videos into your Web site or application.

Source: http://code.google.com/intl/en/more/.

Working with the Google App Engine

Google App Engine (GAE) is a Platform as a Service (PaaS) cloud-based Web hosting service on Google's infrastructure. Figure 8.8 shows the GAE home page at http://code.google.com/ intl/en/appengine/. This service allows developers to build and deploy Web applications and have Google manage all the infrastructure needs, such as monitoring, failover, clustering, machine instance management, and so forth. For an application to run on GAE, it must comply with Google's platform standards, which

narrows the range of applications that can be run and severely limits those applications' portability.

GAE supports the following major features:

- Dynamic Web services based on common standards
- Automatic scaling and load balancing
- Authentication using Google's Accounts API
- Persistent storage, with query access sorting and transaction management features
- Task queues and task scheduling
- A client-side development environment for simulating GAE on your local system
- One of either two runtime environments: Java or Python

When you deploy an application on GAE, the application can be accessed using your own domain name or using the Google Apps for Business URL.

FIGURE 8.8

The Google App Engine page at http://code.google.com/intl/en/appengine/

Google App Engine currently supports applications written in Java and in Python, although there are plans to extend support to more languages in the future. The service is meant to be languageagnostic. A number of Java Virtual Machine languages are compliant with GAE, as are several Python Web frameworks that support the Web Server Gateway Interface (WSGI) and CGI. Google has its own Webapp framework

designed for use with GAE. The AppScale (http://appscale. cs.ucsb.edu/) open-source framework also may be used for running applications on GAE.

To encourage developers to write applications using GAE, Google allows for free application development and deployment up to a certain level of resource consumption. Resource limits are described on Google's quota page at http://code.google.com/appengine/docs/ quotas.html, and the quota changes from time to time.

Google uses the following pricing scheme:

- CPU time measured in CPU hours is $0.10 per hour.
- Stored data measured in GB per month is $0.15 per GB/month.
- Incoming bandwidth measured in GB is $0.10 per GB.
- Outgoing bandwidth measured in GB is $0.12 per GB.
- Recipients e-mailed is $0.0001 per recipient.

The pricing page for Google AppEngine may be found at: http://code.google.com/ appengine/docs/billing.html. The current resource limits are shown in Table 8.3. Consumption of resources beyond the free limit is generally on a pay-as-you-go basis, although in certain circumstances, Google may allow for additional free usage. When you enable billing for an application deployed to GAE, you pay for consumption of CPU, network I/O, and other usage above the level of the free quotas that GAE allows.

TABLE 8.3

Apps Quota Limits

Resource Quotas	Free Default Quota	Billing Enabled Default Quota
Applications per developer	10	No fixed limit
Application size	150MB	No fixed limit
Bandwidth limit (in and out)	1GB (each), up to 56MB/minute	1GB free and 1,046GB max, up to 10GB/min rate
CPU usage	6.5 CPU-hours/day, up to 15 CPU-minutes/minute	6.5 CPU-hours/day free to 1,729 CPU-hours/day maximum, up to 72 CPU-minutes/minute maximum rate
Datastore API calls	10 million/day, up to 57,000 queries/min	200 million queries/day, up to 129 queries/min
Data received from API	115GB, up to 659MB/min	695GB, up to 1,484MB/min

(continued)

TABLE 8.3 *(continued)*

Resource Quotas	Free Default Quota	Billing Enabled Default Quota
Data sent to API	12GB, up to 68MB/min	72GB, up to 153MB/min
Data storage	1GB	1GB free, no maximum
Datastore CPU Time	60 CPU-hours, up to 20 CPU-min/min	1,200 CPU-hours, up to 50 CPU-min/min
E-mails	2,000/day, up to 8 recipients/min	2,000 free to 7.4 million recipients max, up to 5,100 recipients/min
HTTP requests	1,300,000/day, up to 7,400 requests/minute	43,000,000 requests, up to 30,000 requests/min rate
Indexes	100	200
Storage per application (Blobstore)	1GB	1GB free, no limit
Storage API calls (Blobstore)	No free quota	140 million calls/day, up to 72,000 calls/min
Storage item limit	1GB	1 GB free, no maximum
Time per request allowed	30 sec	30 sec
URLFetch API calls	657,000/day up to 3,000 calls/min	46 million calls/day up to 32,000 calls/min

Source: `http://code.google.com/appengine/docs/quotas.html`.

Applications running in GAE are isolated from the underlying operating system, which Google describes as running in a sandbox. This allows GAE to optimize the system so Web requests can be matched to the current traffic load. It also allows applications to be more secure because applications can connect only to computers using the specified URLs for the e-mail and fetch services using HTTP or HTTPS over the standard well-known ports. URL fetch uses the same infrastructure that retrieves Web pages on Google. The mail service also supports Gmail's messaging system.

Applications also are limited in that they can only read files; they cannot write to the file system directly. To access data, an application must use data stored in the memcache (memory cache), the datastore, or some other persistent service. Memcache is a fast in-memory key-value cache that can be used between application instances. For persistent data storage of transactional data, the datastore is used. Additionally, an application responds only to a specific HTTP request—in real-time, part of a queue, or scheduled—and any request is terminated if the response requires more than 30 seconds to complete.

GAE has a distributed datastore system that supports queries and transactions. This datastore is non-relational or ‑schema-less,‖ but it does store data objects or entities that are assigned properties. In your queries, you can use entities filtered by

kind or type and also sorted by properties. You can find a list of the various property types at http://code.google.com/appengine/docs/python/datastore/typesandpropertyclasses.html; the list includes strings, booleans, float, datetime, blob, text, and other property types. Each application can structure its own sets of data entities. The datastore uses an optimistic concurrency control and maintains strong consistency. An application can execute transactions with multiple operations, and they either all succeed or fail as a unit. To support the distributed nature of the datastore, the concept of an entity group is employed. Transactions manage entities as a single group, and entity groups are stored together in the system so operations can be performed faster.

The App Engine relies on the Google Accounts API for user authentication, the same system used when you log into a Google account. This provides access to e-mail and display names within your app, and it eliminates the need for an application to develop its own authentication system. Applications can use the User API to determine whether a user belongs to a specific group and even whether that person is an administrator for your application.

Many applications have been built and are running on Google App Engine. To get some idea of the range of applications that have been developed, you may want to visit the Google App Engine Gallery.

Using Amazon Web Services

Amazon.com is one of the most important and heavily trafficked Web sites in the world. It provides a vast selection of products using an infrastructure based on Web services. As Amazon.com has grown, it has dramatically grown its infrastructure to accommodate peak traffic times. Over time the company has made its network resources available to partners and affiliates, which also has improved its range of products.

Starting in 2006, Amazon.com made its Web service platform available to developers on a usage-basis model. The technologies described in this chapter represent perhaps the best example of Web services achieved through the Service Oriented Architecture of components. Through hardware virtualization on Xen hypervisors, Amazon. com has made it possible to create private virtual servers that you can run worldwide. These servers can be provisioned with almost any kind of application software you might envisage, and they tap into a range of support services that not only make distributed cloud computing applications possible, but make them robust. Some very large Web sites are running on Amazon. com's infrastructure without their client audience being any the wiser.

Amazon Web Services is based on SOA standards, including HTTP, REST, and SOAP transfer protocols, open source and commercial operating systems, application servers, and browser-based access. Virtual private servers can provision virtual private clouds connected through virtual private networks providing for reasonable security and control by the system administrator.

AWS has a great value proposition: You pay for what you use. While you may not save a great deal of money over time using AWS for enterprise class Web applications, you encounter very little barrier to entry in terms of getting your site or application up and running quickly and robustly. AWS has much to teach us about the future of cloud computing and how virtual infrastructure can be best leveraged as a business asset.

Understanding Amazon Web Services

The Amazon is the world's largest river. Amazon.com is the world's largest online retailer with net sales in $24.51 billion, according to their 2009 annual report. The company is a long way past selling books and records. While Amazon.com is not the earth's biggest retailer (that spot is reserved for Wal-Mart), Amazon.com offers the largest number of retail product SKUs through a large ecosystem of partnerships. By any measure, Amazon.com is a huge business. To support this business, Amazon.com has built an enormous network of IT systems to support not only average, but peak customer demands. Amazon Web Services (AWS) takes what is essentially unused infrastructure capacity on Amazon.com's network and turns it into a very profitable business. Figure 9.1 shows the Amazon Web Services home page (http://aws.amazon.com/).

AWS is having enormous impact in cloud computing. Indeed, Amazon.com's services represent the largest pure Infrastructure as a Service (IAAS) play in the marketplace today. It is also one of the best examples of what is possible using a Service Oriented Architecture (SOA), which is described in Chapter 13. The structure of Amazon.com's Amazon Web Services (AWS) is therefore highly educational in understanding just how disruptive cloud computing can be to traditional fixed assetIT deployments, how virtualization enables a flexible approach to system rightsizing, and how dispersed systems can impart reliability to mission critical systems.

FIGURE 9.1

Amazon Web Services home page

For these reasons, even though Amazon.com'sIaaS services are described in other chapters individually, this chapter provides background to the entire portfolio and shows why Amazon Web Services is a $500 million business that hosts eight of the top ten Facebook games (http://gigaom.com/2010/08/02/amazon-web-services-revenues/; and http:// venturebeat.com/2010/08/03/amazon-web-services-generating-an-estimated-500m-in-revenue-thanks-in-part-to-growth-of-social-games/). In 2008 AWS claimed 330,000 unique accounts, although the press release for that claim has now disappeared.

Amazon Web Services represents only a small fraction of Amazon's overall business sales at the moment, but it is a rapidly growing component. Amazon doesn't break down its sales by individual areas in its annual report, but according to Randy Bias who blogs on the site Cloudscaling.com (http://cloudscaling.com/blog/cloud-computing/amazons-ec2-generating-220m-annually) the largest component of Amazon's offerings is Amazon's Elastic Compute Cloud (EC2), which generates in excess of $220 million annually as of October 2009. EC2 is estimated to run on over 40,000+ servers worldwide divided into six availability zones. (You learn about EC2 later in this chapter.) EC2 is an Infrastructure as a Service (IaaS) play, a market that

was pegged to be around $400-$600 M/year and growing 10%-20%/year even in the face of a dramatic market slowdown. Rackspace Cloud (http://www.rackspacecloud.com/), EC2's nearest competitor, is pegged to be around 10% the size of EC2 by Bias.

Amazon Web Service Components and Services

Amazon Web Services is comprised of the following components, listed roughly in their order of importance:

- **Amazon Elastic Compute Cloud** (EC2; http://aws.amazon.com/ec2/), is the central application in the AWS portfolio. It enables the creation, use, and management of virtual private servers running the Linux or Windows operating system over a Xen hypervisor. Amazon Machine Instances are sized at various levels and rented on a computing/ hour basis. Spread over data centers worldwide, EC2 applications may be created that are highly scalable, redundant, and fault tolerant. EC2 is described more fully the next section. A number of tools are used to support EC2 services:
- **Amazon Simple Queue Service** (SQS; http://aws.amazon.com/sqs/) is a message queue or transaction system for distributed Internet-based applications. See ‒Examining the Simple Queue Service (SQS)‖ later in this chapter for a description of this AWS feature. In a loosely coupled SOA system, a transaction manager is required to ensure that messages are not lost when a component isn't available.
- **Amazon Simple Notification Service** (SNS; http://aws.amazon.com/sns/) is a Web service that can publish messages from an application and deliver them to other applications or to subscribers. SNS provides a method for triggering actions, allowing clients or applications to subscribe to information (like RSS), or polling for new or changed information or perform updates.

EC2 can be monitored by **Amazon CloudWatch** (http://aws.amazon.com/ cloudwatch/), which provides a console or command line view of resource utilization, site Key Performance Indexes (performance metrics), and operational indicators for factors such as processor demand, disk utilization, and network I/O. The metrics obtained by CloudWatch may be used to enable a feature called **Auto Scaling** (http://aws.amazon.com/autoscaling/) that can automatically scale an EC2 site based on a set of rules that you create. Autoscaling is part of Amazon Cloudwatch and available at no additional charge.

Amazon Machine Instances (AMIs) in EC2 can be load balanced using the **Elastic Load Balancing** (http://aws.amazon.com/elasticloadbalancing/) feature. The Load Balancing feature can detect when an instance is failing and reroute traffic to a healthy instance, even an instance in other AWS zones. The Amazon CloudWatch

metrics request count and request latency that show up in the AWS console are used to support Elastic Load Balancing.

- **Amazon Simple Storage System** (S3; http://aws.amazon.com/s3/**)** is an online backup and storage system, which is described in ‒Working with Amazon Simple Storage System (S3)‖.

A high speed data transfer feature called AWS Import/Export (http://aws.amazon. com/importexport/) can transfer data to and from AWS using Amazon's own internal network to portable storage devices.

- **Amazon Elastic Block Store** (EBS; http://aws.amazon.com/ebs/**)** is a system for creating virtual disks (volume) or block level storage devices that can be used for Amazon Machine Instances in EC2.
- **Amazon SimpleDB**(http://aws.amazon.com/simpledb/) is a structured data store that supports indexing and data queries to both EC2 and S3. SimpleDB isn't a full database implementation, as you learn in ‒Exploring SimpleDB (S3)‖ later in this chapter; it stores data in ‒buckets‖ and without requiring the creation of a database schema. This design allows SimpleDB to scale easily. SimpleDB interoperates with both Amazon EC2 and Amazon S3.
- **Amazon Relational Database Service** (RDS; http://aws.amazon.com/rds/) allows you to create instances of the MySQL database to support your Web sites and the many applications that rely on data-driven services. MySQL is the ‒M‖ in the ubiquitous LAMP Web services platform (for Linux, APACHE, MySQL, and PERL), and the inclusion of this service allows developers to port applications, their source code, and databases directly over to AWS, preserving their previous investment in these technologies. RDS provides features such as automated software patching, database backups, and automated database scaling via an API call.
- **Amazon Cloudfront**(http://aws.amazon.com/cloudfront/) is an edge-storage or content-delivery system that caches data in different physical locations so that user access to data is enhanced through faster data transfer speeds and lower latency. Cloudfront is similar to systems such as Akamai.com, but is proprietary to Amazon.com and is set up to work with Amazon Simple Storage System (Amazon S3). Cloudfront is currently in beta, but has been well received in the trade press. See ‒Defining Cloudfront‖ later in this chapter for more details.

While the list above represents the most important of the AWS offerings, it is only a partial list—a list that is continually growing and very dynamic. A number of services and utilities support Amazon partners or the AWS infrastructure itself. These are the ones you may encounter:

- **Alexa Web Information Service** (http://aws.amazon.com/awis/) and **Alexa Top Sites** (http://aws.amazon.com/alexatopsites/) are two services that collect and expose information about the structure and traffic patterns of Web sites.

This information can be used to build or structure Web sites, access relatedsites, analyze historical patterns for growth and relationships, and perform data analysis on site information. Alexa Top Sites can rank sites based on their usage and be used to structure awareness of site popularity into the structure of Web service you build.

- **Amazon Associates Web Services** (A2S) is the machinery for interacting with Amazon's vast product data and eCommerce catalog function. This service, which was called Amazon E-Commerce Service (ECS), is the means for vendors to add their products to the Amazon.com site and take orders and payments.
- **Amazon DevPay** (http://aws.amazon.com/devpay/) is a billing and account management service that can be used by businesses that run applications on top of AWS. DevPay provides a developer API that eliminates the need for application developers to build order pipelines, because Amazon does the billing based on your prices and then uses Amazon Payments to collect the payments.
- **Amazon Elastic MapReduce** (http://aws.amazon.com/elasticmapreduce/) is an interactive data analysis tool for performing indexing, data mining, file analysis, log file analysis, machine learning, financial analysis, and scientific and bioinformatics research. Elastic MapReduce is built on top of a Hadoop framework using the Elastic Compute Cloud (EC2) and Simple Storage Service (S3).
- **Amazon Mechanical Turk** (http://aws.amazon.com/mturk/) is a means for accessing human researchers or consultants to help solve problems on a contractual or temporary basis. Problems solved by this human workforce have included object identification, video or audio recording, data duplication, and data research. Amazon.com calls this type of work Human Intelligence Tasks (HITs). The Mechanical Turk is currently in beta.
- **AWSMulti-FactorAuthentication**(AWSMFA; http://aws.amazon.com/mfa/**)** is a special feature that uses an authentication device you have in your possession to provide access to your AWS account settings. This hardware key generates a pseudo-random sixdigit number when you press a button that you enter into your logon. This gives you two layers of protection: your user id and password (things you know) and the code from your hardware key (something you have). This multifactor security feature can be extended to Cloudfront and Amazon S3. The Enzio Time Token from Gemalto (http://online noram.gemalto.com/) is available for use with Amazon Web Service; the key costs $12.99. Secure access to your EC2 AMIs is controlled by passwords, Kerberos, and 509 Certificates.
- **Amazon Flexible Payments Service** (FPS; http://aws.amazon.com/fps/) is a payments-transfer infrastructure that provides access for developers to charge

Amazon's customers for their purchases. Using FPS, goods, services, donations, money transfers, and recurring payments can be fulfilled. FPS is exposed as an API that sorts transactions into packages called Quick Starts that make this service easy to implement.

- **Amazon Fulfillment Web Services** (FWS; http://aws.amazon.com/fws/) allows merchants to fill orders through Amazon.com fulfillment service, with Amazon handling the physical delivery of items on the merchant's behalf. Merchant inventory is prepositioned in Amazon's fulfillment centers, and Amazon packs and ships the items. There is no charge for using Amazon FWS; fees for the Fulfillment by Amazon (FBA; http://www.amazon.com/gp/seller/fba/fulfillment-by-amazon.html) service apply. Between FBA and FWS, you can create a nearly virtual store on Amazon.com.
- **Amazon Virtual Private Cloud** (VPC; http://aws.amazon.com/vpc/) provides a bridge between a company's existing network and the AWS cloud. VPC connects your network resources to a set of AWS systems over a Virtual Private Network (VPN) connection and extends security systems, firewalls, and management systems to include their provisioned AWS servers. Amazon VPC is integrated with Amazon EC2, but Amazon plans to extend the capabilities of VPC to integrate with other systems in the Amazon cloud computing portfolio.
- **AWS Premium Support** (http://aws.amazon.com/premiumsupport/) is amazon's technical support and consulting business. Through AWS Premium Support, subscribers to AWS can get help building or supporting applications that use EC2, S3, Cloudfront, VPC, SQS, SNS, SimpleDB, RDS, and the other services listed above. Service plans are available on a per-incidence, monthly, or unlimited basis at different levels of service.

With this overview of AWS components complete, let's look at the central part of Amazon Web Service's value proposition, the creation and deployment of virtual private servers using the Elastic Compute Cloud (EC2) service.

Working with the Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (EC2) is a virtual server platform that allows users to create and run virtual machines on Amazon's server farm. With EC2, you can launch and run server instances called Amazon Machine Images (AMIs) running different operating systems such as Red Hat Linux and Windows on servers that have different performance profiles. You can add or subtract virtual servers elastically as needed; cluster, replicate, and load balance servers; and locate your different servers in different data centers or ‒zones‖ throughout the world to provide fault tolerance. The term *elastic* refers to the ability to size your capacity quickly as needed.

The difference between an instance and a machine image is that an instance is the emulation of a hardware platform such as X86, IA64, and so on running on the Xen hypervisor. A machine image is the software and operating system running on top of the instance. A machine image may be thought of as the contents of a boot drive, something that you could package up with a program such as Ghost, Acronis, or TrueImage to create a single file containing the exact contents of a volume. A machine image should be composed of a hardened operating system with as few features and capabilities as possible and locked down as much as possible.

Consider a situation where you want to create an Internet platform that provides the following:

- A high transaction level for a Web application
- A system that optimizes performance between servers in our system
- Data driver information services
- Network security
- The ability to grow your service on demand

Implementing that type of service might require a rack of components that included the following:

- An application server with access to a large RAM allocation
- A load balancer, usually in the form of a hardware appliance such as F5's BIG-IP
- A database server
- Firewalls and network switches
- Additional rack capacity at the ISP

A physical implementation of these components might cost you something in the neighborhood of $25,000 depending upon the scale of your application. With AWS, you might be able to have an equivalent service for as little as $1,000 and have a high level of availability and reliability to boot. This difference may surprise you, but it is understandable when you consider that AWS can run its services with a much greater efficiency than your company would alone and therefore amortize its investment in hardware over several customers. That is the promise and the potential of cloud computing realized and why large Web sites such as Recovery.gov have moved to AWS.

Amazon Machine Images

AMIs are operating systems running on the Xen virtualization hypervisor. Each virtual private server is accorded a size rating called its *EC2 Compute Unit,* which is pegged to the equivalent of a 1.0–1.2 GHz 2007 Opteron or 2007 Xeon processor.

Table 9.1 shows the current set of Instance types, which broadly fall into the following three classes:

1. **Standard Instances:** The standard instances are deemed to be suitable for standard server applications.
2. **High Memory Instances:** High memory instances are useful for large data throughput applications such as SQL Server databases and data caching and retrieval.
3. **High CPU Instances:** The high CPU instance category is best used for applications that are processor- or compute-intensive. Applications of this type include rendering, encoding, data analysis, and others.

TABLE 9.1

Amazon Machine Image Instance Types

Type	Compute Engine	RAM (GB)	Storage (GB)1	Platform	I/O Performance	API Name
Micro instance	Up to 2 EC2 Compute Units (1 virtual core) in short bursts	0.613	EBS (Elastic Block Storage) storage only	32-bit or 64-bit	Low	T1.micro
Standard instance – small (default)	1 EC2 Compute Unit (1 virtual core)	1.7	160	32-bit	Moderate	m1.small
Standard instance – large	4 EC2 Compute Units (2 virtual cores X 2 EC2 Units)	7.5	850	64-bit	High	m1.large
Standard instance – extra large	8 EC2 Compute Units (4 virtual cores X 2 EC2 Units)	15	1,690	64-bit	High	m1.xlarge
High Memory Double Extra Large Instance	13 EC2 Compute Units (4 virtual cores X 3.25 EC2 Units)	34.2	850	64-bit	High	m2.2xlarge

Type	Compute Engine	RAM (GB)	Storage (GB)1	Platform	I/O Performance	API Name
High Memory Quadruple Extra Large Instance	26 EC2 Compute Units (8 virtual cores X 3.25 EC2 Units)	68.4	1,690	64-bit	High	m2.4xlarge
High CPU Medium Instance	5 EC2 Compute Units (2 virtual cores X 2.5 EC2 Units)	1.7	350	32-bit	Moderate	c1.medium
High CPU Extra Large Instance	20 EC2 Compute Units (8 virtual cores X 2.5 EC2 Units)	7	1,690	64-bit	High	c1.xlarge

1. Storage is not persistent. All assigned storage is lost upon rebooting. To store data on AWS, you need to create a Simple Storage Service (S3) bucket or an Elastic Block Storage (EBS) volume.

Pricing models

The pricing of these different AMI types depends on the operating system used, which data center the AMI is located in (you can select its location), and the amount of time that the AMI runs. Rates are quoted based on an hourly rate. Additional charges are applied for:

- the amount of data transferred
- whether Elastic IP Addresses are assigned
- your virtual private server's use of Amazon Elastic Block Storage (EBS)
- whether you use Elastic Load Balancing for two or more servers
- other features

AMIs that have been saved and shut down incur a small one-time fee, but do not incur additional hourly fees. The three different pricing models for EC2 AMIs are as follows:

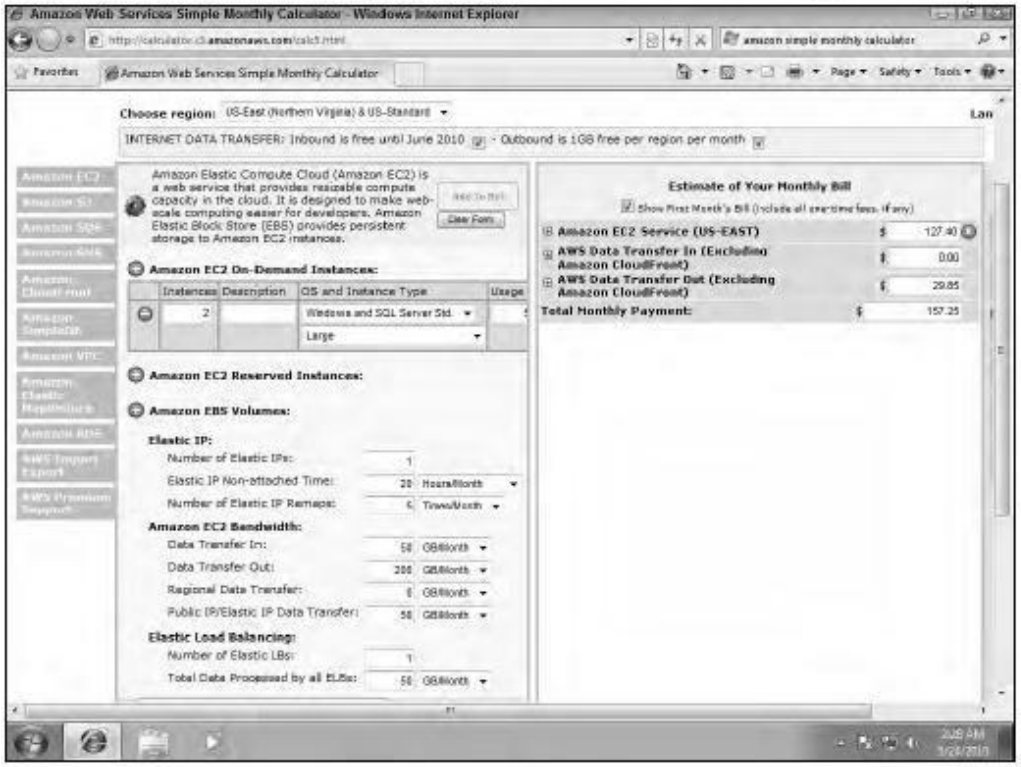
- **On-Demand Instance:** This is the hourly rate with no long-term commitment.
- **Reserved Instances:** This is a purchase of a contract for each instance you use with a significantly lower hourly usage charge after you have paid for the reservation.
- **Spot Instance:** This is a method for bidding on unused EC2 capacity based on the current spot price. This feature offers a significantly lower price, but it varies over time or may not be available when there is no excess capacity.

Pricing varies by zone, instance, and pricing model. A chart of the different current prices may be found at http://aws.amazon.com/ec2/. This page also includes current

Amazon Elastic Block Store volume and snapshot charges to Amazon S3, as well as data transfer rates. Figure 9.2 shows the AWS Simple Monthly Calculator that you can find at http://calculator.s3. amazonaws.com/calc5.html to help you estimate your monthly charges.

FIGURE 9.2

The Amazon Web Services Simple Monthly Calculator for determining system costs on AWS



System images and software

You can choose to use a template AMI system image with the operating system of your choice or create your own system image that contains your custom applications, code libraries, settings, and data. Security can be set through passwords, Kerberos tickets, or certificates.

These operating systems are offered:

- RedHat Enterprise Linux
- OpenSuse Linux
- Ubuntu Linux
- Sun OpenSolaris
- Fedora
- Gentoo Linux

- Oracle Enterprise Linux
- Windows Server 2003/2008 32-bit and 64-bit up to Data Center Edition
- Debian

Most of the system image templates that Amazon AWS offers are based on Red Hat Linux, Windows Server, Oracle Enterprise Linux, and OpenSolarisfrom the list above. Table 9.2 lists some of the more common enterprise applications that are available from AWS either as part of its canned templates or for use in building your own AMI system image. Hundreds of free and paid AMIs can be found on AWS.

TABLE 9.2

EC2 Enterprise Software Types

Application Type	Software
Application Development Environments	IBM sMash, JBoss Enterprise Application Platform, and Ruby on Rails
Application Servers	IBM WebSphere Application Server, Java Application Server, and Oracle WebLogic Server
Batch Processing	Condor, Hadoop, and Open MPI
Databases	IBM DB2, IBM Informix Dynamic Server, Microsoft SQL Server Standard 2005, MySQL Enterprise, and Oracle Database 11g
Video Encoding and Streaming	Windows Media Server and Wowza Media Server Pro
Web Hosting	Apache HTTP, IIS/ASP.Net, IBM Lotus Web Content Management, and IBM WebSphere Portal Server

When you create a virtual private server, you can use the Elastic IP Address feature to create what amounts to a static IP v4 address to your server. This address can be mapped to any of your AMIs and is associated with your AWS account. You retain this IP address until you specifically release it from your AWS account. Should a machine instance fail, you can map your Elastic IP Address to fail over to a different AMI. You don't need to wait until a DNS server updates the IP record assignment, and you can use a form to configure the reverse DNS record of the Elastic IP address change.

There are currently four different EC2 service zones or regions:

- US East (Northern Virginia)
- US West (Northern California)
- EU (Ireland)
- Asia Pacific (Singapore)

Creating an account and instance on EC2

The process for signing up for Amazon Web Services, creating an Amazon Machine Instance, and provisioning the image with software is relatively straightforward. You begin the process by clicking the Sign Up Now button on the Amazon Web Services home page (http://aws.amazon. com) shown in Figure 9.4. You see a page on which you name your account, provide a password, and select a payment option. If you have an Amazon.com user account, you can opt to use that account for your AWS account. After you create your account, you want to sign into the Amazon EC2 Management Console, where you see a dashboard similar to the one shown in Figure 9.3.

To create an AMI instance, do the following:

1. Click the Launch Instance button in the Getting Started Section to launch the Request Instances wizard shown in Figure 9.4.
2. Scroll the list to find the type of system image you want, and click Select.
3. Specify the Number of Instance(s) desired, the Availability Zone where the instance(s) should be located, the Instance type in the Instance Details step shown in Figure 9.5, and click Continue.
4. In the Advanced Instance Options step shown in Figure 9.6, enter an ID for the Kernel or RAM and enable CloudWatch, if desired; then click Continue.

FIGURE 9.3

The AWS EC2 Management Console with no instances

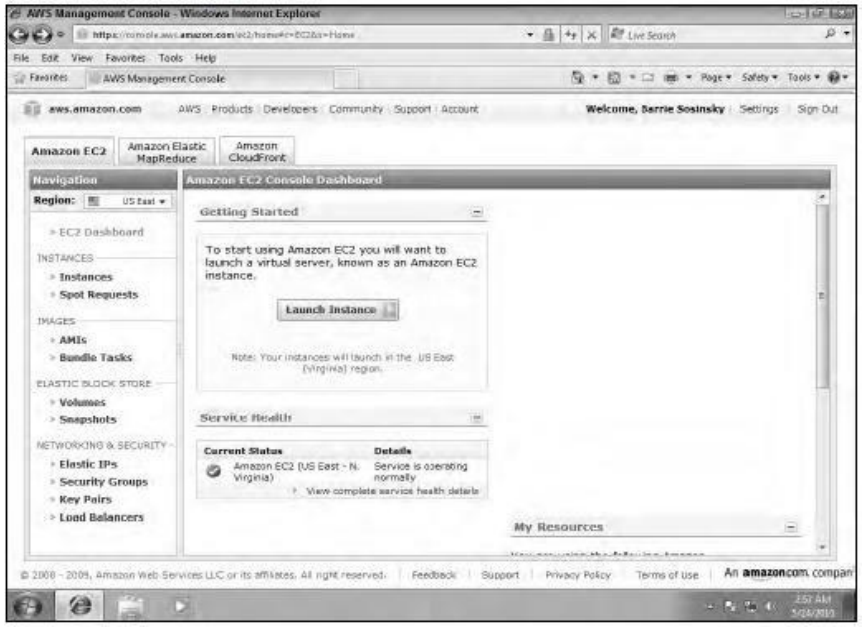


FIGURE 9.4

Select an Instance type from one of the templates shown, or create your own AMI in this step.

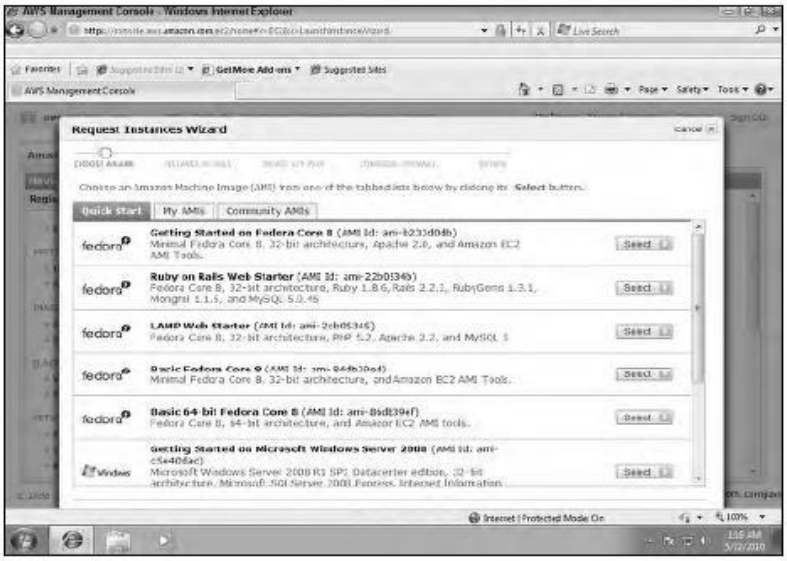
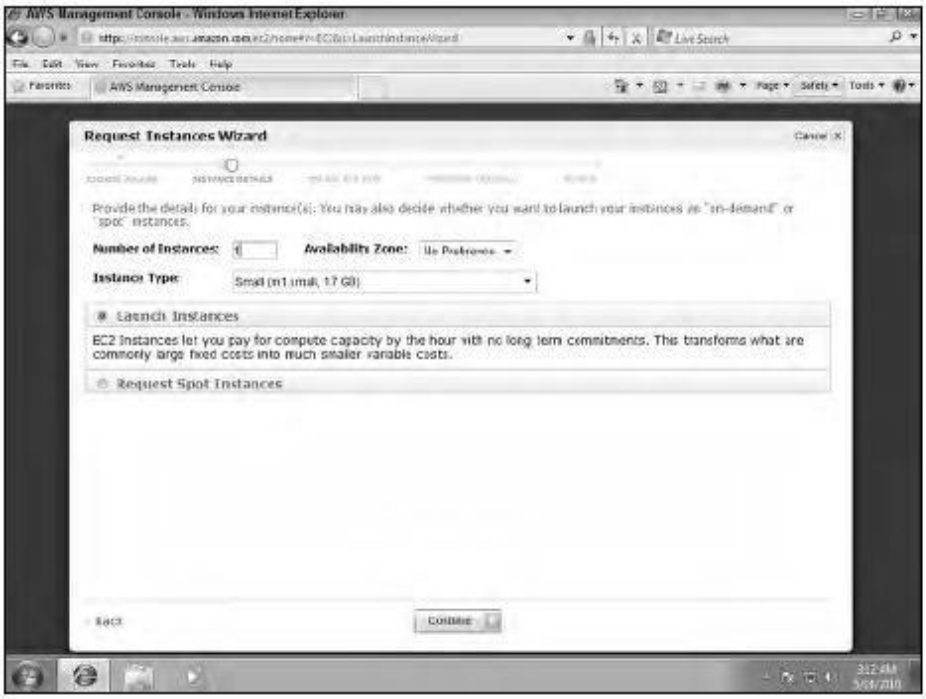


FIGURE 9.5

Fill in the instance details in this step.



5. In the Create a Key Pair step, you are asked to either create a new key pair or apply a key pair that you have already created. Make a selection as shown in Figure 9.7, and click Continue.

Creating a key pair generates a public/private key that you download from AWS. When you want to provide someone access to your secured server, you supply them with the private key they need to connect to the server.

6. The Configure Firewall page as shown in figure 9.8 allows you to set the applications that have access to your server, their transport protocols, ports, and the security group that can access your server. You can create a new security group or apply one that already exists.

FIGURE 9.6

In the Advanced Instance Options step, you can provide an identifier for your instance's kernel and RAM disk and enable the CloudWatch monitoring feature.

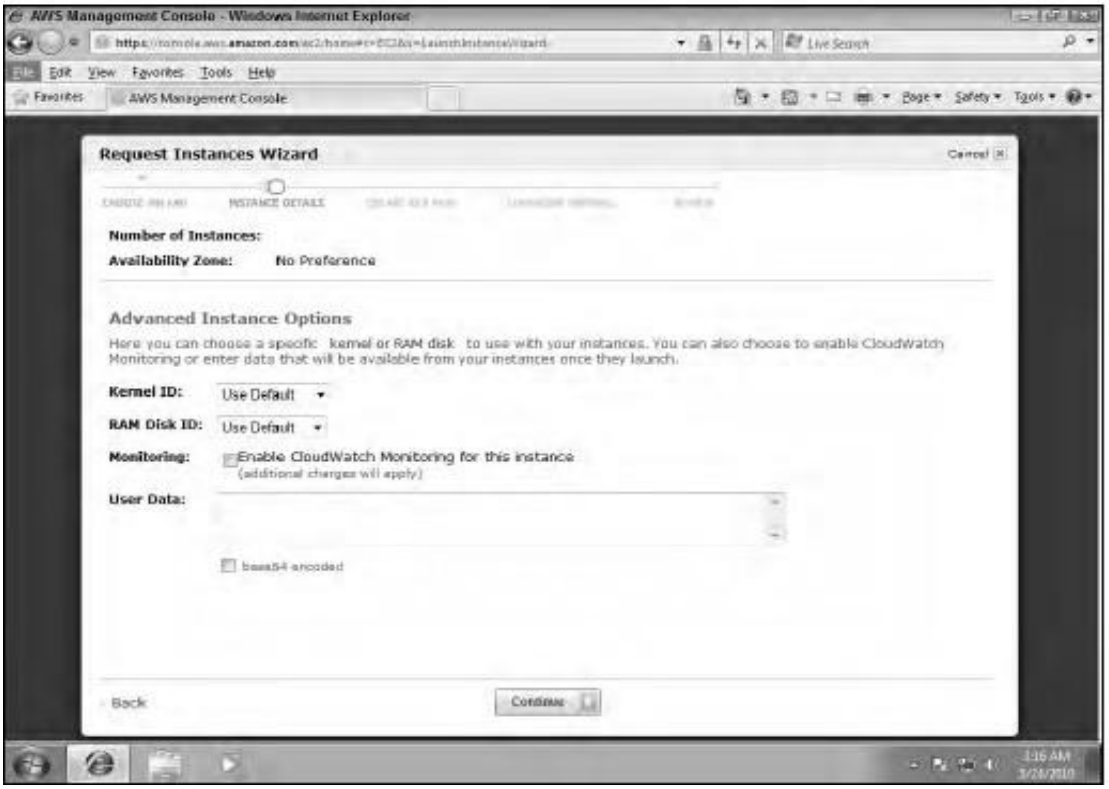


FIGURE 9.7

For secure access to your AMI, you can assign a public/private key pair.

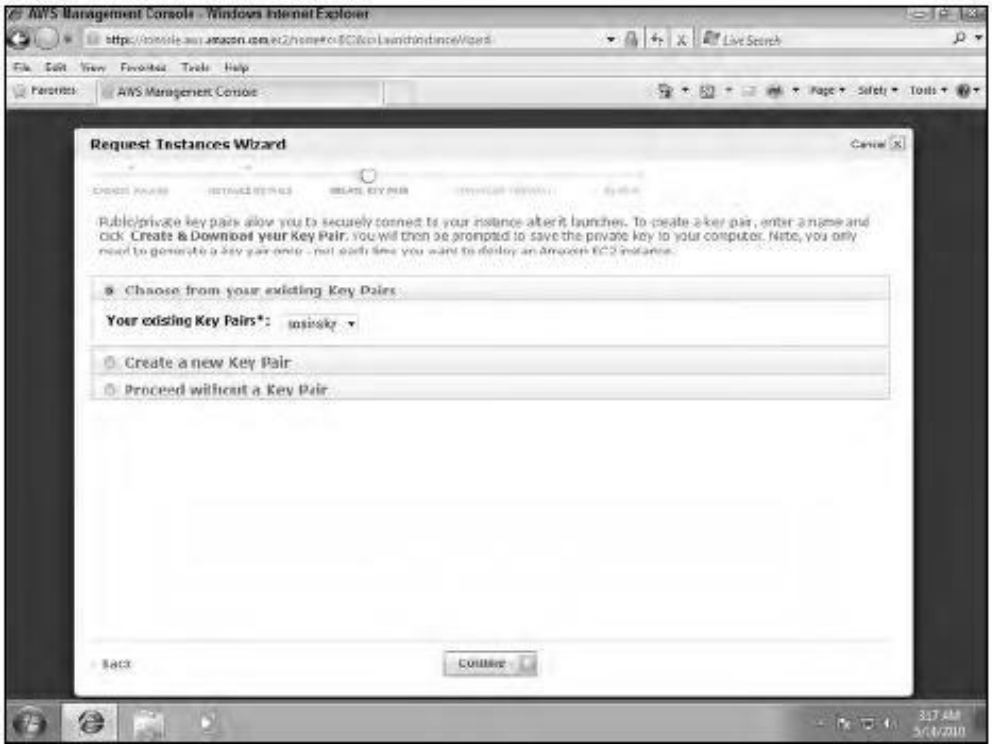
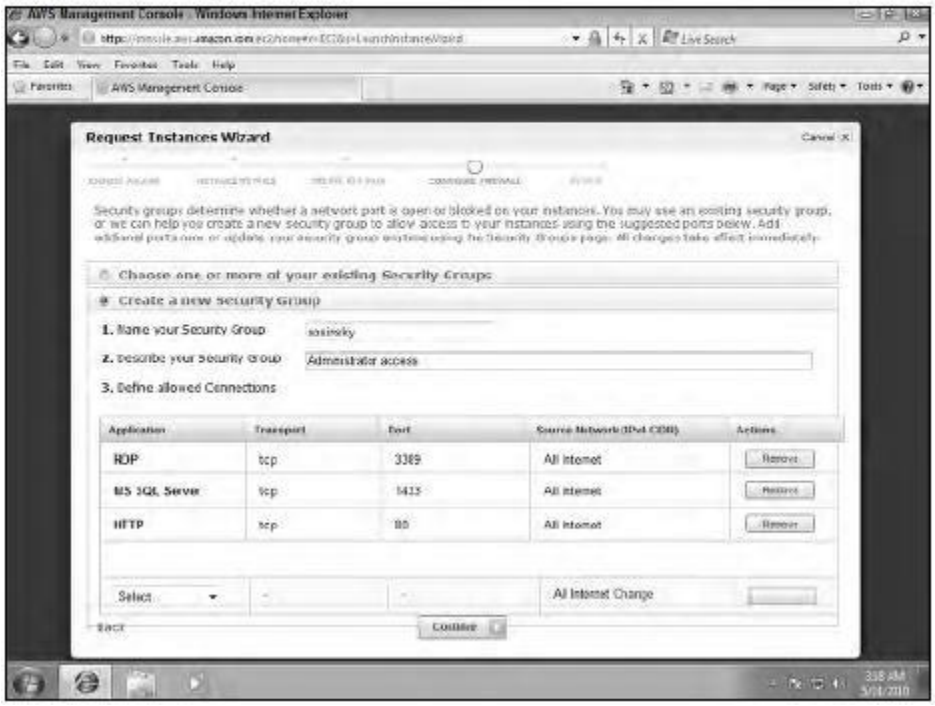


FIGURE 9.8

Firewall settings allow you to filter by service and protocol, as well as set a security group membership for access.

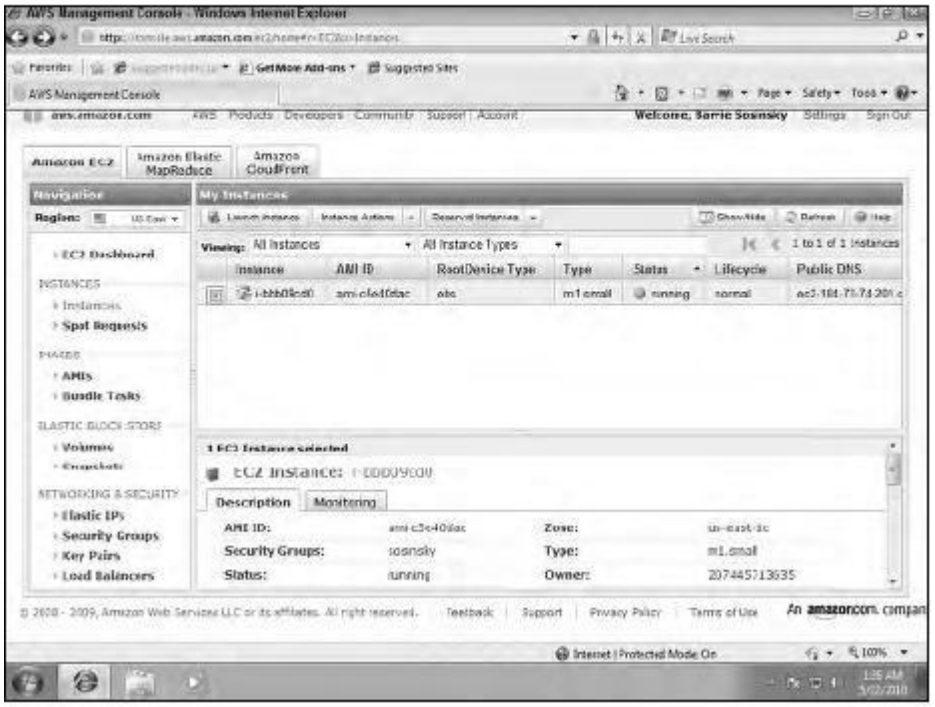


7. Click Continue to go the Review summary page shown in Figure 9.9; review the settings on that page and then click Continue again.

You are returned to the AWS Management Console, and your AMI is shown in the My Instances pane. The instance is created, and after a moment starts running, as shown in Figure 9.10.

FIGURE 9.10

The AWS Management Console with an active AMI showing



8. After the instance is running, you need to connect to the instance. Use the Connect command on the Instance context menu, as shown in Figure 9.11. That menu also allows you to suspend, reboot, terminate (deleting or killing), clone, snapshot, set passwords, and perform other actions that are specific to the type of system image you created.

FIGURE 9.9

This Review page allows you to see the type of Amazon Machine Instance and the system image it will run before you create it.

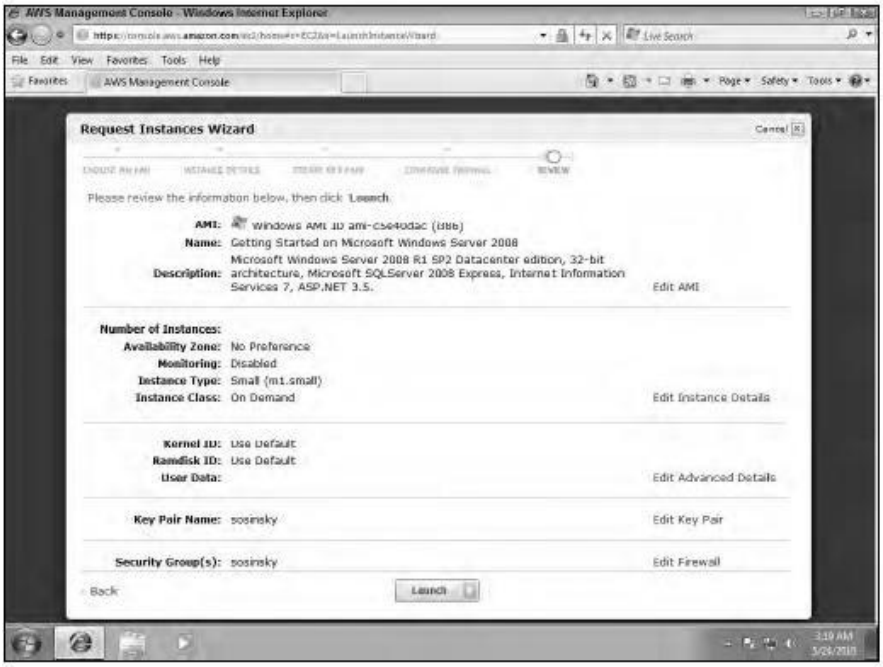


FIGURE 9.10

The AWS Management Console with an active AMI showing

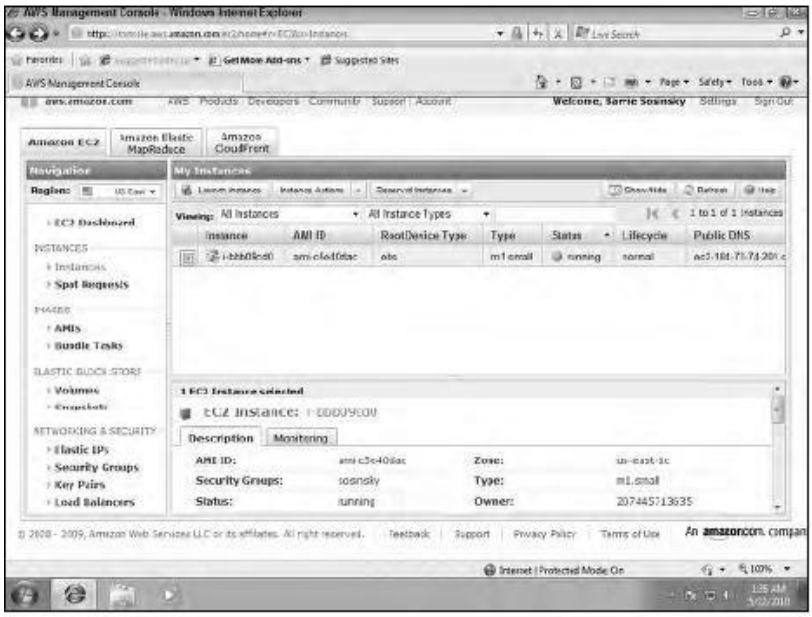
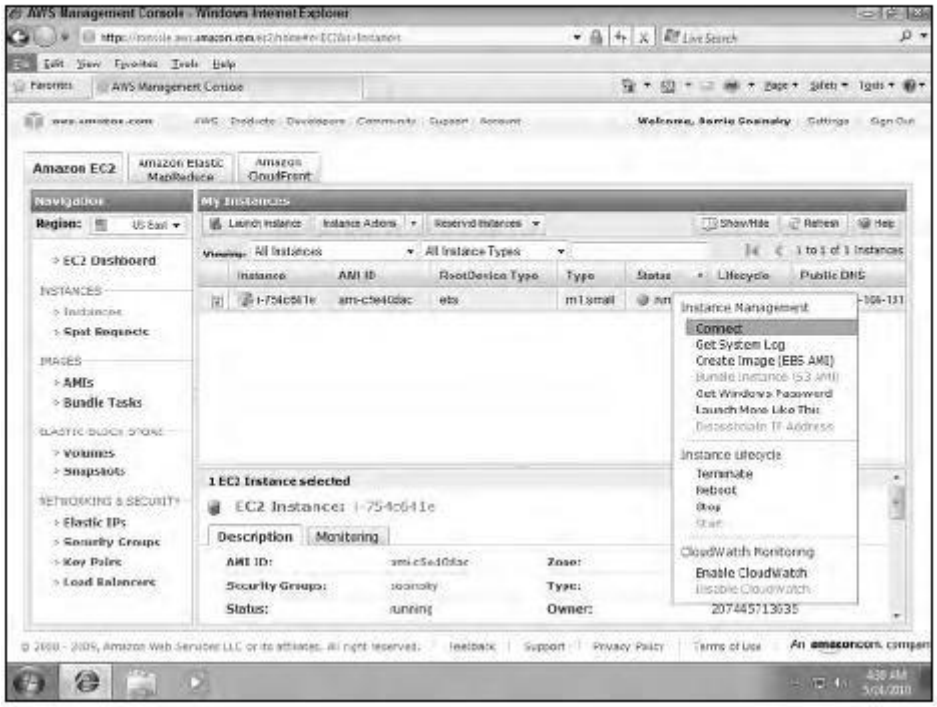


FIGURE 9.11

Context menu for a Windows system image running in an AMI



This Review page allows you to see the type of Amazon Machine Instance and the system When you are finished creating an AMI instance, you can create a shortcut to directly connect with that instance. With Windows Server, that shortcut is a Microsoft Remote Desktop Connection (or Terminal Server Client), which connects to your server using the Remote Desktop Protocol. Other operating systems create different types of Virtual Private Network connections.

Working with Amazon Storage Systems

When you create an Amazon Machine Instance you provision it with a certain amount of storage. That storage is temporal, it only exists for as long as your instance is running. All of the data contained in that storage is lost when the instance is suspended or terminated, as the storage is reassigned to the pool for other AWS users to use. For this and other reasons you need to have access to persistent storage. The Amazon Simple Storage System provides block storage, but is set up in a way that is somewhat unique from other storage systems you may have worked with in the past.

Amazon Simple Storage System (S3)

Amazon S3's cloud-based storage system allows you to store data objects ranging in size from 1 byte up to 5GB in a flat namespace. In S3, storage containers are referred to as buckets, and buckets serve the function of a directory, although there is no object hierarchy to a bucket, and you save objects and not files to it. It is important that you do not associate the concept of a filesystem with S3, because files are not supported; only objects are stored. Additionally, you do not ―mount‖ a bucket as you do a filesystem.

The S3 system allows you to assign a name to a bucket, but that name must be unique in the S3 namespace across all AWS customers. Access to an S3 bucket is through the S3 Web API (either with SOAP or REST) and is slow relative to a real-world disk storage system. S3's performance limits its use to non-operational functions such as data archiving and retrieval or disk backup. The REST API is preferred to the SOAP API, because it is easier to work with large binary objects with REST.

You can do the following with S3 buckets through the APIs:

- Create, edit, or delete existing buckets
- Upload new objects to a bucket and download them
- Search for and find objects and buckets
- Find metadata associate with objects and buckets
- Specify where a bucket should be stored
- Make buckets and objects available for public access

One tool commonly used to manage data for Amazon S3 is the s3cmd command line client (http://s3tools.org/s3cmd).

The S3 service is used by many people as the third level backup component in a 3-2-1 backup strategy. That is, you have your original data (1), a copy of your data (2), and an off-site copy of your data (3); the latter of these may be S3. In this regard, S3 acts as a direct competitor to Carbonite's backup system. One of the options available to you is versioning for Amazon S3. With versioning, every version of an object stored in an S3 bucket is retained, provided you enable the versioning feature. Any HTTP or REST operation such as PUT, POST, COPY, or DELETE creates a new object that is stored along with the older version. A GET operation retrieves the newest version of the object, but the ability to recover and undo actions is available. Versioning also can be used for preserving data and for archiving purposes.

Amazon S3 provides large quantities of reliable storage that is highly protected but to which you have low bandwidth access. S3 excels in applications where storage is archival in nature. For example, you find S3 in use by large photo sharing sites. In the next section you'll see Amazon's Elastic Block Storage or EBS. In EBS you create virtual drives that you can use with your machine instances in the same way that you would use a hard drive with a physical system. EBS tends to be used in transactional systems where high-speed data access is required.

Amazon Elastic Block Store (EBS)

The third of Amazon's data storage systems are devoted to Amazon Elastic Block Storage (EBS), which is a persistent storage service with a high operational performance. Advantages of EBS are that it can store file system information and its performance is higher and much more reliable than Amazon S3. That makes EBS valuable as an operational data storage medium for AWS. The cost of creating an EBS volume is also greater than creating a similarly sized S3 bucket.

An EBS volume can be used as an instance boot partition. The advantages of an EBS boot partition are that you can have a volume up to 1TB, retain your boot partition separately from your EC2 instance, and use a boot partition volume as a means for bundling an AMI into a single package. EBS boot partitions can be stopped and started, and they offer fast AMI boot times.

EBS is similar in concept to a Storage Area Network or SAN; you create block storage volumes varying in size from 1GB to 1TB and make those volumes available to your machine instances. The performance of a volume is dependent upon the network I/O and therefore varies as a function of the size of your instance (see Table 9.3), as well as the type of disk I/O operations (random, sequential, request size, and READS or WRITE) that are in progress.

When you create volumes, they appear first as raw block storage devices that must be formatted for use. A volume is mounted on a particular instance and is available to that instance alone; that is, volumes may not be shared between instances. Volumes may be located in the same zone as the AMI to which they are attached. Volumes appear as if they are devices (physical drives) when attached to an instance. You can mount multiple volumes on a single instance, if desired, and create striped RAID volumes for faster performance. The filesystem for mounted volumes appears when you open the volume, and you can install applications or copy data to mounted volumes as you would any physical disk.

EBS supports volume replication within the same availability zone, which can add an extra level of fault tolerance to the data set. The use of replication means that mirroring a volume won't add much additional fault tolerance. Snapshots are the recommended approach to improving your volume's reliability.

You can make an instance image or snapshot of your AMI, and these point-in-time snapshots are then copied out to Amazon S3. You can use these snapshots as system images to create new AMIs or to restore a volume (and instance) to that point-in-time snapshot when needed. You can share snapshots with other authorized users by using a volume's context menu in the AWS Management Console and selecting the Snapshot Permissions command.

When you create a new volume from an S3 snapshot, the data is slowly copied to the new volume. As you start working on the new volume, any missing data is downloaded preferentially as needed.

Each snapshot you take adds incremental changes to the previous snapshot, which means that while the first snapshot takes a fair amount of time, subsequent snapshots are usually executed quickly and with only a modest amount of extra storage space required.

EBS is a service priced on the amount of storage space used, how long you use it, and the number of I/O requests made to the volume. You can use a utility like IOSTAT to measure I/O of your systems to estimate these transaction costs, which vary greatly by operating system and application. Amazon quotes an example of a medium-sized database of 100GB with 100 I/O per sec costing about $10 per month for the allocated storage and $26 per month for the I/O (there are 2.6 million seconds in a month). Snapshots are priced on the storage blocks used, not on the size of the volume being stored. Amazon also charges for the amount of data transferred to Amazon S3 during a snapshot.

Table 9.3 summarizes the various properties of the three different forms of EC2 data storage devices.

TABLE 9.3

EC2 Storage Type Properties

Property	AMI Instance	Amazon Simple Storage Service (S3)	Amazon Elastic Block Storage (EBS)	Amazon CloudFront
Adaptability	Medium	Low	High	Medium
Best usage	Transient data storage	Persistent or archival storage	Operational data storage	Data sharing and large data object streaming

Property	AMI Instance	Amazon Simple Storage Service (S3)	Amazon Elastic Block Storage (EBS)	Amazon CloudFront
Cost	Low	Medium	High	Low
Ease of use	Low	High	High	High
Data protection	Very Low	Very High	High	Low
Latency	Medium	Low	High	High
Least best used as	Persistent storage	Operational storage	For small I/O transfers	Operational data
Reliability	High	Medium	High	Medium
Throughput	Variable	Slow	High	High

CloudFront

Amazon CloudFront is referred to as a content delivery network (CDN), and sometimes called *edge computing*. In edge computing, content is pushed out geographically so the data is more readily available to network clients and has a lower latency when requested. You enable CloudFront through a selection in the AWS Management Console.

You can think of a CDN as a distributed caching system. CloudFront servers are located throughout the world—in Europe, Asia, and the United States. As such, CloudFront represents yet another level of Amazon cloud storage. A user requesting data from a CloudFront site is referred to the nearest geographical location. CloudFront supports ‑geo-caching‖ data by performing static data transfers and streaming content from one CloudFront location to another.

At the time this chapter was written CloudFront was in beta, but it has been well received. Direct competitors for CloudFront include Akamai Technologies (http://www.akamai.com/), Edgecast Networks (http://www.edgecast.com/), and Limelight Networks (http://www. limelightnetworks.com/). CloudFront's aggressive pricing model is expected to put pressure on these other services over time. Pricing for

CloudFront is based on how much data is transferred to clients, and it doesn't require a service contract. You can estimate CloudFront's costs using the AWS Simple Monthly Calculator (refer to Figure 9.3); costs vary by region.

When you create a CloudFront implementation, a CloudFront domain name is registered for your domain name in the form <domainname>.cloudfront.net, and objects in the CloudFront domain can be mapped to your own domain. You store your source files on CloudNet servers in Amazon S3 buckets and then use the CloudFront API to register the S3 bucket with the CloudNet distribution. Then in your applications, Web pages, and links, you reference the distribution location.

CloudFront represents the last of the Amazon Web Services that store and serve objects and files. To store data in a way that makes it searchable and organizes it, Amazon offers two different database services that are covered in the next section.

Understanding Amazon Database Services

Amazon offers two different types of database services: Amazon SimpleDB, which is non-relational, and Amazon Relational Database Service (Amazon RDS), both of which were in beta at the time of this writing. Dynamic data access is a central element of Web services, particularly "Web 2.0" services, so although AMIs support several of the major databases, it isn't surprising that they would create their own databases as part of the AWS Service Oriented Architecture.

Amazon SimpleDB

Amazon SimpleDB is an attempt to create a high performance data store with many database features but without the overhead. This is analogous to the goals used to create the Amazon Simple Storage System (S3). The service is meant to be low touch, in that it abstracts many of the common concerns of database administrators for hardware requirements, software maintenance, indexing, and performance optimization.

To create a high performance "simple" database, the data store created is flat; that is, it is non-relational and joins are not supported. Data stored in SimpleDB domains doesn't require maintenances of a schema and is therefore easily scalable and highly available because replication is built into the system. Data is stored as collections of items with attribute-value pairs, and the system is akin to using the database function within a spreadsheet. To support replication, a set of two consistency functions are part of SimpleDB that check data across the different copies. Transactions are performed as a set of conditional PUTS and DELETES, and you can INSERT, REPLACE, or DELETE values for item attributes. These transaction capabilities do not

enable features like ROLLBACK, but they allow you to create solutions that maintain optimistic concurrency control and will perform an INSERT based on the value of a counter or timestamp.

You grow a SimpleDB database by scaling out and creating additional data domains, and SimpleDB integrates with EC2 instances and S3 storage. Data objects stored in S3 can be queried in SimpleDB, returning information about the objects' metadata and pointers to the objects' location.

Data in SimpleDB is automatically indexed and may be queried as needed. The API is relatively simple, consisting of domain creation, put and get attributes, and SELECT statements. According to Amazon, query performance is near the level you would see for a database on a LAN, as access through a browser. Although a SimpleDB database is replicated and therefore made highly available and fault tolerant, the service lacks many of the speed enhancements available to relational systems. A data domain may be located geographically in any of AWS's regions.

The design goal was to remove as much of the database system maintenance as possible. In a Web services architecture, many applications don't require the performance level of a relational database. Among the featured uses of SimpleDB are data logging, online gaming, and metadata indexing. SimpleDB would not be the best choice for a high-volume transaction system. Data transfers within regions between SimpleDB and other AWS services are free. Service charges accrue based on SimpleDB Machine Hours and inter-regional data transfers.

The three areas of use for SimpleDB that Amazon Web Services highlights are: logging (http:// aws.amazon.com/simpledb/usecases_logging/), online gaming (http://aws.amazon.com/simpledb/usecases_online_gaming/), and metadata indexing (http:// aws.amazon.com/simpledb/usecases_metadata_indexing/).

Amazon Relational Database Service (RDS)

Amazon Relational Database Service is a variant of the MySQL5.1 database system, but one that is somewhat simplified. The purpose of RDS is to allow database applications that already exist to be ported to RDS and placed in an environment that is relatively automated and easy to use. RDS automatically performs functions such as backups and is deployable throughout AWS zones using the AWS infrastructure.

In RDS, you start by launching a database instance in the AWS Management Console and assigning the DB Instance class and size of the data store. The DB Instance is then connected to your MySQL database. Any database tool that works with

MySQL 5.1 will work with RDS. Additionally, you can monitor your database usage as part of Amazon CloudWatch.

Table 9.4 shows the different Instance Classes for an Amazon RDS database. Pricing is based on machine hour rates by class, by amount of storage per month, and per million requests.

TABLE 9.4

Amazon Relational Database Service Instance Class

Type1	Compute Engine	RAM (GB)	Platform	Price2
Small DB Instance (default)	1 EC2 Compute Unit (1 virtual core)	1.7	64-bit	$0.11
Large DB Instance	2 EC2 Compute Units (2 virtual cores X 2 EC2 Units)	7.5	64-bit	$0.44
Extra Large DB Instance	8 EC2 Compute Units (4 virtual cores X 2 EC2 Units)	15	64-bit	$0.88
Double Extra Large DB Instance	13 EC2 Compute Units (4 virtual cores X 3.25 EC2 Units)	34	64-bit	$1.55
Quadruple Extra Large DB Instance	26 EC2 Compute Units (8 virtual cores X 3.25 EC2 Units)	68	64-bit	$3.10

1. Storage available is from 5GB to 1TB.

2. Price for U.S. N. Virginia deployment for database machine; storage price is $0.10 per GB-month; and I/O rate price is $0.10 per 1 million requests for the same location. Data transfer rates also apply.

Among the important features of RDS is the automated point-in-time backup system for data in the database as well as for the MySQL transaction logs. Backups can be saved for up to eight days. In addition to backup, RDS supports database snapshots. A DB Snapshot is stored as a full database backup and is retained until you specifically delete it from your storage container. Snapshots may be scheduled or may be manually initiated by an administrator.

The deployment of RDS databases can be spread among multiple availability zones for increased fault tolerance and data availability. These so-called ‒Multi-AZ Deployments‖ can be automatically replicated and maintain a standby replica in another availability zone, with automatic failover when a database disruption is detected. The conversion of a single location RDS database to a Multi-DB deployment may be accomplished with a single API call. Other API calls support instance creation and maintenance, snapshots, and restores.

Choosing a database for AWS

In choosing a database solution for your AWS solutions, consider the following factors in making your selection:

- Choose SimpleDB when index and query functions do not require relational database support.
- Use SimpleDB for the lowest administrative overhead.
- Select SimpleDB if you want a solution that autoscales on demand.
- Choose SimpleDB for a solution that has a very high availability.
- Use RDS when you have an existing MySQL database that could be ported and you want to minimize the amount of infrastructure and administrative management required.
- Use RDS when your database queries require relation between data objects.
- Chose RDS when you want a database that scales based on an API call and has a pay-asyou-use-it pricing model.
- Select Amazon EC2/Relational Database AMI when you want access to an enterprise relational database or have an existing investment in that particular application.
- Use Amazon EC2/Relational Database AMI to retain complete administrative control over your database server.

Using Microsoft Cloud Services

Microsoft has a very extensive cloud computing portfolio under active development. Efforts to extend Microsoft products and third-party applications into the cloud are centered around adding more capabilities to existing Microsoft tools. Microsoft's approach is to view cloud applications as software plus service. In this model, the cloud is another platform and applications can run locally and access cloud services or run entirely in the cloud and be accessed by browsers using standard Service Oriented Architecture (SOA) protocols.

Microsoft calls their cloud operating system the Windows Azure Platform. You can think of Azure as a combination of virtualized infrastructure to which the .NET Framework has been added as a set of .NET Services. The Windows Azure service itself is a hosted environment of virtual machines enabled by a fabric called Windows Azure AppFabric. You can host your application on Azure and provision it with storage, growing it as you need it. Windows Azure service is an Infrastructure as a Service offering. A number of services interoperate with Windows Azure, including SQL Azure (a version of SQL Server), SharePoint Services, Azure Dynamic CRM, and many of Windows Live Services comprising what is the Windows Azure Platform, which is a Platform as a Service cloud computing model. Eventually, many more services will be added, encompassing the whole range of Microsoft's offerings. This architecture positions Microsoft to either extend its product into the Web or to license its products, whichever way the cloud computing marketplace develops. From Microsoft's position and that of its developers, Windows Azure has lots of advantages.

Windows Live Services is a collection of applications and services that run on the Web. Some of these applications called Windows Live Essentials are add-ons to Windows and downloadable as applications. Other Windows Live Services are standalone Web applications viewable in a browser. An important subset of these Windows Live Services is available to Windows Azure applications through the Windows Live Messenger Connect API. A set of Windows Live for Mobile applications also exists. These applications and services are more fully described in this chapter.

Exploring Microsoft Cloud Services

Microsoft CEO Steve Balmer recently said at a University of Washington speech that Microsoft was ―betting our company‖ on the cloud. Balmer also claimed that about 70 percent of Microsoft employees were currently working on cloud-related projects and that the number was expected to rise to about 90 percent within a year. Plans to integrate cloud-based applications and services into the Microsoft product portfolio dominates the thinking at Microsoft and is playing a central role in the company's ongoing product development. The starting place for Microsoft's cloud computing efforts may be found at Microsoft.com/cloud, shown in Figure 10.1.

Microsoft has a vast array of cloud computing products and initiatives, and a number of industry leading Web applications. Although services like America Online Instant Messenger (AIM) may garner mindshare in the United States, surprisingly Microsoft Messenger is the market leader in many other countries. Product by product in any category you can name—calendars, event managers, photo galleries, image editors, movie making, and so on—Microsoft has a Web application for it. Some of these products are also-rans, some are good, some are category leaders, and a few of them are really unique. What is also true is that Web apps are under very active development. Microsoft sees its on-line application portfolio as a way of extending its desktop applications to make the company pervasive and to extend its products' lives well into the future.

Going forward, Microsoft sees its future as providing the best Web experience for any type of device, which means that it structures its development environment so the application alters its behavior

depending upon the device. For a mobile device, that would mean adjusting the user interface to accommodate the small screen, while for a PC the Web application would take advantage of the PC hardware to accelerate the application and add richer graphics and other features. That means Microsoft is pushing cloud development in terms of applications serving as both a service and an application. This duality—like light, both a particle and a wave—manifests itself in the way Microsoft is currently structuring its Windows Live Web products. Eventually, the company intends to create a Microsoft app store to sell cloud applications to users.

Microsoft Live is only one part of the Microsoft cloud strategy. The second part of the strategy is the extension of the .NET Framework and related development tools to the cloud. To enable .NET developers to extend their applications into the cloud, or to build .NET style applications that run completely in the cloud, Microsoft has created a set of .NET services, which it now refers to as the Windows Azure Platform. .NET Services itself had as its origin the work Microsoft did to create its BizTalk products.

Azure and its related services were built to allow developers to extend their applications into the cloud. Azure is a virtualized infrastructure to which a set of additional enterprise services has been layered on top, including:

- A virtualization service called Azure **AppFabric** that creates an application hosting environment. AppFabric (formerly .NET Services) is a cloud-enabled version of the .NET Framework.
- A high capacity non-relational storage facility called Storage.
- A set of virtual machine instances called Compute.
- A cloud-enabled version of SQL Server called SQL Azure Database.
- A database marketplace based on SQL Azure Database code-named ―Dallas.‖ An xRM (Anything Relations Management) service called Dynamics CRM based on Microsoft Dynamics.
- A document and collaboration service based on SharePoint called SharePoint Services. Windows Live Services, a collection of services that runs on Windows Live, which can be used in applications that run in the Azure cloud.

FIGURE 10.1

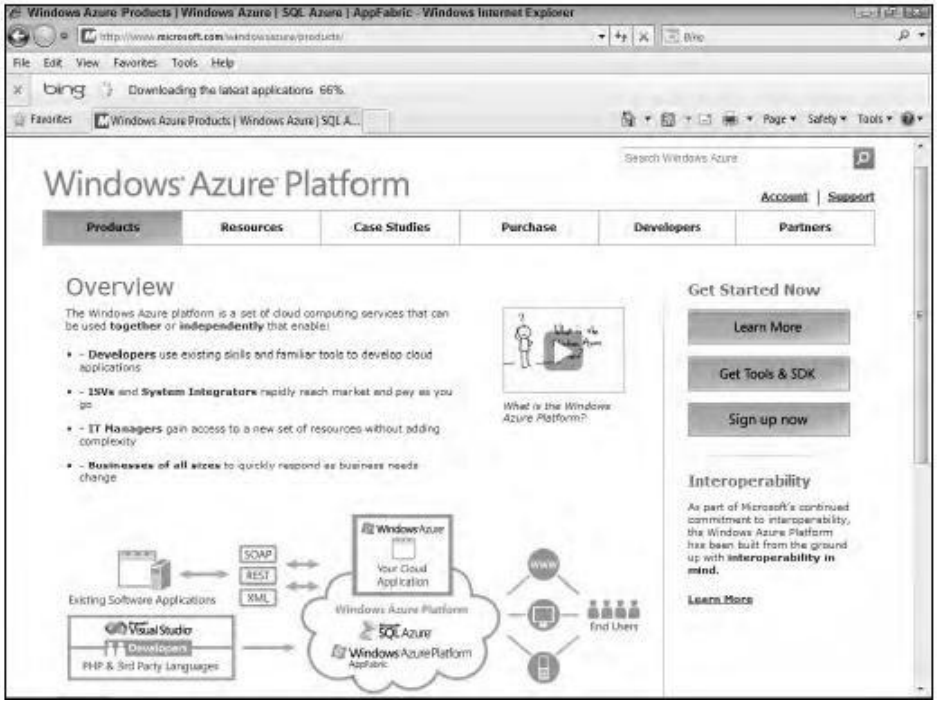
Microsoft maintains a home page for cloud computing at `http://www.microsoft.com/cloud`.

Eventually the entire Microsoft server portfolio will be available as a cloud-based application or service, including Exchange. So the Windows Azure Platform can be viewed in a sense as the next Microsoft operating system, the first one that is a cloud OS. The Microsoft vision for the Windows Azure Platform is shown in Figure 10.2, where the company sees applications developed in Visual Studio or through PHP and other languages deployed to the cloud, existing local (on-premises) applications interacting with Azure with standard SOA protocols (SOAP, REST, and XML), all running on the Windows Azure virtualized infrastructure.

The end result is pervasive computing available to users on the device of their choice. Just how Microsoft intends to integrate all these technologies into a unified offering is the story of this chapter.

FIGURE 10.2

The integrated vision for application development and deployment with Azure is illustrated in this overview page of the Azure platform (http://www.microsoft.com/windowsazure/products/).



Defining the Windows Azure Platform

Azure is Microsoft's Infrastructure as a Service (IaaS) Web hosting service. Azure is a deep blue color, the color of the clear sky onto which you can paint clouds. Taken together as a unit, Windows Azure Platform becomes a Platform as a Service (PaaS) offering. Hence, you may run into some people calling Azure an infrastructure service and others calling it a platform; in context, both are correct. Compared to Amazon's and Google's cloud services, Azure (the service) is a competitor to AWS. Windows Azure Platform is a competitor to Google's App Engine.

Figure 10.3 shows the home page of the Windows Azure Platform found at http://www.microsoft.com/windowsazure.

A developer creates an Azure application by first logging onto the Azure portal from the Sign up now button shown in Figure 10.3, supplying a Windows Live ID, creating a hosted account, and provisioning a storage account. The completed application can then be made available to users as a hosted application or service.

The software plus services approach

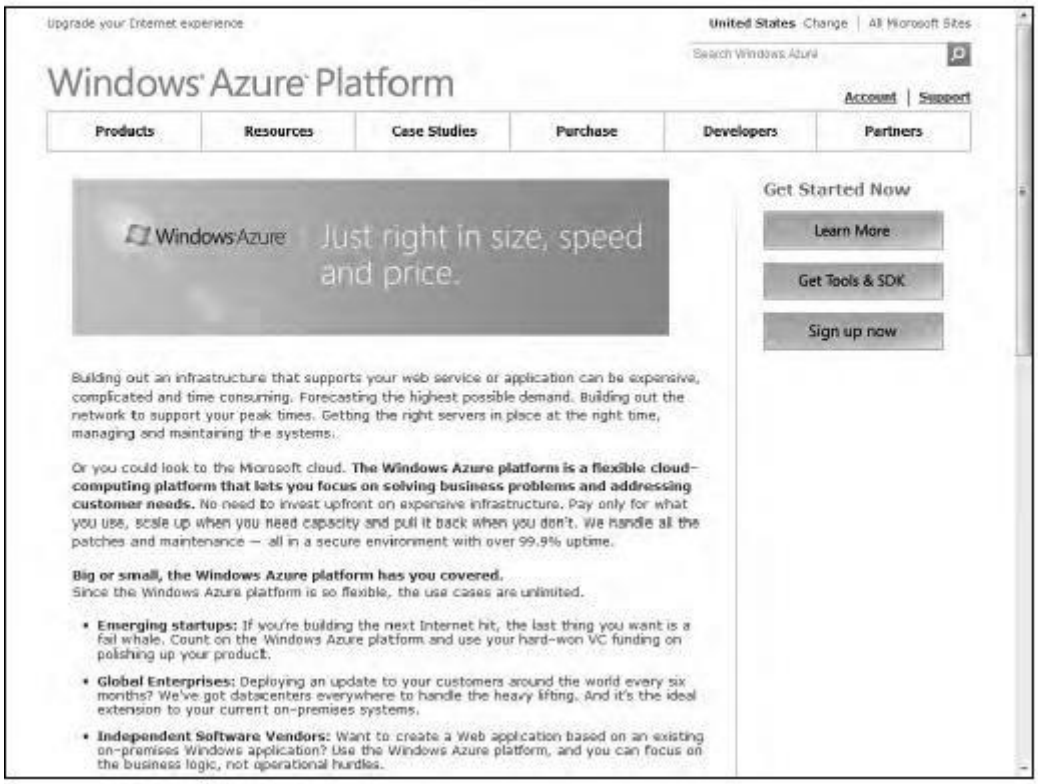
Microsoft has a very different vision for cloud services than either Amazon or Google does. In Amazon's case, AWS is a pure infrastructure play. AWS essentially rents you a (virtual) computer on which to run your application. An Amazon Machine Image can be provisioned with an operating system, an enterprise application, or application stack, but that provisioning is not a prerequisite. An AMI is your machine, and you can configure it as you choose. AWS is a deployment enabler.

Google's approach with its Google App Engine (GAE) is to offer a cloud-based development platform on which you can add your program, provided that the program speaks the Google App Engine

API and uses objects and properties from the App Engine framework. Google makes it possible to program in a number of languages, but you must write your applications to conform to Google's infrastructure. Google Apps lets you create a saleable cloud-based application, but that application can only work within the Google infrastructure, and the application is not easily ported to other environments.

FIGURE 10.3

Window Azure Platform's home page may be found at `http://www.microsoft.com/windowsazure`, and is shown in this figure.



Microsoft sees the cloud as being a complimentary platform to its other platforms. The company envisages a scenario where a Microsoft developer with an investment in an application wants to extend that application's availability to the cloud. Perhaps the application runs on a server, desktop, or mobile device running some form of Windows. Microsoft calls this approach *software plus services.*

The Windows Azure Platform allows a developer to modify his application so it can run in the cloud on virtual machines hosted in Microsoft datacenters. Windows Azure serves as a cloud operating system, and the suitably modified application can be hosted on Azure as a runtime application where it can make use of the various Azure Services. Additionally, local applications running on a server, desktop, or mobile device can access Windows Azure Services through the Windows Services Platform API.

Given that Microsoft owns the Office application market as well as the desktop OS market, this approach makes lots of sense. It is also quite possible that a hybrid application that can reside either locally or in the cloud will have lots of appeal not only to developers but to users who would prefer more control over their data and more security than the cloud might offer.

The Azure Platform

With Azure's architecture (shown in Figure 10.4), an application can run locally, run in the cloud, or some combination of both. Applications on Azure can be run as applications, as background

processes or services, or as both. The Windows Azure service itself is shown as the oval in Figure 10.4 and is a cloud-based operating system with a fabric infrastructure of virtual machines hosted in Microsoft datacenters.

The Azure Windows Services Platform API uses the industry standard REST, HTTP, and XML protocols that are part of any Service Oriented Architecture cloud infrastructure to allow applications to talk to Azure. Developers can install a client-side managed class library that contains functions that can make calls to the Azure Windows Services Platform API as part of their applications. These API functions have been added to Microsoft Visual Studio as part of Microsoft's Integrated Development Environment (IDE). There are plans to add IPsec connectivity to Azure in the near future. *IPsec* refers to the Internet Protocol Security protocol suite for creating a secure Internet connection between two endpoints. IPsec provides for authenticated communication using sessionbased negotiation and the exchange of cryptographic keys to enable encrypted communication to be sent and decrypted. IPsec is an IETF standard that is in wide use.

The Azure Service Platform hosts runtime versions of .NET Framework applications written in any of the languages in common use, such as Visual Basic, C++, C#, Java, and any application that has been compiled for .NET's Common Language Runtime (CLR). Azure also can deploy Web-based applications built with ASP.NET, the Windows Communication Foundation (WCF), and PHP, and it supports Microsoft's automated deployment technologies. Microsoft also has released SDKs for both Java and Ruby to allow applications written in those languages to place calls to the Azure Service Platform API to the AppFabric Service.

The Windows Azure service

Windows Azure is a virtualized Windows infrastructure run by Microsoft on a set of datacenters around the world. In Figure 10.4, the dashed oval encloses the portion of the Windows Azure Platform that is Azure itself—that is, the portion of the platform that is the IaaS part, which is shown in more detail in Figure 10.5.

Six main elements are part of Windows Azure:

- **Application:** This is the runtime of the application that is running in the cloud.
- **Compute:** This is the load-balanced Windows server computation and policy engine that allows you to create and manage virtual machines that serve either in a Web role and a Worker role.

A Web role is a virtual machine instance running Microsoft IIS Web server that can accept and respond to HTTP or HTTPS requests. A Worker role can accept and respond to requests, but doesn't run IIS in that virtual machine. Worker roles can communicate with Azure Storage or through direct connections to clients.

- **Storage:** This is a non-relational storage system for large-scale storage.

Azure Storage Service lets you create drives, manage queues, and store BLOBs (Binary Large Objects). You manipulate content in Azure Storage using the REST API, which is based on standard HTTP requests and is therefore platform-independent. Stored data can be read using GETs, written with PUTs, modified with POSTs, and removed with DELETE requests.

Azure Storage plays the same role in Azure that Amazon Simple Storage Service (S3) plays in Amazon Web Services. For relational database services, SQL Azure may be used.

- **Fabric:** This is the Windows Azure Hypervisor, which is a version of Hyper-V that runs on Windows Server 2008.
- **Config:** This is a management service.

- **Virtual machines:** These are instances of Windows that run the applications and services that are part of a particular deployment.

FIGURE 10.4

The Windows Azure Platform extends applications running on other platforms to the cloud using Microsoft infrastructure and a set of enterprise services.

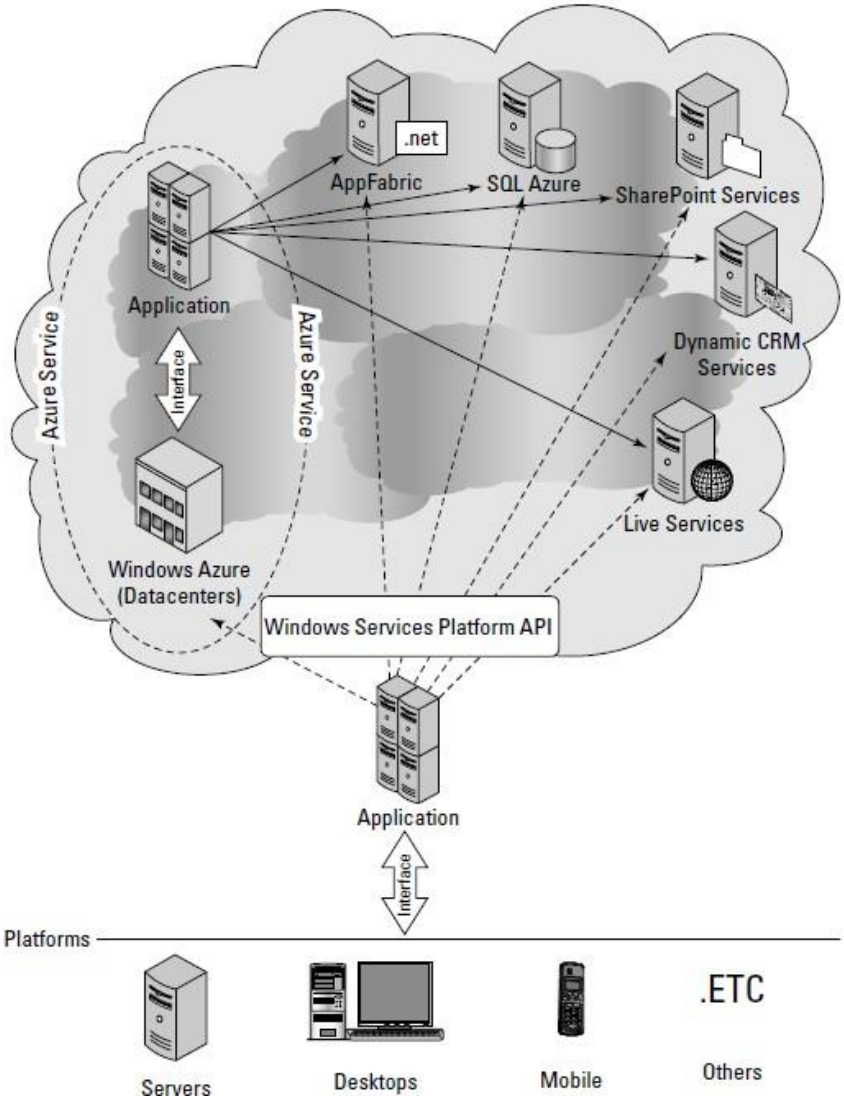


FIGURE 10.5

Windows Azure is a virtualized infrastructure that provides configurable virtual machines, independent storage, and a configuration interface.

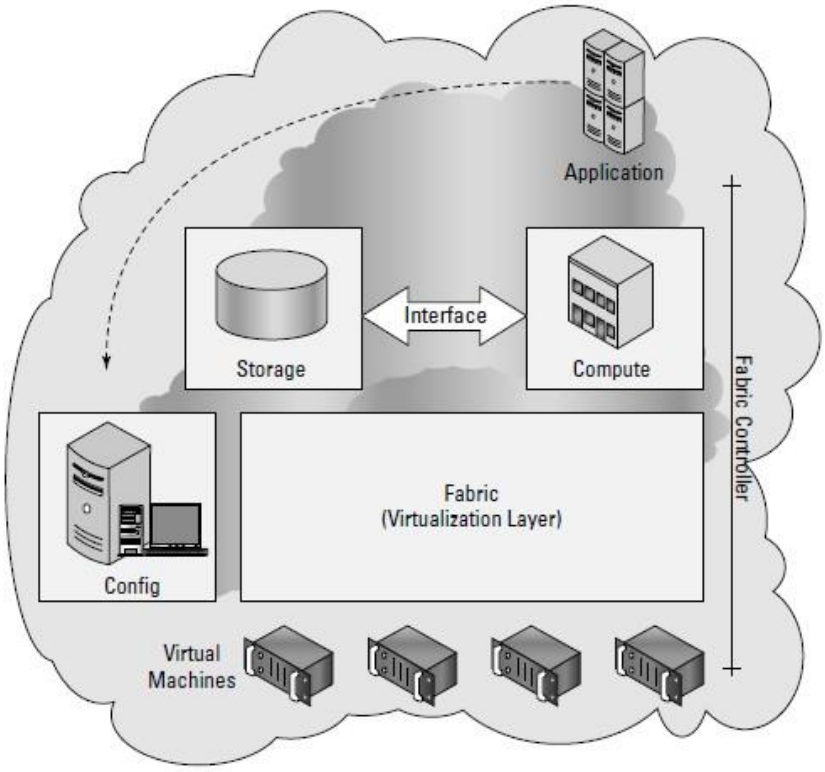


Table 10.1 shows the different Virtual Machine sizes available on Windows Azure.

TABLE 10.1

Windows Azure Virtual Machine Sizes

VM Size1	CPU Cores	Memory (GB)	Disk Space for Local Storage Resources (GB)
Small	1	1.7	250
Medium	2	3.5	500
Large	4	7	1000
ExtraLarge	8	14	2000

1. Microsoft has not released information that would allow you to match VM sizes to physical systems based on real CPUs.

1. Microsoft has not released information that would allow you to match VM sizes to physical systems based on real CPUs.

The portion of the Azure environment that creates and manages a virtual resource pool is called the Fabric Controller. Applications that run on Azure are memory-managed, load-balanced, replicated, and backed up through snapshots automatically by the Fabric Controller.

Windows Azure AppFabric

Azure AppFabric (http://msdn.microsoft.com/en-us/windowsazure/netservices.aspx) is a Service Bus and Access Control facility based on .NET technology for client requests to Web services on Azure.

Previously, these services were called Microsoft .NET Services. Azure AppFabric supports the standard Service Oriented Architecture (SOA) protocols such as REST and SOAP and the WS- protocols.

The function of a service bus in an SOA is to expose distributed services as an endpoint with a specific URI that clients can request services from, as shown in Figure 10.6. A particular set of endpoints and its associated Access Control rules for an application is referred to as the service namespace. Each namespace is assigned a management key that is part of the security mechanism. The Service Bus service registry makes endpoints discoverable, if so configured.

Azure AppFabric manages requests by locating the service, communicating the request, and making the necessary connection possible by performing network address translation, opening appropriate ports in any intervening firewalls. AppFabric manages the transaction to ensure that it is completed and that a response is sent to the client. A service bus also can serve to negotiate the exchange of information between a client and the service.

Azure AppFabric acts as an SOA service bus, as shown in Figure 10.6. AppFabric can provide a negotiated traversal of services through firewalls and NATs as a relay service using the Service Bus' rendezvous address. A rendezvous address not only includes the service URI, but also includes the namespace of the service bus. Alternatively, if both applications comply to .NET Services a direct connection between the applications can be used instead with the required NAT traversal information for the direct connection provided by the relay service of the Service Bus. NAT (Network Address Traversal) is a system for creating and maintaining Internet connections for TCP or UDP traffic where the connection point is hidden behind a router or a firewall and routing is performed by one of several possible mechanisms.

The Access Control portion of Azure AppFabric is a claims access control system that provides a token-based trust mechanism for identity management. An application or user, as shown on the right of Figure 10.7, presents a claim for a service from an application on the left. The Access Control examines the request, and if it finds it to be valid, it grants a security token to the client.

These steps are associated with Access Control:

1. The client requests authentication from Access Control.
2. Access Control creates a token based on the stored rules for server application.
3. A token is signed and returned to the client application.
4. The client presents the token to the service application.
5. The server application verifies the signature and uses the token to decide what the client application is allowed to do.

Access Control allows one application to trust the identity of another application. This mechanism can federate with identity providers such as Active Directory Federation Services (ADFS v2) to create distributed systems based on SOA.

FIGURE 10.6

Azure AppFabric service pathways

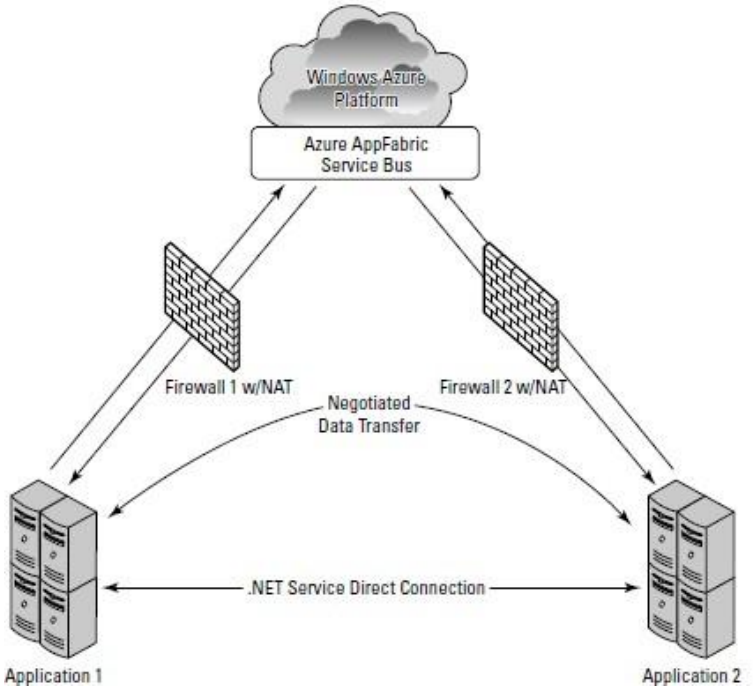
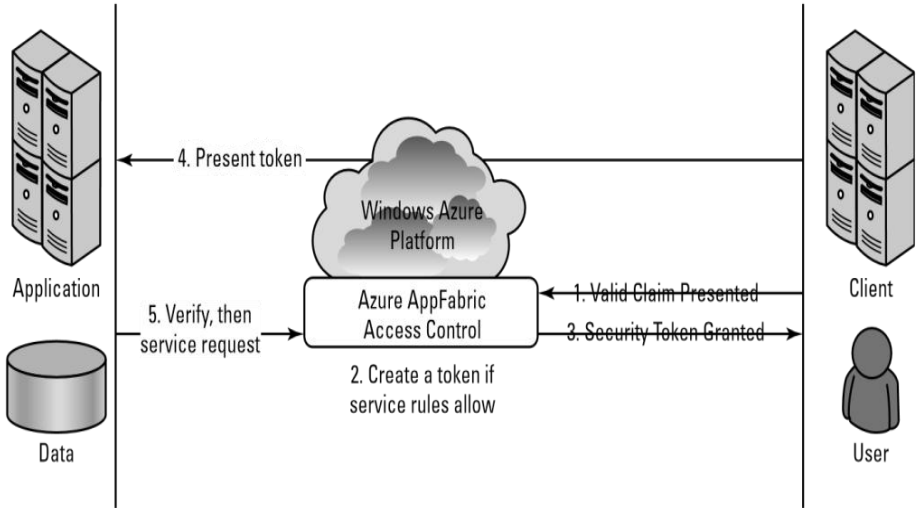


FIGURE 10.7

Azure AppFabric Access Control enables secure application requests through a token mechanism.



Microsoft likes to refer to the Azure AppFabric as an ―Internet Service Bus‖ to differentiate it from the standard Enterprise Service Bus (ESB) that you find in SOA architectures. AppFabric has all the same components of an ESB, namely service orchestration, federated identity, access control, a namespace, service registry, and a messaging fabric, but it locates these components in the cloud. Often ESBs are located on LANs. According to Microsoft, this approach abstracts away from application developers the challenges related to NAT traversal, DDNS (Dynamic DNS), and UPnP. ESBs are described in Chapter 13; please refer to that chapter for further discussion on this topic.

Azure Content Delivery Network

The Windows Azure Content Delivery Network (CDN) is a worldwide content caching and delivery system for Windows Azure blob content. Currently, more than 18 Microsoft datacenters are

hosting this service in Australia, Asia, Europe, South America, and the United States, referred to as endpoints. CDN is an edge network service that lowers latency and maximizes bandwidth by delivering content to users who are nearby.

Any storage account can be enabled for CDN. In order to share information stored in an Azure blob, you need to place the blob in a public blob container that is accessible to anyone using an anonymous sign-in. The Azure portal lists the domain name of the blob container in the form http://<*guid*>.vo.msecnd.net/. You also can register a custom domain name for a Windows Azure CDN endpoint.

For content in a public container named ―Box‖ in the storage account named ―MyAccount,‖ a user would access the content with one of the following URLs: Windows Azure Blob services URL: http://<*MyAccount*>.blob.core.windows. net/<*Box*>/

Windows Azure CDN URL: http://<*guid*>.vo.msecnd.net/<*Box*>/

When the Blob service URL is used, the request is redirected to the closest CDN endpoint to the client. The CDN service searches that location and serves the content; if the content isn't found, the CDN retrieves the Blob from the Blob service, caches the content, and then serves it to the user. Parameters can be set that determine how long content is cached (Time-To-Live, TTL), with the default being 72 hours.

SQL Azure

SQL Azure is a cloud-based relational database service that is based on Microsoft SQL Server. Initially, this service was called SQL Server Data Service. An application that uses SQL Azure Database can run locally on a server, PC, or mobile device, in a datacenter, or on Windows Azure. Data stored in an SQL Azure database is accessed using the Tabular Data Stream (TDS) protocol, the same protocol used for a local SQL Server database. SQL Azure Database supports TransactSQL statements.

Azure data is replicated three times for data protection and writes are checked for consistency. SQL Azure eventually will support the Microsoft Sync Framework providing a facility for SQL Azure Databases to synchronize their data with local databases.

There is a current limit of 10GB for each SQL Azure Database. Queries against a single database are unified. However, if the storage size exceeds the limit, then data must be partitioned into logical sets and queries need to be structured to account for this partitioning. For example, names in a database might have to be partitioned A-K, L-R, and S-Z. SQL Azure Database is a shared database environment, and limitations are placed on how long a query can run or how many resources a query can use.

From the standpoint of any application, an SQL Azure Database looks like and behaves like a local database with a few exceptions. The current exceptions are that the SQL Common Language Runtime (CLR) and support for spatial data were not included, although support will be added later for them. The biggest difference is that because SQL Azure is managed in the cloud, there are no administrative controls over the SQL engine. You can't shut the system down, nor can you directly interact with the SQL Servers.

Windows Azure pricing

Prices for working with the Windows Azure Platform are based either on a ―consumption‖ (pay-asyou-go) model or through various contracts for levels of monthly service that Microsoft calls

–commitments.‖ When you exceed the subscription level of your commitment, the additional usage is charged on the consumption model.

Current pricing for Windows Azure is as follows:

- Compute: $0.12 / hour
- Storage: $0.15 / GB stored / month
- Storage transactions: $0.01 / 10K
- Data transfers (excluding CDN): $0.10 in / $0.15 out / GB ($0.30 in / $0.45 out / GB in Asia)
- CDN data transfers: $0.15 GB for North America and Europe ($0.20 GB elsewhere)
- CDN transactions: $0.01 / 10K

A transaction is an application request.In the Windows Azure Service Level Agreement, Microsoft states that it guarantees an external connectivity between two or more role instances that are located in different Azure domains of at least 99.95 percent uptime. The connection between storage and Microsoft's Content Delivery Network (CDN, described below) is stated to be at least 99.9 percent uptime.

SQL Azure charges are based on two different programs:

- Web Editions: Up to 1GB database = $9.99 / month; up to 5GB database = $49.95 / month
- Business Edition: Up to 10GB database = $99.99 / month; up to 20GB database = $199.98 / month; up to 30GB database = $299.97 / month; up to 40GB database = $399.96 / month; up to 50GB database = $499.95 / month Data transfers: $0.10 in / $0.15 out / GB ($0.30 in / $0.45 out / GB in Asia)

These are the charges for Windows Azure Platform AppFabric:

- Access Control transactions of $1.99 / 100K transactions
- Service Bus connections: $3.99 per connection on a –pay-as-you-go‖ basis, $9.95 for a pack of 5 connections, $49.75 for a pack of 25 connections, $199 for a pack of 100 connections, and $995 for a pack of 500 connections
- Data transfers: $0.10 in / $0.15 out / GB ($0.30 in / $0.45 out / GB in Asia)

Given that Windows Azure is a relatively new service and that IaaS likely will become a very competitive market, pricing is sure to change over time. You should definitely check the pricing page for current pricing if you are thinking of deploying on Azure.

Microsoft offers a TCO calculator for the Windows Azure Platform that you may find useful in determining your costs and savings. To access the calculator use the following link: http://www.microsoft.com/windowsazure/economics/#tcoCompare-LB.

Windows Live services

Windows Live is a collection of cloud-based applications and services, some of which can be used inside applications that run on Windows Azure Platform. Some Windows Live applications run as standalone applications and are available to users directly through a browser. Others are services that add capabilities to the Windows Azure Platform as part of Microsoft's software plus services strategy.

Microsoft has rolled out Windows Live in sets of releases they describe as four waves. The first wave was a rebranding of several Microsoft MSN applications and services in late 2005. More applications including Windows Mail, Windows Photo Gallery, and Windows Movie Maker were unbundled from Vista and rolled into a downloadable software suite called Windows Live Essentials. There has been continuous development, branding, marketing, and rebranding of the Windows Live portfolio that has had many people scratching their heads. Many Windows Live applications have been

rolled into other services or discontinued entirely.Here's what I believe the current situation is with Windows Live. If an application is bundled as part of an additional download for desktop users, it is part of the Windows Live Essentials package. Some applications that are part of Windows Live are standalone products, while others are extensions of existing Microsoft commercial software. An example of a standalone product would be Windows Live Calendar. An example of a cloud-based line extension is Windows Live Office, described more fully in Chapter 16.

Some parts of the Windows Live portfolio are shared applications and services that are accessible to developers, and those services are the Windows Live Services that are one component of the Windows Azure Platform. Developers access the services for Windows Live Services through a collection of APIs and controls called Windows Live Messenger Connect (previously called Live Services and Windows Live Dev). Using these APIs and controls, developers can add Windows Live Services capabilities and data to their application.

Messenger Connect was released as part of the Windows Live Wave 4 at the end of June 2010, and it unites APIs such as Windows Live ID, Windows Live Contacts, and Windows Live Messenger Web Toolkit into a single API. Messenger Connect works with ASP.NET, Windows Presentation Foundation (WPF), Java, Adobe Flash, PHP, and Microsoft's Silverlight graphics rendering technology through four different methods:

- Messenger Connect REST API Service
- Messenger Connect .NET and Silverlight Libraries
- Messenger Connection JavaScript Libraries and Controls
- Web activity feeds, either RSS 2.0 or ATOM

Table 10.2 lists the current services that can be used by Windows Live Messenger Connect in applications and Web sites.

TABLE 10.2

Windows Live Services

Service Name	URL	Microsoft Description
Admin Center	Windows Live Admin Center SDK	A management utility for a domain using SOAP and RPC.
Alerts	Windows Live Alerts for RSS Feeds	Enables Windows Live Alerts from an RSS feed.
Alerts	Windows Live Alerts SDK	Allows developers to add Windows Live Alerts notification service to an application using SOAP.
Contacts	Windows Live Contacts API	Allows developers to use REST to query the Windows Live People Address Book, as well as to adjust permission to contact data based on the Windows Live ID Delegated Authentication protocol.
FeedSync	FeedSync	Synchronizes information obtained from RSS and ATOM sources.
Live Framework	Live Framework SDK	An API for building Live Mesh application based on Windows Live Services.
Live Framework	Live Framework Tools for Visual Studio	Includes the Live Mesh tools from Visual Studio 2008 and Visual Web Developer Express Edition 2008.
Messenger	Web Toolkit	UI controls for building Web applications using Windows Live Messenger.
Messenger	IM Control	A set of controls that can enable instant messaging in an application.
Messenger	Presence API	An API that can be used to indicate a Windows Live Messenger's presence and control instant messages to that person's browser using a set of HTTP commands.
Photo Gallery	Windows Live Photo Gallery SDK	Allows for the creation and editing of photos and videos using the Publishing Plug-in Platform of Windows Live Photo Gallery inside applications.
Spaces	Windows Live Spaces MetaWeblog API	An API that can use XML-RPC calls to get and send Weblog data.
Spaces	Windows Live Spaces API and Feeds	An API that integrates Windows Live Spaces, Windows Live Events, Windows Live Photos, and Windows Live Profile into applications.
Web Gadgets	Gadgets SDK	Lightweight, single-purpose applets that can run on Windows Live Personalized Experience and Windows Live Spaces.
Windows Live ID	Web Authentication	Used to integrate Windows Live ID authentication into a Web site.
Windows Live ID	Delegated Authentication	Allows an application to access data for an authenticated Windows Live ID user from Web services and sites that accept that authentication.
Windows Live ID	Client Authentication	An API for Windows Live ID sign-in from a desktop application.
Writer	Windows Live Writer SDK	Allows applications to incorporate the features of the Windows Live Writer in their application. Additional capabilities include features for creating and managing blogs, adding more content, and customizing the Windows Live Writer user interface.

Reference: Based on `http://en.wikipedia.org/wiki/Windows_Live_Messenger_Connect`. An API for Bing and the toolbar is also available as a service.

Using Windows Live

Windows Live includes several popular cloud-based services. The two best known and most widely used are Windows Live Hotmail and Windows Live Messenger, with more than 300 million users worldwide. Windows Live is based around five core services:

- E-mail
- Instant Messaging
- Photos
- Social Networking
- Online Storage

A user or application can consume Windows Live in a number of ways. Some Windows Live applications are entirely cloud-based Web services, so users can use these applications from within any browser. The Office Live applications described more fully in Chapter 16, ―Microsoft Office Web Apps,‖ is an example of this sort of service. Some of these services are aimed at mobile devices and are referred to as Windows Live for Mobile (described below), and they are consumed on conforming mobile devices. Some of these applications are client-side applications that you download from Windows Live for use on your desktop, of which Windows Live Essentials is the primary example.

You can access Windows Live services in one of the following ways:

- By navigating to the service from the command on the navigation bar at the top of Windows Live
- By directly entering the URL of the service
- By selecting the application from the Windows Live Essentials folder on the Start menu If you haven't signed into Windows Live during your session, Windows Live requests that you do so before allowing you to proceed.

Table 10.3 lists the current offerings of Windows Live Services.

TABLE 10.3

Windows Live Services Offerings

Service Name	URL	Description
Windows Live Account	http://account.live.com/	Management service for Windows Live ID and relationships.
Windows Live Admin Center	http://admin.live.com/	E-mail hosting for Web site owners.
Windows Live Alerts	http://alerts.live.com/	Generates alerts sent to e-mail, mobile device, or Windows Messenger.
Windows Live Calendar	http://calendar.live.com/	Calendar service with appointments, meetings, and events; can be shared with others.
Windows Live Contacts	http://contacts.live.com/	Address book service with synchronization feature.
Windows Live Devices	http://devices.live.com/	Synchronization and remote access service for files stored on PCs and mobile devices.
Windows Live Essentials	http://essentials.live.com/	Downloadable applications that supplement Microsoft Windows.
Windows Live Family Safety	http://fss.live.com/	Allows you to manage and monitor your children's Internet activity so they can surf the Web more safely.
Windows Live FrameIt	http://frameit.live.com/	Adds an RSS feed to digital photo frame devices.
Windows Live Gallery	http://gallery.live.com/	A collection of developer add-ons for Windows Live products.
Windows Live Groups	http://groups.live.com/	A group discussion, collaboration, sharing, and coordination tool.
Windows Live Home	http://home.live.com/	A personalization Web page and tool for Windows Live with status information and navigation features.
Windows Live Hotmail	http://hotmail.com/	A Web-based free e-mail service with contacts and calendar.
Windows Live ID	http://login.live.com/	A sign-on service shared by Windows Live applications.

Service Name	URL	Description
Windows Live Mail	http://mail.live.com	Desktop e-mail client with RSS; replaces Outlook Express and Windows Mail. You can use Live Mail to manage Gmail or Yahoo! Plus Mail accounts, as well as your POP e-mail services.
Windows Live Messenger	http://messenger.live.com/	Allows you to chat instantly with friends and family on your desktop, on the Web, and on your mobile phone.
Windows Live Messenger Companion	http://essentials.live.com/	Windows Live Essentials add-on for Internet Explorer; shares link to a page on the site you're visiting. You can see the page and add a comment.
Windows Live Movie Maker	http://essentials.live.com/	Allows you to create beautiful, memorable movies and then publish to the Web in a few clicks.
Windows Live Office	http://office.live.com/	Contains document creation and editing tools based on Office, Excel, PowerPoint, and OneNote with Windows Live SkyDrive storage.
Windows Live OneCare Safety Scanner	http://safety.live.com/	Consists of a PC scanner for viruses, spyware, and other malware. Features include disk cleaner, defragmenter, port scanner, and registry cleaner.
Windows Live Photo Gallery	http://photogallery.live.com/	Allows you to edit, organize, tag, and share your photos.
Windows Live Photos	http://photos.live.com/	Photo storage and sharing service. You can use the service to publish photos to third-party photo services.
Windows Live Profile	http://profile.live.com/	Profile information management service for user information.
Windows Live SkyDrive	http://skydrive.live.com/	Online file storage system service.
Windows Live Spaces	http://spaces.live.com/	Social networking, blogging, and photo-sharing site.
Windows Live Sync	http://sync.live.com/	File synchronization and sharing site based on Live Mesh; originally called folder share.
Windows Live Writer	http://writer.live.com/	Allows you to compose a blog post, add your photos and links to your videos, and then publish to the Web. You can post the blogs from Writer to Blogger, WordPress, and other services.

The following Windows Live services have been discontinued or rebranded:

- Windows Live Agent
- Windows Live Barcode
- Windows Live Call (now part of Windows Live Messenger)
- Windows Live Events
- Windows Live Expo
- Windows Live Favorites (now Part of Windows Live SkyDrive)
- Windows Live Help Community
- Windows Live Hotspot Locator (now MSN WiFi Hotspots)
- Windows Live OneCare (now Microsoft Security Essentials)
- Windows Live Personalized Experience
- Windows Live QnA (now MSN QnA)

- Windows Live Search Center (now Windows Search 4)
- Windows Live Shopping (now Bing Shopping)
- Windows Live Toolbar
- Windows Live TV
- Windows Live Video Messages
- Windows Live Web Messenger (now part of Windows Live Web services)
- Windows Live WiFi Center

Windows Live Essentials

Windows Live Essentials applications are a collection of client-side applications that must be downloaded and installed on a desktop. Some of these applications were once part of Windows and have been unbundled from the operating system; others are entirely new. Live Essentials rely on cloud-based services for their data storage and retrieval, and in some cases for their processing.

Windows Live Essentials currently includes the following:

- Family Safety
- Windows Live Messenger
- Photo Gallery
- Mail
- Movie Maker

The download page for Windows Live Essentials (http://essentials.live.com/) is shown in Figure 10.8. All the Windows Essentials are downloaded as a single file. This page also has links to download related software such as the Bing bar (which replaces the Windows Live Toolbar), Microsoft Office Outlook Connection, Office Live Add-in, and Microsoft Silverlight. When you install Windows Live Essentials, shortcuts for these programs are placed on the Windows Start menu.

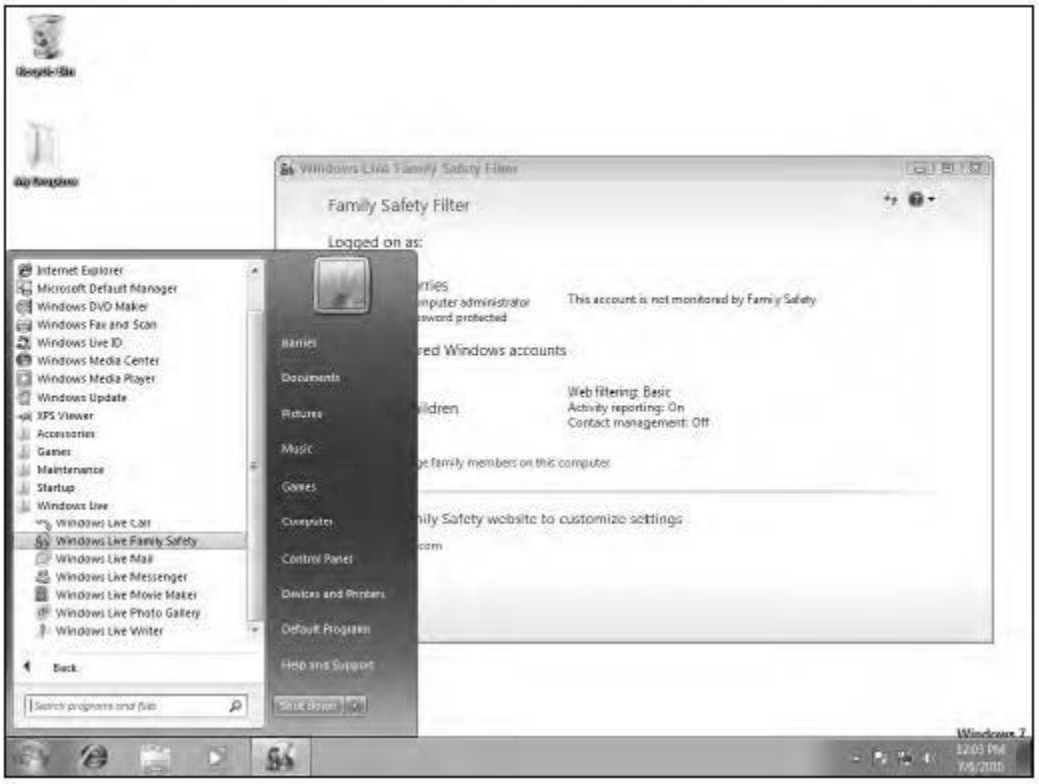
Windows Live Essentials help alleviate a long-standing problem of Microsoft with Windows by allowing Microsoft to unbundle some of its add-on applications for the operating system so they don't compete with other vendors' products unfairly. Live Essentials moves these applications partially onto the cloud, while making them available easily as a download and a service. Shown in Figure 10.9 is Windows Live Family Safety, which is a Web filter and activity reporting tool for Windows accounts on a per-machine basis.

FIGURE 10.8

The Windows Live Essentials home page (`http://essentials.live.com/`) provides links to the downloads of Microsoft's cloud-based client-side applications, the application's own page, as well as links to download related software.

FIGURE 10.9

Windows Live Essentials is available from the Start menu as a set of commands.



Windows Live Home

Windows Live Home is the central access page or portal for the Windows Live suite. An example is shown in Figure 10.10. The page provides navigation, lists activities, provides access to e-mail, shows your RSS feeds, and lists your account name and some related information. What you see on this page is customizable and depends on the services to which you are subscribed. The page can be themed, which changes the color, fonts, and look of the page.

These are the most commonly used features on Windows Live Home:

- Launching other Windows Live services
- Viewing e-mail headers from Hotmail and private messages from other users
- Viewing activity of people you follow
- Displaying weather information and RSS feed updates
- Managing calendars and events
- Viewing photos
- Modifying profile and relationships

FIGURE 10.10

Your personalized home page for Windows Live (`http://live.microsoft.com/home`) contains content and ads.

Windows Live for Mobile

Microsoft has a number of Windows Live services that are specifically meant to be run on mobile devices or cell phones that it calls Windows Live for Mobile (http://mobile.live.com). Some of these services run on the Windows Mobile platform, some are Web-based applications that conform to the lightweight Wireless Application Protocol (WAP) or on GPRS (General Packet Radio Service) browser, and some support SMS (Simple Message Service) systems.

The current list of these services includes the following:

- Live Mesh Mobile
- Windows Live Calendar Mobile
- Windows Live Contacts Mobile
- Windows Live Groups Mobile
- Windows Live Home Mobile
- Windows Live Messenger Mobile
- Windows Live Office Mobile
- Windows Live Profile Mobile
- Windows Live SkyDrive Mobile
- Windows Live Spaces Mobile