CCS : ch 3a [cloud computing services]

In this chapter, we will discuss about various cloud-based services. You will see that these end-services are at least, to some extent, similar to what IT managers offer from internally-hosted, traditional infrastructure. Both cloud-based services and the services provided by IT managers offer compute power and storage, an application-development and hosting platform, which users can use with little or no customization.

But once you look under the covers, you will see differences. For example, the virtualized (and usually vast) resource pooling and self-service characteristics are fundamental aspects of the cloud. In a public cloud, the infrastructure is multi-tenant. Most public clouds have hundreds or thousands of servers spread across the country or the world. A user can set up the environment easily using the self-service portal. There is often no initial expense. You are charged on a monthly basis (or whatever the billing period is) for the actual amount of resources you use for that month. It is just like a utility bill you get at home for phone or electricity usage.

Cloud makes it convenient to set up a new environment. In the cloud, it takes a short time (for example, a few hours) to set up servers with required storage, OS, database, and development tools that you want to use. On the other hand, to set up an internal, non-cloud server and storage system for a new development project or application, the roll-out will take weeks or months. In order to set up a new environment, you will have to go through various phases, namely, architecture, approval, negotiation, procurement, setup, installation, testing, and go-live. Each of these phases can take several weeks to complete.

The cloud services have been in use for years before the term "cloud computing" was coined. In fact, all services available in the cloud have been available since the dawn of computing. But now, the cost of setting up cloud infrastructure has become cheaper, the cloud delivery mechanisms have become more convenient to use, and the related bandwidth has become more widespread and affordable.

EXAM PRISM

In reality, the change or metamorphosis from using internal resources to use an external cloud has been so drastic and payback so compelling that small and mid-sized businesses in the future won't be required to spend money to set up their own internal, dedicated datacenters or IT infrastructure.

For larger businesses, several of their internal business units will move towards the cloud, unless compelled by government regulations to keep the data and applications within their internal premises. There are 3 key types of cloud-based service mechanisms:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

} SPI model

A term coined to capture the have-all services is **Anything as a Service** or XaaS. The most widespread examples of *XaaS* are *SaaS, PaaS, and IaaS*, collectively termed as the SPI model. Other examples of *XaaS* include *Security as a service (SaaS), Identity Management as a service (IdMaaS),*

Datacenter as a Service (DCaaS), Database as a Service (DBaaS), Storage as a Service, Hardware as a Service, or *ERP as a Service.*

Infrastructure as a Service (IaaS)

IaaS is a model in which you, as a customer, pay for the resources (such as compute power, memory, storage, bandwidth, security devices, load balancers, etc) kept at the provider's facility or wherever the provider keeps its hardware. The provider owns the equipment and maintains it at a level specified in the previously-agreed upon **Service Level Agreement (SLA)**. As a customer, all you need to do is to pay for the part of the resources dedicated permanently to your account or resources that you provision temporarily to meet the short-term needs.

What customer pays for?

EXAM PRISM

IaaS provider offers template OS images for the virtual servers. They also have APIs to add, start, stop, access, configure, and delete the virtual host (or virtual machine) and storage. The customer can neither control the hardware or network nor manage the underlying cloud infrastructure.

The customer can install OS, applications, libraries, and other software within the virtual machine and has limited control on networking and security equipment, such as host firewalls. Since the early 1980s, various offerings such as mainframes and thin computing (which are similar to today's *IaaS* and private clouds) have been built.

One of the pioneers of *IaaS* was a storage provider called **Storage Networks**, which was set up in the late 1990s in Waltham, Massachusetts. They marketed storage-as-a-utility with pay-per-use billing. They offered a menu of services, such as data archiving, backups, and replication. The user community was; however, not ready for the services and they were unfortunately far ahead of their times to be commercially successful.

In July 2003, its Board of Directors announced the closure of the business and approved a plan to liquidate the company. The company terminated all remaining employees of the company, with the exception of a small transition team to oversee the wind down of the business. In December 2003, a company called Rocket Software (**http://www.rocketsoftware.com/**) of Newton, Massachusetts bought some of the technical assets of the company. That was in early 2000s, when server virtualization or pay-per-use for hardware resources was uncommon as a marketable utility. Even the term cloud computing had not been coined. Fast forward it to the scenario today. Today with robust virtualization technologies (such as *Xen* from **Citrix**, *Hyper-V* from **Microsoft**, *vSphere* from **VMware** and open source software), high-speed broadband, and shrinking corporate budgets for IT, businesses are looking to buy or rent only what they need for the day. *IaaS* is the solution for them.

the early IaaS

To be commercially successful, the *IaaS* service must include the following:

What IaaS service should have?

- Utility-style computing service with pay-per-use billing
- Superior, world-class IT infrastructure and support
- Virtualized servers, storage, and network to form a shared pool of resources

- Dynamic scalability of memory, bandwidth, storage, and servers (usually to thousands of servers) to meet user needs in real-time
- Flexibility for users to add more or reduce the allocated resources
- Automation of administrative tasks
- Ability to view and manage resource utilization

As a user, you need to be sure that your *IaaS* provider has the above-mentioned attributes. Here is a list of questions to ask your provider:

1. What does the provider have in place to protect non-IT infrastructure, such as its datacenter, UPS, diesel generator, racks, and air-conditioners, and IT infrastructure, such as servers and storage?
2. How does it configure the security of the virtual machines?
3. How does the provider validate the integrity of the Virtual Machine Images (VMIs)?
4. How does it protect customer data, applications, and infrastructure from attacks by other tenants in the same cloud?
5. What tools does the provider use to detect security flaws?
6. What are the physical locations where data will be stored? This is required for compliance with certain regulations that need data to be in the same country.
7. How and at what frequency are the backups provided? Is backup data encrypted?
8. What are the DR and BCP plans?

Case Study 1: Content Distribution by a TV Channel Provider Using Public Cloud

A cable TV operator plans to expand its services to offer on-demand movies to users over an IP network. The users should be able to stream movies to their laptops, desktops or IP-enabled TVs over a DSL or broadband. The problem is that the datacenter location is far from the customer base. The operator does not have high-speed IP access to all its users. The movies are very large and must be available on-demand. The operator decides to host the movies on the public cloud. It will allow the operator to manage massive data files and bandwidth demands without increasing physical resources or link speeds. The operator will pay only for storage space and bandwidth consumed to stream videos to the paying users. The cloud provider has user authentication and Role-Based Access Control (RBAC) to control administrators from the TV operator company who are authorized to change the movie database.

Platform as a Service (PaaS)

PaaS is a cloud service where the customer gets a set of application and product development tool hosted on the provider's infrastructure. The customer can deploy acquired applications or those created using programming languages and tools supported by the provider. The consumer does not control, own or manage the underlying cloud infrastructure, including servers, OS, storage, security devices, network, but has a control over the deployed applications and possibly also the configurations of the hosting environment.

EXAM PRISM

PaaS democratizes the development of Web-based applications. It is particularly useful for Small and Mid-sized Enterprises (SMEs). The SMEs can have developers sitting at various remote locations and working together on the same platform, without the need to buy and set up servers for each of them. With PaaS, they can easily and instantly add more remote developers.

Developers on the *PaaS* platform create applications on the cloud platform using APIs, website portals or gateway software installed on the cloud servers. **Force.com** (a part of Salesforce.com), **Microsoft Azure** and **Google AppEngine** are a few leading *PaaS* providers.

Despite the advancement in this area, *PaaS* suffers from absence of widely-adopted standards for application portability between clouds.

Table 1 shows a comparison between a traditional development environment and *PaaS* on the basis of the features offered:

Table 1: Features of an Internal, Traditional Development Environment and PaaS

Feature	In-House Application Development	PaaS for Application Development
Multi-Tenancy	Intended for a single or a small group of users.	Supports hundreds to thousands of users, each with multiple active projects. Partition of data is must to protect several users.
User End-Points	Application-based tools, browsers.	Web browser-based tools.
Deployment	Deployment and scalability are left for installation and go-live phases.	Scalability, failover, and load-balancing are the basic building blocks.
Runtime Monitoring	Development solutions are usually not associated with runtime monitoring.	Built-in monitoring available with the development platform.
Virtual Machines, Servers, Storage, Databases	Multiple options are available and can be customized to meet any user requirement.	Need to work with the infrastructure offered by PaaS providers.
Integrated Development Environment (IDE)	May have separate environment and infrastructure for development, test, debugging, and production.	Same environment for all phases.

Table 2 shows the list of various categories of *PaaS*:

Table 2: PaaS Categories and Examples

Category	Description	Examples
I	These are *PaaS* offerings where software developers can use current tools, which they prefer to locally develop the software. Once they compile and test the application, they can upload it on the cloud.	Force.com, Google, LongJump, Magic Software, Microsoft, NetSuite, TIBCO, and Wavemaker.
II	The *PaaS* vendors provide a cloud-based development environment, which includes stages, such as design, coding, debugging, testing, staging, and deployment. All phases provide browser-based access. The developers build and test applications at the *PaaS* provider's site.	Force.com, Intuit, Trackvia, and Wolf Frameworks
III	These *PaaS* vendors target business experts, not coders or developers. The vendors provide tools, templates that can be easily customized to build applications. There is no need to write original code, it speeds up application delivery timeline.	Caspio, Cordys, Mendix, WorkXpress, and Zoho.
IV	These *PaaS* vendors enable developers to use tools for building the applications, either locally or in the cloud. The developers can use the cloud to stage, deploy, scale, and manage the applications.	Amazon, CloudBees, Engine Yard, Gigaspaces, IBM, Joyent, Microsoft, Red Hat, Standing Cloud, and Cloud Foundry (VMWae).

PaaS categories

Despite the availability of various *PaaS* vendors, consumers need to be cautious. Small, start-up vendors may not be in business a few years down the line and large established *PaaS* vendors may be slow to keep up with new offerings and attempt to sell their traditional products. There are a few well-established *PaaS* providers, such as **Force.com**, **Google**, and **Microsoft**.

EXAM PRISM

When you are looking for a PaaS provider, the key goal should be reduced time-to-market rather than cost savings. In addition, other factors, such as high availability, security, and scalability are important for the developers and testers.

The existing development user community must be familiar with the technologies and offerings within the selected *PaaS* service.

When PaaS Provider can be successful?

Leveraging PaaS for Productivity Q.1.1.

The following key features will increase a developer's productivity if they are effectively implemented on a *PaaS* site:

- **Ubiquitous Access and Quick Deployment**—This is particularly important for organizations whose developers are geographically spread out, mobile, and always changing in number. *PaaS* enables rapid implementation, scalability, and collaboration.
- **Caching**—A *PaaS* environment that supports caching for cloud resources will boost application performance. Developers would need an API to put an object or resource in the cache and to flush the cache.
- **Integrated Development Environment (IDE)**—A *PaaS* environment must have a browser-based development studio with an IDE for development, test, and debugging of applications. *PaaS* must provide the same development and production environment and the capability to build apps in various languages, such as Java, Python, C#, Ruby, or PHP.
- **Database**—Each *PaaS* must provide a database for developers to store and access data. For example, for *PaaS* cloud, **Force.com** has a service called **database.com** that enables you to build tables, fields, and relationships. It includes file storage, user management, authentication and development tools that make it easy to build applications. The database and APIs must support open standards, such as REST, OAuth, SAML, and SOAP.
- **Integration**—Integration with external databases and Web services and their compatibility is ensured with leading cloud providers, such as **Google AppEngine**, **Microsoft Azure**, **Amazon** or **Force.com**.
- **Logging**—A common requirement for all developers, regardless of the application they are developing. The *PaaS* environment must have **APIs** to open and close log files, write event logs, examine entries, and send alerts for certain events detected in the log files.
- **Identity Management**—Developers in a *PaaS* or traditional environment need to authenticate and manage users within their applications. Each user has a set of privileges managed by a role-based access mechanism. The *PaaS* cloud must support federated identity management system where a user once authenticated is given credentials to access services within the application as well as on other clouds. **APIs** with the *PaaS* should cache, use, and delete credentials as needed.
- **Messaging**—The *PaaS* cloud must provide ability to **APIs** to manage messages, such as the ability to post messages to any queue, consume messages, and examine message content without consuming them. It must support a highly-secured and on-demand collaboration throughout the *Software Development LifeCycle (SDLC)* with phases, such as design, development, testing, deployment, production, and support.
- **Job Processing**—The *PaaS* must provide ability to **APIs** to allow developers to start, monitor, pause, and stop large processing jobs, such as **Hadoop** style data mining. **Hadoop** is a software framework that enables applications to conveniently work with thousands of nodes and petabytes of data and is based on **Java** programming language.

- **Session Management**—*PaaS* must provide the ability to view access or change user sessions. When a user session migrates from one node to another in the cloud (say to server failure, for example) the *PaaS* environment must maintain the user session.
- **Service Discovery**—*PaaS* platform must give developers a convenient way to discover available services and the ability to search the cloud by service types. It must provide a range of search criteria for different service categories offered by the provider.

Besides the above-mentioned features, you must make sure that the *PaaS* environment meets your specific programming needs. Here is a list of questions you need to ask a potential *PaaS* provider:

- What development environments does the vendor provide?
- Can your existing and read-for-market applications be hosted from the *PaaS* cloud? This will prove the readiness of the provider.
- What is the security responsibility (for user authentication, data, etc.) of the provider and the consumer?
- Does the provider have the expertise and willingness to work directly with you for porting, customization, and deployment, as often as required?
- What facilities are there for logging of security events and user activity? Do they use log servers, independent of the *PaaS* platform?
- What tools are available to detect security flaws in applications? Does the provider have sound intrusion detection or prevention systems?
- What about change management? Does the provider have a robust, centralized mechanism for maintenance without incurring downtime or performance degradations?
- What are the provisions for backups and DR?
- How does the cloud vendor ensure that application flaws of one customer do not impact the shared infrastructure or someone else's data?
- Can the public *PaaS* provider offer the services within a hybrid cloud model? This will assure enhanced security and flexibility.

also some key questions to PaaS & how to choose PaaS?

EXAM PRISM

Developers are turning to PaaS to simplify application development and deployment. After all, configuring and maintaining the OS, messaging queues, activity logs, security and OS updates, and application containers running below the application is complex, expensive, and requires trained manpower. The promise of PaaS is the delivery of an application infrastructure, where the provider takes care of the underlying stack.

Out of all the applications you have to develop, some will not be suitable for *PaaS*. For example, if you need to develop and test on a mainframe or Unix system, that is not offered by *PaaS* provider, you will need to set up and use an internal environment. For other applications, you can use environments offered by *PaaS* providers.

Guidelines for Selecting a *PaaS* Provider

9.2.d

how to choose PaaS

There are two categories of *PaaS* providers. The first are large IT operating system and software vendors, who offer *PaaS* because it is a natural extension of what they have traditionally sold. The second are small, independent vendors for whom *PaaS* is the sole or key offering. Here are a few guidelines for choosing a suitable *PaaS* provider:

- **Compatibility with Other Clouds**—*PaaS* providers will claim portability to all other clouds. This cannot be true as the cloud environments are unique and standards are still evolving. However, you need to be aware of providers who claim they have everything for everyone.
- **Target Customers**—*PaaS* providers have certain target customers and architect their environment to appeal to particular group of users. For example, **Microsoft** focuses on *.NET* and *PHP* developers. **Oracle** focusses on *Java* developers. Make sure that the *PaaS* provider you select offers and supports the development environment you need.
- **Avoid Vendor Lock-in**—You must select a provider who facilitates cloud interoperability for your application. It must be easily ported to another public or hybrid cloud or even to a non-virtualized internal infrastructure. For this reason, the provider must abstract application runtime, data handling, and middleware from the underlying infrastructure.

EXAM PRISM

PaaS must also support a broad range of languages, such as Java, .NET, PHP, Ruby, and Grails. They must also offer a range of deployment options for you to choose from.

- **Platform Management**—Make sure that the *PaaS* provider can manage and maintain the environment. Many *PaaS* platforms are built using open-source projects and applications for which the provider may not have skills or control.
- **The Test of Time**—Make sure that the cloud vendor will be in business in the times to come.
- With the ephemeral nature of cloud businesses, make sure that the provider has a reliable set of partners, satisfied customers, and a sound financial base and can continue to survive even in harsh times.

Concerns with PaaS

The convenience attained with *PaaS* (as per the above points) leads to a great loss of control over the environment and security. The following are the drawbacks of using *PaaS*:

- **Lack of Visibility**—It is difficult to know if you are running in a secure, robust environment. There is no standard way to determine the patch levels, view, and analyze the activity logs, or perform a vulnerability audit on the platform. Remote tests are usually banned.
- **Portability/Interoperability with Applications On Another Cloud**—Unlike *IaaS*, where OS images can be moved between clouds, applications developed on a *PaaS* involve cloud-provider's **APIs** and customized language extensions. This makes porting of applications difficult.

- **Security**—Some *PaaS* providers include built-in security services, but the end-user has no information on the implemented security mechanisms. The customers cannot install host-level security applications for antivirus, WAF (python-based framework for configuring, compiling, and installing applications), host-based firewalls or disable services or ports.
- **Security for Development Code**—Since the development code resides on a third-party, shared infrastructure, the customers are wary of security and privacy of the code, which is the prime **Intellectual Property (IP)** for the **Independent Software Vendors (ISVs)**.

Language and *PaaS*

Select a *PaaS* provider with the right type of orientation and support for various software languages that will help to build a flexible and portable application:

- **Programming-Language Specific PaaS**—These were common in the early 2000s but are now less in number. They support one language, be it **Java, Ruby on Rails, Python or .NET** or any another language. The advantages they provide are the use of language-specific tools, customized support, extensive libraries, and specific code to run compute-intensive tasks in the clouds. It allows scientists and analysts to access extensive compute requirements they are comfortable with and need from, within a development and test environment.
- **Language-Agnostic Providers**—They were designed for developers working with multiple programming languages, databases, and frameworks. For example, they offer support for various languages, such as **Python, Java, .NET,** and **Ruby**, and databases, such as **MS SQL, MySQL, Postgres,** and **MongoDB**. Potential drawbacks are that you may not find the same degree of language-specific customizations or libraries to distribute your code to several other clouds.
- **Hybrid PaaS**—It is a cloud service that gives you the freedom to work with a complex mix of off-premises and on-premises applications and data. They abstract the infrastructure layer and on-premise resources. For example, if you have a large, on-premise **Oracle** database that you do not want to replicate to a public cloud for its size, security or cost reasons, you can access it from applications in a hybrid *PaaS*. However, a potential drawback of hybrid *PaaS* is the need to set up and manage the hardware abstraction layer, which requires in-house expertise and management efforts.

The *PaaS* market is fast maturing with several commercial *PaaS* providers. Your selection would depend on your need for a single language, mix of stack component and several languages or a *PaaS* that would allow you to access in-house data or applications.

Software as a Service (SaaS)

SaaS is the capability to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a Web browser. As in the case of *PaaS*, the customer does not manage or control the underlying cloud infrastructure (servers, storage,

OS, network) or the application features. The customer can; however, configure user-specific application parameters and settings.

EXAM PRISM

In the SaaS cloud, the vendor supplies the hardware infrastructure, software, and applications. The customer interacts with the application through a portal. As the service provider hosts the application as well as stores the user data, the end user is free to use the service from anywhere.

SaaS provides several benefits. Users get to use the application over the Internet without the onus of buying, implementing or managing the software. Like *IaaS* and *PaaS*, the fee is priced on the usage-basis, whereby customers purchase rights to use certain or all modules as required. Although *SaaS* and *ASP* may appear to be the same, they are different. **Table 3** lists the differences between *ASP* and *SaaS*:

Table 3: Differences between Application Service Provider (ASP) and SaaS

Feature	ASP	SaaS
Ownership	ASP applications are usually single-tenant with client-server architecture hosted by a third-party with an HTML-front end to make it easy for remote use.	Multi-tenant, application hosted by the application developer, with regular updates directly from the developer.
Infrastructure	May be a non-virtualized environment with direct attached storage; with server and storage dedicated to the application.	Shared, virtualized servers, network and storage systems form a resource pool; server and storage are shared with other services.
Web-based	Not originally written to be Web-based and used over the Internet; hence, there is performance degradation.	Built to be Web-based and used over the public Internet.

There are a large number of *SaaS* providers, such as **Microsoft LiveCRM, Google Apps, Trend Micro, Symantec,** and **Zoho.** In September 2007, SAP launched ***Business ByDesign*** – an online version of ERP service targeted at small and medium-sized businesses who do not want to invest in a large IT deployment. It enables a preconfigured suite for managing financials, customer relationships, HR, projects, procurement and supply chain. About a year later, in October 2008, **Infor** (based in Alpharetta, Georgia) entered the cloud market with the launch of a *SaaS* version of *ERP* ***SyteLine***, a very unique offering that allows the user to move seamlessly between on-premises deployment and public cloud-based *SaaS* or vice-versa.

Microsoft Dynamics entered the *SaaS* market in 2007 with the introduction of ***CRMLive***, which is run at **Microsoft** datacenters around the world, along with all the other "Live" products, such as ***Live Small Business Office***. Software-plus-services for ***Microsoft Dynamics ERP*** is the new capability being offered. It allows the user to implement the ***Microsoft Dynamics*** software as a wholly-owned on-site solution, or via **Microsoft** online services, or as a combination. **Oracle** entered the market in 2009 with the introduction of services comprising of ***Oracle Sourcing*** and ***Oracle Sourcing Optimization*** products. In addition, **Oracle** also offers ***CRMOnDemand***, a CRM as a *SaaS*.

what to look for in SaaS

But *SaaS* needs cautious thinking and a well-planned deployment. There must be a tested confidence that organizational data is secure in the remote site. The issue of confidentiality, integrity, and service availability has to be addressed by the provider. Authentication to the application needs to be tightly protected using tested **Identity Access Management (IdAM)** applications. Here is a list of questions you need to ask your *SaaS* provider:

- How does the provider make sure that the users who sign up are not fraudsters and will not start malicious activity?
- How and to what extent is security integrated with the SDLC at different phases, such as architecture, coding, testing, and deployment?
- What are the design and coding standards?
- What Web security standards are being followed?
- How is customers' data protected from attacks by other tenants?
- How is data at-rest and in-motion protected from other application users in the cloud?

Case Study 2: Employee and HR Information Processing in the Cloud

A customer wants to migrate all employee data and related HR processing (such as payroll, tax calculations, vacation approvals, and retirement contribution) to a public cloud service provider. Traditionally, they used distributed environment with internal application servers, database servers with SQL database, customized HR applications, and Storage Area Network (SAN) based storage. The internal immediate needs were for more compute and storage resources. Data replication for Business Continuity Planning (BCP) would be an enormous capital expense.

The customer selected an *IaaS* provider and deployed *VMs* and external storage (See Figure 1). The HR application used 3 *VMs*, running the *Redhat Linux* operating system. The applications are provided by an *ISV* but had been internally customized. The *SQL* database is hosted on a fourth *VM*. Alternatively, the customer could have used a database-as-a-service but that would have required rewriting and porting the application to the cloud provider's database. The HR data is maintained on a cloud storage device offered by the cloud service provider. Essential security requirements are:

1. The company must make sure that the HR applications are accessed by authorized users only. It must also implement Role-Based Access Control (RBAC) so users are limited to certain privileges related to their business role and responsibility.
2. Similarly, the customer must ensure that if a *VM* is shut down or deleted, it has the option to either move the data to another *VM* or delete all the data related to services on the *VM*.

Figure 1 shows the use of VMs for hosting applications and databases at a datacenter:

case study: if company wants to move to cloud?
- decide IaaS, how many VM and for what?
- which aspect of business on cloud Vs local?
- which CSP? cost and benefits of CSP?
- security concerns?

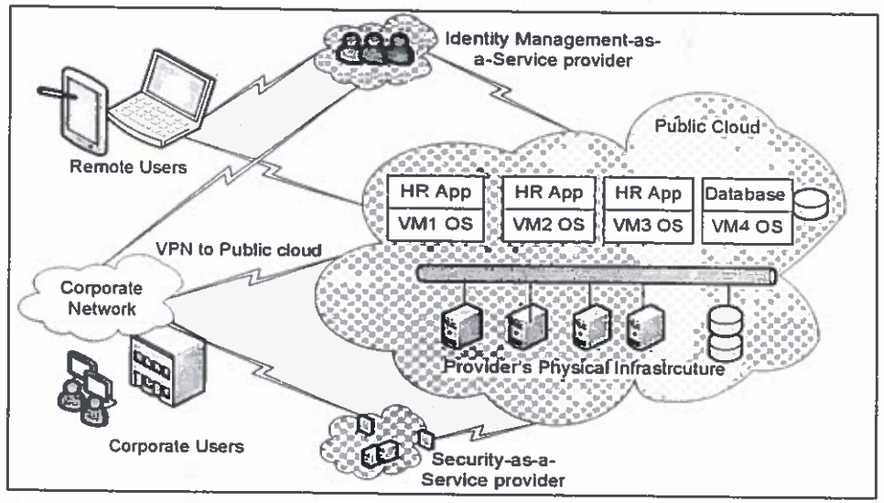

Figure 1: HR Applications and Database Hosted on VMs at a Public Cloud Provider's Datacenter

The advantages include scalability to increase the compute resources so as to meet peak load requirement and reduce them during low-use periods to save on operational expenses. The customization includes subscribing to backup and DR services. This eliminates the need for imminent and substantial capital expense for data replication to a remote datacenter. The customer created a set of best practices that are meant for pending projects to use a public cloud:

- Determine which aspects of your business (backups, replications, infrastructure applications, ERP, emails, software development) could be moved to the cloud.
- Prepare a tender or RFP and send it to several cloud service providers. Compare technical strengths, offerings and prices of the providers.
- Research the cloud market and compare available options. Contact customer references as part of the research.
- Devise a cost-effective, non-disruptive, phased project plan to migrate applications to the cloud.
- Negotiate and execute a primary contract, a service level agreement, and a maintenance contract.
- Keep a period aside for overlap during which your internal, traditional infrastructure is available when a need arises.

Case Study 3: Use of a Public Cloud for Backups and DR

A pharmaceutical company needs to implement DR. There are several government regulations that it must abide by. The first requirement is that data must be within the country where it operates. The DNA sequencing research division has a huge data repository and regularly executes large, computer-intensive batch jobs. The data needs to be backed up. Several versions of the data must be kept for comparative and trend analysis to determine and understand the biological development of

humans as well as disease-causing bacteria and viruses. The second requirement is confidentiality of data. All pharmaceutical **Intellectual Property (IP)** and early-stage research data on upcoming medicines and antibiotics are online. They need to be protected from threats and hacking. The third requirement is the need to keep the investment and capital expenses low and within the allocated budget. The company evaluated several options:

The first option was to select a remote datacenter site, establish a datacenter, procure IT equipment for backups and data storage and start data replication from the production site. However, this required large initial expenses for the datacenter, power, cooling, physical security, racks, backup equipment, storage arrays, network security, etc. The operating expenses of maintenance, monitoring, security personnel, IT administrators, and annual support costs for software and hardware were extremely high.

The second option was to lease rack space at a host service provider datacenter. The provider would take care of the space, cooling, network bandwidth, and physical security for a monthly fee. But the company would still have to purchase and manage the servers, storage, backup tape libraries, and data replication software. In addition, the storage growth is unpredictable, as it depends on the research activity and number of drug tests the company has in the pipeline. It is not possible to invest in the IT hardware, as it is difficult to forecast the future storage spikes.

The third option was to use an *IaaS* cloud provider. This took care of removing initial capital expenses but several other issues, such as compliance, cooperation during audits, data confidentiality, and probable longevity of the provider, could not have been taken lightly. In order to comply with government regulations on data and IP protection, the company also needed to make sure that the data is stored within the country.

After diligent deliberations, the company chose the third option, initially motivated by economic reasons. The company decided on the deployment roadmap, which was divided into various phases. **Table 4** lists these phases along with their brief description:

Table 4: Migration to a Cloud-based DR Site

	Phases	Description
1	Cloud service provider selection	This stage is probably the most important. You need to ask if the provider throttles the speed of uploads. Most Internet service providers provide a higher speed for downloads than for uploads. Also, check if your cloud provider blocks certain file types (such as videos, compressed files, database files, and OS files) from being backed up. If the provider has free trails, take advantage of it. They have an option of using a full-service DR provider with backup and recovery assets as standby at the DR site.
2	Backup and restore for non-critical data	Test if the cloud provider is suitable for you by backing up non-critical data. Try file-level, partial, and full restores. Also, see if you can make application-level backups, such as for databases or messaging. Try bare-metal restores of the OS.

Table 4: Migration to a Cloud-based DR Site

	Phases	Description
3	Asynchronous Data Replication to Meet RPO requirements	Set up scripts to automatically backup incremental updates to the cloud. It is advisable that your traditional in-house backup infrastructure should not be removed. Some cloud providers do not allow several application-level or snapshot-based backups.
4	DR Tests for Applications	Once you are able to backup most file types, you must test file-level, application-level, and bare-metal restores. Also, test the time it takes for services to be restored after retrieving the applications and associated user data from the cloud to your corporate datacenter. All relevant documentation must be updated to reflect the new DR plans. Perform a dry run of the DR test every 3 months.

Over time and with due diligence and planning, backup and DR has become a valuable cloud use case for the pharmaceutical company. The cloud has solved the problem of having one or more remote sites with a data copy. The cloud vendor maintains at least two data copies at different sites. It has proven to be a cost-effective way to have a backup of all critical data, applications, and a few operating systems supported by the provider. The cloud data is easily accessible for restoration from any site that the customer selects, to use it as an alternate location to run his/her services. The pharmaceutical company has two options to create a back up of data:

- **Cloud Backup Services**—It can replicate its data to a provider's datacenter and configure servers for use in the event of a disaster.

  The second option is to take service of a full-service DR provider and pay fee each month to them to manage the replications and maintain warm-standby servers for use in the event of a disaster. The customer has found several public DR-as-a-service providers:

  Double-Take Cloud Services from Vision Solutions (http://www.visionsolutions.com/)—System state, applications, and data at a customer site are replicated at a byte-level to a cloud-based repository. If the primary site is down, *Double-Take Cloud* creates servers on the basis of the information in the repository. These servers have identities that the end-users are familiar with.

- **EVault Services (http://www.evault.com/ from Seagate)**—It enables you to efficiently backup your physical or virtual systems and data and extend IT infrastructure to the cloud.

- **Sungard's Availability Services (http://www.sungardas.com/)**—It offers multi-site facilities and infrastructure to mitigate disasters. They have a Secure2Disk solution that is an online, disk-based service for rapid backups, increased reliability, and shorter recovery time objective (RTO).

There are numerous benefits of a cloud DR, which are given as follows:

- No upfront investment on deployment
- Access to seemingly-unlimited amount of resources
- Pay for what you use and predictable operating costs
- Reliable, multi-location replication

- Higher availability and uptime
- Available uniformly to users in all geographical locations
- Seamless upgrade to latest versions for all users
- Compliance to all regulatory requirements

Backups, DR, business continuity, and content storage are among the top use cases for cloud computing today.

EXAM PRISM

*For backup and DR in the cloud, it is beneficial to use **de-duplication**, **compression**, and **encryption** at source. These three will reduce expenses by reducing bandwidth and storage utilization. Besides, it will improve performance and security.*

Database as a Service (DBaaS)

Databases are an essential component for all enterprise applications. Database services for on-premise applications are configured using internal, purchased servers and database. Large organizations have a shared infrastructure for internal teams, where the database may be a shared service and used for different applications. Thus, several applications can simultaneously access a single database running on clustered servers and centralized SAN or NAS storage. The applications are; however, isolated from each other. Organizations can use a shared database on a private or public cloud. In either case, the data for each application is explicitly protected by the centralized service called *Database-as-a-Service (DBaaS)*.

Cloud providers, especially *PaaS*, offer a database based on commercial products, such as *PostgreSQL*, *MySQL*, *Oracle* or *Microsoft SQL*. Nowadays, there are several available database service providers. Here are a few factors to consider before selecting one for your requirement:

- **Research**—Find the available options and scrutinize the SLA, the offered services, commitments and support.
- **Estimate Your Needs**—Calculate what you will really need. It is expensive to add resources incrementally in real-time to meet load spikes.
- **Vendor and Community Support**—Ask the cloud provider for references and take time to speak with them. Check if the provider has a capable team, which is willing to help you with technical issues on a 24/7 basis. Make sure that the database has support from an active user community and user forums.
- **API Support for Databases**—Make sure that your provider supports and that you use APIs that will work on other clouds. This lets you focus on your application without worrying about infrastructure and compatibility.
- **Price**—Cost for cloud databases run from free versions to variable pay-per-use to fixed monthly fee option. Open-source solutions are inexpensive (and have good community support).

- **Tuned for the Cloud**—Make sure that the database configuration, scalability, reliability, and performance are all tuned to work in a cloud environment where it may be subjected to immense load spikes and large number of concurrent users.
- **Compatibility with On-Premise Databases**—Vendors offer public cloud database that is different from their on-premise edition. Understand the differences and try to use features that are supported on-premise to ease your migration to a hybrid cloud when required.

Sharding a database helps in improving performance. It is a process of breaking a large database into a number of smaller databases that are located on one or more servers.

EXAM PRISM

*Sharding improves performance, especially for high-transaction, large database-centric business applications. In a cloud environment, **Sharding** reduces the index size and helps in improving search performance.*

EXAM PRISM

DBaaS service providers offer sharded environments in a cloud by horizontally scaling up server resources. Another common practice in DBaaS is database profiling. It profiles and analyzes source databases and examines the data quality in them for inconsistencies in structure, content, relationships, and derivation rules.

The goal of database profiling is to highlight any predictable issues that may arise. This helps in avoiding any erratic problems that may come up during implementation or run-time.

Specialized Cloud Services

There are other specialized cloud services, such as Distributed Computing as a Service (DCaaS) and Parallelism as a Service. *Distributed Computing as a Service (DCaaS)* is a method of performing a single, common job by engaging various geographically-dispersed resources. It is a Web-based service that makes it easy and convenient to process massive amounts of data and perform compute-intensive tasks. The service check for available resources across the network splits a job to run on different servers and later collects and assembles the results. It helps reduce processing time, improves the performance of the system, allows use of multiple resources and reduces the cost of computation.

In distributed computing environment, each processor or server has its own private memory for its work. Information is exchanged by passing messages between the processors. Examples include wireless-sensor network, telephone networks, network file systems, distributed databases, etc. On the other hand, in *Parallelism as a Service*, all system and user processors have access to a shared memory space to enable efficient inter-process communication. It improves performance by eliminating or reducing communication. Examples include cluster computing and volunteer computing. *Parallelism as a Service* enables multiple tasks or actions to be performed simultaneously. A cloud is a convenient platform for *Parallelism as a Service*, as it made up of various virtual resources that can do multiple actions or run multiple applications in parallel. In a cloud, you can leverage its multiple resources to execute and process multiple tasks simultaneously, without any variation, instability or interruption in the speed or quality of the service.