Realtime publishers

The Definitive Guide™ To

Cloud Computing

sponsored by

IBM®

Dan Sullivan

Copyright Statement

Chapter 5: Strategies for Moving to the Cloud

Cloud computing is a framework for delivering services that, as we have seen in previous chapters, offers a number of compelling benefits. Now it is time to turn our attention to strategies for moving an organization from thinking about cloud computing to using cloud computing. Many of the same rational methods and management techniques we use in IT planning and deliver today are relevant to cloud computing. This is not surprising. As I have noted in this book, cloud computing is a phase in the evolution of IT services delivery; it builds on previous practices to deliver new levels of efficiency, control, and manageability.

This chapter focuses on how to plan for the organizational and technical issues around the move to cloud computing. It is specifically structured around three broad topics:

- Planning principles
- Architectural principles
- Use case scenarios

The first section on planning principles will describe a process for understanding the current state of IT services and framing them in such a way that we can properly start delivering these services in a cloud-based environment. In the second section on architectural principles, we examine issues such as scalability, manageability, and service delivery in terms of design and implementation issues. High-level discussions about planning and architecture in the first two sections of this chapter are complemented by a set of use case scenarios in the third section of this chapter. The goal of the use cases is to provide concrete examples of applying the planning and architectural principles to typical scenarios facing cloud computing adopters.

Planning Principles for Moving to Cloud Computing

Planning a move to cloud computing starts pretty much the same as any other planning process: understanding where you are and where you are trying to go. In the realm of IT, this generally means understanding the business drivers that dictate the services to be delivered, the expectations for those services, and the constraints on actually delivering them. From there, we can move to a detailed definition of requirements. With a clear and well-defined set of requirements, we can document workloads that we expect to utilize the cloud. Each of these steps will be considered in turn.

Prioritizing According to Business Drivers

Business drivers are the strategic objectives of an organization that frame the need for IT services. These can include:

- Increasing productivity
- Reducing time to market in new product development
- Reducing production costs
- Optimizing product distribution and delivery
- Increasing market share
- Increasing customer retention

Business drivers are so high level that they can apply to many different businesses. This is expected because businesses all have the same high-level goals of maximizing returns for owners.

What distinguishes businesses in terms of strategies is how they prioritize these objectives and how they define and implement strategies to realize their goals. For example, one company may decide to focus on increasing productivity in order to remain competitive in an increasingly global market. Another company may realize that it is their intellectual property (IP) that drives their growth, and they need to invest more in computational resources to develop new IP. Still another company operating in a mature market may decide to grow by acquiring new customers by targeting perceived weaknesses in their competitor's product line.

The first step in planning a move to the cloud, then, is understanding what business objective is served by that move. Certainly, moving to cloud computing because it is a more efficient vehicle for delivering computing services is a sound reason. We do not need to settle for just that, though. If we press for an even more detailed set of drivers, we can more precisely plan our cloud services. This will help us to plan for short-term capacity demands, plan for long-term needs, as well as deploy needed applications and other software to support those objectives.

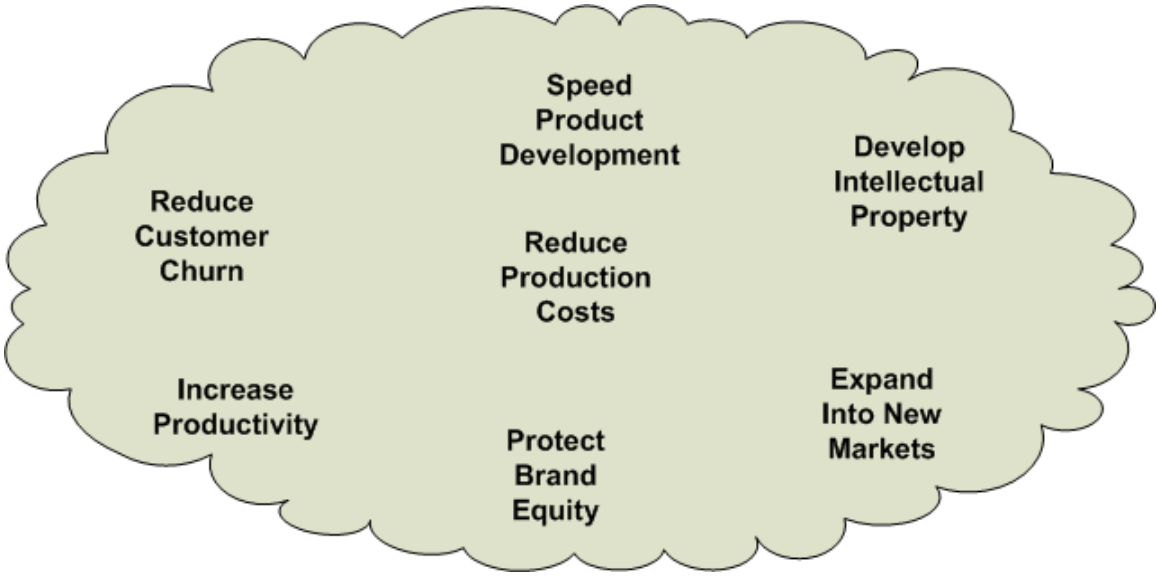

Figure 5.1: High-level, non-prioritized business objectives are less helpful in shaping cloud computing planning than more precise, prioritized objectives.

Defining Requirements

Defining requirements that will drive a cloud computing adoption can be a daunting task. It is difficult enough to elucidate and define requirements for one application let alone gathering requirements for multiple applications serving different business needs and managed by a range of departments. Fortunately, we are not starting from scratch. Applications, documentation, policies, and operational procedures are probably already in place. Our job is then one of understanding the details of existing systems because these reflect, at least to some degree, the current application requirements. We can then build on this by assessing additional requirements going forward.

Existing Applications Infrastructure: The Current State of Affairs

An inventory of existing applications and workloads is a valuable asset for planning a move to the cloud. An inventory should include all applications that might migrate to the cloud. Implementation details will vary from one application to another (even among the same software used by different departments or for different business purposes), so it is important to include in the inventory key information in three areas:

- Business requirements and related details
- Technical and implementation requirements
- Operational details and requirements

Business requirements specify who within the organization is responsible for a service, how critical that service is to the business, and what strategic objective is served by the application. These requirements are not necessarily long, detailed documents; a simple half-page summary is probably enough. Our goal is not to create an encyclopedic resource on every application in the organization but to create a planning tool that highlights the services that will run in the cloud and identify the core requirements for those services.

The technical details catalog some of the implementation details about existing services. This includes details such as:

- Server configuration
- Workloads on servers
- Dependencies and interoperability considerations
- Use of shared resources, such as disk arrays

All the necessary details about existing services may be documented in a form like that shown in Table 5.1.

Type	Requirement Area	Description
Business	Service Description	A high-level description of the service
	Business Owner	Person or department that funds and governs the IT service
	Service Level Agreements	Key requirements on service delivery
	Business Objective	Describes the strategic business objective that is served by this IT service
	Criticality	Ranking of relative importance of this service.
Technical	Servers	List of servers and description of configuration; role of each server
	Shared resources used	Shared IT resources, such as disk arrays, network, backup services
	Platform Services	Operating system required, libraries, utilities and other packages required to run the applications
	Applications	Commercial, open source and custom applications
	Physical distribution of servers	Location of primary servers, backup servers and disaster recovery sites
	Utilization	Description of server, disk array, network utilization.
	Peak Periods	Times and duration of peak loads, frequency of peak periods, periodicity of peak demands
	Dependency on other Services	Other IT services that are required to deliver this service
Operations	Backup requirements	Recovery point objectives, recovery time objectives, etc.
	Disaster recovery	Time to recover services, level of services to be restored, critical dependencies
	Compliance issues	Summary of key compliance and governance issues with this service

Table 5.1: Requirement categories for summarizing existing applications, software stacks, servers, and related hardware

Additional Requirements for New Applications

If there is one thing we can count on with IT services, it is that requirements will change. A move to the cloud will open new opportunities to deploy additional services, change the way services are consumed, and consolidate resources. These should also be captured during the requirements-gathering stage. We certainly want to capture applications and workloads that fall into the "more of the same category" (for example, more departments will stand up small databases because the overhead with managing them is reduced) but the most interesting, and perhaps the most influential in the long term, are those that change the way we do business. Consider examples such as:

- Using cloud storage to store single copies of data that are accessed by multiple applications rather than duplicating data sets
- Reducing the number of ad hoc reporting tools as users standardized on the "best of the breed" tools offered in the cloud's service catalog
- New applications, such as statistical analysis and data mining of large customer transaction data sets enabled by on-demand access to compute and storage resources

In the best cases, we will be able to devise reasonable estimates on compute and storage impact of some of these new requirements. For example, in the case of reducing duplicate data for business intelligence applications, we can develop fairly accurate estimates. The more innovative applications, such as advanced analytics, are more difficult to pin down. The CPU demands of such applications are highly dependent on the type of analysis, the algorithms used, the implementation of the algorithms, and the amount of data we are analyzing. Even with these limitations, we can at least provide best estimates (sometimes guesses) for these new types of applications. The next step in the planning process after prioritizing business drivers and defining known and estimated requirements is to analyze the potential workload for the cloud.

Assessing Workloads

Workloads are as varied as business requirements. Some workloads place a heavy load on CPUs while others are more I/O intensive. Sometimes workloads are fairly consistent over time and others have well-defined peak demand periods. It is important to understand workload profiles for a few reasons.

Capacity Planning

First, it helps to estimate the overall capacity of cloud services the business will consume. This is especially important if you are implementing a private cloud and want to ensure adequate capacity for peak demand periods. Public cloud customers will also find this data useful for budgeting and long-term planning although there is no need to be concerned about the hardware capacity of your provider (at least in theory). For hybrid cloud configurations, this type of detail can help you understand when internal capacity will be exceeded and public cloud resources will be required.

Scheduling

Another reason to assess workloads is for scheduling purposes. Some jobs have fairly predictable workloads. For example, services provided to the customers through Web applications will have generated historical data that can be used to determine demand patterns. These applications may have minor periodic variations, for example, Mondays have heavier workloads than Fridays, or longer, seasonal variations such as those retailers experience just before the Christmas holiday.

Cloud providers can use knowledge of workloads to optimize scheduling. Ideally, at any time, we would have a mix of jobs that have different levels of demand on CPU, I/O, and networking. We would not want, for example, to have all the I/O and CPU intensive extraction, transformation, and load (ETL) processes running at one time. Depending on the level of control one has over the workload scheduling, a cloud provider can schedule jobs in an optimal manner or use variations in pricing schedules to provide incentives for users to schedule their jobs in ways that coincide with the scheduling goals of the provider.

One way to globally optimize scheduling is with a bid/accept model for pricing. Cloud consumers can bid a price for a server or CPU time based on the value of having a particular job run. If it is a high-priority job, the customer will bid a higher price; if the job can wait, the customer will bid less. This approach will optimize the allocation of resources in the way a free market optimally allocates resources. This model, however, is subject to the same limitations as free markets; the model breaks down when there is, for example, insufficient information or time to fully evaluate options.

Cost Recovery

Public cloud providers set their rates to cover costs and earn a profit. The IT department, or other organization structure charged with providing private cloud services, will likely charge for services provided as well. Internal service providers generally are more concerned with recovery costs than making a profit, and a shared cost model is a common means for charging for these services. Charges are based on a simple formula:

(Total Cost of Providing Service / Number of Units Consumed) = Cost Per Unit

Units of service can be CPU hours, server hours, or gigabytes of storage per month. Basically the idea is that the service providers recover whatever the cost of providing a service.

> **Note**
>
> This is different from a simple market model in which price is determined by supply and demand. In the case of a cost recovery model, when demand goes down, price per unit could actually go up because the number of units consumed goes down. Conventional free market economics predicts the price will drop in such situations.

The mix of workloads and their distribution over time are important factors when aligning requirements to the cloud model.

Aligning Requirements to Cloud Services

At the end of the planning phase, we should have:

- A set of high-level requirements for existing applications that will move to the cloud described in terms of business, technical, and operational requirements
- Rough estimates for new applications enabled by the cloud
- Workload information that can provide the basis for capacity planning, scheduling, and cost recovery

To ensure a cloud service meets the expected needs, we want to have sufficient capacity. How we do so will depend on whether we are using public, private, or hybrid cloud services. When a private or hybrid cloud model is used, we are both the provider (for some of the services in the hybrid case) and the consumer. As the provider of cloud services, we have to redeploy existing hardware and/or procure additional hardware and deploy it in a cloud infrastructure along with management applications and a service catalog of machine images and related software. When a public cloud provider is used, we have to demonstrate the provider can offer the levels of service needed at the times they are required. As we get into these issues, we move away from the planning aspects and start to focus on more architecture-oriented issues related to moving to the cloud.

Architectural Principles for Cloud Services

The architectural principles underlying the cloud model are designed to maximize the utility of computing infrastructure by making it available to a broad range of users for a variety of applications without unnecessarily coupling hardware and software to single uses. To do so, we design around a number or architectural principles focused on:

- Designing for scalability
- Designing for manageability
- Deploying layered technical services
- Delivering business services

Before discussing each of these in detail, it is worth noting the importance of virtualization to cloud architectures. Virtualization is a fundamental aspect of cloud computing and is used at numerous levels of service delivery. We virtualize computing and storage, which hides the implementation details of these low-level services. Higher-level services, such as database management, content management, and identity management, are provided as services abstracted away from implementation details.

An immediate benefit of virtualization is flexibility. Hardware can run different operating systems (OSs) at different times. Different software stacks can be deployed to run for some period of time and then shut down. Legions of IT professionals are not needed to do this; virtualization enables greater levels of self service than have been possible in the past.

The degree of flexibility, and the benefits derived from it, varies with the amount and method of virtualization. For example, at one end of the spectrum, we can deploy single servers dedicated to single tasks. If additional resources are needed to accommodate growing workloads, either the server needs to be upgraded or additional servers need to be dedicated to that purpose. This is especially costly if the additional resource requirements are only for short peak demand periods.

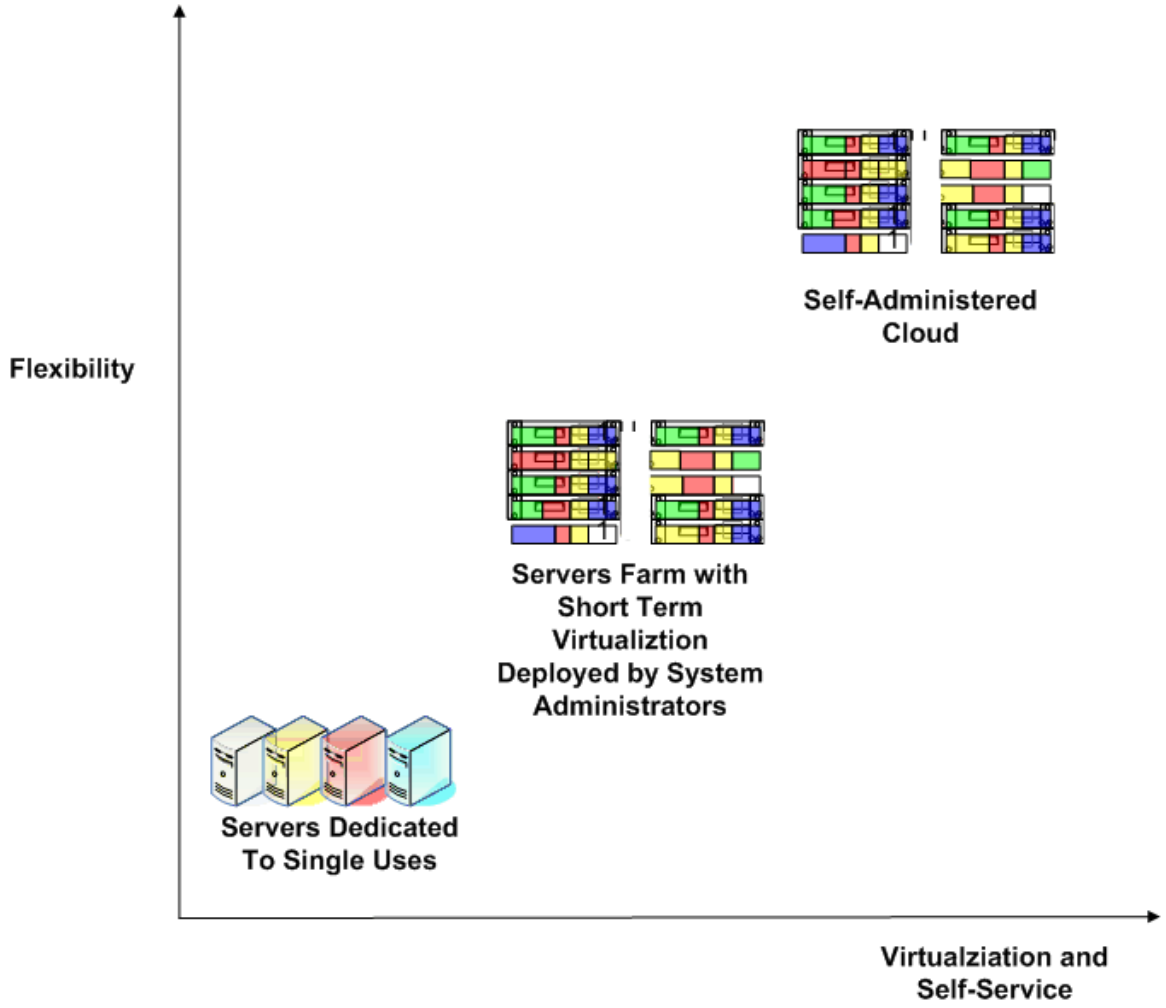

Figure 5.2: The greater the virtualization and support for self-administration, the greater the flexibility in adapting computing resources to changing service needs.

A step away from the dedicated server model toward a highly virtualized environment like the cloud is a server farm in which servers are reallocated according to changing needs. There are a number of advantages of this approach over the dedicated server model. First, policies and procedures are in place to change the roles of servers fairly rapidly. Systems administrators shut down applications and supporting software, install machine images with other applications needed at the time, and redeploy the servers in their new roles. A second advantage is that hardware is fairly easily reallocated; there is no need to procure new hardware for small, incremental increases in workloads.

Although the virtualized server farm is a step in the right direction, it is still hampered by the need for IT support to reallocate resources. This creates a certain amount of overhead cost associated with the switch. Granted, it is smaller than the cost associated with switching dedicated servers, but it is still greater than the cost associated with the self-service switching costs found in cloud environments.

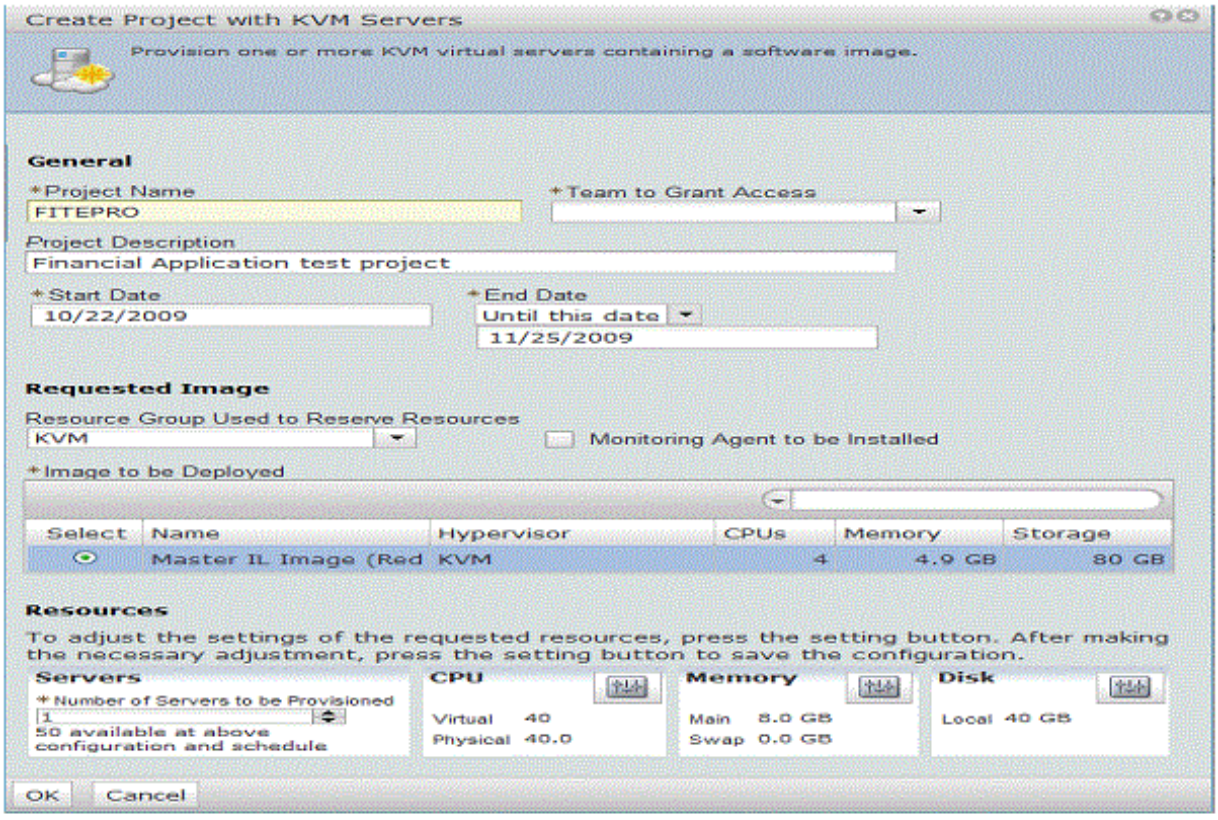

Figure 5.3: Virtualization combined with self-service administration lowers virtual machine deployment costs. Non-technical cloud consumers can manage their own workloads.

In a cloud environment, the process of deploying virtual machines is highly automated with the use of self-service software. In addition, resource tracking modules in the cloud administration software can track the images used, the time servers are up and running, and the amount of storage used by job. This can further reduce administration costs and facilitate charge backs and cost recovery. In addition to flexibility, virtualization enables critical qualities such as scalability and manageability.

Designing for Scalability

Concerns about scalability affect both cloud providers and cloud service consumers. In the case of cloud providers, the designing for scalability entails addressing several requirements for meeting varying workload demands. For cloud consumers, the issues tend to be around the question of how to most effectively utilize the computational resources available in the cloud.

Providing Scalable Computing Resources

At first glance, cloud scalability may look like just a matter of hardware. With enough physical servers, disks in storage arrays, and network bandwidth, we can meet scalability demands, right? Not exactly, or at least that is not the entire story. Cloud service providers also have to provide services and features in addition to raw hardware to enable a functional, scalable cloud. Some of these services and features include:

- Security services
- Standardized catalog of applications
- A service oriented architecture (SOA)

These requirements are comparable to those we find outside the cloud.

Security Services in the Cloud

Security in the cloud looks much like security outside a cloud environment. When we deploy applications to the cloud, we have to concern ourselves with several security requirements:

- Identity management
- Access controls
- Auditing and logging
- Vulnerability management and threat assessment

Identities are independent of workloads running in the cloud. Identities persist over time and should be maintained with authentication and authorization information as well as encryption keys. This type of information is needed, for example, to control limits on resource allocation in the cloud and to store keys used to encrypt data stored in the cloud.

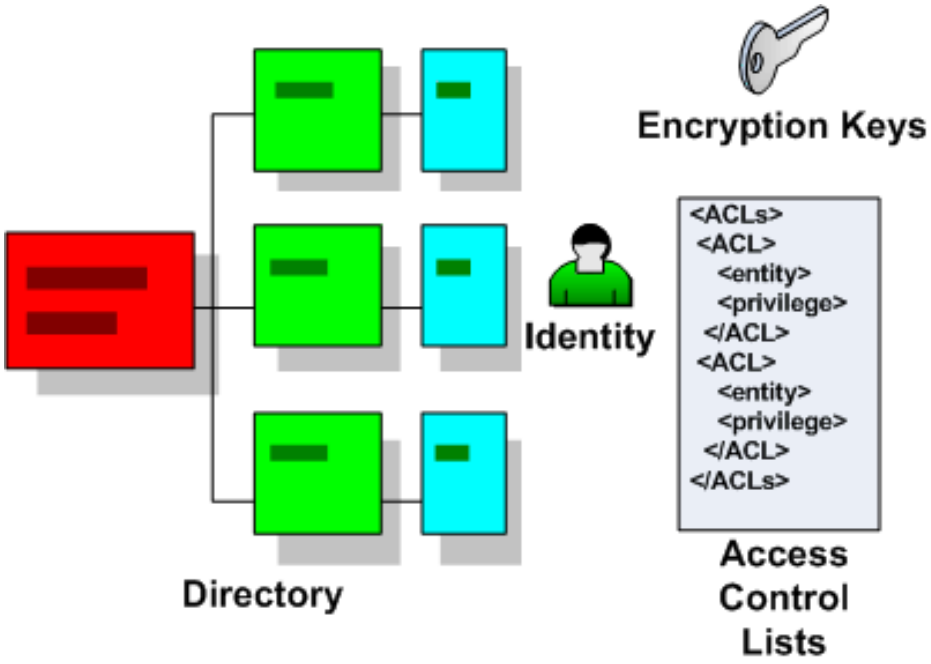

Figure 5.4: Security controls in the cloud depend on identity, access control lists, and encryption keys.

In addition to user-centric security information, cloud providers need to support process-oriented auditing and logging. As in any deployment architecture, audit and other logs must be tamper-proof and sufficiently detailed to meet security and compliance requirements.

The images that comprise the service catalog will support a wide range of OSs, utilities, libraries, and applications. These are all sufficiently complex to require regular vulnerability scanning, patching, and upgrading. Cloud providers will also need to have procedures in place to perform vulnerability scans on images, track patch levels, and update images as needed. One of the advantages of cloud architectures is that once an image is scanned or patched, every cloud user that deploys that image will have access to the latest version. There is no need to push patches to servers or desktops, verify installation, and then manually correct failed patches.

Standardized Catalog of Services

Scalability often implies repeated use of a small set of constructs. Take, for example, a cluster of computers comprising identically configured servers, distributed database running the same database management system in different sites, or even the ubiquitous desktop OS. These examples show that benefits of standardization can often outweigh the disadvantages of not having customized solutions to a particular problem.

In the cloud, standardization at the platform and application level comes with a standardized catalog of services. Cloud users can instantiate virtual machines running images from the catalog. The data we collect in the planning stages about application requirements can form the basis for building the service catalog. Cloud users are still free to bring or develop their own custom applications, but the service catalog provides a supported foundation for all cloud users. Cloud providers have to weigh the benefits of adding specialized images to the catalog against the additional overhead of managing more images.

SOA

Services in any architecture have to be sufficiently accessible to be of use; when we are working with highly-scalable architectures such as the cloud, it is even more important. In the cloud, we have the possibility of running a large number of services under varying workload conditions which are subject to different constraints. In environments such as this, there should be as few dependencies as possible between applications.

SOAs decouple services through agreed upon interfaces and message passing. This model scales to different types of services, a wide range of inputs and outputs, and can scale to a large number of services.

Scalability requires design and implementation considerations beyond those of just hardware and infrastructure. Scalability in the cloud requires providers to plan for and support security services, a standardized catalog of services, and an SOA.

Using Cloud Services in Scalable Ways

A cloud architecture is, by definition, scalable; however, to realize the full benefit of the cloud, we as cloud consumers need to use application architectures that take advantage of the cloud's underlying scalability. This requires that our applications avoid processing bottlenecks, such as a service that is provided only on a single server. As other parts of the application scale up to meet demands, that service would be bound by the constraints of the single server. Two common ways of avoiding this type of bottleneck are to distribute workloads in either a round robin manner or by partitioning workloads.

Scaling with Round Robin Load Balancing

Consider an online retailer that experiences peak demands during the holiday shopping period. The holiday season lasts several weeks, so scaling their Web site with cloud-based applications makes sense. There will be many users all accessing the Web site and most of the demands on the server will be to deliver Web pages, so the retailer will deploy multiple Web servers each hosting the same content. A load balancer receives all HTTP requests from shoppers and distributes them evenly across all the Web servers. In this way, no single server becomes a bottleneck and additional Web servers can be deployed from the cloud if needed. Furthermore, this approach provides high availability as well because the failure of any one Web server will be compensated for immediately by other servers in the cluster.

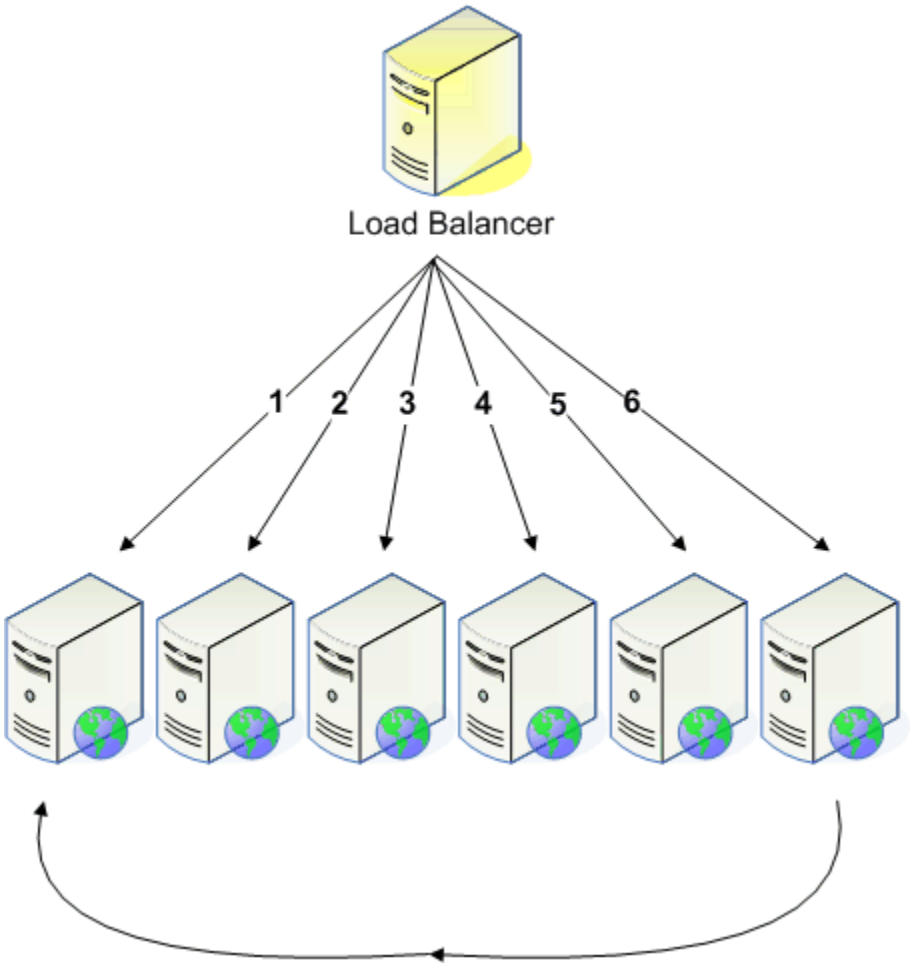

Figure 5.5: Round robin load balancing assigns each new connection or transaction to the next server in an ordered list of servers; when the last server is reached, the next connection or transaction is assigned to the first server in the list.

Partitioning by Data Characteristics

Another way to scale applications is to divide workloads by some characteristic of the data:

- Geographic location of customer
- Distribution center fulfilling an order
- Product category
- Customer name

Ideally, the criteria for dividing a workload will lead to roughly equal size partitions of the data. This helps to ensure scalability because no one server supporting a partition would become overloaded faster than the others. Also, it can help long-term maintenance if the partitioning scheme allows for changes to the partition criteria without significant overhead. For example, if a geographic partitioning scheme is used and one area grows faster than the others, one could subdivide the fast-growing geographic area into two subdivisions.

Partitioning data and storing it in different databases is sometimes used when a single database server cannot keep pace with workloads. Geographic distribution is especially helpful in localizing network traffic and improving the responsiveness of applications that run on the same local network as the database server. In the cloud, this is less of a concern at least for the cloud service consumer. Nonetheless, this type of partitioning is still useful for performance.

Databases use a combination of in-memory caches and persistent disk storage. Queries that can be answered using cached data are significantly faster than those that require disk operations. In the cloud, multiple instances of a database can run on multiple servers. Each server will maintain a cache of partitioned data and, presumably, use cloud storage for persistence. The total amount of memory available for caching is the sum of cache memory across all database servers. This can result in a much higher ratio of queries being answered from the cache rather than from disk.

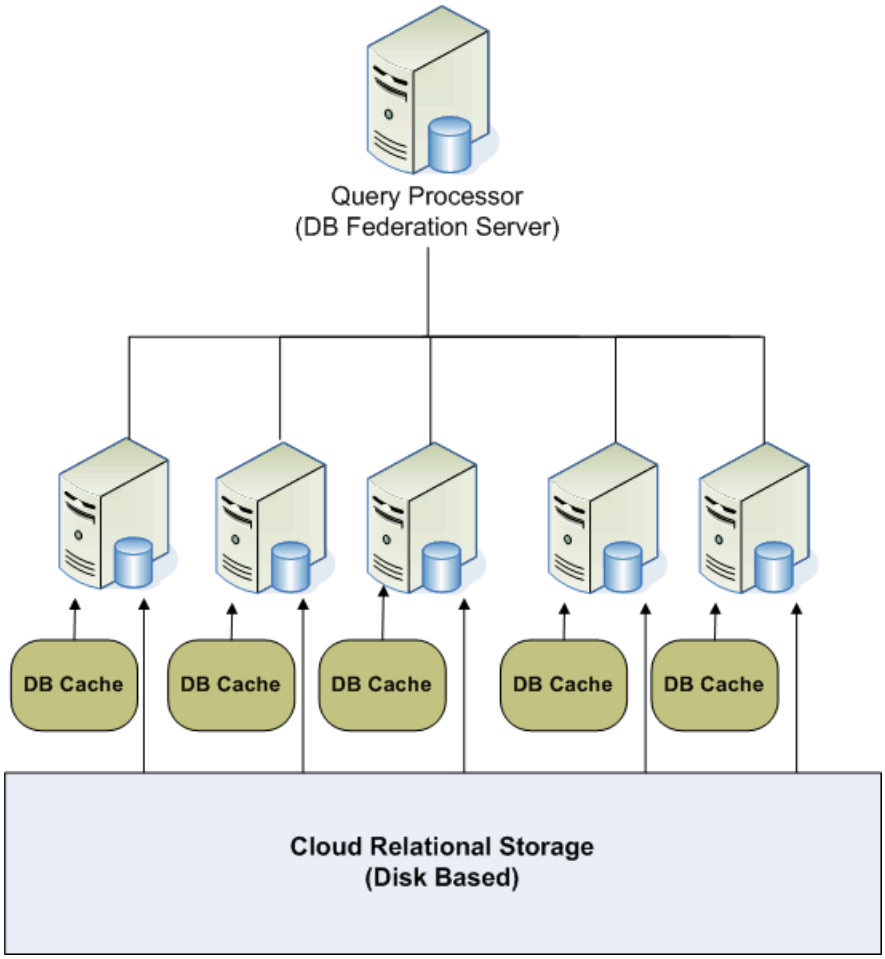

Figure 5.6: Partitioning data across multiple database servers can improve the scalability of data-intensive applications.

Designing for scalability is concern for both cloud providers and cloud consumers. Providers need to address obvious hardware and networking infrastructure issues with scalability, but those are not the only scalability issues they face. Security, a standardized catalog of applications, and an SOA are also essential for ensuring scalability. Cloud consumers also have a role in ensuring scalability by designing their applications appropriately using techniques such as round robin load balancing and data partitioning.

Designing for Manageability

Manageability is another architectural principle that strongly influences how we implement and consume cloud services. This is an important principle for both cloud providers and consumers. Three key points in this area are:

- Provisioning
- Monitoring
- Usage and accounting

The more these services can be automated, the more efficiently a cloud can deliver services to its users.

Managing Cloud Provisioning

Provisioning in the cloud is the process of instantiating one or more virtual servers running a particular machine image. In the simplest case, a user needs to start a single server, and after running a process, the user shuts down the server. This is a fairly straightforward task but still requires management software to allow non-IT personnel to manage the process. Even in a simple case, there are issues:

- Selecting a machine image to run on the virtual machine
- Determining the time to start the virtual instance
- Deploying additional applications needed to process the particular workload
- Starting services on the virtual machine
- Executing a workflow
- Shutting down the virtual server

Provisioning operations can be more complex if they involve multiple instances running different applications. For example, a workflow may require six virtual servers running a Java application server and a load balancer for distributing transactions across the six other servers. The servers may be shut down at different times as the workload varies or other application servers may be added to the set of servers to meet peak demand. Easy-to-use software is essential to low-cost provisioning.

Monitoring Jobs in the Cloud

Once servers are provisioned and jobs are running, we will need to monitor them. This includes tracking:

- CPU and memory utilization to determine whether additional resources are required or some should be shut down
- Disk I/O to ensure sufficient throughput on I/O operations to meet requirements and service level agreements (SLAs)
- Application logs to look for adverse events or warnings of potential problems
- Jobs and workflows running in the cloud, including running time, resources allocated, and costs for those resources

This type of monitoring is primarily for managing running jobs. It is also important to have management reports that summarize jobs, resources used, and costs over longer periods of time.

Usage and Accounting Reports

Usage and accounting reports are especially important for verify billing and analyzing trends in cloud usage. For providers, these reports show aggregate information about:

- Who is using cloud services
- Number of virtual servers run per job and the duration of jobs
- Machine images instantiated in the cloud
- The amount of storage in use
- The amount and type of I/O operations

Cloud users may find these reports especially useful for optimizing how they schedule jobs. Unlike running a dedicated server, there are easily controlled marginal costs associated with running jobs in the cloud. There may be cost advantages to running jobs on larger servers but running fewer instances when the pricing scheme provides such an advantage. There may be advantages to aggregating jobs and running them less frequently. This can be the case when cloud providers charge in minimum units of one hour and jobs are consistently finishing in well under one hour.

Designing for manageability means planning for end user provisioning, process monitoring, and usage and accounting reports from the start. Cloud service consumers should make use of these reports to run their jobs in the most efficient manner possible.

Deploying Layered Technical Services

Layering services is a long-standing approach to dealing with software complexity. OSs have long used layering to isolate the need to deal with hardware-specific issues or manage low-level operations, like virtual memory. Layering services is a sound approach in cloud environments as well. At the most course description, cloud services are layered as:

- Infrastructure services
- Software platforms
- Applications and information services

Infrastructure services are the lowest-level service and include virtual machines, virtualized storage, and network services. On top of this layer, we run middleware software such as relational database management systems, Java application servers, content management systems, portals, and so on. This middle tier provides the building blocks for business-specific applications such as customer relationship management (CRM) systems, business intelligence reporting systems, and customer-facing Web applications.

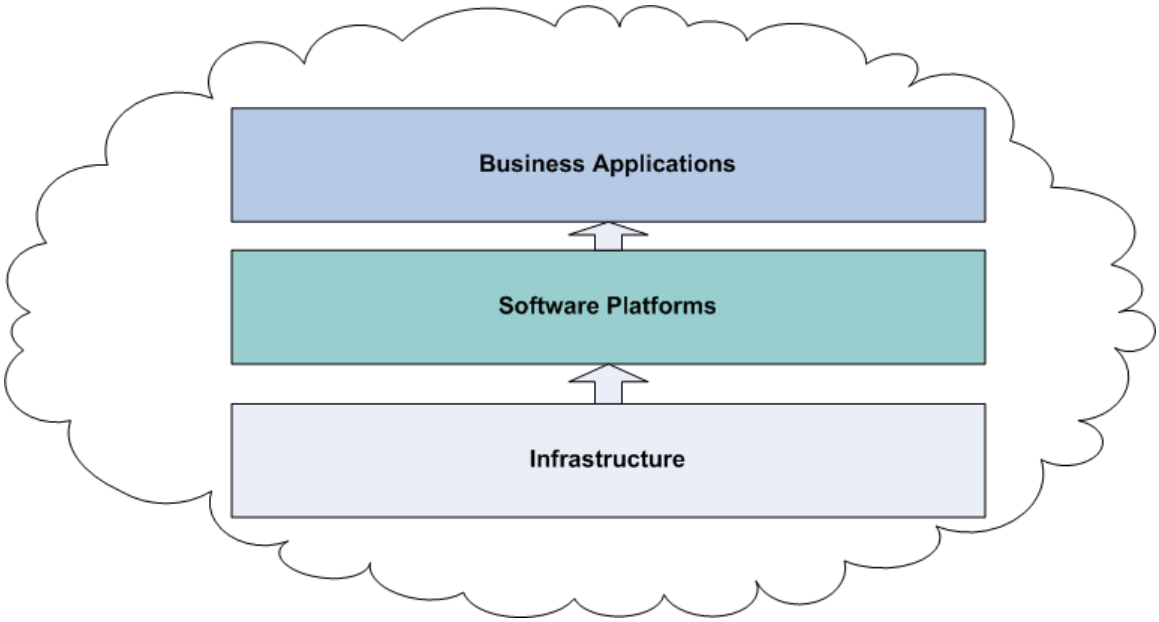

Figure 5.7: Cloud services are delivered in layers, each providing service to the layer above with the top-most layer providing end use business applications.

Delivering Business Services

Usually we would stop discussing architectural principles once we reach the top of the application stack where business services are delivered. We'll veer from the normal case here to address one other essential part of delivering and consuming cloud services: the need for managing service delivery.

The service catalog discussed earlier is part of this process. As noted, the contents of the service catalog are driven by existing and anticipated business requirements. The service catalog has its own long-term maintenance issues, just as software distributed throughout the organization. One of the advantages of the cloud is that service management is less complex. Servers are generally concentrated in the data center and there is less need for maintaining desktop clients.

Policies are needed to govern cloud operations and services to ensure their long-term stability. Basic policies, such as the following, should be in place:

- Pricing and cost recovery
- Patch management
- Security policies
- Acceptable usage
- Auditing
- Data retention

Policies define how cloud services will be governed and managed and provide the final piece of the planning processes for deploying business services in the cloud. In the next section, we will turn our attention to two use cases to provide examples of applying the planning process and architectural principles to typical business requirements.

Business Services in the Cloud: Use Case Scenarios

We will consider two use cases: a new customer service initiative and a business intelligence application. We will also examine some of the workload considerations that factor into managing cloud-based services.

New Customer Initiative Use Case

The first use case scenario is motivated by the business driver to improve customer retention. A company has been experiencing moderate but increasing turnover in the customer base; this is commonly known as *churn*. In an effort to reduce churn, the company has determined that it can gain a competitive advantage over others in the market by improving customer experience. In particular, the company has decided on a two-pronged approach. First, it will allow customers to access their entire account history rather than just the past 4 months, as currently implemented. Second, it will provide more targeted offers based on a customer's account history.

As part of the planning process, the company reviews the business, technical, and operational requirements for these services (see Table 5.1 for a list of requirement categories). The business area requirements focus on this imitative as mid-level criticality (that is, not essential for core day-to-day operations but a long-term priority).

The technical requirements include platform services such as relational database management services, customer identity management services, and access to a portal to provide presentation-level services. Estimates are compiled on the amount of data that will be stored, the number of customers querying their account histories each day, and the processing load required to update account histories on a daily basis.

Operational requirements include backup recovery and, because this is a customer-facing application, disaster recovery. Compliance requirements are minimal, but company policies protecting private customer information must be followed.

The requirements are well met by cloud architecture. Accessing entire account histories for all active customers requires the ability to rapidly scale both computing and storage resources. The incremental growth in storage required to accommodate new customer activity is also readily met by the cloud. Analyzing customer account history to generate custom offers is a compute-intensive process but will not require significant additional storage. This type of analysis will be done periodically but not more frequently than once a month. The peak CPU demands generated by this process will last for 1 to 2 days. The need for additional compute resources can be met by the cloud as well.

The service catalog already supports the middleware required, including the database, portal, and statistical analysis software. Each of these platform services is available in different images, so each will be running on one or more virtual machines. This is a customer-facing Web application, so the portal servers will be configured in a load-balanced cluster and the data will be partitioned to evenly distribute the customer database over multiple database servers.

Business Intelligence Use Case

A company has decided to consolidate its business intelligence reporting services to improve the efficiency of business intelligence operations and lower overall costs. One of the defining characteristics of business intelligence and advanced analytic operations is that they entail large amounts of data and they are computing intensive.

Traditional data warehouses and similar business intelligence architectures inefficiently allocate resources. They can be deployed around dedicated department-level servers and storage. This tends to lead to low CPU utilization between data loads and report generation. Unless there is high demand for ad hoc queries outside of data loads and report generation operations, the server runs well below capacity.

Another potential area of significant inefficiency is in storage. It can be difficult to estimate storage requirements, especially when various performance techniques, such as excessive indexing, denormalization, and materialized views, may be used to improve performance. The best combination of optimization techniques may not be discovered until the business intelligence system has been in use for some time. In a traditional deployment, that storage hardware would have been purchased already. That inconvenient fact often leads to purchasing more storage than is needed for fear of not having adequate storage.

Business intelligence as a cloud service can be implemented more efficiently. Let's assume the business drivers behind this project include improving sales by providing detailed and timely reports to sales managers while reducing the total cost of business intelligence services in the company. Technical requirements include large volumes of storage and a large number of servers to perform ETL operations to populate and update the data warehouse on a daily basis. Once the ETL process is complete, reports will be generated. Once the reports are complete, the peak demand period is over but an estimated 25% of peak computing resources will be needed during the rest of the data for ad hoc reporting.

The cloud allows this initiative to start servers as needed for the ETL and reporting operations, then scale back to a smaller number of servers. An additional benefit is that a single copy of data can be shared among different departments. For example, the marketing department and the quality control group may both want to use sales data but in different ways. In cases where each department maintains its own data mart, the sales data would be duplicated. The same data marts can run in the cloud but share a single copy of the source data.

Mixing Workloads

Jobs that do not need to run on strict time schedules can be arranged to optimize utilization. For example, loading schedules can be optimize to increase utilization by performing extraction and copy operations during times when there is a low demand on cloud resources. Similarly, workloads can be mixed so that some I/O intensive jobs are run at the same time as other CPU intensive jobs that can run at the same time as jobs with more constant and predictable workloads, such as development and test environments or collaboration services.

Both of these use cases demonstrate common characteristics of business services that fit well with the cloud model:

- Minimal or moderate security requirements
- Minimal dependencies between services
- Moderate audit requirements
- Minimal customization

As a result, these applications can meet the requirements of the business drivers that motivate their development; they can be deployed using the infrastructure, platform, and application services provided by the cloud; and they can be managed using the self-service provisioning, monitoring, and usage accounting services provided by the cloud management software.

Summary

When formulating a strategy for moving a business to adopt cloud services, we should bear in mind both business planning and architectural considerations. On the planning front, start with the business drivers and ensure that services deployed in a cloud support those drivers. To do so, be sure to analyze requirements in terms of business, technical, and operational needs. Also understand workloads and related issues, such as capacity planning, scheduling, and cost recovery.

Key architecture and design consideration also have to be taken into account by cloud service providers and cloud service consumers. Scalability is essential. Cloud service providers ensure scalability by providing sufficient hardware, software, and networking services but also by supporting security services and a standardized catalog of applications in an SOA. Manageability is also a factor in realizing scalable services, especially related to provisioning, monitoring, and usage reporting.

In the next chapter, we will delve deeper into technical and architectural issues with a look at identifying further details of cloud architectures and their impact on your business.