

Chapter 10

Cloud Computing Solution Patterns: Infrastructural Solutions

Shyam Kumar Doddavula, Ira Agrawal, and Vikas Saxena

Abstract Cloud computing is an important emerging paradigm that is affecting the way enterprise IT is being managed. Cloud computing offers several useful features like high scalability, agility through elasticity, on-demand self-service, and pay-per-use models when consuming and delivering IT capabilities. There are several scenarios in enterprise IT context where solution architecture approaches leveraging cloud computing technologies offer a better solution than established traditional options. However, the architectural design and deployment approaches for the emerging solutions based on cloud computing paradigm are different from traditional approaches, so there is a need for a new set of solution architectural patterns and best practices. There are infrastructural layer solution patterns and application layer solution patterns addressing concerns at the corresponding layers. The infrastructure layer solution patterns deal with concerns like how to architect compute infrastructure that deals with unpredictable workloads while keeping the costs down and how to architect storage infrastructure that handles storage of large volumes of data. This chapter describes a number of common infrastructure layer scenarios and use cases, the limitations of traditional solutions, and the cloud computing-based infrastructural solution patterns that a software architect can leverage.

Keywords Cloud computing • Cloud computing patterns • Cloud infrastructure • IaaS • Cloud architecture • Cloud building blocks

S.K. Doddavula (✉) • I. Agrawal • V. Saxena
Cloud Computing CoE, Infosys Labs, Infosys Ltd., Bangalore, Karnataka, India
e-mail: shyamkumar.d@gmail.com; Ira_Agrawal@infosys.com;
vikas.saxena.2006@gmail.com

10.1 Introduction

The National Institute of Standards and Technology (NIST) defines cloud computing as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [1]. NIST further defines that the cloud model is composed of the following [1]:

- Five essential characteristics, namely, “on-demand self-service,” “broad network access,” “resource pooling,” “rapid elasticity,” and “measured service”
- Three cloud service models, namely, “Software as a Service (SaaS),” “Platform as a Service (PaaS),” and “Infrastructure as a Service (IaaS)”
- Four deployment models, namely, “private cloud,” “community cloud,” “public cloud,” and “hybrid cloud”

Cloud computing technologies are causing disruptive changes in how IT resources and services across the infrastructure, platform, and application layers are getting provisioned and in the way they are getting consumed. IT consumers are looking toward a pay-per-use model in procuring computing resources to align the IT costs to consumption, thereby reducing inefficiencies. IT providers are expected to provide computing resources as services, on demand accessible from anywhere over a network. Further, these resources are expected to be scaled up and down with rapid elasticity to meet the changing needs. IT solution architects are also increasingly looking at leveraging these technologies and models while architecting solutions. This chapter also provides an overview of the key architectural building blocks of a typical cloud solution and focuses on the infrastructure layer concerns and cloud computing solutions relevant to this layer. A companion chapter titled “Cloud Computing Solution Patterns: Application and Platform Solutions” that also appears in this book focuses on the platform and application layer concerns and solutions.

Infrastructure solution architects typically face same problem scenarios [3] again and again in the context of several solutions like how to optimize server infrastructure, how to design a cost-effective and highly scalable storage layer, and how to address business continuity and disaster recovery. So, there is a need for reusable solution patterns [2, 11, 12] that describe the architectural solutions for these scenarios so that each solution does not need to reinvent the solution strategy. Patterns [13] can be defined at several levels, from business to architecture, design, and programming to deployment and runtime administration. They assist in the creation of artifacts according to best practices through all stages of the software development life cycle. Patterns with their specification and implementation help the creation of various artifacts needed for a software solution. Patterns provide a powerful way to improve IT solution development by identifying best practices and design expertise that can be captured and then made available for reuse. Some of the advantages of implementing the patterns are improved productivity, reduced development time,

minimized complexity, increased quality, improved governance, improved cost, etc. Solution patterns capture standardized, reusable solution architectures for various commonly occurring problem scenarios. This chapter describes a few commonly required infrastructural solution patterns. The chapter also provides a comparison of traditional approaches with the cloud-based solutions for the scenarios and highlights the benefits and limitations of approaches.

10.2 Cloud Computing Solution Patterns

10.2.1 Cloud Solution Architectural Building Blocks

Figure 10.1 gives a high-level view of some of the key architectural building blocks of cloud solutions.

It consists of the following layers of capabilities:

- Cloud infrastructure layer which provides the basic infrastructural capabilities
- Cloud platform layer that provides the next level specialized platforms and frameworks

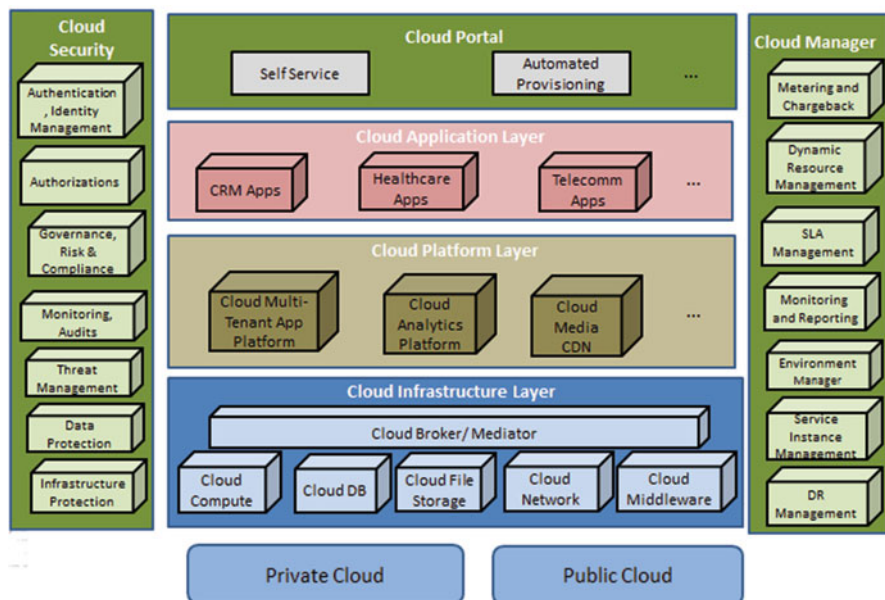


Fig. 10.1 Cloud solution architectural building blocks: high-level view

- Cloud portal, cloud manager, and cloud security layers which provide the crosscutting management and security components

A more detailed description of the various layers and building blocks is provided in the following subsections.

10.2.1.1 Cloud Infrastructure

This comprises the following infrastructural building blocks:

- *Cloud compute*: This provides a scalable and elastic compute infrastructure so that the processing layers of application can be developed over this component.
- *Cloud DB*: This provides a scalable and elastic data stores that can scale up and down to meet the application's storage needs. There are several implementation options for cloud data stores like document stores, key-value data stores, relational data stores, in-memory data grids, and graph data stores.
- *Cloud storage*: This provides a scalable file storage component that provides the foundation for several higher-level storage components. This is generally implemented as a distributed file system.
- *Cloud network*: This enables scalable network communication supporting different scenarios like Enterprise Application networks, Machine-to-Machine networks, and Compute grid networks.
- *Cloud middleware*: This provides messaging and other such integration enabler components which the layers above use for inter-component and inter-application collaborations.
- *Cloud broker/mediator*: This provides abstractions and inter-cloud integration components that enable easy migration of next set of layers across the various cloud implementations.

10.2.1.2 Cloud Platform

This comprises the platform building blocks like:

- *Cloud multi-tenant application platform*: This provides a scalable and elastic application development platform with features like dynamic auto-scaling application runtime environment and inbuilt multi-tenancy support that enables building scalable cloud applications without having to deal with such common concerns.
- *Cloud analytics platform*: This provides a scalable and elastic analytics platform that enables handle "Big data" addressing volume, velocity, and variety-related complexities.
- *Cloud media content delivery network (Cloud CDN)*: This provides a scalable media content distribution platform leveraging the cloud storage and network infrastructure components.

10.2.1.3 Cloud Portal

This provides self-service and automated provisioning capabilities. This component enables cloud consumers to request and get the various cloud-based infrastructure, platform, and application components provisioned on demand. It also enables administration-, monitoring-, and management-related interactions.

10.2.1.4 Cloud Manager

This provides the various management capabilities addressing operational and support concerns like dynamic resource management for auto-scaling, service SLA management, metering and chargeback, monitoring and reports, instance management, and availability management.

10.2.1.5 Cloud Security

This provides the various capabilities that address security concerns like authentication and identity management, authorization, access controls and privilege management, governance and risk management, data protection, security of data in transit and at rest, threat and vulnerability management, and infrastructure protection.

10.2.2 *Benefits and Limitations of Cloud Solutions*

Some of the common benefits of using the cloud solutions include:

- Lower overall costs of infrastructure
 - Higher infrastructure utilization with resource pooling and sharing
 - Lower TCO for applications with variable workloads with dynamic scaling and pay-per-use models
 - Lower costs of business continuity and disaster recovery of IT infrastructure
- Improved service levels with self-service, elasticity, and automation
 - Better provisioning experience with self-service and automatic provisioning of applications, platforms, and infrastructure
 - Better application SLAs with dynamic resource management and auto-scaling
 - Faster time to market for business applications with faster provisioning of infrastructure and platforms
- Reduced maintenance and management concerns

- Higher scalability, elasticity, and lower administration overheads with abstraction of infrastructure and platform concerns

Some of the common limitations include:

- Security and compliance issues in deploying applications and data over infrastructure provided by third party.
- Lack of maturity, thereby making it difficult to migrate existing complex applications to cloud.
 - Applications may need re-engineering to be ported over to some cloud platforms.
 - Lack of advanced SQL support and ACID properties in NoSQL-based cloud data store options makes it difficult to migrate existing applications to cloud platforms.
 - It may be difficult to migrate specialized applications and platforms like grid computing applications.
- Scalability of private cloud deployment options may be limited.
- Use of virtualization may add performance overheads and also introduce limitations in supporting applications that cannot use virtualized infrastructure like real-time applications.

10.2.3 *Pattern Documentation Structure*

A solution architecture pattern captures standardized, reusable solution architectures for various commonly occurring problem scenarios. There are many different formats used for describing architecture patterns, but none of them has achieved widespread acceptance. The elements described below are some of the common aspects that can be found in most pattern definitions. So, these are the elements that are captured for the cloud solution architecture patterns in the subsequent sections.

Name

It represents a meaningful way to refer to a pattern. It is generally a single word or short phrase.

Intent

This section contains description of the problem indicating the intent in applying the pattern – the intended goals and objectives to be reached within the category and motivations described below.

Category

This section describes the context for which the solution is applicable.

Motivations

This section defines the various motivations and factors.

Solution

This section contains a description, using text and/or graphics for describing how to achieve the intended goals and objectives. The different approaches or implementation methods and their variants should be described in this section. The section also describes the key challenges associated with traditional solutions for the problem and how this solution meets up those challenges. The section also describes various components required to implement the solution.

Applicability

This section provides a description of scenarios and situations where this pattern can be applied.

Consequences

This section describes the implications, benefits, and limitations of applying the pattern.

Known Uses

This section describes known applications of the pattern. Known Uses can also serve as examples.

Related Patterns

This section describes the relationships between this pattern and others. The pattern may be inspired from other patterns. These may be predecessor patterns, whose resulting contexts correspond to the initial context of this one.

10.3 Common Use Cases and Cloud Solution Patterns

This section describes some of the common use cases, challenges with traditional solution options, cloud solution architecture, as well as benefits and limitations.

10.3.1 *Server Infrastructure Optimization Through Enterprise Private Clouds*

Name

Private cloud

Intent

The intent of the pattern is to optimize [4] server infrastructure used among the pool of applications with different workload peaks at different times.

Category

Deployment

Motivations

There are several scenarios in enterprises where applications require most of the resources only for a short duration of time, but the infrastructure is over provisioned to meet the peaks of such application. For example, a month-end processing application requires 50 CPU for the last 5 days of the month and requires 5 CPU for the rest of the month, but the 50 CPU infrastructure is provisioned and so the overall utilization of the resources is low.

Also, with traditional approach, in order to make sure that failure of such applications does not hamper other running applications, they are provisioned in separate set of physical servers; therefore, reallocation of unutilized resources to other applications when not needed is more difficult.

Cloud Solution

The solution is to create a cloud computing platform in the enterprise private internal network that enables pooling of available infrastructure resources and helps organization to leverage the existing infrastructure setup to provision the applications (along with all required underlying infrastructure, software platforms, etc.) on demand with reduced time and intelligent allocation of resources. This overcomes several challenges like security and compliance issues faced with public cloud adoption while leveraging existing investments. Enterprise private clouds [5] are seen as a natural progression of initiatives like virtualization of resources already taken up by several organizations. Enterprise private cloud solutions add capabilities like self-service and automation and charge back over the virtualized infrastructure. There are several implementation options like those based on VMware ESX, Citrix Xen, Citrix XenServer, Microsoft Hyper-V, Solaris zones, and IBM AIX WPARs and LPARs.

The conceptual view of the solution shown in Fig. 10.2 provides an overview of the service centric provisioning model with cloud computing.

The IT administrator uses a cloud portal created over a private cloud platform to populate a service catalog with the application(s) service templates capturing the commonly used and applicable deployment patterns in the enterprise. The solution architect then creates a service request using the service catalog based on a selected template in the cloud portal and deploys the application on the cloud infrastructure and makes it available for the end users to access them. Application developers use application development and testing tools to develop and test the applications. The operational support personnel use various tools to manage the applications over the cloud infrastructure. The cloud manager monitors the application usage and dynamically provisions additional infrastructure and scales the application when there is more usage and similarly de-provisions infrastructure when there is lower usage. It also meters the usage and charges the users based on it.

A key strategy is to dynamically reallocate the resources to different applications during their workload peak so that the infrastructure can be used efficiently. In traditional solution, resources are allocated to application with static binding which results in silos of infrastructure of each application. It also results in upfront estimation of the resources needed, and the consequent long hardware procurement cycles make deployment architecture changes difficult. The strategy with the cloud solution is to use virtualized infrastructure so that resources can be pooled and dynamically

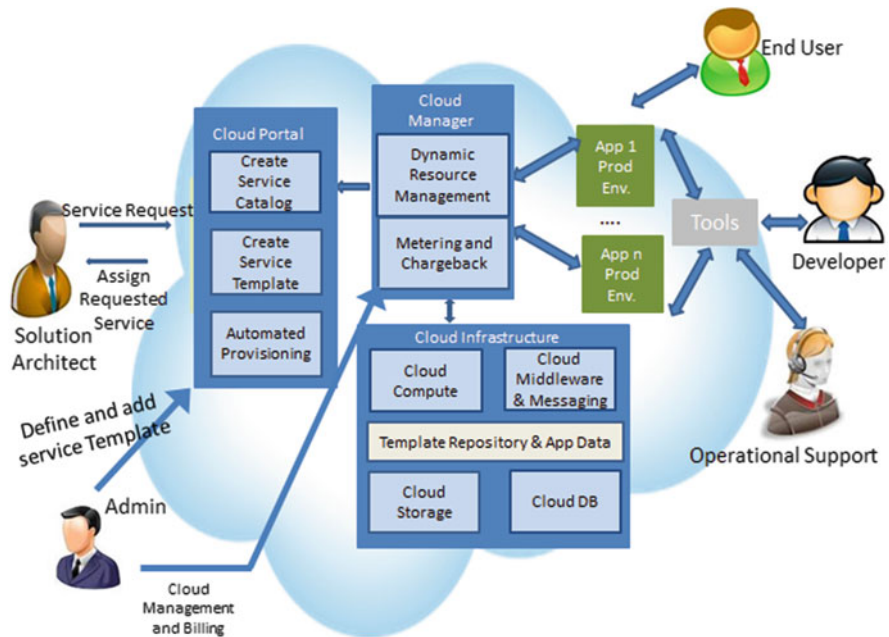


Fig. 10.2 Enterprise private cloud platform

reallocated based on current workloads and priorities of the various applications and also it enables on-demand self-service provisioning of infrastructure.

Some of the architectural building blocks needed to implement this pattern are:

- Cloud infrastructure:
 - Cloud DB: Elastic data stores that can scale up and down to meet the application's storage needs.
 - Cloud storage: Elastic storage for storing application-specific files including the application-specific template repository.
 - Compute cloud: The processing layer of the application is developed over this component.
 - Cloud middleware: Integration across application is achieved using the cloud middleware.
- Cloud manager:
 - Automated provisioning manager: For automating the provisioning of the cloud infrastructure and the application components
 - Cloud manager: For monitoring the application usage and dynamically scaling the infrastructure up and down and also to meter the usage and charge the users based on it
 - Cloud portal: For self-service provisioning of the cloud infrastructure and to define the service catalog and template specific to the application(s)

Applicability

This pattern is applicable to the scenarios in large enterprises with a big pool of applications with varying workloads deployed on existing infrastructure. In such scenarios, there will be existing investments that the enterprises will want to leverage while improving the overall utilization of the infrastructure using private clouds. Enterprises operating in domains with constant change, which need agile infrastructure to be able to reallocate them as per changing business needs, will also find this solution architecture relevant.

Consequences

With a private cloud solution, economies of scale possible within an enterprise across the various lines of business are utilized by pooling the infrastructure in the enterprise, so the consequences are that it works better in organizations which have sufficient scale, and it also means that the scalability of the cloud is limited to the scale of the single enterprise. So, the scalability of private cloud solutions is limited to the resources pooled unlike in public cloud which can have higher scalability. Private clouds are successful only when the infrastructure across the various lines of business is pooled; this requires proper governance and sharing policies so that the priorities of the various lines of business are balanced with that of savings costs across the organization. Most of the private cloud solutions available to implement need virtualization, so the applications that cannot use virtualized infrastructure like real-time applications and those that need specialized hardware cannot be provisioned over cloud infrastructure. The virtualized infrastructure may introduce performance overheads.

Known Uses

Private cloud infrastructures are majorly built for enhancing the in-house capabilities of the organizations. They are mainly customized implementations on virtualization software like VMware ESX, Citrix XenServer, and Microsoft Hyper-V. Terremark also offers private cloud infrastructure for the enterprises.

Related Pattern

Cloud burst

10.3.2 Scalable and Low-Cost Storage, Backup, and Archival: Internet Scale Content Distribution Solutions

Name

Cloud storage

Intent

The intent of the pattern is to provide a scalable and low-cost storage solution for the increasing unstructured data generated from different sources like social media and rich media data like video, from the sensor devices. The solution should also provide the efficient backup and archival mechanism for such huge data.

Category

Storage

Motivation

With increasing adoption of social media by consumers, several enterprises have been creating their own social media platforms that enable their customers form online communities, use online discussion forums, and provide reviews and ratings of the services and solutions. This is resulting in large amounts of data that needs to be stored.

Use of rich media for better user experience is common these days, so there are scenarios where an enterprise wants a web-based portal for its applications. This portal is content heavy as it uses a lot of multimedia content like a rich promotion campaign with a lot of video and image files. So, this huge amount of data that is generated needs to be stored and its backup and archival also needs to be maintained. As the amount of information collected is growing, the backup of such information is also growing. Using traditional storage solutions involve a lot of CAPEX and a lot of administration overheads. Traditional solutions are better suited for scenarios where the value per byte of information is high like transactional data, but for scenarios like social media data where the value per byte is not well defined but can be a potential source of insights and competitive differentiation, there is a need for alternative solutions. Another aspect is lack of well-defined structure for these new kinds of information requiring solutions based on alternative architectures that are better suited for these.

Cloud Solution

The solution is to leverage several cloud storage [6] and implementation options based on private and public cloud deployment models which will provide a scalable and low-cost storage. There are public cloud storage solutions like Amazon S3 and Azure Blob Storage that provide a scalable solution for storing files through a public API (REST and WebServices). These solutions provide highly scalable storage with a pay-per-use model for storage and retrieval.

There are several content delivery network solutions available, e.g., CloudFront from Amazon, Windows Azure CDN, and CloudFlare that provide the ability to syndicate content to multiple servers across the globe to enable faster response times in content downloads. Some of these are integrated with the cloud storage solutions, thus making it easier for developing content-centric web applications.

There are several public cloud database solutions, like Amazon Simple DB, SQL Azure, and Google Big Table, which provide scalable and on-demand object storage solutions with public API. There are also NoSQL solutions like HBase, MongoDB, Cassandra, and CouchDB that enable implementing scalable in-house data stores leveraging commodity hardware.

There are also open-source solutions like Hadoop Distributed File System and Commercial Off-The-Shelf (COTS) storage solutions from vendors like Nirvanix and Parascala that enable creating scalable file storage based on commodity hardware within the enterprise network. These use technologies similar to those used by the public cloud storage solutions to provide a scalable solution.

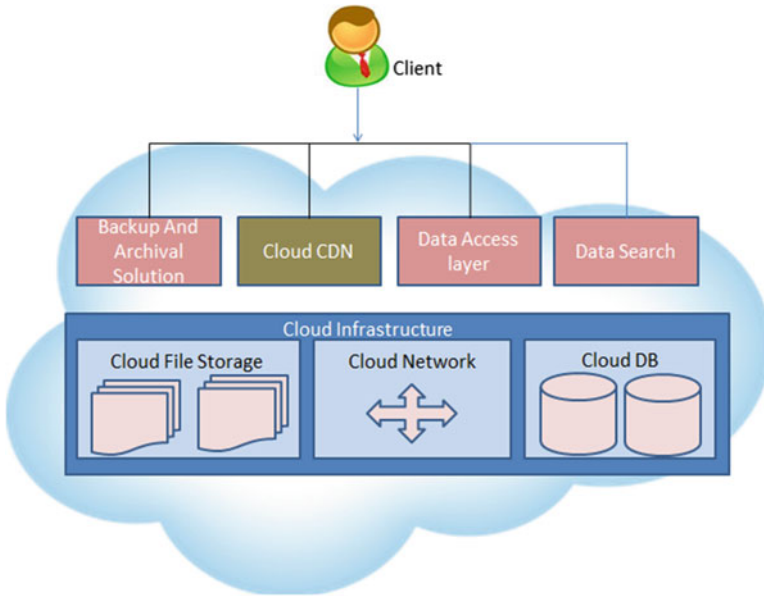


Fig. 10.3 Cloud storage, archival, content delivery solutions

With disk slowly becoming the next tape and memory becoming the next disk due to advances in technologies and falling prices, there are several in-memory storage solutions emerging. There are open-source distributed cache solutions like memcached and COTS from vendors like Oracle (Coherence), GemFire, Terracotta, GigaSpaces, and SAP (HANA) that provide scalable in-memory data stores.

The conceptual view of solution to address storage needs based on these technologies is shown in Fig. 10.3.

A scalable storage infrastructure layer is created leveraging public cloud file storage, public cloud data stores, in-house NoSQL data stores, in-memory data stores, and distributed file system-based solutions. A data access and search layer is created which provides abstractions for the various logical categories of data stores (like File Store, Document Store, Key-Value Store, Column Store, Graph Store) hiding the implementation and deployment details of the underlying infrastructure layer. The data access layer is then leveraged by the platform and application layers. Specialized frameworks like “backup and archival,” “content delivery network (CDN)” based on the underlying cloud storage infrastructure layer, addressing the specific needs of a particular use case, are then developed or bought/leveraged if COTS solutions are available.

Here, a key challenge is that huge amount of data needs to be handled, and existing traditional storage solutions are too expensive and cannot scale. The cost of maintenance of such high-end storage and network infrastructure is also very high. So, the strategy with the cloud solution is to provide a horizontally scalable solution

with the commodity hardware. Public cloud deployment models can help if there is a need to convert from CAPEX- to OPEX-based financial model.

Some of the architectural building blocks needed to implement this solution are:

- Cloud infrastructure:
 - Cloud DB: Horizontally scalable and elastic data stores based on commodity hardware that can scale up and down to meet the application's storage needs
 - Cloud storage: Elastic storage for storing application-specific files
- Cloud CDN: These help address media content distribution use cases.

Applicability

The pattern is applicable for the scenarios where huge volumes of data where the cost per unit of data is low. Such scenarios need a storage solution that should be scalable with low cost while providing acceptable response time.

Consequences

Data leaves the organization boundary for public cloud storage options, and consequently data protection and regulatory compliance issues need to be dealt with though there will be benefits of higher scalability and probably lower costs too.

Several of these cloud storage and database solutions are still evolving so they may lack features like strict ACID properties and extensive support of SQL. Therefore, migration of existing complex application over to cloud infrastructure may not be straightforward.

As per the CAP (aka Brewer's) theorem, it is not possible to provide consistency, availability, and partition tolerance simultaneously. Several of these NoSQL storage solutions may have made a choice of guaranteeing one or two of these architectural concerns and so may suit only those use cases where the choice made is appropriate.

Known Uses

Google Drive, Dropbox, and Microsoft's SkyDrive are a few popular cloud-based storage solutions.

Related Pattern

Big data analytics

10.3.3 Handling Workload Spikes with Cloud Burst

Name

Cloud burst

Intent

The intent of the pattern is to provide a cost-effective solution that enables an application to meet minimum service levels during unexpected workload spikes.

Category

Deployment, application

Motivations

There are scenarios in enterprises where there are existing applications deployed over existing infrastructure for which investments have already been done. Existing infrastructure is able to handle normal workloads but is unable to meet the occasional spikes in workloads with acceptable service levels, for example, an e-commerce portal that receives a lot of requests after a marketing campaign.

In the traditional solution, there will always be a trade-off between meeting the service levels and keeping a lot of infrastructure idle with over provisioning. Like if we have the infrastructure for normal workload, the SLAs will be met at normal times only, but during the peak times the SLAs will not be met. If we design the infrastructure to handle peak workload, then most of the time the resources will lie idle.

Cloud Solution

The solution strategy is to make use of public cloud platforms to handle the workload bursts while leveraging in-house private clouds for the normal workloads. Cloud bursting [8] refers to the ability to get infrastructure at runtime from a cloud provider and enabling the applications to use that to meet the service levels during a temporary spike in the workload.

There are solutions that allow such seamless migration from private cloud setup to a public cloud during temporary workload spikes to maintain the SLAs like vCloud Express which is based on VMware technologies that are designed to enable seamless migration of workloads from internal VMware cloud to a public cloud provider supporting vCloud API. Similarly, there are other hybrid cloud solutions from vendors like CloudSwitch and Elastra that are designed for these use cases. Hybrid cloud solutions enable organizations to leverage their existing investments in the application infrastructure for normal workloads while at the same time meeting the service levels by paying for additional infrastructure only when needed.

Figure 10.4 provides a conceptual view of the cloud burst solution pattern.

In Fig. 10.4, the application seamlessly bursts over onto a public cloud using a cloud mediator. The cloud manager will take care of auto-provisioning of application through cloud portal, meters the usage, and bills the customers accordingly.

One of the key challenges is to maintain the SLAs in case of temporary spikes in the workload. So, the cloud solution strategy here will be to have a hybrid cloud setup with application on private cloud and burst to public cloud to handle spikes. It is basically about getting the extra infrastructure from the public cloud provider to handle the workload thereby maintaining the SLAs and reducing the costs.

Some of the architectural building blocks needed to implement this are:

- Cloud infrastructure:
 - Cloud compute and cloud DB: Elastic compute platform and the data stores that can scale up and down to meet the application's processing and storage needs
- Cloud manager
 - Automated provisioning manager: For automating the provisioning of the cloud infrastructure and the application components
 - Cloud manager: For monitoring the application usage and provisioning infrastructure from the cloud to handle the workload burst

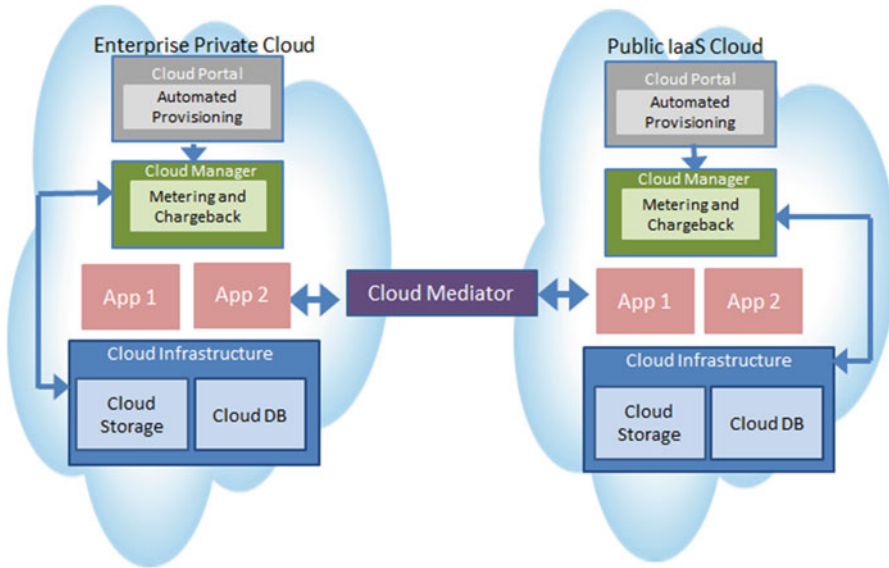


Fig. 10.4 Hybrid cloud solutions for “cloud burst”

- Cloud mediator: This component enables seamless migration of applications from private to public cloud setup.

Applicability

The pattern is applicable for the scenarios that involve an application that experiences occasional workload bursts.

Consequences

This solution architecture involves use of public cloud infrastructure so there will be associated benefits and issues. These solutions typically work only if bottleneck is compute capacity, but in several scenarios the bottleneck may not just be the compute platform and can be other components of the application architecture like the database. It is harder to make a typical commercial database leverage additional infrastructure from a public cloud and scale as they are typically designed to scale up vertically than scale up horizontally. Also, migration of applications over public cloud infrastructure may result in security and compliance issue. Many of these solutions are still not mature enough to handle complex application.

Known Uses

WebSphere CloudBurst and IBM CloudBurst, are the known implementations for handling cloud bursts.

Related Pattern

Public cloud applications, private cloud

10.3.4 Cloud-Based On-Demand Development and Testing

Name

Test and development clouds

Intent

The intent of the pattern is to enable on-demand development and test platforms.

Category

Structural

Motivations

During the software/application development process, creation of separate development (Dev) and test environments is a very common practice. Nearly 30–50 % servers in a typical large enterprise are dedicated for such development and test environment, and these environments are idle for most of the time as they are used only during the application software releases. Each application may have its own test and development environment. Such silos of development and test environment for each application may result in underutilization of hardware resources.

It may also be possible that an application may require multiple test environments. In such cases if the infrastructure sizing is done for the peak, there will be a significant underutilization of the infrastructure. Moreover, the setting up and maintenance of these environments is not an easy task. It requires a lot of effort and time to set up these environments. Because of the complexities involved, setting up test environments results in long provisioning times, thereby delaying projects.

Organizations are also facing challenges in creating test environments for Internet applications at the scale that matches production setup, so usually the test environment may not be at the same scale as the production environment because of which the testing may not reflect a real production scenario and so may not be adequate.

Cloud Solution

The solution is to leverage an on-demand development and test platforms based on cloud infrastructure [7] on pay-per-use model that can be provisioned as needed and decommissioned when they are not needed. These solutions enable creation of test and development environments on demand through automation and self-service. Multiple deployment models based on combinations of public and private clouds are possible for the testing tools and environments.

The conceptual view of the solution is shown in Fig. 10.5.

Templates of the test and development environments of the applications are created using virtual machine snapshots and are then used to provision the development and test environments on demand and made available to the developers and testers. Testing tools can be provisioned on demand on the cloud infrastructure itself which are used for manual or automated testing of the applications. The cloud portal provides the self-service portal and the catalog with the environment templates.

A key challenge is to provision a complete environment with a single click. The challenge here is to provision all the elements of the environment such as

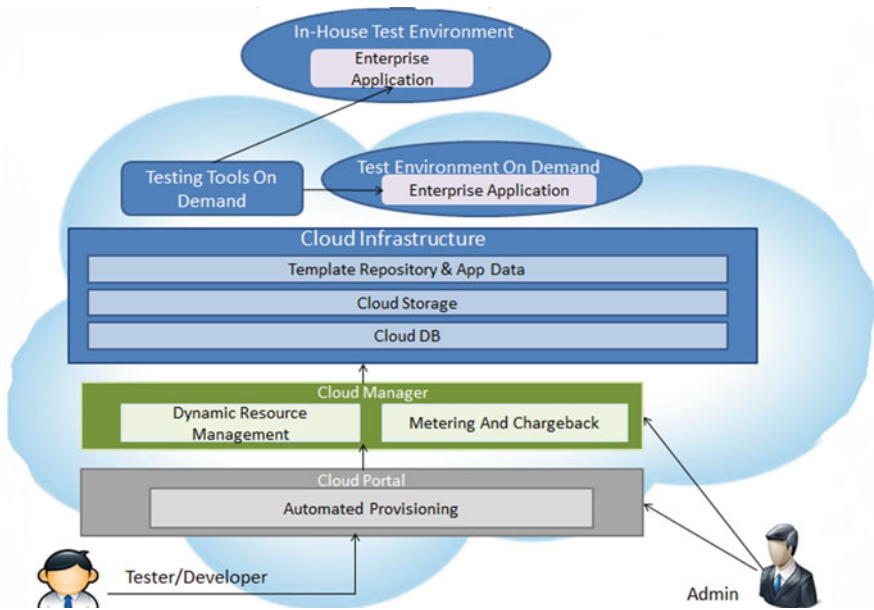


Fig. 10.5 Testing on cloud

web servers, application servers, database servers, development tools, and testing tools with one click. This abstraction of the complete environment as a single service instance enables easier management as the system can track the various infrastructure and software components provisioned as part of an environment instance that has been provisioned.

The various architectural components of this solution are:

- Cloud infrastructure:
 - Cloud storage and DB: To provide the storage needs of the test and development environments including test data and virtual machine snapshots
- Cloud manager
 - Automated provisioning manager: For automating the provisioning of the cloud infrastructure, the application components, testing, and development tools
 - Cloud manager: For monitoring and management, metering and billing, etc.
 - Cloud portal: For self-service provisioning of the cloud infrastructure

Applicability

The pattern is applicable for large enterprises with a lot of software application development and releases. Such enterprises can benefit from being able to provision and de-provision test and development environments on demand.

Consequences

Provisioning virtualized test environments can cause a problem if the applications use non-virtualized infrastructure in production environment as the use of virtualized environment for testing may not uncover all potential problems in production.

Also, security and compliance issues for public cloud-based deployment models still remain open for this solution as well.

Known Uses

PushToTest is an example of a public cloud-based testing tool. Test and development environment provisioning is one of the most common use cases for public cloud usage.

Related Pattern

High-performance computing cloud, cloud desktops

10.3.5 Business Continuity with Cloud Infrastructure

Name

Disaster recovery in cloud

Intent

The intent of the pattern is to provide a cost-effective disaster recovery solution leveraging cloud infrastructure.

Category

Behavioral

Motivations

Enterprises which adhere to regulatory compliances such as PCI DSS, HIPPA, and SOX generally have a DR site which is an exact replication of the production site. The traditional method of setting up and maintaining a DR site involves maintaining a replica of data center with the servers, storage, and networking in a different place. This redundancy incurs high costs.

Also, testing the DR process each time for any change in the production environment is a complex and costly task.

Cloud Solution

An alternative to the traditional approach is to leverage cloud infrastructure for business continuity/disaster recovery [9]. The production environment is virtualized; after that the servers are backed up as VM templates on cloud storage and provisioned on demand during disaster recovery from these templates, thus reducing dependency on the physical servers and enabling automation in provisioning the servers and the applications making the DR process simpler than it is otherwise with traditional solution using physical servers and applications. The pay-per-use model reduces capital expenditure in maintaining the infrastructure. The cloud infrastructure makes the solution more scalable.

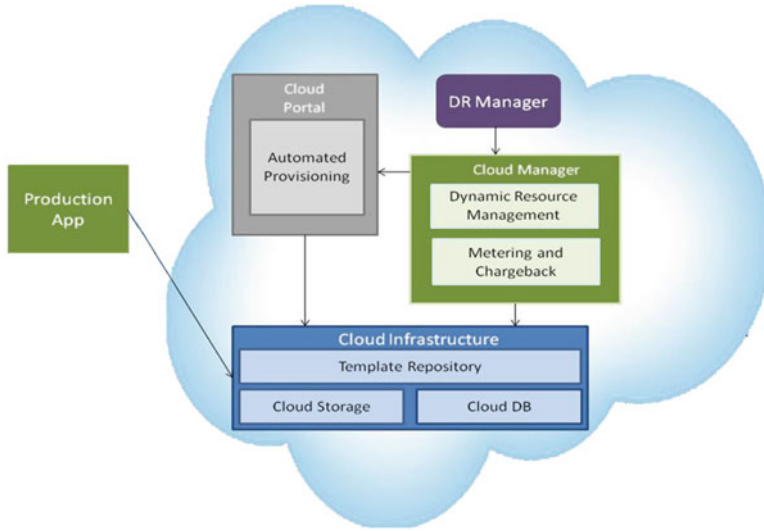


Fig. 10.6 Business continuity using cloud infrastructure

The conceptual view of the solution pattern is shown in Fig. 10.6.

The production servers are virtualized, and periodically virtual machine snapshots are taken of the production servers. These snapshots are transferred and stored on a cloud storage, and when needed the servers are recreated over the cloud using those snapshots. Similarly replicas of data stores and file systems are also created and stored on cloud storage. They are then used during disaster scenarios to restore the data. Cloud providers generally provide multiple data centers spread geographically across multiple locations, thus providing high availability. By leveraging such infrastructure, the disaster recovery costs are reduced.

Cloud providers offer only standardized infrastructure so there will be limitations. Security and regulatory concerns with public cloud need to be addressed to be able to use this effectively. There can also be issues with transferring large amounts of data which may affect the recovery point objective (RPO) and recovery time objectives (RTO) of the DR solution using this approach.

Some of the architectural building blocks needed to implement this are:

- Cloud infrastructure:
 - Cloud storage and DB: For taking the backups of the virtual machines and data
- Cloud manager
 - Automated provisioning manager: For automating the provisioning of the applications during recovery
 - Cloud manager: For monitoring and managing the cloud infrastructure and also for metering and chargebacks
 - Cloud portal: For self-service provisioning of the DR site and applications

Applicability

The pattern is applicable for scenarios where DR is needed for simple applications, especially for small and medium businesses (SMBs) who cannot afford the costs of a redundant DR site.

Consequences

There can be performance issues with replication and backups over a WAN which may make it technically difficult to meet some of the expected RPO and RTO service levels in some cases.

Known Uses

Dell AppAssure's Cloud Backup and Disaster Recovery and Citrix Cloud Solution for Disaster Recovery are examples of such cloud-based business continuity solutions.

Related Pattern

Private cloud, public cloud applications

10.3.6 Desktop as a Service Using Cloud Infrastructure

Name

Cloud desktops

Intent

The intent of the pattern is to provide low-cost solution providing desktops on demand with appropriate work environments.

Category

Deployment

Motivations

A large portion of IT resources is spent in managing and operating desktop environments for employees. There are several complexities involved in maintaining such an environment like periodic technology refresh, providing patches and updates, providing disaster recovery and backups, and support for employee mobility. Traditional approach of procuring physical desktops and using them involves several complexities because of the physical distribution of the desktops across various locations and incurs high costs.

Another common scenario is that of a large enterprise starting a new branch in a different country. The complexities of managing physical desktops and local compliance requirements make it highly cost prohibitive to manage employee desktops at remote locations.

Cloud Solution

The solution is to set up a desktop cloud to provision remote desktops on cloud infrastructure [10]. A desktop cloud solution helps to provide "anytime, anywhere" access to applications, information, and resources, thereby improving

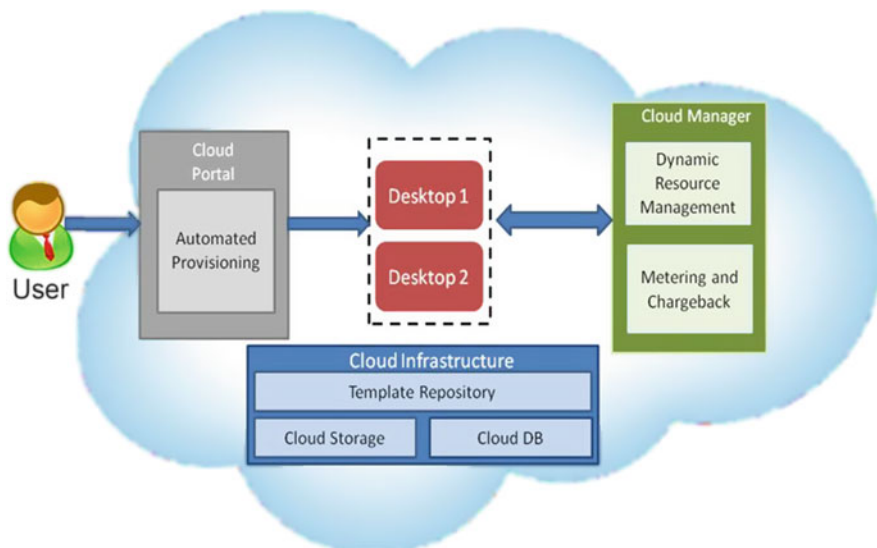


Fig. 10.7 Desktop cloud solution

employee productivity. These solutions generally enable thin clients or any other Internet-connected device to access virtualized desktops with hosted applications. Organizations are looking to use these solutions to avoid the costs associated with constant refresh of desktop hardware. There are also multiple deployment models with private cloud being a popular choice. Refer to Fig. 10.7.

Cloud infrastructure is used to provision the remote desktops for users/employees. The cloud manager monitors the application usage and dynamically provisions additional infrastructure and scales the number of desktops required based on the usage and similarly de-provisions infrastructure when there is lower usage. This significantly cuts the expenditure on infrastructure. The cloud portal can be used by the user to provision desktops and storage on demand.

A key challenge here is to secure the data and desktops on cloud. There are several cloud security solutions like virtual private clouds (e.g., Amazon AWS VPC), disk encryption software, and single sign-on and identity management solutions that can be leveraged to address several security concerns.

Another key challenge is addressing performance and latency issues, especially if the users have to access the resources over a WAN.

Some of the architectural building blocks needed to implement this solution are:

- Cloud infrastructure:
 - Compute cloud: For the desktop computing needs
 - Cloud storage: For storing application-specific files and repository of VM templates

- Cloud manager
 - Auto-provisioning manager: For automating the provisioning of remote desktops and applications
 - Cloud manager: For monitoring the individual desktop resource usage and dynamically scaling the infrastructure up and down
 - Cloud portal: For self-service provisioning of the desktops

Applicability

This pattern is applicable for large organizations having a large number of employees. The solution not only results in significant cost cutting but also provides scalability of infrastructure.

Consequences

This solution results in centralization of the computing resources of all desktops across the organization, which results in reduced complexity in managing them ensuring timely updates, security, easier employee mobility, etc. The flip side is that there can be loss of employee productivity if there are latency and performance issues.

Known Uses

IBM Smart Business Desktop Cloud is an example of an implementation of this pattern.

Related Pattern

High-performance computing cloud

10.4 Conclusion

This chapter has provided an overview of the architectural building blocks of a typical cloud solution and then described a few solution patterns elaborating on common problem scenarios that enterprise solution architects face at the infrastructure layer while developing enterprise solutions, the traditional solution approaches and their limitations, and cloud computing solution approaches and their benefits and limitations. Enterprise infrastructure solution architects can benefit from reusing these solution patterns. As the technologies evolve, the solution strategies also need to evolve. So, these should be treated as starting points for capturing these evolving solutions.

References

1. Mell, P., Grance, T.: The NSIT definition of cloud computing. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf> (2011)
2. Guest, S.: 5 cloud computing patterns along with examples and Azure solutions. <http://www.infoq.com/presentations/Patterns-for-Cloud-Computing> (2010)

3. Amrhein, D., Anderson, P.: Cloud computing use cases. http://www.opencloudmanifesto.org/Cloud_Computing_Use_Cases_Whitepaper-2_0.pdf (2009)
4. CA Technologies: IT optimization through workload automation. http://whitepapers.business-week.com/detail/RES/1335968686_109.html (2012)
5. Doddavula, S., Wasudeo, A.: Adopting cloud computing – enterprise private clouds: <http://www.infosys.com/infosys-labs/publications/infosyslabs-briefings/documents/cloud-computing-enterprise-private-clouds.pdf> (2009)
6. Kim, S.-G., Han, H., Eom, H., Yeom, H.Y.: Toward a cost-effective cloud storage service. In: Advanced Communication Technology (ICACT), The 12th International Conference. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5440497&isnumber=5440112>
7. Tilley, S., Parveen, T.: Migrating software testing to the cloud. In: IEEE International Conference on Software Maintenance (ICSM). <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5610422&isnumber=5609528> (2010)
8. Bicer, T., Chiu, D., Agrawal, G.: A framework for data-intensive computing with cloud bursting. In: IEEE International Conference on Cluster Computing (CLUSTER). <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5610422&isnumber=5609528> (2011)
9. Chris, P.: Business continuity with cloud. <http://www.eweek.com/c/a/Cloud-Computing/How-to-Ensure-Business-Continuity-with-Cloud-Computing/> (2010)
10. Carta, D.P.: Your office in the clouds: the best online virtual desktops. <http://gigaom.com/collaboration/your-office-in-the-clouds-the-best-online-virtual-desktops/> (2009)
11. Fehling, C., Retter, R.: Cloud computing patterns. <http://cloudcomputingpatterns.org/> (2011)
12. Ho, R.: Cloud computing patterns. <http://architects.dzone.com/news/cloud-computing-patterns> (2009)
13. IBM: Pattern solutions. <http://www.ibm.com/developerworks/rational/products/patternsolutions/> (2011)