UNIT – II

Introduction:

- Utility-oriented data centers are the first outcome of cloud computing, and they serve as the infrastructure through which the services are implemented and delivered.
- Any cloud service, whether virtual hardware, development platform, or application software, relies on a distributed infrastructure owned by the provider or rented from a third party.
- Cloud can be implemented using a datacenter, a collection of clusters, or a heterogeneous distributed system composed of desktop PCs, workstations, and servers.
    - In most cases hardware resources are virtualized to provide isolation of workloads and to best exploit the infrastructure.
- According to the specific service delivered to the end user, different layers can be stacked on top of the virtual infrastructure:
    - a virtual machine manager, a development platform, or a specific application middleware.
- A broad definition of the phenomenon could be as follows:
    - Cloud computing is a utility-oriented and Internet-centric way of delivering IT services on demand.
    - These services cover the entire computing stack: from the hardware infrastructure packaged as a set of virtual machines to software services such as development platforms and distributed applications.

The cloud reference model:

- Cloud computing supports any IT service that can be consumed as a utility and delivered through a network, most likely the Internet.
- Such characterization includes quite different aspects: infrastructure, development platforms, application and services.
- **Architecture:**
- It is possible to organize all the concrete realizations of cloud computing into a layered view covering the entire stack (see Figure 4.1), from hardware appliances to software systems.
- Cloud resources are bound to offer “computing horsepower” required for providing services.
    - Often, this layer is implemented using a datacenter in which hundreds and thousands of nodes are stacked together.

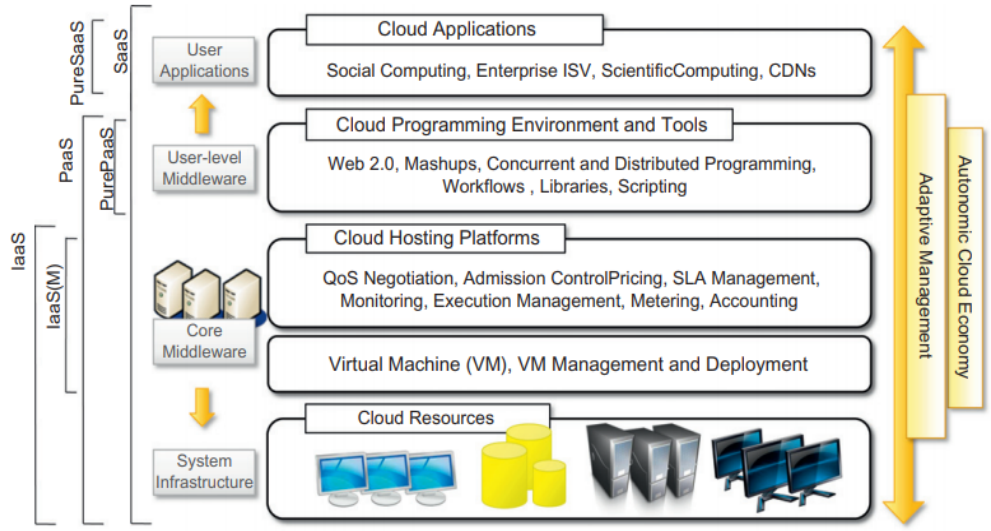


FIGURE 4.1

The cloud computing architecture.

- Cloud infrastructure can be heterogeneous in nature because a variety of resources, such as clusters and even networked PCs, can be used to build it.
- Moreover, database systems and other storage services can also be part of the infrastructure.

- The physical infrastructure is managed by the core middleware, the objectives of which are to provide an appropriate runtime environment for applications and to best utilize resources.
- At the bottom of the stack, virtualization technologies are used to guarantee runtime environment customization, application isolation, sandboxing, and quality of service.
- Hypervisors manage the pool of resources and expose the distributed infrastructure as a collection of virtual machines.
- By using virtual machine technology it is possible to finely partition the hardware resources such as CPU and memory and to virtualize specific devices, thus meeting the requirements of users and applications.
- This solution is generally paired with storage and network virtualization strategies, which allow the infrastructure to be completely virtualized and controlled.
- Infrastructure management is the key function of core middleware, which supports capabilities such as negotiation of the quality of service, admission control, execution management and monitoring, accounting, and billing.
- The combination of cloud hosting platforms and resources is generally classified as a Infrastructure-as-a-Service (IaaS) solution.
- We can organize the different examples of IaaS into two categories:

- Some of them provide both the management layer and the physical infrastructure; others provide only the management layer (IaaS (M)).
- In this second case, the management layer is often integrated with other IaaS solutions that provide physical infrastructure and adds value to them.

- In a scenario, users develop their applications specifically for the cloud by using the API exposed at the user-level middleware, this approach is also known as Platform-as-a-Service (PaaS).
- PaaS offered to the user is a development platform rather than an infrastructure, which generally include the infrastructure as well, which is bundled as part of the service provided to users.
- The top layer of the reference model depicted in Figure 4.1 contains services delivered at the application level.
- These are mostly referred to as Software-as-a-Service (SaaS).
  - In most cases these are Web-based applications that rely on the cloud to provide service to end users.
  - The horsepower of the cloud provided by IaaS and PaaS solutions allows independent software vendors to deliver their application services over the Internet.
- As a vision, any service offered in the cloud computing style should be able to adaptively change and expose an autonomic behavior, in particular for its availability and performance.
- As a reference model, it is then expected to have an adaptive management layer in charge of elastically scaling on demand.

Table 4.1 Cloud Computing Services Classification

Category	Characteristics	Product Type	Vendors and Products
SaaS	Customers are provided with applications that are accessible anytime and from anywhere.	Web applications and services (Web 2.0)	SalesForce.com (CRM) Clarizen.com (project management) Google Apps
PaaS	Customers are provided with a platform for developing applications hosted in the cloud.	Programming APIs and frameworks Deployment systems	Google AppEngine Microsoft Azure Manjrasoft Aneka Data Synapse
IaaS/HaaS	Customers are provided with virtualized hardware and storage on top of which they can build their infrastructure.	Virtual machine management infrastructure Storage management Network management	Amazon EC2 and S3 GoGrid Nirvanix

Infrastructure- and hardware-as-a-service:

- Infrastructure- and Hardware-as-a-Service (IaaS/HaaS) solutions are the most popular and developed market segment of cloud computing, which deliver customizable infrastructure on demand.
- The available options within the IaaS offering umbrella range from single servers to entire infrastructures, including network devices, load balancers, and database and Web servers.

- The main technology used to deliver and implement these solutions is hardware virtualization.
- Virtual machines also constitute the atomic components that are deployed and priced according to the specific features of the virtual hardware: memory, number of processors, and disk storage.
- IaaS/HaaS solutions bring all the benefits of hardware virtualization: workload partitioning, application isolation, sandboxing, and hardware tuning.
    - From the perspective of the service provider, IaaS/HaaS allows better exploiting the IT infrastructure and provides a more secure environment where executing third party applications.
    - From the perspective of the customer it reduces the administration and maintenance cost as well as the capital costs allocated to purchase hardware.
- At the same time, users can take advantage of the full customization offered by virtualization to deploy their infrastructure in the cloud.
- Besides the basic virtual machine management capabilities, additional services can be provided, generally including the following……
    - SLA resource-based allocation, workload management, support for infrastructure design through advanced Web interfaces, and the ability to integrate third-party IaaS solutions.
- Figure 4.2 provides an overall view of the components forming an Infrastructure-as-a-Service solution.

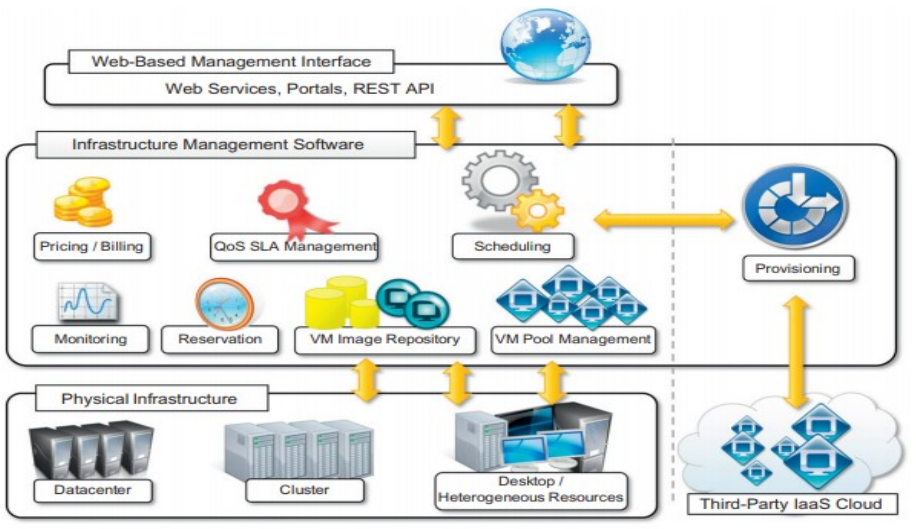


FIGURE 4.2
Infrastructure-as-a-Service reference implementation.

- It is possible to distinguish three principal layers: the physical infrastructure, the software management infrastructure, and the user interface.
- At the top layer the user interface provides access to the services exposed by the software management infrastructure.

- Such an interface is generally based on Web 2.0 technologies, which allow either applications or final users to access the services exposed by the underlying infrastructure.

- The core features of an IaaS solution are implemented in the infrastructure management software layer.
  - In particular, management of the virtual machines is the most important function performed by this layer.
- A central role is played by the scheduler, which is in-charge of allocating the execution of virtual machine instances.
- The scheduler interacts with the other components that perform a variety of tasks:
  - The pricing and billing component takes care of the cost of executing each virtual machine instance and maintains data that will be used to charge the user.
  - The monitoring component tracks the execution of each virtual machine instance and maintains data required for reporting and analyzing the performance of the system.
  - The reservation component stores the information of all the virtual machine instances that have been executed or that will be executed in the future.
  - If support for QoS-based execution is provided, a QoS/SLA management component will maintain a repository of all the SLAs made with the users; together with the monitoring component, this component is used to ensure for the desired quality of service.
  - The VM repository component provides a catalog of virtual machine images that users can use to create virtual instances.
  - A VM pool manager component is responsible for keeping track of all the live instances.
  - Finally, for the integration of additional resources belonging to a third-party IaaS provider, a provisioning component interacts with the scheduler to provide a VM instance that is external, directly managed by the pool.
- The bottom layer is composed of the physical infrastructure, on top of which the management layer operates.
- From an architectural point of view, the physical layer also includes the virtual resources that are rented from external IaaS providers.

Platform as a service:

- Platform-as-a-Service (PaaS) solutions provide a development and deployment platform for running applications in the cloud.
- They constitute the middleware on top of which applications are built. A general overview of the features characterizing the PaaS approach is given in Figure 4.3.

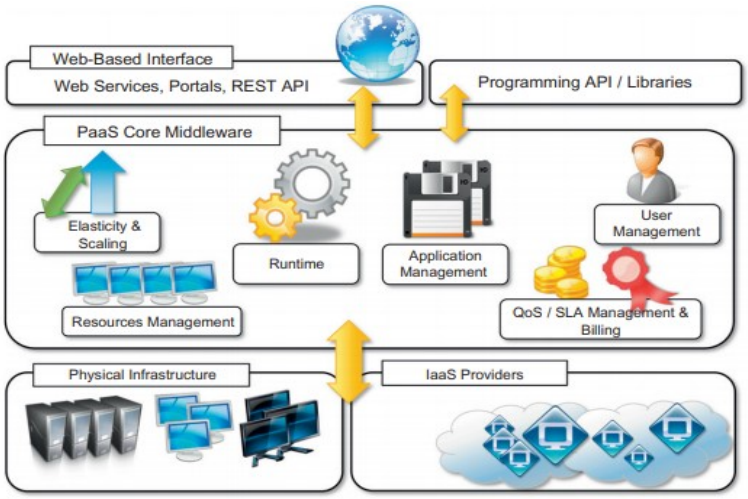


FIGURE 4.3
The Platform-as-a-Service reference model.

- Application management is the core functionality of the middleware.
  - PaaS implementations provide applications with a runtime environment and do not expose any service for managing the underlying infrastructure.
  - They automate the process of deploying applications to the infrastructure, configuring application components, provisioning and configuring supporting technologies such as load balancers and databases, and managing system change based on policies set by the user.
  - Developers design their systems in terms of applications and are not concerned with hardware (physical or virtual), operating systems, and other low-level services.
  - The core middleware is in-charge of managing the resources and scaling applications on demand or automatically, according to the commitments made with users.
- From a user point of view, the core middleware exposes interfaces that allow programming and deploying applications on the cloud.
  - These can be in the form of a Web-based interface or in the form of programming APIs and libraries.
  - The specific development model decided for applications determines the interface exposed to the user.
- Some implementations provide a completely Web-based interface hosted in the cloud and offering a variety of services.
- PaaS solutions can offer middleware for developing applications together with the infrastructure or simply provide users with the software that is installed on the user premises.
  - In the first case, the PaaS provider also owns large datacenters where applications are executed; in the second case, referred as Pure PaaS, the middleware constitutes the core value of the offering.

- It is also possible to have vendors that deliver both middleware and infrastructure and ship only the middleware for private installations.
- The PaaS umbrella encompasses a variety of solutions for developing and hosting applications in the cloud. Despite this heterogeneity, there are some essential characteristics that identify a PaaS solution……
- **Runtime framework.**
    - This framework represents the "software stack" of the PaaS model and the most in-built aspect that comes to people's minds when they refer to PaaS solutions.
    - The runtime framework executes end-user code according to the policies set by the user and the provider.
- **Abstraction:**
    - PaaS solutions are distinguished by the higher level of abstraction that they provide.
    - Whereas in the case of IaaS solutions the focus is on delivering "raw" access to virtual or physical infrastructure, in the case of PaaS the focus is on the applications the cloud must support.
    - This means that PaaS solutions offer a way to deploy and manage applications on the cloud rather than a bunch of virtual machines on top of which the IT infrastructure is built and configured.
- **Automation:**
    - PaaS environments automate the process of deploying applications to the infrastructure, scaling them by provisioning additional resources when needed.
    - This process is performed automatically and according to the SLA made between the customers and the provider.
    - This feature is normally not native in IaaS solutions, which only provide ways to provision more resources.
- **Cloud services:**
    - PaaS offerings provide developers and architects with services and APIs, helping them to simplify the creation and delivery of elastic and highly available cloud applications.
    - These services are the key differentiators among competing PaaS solutions and generally include specific components for developing applications, advanced services for application monitoring, management, and reporting.
- Another essential component for a PaaS-based approach is the ability to integrate third-party cloud services offered from other vendors by leveraging service-oriented architecture.

- PaaS environments deliver a platform for developing applications, which exposes a well-defined set of APIs and, in most cases, binds the application to the specific runtime of the PaaS provider.
- Finally, from a financial standpoint, PaaS solutions can cut the cost across development, deployment, and management of applications.
- It helps management reduce the risk of ever-changing technologies by offloading the cost of upgrading the technology to the PaaS provider.

Software as a service:

- Software-as-a-Service (SaaS) is a software delivery model that provides access to applications through the Internet as a Web-based service.
- It provides a means to free users from complex hardware and software management by offloading such tasks to third parties, which build applications accessible to multiple users through a Web browser.
- In this scenario, customers neither need install anything on their premises nor have to pay considerable up-front costs to purchase the software and the required licenses.
    - They simply access the application website, enter their credentials and billing details, and can instantly use the application, which can be further customized for their needs.
- On the provider side, the specific details and features of each customer's application are maintained in the infrastructure and made available on demand.
- The SaaS model is appealing for applications serving a wide range of users and that can be adapted to specific needs with little further customization.
- This requirement characterizes SaaS as a "one-to-many" software delivery model, whereby an application is shared across multiple users.
    - Example: CRM3 and ERP4 applications that constitute common needs for almost all enterprises, from small to medium-sized and large business.
    - Every enterprise will have the same requirements for the basic features concerning CRM and ERP; different needs can be satisfied with further customization.
- SaaS applications are naturally multitenant.
    - Multitenancy, which is a feature of SaaS compared to traditional packaged software, allows providers to centralize and sustain the effort of managing large hardware infrastructures, maintaining and upgrading applications

transparently to the users, and optimizing resources by sharing the costs among the large user base.

- On the customer side, such costs constitute a minimal fraction of the usage fee paid for the software.
- The analysis carried out by Software Information & Industry Association (SIIA) was mainly oriented to cover application service providers(ASPs) and all their variations, which capture the concept of software applications consumed as a service in a broader sense.
- ASPs already had some of the core characteristics of SaaS:
  - The product sold to customer is application access.
  - The application is centrally managed.
  - The service delivered is one-to-many.
  - The service delivered is an integrated solution delivered on the contract, which means provided as promised.
- The benefits delivered from SaaS are as the following:
  - Software cost reduction and total cost of ownership (TCO) were paramount
  - Service-level improvements and Rapid implementation
  - Standalone and configurable applications.
  - Subscription and pay-as-you-go (PAYG) pricing.
- Office automation applications are also an important representative for SaaS applications:
- Google Documents and Zoho Office are examples of Web-based applications that aim to address all user needs for documents, spreadsheets, and presentation management.

Open challenges:

- Still in its beginning, cloud computing presents many challenges for industry and academia.
- In this section, we highlight the most important ones:
  - The definition and the formalization of cloud computing,
  - The interoperation between different clouds,
  - The creation of standards,
  - Security,
  - Scalability,
  - Fault tolerance, and
  - Organizational aspects.

Cloud definition:

- As discussed earlier, there have been several attempts made to define cloud computing and to provide a classification of all the services and technologies identified as such.
- One of the most comprehensive formalizations is noted in the NIST working definition of cloud computing.
    - It characterizes cloud computing as on-demand self-service, broad network access, resource-pooling, rapid elasticity, and measured service;
    - It classifies services as SaaS, PaaS, and IaaS; and
    - It categorizes deployment models as public, private, community, and hybrid clouds.
- Despite the general agreement on the NIST definition, there are alternative taxonomies for cloud services.
    - David Linthicum, founder of Blue-Mountains Labs, provides a more detailed classification, which understands 10 different classes and better suits the vision of cloud computing within the enterprise.
    - A different approach has been taken at the University of California, Santa Barbara(UCSB), which departs from the XaaS concept and tries to define an ontology for cloud computing.
    - In their work the concept of a cloud is dissected into five main layers: applications, software environments, software infrastructure, software kernel, and hardware.
        - Each layer addresses the needs of a different class of users within the cloud computing community and most likely builds on the underlying layers.
        - It provides a more robust interaction model between the different cloud entities on both the functional level and the semantic level.
- These characterizations and taxonomies reflect what is meant by cloud computing at the present time, but being in its start the phenomenon is constantly evolving, and the same will happen to the attempts to capture the real nature of cloud computing.

Cloud interoperability and standards:

- Cloud computing is a service-based model for delivering IT infrastructure and applications like utilities such as power, water, and electricity.
- To fully realize this goal, introducing standards and allowing interoperability between solutions offered by different vendors are the fundamental objectives.
- Vendor lock-in constitutes one of the major strategic barriers against the seamless adoption of cloud computing at all stages.

- Vendor lock-in can prevent a customer from switching to another competitor's solution, or when this is possible, it happens at considerable conversion cost and requires significant amounts of time.
    - This can occur either because the customer wants to find a more suitable solution for customer needs or because the vendor is no longer able to provide the required service.
- The presence of standards that are actually implemented and adopted could give room for interoperability and then lessen the risks resulting from vendor lock-in.
- The current state of standards and interoperability in cloud computing resembles the early Internet era.
- Yet the first steps toward a standardization process have been made, and a few organizations, such as the Cloud Computing Interoperability Forum (CCIF), the Open Cloud Consortium, and the DMTF Cloud Standards Incubator, are leading the path.
- Another interesting initiative is the Open Cloud Manifesto, which represents the point of view of various stakeholders on the benefits of open standards in the field.
- The standardization efforts are mostly concerned with the lower level of the cloud computing architecture, which is the most popular and developed.
    - In particular, in the IaaS market, the use of a proprietary virtual machine format constitutes the major reasons for the vendor lock-in, and efforts to provide virtual machine image compatibility.
    - The Open Virtualization Format (OVF) is an attempt to provide a common format for storing the information and metadata describing a virtual machine image.
- The challenge is providing standards for supporting the migration of running instances, thus allowing the real ability of switching from one infrastructure vendor to another in a completely transparent manner.
- Another direction in which standards try to move is planning a general reference architecture for cloud computing systems and providing a standard interface through which one can interact with them.
- At the moment the compatibility between different solutions is quite restricted, and the lack of a common set of APIs make the interaction with cloud-based solutions vendor specific.

Scalability and fault tolerance:

- The ability to scale on demand constitutes one of the most attractive features of cloud computing.

- Clouds allow scaling beyond the limits of the existing in-house IT resources, whether they are infrastructure (compute and storage) or applications services.
- To implement such a capability, the cloud middleware has to be designed with the principle of scalability along different dimensions in mind—for example, performance, size, and load.
- The cloud middleware manages a huge number of resource and users, which rely on the cloud to obtain the horsepower that they cannot obtain within the premises without bearing considerable administrative and maintenance costs.
- These costs are a reality for whomever develops, manages, and maintains the cloud middleware and offers the service to customers.
- In this scenario, the ability to tolerate failure becomes fundamental, sometimes even more important than providing an extremely efficient and optimized system.
- Hence, the challenge in this case is designing highly scalable and fault-tolerant systems that are easy to manage and at the same time provide competitive performance.

Security, trust, and privacy:

- Security, trust, and privacy issues are major obstacles for massive adoption of cloud computing.
- The traditional cryptographic technologies are used to prevent data tampering and access to sensitive information.
- The massive use of virtualization technologies exposes the existing system to new threats, which previously were not considered applicable.
    - For example, it might be possible that applications hosted in the cloud can process sensitive information. which can be stored using the most advanced technology in cryptography to protect data and then be considered safe.
- Since the application is hosted in a managed virtual environment it becomes accessible to the virtual machine manager that by program is designed to access the memory pages of such an application.
    - In this case, what is experienced is a lack of control over the environment in which the application is executed
- The lack of control over their own data and processes also poses severe problems for the trust we give to the cloud service provider and the level of privacy we want to have for our data.
- On one side we need to decide whether to trust the provider itself; on the other side, specific regulations can simply prevail over the agreement the provider is willing to establish with us concerning the privacy of the information managed on our behalf.

- Moreover, cloud services delivered to the end user can be the result of a complex stack of services that are obtained by third parties via the primary cloud service provider.
- The challenges are, then, mostly concerned with devising secure and trustable systems from different perspectives: technical, social, and legal.

Organizational aspects:

- Cloud computing introduces a significant change in the way IT services are consumed and managed.
- More precisely, storage, compute power, network infrastructure, and applications are delivered as metered services over the Internet.
- This introduces a billing model that is new within typical enterprise IT departments, which requires a certain level of cultural and organizational process maturity.
- In particular, a wide acceptance of cloud computing will require a significant change to business processes and organizational boundaries.
- Some interesting questions arise in considering the role of the IT department in this new scenario. In particular, the following questions have to be considered……
    - What is the new role of the IT department in an enterprise that completely or significantly relies on the cloud?
    - How will the compliance department perform its activity when there is a considerable lack of control over application workflows?
    - What are the implications (political, legal, etc.) for organizations that lose control over some aspects of their services?
    - What will be the perception of the end users of such services?

The Fundamentals of Cloud Architectures:

- Here we introduce and describes several of the more common foundational cloud architectural models, each demonstrating a common usage and characteristic of contemporary cloud-based environments.

1. Workload Distribution Architecture:

- IT resources can be horizontally scaled via the addition of one or more identical IT resources, and a load balancer that provides runtime logic capable of evenly distributing the workload among the available IT resources (Figure 11.1).
- The resulting workload distribution architecture reduces both IT resource overutilization and under-utilization to an extent dependent

upon the sophistication of the load balancing algorithms and runtime logic.

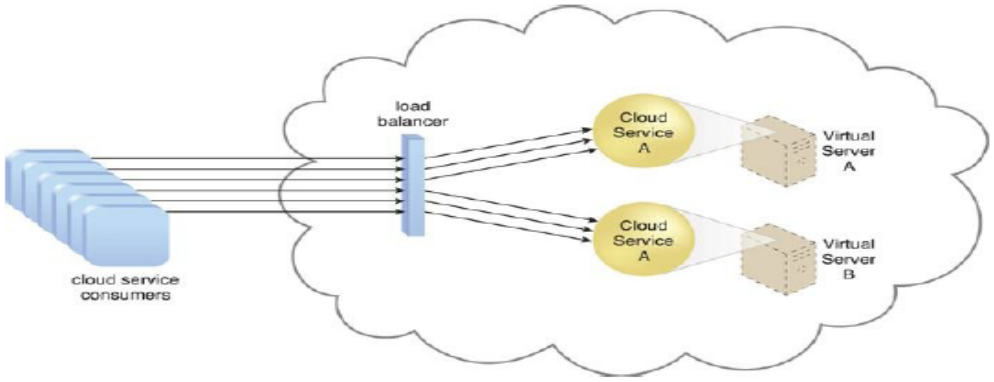


Figure 11.1. A redundant copy of Cloud Service A is implemented on Virtual Server B. The load balancer intercepts cloud service consumer requests and directs them to both Virtual Servers A and B to ensure even workload distribution.

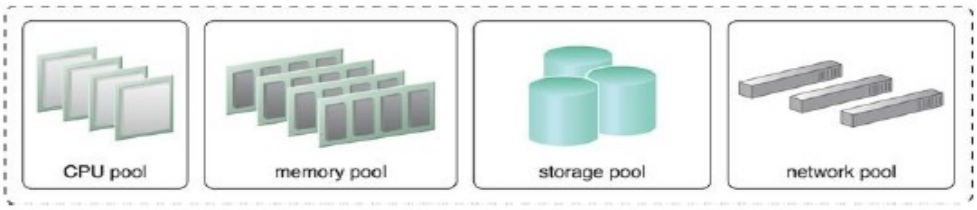
- This fundamental architectural model can be applied to any IT resource, with workload distribution commonly carried out in support of distributed virtual servers, cloud storage devices, and cloud services.
- Load balancing systems applied to specific IT resources usually produce specialized variations of this architecture that incorporate aspects of load balancing, such as:
- The service load balancing, the load balanced virtual server and the load balanced virtual switches architecture.

2. Resource Pooling Architecture:

- A resource pooling architecture is based on the use of one or more resource pools, in which identical IT resources are grouped and maintained by a system.
- The responsibility of the system is automatically ensures that they remain synchronized in order to serve the requests. The following are common examples of resource pools……
- **Physical server pools:**
  - Physical server pools are composed of networked servers that have been installed with operating systems and other necessary programs and/or applications and are ready for immediate use.
- **Virtual server pools:**
  - Virtual server pools are usually configured using one of several available templates chosen by the cloud consumer during provisioning.
  - For example, a cloud consumer can set up a pool of mid-tier Windows servers with 4 GB of RAM or a pool of low-tier Ubuntu servers with 2 GB of RAM.
- **Storage pools:**
  - Storage pools, or cloud storage device pools, consist of file-based or block-based storage structures that contain empty and/or filled cloud storage devices.

- **Network pools (or interconnect pools):**
  - Network pools (or interconnect pools) are composed of different preconfigured network connectivity devices.
  - For example, a pool of virtual firewall devices or physical network switches can be created for redundant connectivity, load balancing, or link aggregation.
- **CPU pools:**
  - CPU pools are ready to be allocated to virtual servers, and are typically broken down into individual processing cores.
  - Dedicated pools can be created for each type of IT resource and individual pools can be grouped into a larger pool, in which case each individual pool becomes a sub-pool (Figure 11.2).

Figure 11.2.



- Resource pools can become highly complex, with multiple pools created for specific cloud consumers or applications.
- A hierarchical structure can be established to form parent, sibling, and nested pools in order to facilitate the organization of diverse resource pooling requirements.

3. Dynamic Scalability Architecture:

- The dynamic scalability architecture is an architectural model based on a system of predefined scaling conditions that trigger the dynamic allocation of IT resources from resource pools.
- Dynamic allocation enables variable utilization as dictated by usage demand fluctuations, since unnecessary IT resources are efficiently reclaimed without requiring manual interaction.
- The automated scaling listener is configured with workload thresholds that dictate when new IT resources need to be added to the workload processing.
  - This mechanism can be provided with logic that determines how many additional IT resources can be dynamically provided, based on the terms of a given cloud consumer's provisioning contract.
- The following types of dynamic scaling are commonly used:
  - **Dynamic Horizontal Scaling** – IT resource instances are scaled out and in to handle fluctuating workloads. The automatic scaling listener monitors requests and signals resource replication to initiate IT resource duplication, as per requirements and permissions.
  - **Dynamic Vertical Scaling** – IT resource instances are scaled up and down when there is a need to adjust the processing capacity of a single IT resource. For example, a virtual server that is being

overloaded can have its memory dynamically increased or it may have a processing core added.

- **Dynamic Relocation** – The IT resource is relocated to a host with more capacity. For example, a database may need to be moved from a tape-based SAN storage device with 4 GB per second I/O capacity to another disk-based SAN storage device with 8 GB per second I/O capacity.

- Figures 11.5 to 11.7 illustrate the process of dynamic horizontal scaling. The brief steps are as follows……
  - Cloud service consumers are sending requests to a cloud service.
  - The automated scaling listener monitors the cloud service to determine if predefined capacity thresholds are being exceeded.
  - The number of requests coming from cloud service consumers increases.
  - The workload exceeds the performance thresholds. The automated scaling listener determines the next course of action based on a predefined scaling policy.
  - If the cloud service implementation is deemed eligible for additional scaling, the automated scaling listener initiates the scaling process.
  - The automated scaling listener sends a signal to the resource replication mechanism.
  - The Resource replication mechanism which creates more instances of the cloud service.
  - Now that the increased workload has been accommodated, the automated scaling listener resumes monitoring and detracting and adding IT resources, as required.

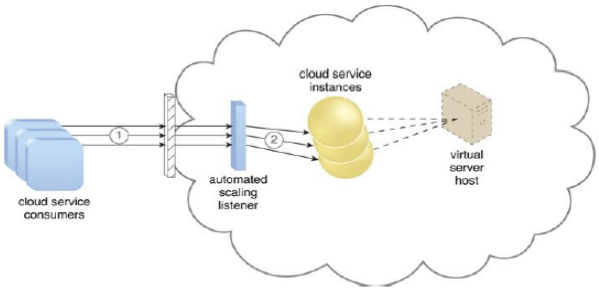
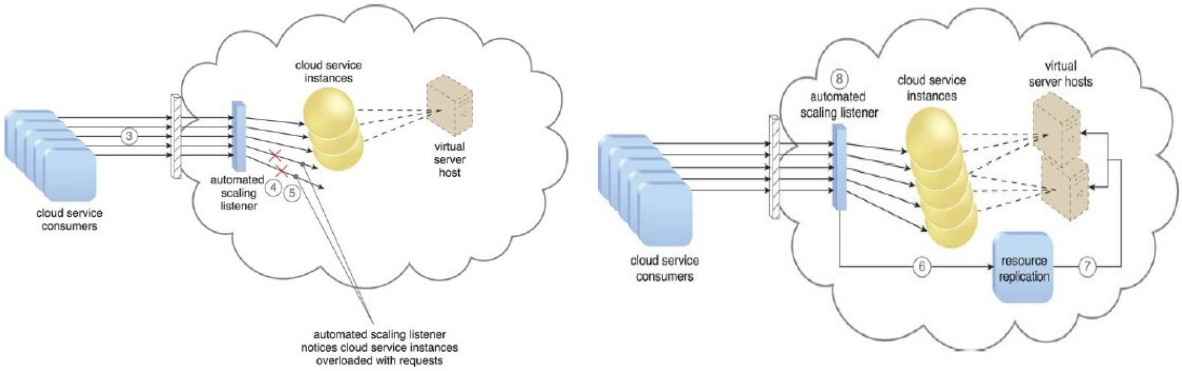


Figure 11.5. Cloud service consumers are sending requests to a cloud service (1). The automated scaling listener monitors the cloud service to determine if predefined capacity thresholds are being exceeded (2).

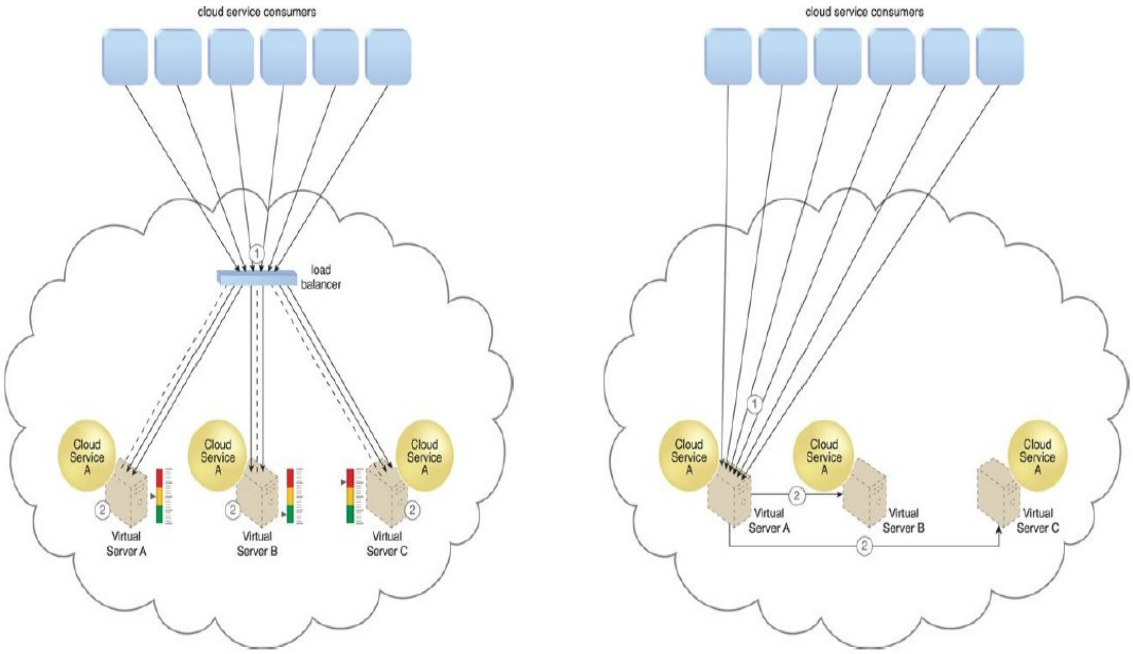


- Besides the core automated scaling listener and resource replication mechanisms, the following mechanisms can also be used in this form of cloud architecture.

- **Cloud Usage Monitor** – Specialized cloud usage monitors can track runtime usage in response to dynamic fluctuations caused by this architecture.
- **Hypervisor** – The hypervisor is invoked by a dynamic scalability system to create or remove virtual server instances, or to be scaled itself.
- **Pay-Per-Use Monitor** – The pay-per-use monitor is engaged to collect usage cost information.

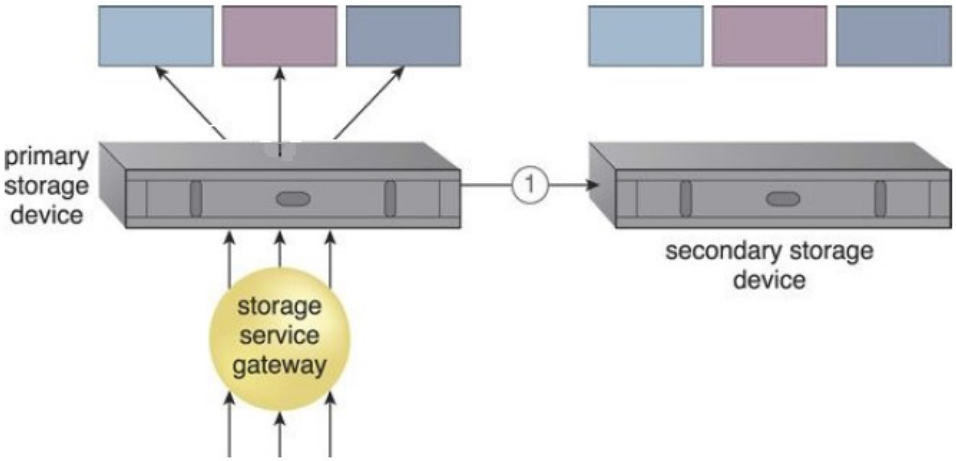
4. Service Load Balancing Architecture:

- The service load balancing architecture can be considered a specialized variation of the workload distribution architecture that is geared specifically for scaling cloud service implementations.
- Redundant deployments of cloud services are created, with a load balancing system added to dynamically distribute workloads.
- The duplicate cloud service implementations are organized into a resource pool, while the load balancer is positioned as either an external or built-in component to allow the host servers to balance.
- Depending on the anticipated workload and processing capacity multiple instances of each cloud service implementation can be generated.
- The load balancer can be positioned either independent of the cloud services and their host servers (Figure 11.10), or built-in as part of the application or server's environment.
  - The load balancer intercepts messages sent by cloud service consumers and
  - Forwards them to the virtual servers so that the workload processing is horizontally scaled.
- In the latter case, a primary server with the load balancing logic can communicate with neighboring servers to balance the workload (Figure 11.11).
  - Cloud service consumer requests are sent to Cloud Service A on Virtual Server A.
  - The cloud service implementation includes built-in load balancing logic that is capable of distributing requests to the neighboring Cloud Service A implementations on Virtual Servers B and C.
- The service load balancing architecture can involve the following mechanisms in addition to the load balancer.
  - **Cloud Usage Monitor:** Cloud usage monitors may be involved with monitoring cloud service instances and their respective IT resource consumption levels, as well as various runtime monitoring and usage data collection tasks.
  - **Resource Cluster** – Active-active cluster groups are incorporated in this architecture to help balance workloads across different members of the cluster.
  - **Resource Replication** – The resource replication mechanism is utilized to generate cloud service implementations in support of load balancing requirements.



5. Redundant Storage Architecture:

- Cloud storage devices are occasionally subject to failure and disruptions that are caused by network connectivity issues, controller or general hardware failure, or security breaches.
- A compromised cloud storage device's reliability can have a ripple effect and cause impact failure across all of the services, applications, and infrastructure components in the cloud that are reliant on its availability.
- The redundant storage architecture introduces a secondary duplicate cloud storage device as part of a failover system that synchronizes its data with the data in the primary cloud storage device.
- Note: A logical unit number (LUN) is a logical drive that represents a partition of a physical drive.
- A storage service gateway diverts cloud consumer requests to the secondary device whenever the primary device fails (Figures 11.16 and 11.17).



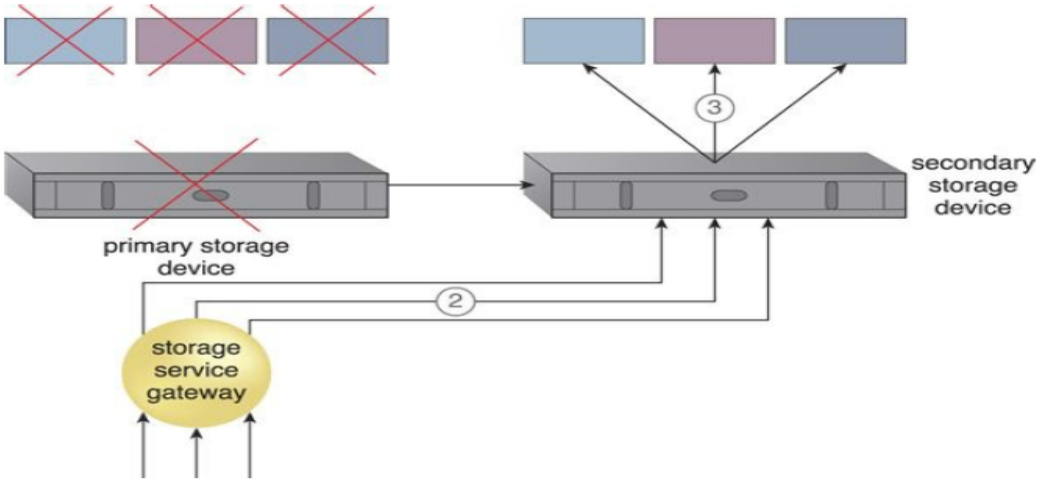


Figure 11.17. The primary storage becomes unavailable and the storage service gateway forwards the cloud consumer requests to the secondary storage device (2). The secondary storage device forwards the requests to the LUNs, allowing cloud consumers to continue to access their data (3).

- This cloud architecture primarily relies on a storage replication system that keeps the primary cloud storage device synchronized with its duplicate secondary cloud storage devices.
- Cloud providers may locate secondary cloud storage devices in a different geographical region than the primary cloud storage device, usually for economic reasons.
- In this case, cloud providers may need to lease a network connection via a third-party cloud provider in order to establish the replication between the two devices
- Some cloud providers use storage devices with dual array and storage controllers to improve device redundancy.

An Overview of Cloud Security:

- Information security is a complex group of techniques, technologies, regulations, and behaviors that collaboratively protect the integrity of and access to computer systems and data.
- IT security measures aim to defend against threats and interference that arise from both malicious intent and unintentional user error.
- We will first define fundamental security terms relevant to cloud computing and describe associated concepts.

1. Confidentiality:

- Confidentiality is the characteristic of something being made accessible only to authorized parties (Figure 6.1).
- Within cloud environments, confidentiality primarily relates to restricting access to data in transit and storage.

2. Integrity:

- Integrity is the characteristic of not having been altered by an unauthorized party (Figure 6.2).

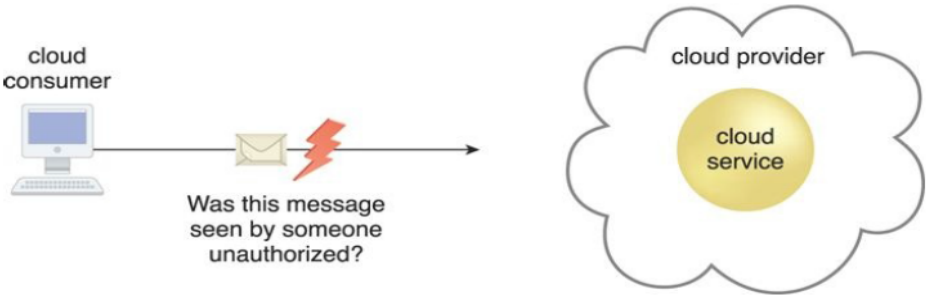


Figure 6.1. The message issued by the cloud consumer to the cloud service is considered confidential only if it is not accessed or read by an unauthorized party.

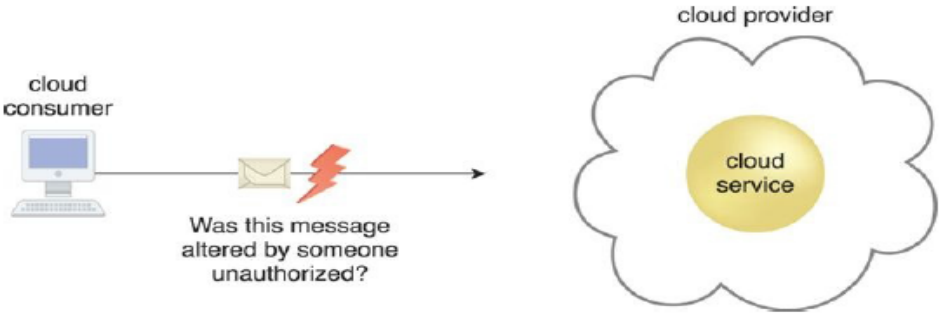


Figure 6.2. The message issued by the cloud consumer to the cloud service is considered to have integrity if it has not been altered.

- An important issue that concerns data integrity in the cloud is whether a cloud consumer can be guaranteed that the data it transmits to a cloud service matches the data received by that cloud service.
- Integrity can extend to how data is stored, processed, and retrieved by cloud services and cloud-based IT resources.

3. Authenticity:

- Authenticity is the characteristic of something having been provided by an authorized source.
- This concept encompasses non-repudiation, which is the inability of a party to deny or challenge the authentication of an interaction.
- Authentication in non-repudiable interactions provides proof that these interactions are uniquely linked to an authorized source.
- For example, a user may not be able to access a non-repudiable file after its receipt without also generating a record of this access.

4. Availability:

- Availability is the characteristic of being accessible and usable during a specified time period.
- In typical cloud environments, the availability of cloud services can be a responsibility that is shared by the cloud provider and the cloud carrier.
- The availability of a cloud-based solution that extends to cloud service consumers is further shared by the cloud consumer.

5. Threat:

- A threat is a potential security violation that can challenge defenses in an attempt to breach privacy and/or cause harm.

- Both manually and automatically instigate threats are designed to exploit known weaknesses, also referred to as vulnerabilities.
- A threat that is carried out results in an attack.

6. **Vulnerability:**

- A vulnerability is a weakness that can be exploited either because it is protected by insufficient security controls, or because existing security controls are overcome by an attack.
- IT resource vulnerabilities can have a range of causes, including configuration deficiencies, security policy weaknesses, user errors, hardware or firmware flaws, software bugs, and poor security architecture.

7. **Risk:**

- Risk is the possibility of loss or harm arising from performing an activity.
- Risk is typically measured according to its threat level and the number of possible or known vulnerabilities.
- Two metrics that can be used to determine risk for an IT resource are:
- The probability of a threat occurring to exploit vulnerabilities in the IT resource.
- The expectation of loss upon the IT resource being compromised.

8. **Security Controls:**

- Security controls are countermeasures used to prevent or respond to security threats and to reduce or avoid risk.
- Details on how to use security countermeasures are typically outlined in the security policy,
- It contains a set of rules and practices specifying how to implement a system, service, or security plan for maximum protection of sensitive and critical IT resources.

9. **Security Mechanisms:**

- Countermeasures are typically described in terms of security mechanisms.
- Security mechanisms are components comprising a defensive framework that protects IT resources, information, and services.

10. **Security Policies:**

- A security policy establishes a set of security rules and regulations.
- Often, security policies will further define how these rules and regulations are implemented and enforced.
- For example, the positioning and usage of security controls and mechanisms can be determined by security policies.

Threat Agents:

- A threat agent is an entity that poses a threat because it is capable of carrying out an attack.

- Cloud security threats can originate either internally or externally, from humans or software programs.
- Figure 6.3 illustrates the role a threat agent assumes in relation to vulnerabilities, threats, and risks, and the safeguards established by security policies and security mechanisms.

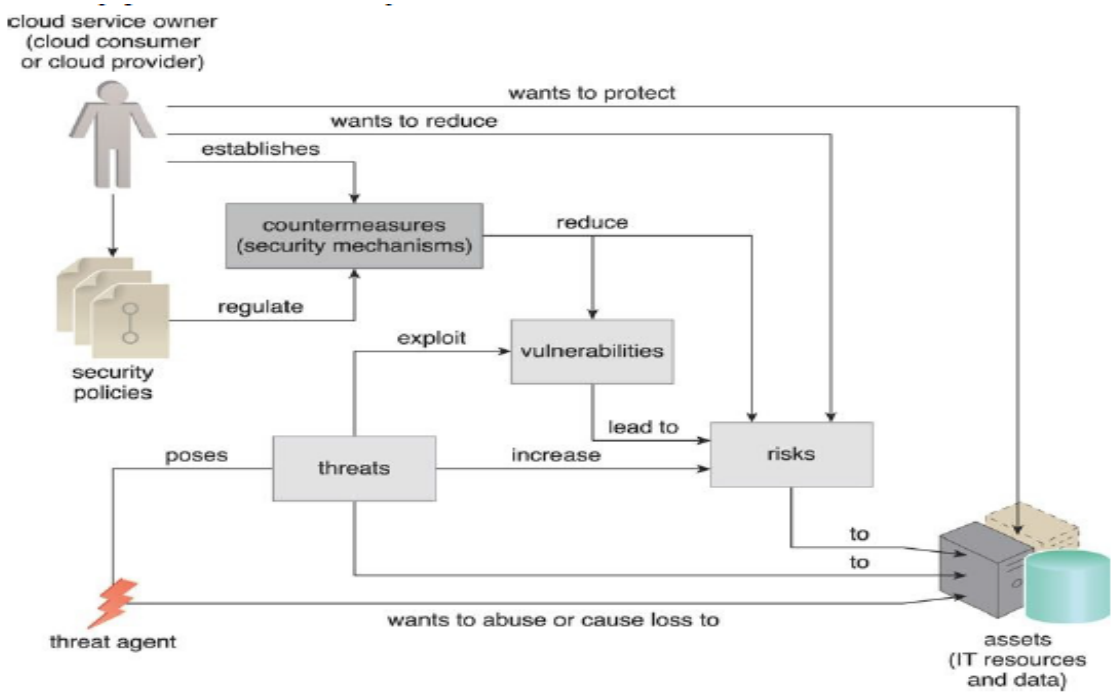


Figure 6.3. How security policies and security mechanisms are used to counter threats, vulnerabilities, and risks caused by threat agents.

Anonymous Attacker:

- An anonymous attacker is a non-trusted cloud service consumer without permissions in the cloud.
- It typically exists as an external software program that launches network-level attacks through public networks.
- When anonymous attackers have limited information on security policies and defenses, it can inhibit their ability to formulate effective attacks.
- Therefore, anonymous attackers often resort to committing acts like bypassing user accounts or stealing user credentials, while using methods that either ensure anonymity or require substantial resources for prosecution.

Malicious Service Agent:

- A malicious service agent is able to intercept and forward the network traffic that flows within a cloud.
- It typically exists as a service agent (or a program pretending to be a service agent) with compromised or malicious logic.
- It may also exist as an external program able to remotely intercept and potentially corrupt message contents.

Trusted Attacker:

- A trusted attacker shares IT resources in the same cloud environment as the cloud consumer and attempts to exploit legitimate credentials to target cloud providers and the cloud tenants with whom they share IT resources.
- The trusted attackers usually launch their attacks from within a cloud's trust boundaries by abusing legitimate credentials or via the appropriation of sensitive and confidential information.
- Trusted attackers (also known as malicious tenants) can use cloud-based IT resources for a wide range of exploitations, including……
- The hacking of weak authentication processes, the breaking of encryption, the spamming of e-mail accounts, or to launch common attacks, such as denial of service campaigns.

Malicious Insider:

- Malicious insiders are human threat agents acting on behalf of or in relation to the cloud provider.
- They are typically current or former employees or third parties with access to the cloud provider's premises.
- This type of threat agent carries tremendous damage potential, as the malicious insider may have administrative privileges.

Cloud Security Threats:

- Here introduces several common threats and vulnerabilities in cloud-based environments and describes the roles of the above-mentioned threat agents.

Traffic Eavesdropping:

- Traffic eavesdropping occurs when data being transferred to or within a cloud (usually from the cloud consumer to the cloud provider) is passively intercepted by a malicious service agent for illegitimate information gathering purposes (Fig. 6.8)
- The aim of this attack is to directly compromise the confidentiality of the data and, possibly, the confidentiality of the relationship between the cloud consumer and cloud provider.
- Because of the passive nature of the attack, it can more easily go undetected for extended periods of time.

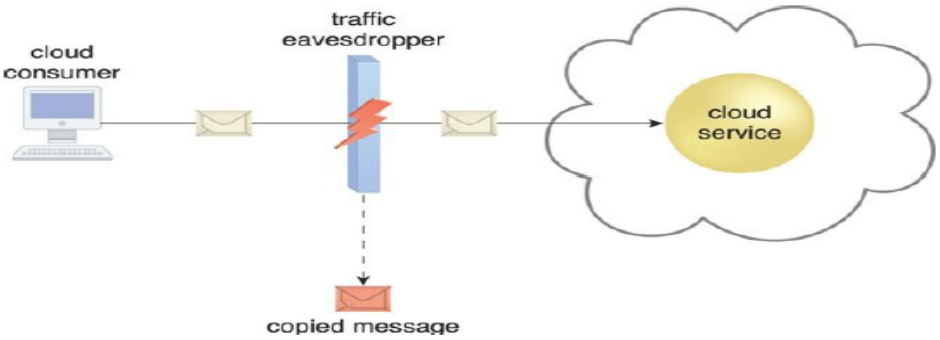


Figure 6.8. An externally positioned malicious service agent carries out a traffic eavesdropping attack by intercepting a message sent by the cloud service consumer to the cloud service. The service agent makes an unauthorized copy of the message before it is sent along its original path to the cloud service.

Malicious Intermediary:

- The malicious intermediary threat arises when messages are intercepted and altered by a malicious service agent, thereby potentially compromising the message's confidentiality and/or integrity.
- It may also insert harmful data into the message before forwarding it to its destination. Below figure illustrates a e.g., of malicious intermediary attack.

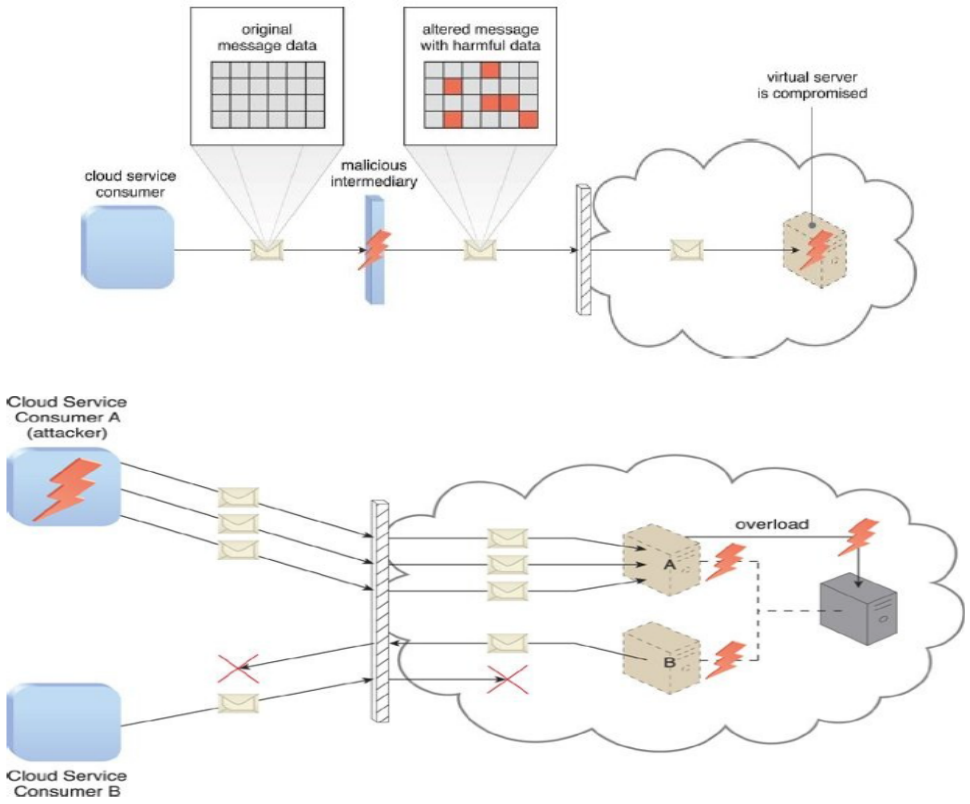


Figure 6.10. Cloud Service Consumer A sends multiple messages to a cloud service (not shown) hosted on Virtual Server A. This overloads the capacity of the underlying physical server, which causes outages with Virtual Servers A and B. As a result, legitimate cloud service consumers, such as Cloud Service Consumer B, become unable to communicate with any cloud services hosted on Virtual Servers A and B.

Virtualization Attack:

- Virtualization provides multiple cloud consumers with access to IT resources that share underlying hardware but are logically isolated from each other.
- Because cloud providers grant cloud consumers administrative access to virtualized IT resources (such as virtual servers), there is an inherent risk that cloud consumers could abuse this access to attack the underlying physical IT resources.
- A virtualization attack exploits vulnerabilities in the virtualization platform to jeopardize its confidentiality, integrity, and/or availability.
- This threat is illustrated in Figure 6.13, where a trusted attacker successfully accesses a virtual server to compromise its underlying physical server.

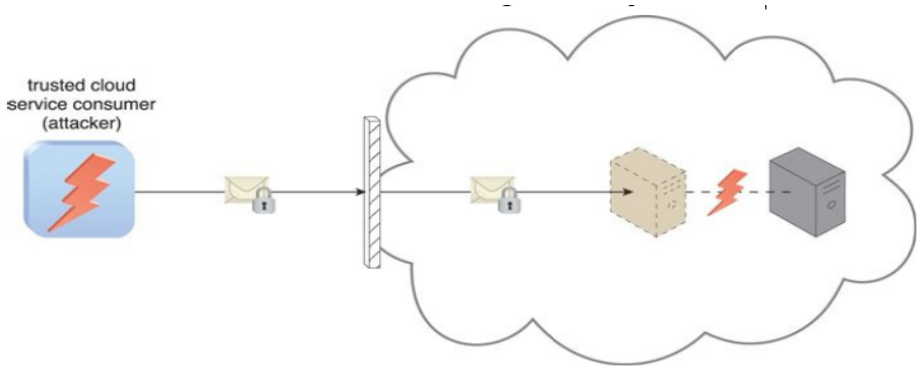


Figure 6.13. An authorized cloud service consumer carries out a virtualization attack by abusing its administrative access to a virtual server to exploit the underlying hardware.

- With public clouds, where a single physical IT resource may be providing virtualized IT resources to multiple cloud consumers, such an attack can have significant repercussions.

References:

- Mastering Cloud Computing: Foundations and Applications Programming – Rajkumar Buyya – Tata Mcgraw Hill publishing.
- Cloud Computing - Concepts, Technology, Security, and Architecture - Second Edition - Thomas Erl, Eric Barcelo – Pearson.

Sample Questions:

1. Explain briefly the Cloud Computing architecture with relevant diagram.
2. Explain the Infrastructure and hardware-as-a-service with a relevant diagram.
3. Define PaaS. Explain the essential characteristics that identify a PaaS solution.
4. SaaS applications are naturally multitenant, Justify with its characteristics and benefits.
5. Discuss the few challenges faced by Cloud computing at the initial stages.
6. Explain the Workload Distribution Architecture with a neat diagram.
7. Explain how Resource Pooling Architecture is beneficial with different Resource pools.
8. Explain the role of Automated Scaling Listener with its working in the Dynamic Scalability architecture.
9. Explain how Service Load Balancing Architecture is differs from Work load balance architecture.

10. The redundant storage architecture introduces a secondary duplicate cloud storage device as part of a failover system. Justify with its advantages.
11. Explain any five fundamental security terms relevant to cloud computing.
12. Explain the role of a threat agent in context of Cloud security with relevant diagram.
13. Define Threat Agent. Explain the different types of Threat Agents.
14. Explain briefly:

    a. Traffic Eavesdropping b. Malicious Intermediary
15. Explain briefly:

    a. Denial of Service b. Virtualization Attack

******