Chap 2. Cloud Architecture

2.1 Cloud Delivery Models

2.2 Cloud Deployment Models

Chap 4. Cloud Working Mechanisms

4.1 Cloud Infrastructure Mechanisms

Cloud infrastructure mechanisms are foundational building blocks of cloud environments that establish primary artifacts to form the basis of fundamental cloud technology architecture. Beyond a vast array of cloud infrastructure mechanisms, this chapter dives into **virtualization**, the most foundational building blocks of cloud environments.

A cloud delivery model represents a specific, pre-packaged combination of IT resources offered by a cloud provider.

The following cloud infrastructure mechanisms are described:

- Virtual Server
- Hypervisor
- Resource Management System

4.1.1 Virtual Server

A *virtual server* is a form of virtualization software that **emulates a physical server**. Virtual servers are used by cloud providers to share the same physical server with multiple cloud consumers by providing individual virtual server instances. Each virtual server can host numerous IT resources, cloud-based solutions, and various other cloud computing mechanisms.

Figure 4.1 shows three virtual servers being hosted by two physical servers. The number of instances a given physical server can share is limited by its capacity.

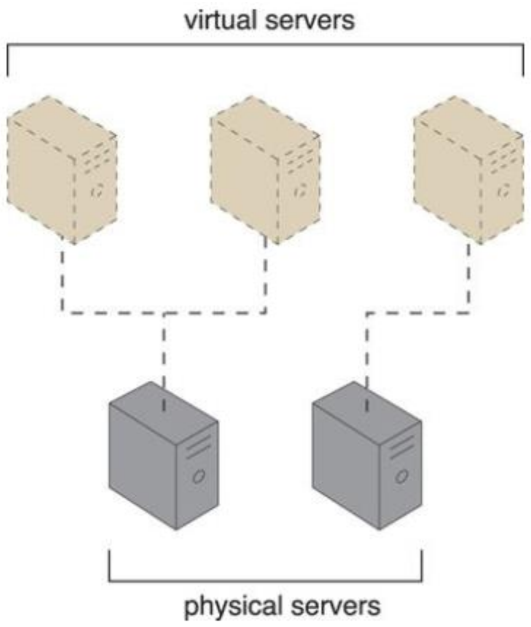

Figure 4.1 The first physical server hosts two virtual servers, while the second physical server hosts one virtual server.

Virtualization technologies include virtual machine techniques, and virtual networks.

- Virtual machines provide virtualized IT-infrastructures on-demand (such as VMware, Xen)
- Virtual networks support users with a customized network environment to access cloud resources (such as VPN)

4.1.2 Hypervisor

The *hypervisor* mechanism is primarily used to generate and manage virtual server instances (VMs) of a physical server. Hypervisor software is installed directly on bare-metal servers and provide a layer of abstraction between the physical hardware and the virtualized environments running on top of it.

Hypervisor’s working mechanism:

- **Abstraction**: The hypervisor abstracts the physical hardware resources such as CPU, memory, storage, and networking, presenting them as virtual resources to the guest operating systems running on the VMs.
- **Isolation**: Each VM operates independently of the others, with its own guest operating system and applications. The hypervisor ensures that each VM is isolated from one another, providing security and stability.
- **Resource Allocation**: Hypervisors allocate and manage physical hardware resources among the VMs. They can dynamically adjust resource allocations based on the needs of each VM, ensuring efficient utilization of hardware resources.

- **Virtual Hardware**: To each VM, the hypervisor presents virtual hardware that appears as dedicated resources. For example, each VM may have its own virtual CPU cores, memory, disk space, and network interfaces, even though they are all sharing the underlying physical hardware.
- **Control and Management**: Hypervisors provide features for controlling and managing the VMs, such as starting, stopping, pausing, and migrating VMs between physical hosts.

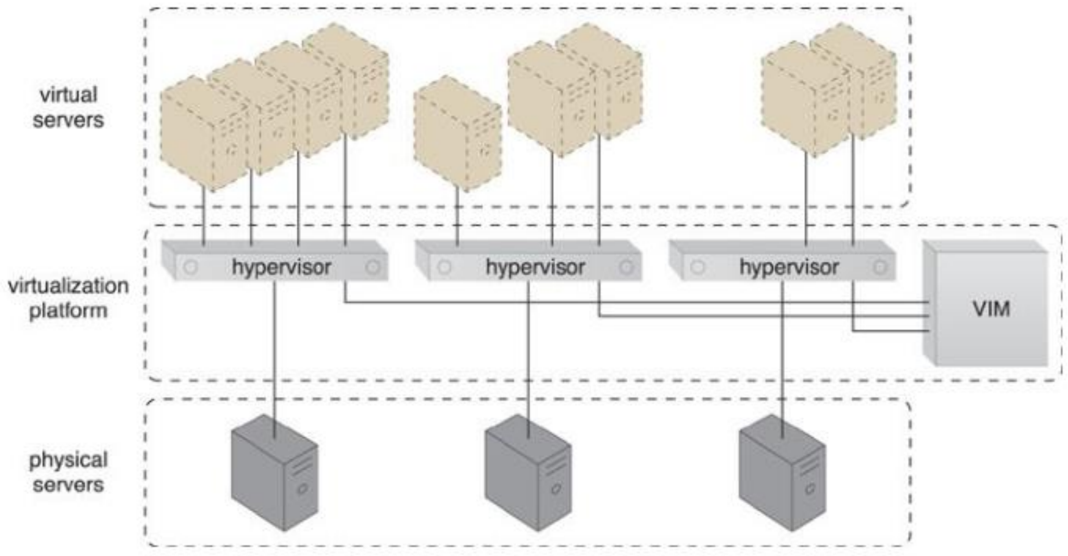

Figure 4.2. Virtual servers are created via individual hypervisor on individual physical servers. All three hypervisors are jointly controlled by the same VIM.

4.1.3 Resource Management System

The *resource management system* mechanism helps **coordinate IT resources** in response to management actions performed by both cloud consumers and cloud providers. Core to this system is the Virtual Infrastructure Manager (VIM), a software tool that provides a centralized platform for administering multiple hypervisors across physical servers. The primary purpose of VIM is to simplify the management and monitoring of virtual servers and their associated resources, regardless of the underlying hypervisor technology being used.

Resource management systems typically expose APIs that allow cloud providers to build remote administration system portals that can be customized to selectively offer resource management controls to external cloud resource administrators acting on behalf of cloud consumer organizations via usage and administration portals

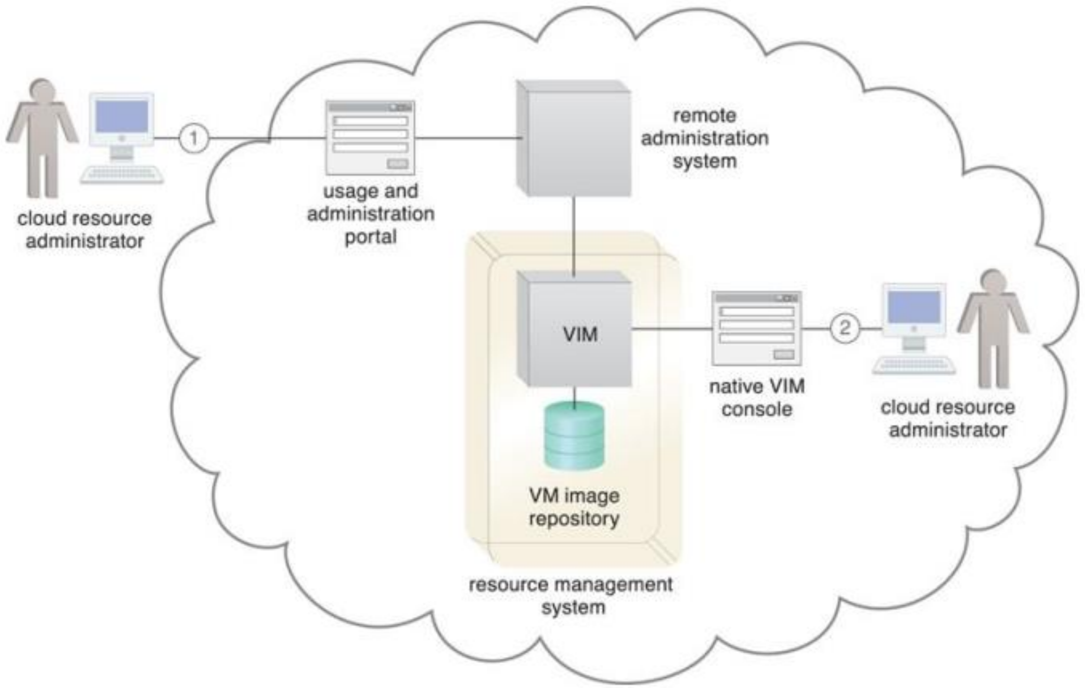
cloud resource administrator
1
usage and administration portal
remote administration system
VIM
VM image repository
resource management system
native VIM console
2
cloud resource administrator

Figure 4.3. The cloud consumer's cloud resource administrator accesses a usage and administration portal externally to administer a leased IT resource (1). The cloud provider's cloud resource administrator uses the native user-interface provided by the VIM to perform internal resource management tasks (2)

4.2 Specialized Cloud Mechanisms

A typical cloud technology architecture contains numerous moving parts to address distinct usage requirements of IT resources and solutions. Each mechanism covered in this chapter fulfills a specific runtime function in supportof one or more cloud characteristics.

The following specialized cloud mechanisms are described in this chapter:

- Automated Scaling Listener
- Load Balancer
- Failover system

All these mechanisms can be considered extensions to cloud infrastructure and can be combined in numerous ways as part of distinct and custom technology architectures.

4.2.1 Automated Scaling Listener

The *automated scaling listener* mechanism is a service agent that monitors and tracks communications between cloud service consumers and cloud services for dynamic scaling purposes. Workloads can be determined by the volume of cloud consumer-generated requests or via back-end processing demands triggered by certain types of requests.

Types of responses to workload fluctuation conditions:

- **Auto-scaling**: Automatically scale out/in based on parameters previously defined by the cloud consumer.
- **Automatic notification** (of the cloud consumer): when workloads exceed current thresholds or fall below allocated resources.
    - ➔ Cloud consumers can choose to adjust its current IT resource allocation.

4.2.2 Load balancer

A common approach to horizontal scaling is to **balance a workload** across two or more IT resources to increase performance and capacity beyond what a single IT resource can provide. A load balancer is programmed or configured with a set of performance and Quality of Service (QoS) rules and parameters with the general objectives of optimizing IT resource usage, avoiding overloads, and maximizing throughput.

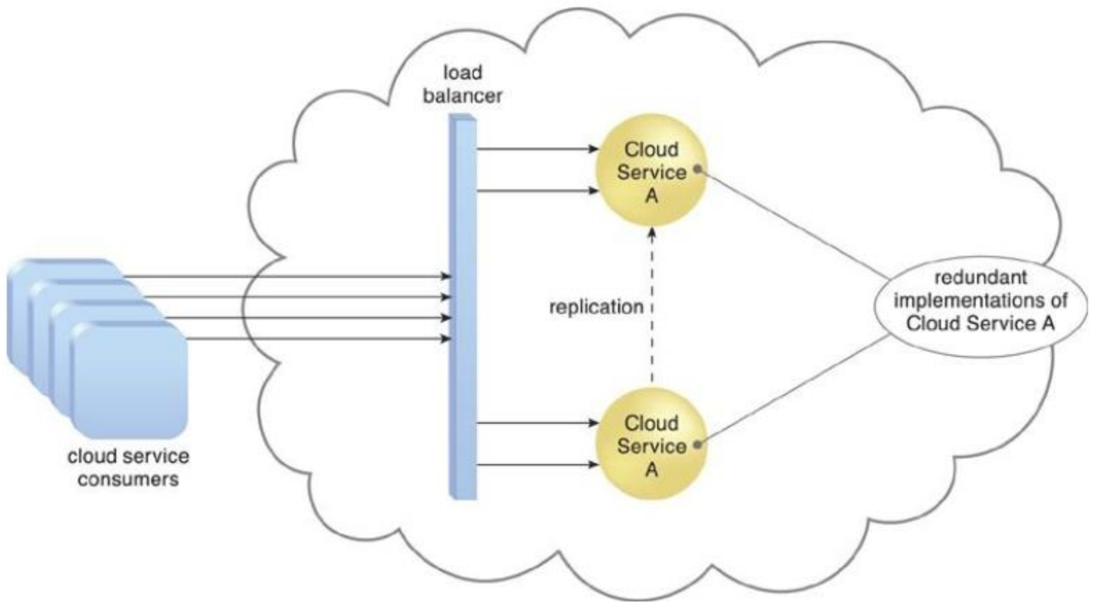

Figure 4.4. A load balancer implemented as a service agent transparently distributes incoming workload request messages across two redundant cloud service implementations, which in turn maximizes performance for the cloud service consumers.

4.2.3 Failover System

The failover system mechanism is used to increase the reliability and availability of IT resources by using established clustering technology to provide redundant implementations. A failover system is configured to automatically switch over to a redundant or standby IT resource instance whenever the currently active IT resource becomes unavailable. Failover systems come in **two basic configurations**:

Active-Active

In an active-active configuration, redundant cloud service implementations of the IT resource actively serve the workload synchronously **(Figure 4.5.1).** When a failure is detected, the failed instance is removed from the load balancing scheduler **(Figure 4.5.2).**

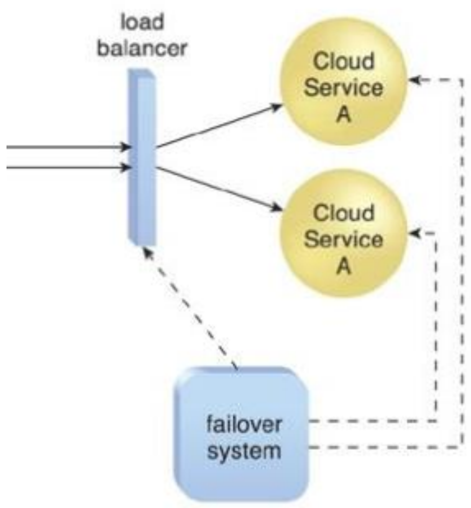

Figure 4.5.1

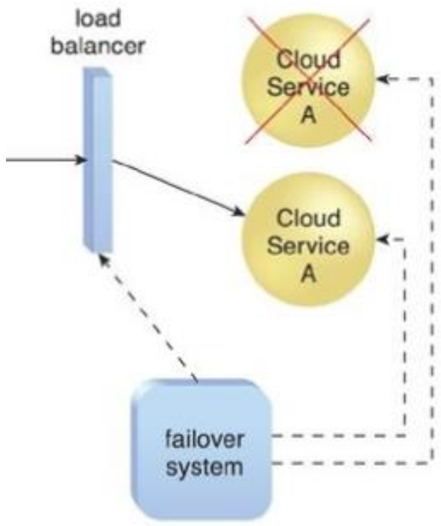

Figure 4.5.2

Active-Passive

In an active-passive configuration, one cloud service implementation acts as the active instance, receiving cloud service requests, while the other acts as the standby instance **(Figure 4.6.1).** When a failure is detected, the standby or inactive implementation is activated to take over the processing from the IT resource that becomes unavailable, and the corresponding workload is redirected to the instance taking over the operation **(Figure 4.7.1).**

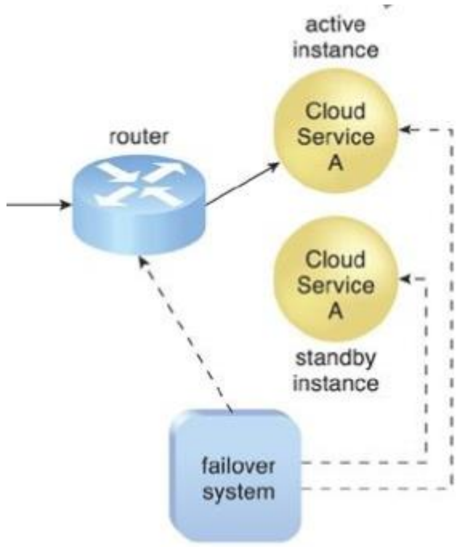

Figure 4.6.1

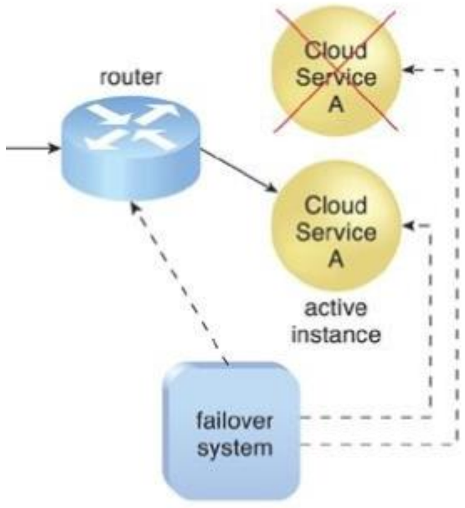

Figure 4.6.2