Lecture 08

Fundamental Cloud Architectures

Disclaimer

- All figures in this presentation are taken from Cloud Computing by Thomas Erl, Zaigham Mahmood, and Ricardo Puttini, (ISBN: 0133387526) Copyright © 2013 Arcitura Education, Inc. All rights reserved.

Fundamental Architectures

- Workload Distribution
- Resource Pooling
- Dynamic Scalability
- Elastic Resource Capacity
- Service Load Balancing
- Cloud Bursting
- Elastic Disk Provisioning
- Redundant Storage

Workload Distribution Architecture

Workload Distribution Architecture

- Reduces both IT resource over-utilization and under-utilization, depending on the sophistication of the load balancing algorithm and runtime logic.
- Work on balancing loads is commonly carried out on:
  - Distributed Virtual Servers
  - Cloud Storage Devices
  - Cloud Services / Applications
- Specialized variations according to specific IT resources, such as:
  - Service load balancing
  - Load balanced virtual server architecture
  - Load balanced virtual switches architecture

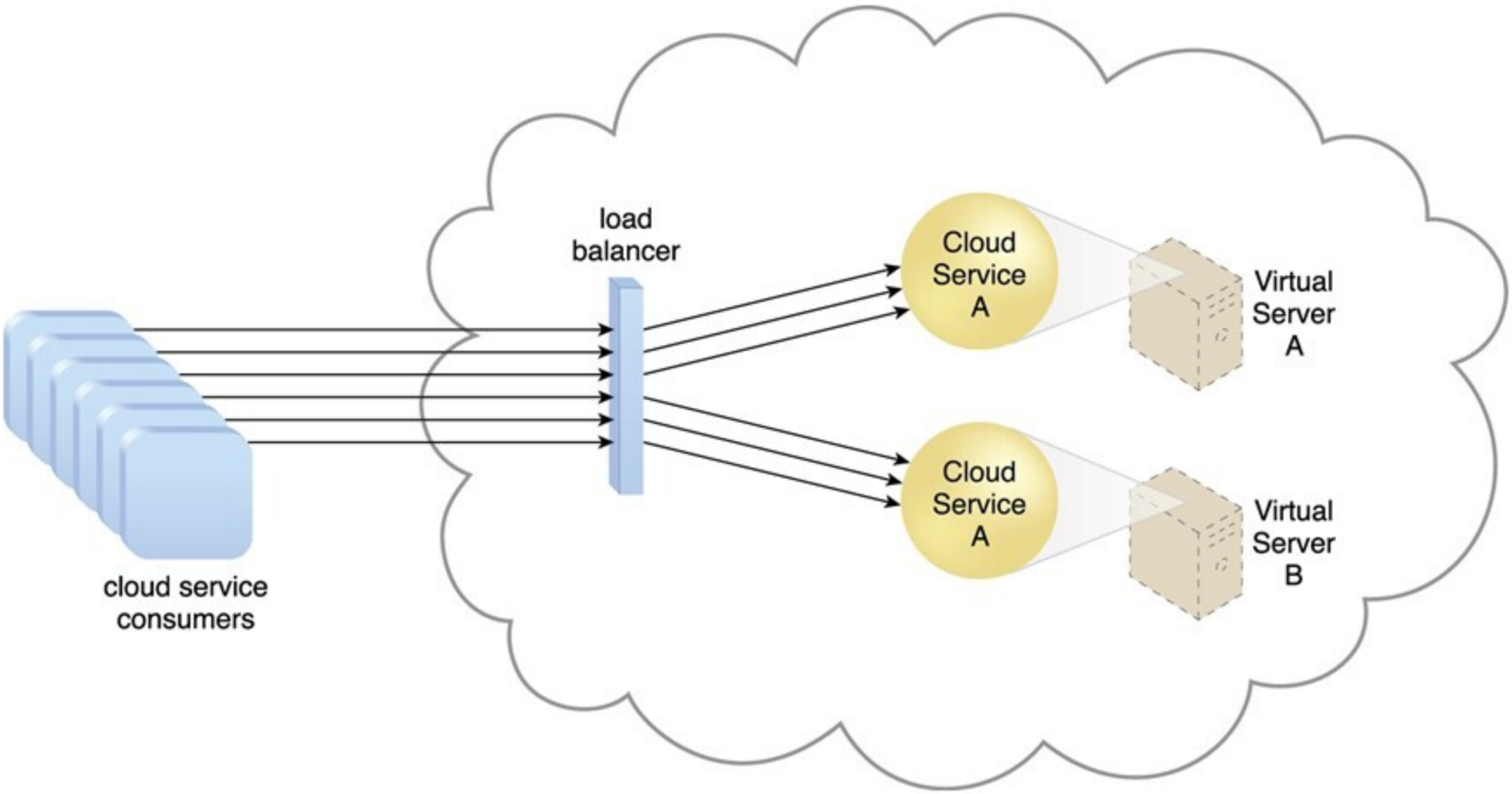


Figure 11.1 A redundant copy of Cloud Service A is implemented on Virtual Server B. The load balancer intercepts cloud service consumer requests and directs them to both Virtual Servers A and B to ensure even workload distribution.

Workload Distribution Architecture

- Cloud mechanisms applied:
  - Audit monitor – To determine whether monitoring is necessary to fulfill legal and regulatory requirements.
  - Cloud usage monitor – Workload tracking and data processing.
  - Hypervisor – Distribution of workloads between hypervisors and virtual servers.
  - Logical network perimeter – Isolates cloud consumer network boundaries in relation to how and where workloads are distributed.
  - Resource cluster – Support workload balancing between different cluster nodes.
  - Resource replication – Generate new instances of virtualized IT resources in response to runtime workload distribution demands.

Resource Pooling Architecture

Resource Pooling Architecture

- Based on the use of one or more resource pools, in which identical IT resources are grouped and maintained by a system that automatically ensures that they remain synchronized.

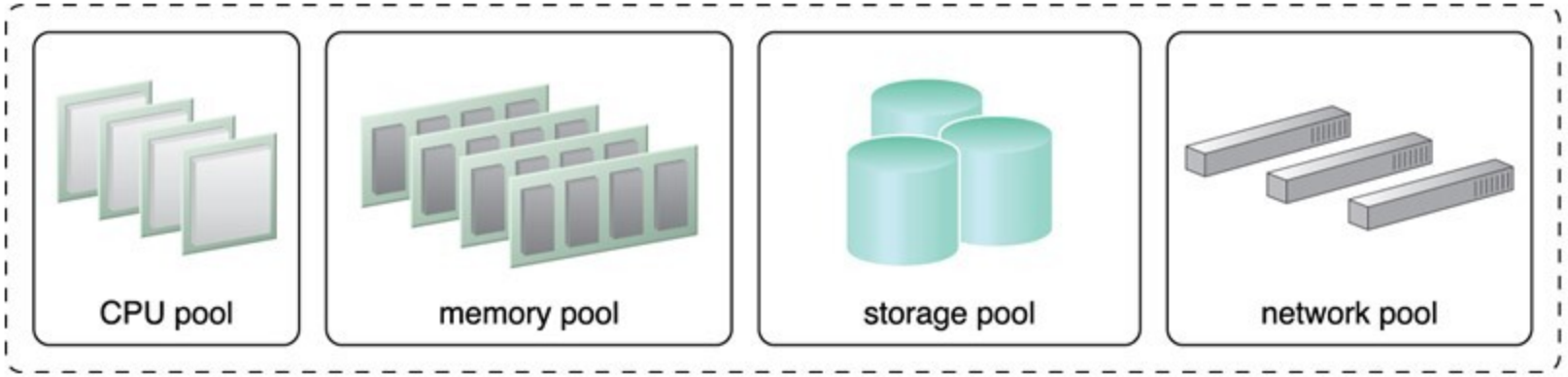


Figure 11.2 A sample resource pool that is comprised of four sub-pools of CPUs, memory, cloud storage devices, and virtual network devices.

Common Examples of Resource Pools

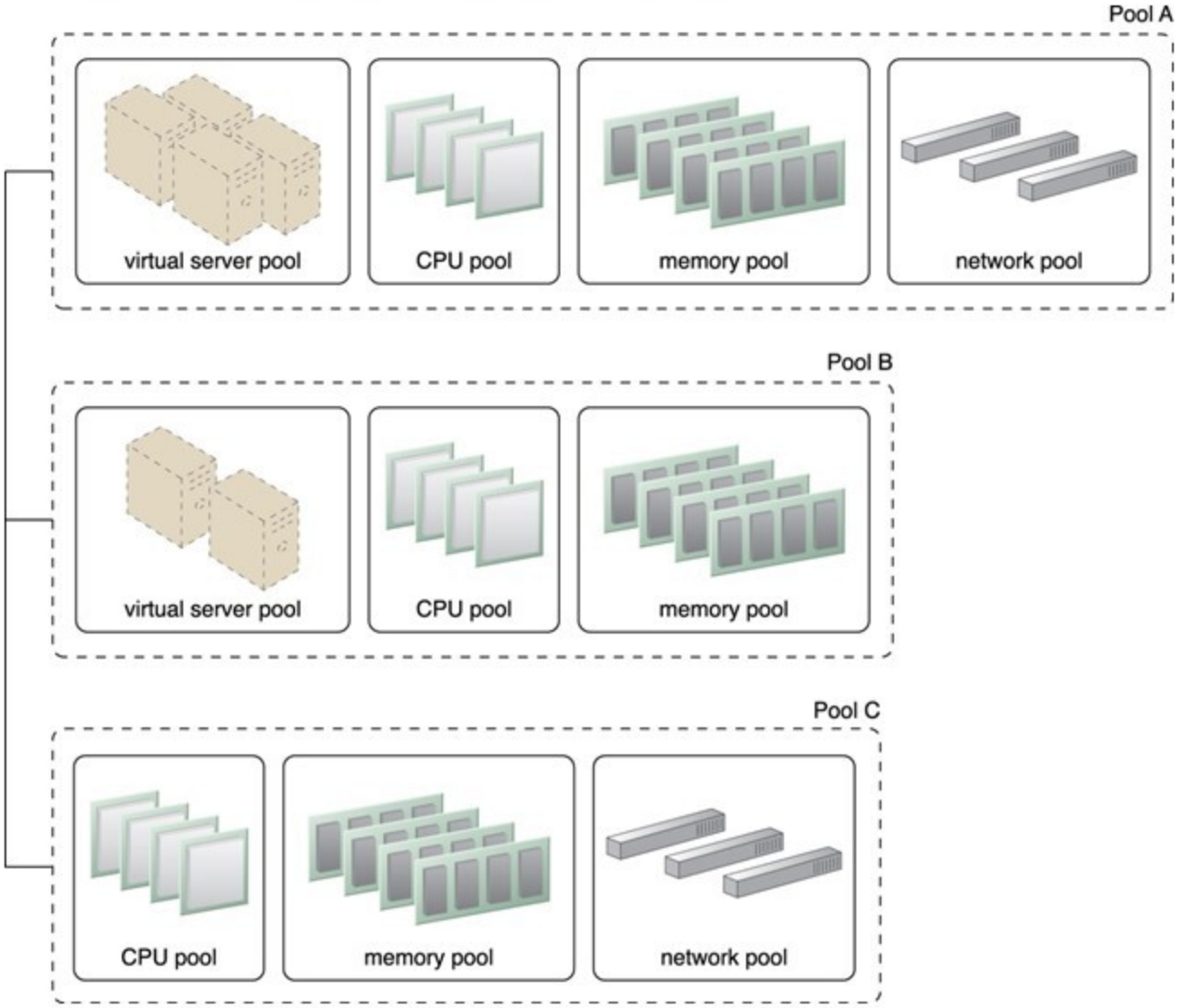
- Physical server pools – networked servers ready for immediate use.
- Virtual server pools – configured using one of several available templates chosen by cloud consumer. Eg. A pool or mid-tier Windows server with 4GB of RAM, or a pool of low-tier Ubuntu servers with 2GB of RAM.
- Storage pools – file-based or block-based, empty or filled.
- Network pools – different configured firewall devices or physical network switches.
- CPU pools – Broken down into individual processing cores, ready to be allocated for virtual servers.
- Memory pool – A pool of RAM, cache, volatile storage.

Types of Resource Pools

- Pools can be complex, when multiple pools created for specific cloud consumers or applications.
- Hierarchical resource pools:
  - Create sibling resource pools from physically grouped IT resources.
  - Sibling pools are isolated from one another so that cloud consumer is only provided access to its respective pool.
- Nested resource pools:
  - Larger pools are divided into smaller pools that individually group similar type of IT resources together.
  - Assign resource pools to different departments or groups in similar cloud consumer organization.

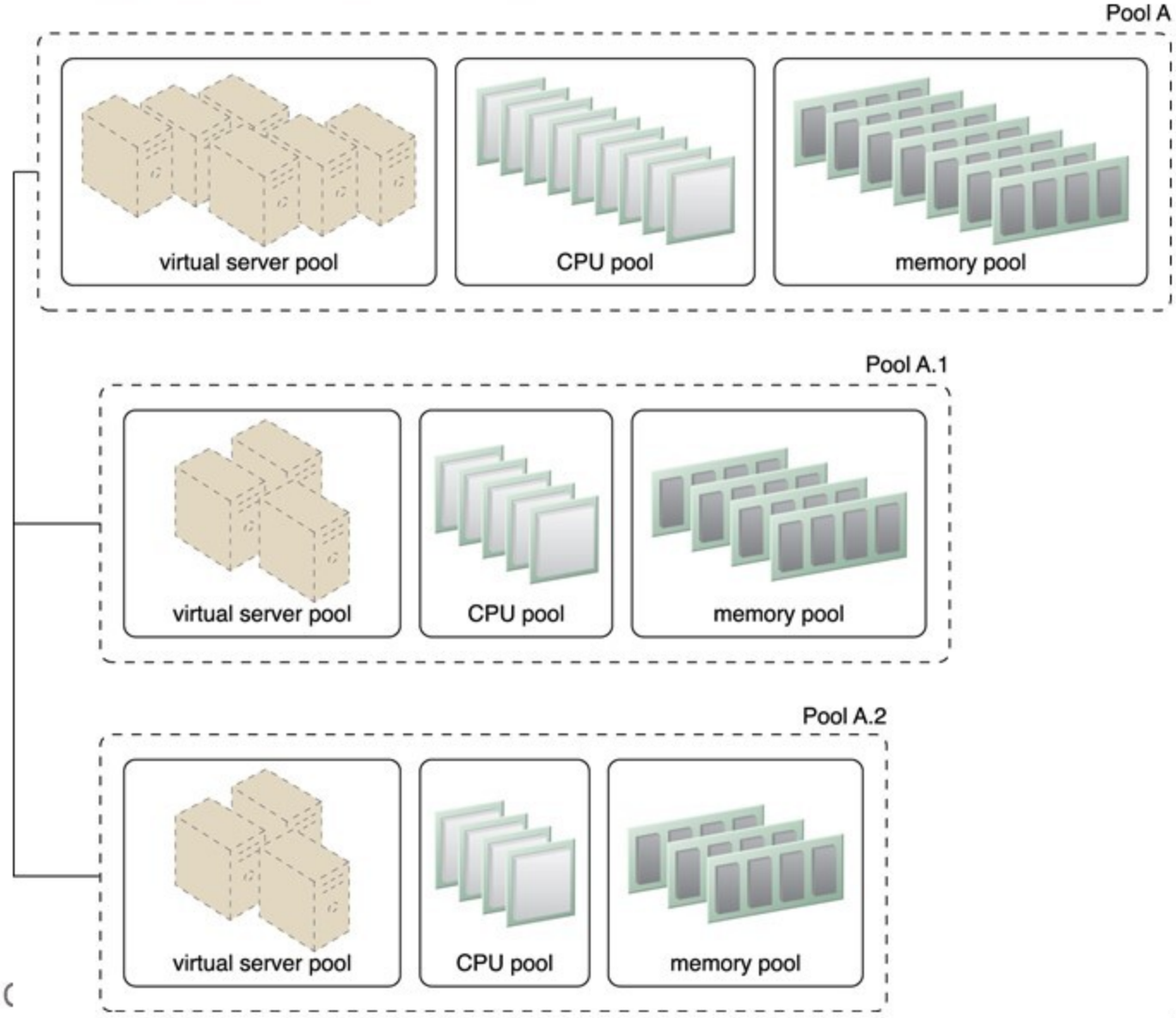
Hierarchical Resource Pool

Figure 11.3 Pools B and C are sibling pools that are taken from the larger Pool A, which has been allocated to a cloud consumer. This is an alternative to taking the IT resources for Pool B and Pool C from a general reserve of IT resources that is shared throughout the cloud.



Nested Resource Pool

Figure 11.4 Nested Pools A.1 and Pool A.2 are comprised of the same IT resources as Pool A, but in different quantities. Nested pools are typically used to provision cloud services that need to be rapidly instantiated using the same type of IT resources with the same configuration settings.



Resource Pooling Architecture

- Mechanisms applied:
  - Audit monitor
  - Cloud usage monitor
  - Hypervisor
  - Logical network perimeter
  - Pay-per-use monitor
  - Remote administration system
  - Resource management system
  - Resource replication

Dynamic Scalability Architecture

Dynamic Scalability Architecture

- Based on a system of predefined scaling conditions that trigger the dynamic allocation of IT resources from resource pools.
- Utilizes the automated scaling mechanism:
  - Workload thresholds that dictate when new IT resources need to be added.
  - Determines how many additional IT resources can be dynamically provided based on the terms of a given cloud consumer's provisioning contract.

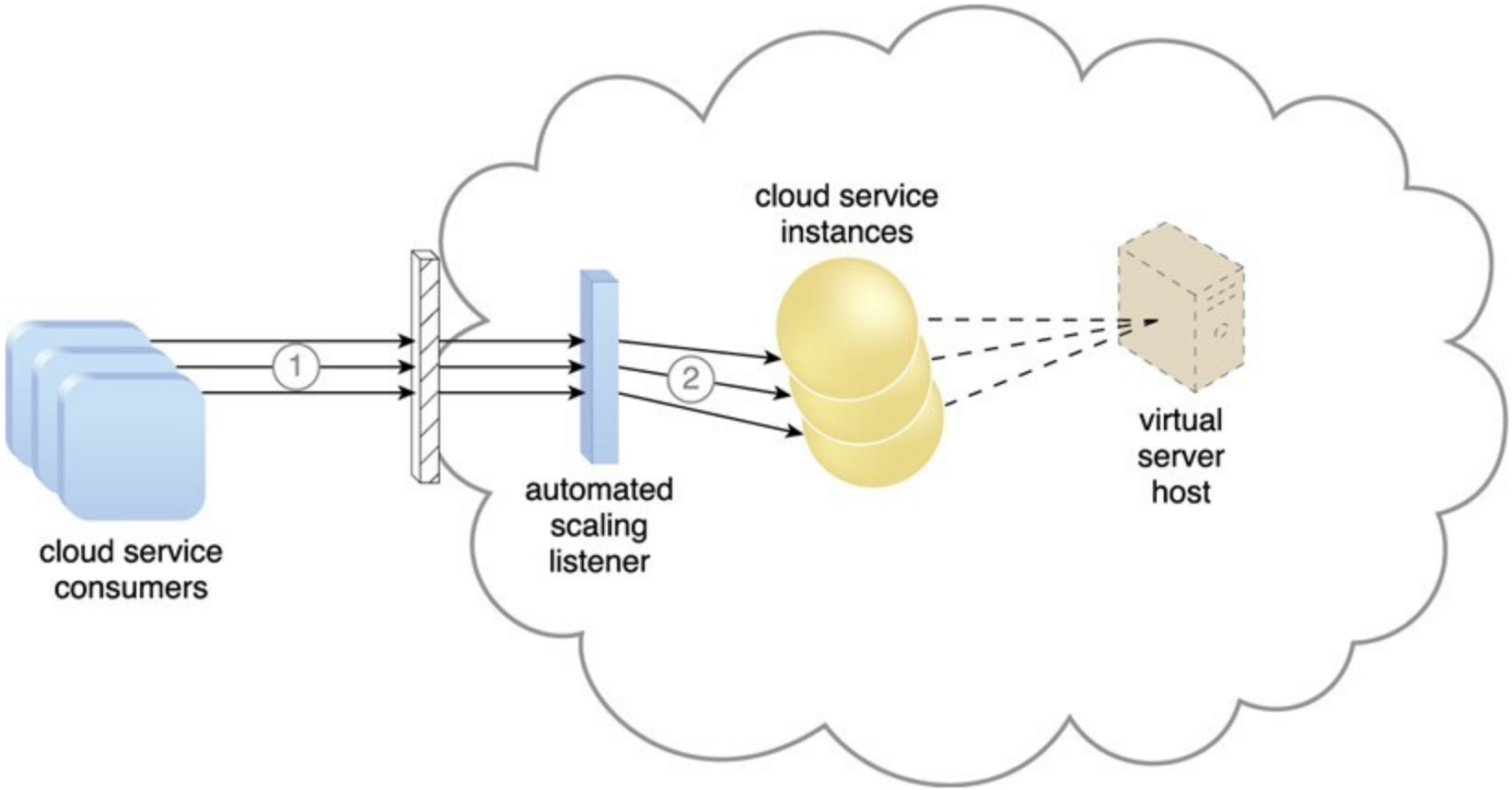


Figure 11.5 Cloud service consumers are sending requests to a cloud service (1). The automated scaling listener monitors the cloud service to determine if predefined capacity thresholds are being exceeded (2).

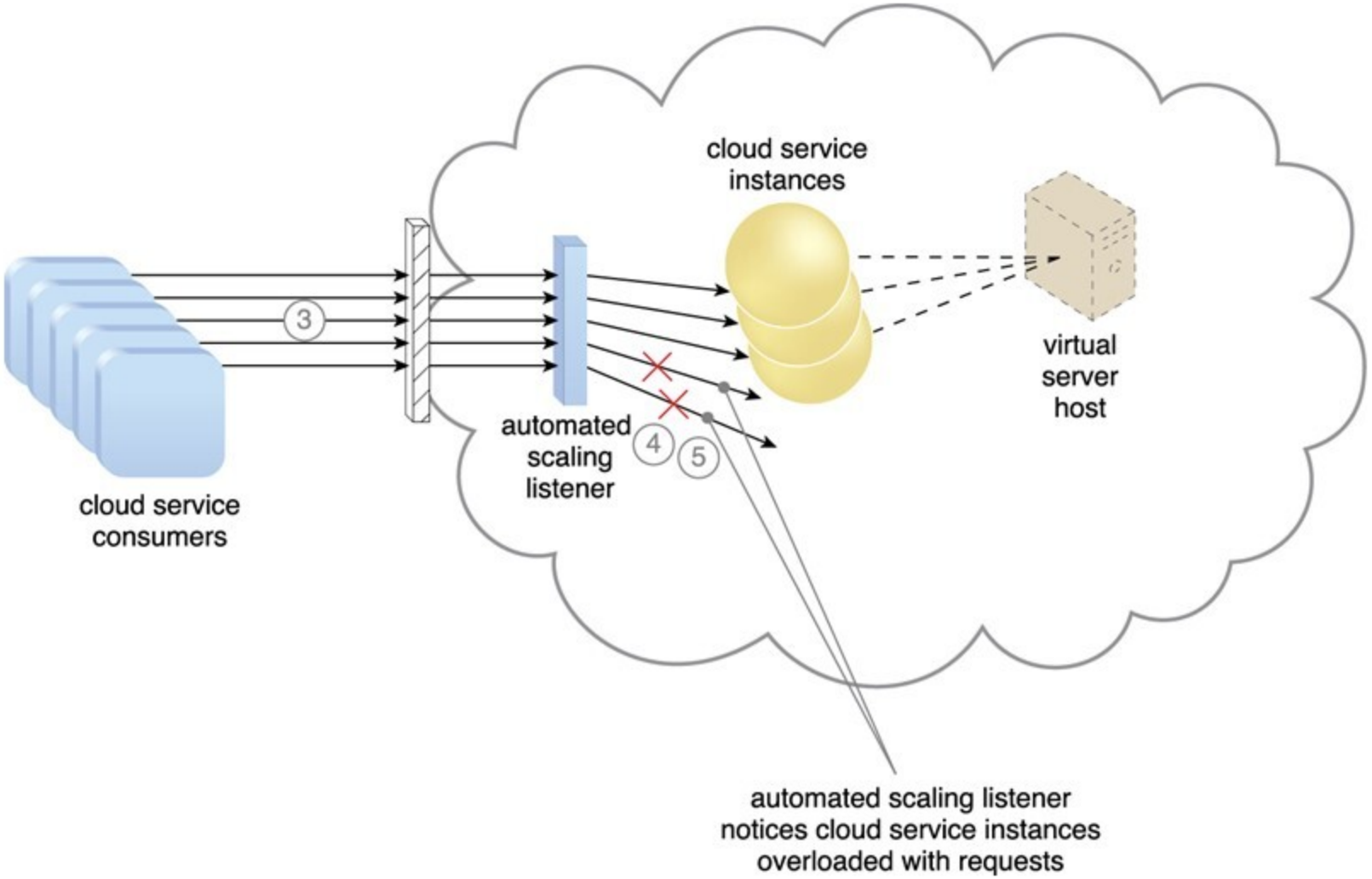


Figure 11.6 The number of requests coming from cloud service consumers increases (3). The workload exceeds the performance thresholds. The automated scaling listener determines the next course of action based on a predefined scaling policy (4). If the cloud service implementation is deemed eligible for additional scaling, the automated scaling listener initiates the scaling process (5).

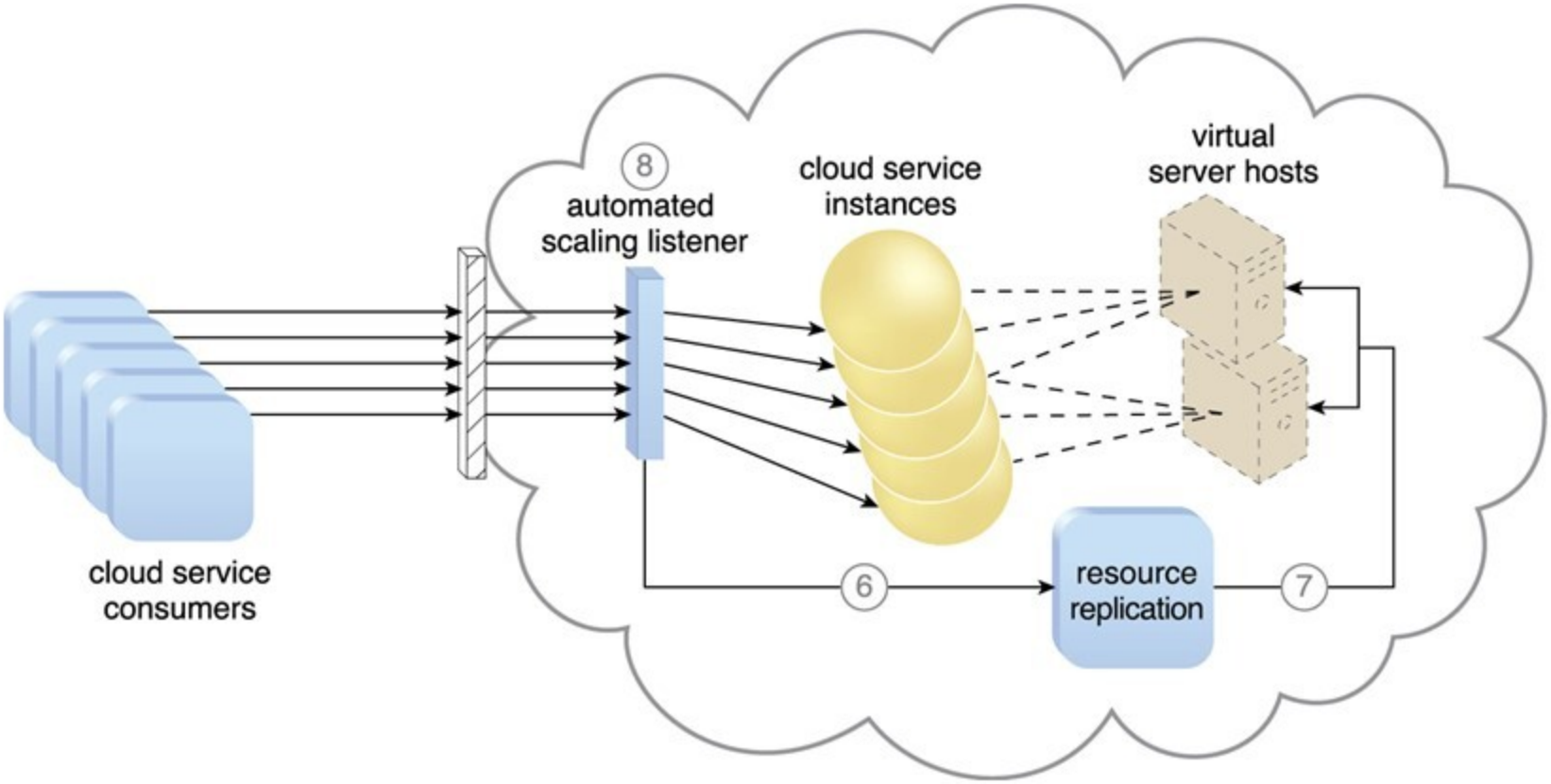


Figure 11.7 The automated scaling listener sends a signal to the resource replication mechanism (6), which creates more instances of the cloud service (7). Now that the increased workload has been accommodated, the automated scaling listener resumes monitoring and detracting and adding IT resources, as required (8).

Types of Dynamic Scaling

- Different types of dynamic scaling:
  - Dynamic horizontal scaling - IT resource instances are scaled out and in to handle fluctuating workloads.
  - 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 (e.g. add processing core, increase memory, etc.)
  - Dynamic relocation – IT resource is relocated to a host with more capacity.

Dynamic Scalability Architecture

- Mechanisms applied:
  - Cloud usage monitor
  - Hypervisor
  - Pay-per-use monitor

Elastic Resource Capacity Architecture

Elastic Resource Capacity Architecture

- Dynamic provisioning of virtual servers, using a system that allocates and reclaims CPUs and RAM to immediate response in the fluctuating processing requirements of hosted IT resources.
- Resource pools are used by scaling technology that interacts with the hypervisor and/or VIM to retrieve and return CPU and RAM resources at runtime.
- Utilizes intelligent automation engine script to automate administration tasks with workflow logic.

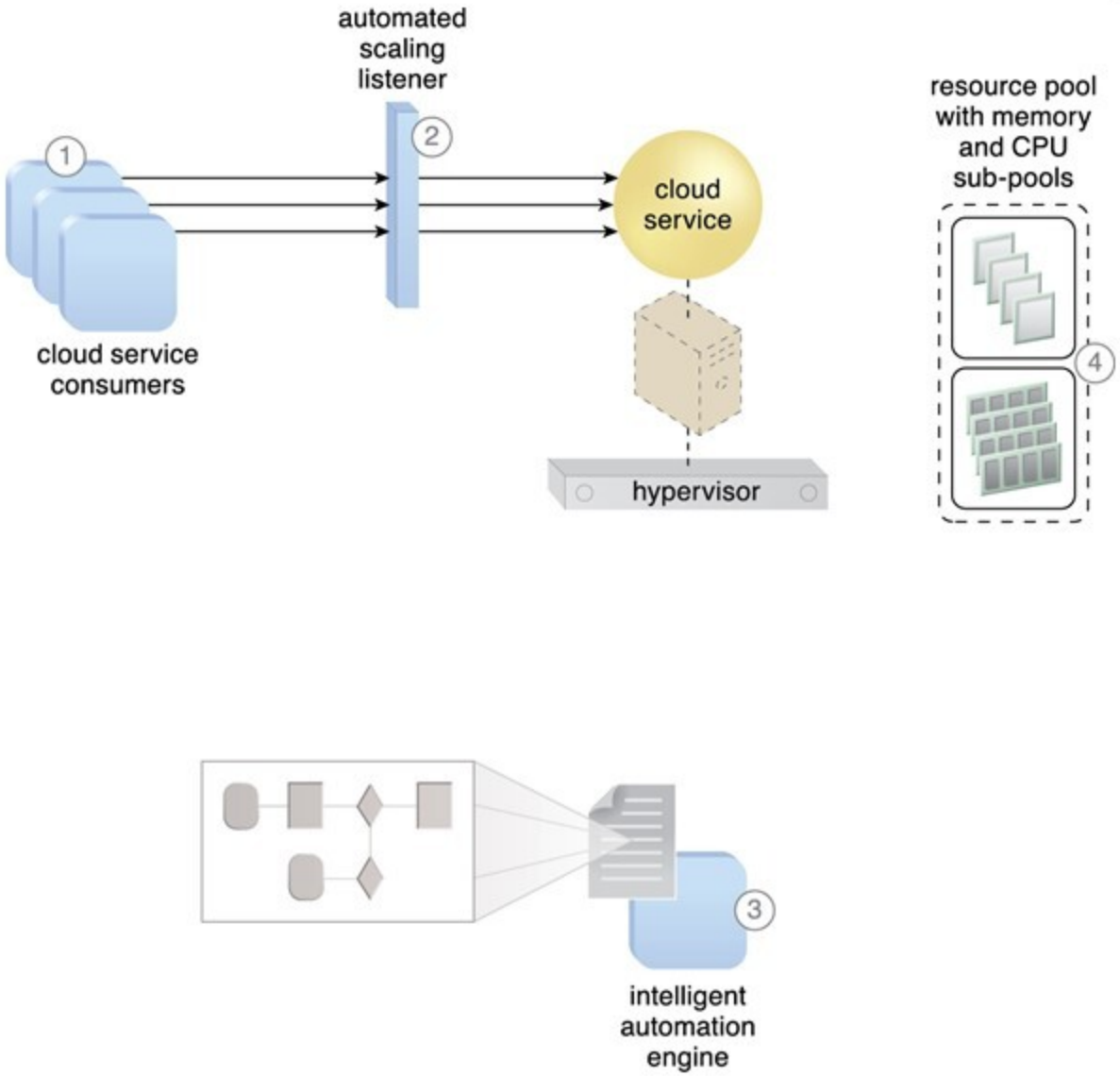


Figure 11.8 Cloud service consumers are actively sending requests to a cloud service (1), which are monitored by an automated scaling listener (2). An intelligent automation engine script is deployed with workflow logic (3) that is capable of notifying the resource pool using allocation requests (4).

Elastic Resource Capacity Architecture

- Mechanisms applied:
  - Cloud usage monitor
  - Pay-per-use monitor
  - Resource replication

Service Load Balancing Architecture

Service Load Balancing Architecture

- Specialized variation of workload distribution architecture, i.e. geared specifically for scaling cloud service implementations by creating redundant deployments of cloud services.
- Implementation of duplicate cloud services are organized into a resource pool, while load balancer is positioned either an external or built-in component to balance workloads.
- Mechanisms applied:
  - Cloud usage monitor
  - Resource cluster
  - Resource replication

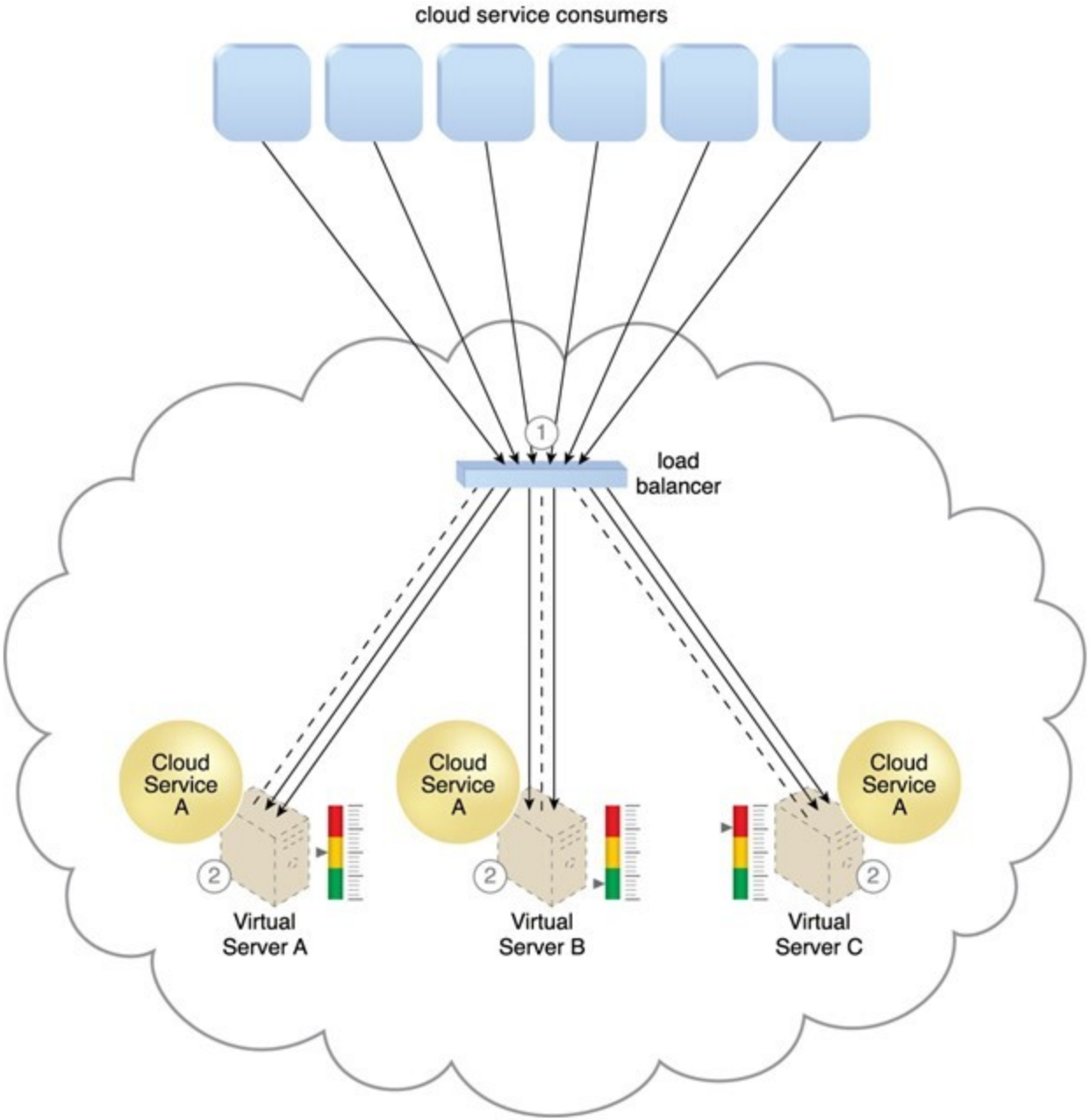
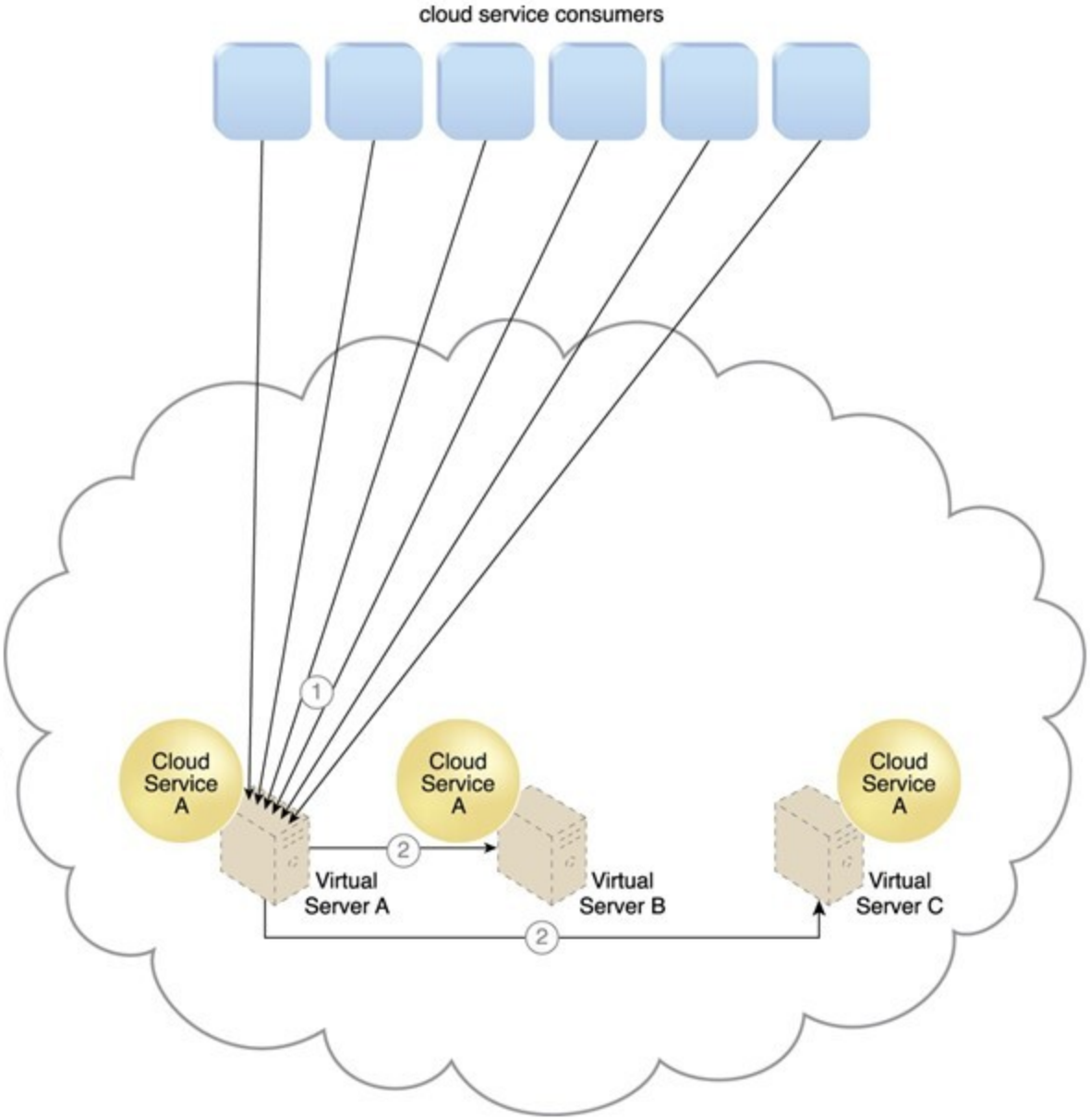


Figure 11.10 The load balancer intercepts messages sent by cloud service consumers (1) and forwards them to the virtual servers so that the workload processing is horizontally scaled (2).

Figure 11.11 Cloud service consumer requests are sent to Cloud Service A on Virtual Server A (1). 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 (2).



Cloud Bursting Architecture

Cloud Bursting Architecture

- Establishes a form of dynamic scaling that scales or "bursts out" on-premise IT resources into a cloud whenever predefined capacity thresholds have been reached.
  - The corresponding cloud-based IT resources are redundantly pre-deployed but remain inactive until cloud bursting occurs – released and "bursts in" back to the on-premise environment.
- A flexible scaling architecture – cloud consumer has option to use resources only to meet higher usage demands.
- Utilizes the automated scaling listener.

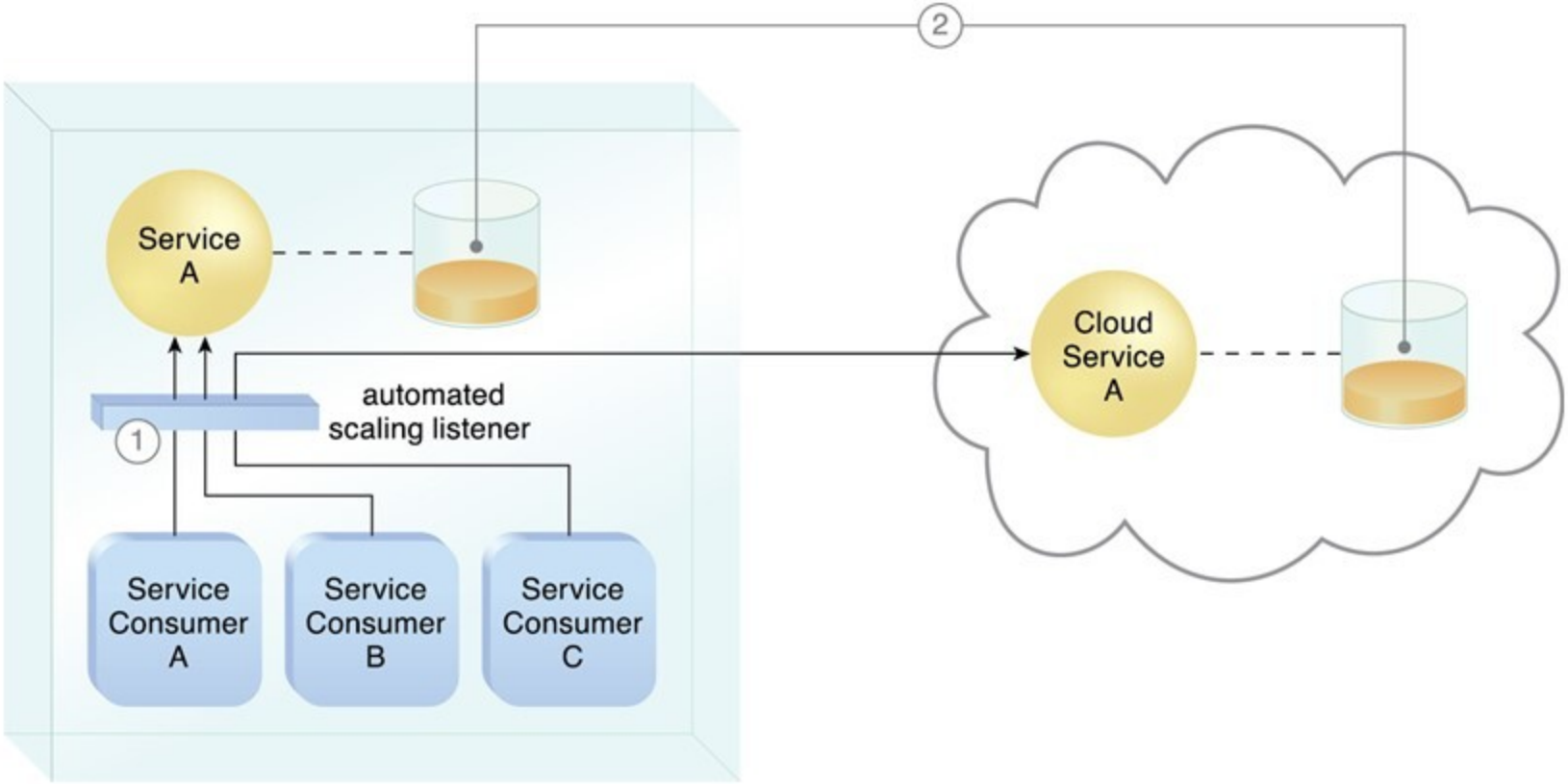


Figure 11.12 An automated scaling listener monitors the usage of on-premise Service A, and redirects Service Consumer C's request to Service A's redundant implementation in the cloud (Cloud Service A) once Service A's usage threshold has been exceeded (1). A resource replication system is used to keep state management databases synchronized (2).

Elastic Disk Provisioning Architecture

Elastic Disk Provisioning Architecture

- Alleviating issues on fixed-disk storage allocation for cloud consumers - Consumers being charged for storage space, although they have yet to utilize it.

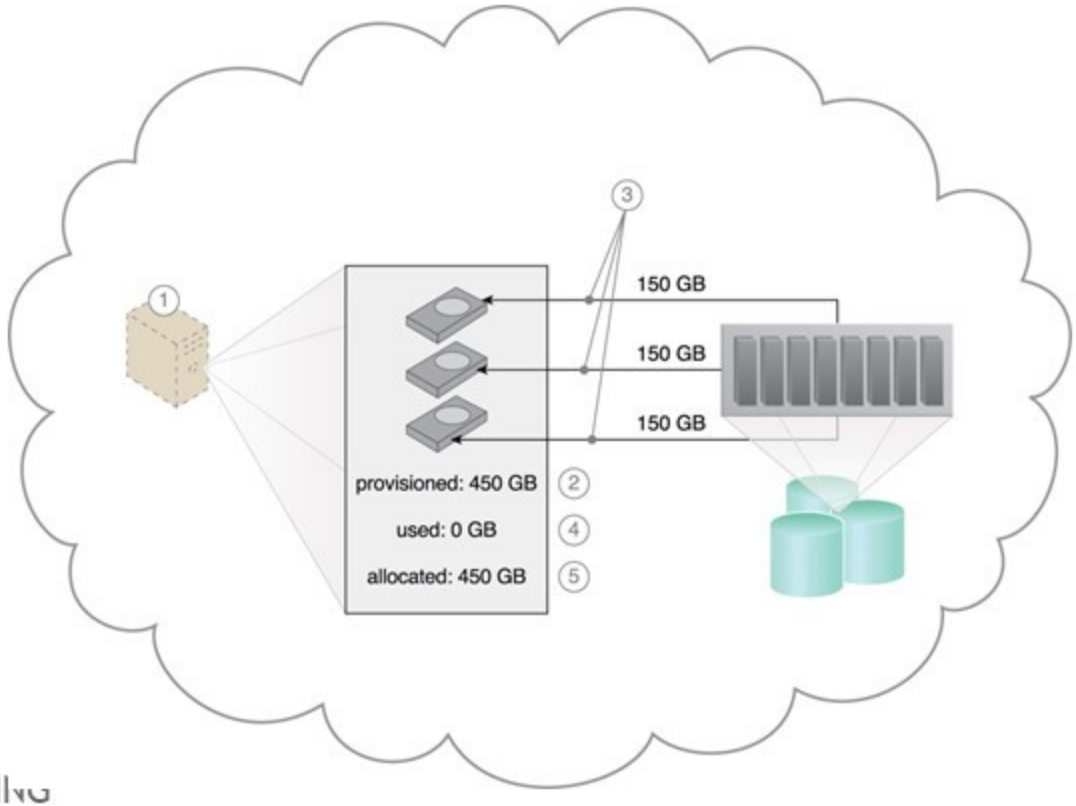
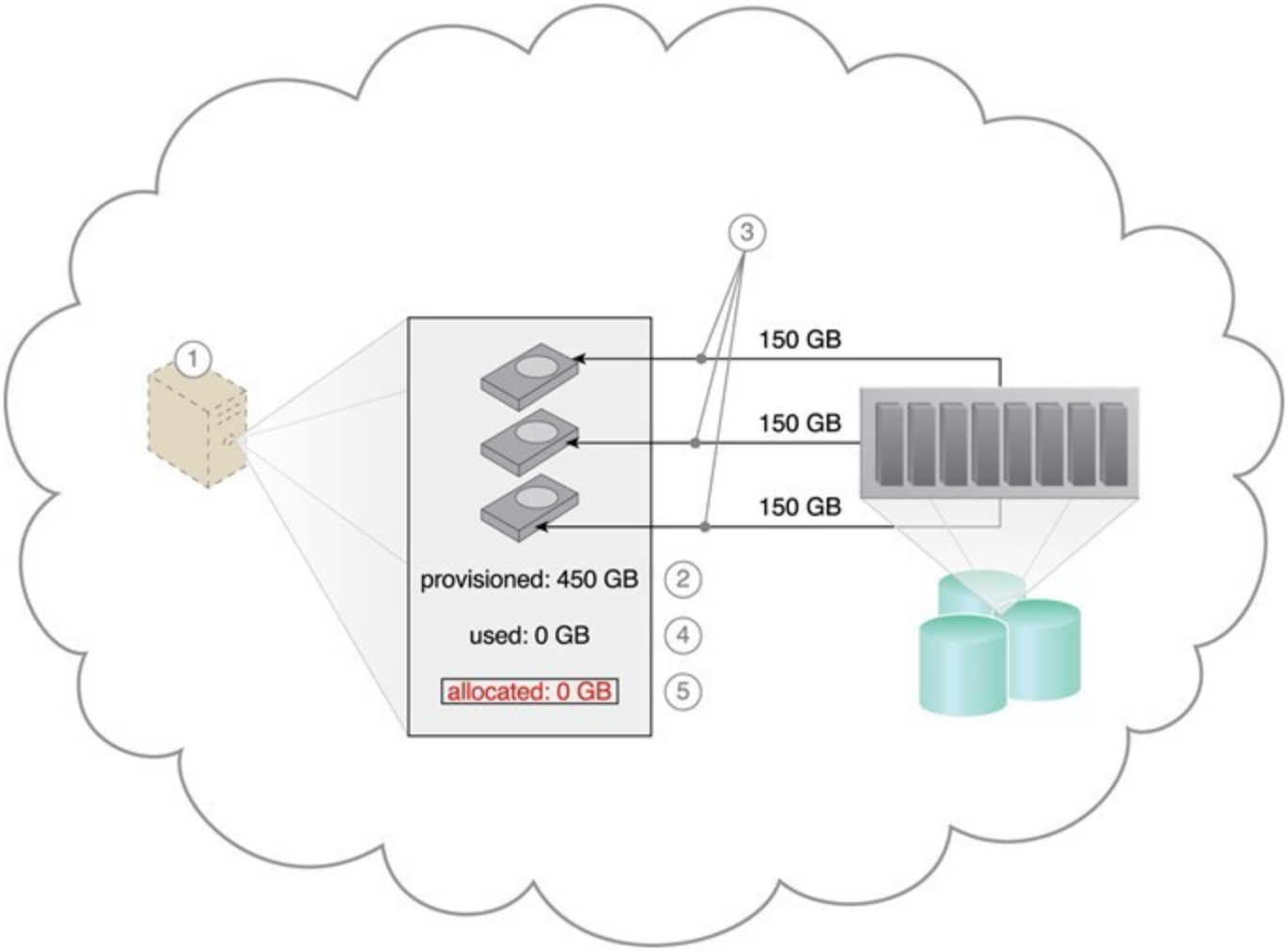


Figure 11.13 The cloud consumer requests a virtual server with three hard disks, each with a capacity of 150 GB (1). The virtual server is provisioned according to the elastic disk provisioning architecture, with a total of 450 GB of disk space (2). The 450 GB is allocated to the virtual server by the cloud provider (3). The cloud consumer has not installed any software yet, meaning the actual used space is currently 0 GB (4). Because the 450 GB are already allocated and reserved for the cloud consumer, it will be charged for 450 GB of disk usage as of the point of allocation (5).

Elastic Disk Provisioning Architecture

- Establishes a dynamic storage provisioning system that ensures that the cloud consumer is granularly billed for the exact amount of storage that it actually uses.
- Uses thin-provisioning technology for the dynamic allocation of storage space and further supported by runtime usage monitor to collect accurate usage data for billing purposes.
- Mechanisms applied:
  - Cloud usage monitor
  - Pay-per-use monitor
  - Resource replication

Figure 11.14 The cloud consumer requests a virtual server with three hard disks, each with a capacity of 150 GB (1). The virtual server is provisioned by this architecture with a total of 450 GB of disk space (2). The 450 GB are set as the maximum disk usage that is allowed for this virtual server, although no physical disk space has been reserved or allocated yet (3). The cloud consumer has not installed any software, meaning the actual used space is currently at 0 GB (4). Because the allocated disk space is equal to the actual used space (which is currently at zero), the cloud consumer is not charged for any disk space usage (5).



Thin-provisioning software is installed on virtual servers that process dynamic storage allocation via the hypervisor.

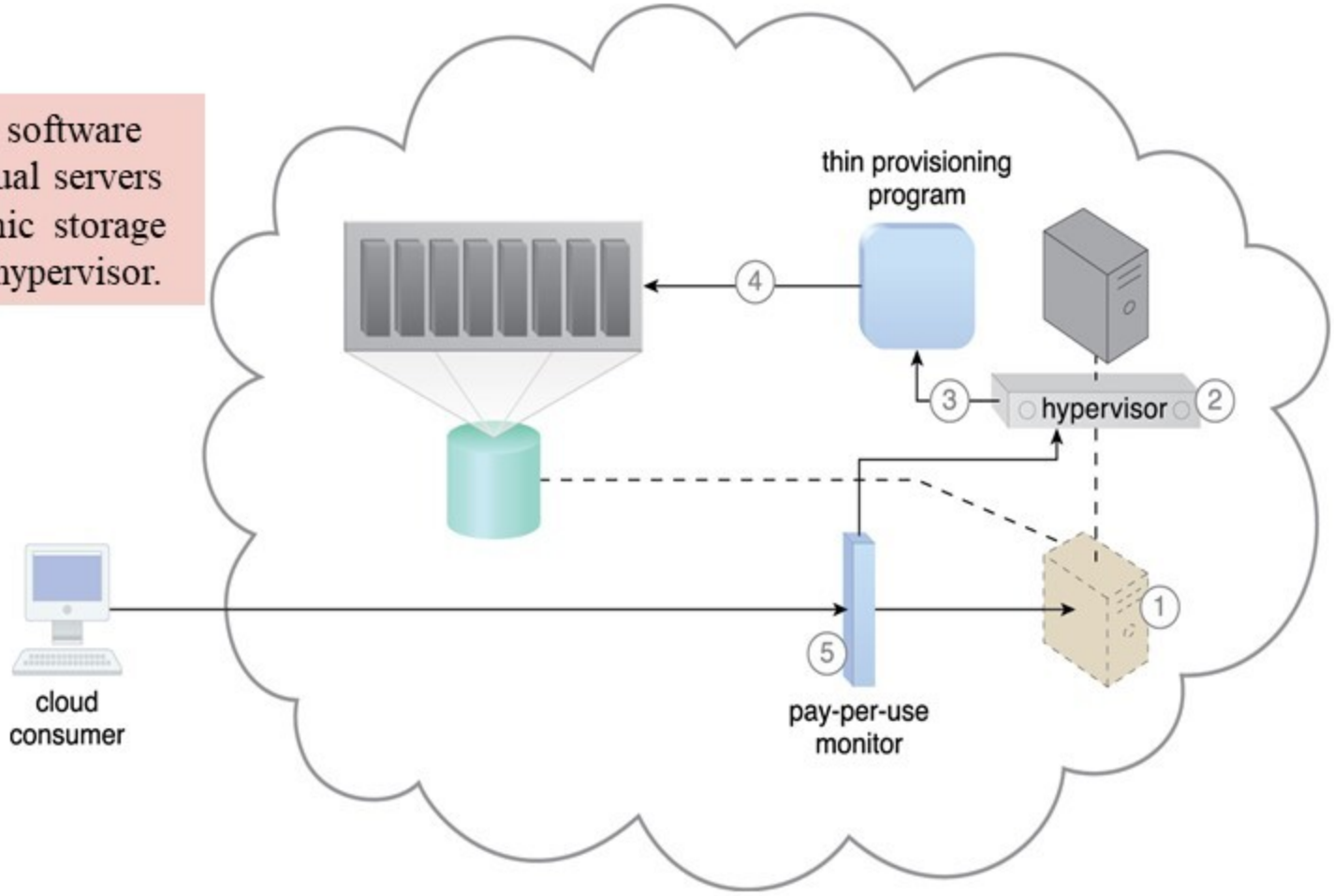


Figure 11.15 A request is received from a cloud consumer, and the provisioning of a new virtual server instance begins (1). As part of the provisioning process, the hard disks are chosen as dynamic or thin-provisioned disks (2). The hypervisor calls a dynamic disk allocation component to create thin disks for the virtual server (3). Virtual server disks are created via the thin-provisioning program and saved in a folder of near-zero size. The size of this folder and its files grow as operating applications are installed and additional files are copied onto the virtual server (4). The pay-per-use monitor tracks the actual dynamically allocated storage for billing purposes (5).

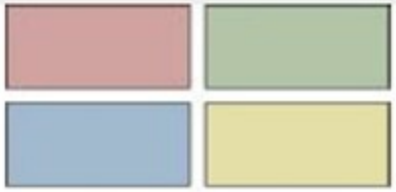
Redundant Storage Architecture

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.
- A storage service gateway diverts cloud consumer requests to secondary device whenever the primary device fails.

Logical Unit Number (LUN)

LUN

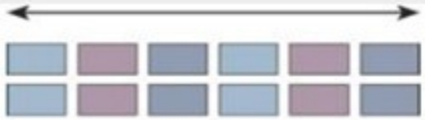


LUNs

A logical unit number (LUN) is a logical drive that represents a partition of a physical drive.

Storage Replication

Storage replication is a variation of the resource replication mechanisms used to synchronously or asynchronously replicate data from a primary storage device to a secondary storage device. It can be used to replicate partial and entire LUNs.



storage replication

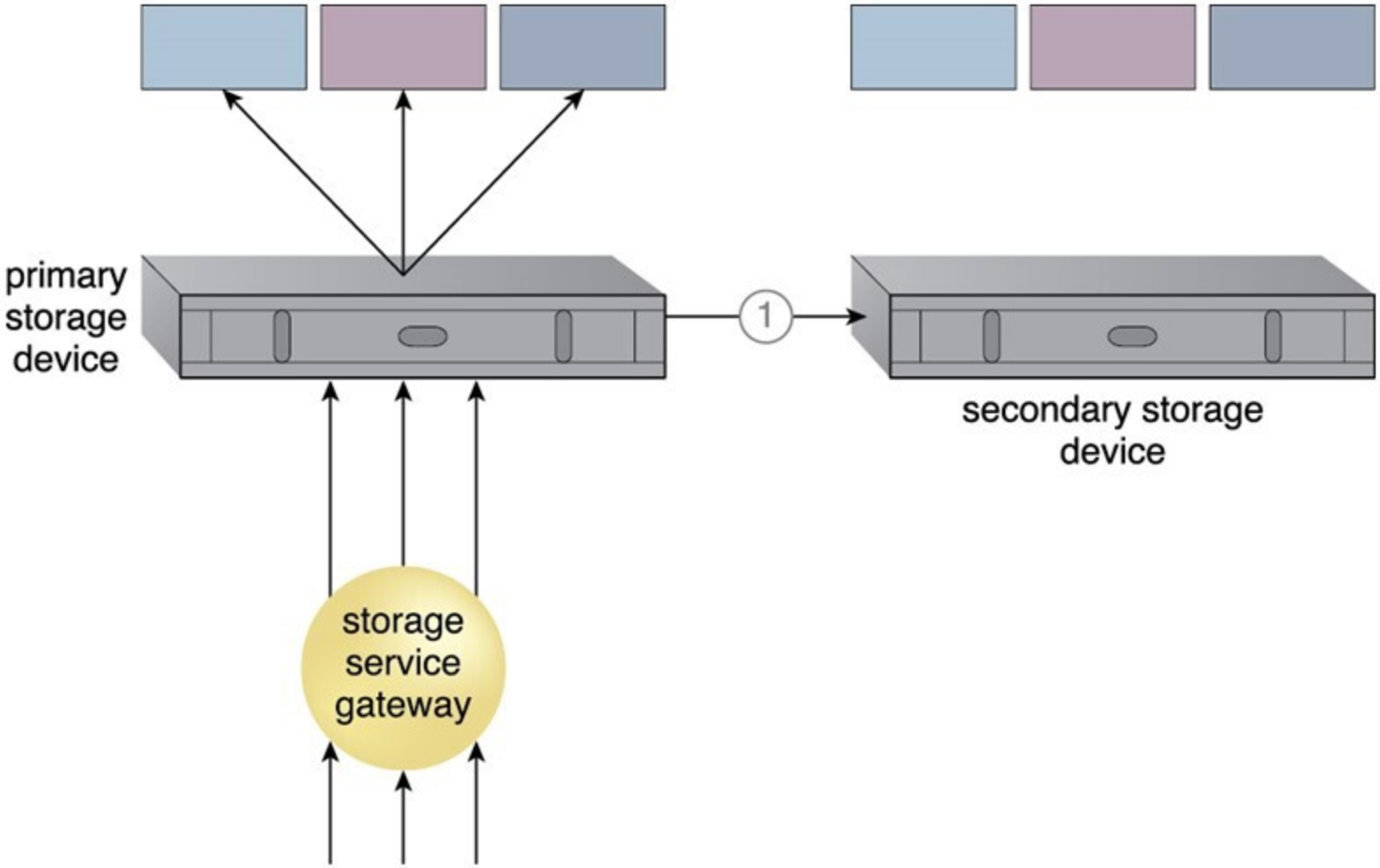


Figure 11.16 The primary cloud storage device is routinely replicated to the secondary cloud storage device (1).

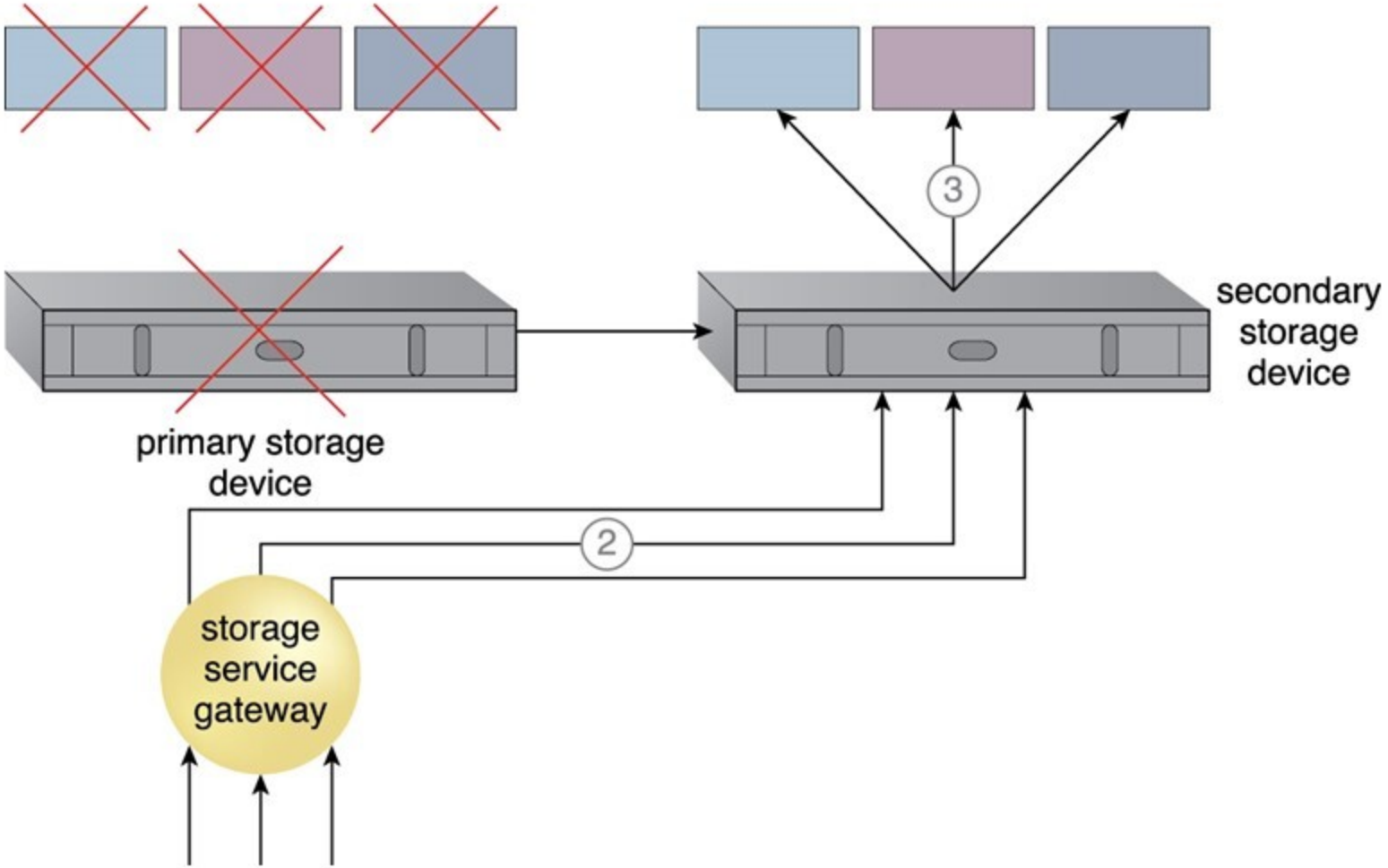


Figure 11.17 The primary storage becomes unavailable and the storage service gateway forwards the cloud consumer requests to the secondary storage device (2). The secondary storage device forwards the requests to the LUNs, allowing cloud consumers to continue to access their data (3).

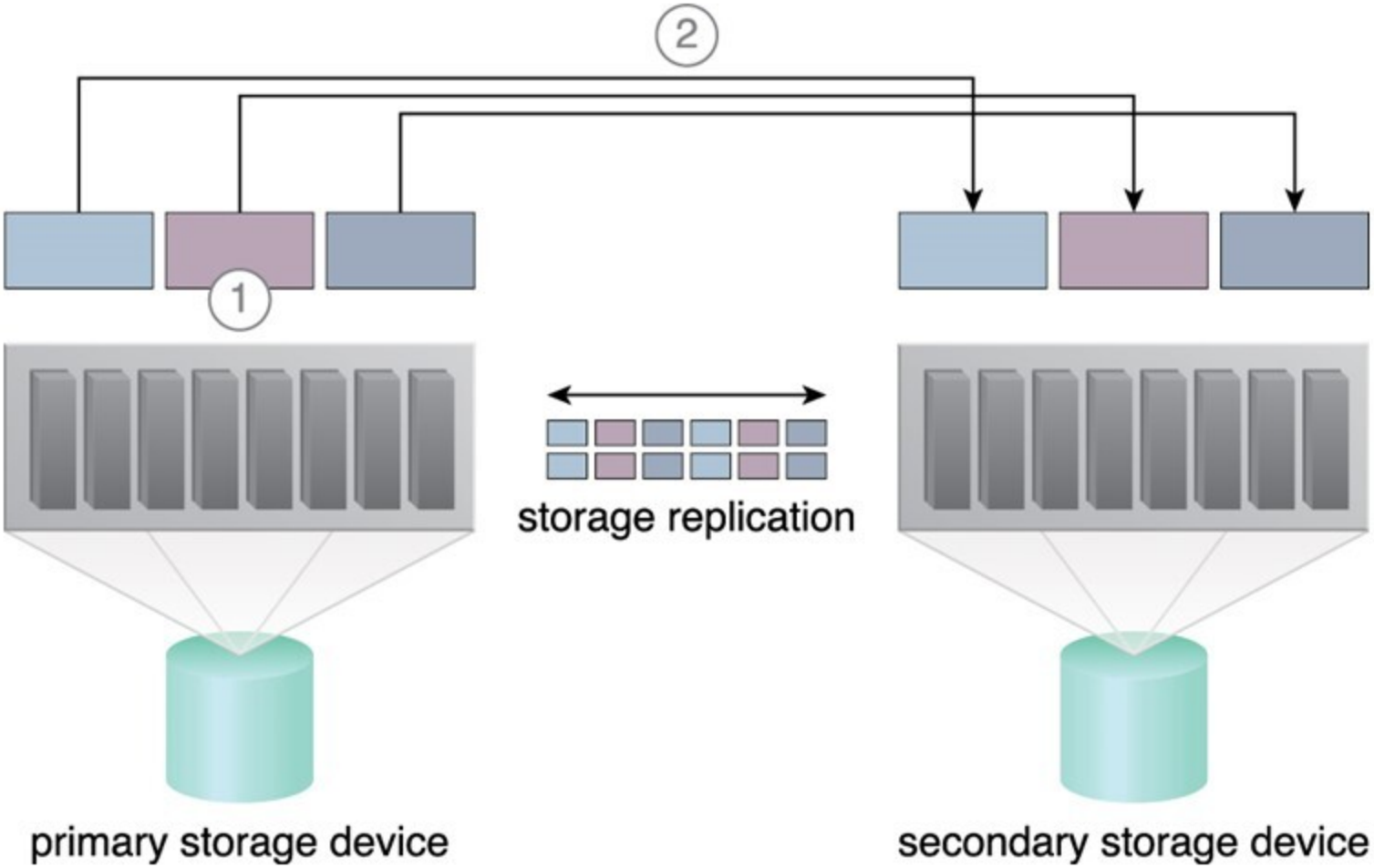


Figure 11.18 Storage replication is used to keep the redundant storage device synchronized with the primary storage device.