Cloud Computing & its Applications

Course Code: SWE4004

Dr Sunil Kumar Singh
Assistant Professor
School - SCOPE
VIT-AP University
sunil.singh@vitap.ac.in
Cabin - AB2 (124D)

Fundamental Cloud Architectures

Outline

- Workload distribution architecture
- Resource pooling architecture
- Dynamic scalability architecture
- Elastic resource capacity architecture
- Service load balancing architecture
- Cloud bursting architecture
- Elastic disk provisioning architecture
- Redundant storage architecture

VIT AP AMARAVATI

Workload distribution architecture

Fundamental cloud architectural models establish baseline functions and capabilities.

- **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 .

- The resulting workload distribution architecture reduces both IT resource over-utilization and under-utilization to an extent dependent upon the sophistication of the load balancing algorithms and runtime logic.

Workload Distribution

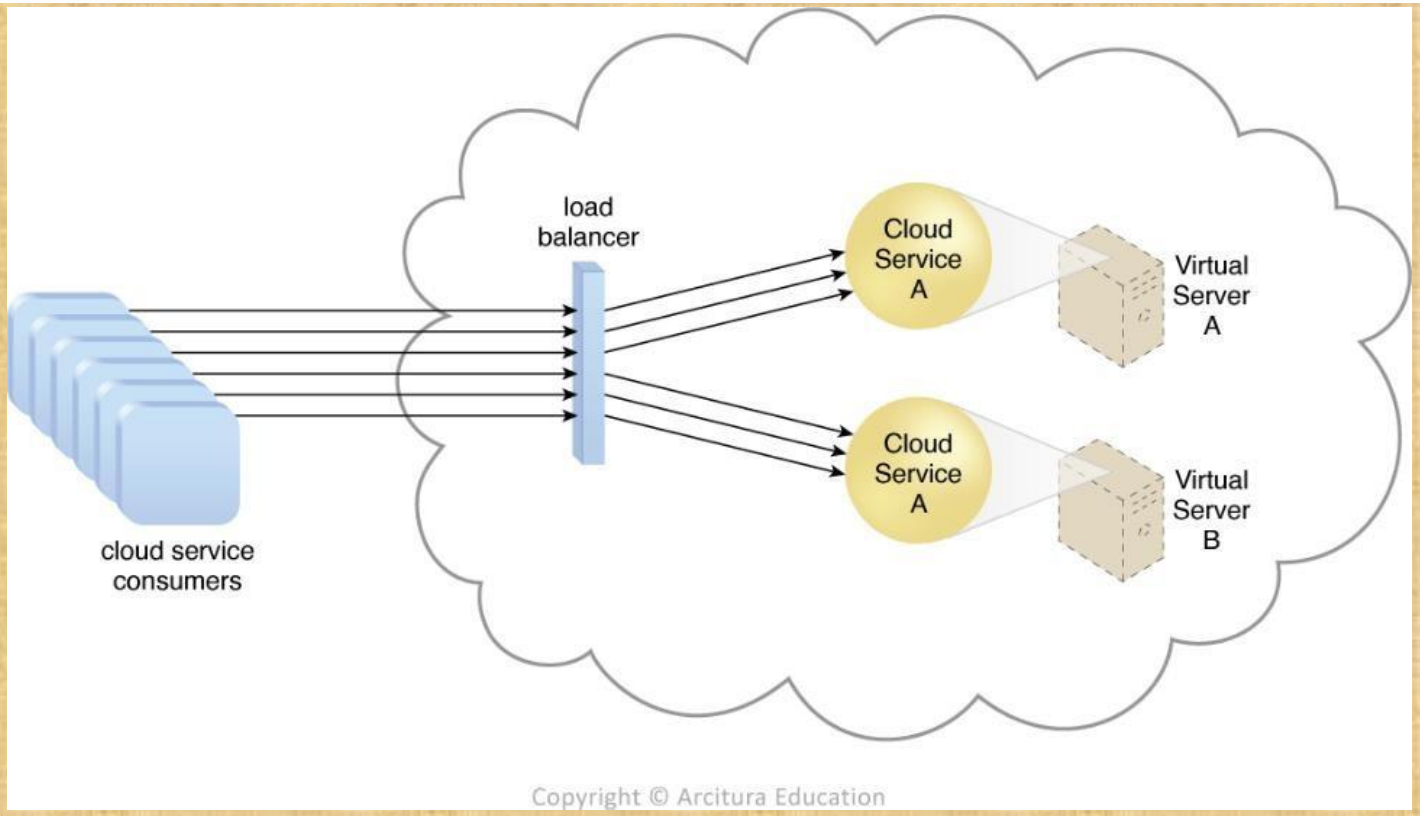


Figure 11.1 - A redundant copy of Cloud Service A is implemented on virtual Server B. The load balancer intercepts cloud service consumer requests and directs them to both virtual Servers A and B to ensure even workload distribution.

Workload distribution

Workload distribution commonly carried out in support of distributed

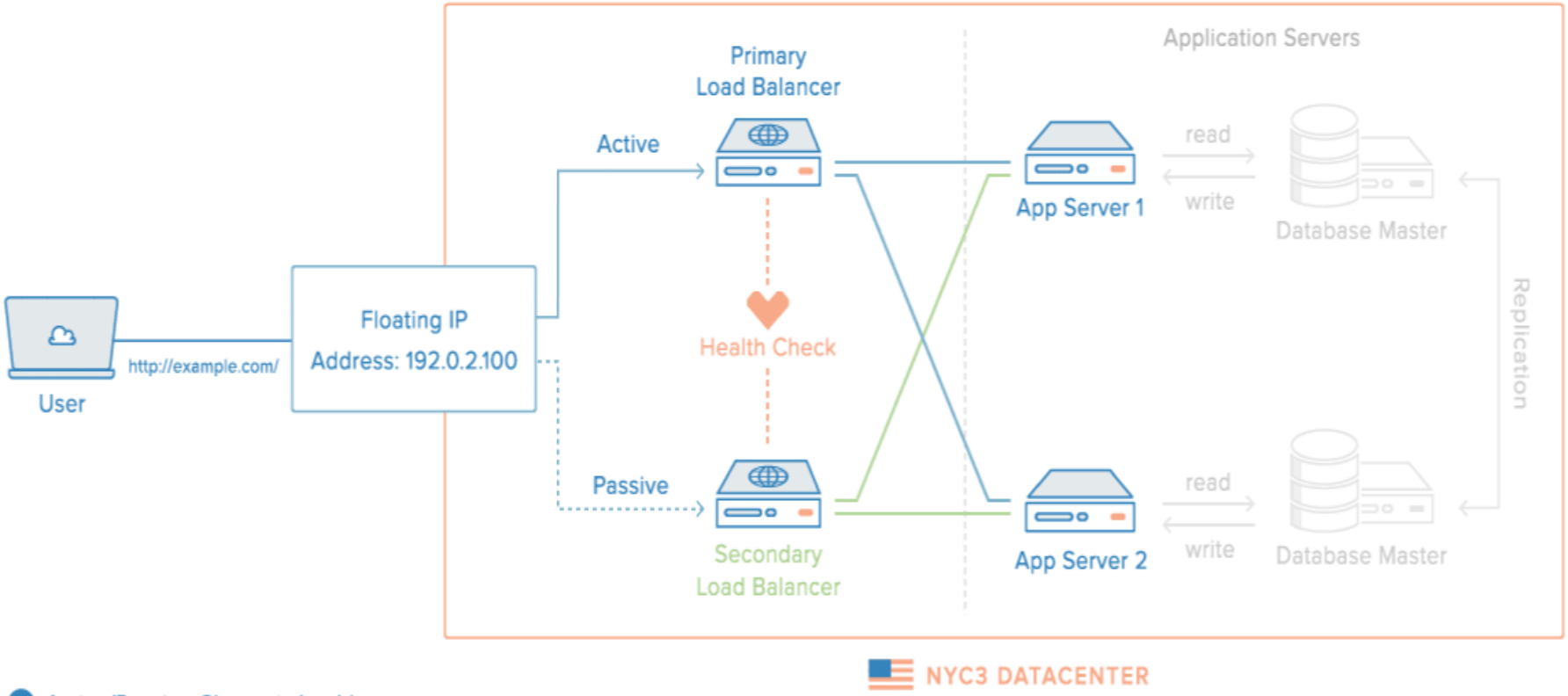
- virtual servers
- cloud storage devices
- cloud services.

Load balancing systems

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 architecture
- the load balanced virtual server
- the load balanced virtual switches architecture

Floating IP



1. Active/Passive Cluster is healthy
2. Primary node fails
3. Floating IP is assigned to Secondary node

Workload Distribution Architecture

The following mechanisms can also be part of workload distribution architecture:

- Audit Monitor
- Cloud Usage Monitor
- Hypervisor
- Logical Network Perimeter
- Resource Cluster
- Resource Replication

Static Load Balancing Algorithms

- Round Robin Load Balancing Algorithm (RR)
- Load Balancing Min-Min Algorithm (LB Min-Min)
- Load Balancing Min-Max Algorithm (LB Min-Max)

Dynamic LB algorithms:

- Honeybee Foraging Behavior Load Balancing Algorithm
- Throttled Load Balancing Algorithm
- ESCE (Equally Spread Current Execution) Load Balancing Algorithm
- Ant Colony Load Balancing Algorithm
- Biased Random Sampling Load Balancing Algorithm
- Modified Throttled Load Balancing Algorithm

VIT AP AMARAVATI

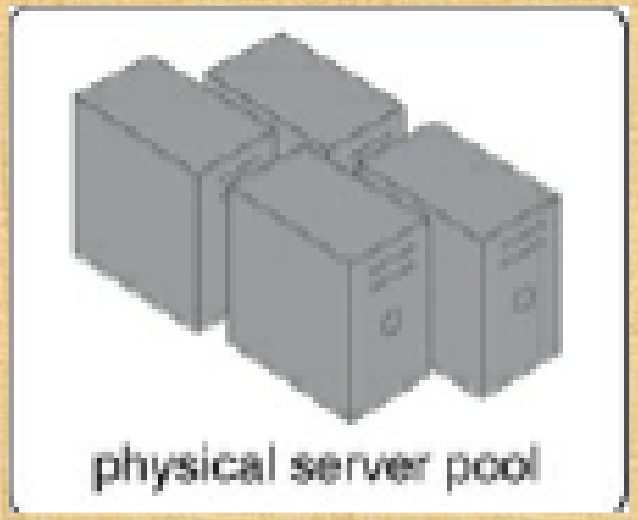
Resource Pooling Architecture

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 that automatically ensures that they remain synchronized.

- Common examples of resource pools:
  - physical server pool
  - virtual server pool
  - storage pool
  - network pool
  - CPU pool
  - memory pool

Physical Server Pool

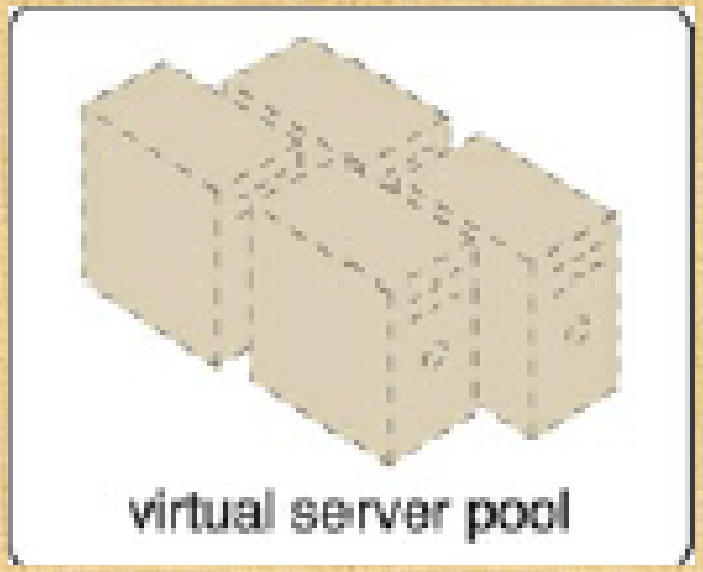
Physical server pools are composed of networked servers that have been installed with operating systems and other necessary programs and/or applications and are ready for immediate use.



Virtual Server Pool

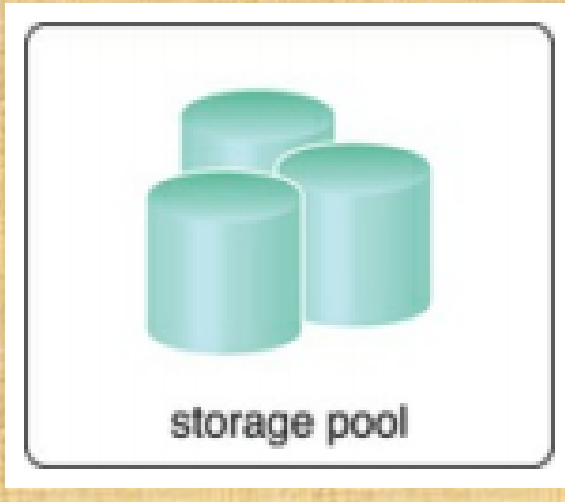
Virtual server pools are usually configured using one of several available templates chosen by the cloud consumer during provisioning.

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.



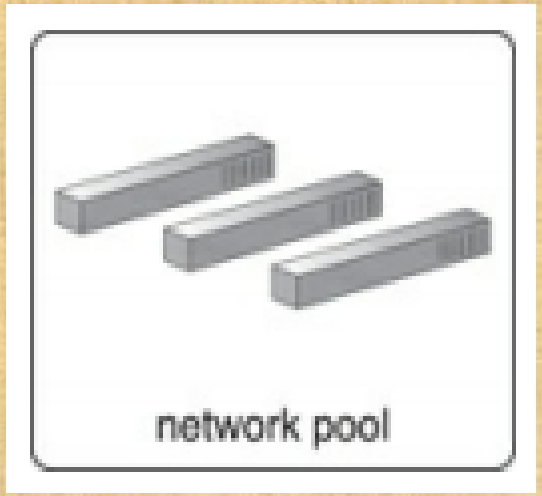
Storage Pool

- A storage pool is a collection of physical disks. A storage pool enables storage aggregation, elastic capacity expansion, and delegated administration. From a storage pool, you can create one or more virtual disks. These virtual disks are also referred to as storage spaces.



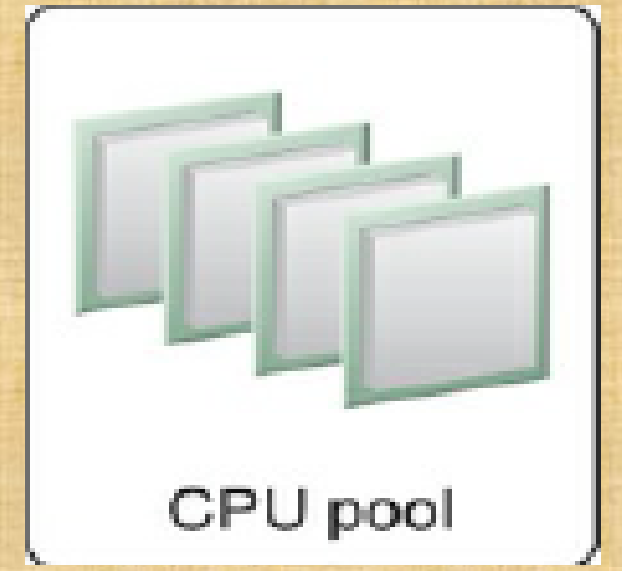
Network Pool

- 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.



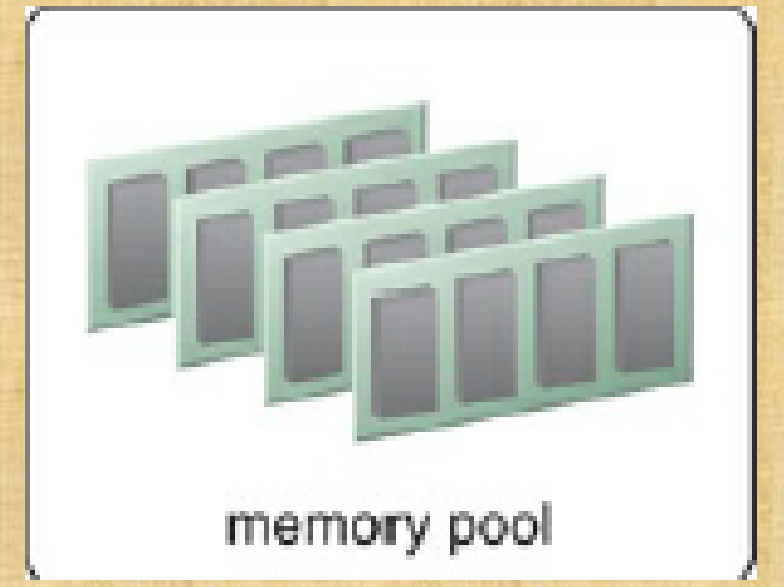
CPU Pool

- CPU pools are ready to be allocated to virtual servers, and are typically broken down into individual processing cores

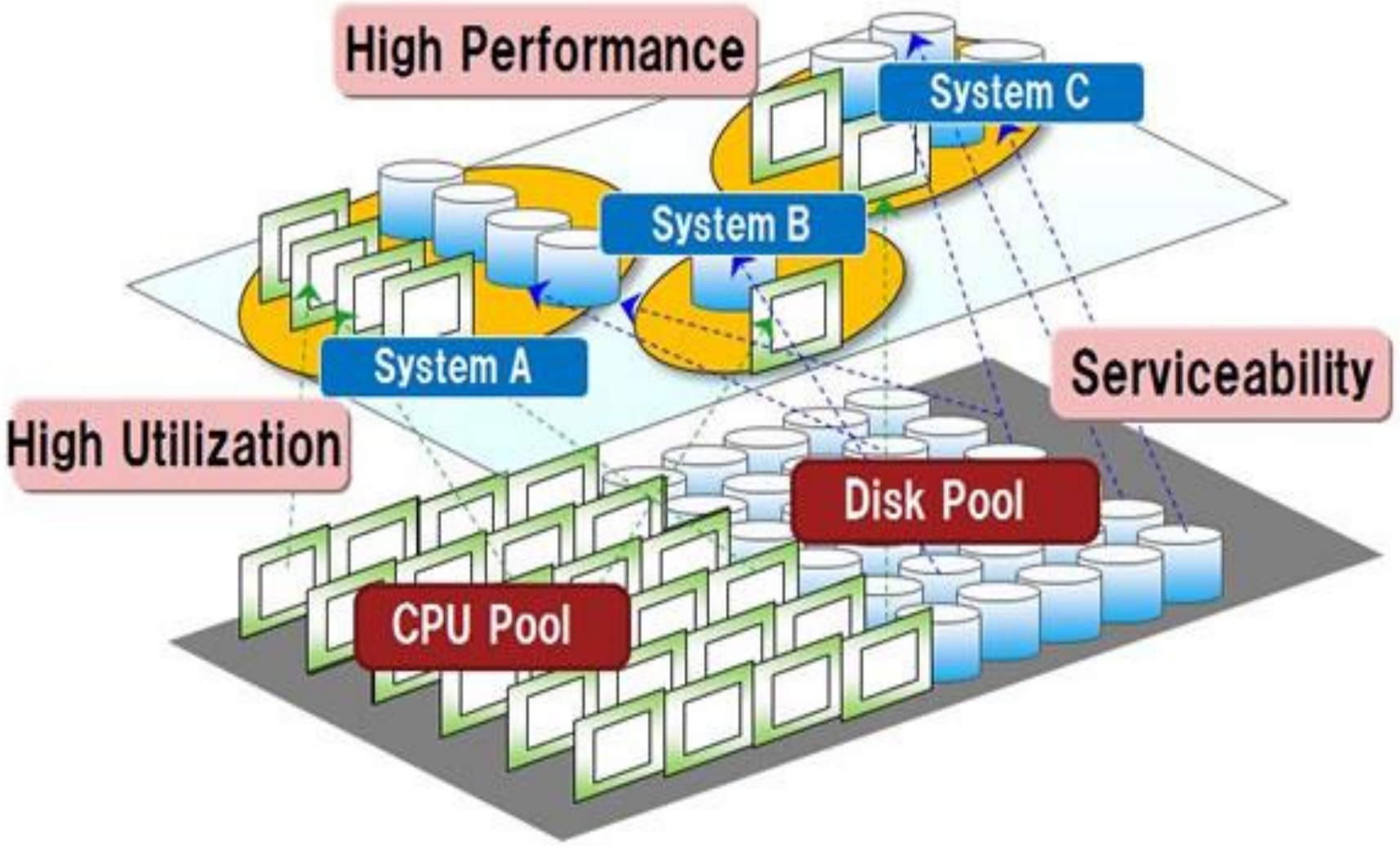


Memory Pool

- Pools of physical RAM can be used in newly provisioned physical servers or to vertically scale physical servers.



IT resource Pool



Sub-Pools

- 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

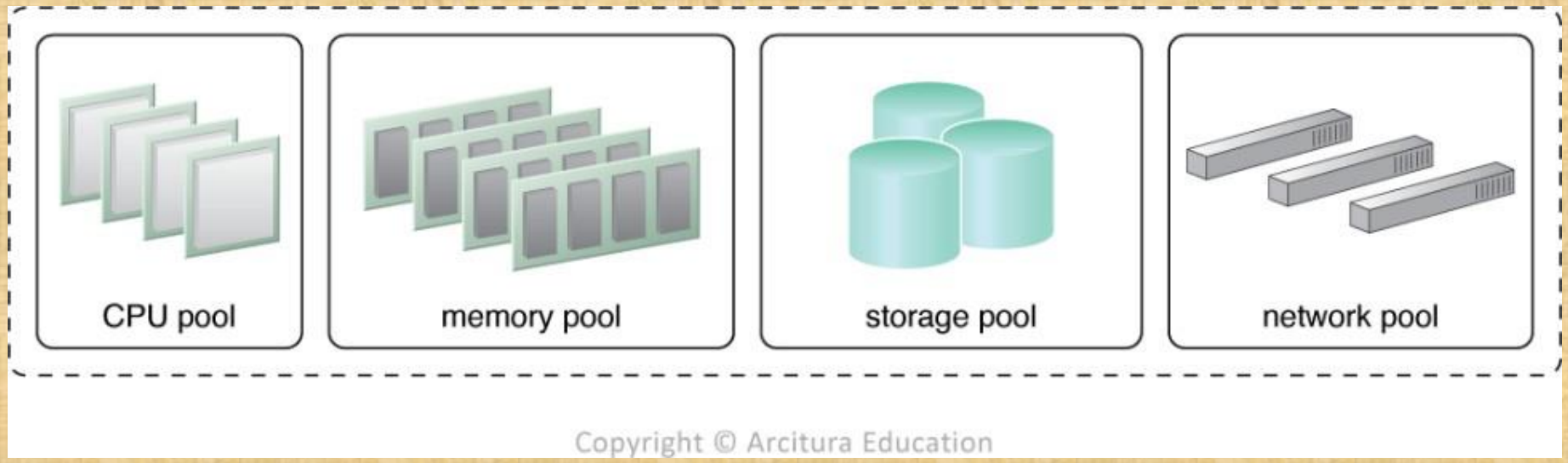
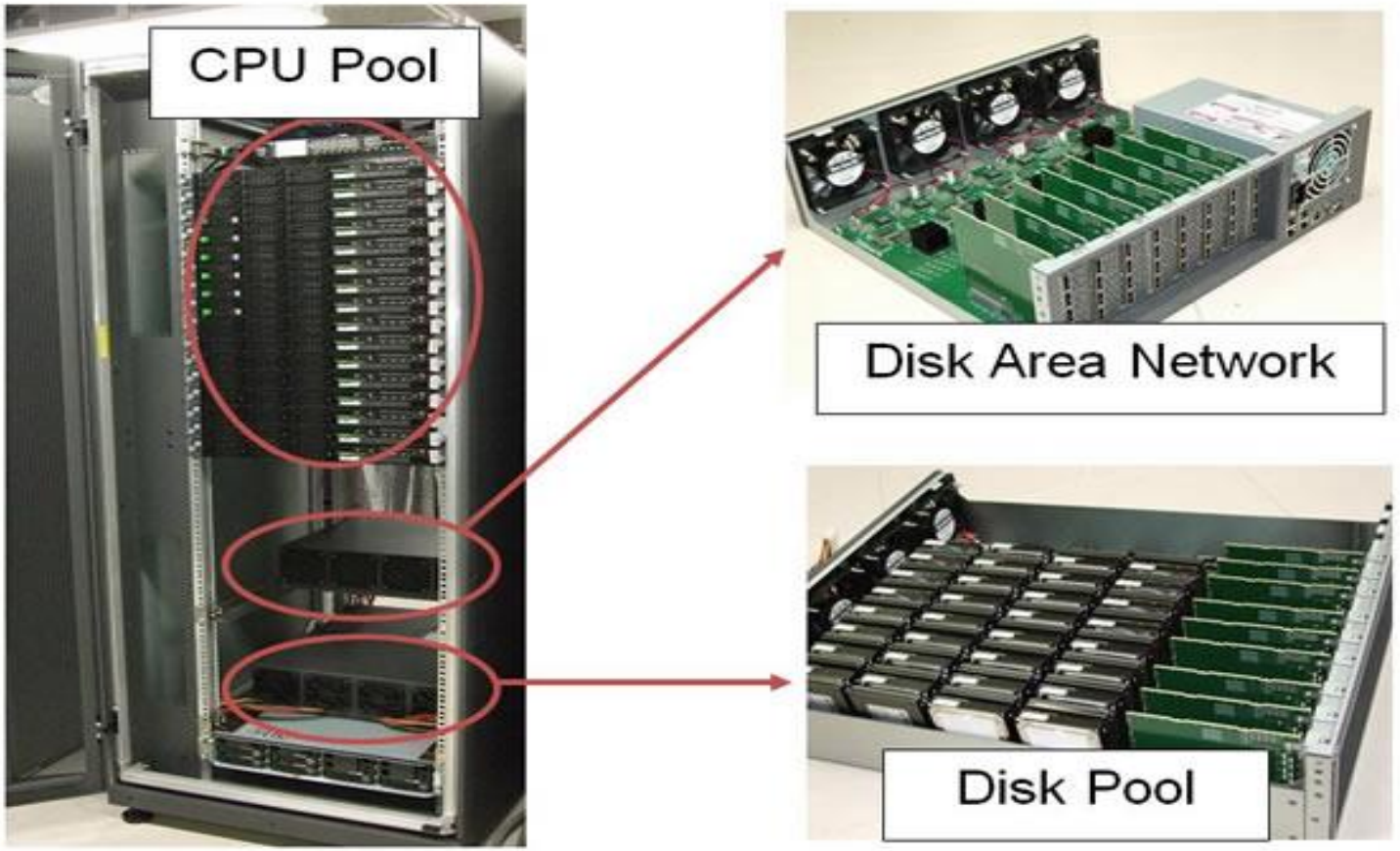


Figure 11.2 - A sample resource pool that is comprised of four sub-pools of CPUs, memory, cloud storage devices, and virtual network devices.


CPU Pool
Disk Area Network
Disk Pool

Resource Pooling Architecture

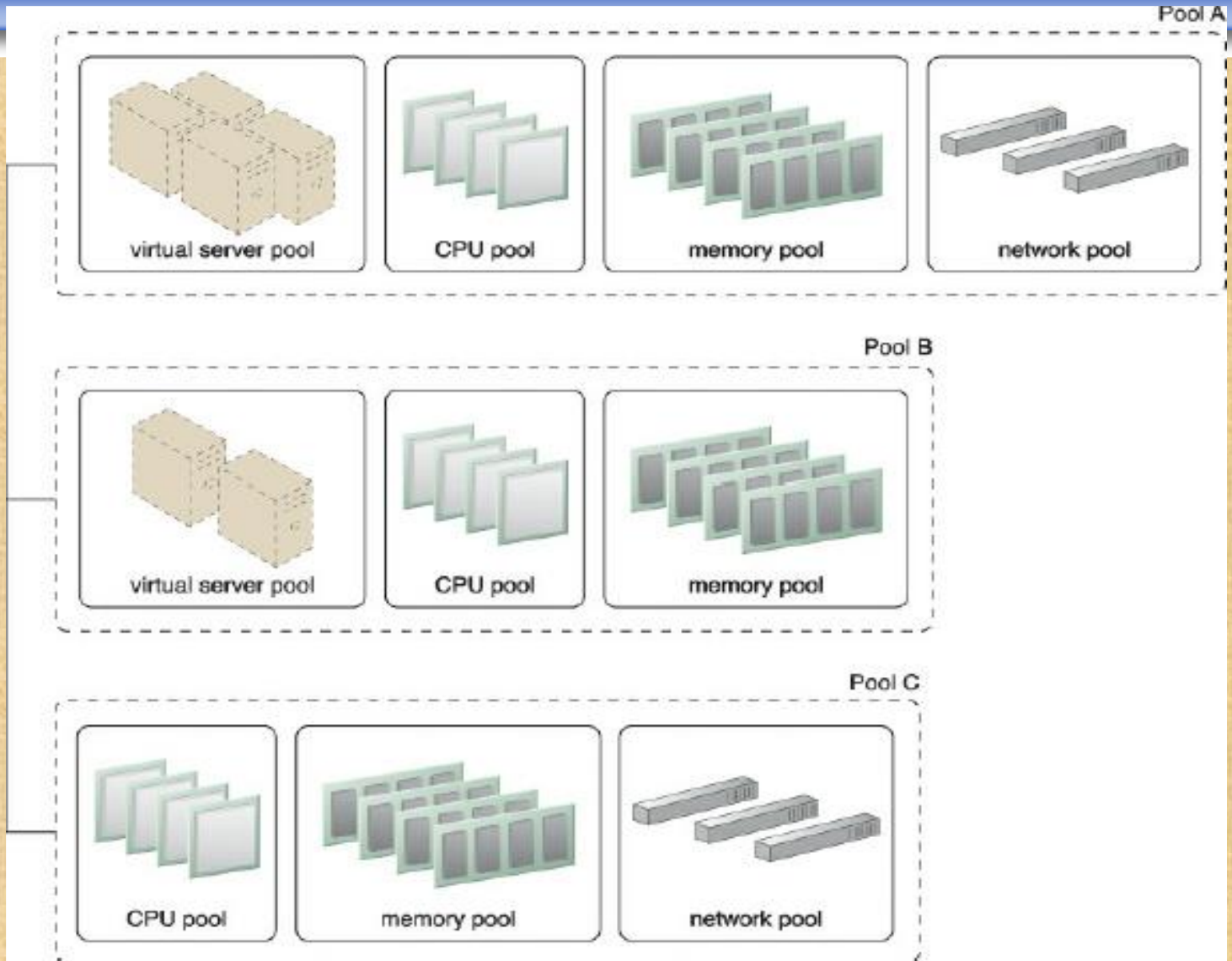
- Resource pools Sharing : It can become highly complex, with multiple pools created for specific cloud consumers or applications. A hierarchical structure can be established to form parent, sibling, and nested pools in order to facilitate the organization of diverse resource pooling requirements

- Sibling pools are isolated from one another so that each cloud consumer is only provided access to its respective pool.

- In Nested pool model : Larger pools are divided into smaller pools that individually group the same type of IT resources together (Figure 11.4). Nested pools can be used to assign resource pools to different departments or groups in the same cloud consumer organization.

VIT AP AMARAVATI

Sibling pools

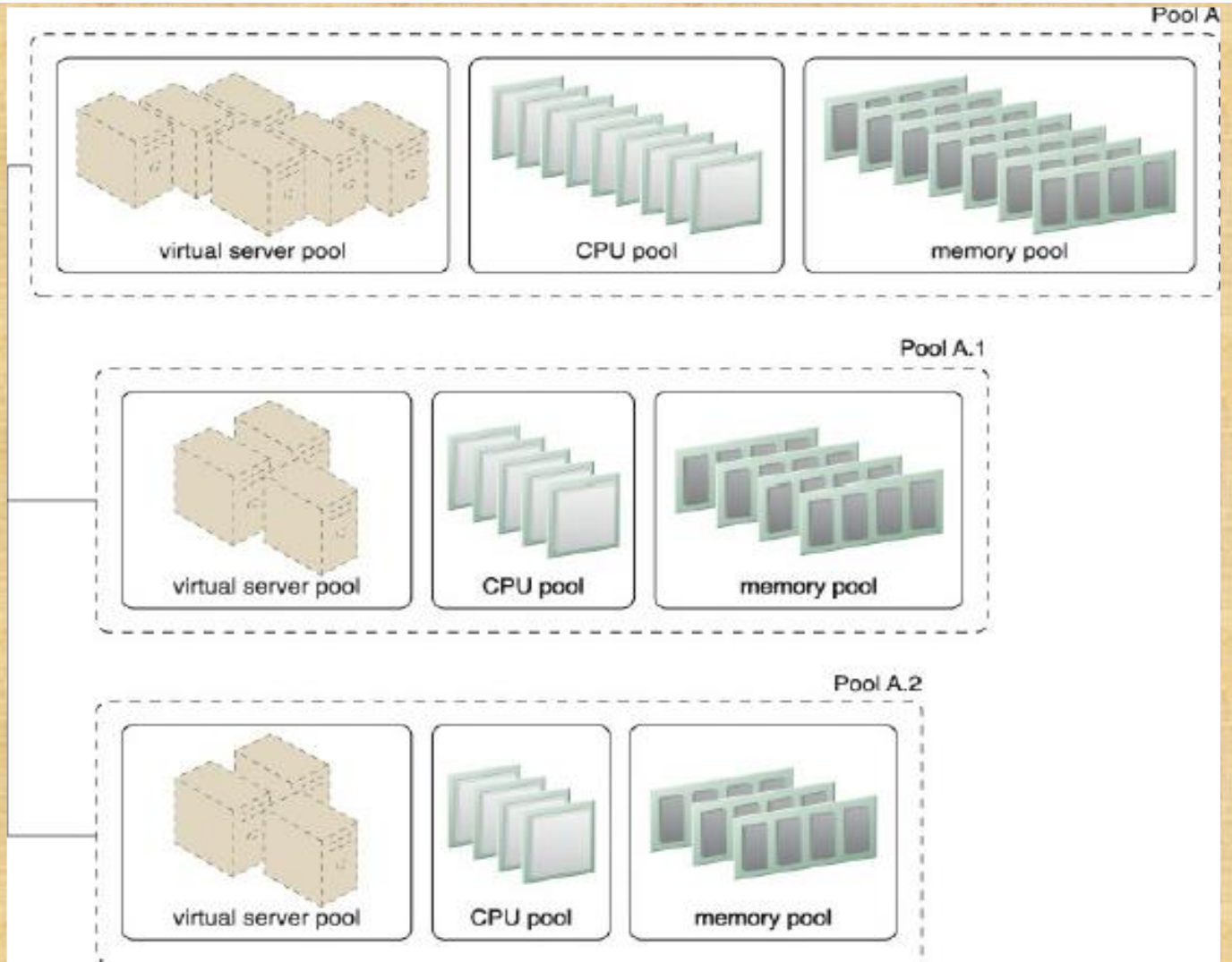


Sibling pools

- 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.

- Sibling resource pools are usually drawn from physically grouped IT resources, as opposed to IT resources that are spread out over different data centers.
- Sibling pools are isolated from one another so that each cloud consumer is only provided access to its respective pool.

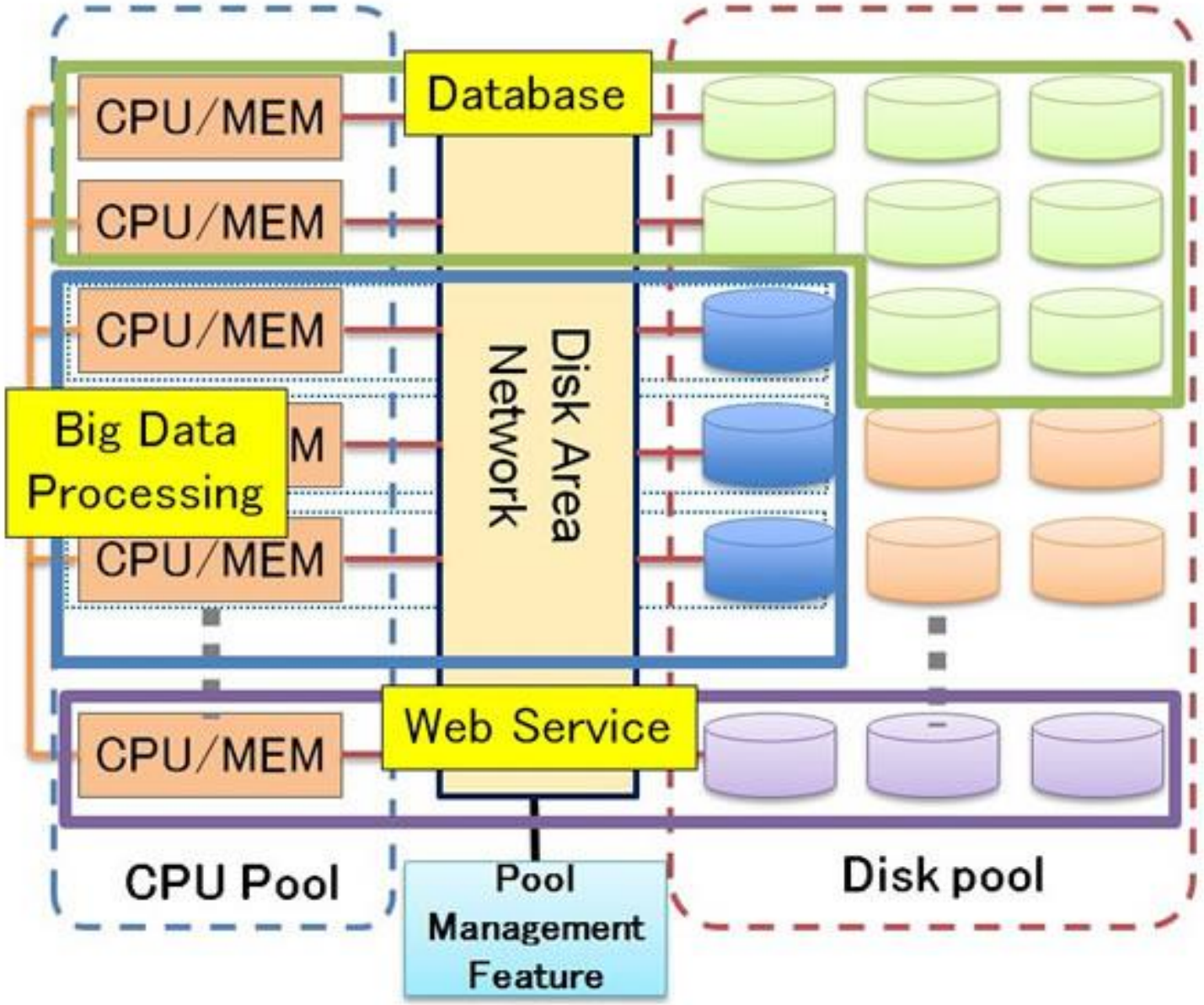
Nested Pool


Pool A
virtual server pool
CPU pool
memory pool
Pool A.1
virtual server pool
CPU pool
memory pool
Pool A.2
virtual server pool
CPU pool
memory pool

Nested Pool

- 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.
- After resources pools have been defined, multiple instances of IT resources from each pool can be created to provide an in-memory pool of “live” IT resources.

Resource Pooling



Resource Pooling Architecture

- In addition to commonly pooled mechanisms of cloud storage devices and virtual servers, the following mechanisms can be part of this resource pooling architecture:
  - Audit monitor
  - Cloud usage monitor
  - Hypervisor
  - Logical network perimeter
  - Pay-per-use monitor
  - Remote administration system
  - Resource management system
  - Resource replication

- Software based data center(VMWare Software)

  https://www.youtube.com/watch?v=gXwt_hi_3Ag

- Case Study

  https://www.fujitsu.com/global/about/resources/news/press-releases/2011/0926-01.html

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.

Dynamic Scalability Architecture

- The automated scaling listener is configured with workload thresholds that dictate when new IT resources need to be added to the workload processing.
- The following types of dynamic scaling are commonly used:
  - Dynamic Horizontal Scaling
  - Dynamic Vertical Scaling
  - Dynamic Relocation (resources are relocated with more capacity)

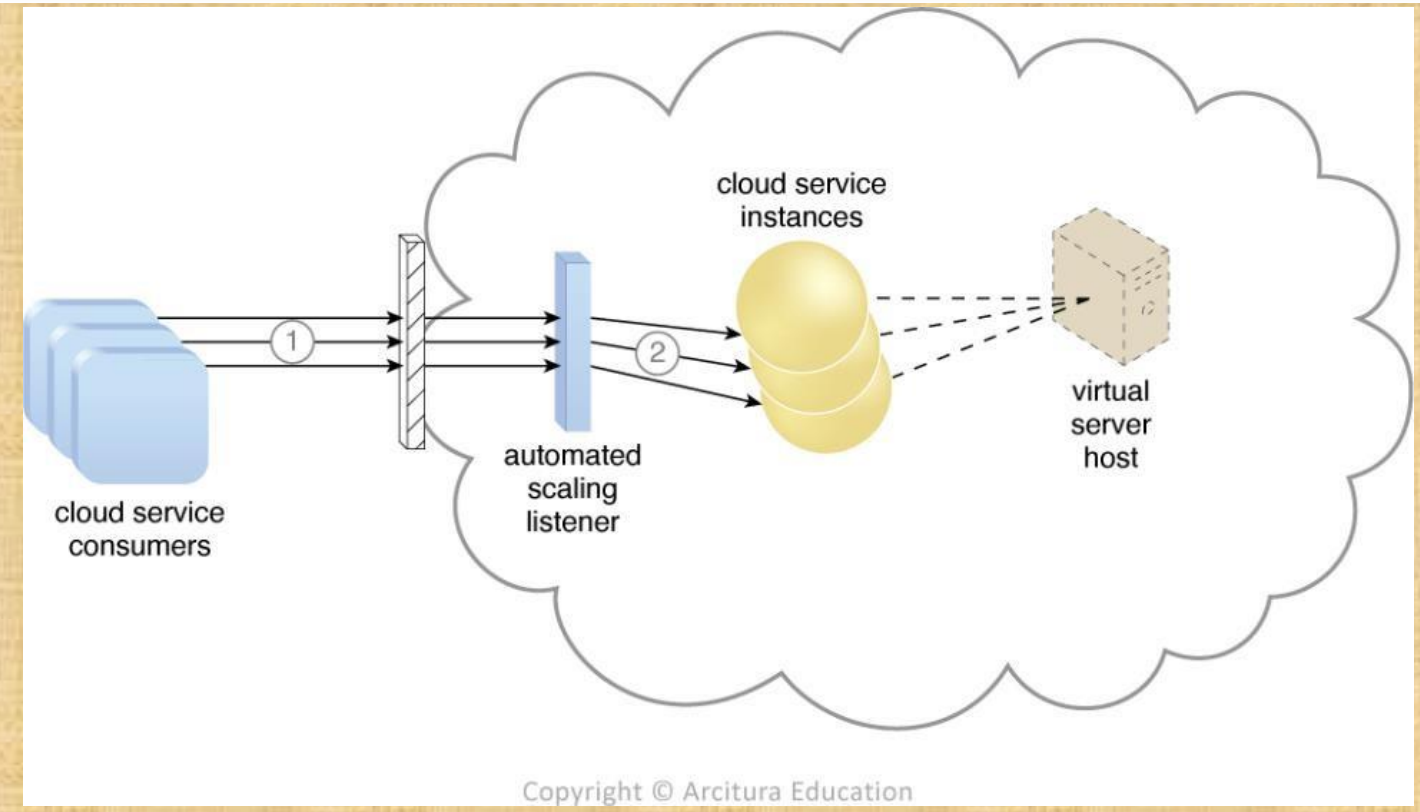
Dynamic Scalability Architecture

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.(storage)

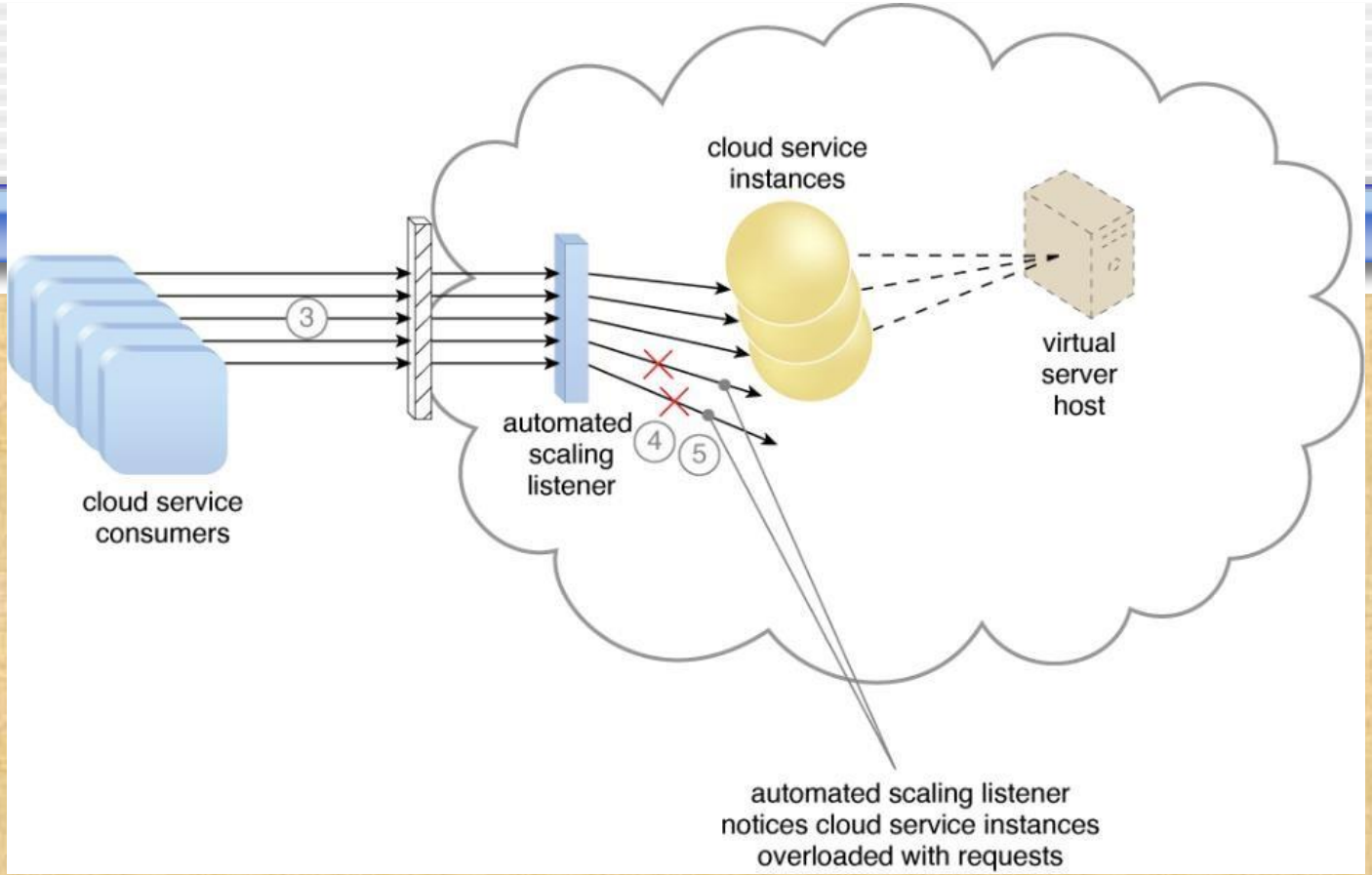
Dynamic Vertical Scaling – IT resource instances are scaled up and down when there is a need to adjust the processing capacity of a single IT resource. For example, a virtual server that is being overloaded can have its memory dynamically increased or it may have a processing core added.(memory)

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.

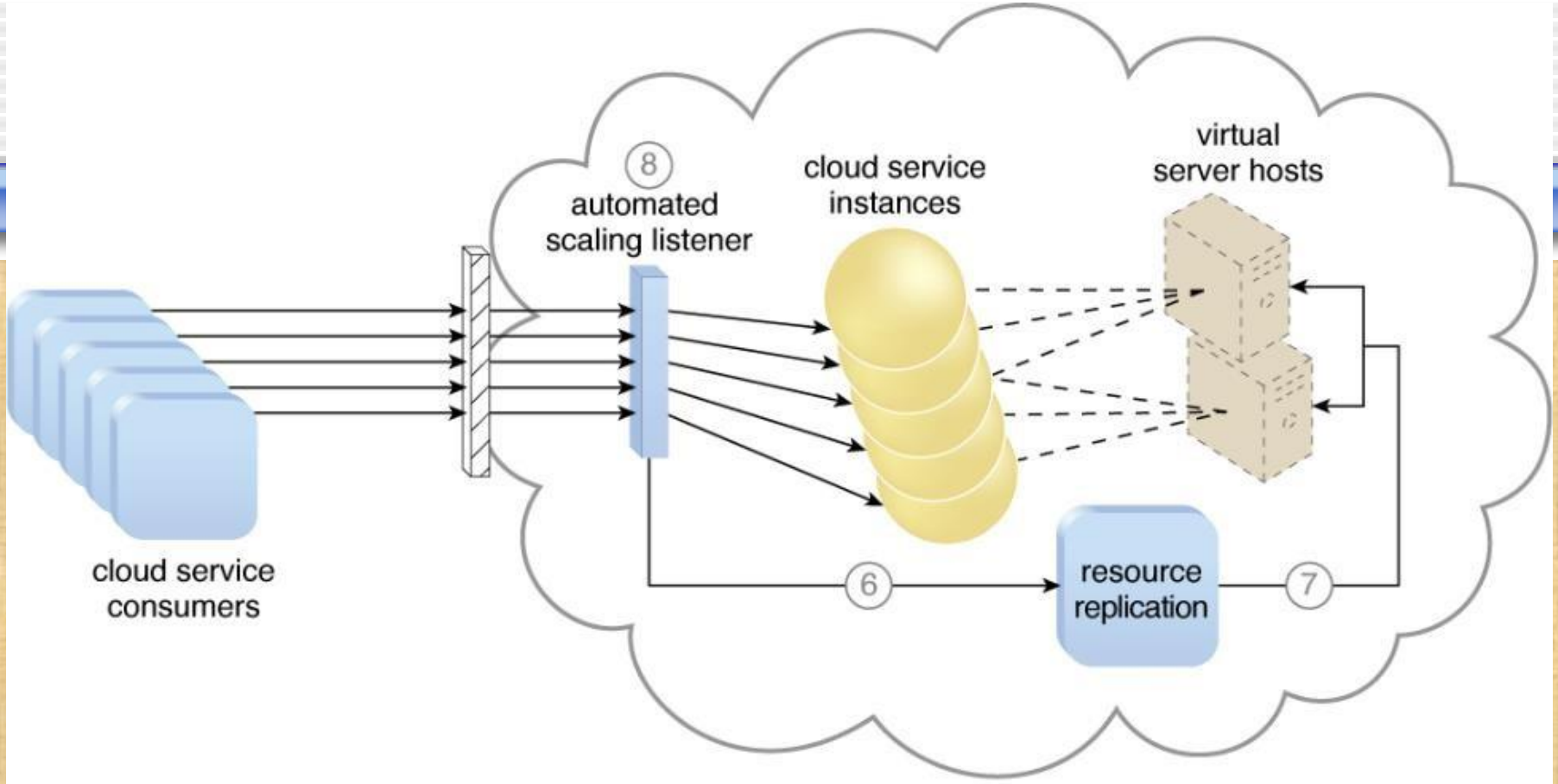
Dynamic Scalability Architecture



- 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).



- The number of service requests coming from cloud service consumers further 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).



- 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).

Dynamic Scalability Architecture

The dynamic scalability architecture can be applied to a range of IT resources, including virtual servers and cloud storage devices. 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 in response to the scaling of IT resources.

Azure App Service Auto-scale

- https://www.youtube.com/watch?v=cDNOTiBRwzA

ELASTIC RESOURCE CAPACITY ARCHITECTURE

What is elasticity?

In cloud computing, elasticity is defined as “the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible”.

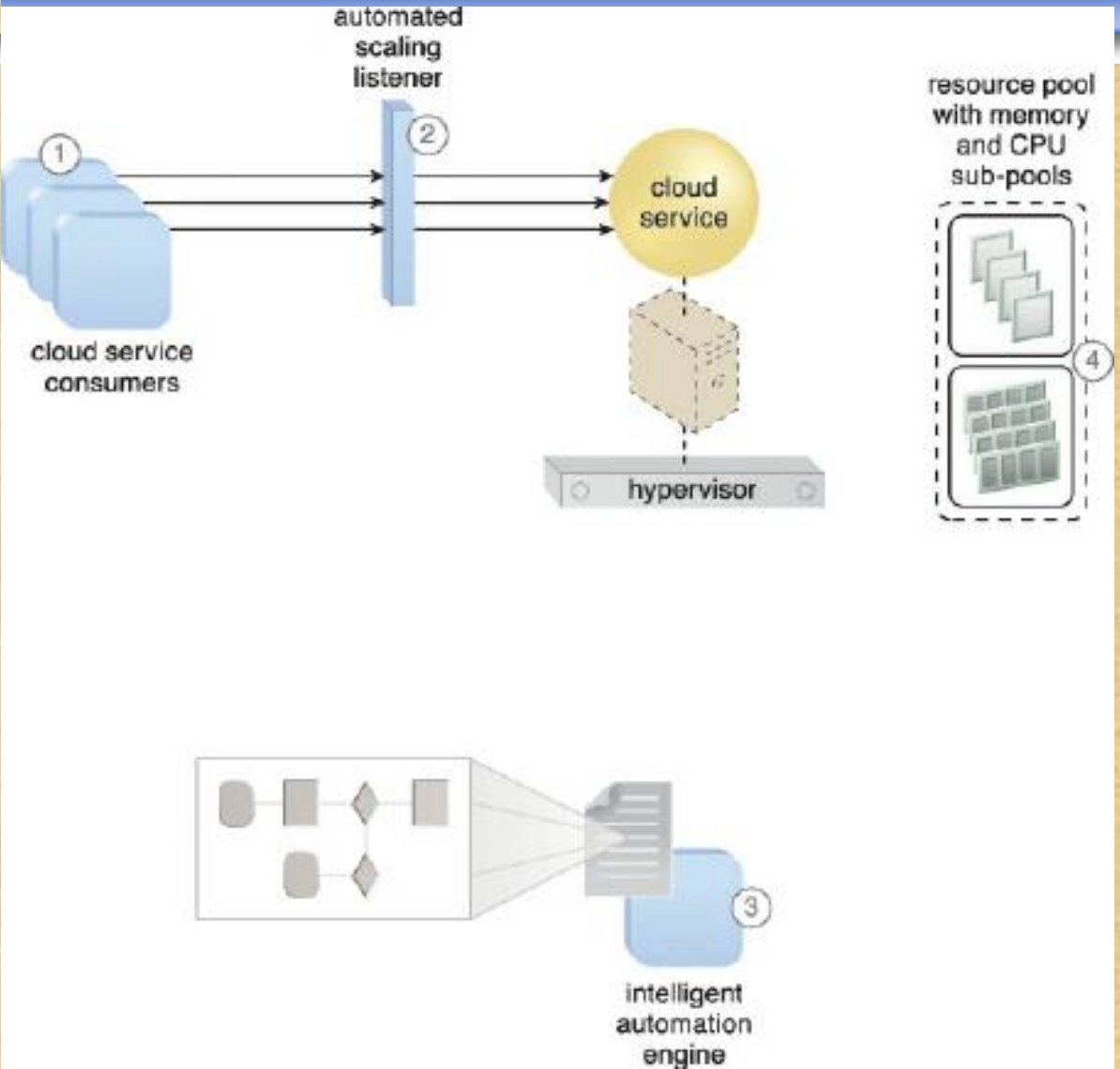
VIT AP AMARAVATI

ELASTIC RESOURCE CAPACITY ARCHITECTURE

- The Elastic Resource Capacity Architecture is primarily related to the dynamic provisioning of virtual servers, using a system that allocates and reclaims CPUs and RAM in immediate response to the fluctuating processing requirements of hosted IT resources (Figures 11.8 and 11.9).
- In an IaaS scenario, **dynamic provisioning** refers to the ability to acquire on demand virtual machines in order to increase the capability of the resulting distributed system and then release them

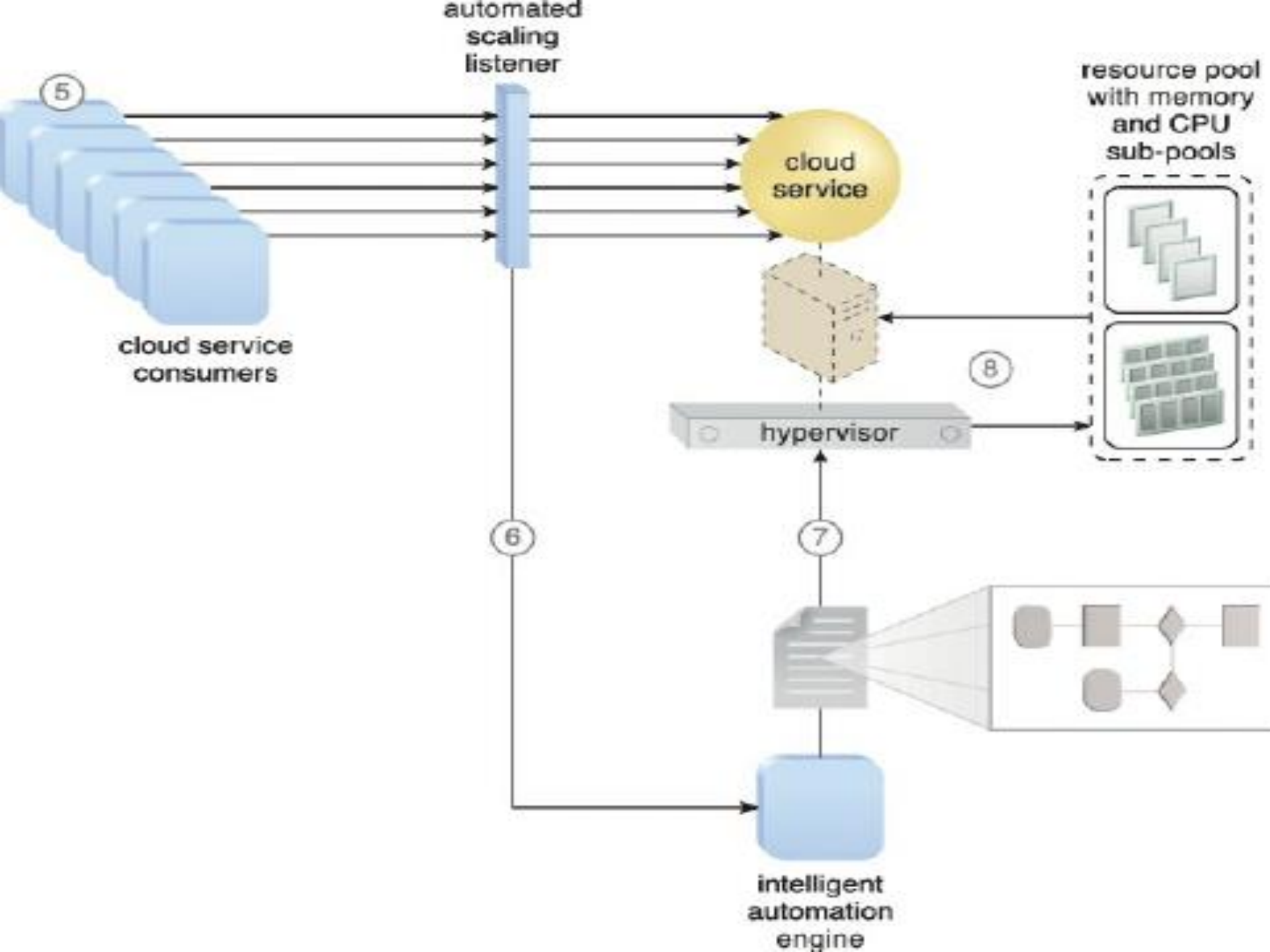
VIT AP AMARAVATI

ELASTIC RESOURCE CAPACITY ARCHITECTURE



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).


automated
scaling
listener
5
cloud service
consumers
cloud
service
resource pool
with memory
and CPU
sub-pools
8
hypervisor
6
7
intelligent
automation
engine

ELASTIC RESOURCE CAPACITY ARCHITECTURE

Cloud service consumer requests increase **(5),** causing the automated scaling listener to signal the intelligent automation engine to execute the script **(6).**

The script runs the workflow logic that signals the hypervisor to allocate more IT resources from the resource pools **(7).**

The hypervisor allocates additional CPU and RAM to the virtual server, enabling the increased workload to be handled **(8).**

VIT AP AMARAVATI

Intelligent Automation Engine

- The intelligent automation engine automates administration tasks by executing scripts that contain workflow logic.
- Some additional mechanisms that can be included in this cloud architecture are the following:
- Cloud Usage Monitor – Specialized cloud usage monitors collect resource usage information on IT resources before, during, and after scaling, to help define the future processing capacity thresholds of the virtual servers.
- Pay-Per-Use Monitor – The pay-per-use monitor is responsible for collecting resource usage cost information as it fluctuates with the elastic provisioning.
- Resource Replication – Resource replication is used by this architectural model to generate new instances of the scaled IT resources.

Service Load Balancing Architecture

- The service load balancing architecture can be considered 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 the workloads.

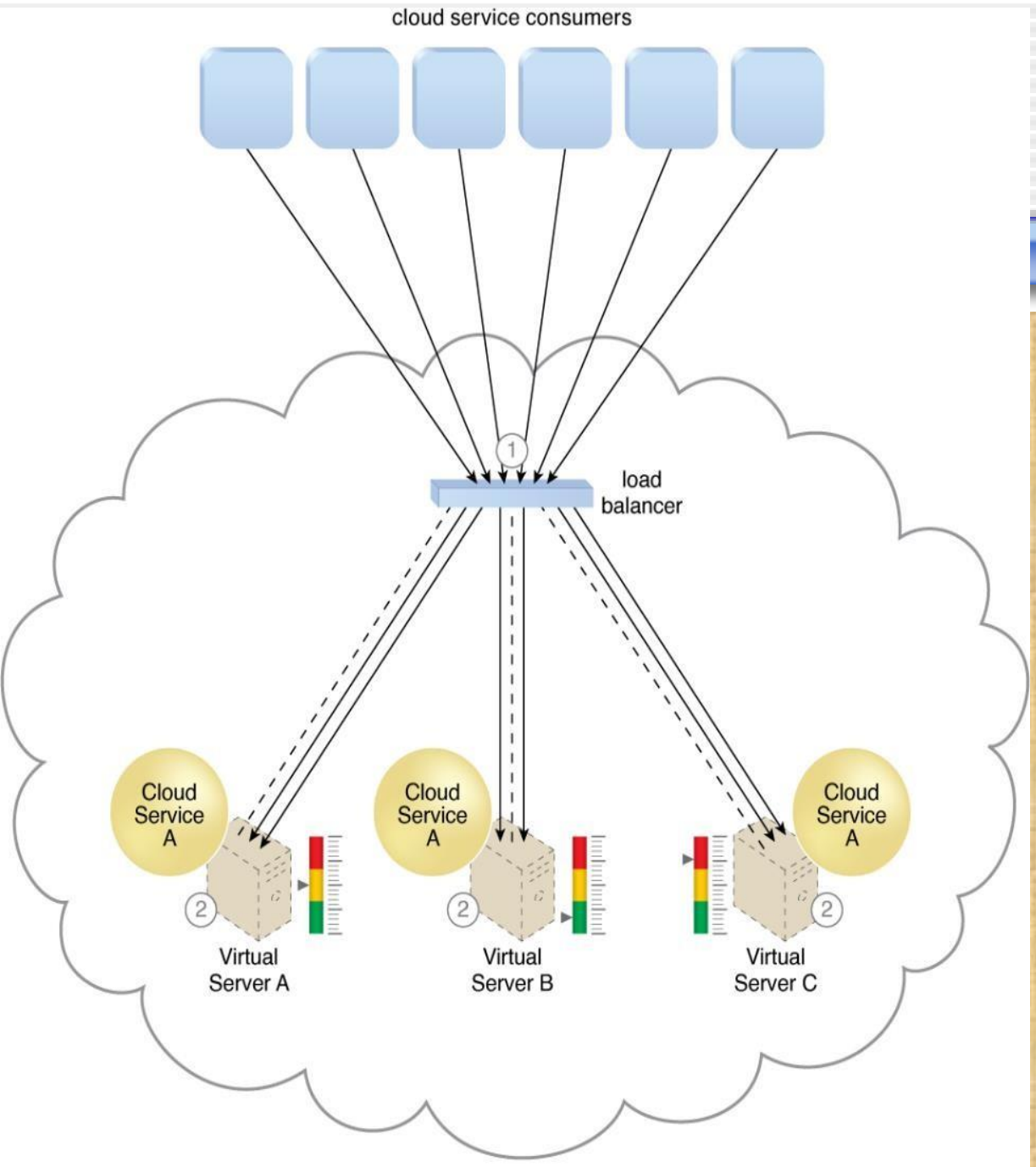
VIT AP AMARAVATI

Service Load Balancing Architecture

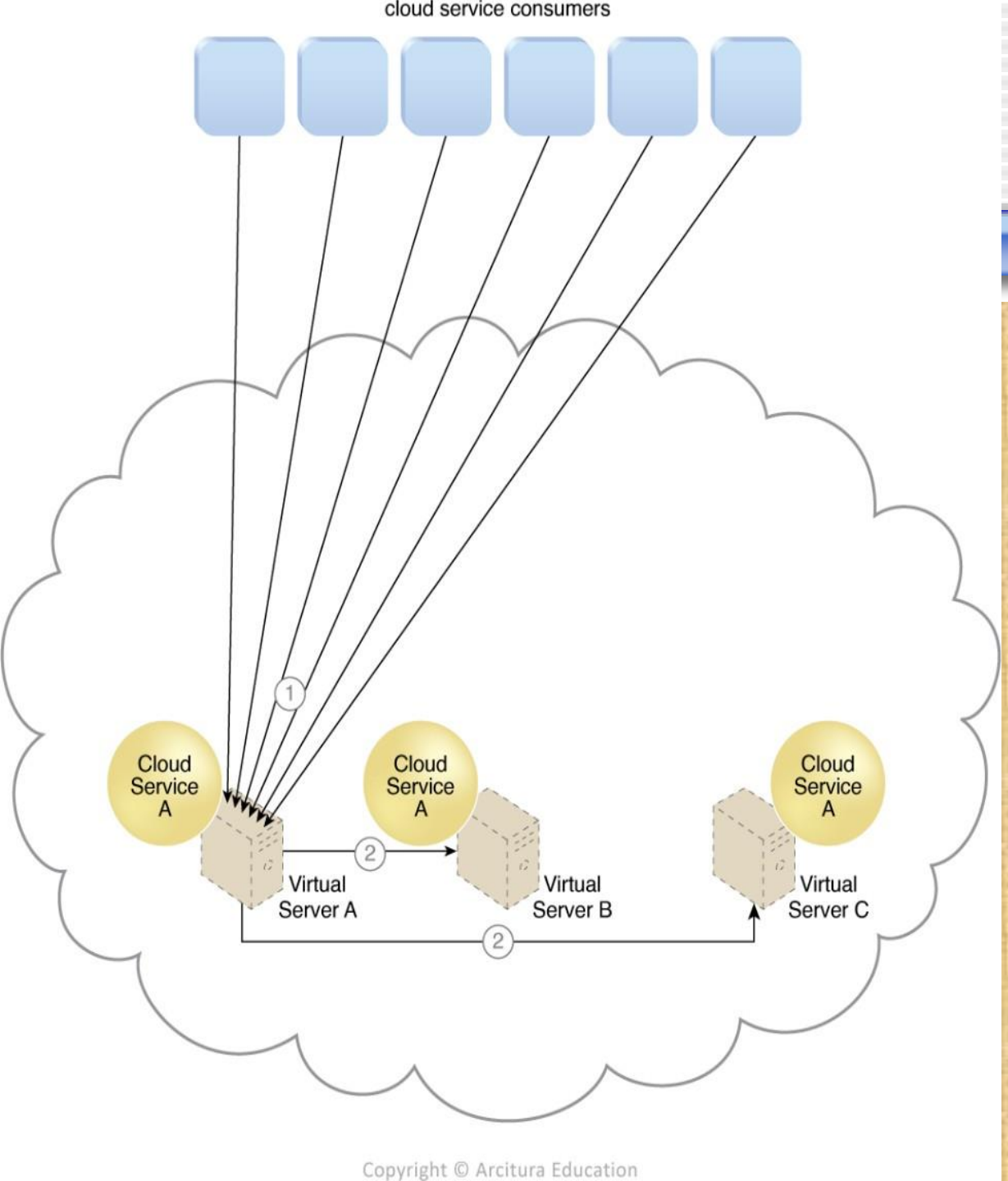
- The load balancer can be positioned either independent of the cloud services and their host servers, or built-in as part of the application or server's environment.

- The service load balancing architecture can involve the following mechanisms in addition to load balancing:

  - Cloud usage monitor
  - Resource cluster
  - Resource replication



- 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).



- 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).

service load balancing architecture

- 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.

VIT AP AMARAVATI

Cloud Bursting Architecture

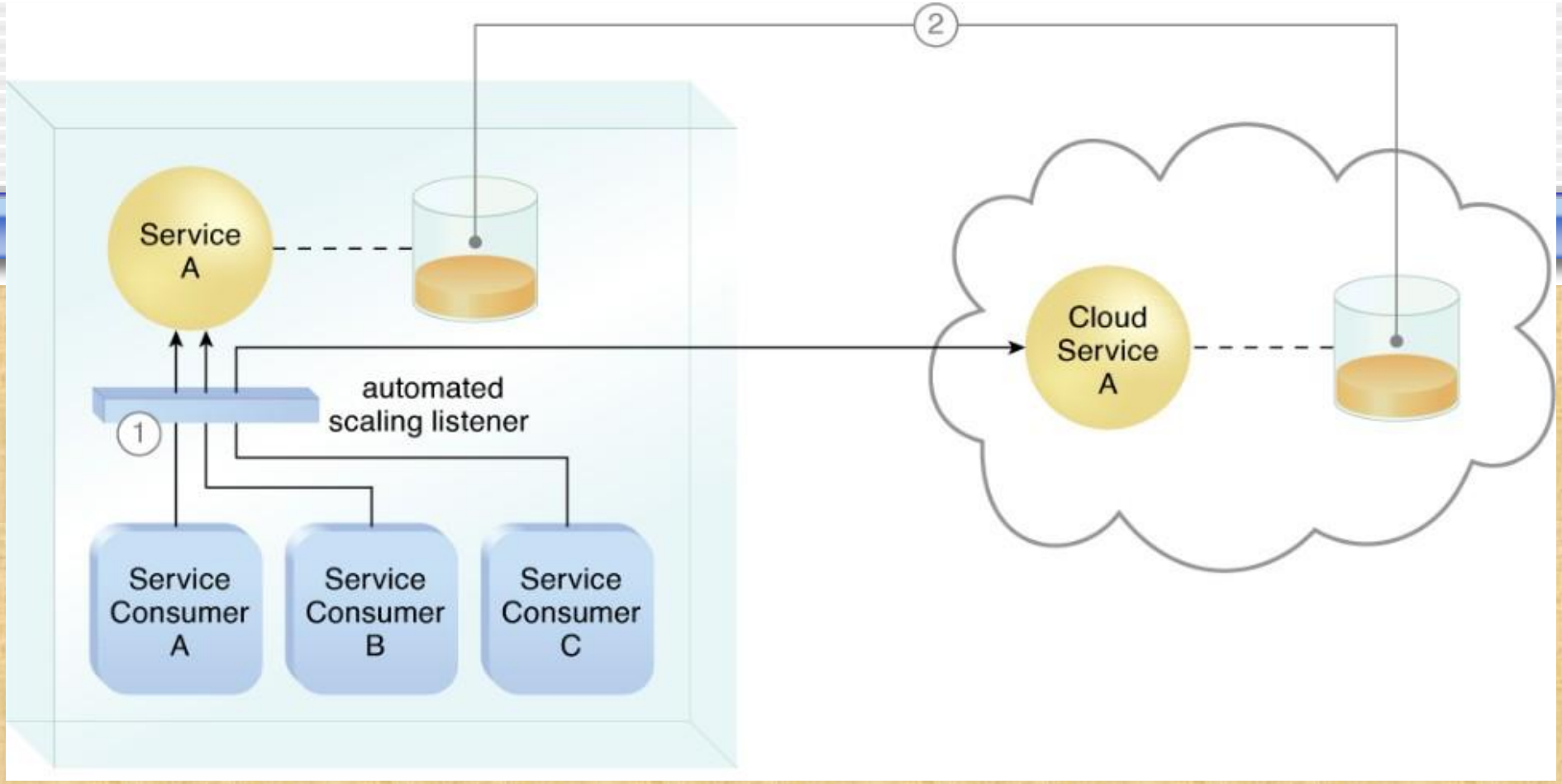
- The cloud bursting architecture establishes a form of dynamic scaling that scales or "bursts out" on- premise IT resources into a cloud whenever predefined capacity thresholds have been reached.
- The corresponding cloud-based IT resources are redundantly pre-deployed but remain inactive until cloud bursting occurs, while burst-in when they are no longer required.
- The foundation of this architectural model is based on the automated scaling listener and resource replication mechanisms.

VIT AP AMARAVATI

Cloud Bursting Architecture

- Cloud bursting is a flexible scaling architecture that provides cloud consumers with the option of using cloud-based IT resources only to meet higher usage demands.

- The foundation of this architectural model is based on the automated scaling listener, to determine when to redirect requests, and resource replication, to maintain synchronicity between on-premise and cloud-based IT resources in relation to state information.

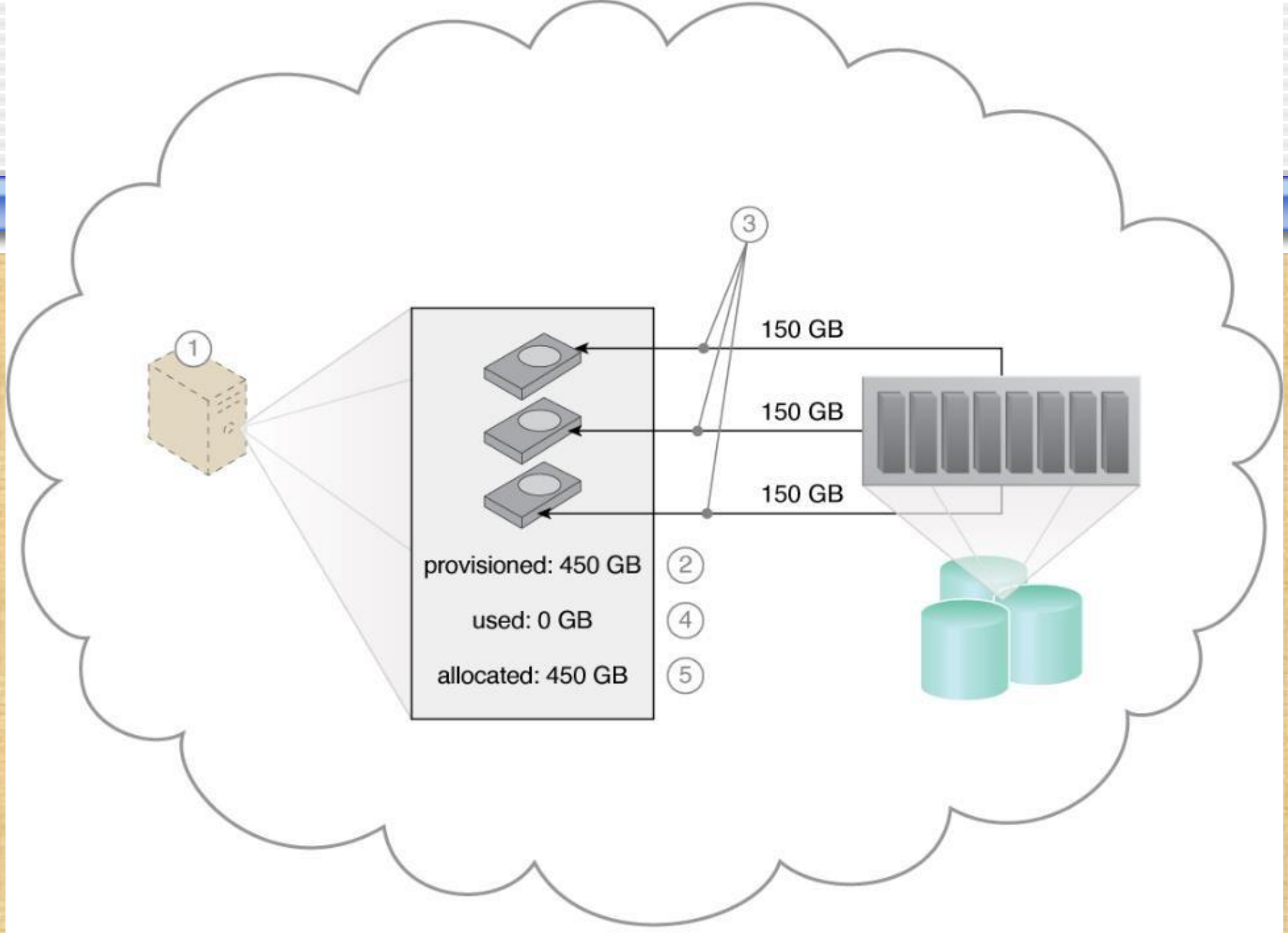


- An automated scaling listener monitors the usage of on-premise Service A, and redirects Service Consumer C's request to Service A's redundant implementation in the cloud (Cloud Service A) once Service A's usage threshold has been exceeded (1). A resource replication system is used to keep state management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

Elastic Disk Provisioning Architecture

- The 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.

- Oppositely, cloud consumers are commonly charged for cloud-based storage space based on disk capacity allocation.

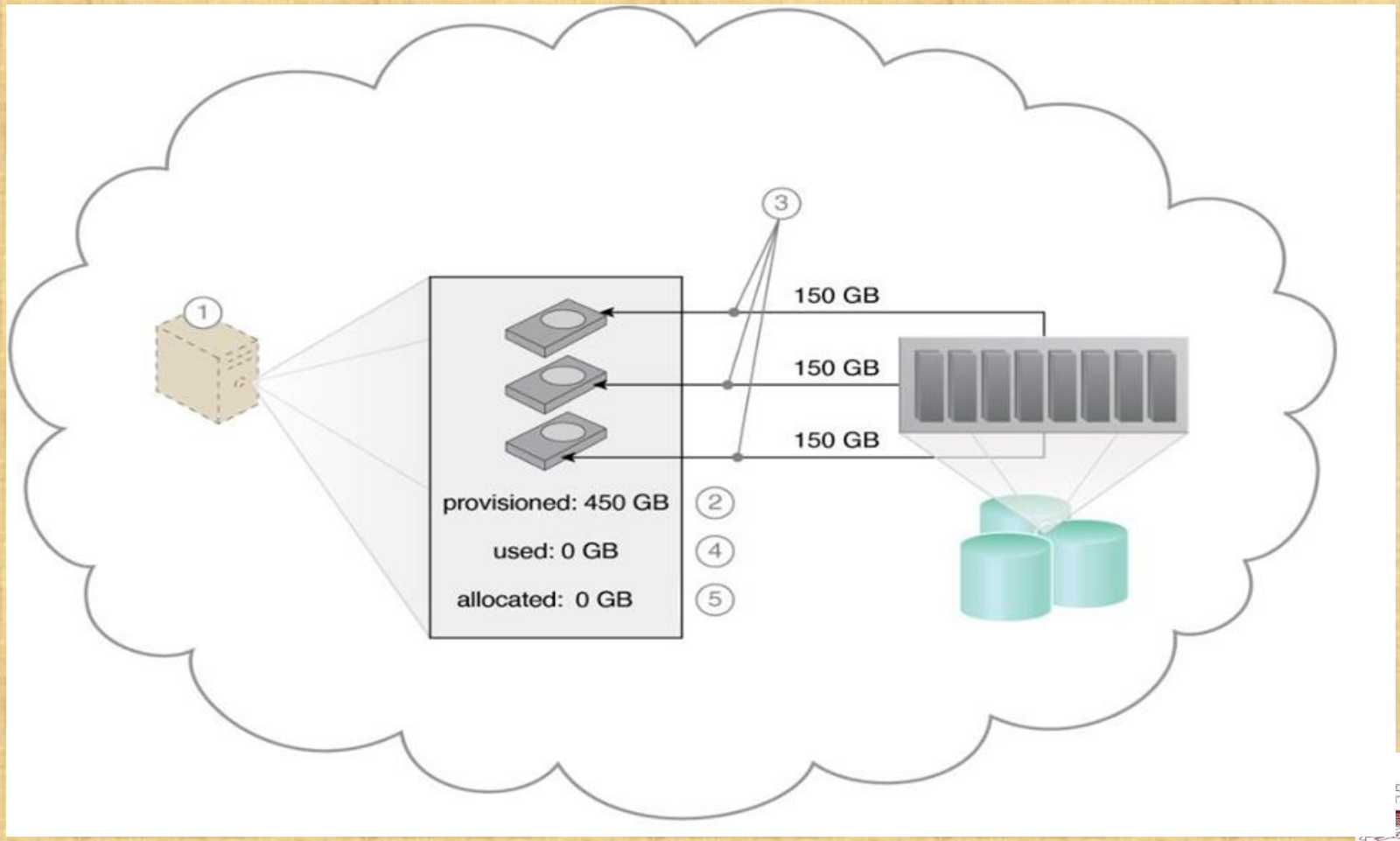


Elastic Disk Provisioning Architecture

- 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 450), the cloud consumer is charged for 450 GB disk space usage (5).

VIT AP AMARAVATI

Elastic Disk Provisioning Architecture

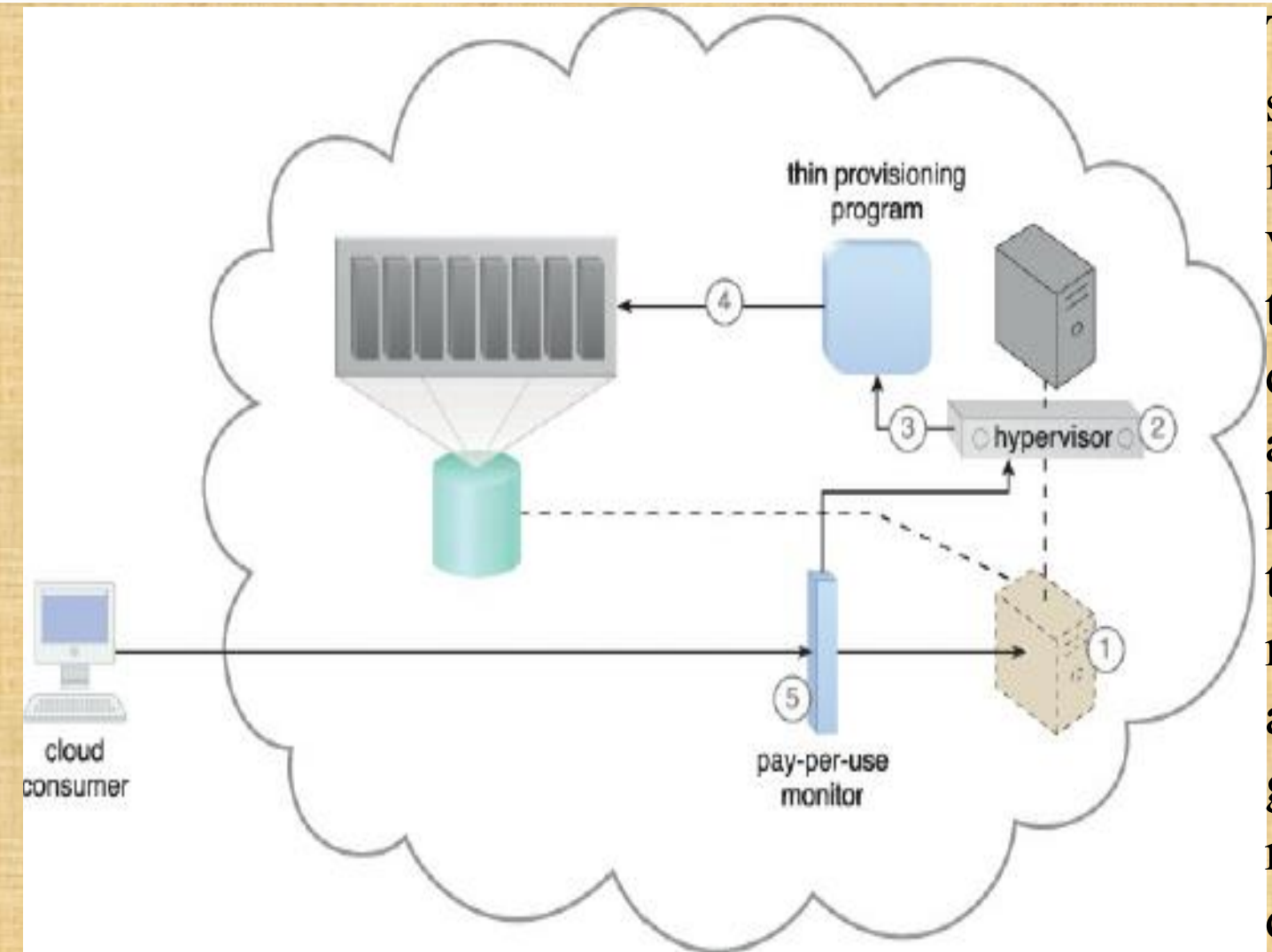


Elastic Disk Provisioning Architecture

- 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).

VIT AP AMARAVATI

Elastic Disk Provisioning Architecture



Thin-provisioning software is installed on virtual servers that process dynamic storage allocation via the hypervisor, while the pay-per-use monitor tracks and reports granular billing-related disk usage data

Elastic Disk Provisioning Architecture

- 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).**

VIT AP AMARAVATI

Elastic Disk Provisioning Architecture

The following mechanisms can be included in this architecture in addition to the cloud storage device, virtual server, hypervisor, and pay-per-use monitor:

- Cloud Usage Monitor – Specialized cloud usage monitors can be used to track and log storage usage fluctuations.

- Resource Replication – Resource replication is part of an elastic disk provisioning system when conversion of dynamic thin-disk storage into static thick-disk storage is required.

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.

Redundant Storage Architecture

- 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. A storage requests to the secondary device whenever the primary device fails.

- The storage service gateway is a component that acts as the external interface to cloud storage services, and is capable of automatically redirecting cloud consumer requests whenever necessary.

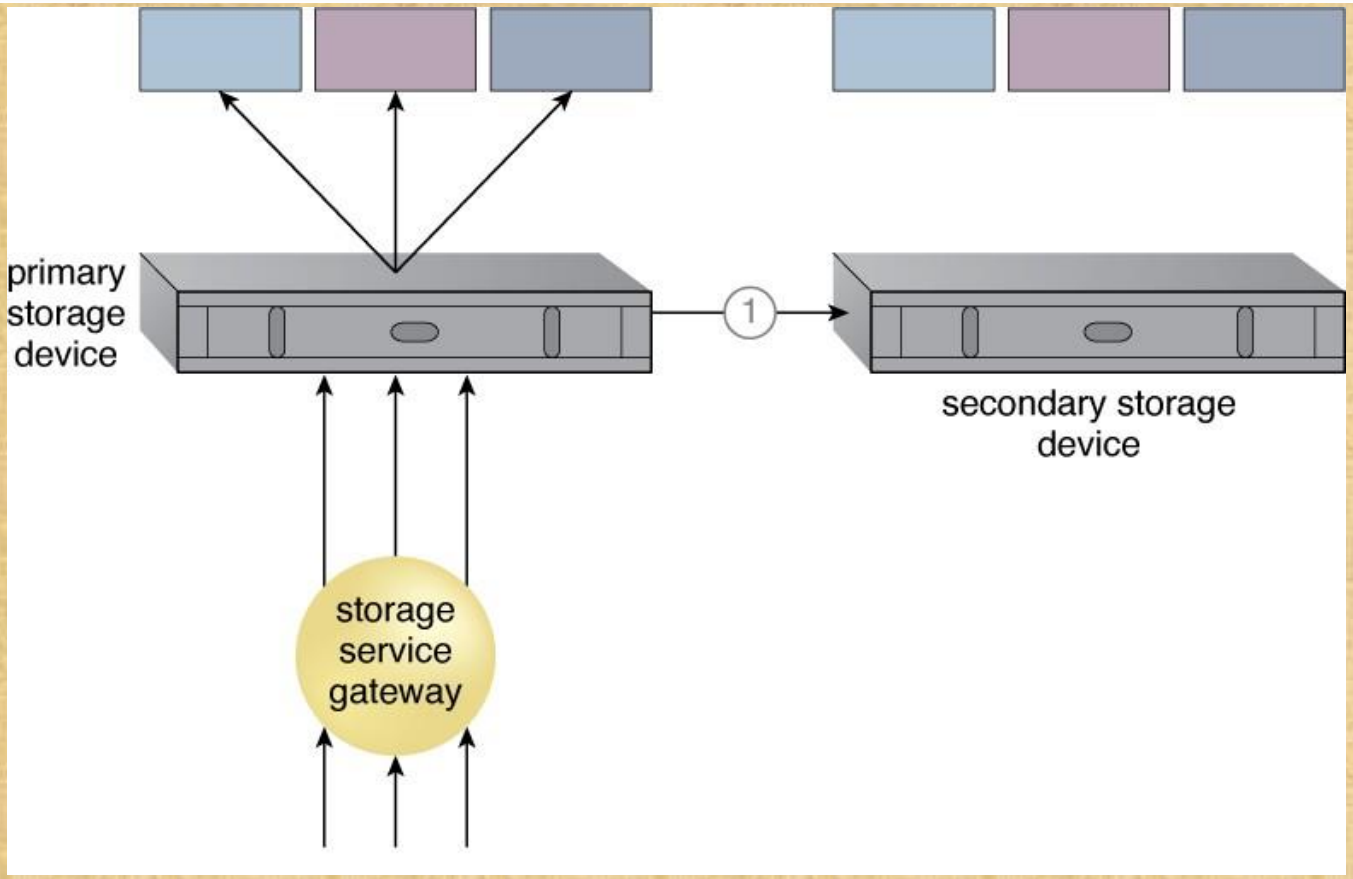
VIT AP AMARAVATI

Redundant Storage Architecture

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

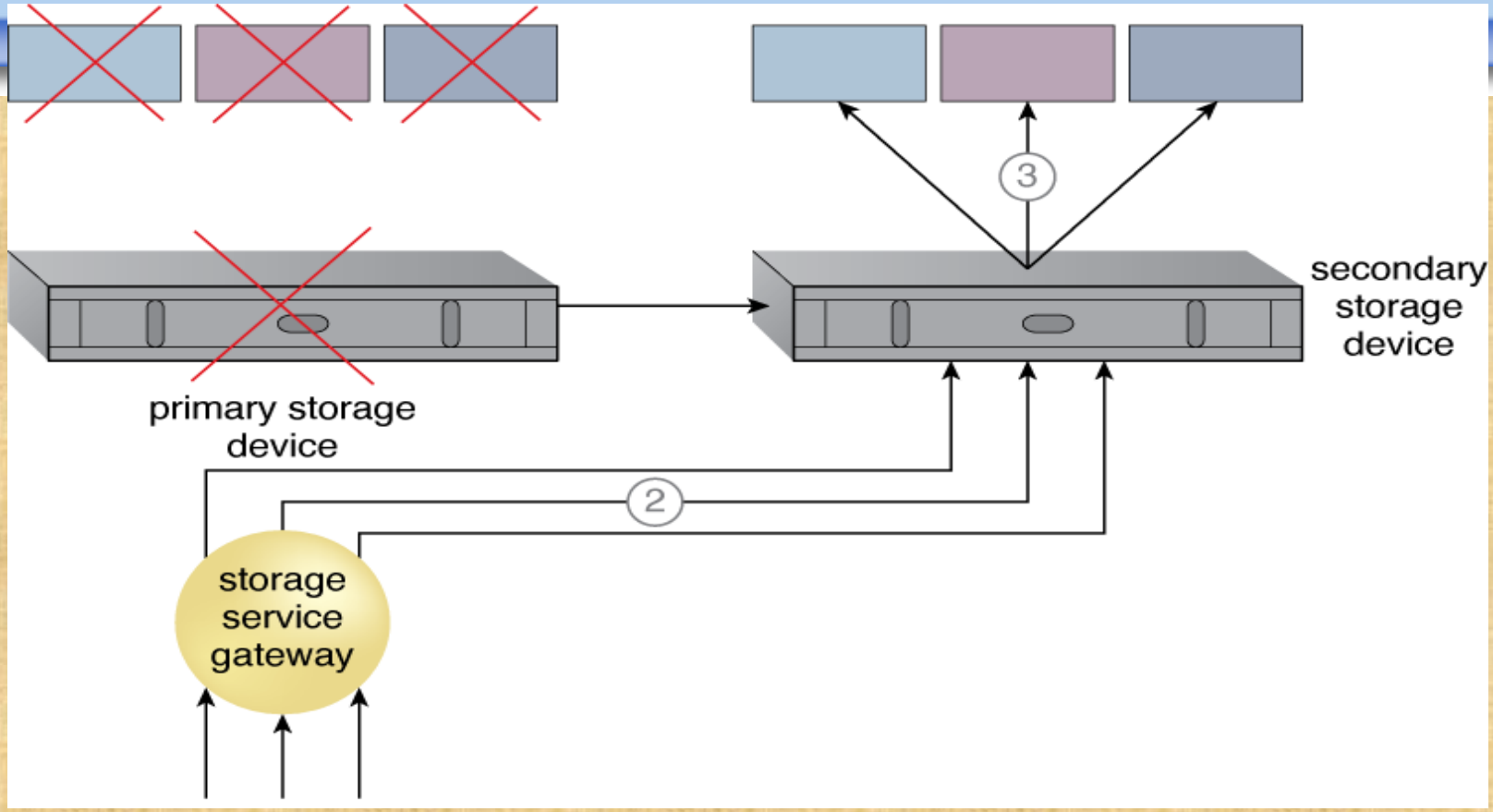
The storage service gateway is a component that acts as the external interface to cloud storage services, and is capable of automatically redirecting cloud consumer requests whenever the location of the requested data has changed.

Redundant Storage Architecture



- The primary cloud storage device is routinely replicated to the secondary cloud storage device .

Redundant Storage Architecture



- 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).

Redundant Storage Architecture

- This architecture primarily relies on a storage replication system that keeps the primary cloud storage device synchronized with secondary devices.
- Cloud providers may locate secondary cloud storage devices in a different geographical region than the primary cloud storage device, usually for economic reasons.
- Some cloud providers use storage devices with dual array and storage devices in a different physical location for cloud balancing and disaster recovery purposes.

Redundant Storage Architecture

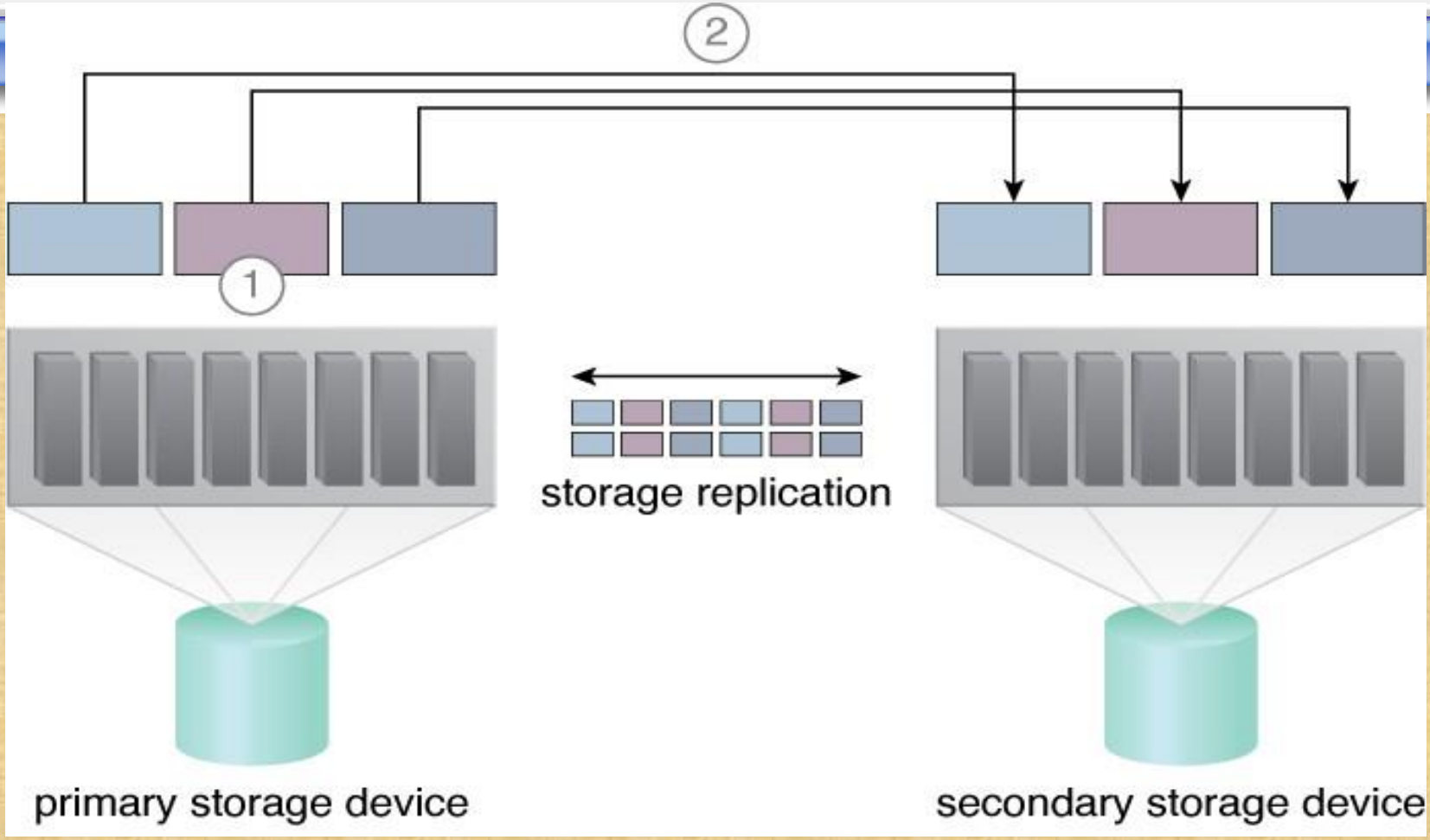


Figure 11.18 - Storage replication is used to keep the redundant storage device synchronized with the primary storage device.

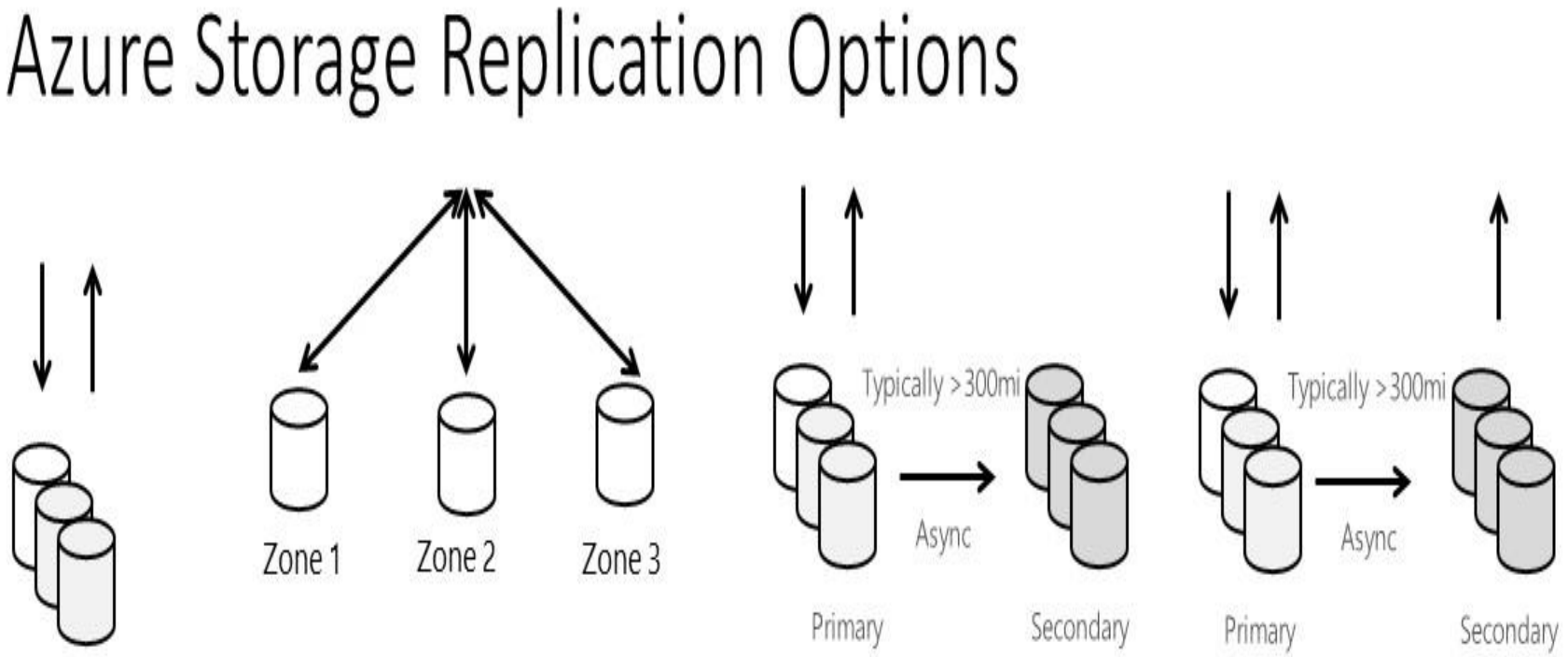
A Case Study : AZURE

Azure has options around the data redundancy using Redundant Storage.

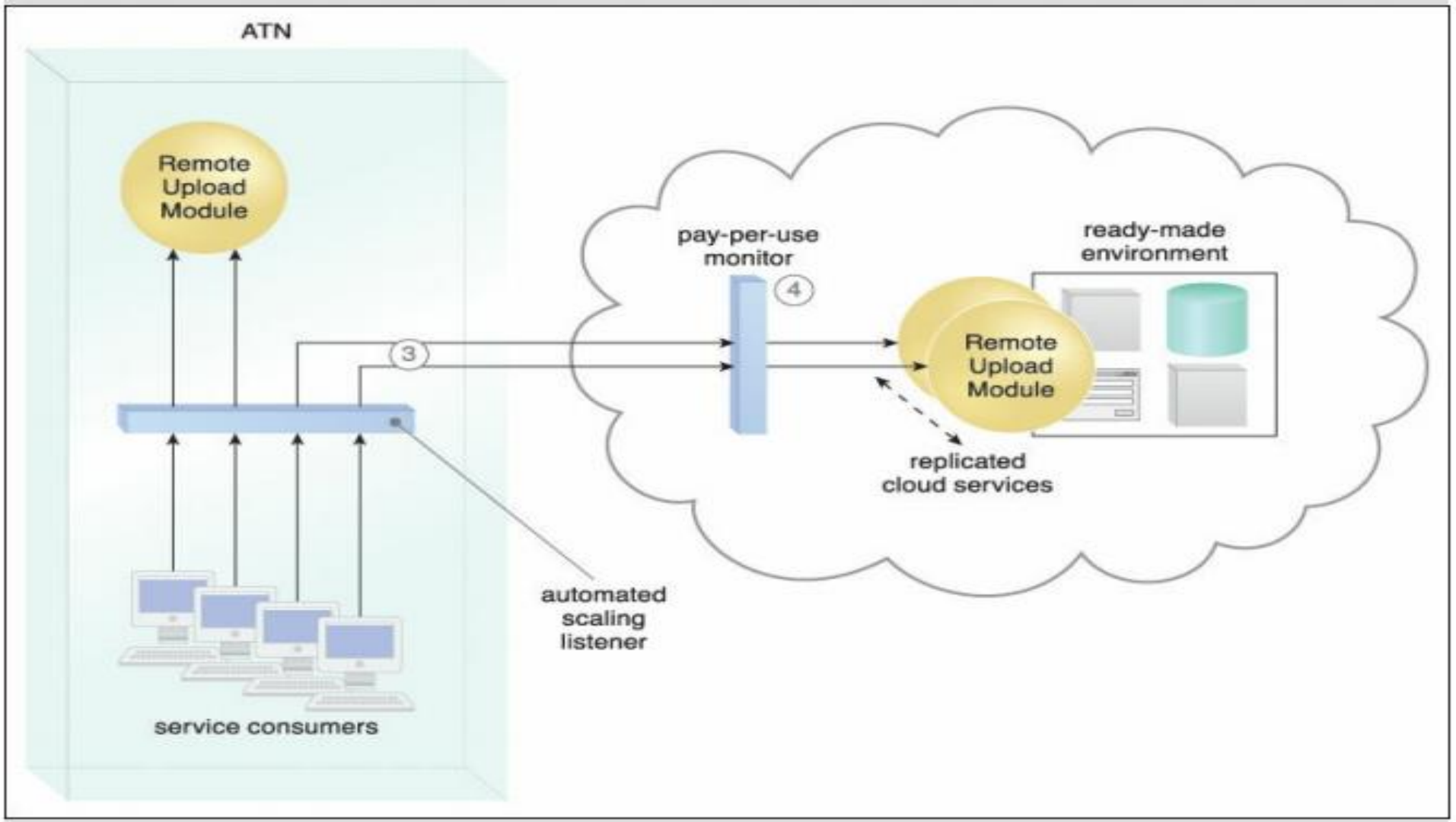
- **LRS (Locally redundant storage)**
- **ZRS (Zone redundant storage)**
- **GRS (Geo-redundant storage)**
- **RA-GRS (Read-Access Geo-redundant)**

VIT AP AMARAVATI

A Case Study


Azure Storage Replication Options
Zone 1
Zone 2
Zone 3
Typically >300mi
Async
Primary
Secondary
Typically >300mi
Async
Primary
Secondary
LRS
ZRS
GRS
RA-GRS

A Case Study


ATN
Remote Upload Module
pay-per-use monitor
4
ready-made environment
3
Remote Upload Module
replicated cloud services
automated scaling listener
service consumers

Conclusion