Unit - V

Cloud Computing Architecture

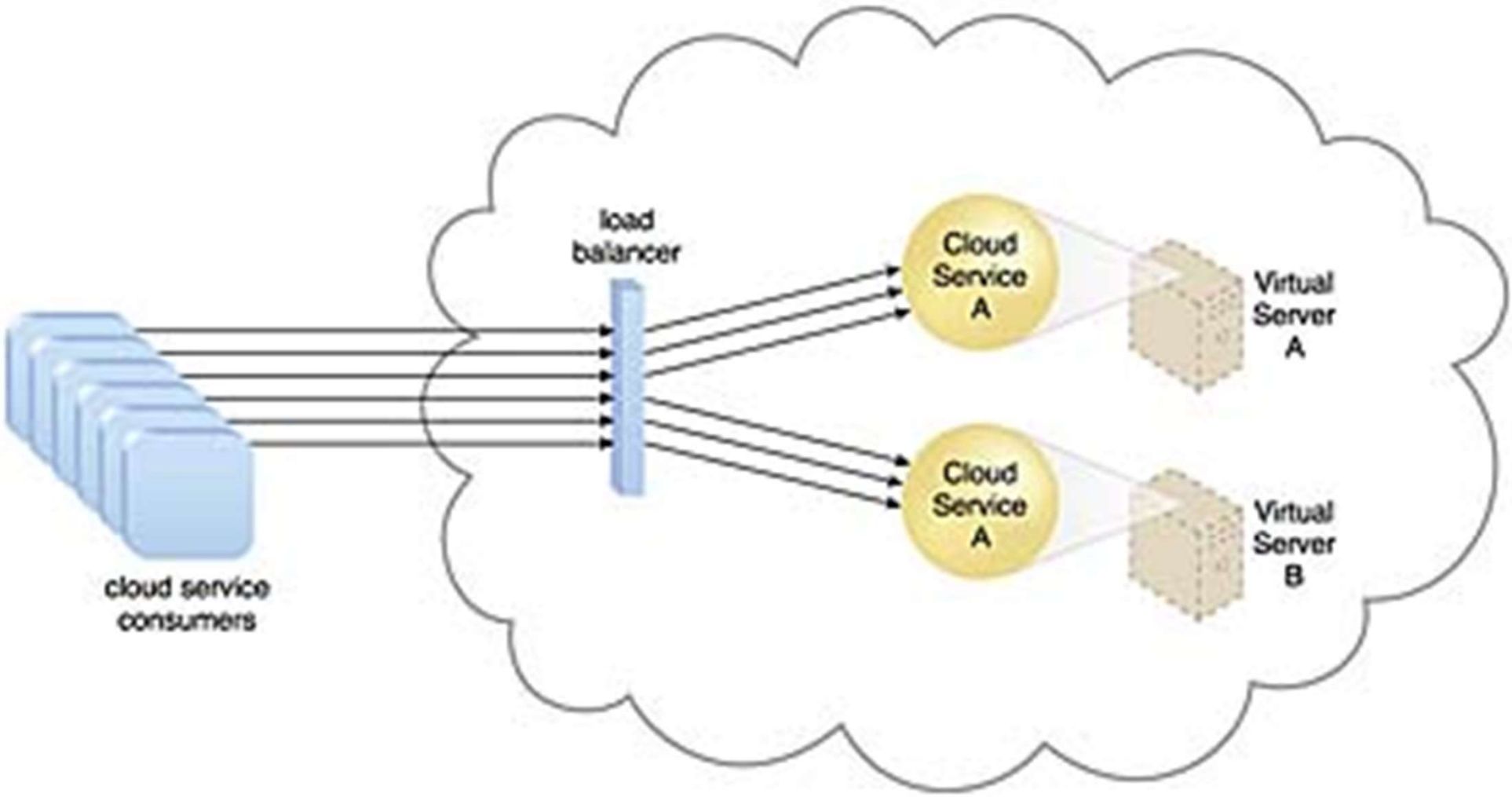
Fundamental Cloud Architectures - Workload Distribution Architecture - Resource Pooling Architecture - Dynamic Scalability Architecture – Elastic Resource Capacity Architecture -Service Load Balancing Architecture – Cloud Bursting Architecture - Elastic Disk Provisioning Architecture – Redundant Storage Architecture. Cloud Computing Reference Architecture (CCRA):

Introduction, benefits of CCRA, Migrating into a Cloud: Introduction, Challenges while migrating to Cloud, Broad approaches to migrating into the cloud, Seven-step model of migration into a cloud, Migration Risks and Mitigation.

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.

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.


load
balancer
Cloud
Service
A
Virtual
Server
A
Cloud
Service
A
Virtual
Server
B
cloud service
consumers

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.

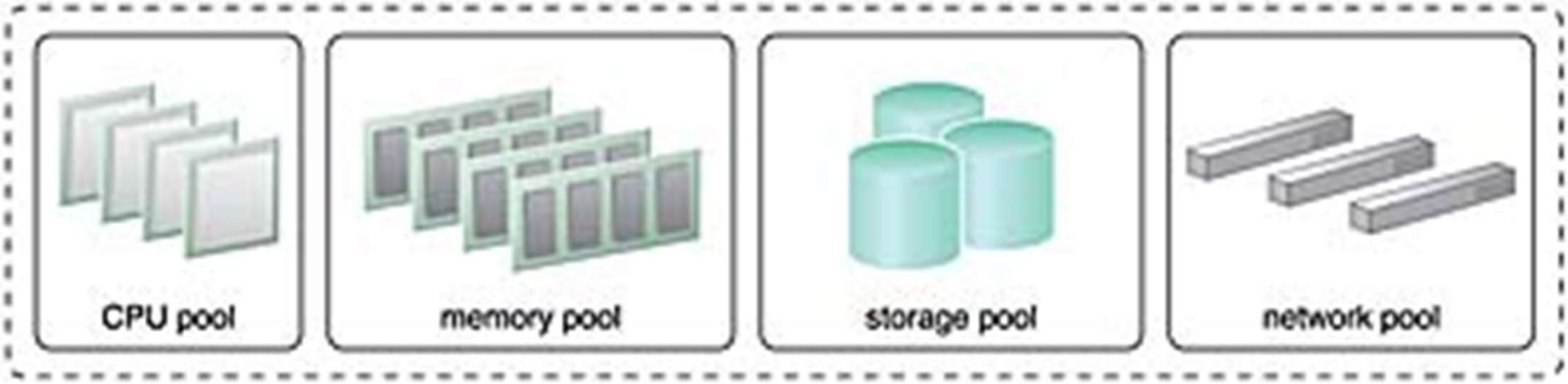
In addition to the base load balancer mechanism, and the virtual server and cloud storage device mechanisms to which load balancing can be applied, the following mechanisms can also be part of this cloud architecture:

- *Audit Monitor* – When distributing runtime workloads, the type and geographical location of the IT resources that process the data can determine whether monitoring is necessary to fulfill legal and regulatory requirements.
- *Cloud Usage Monitor* – Various monitors can be involved to carry out runtime workload tracking and data processing.
- *Hypervisor* – Workloads between hypervisors and the virtual servers that they host may require distribution.
- *Logical Network Perimeter* – The logical network perimeter isolates cloud consumer network boundaries in relation to how and where workloads are distributed.
- *Resource Cluster* – Clustered IT resources in active/active mode are commonly used to support workload balancing between different cluster nodes.
- *Resource Replication* – This mechanism can generate new instances of virtualized IT resources in response to runtime workload distribution demands.

Resource Pooling Architecture

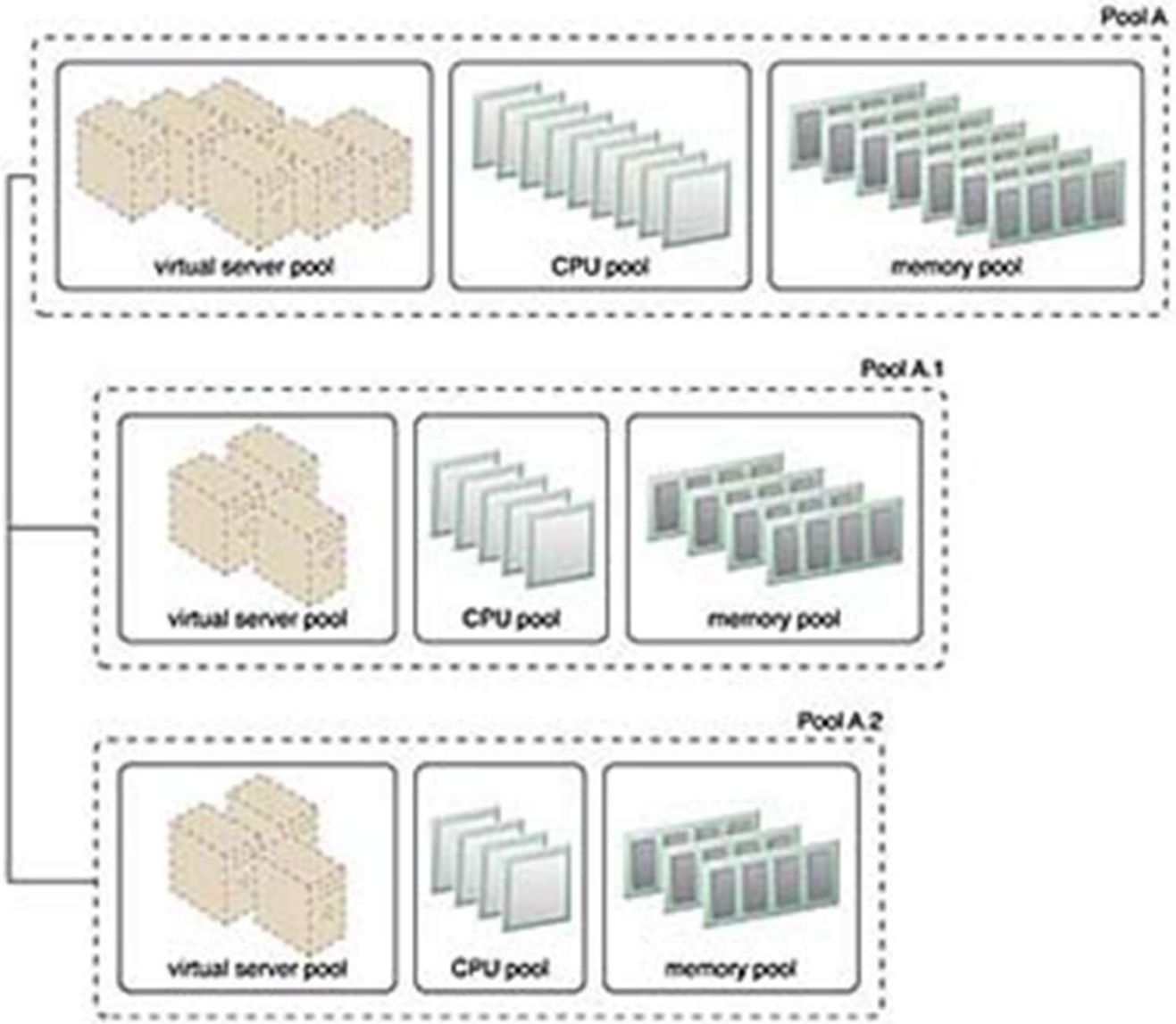
A *resource pooling architecture* is based on the use of one or more resource pools, in which identical IT resources are grouped and maintained by a system that automatically ensures that they remain synchronized.

- 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 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.


CPU pool
memory pool
storage pool
network pool

- 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) 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 are ready to be allocated to virtual servers, and are typically broken down into individual processing cores.
- Pools of physical RAM can be used in newly provisioned physical servers or to vertically scale physical servers.
- 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

- 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.
- 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.
- In the nested pool model, larger pools are divided into smaller pools that individually group the same type of IT resources together. Nested pools can be used to assign resource pools to different departments or groups in the same cloud consumer organization.


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

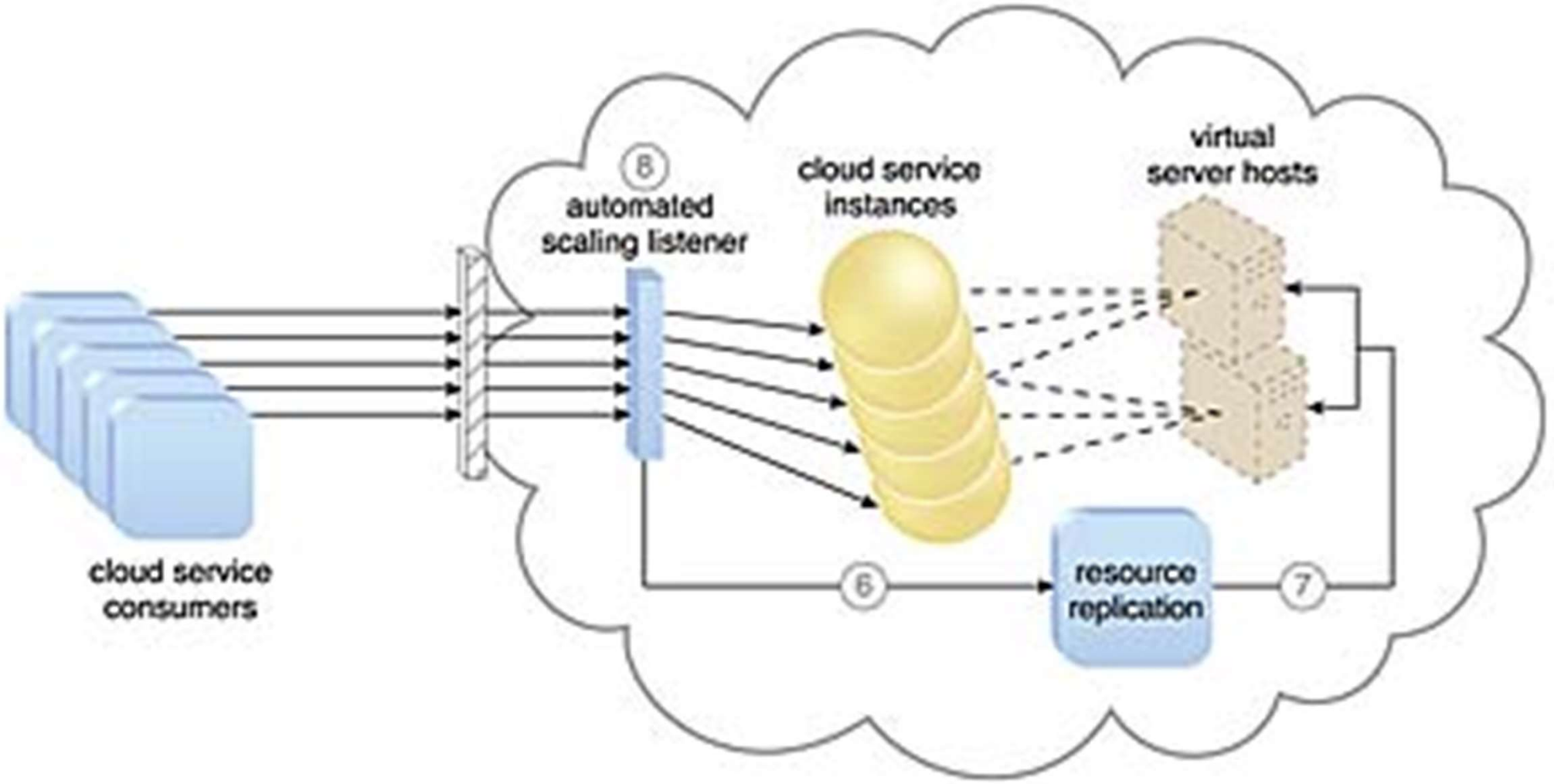
After resources pools have been defined, multiple instances of IT resources from each pool can be created to provide an in-memory pool of "live" IT resources.

In addition to cloud storage devices and virtual servers, which are commonly pooled mechanisms, the following mechanisms can also be part of this cloud architecture:

- *Audit Monitor* – This mechanism monitors resource pool usage to ensure compliance with privacy and regulation requirements, especially when pools contain cloud storage devices or data loaded into memory.
- *Cloud Usage Monitor* – Various cloud usage monitors are involved in the runtime tracking and synchronization that are required by the pooled IT resources and any underlying management systems.
- *Hypervisor* – The hypervisor mechanism is responsible for providing virtual servers with access to resource pools, in addition to hosting the virtual servers and sometimes the resource pools themselves.
- *Logical Network Perimeter* – The logical network perimeter is used to logically organize and isolate resource pools.
- *Pay-Per-Use Monitor* – The pay-per-use monitor collects usage and billing information on how individual cloud consumers are allocated and use IT resources from various pools.
- *Remote Administration System* – This mechanism is commonly used to interface with backend systems and programs in order to provide resource pool administration features via a front-end portal.
- *Resource Management System* – The resource management system mechanism supplies cloud consumers with the tools and permission management options for administering resource pools.
- *Resource Replication* – This mechanism is used to generate new instances of IT resources for resource pools.

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.


8
automated
scaling listener
cloud service
instances
virtual
server hosts
cloud service
consumers
6
resource
replication
7

The following types of dynamic scaling are commonly used:

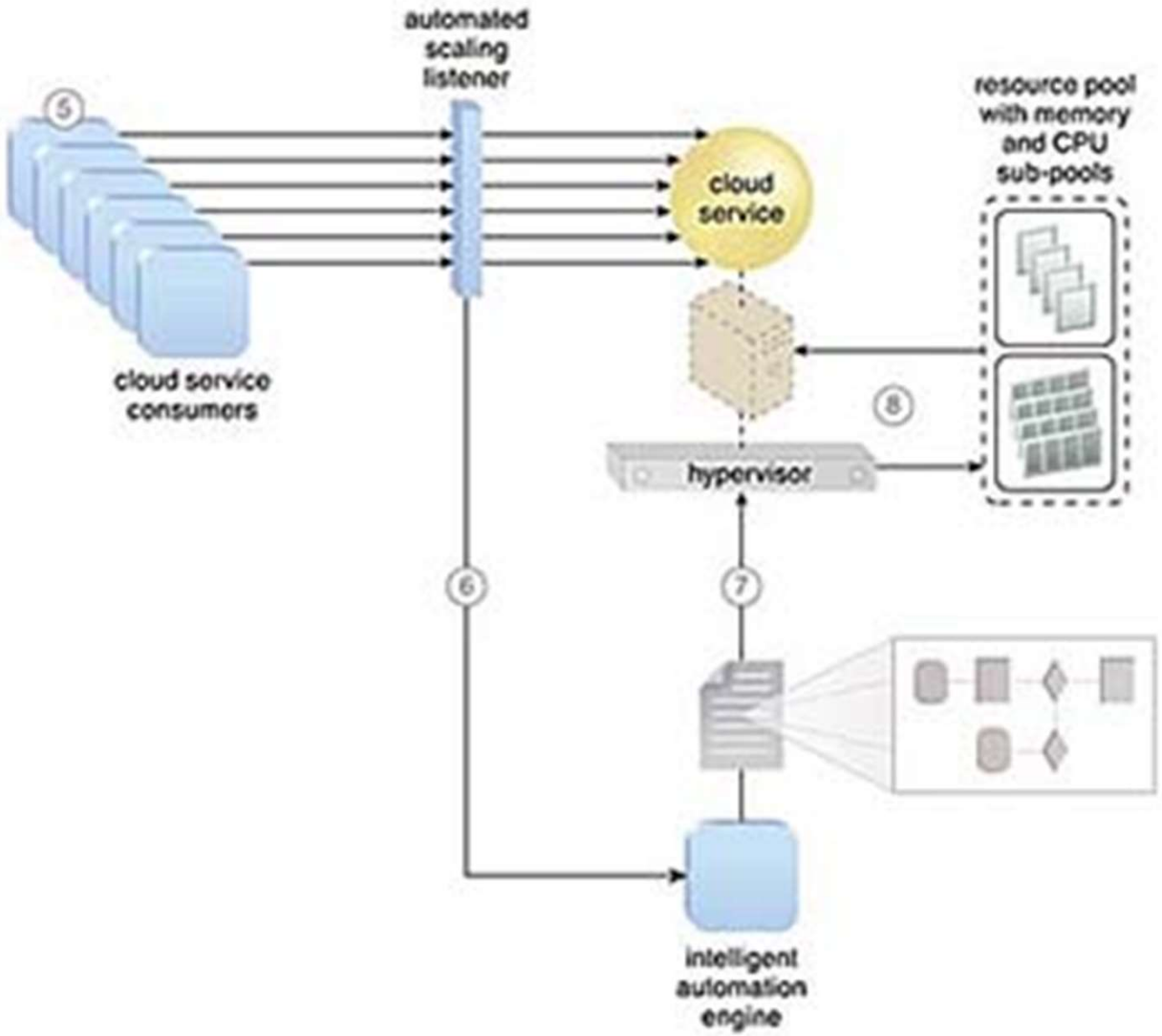
- *Dynamic Horizontal Scaling* – IT resource instances are scaled out and in to handle fluctuating workloads. The automatic scaling listener monitors requests and signals resource replication to initiate IT resource duplication, as per requirements and permissions.
- *Dynamic Vertical Scaling* – IT resource 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.

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.

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


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

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. The runtime processing of the virtual server is monitored so that additional processing power can be leveraged from the resource pool via dynamic allocation, before capacity thresholds are met. The virtual server and its hosted applications and IT resources are vertically scaled in response.

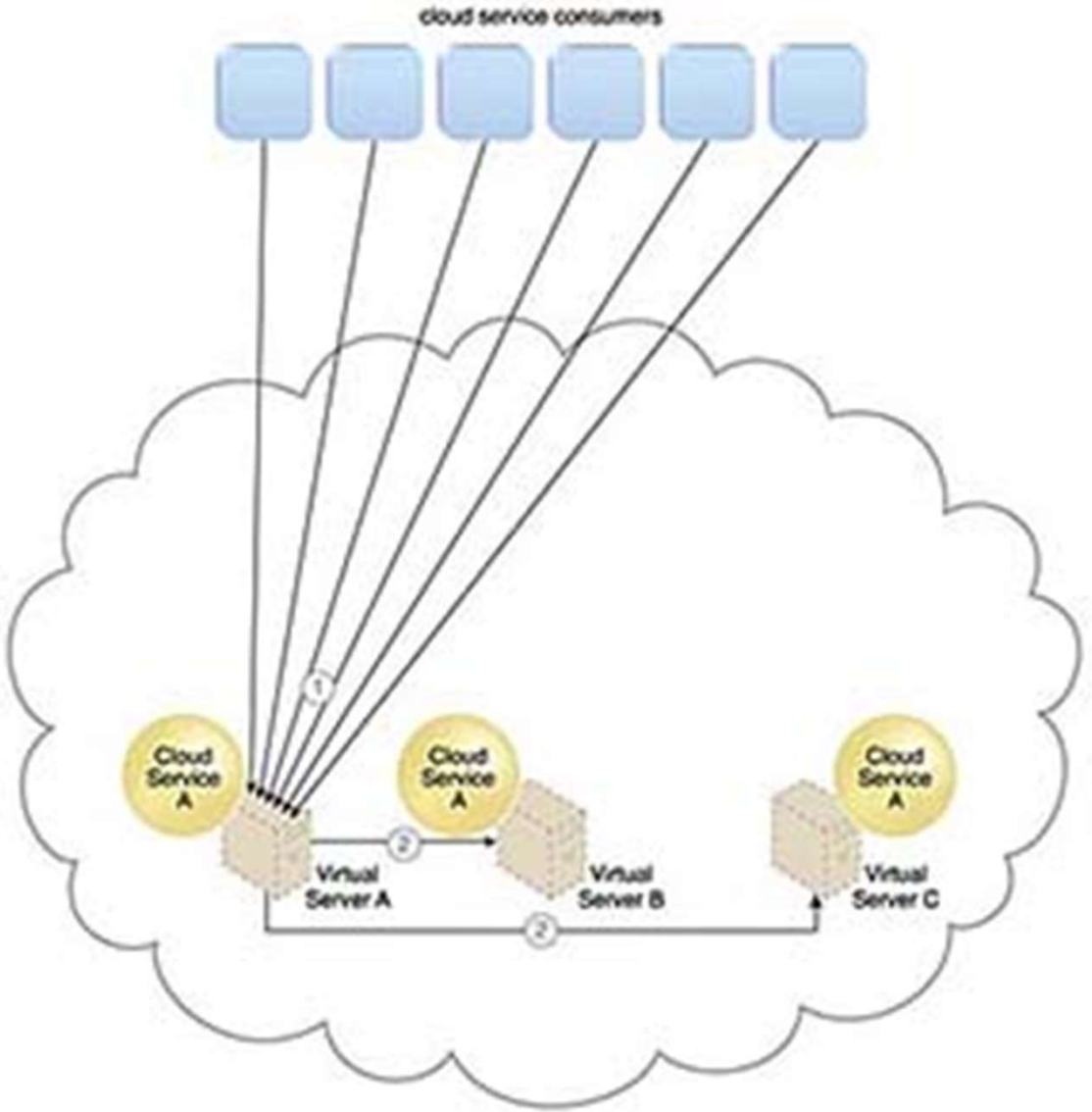
This type of cloud architecture can be designed so that the intelligent automation engine script sends its scaling request via the VIM instead of to the hypervisor directly. Virtual servers that participate in elastic resource allocation systems may require rebooting in order for the dynamic resource allocation to take effect.

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 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 the workloads themselves.
- Depending on the anticipated workload and processing capacity of host server environments, multiple instances of each cloud service implementation can be generated as part of a resource pool that responds to fluctuating request volumes more efficiently.

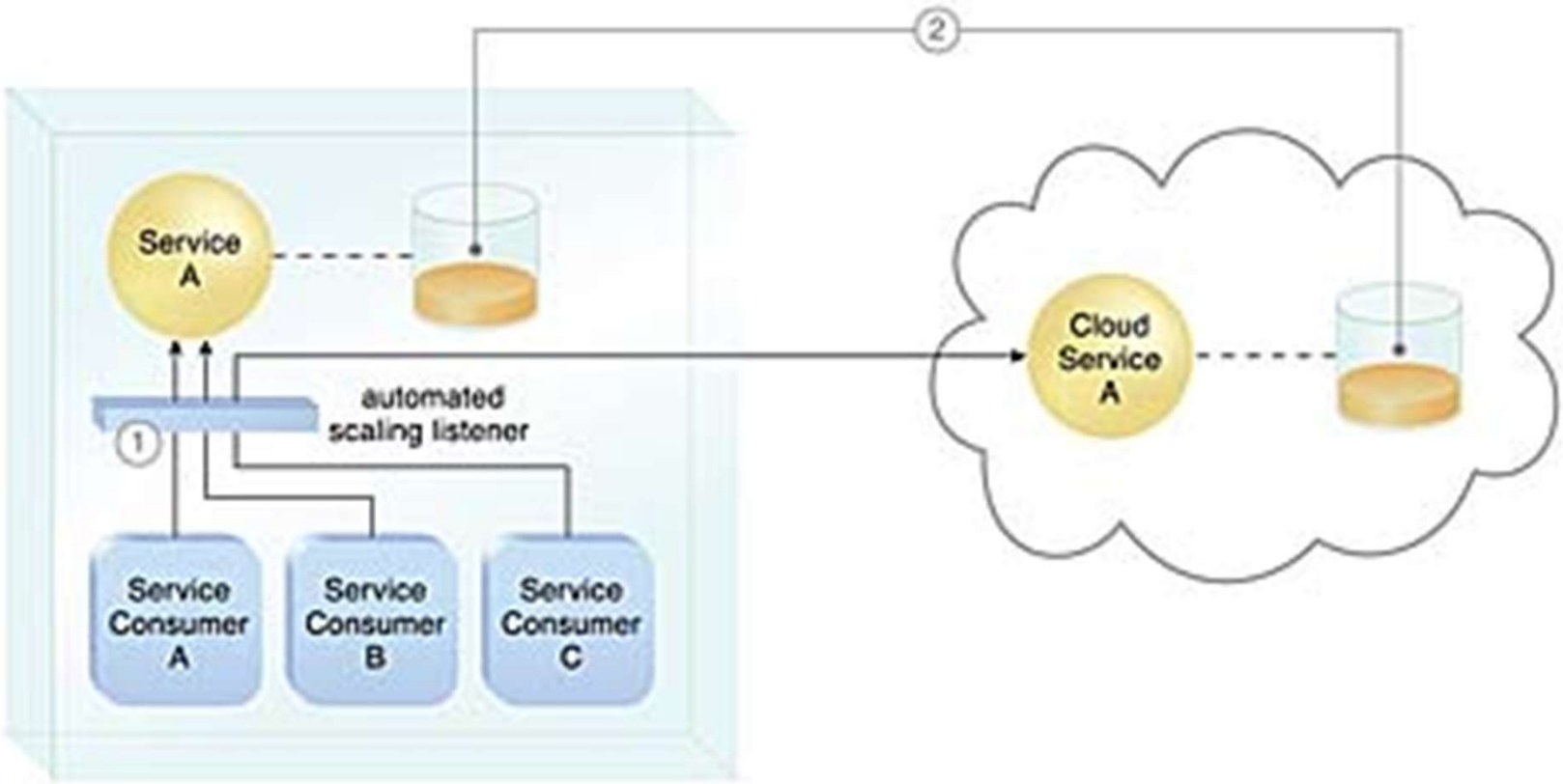

cloud service consumers
Cloud Service A
Cloud Service A
Cloud Service A
Virtual Server A
Virtual Server B
Virtual Server C
1
2
3

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.

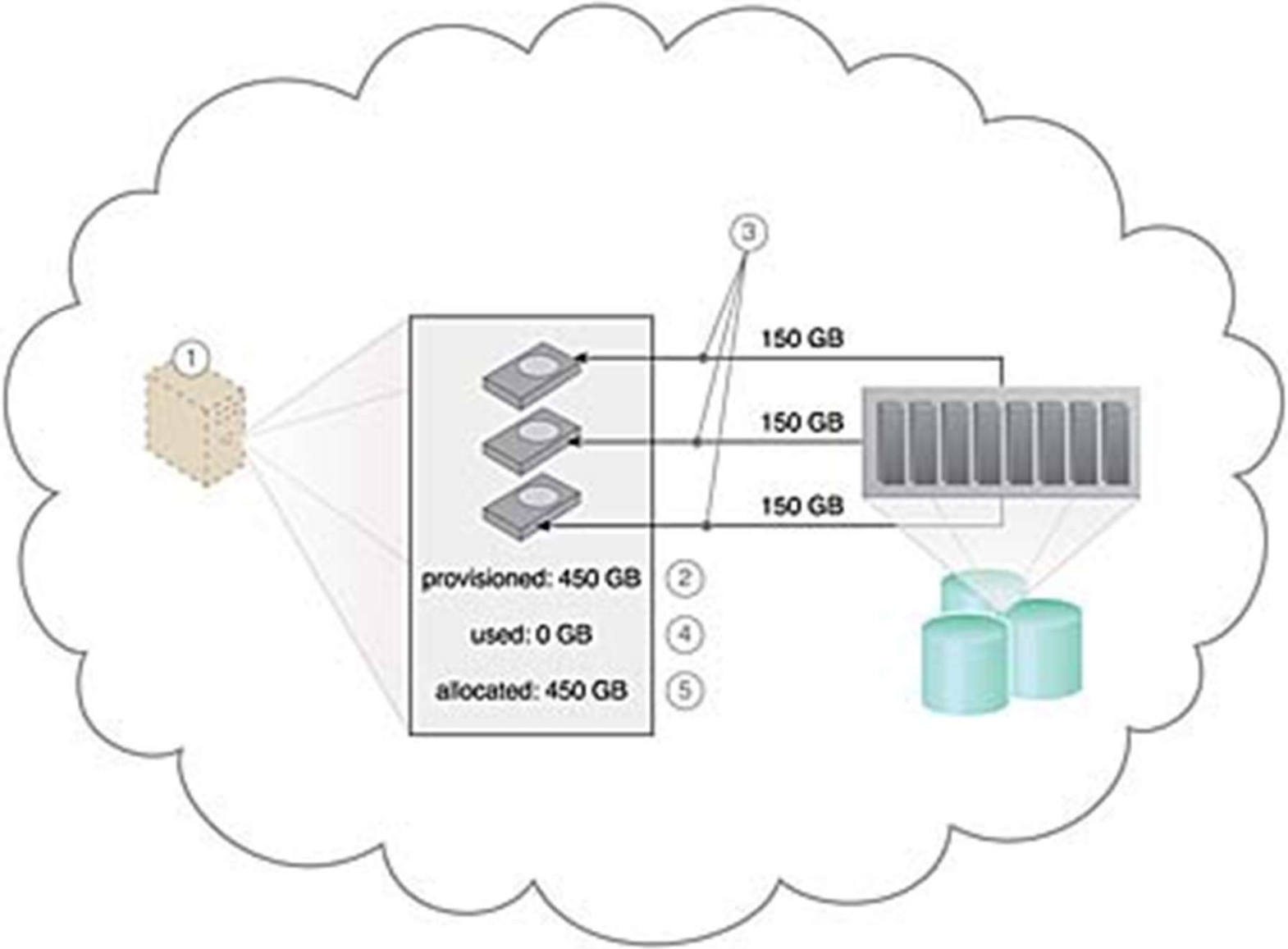
Cloud Bursting Architecture

- 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. After they are no longer required, the cloud-based IT resources are released and the architecture "bursts in" back to the on-premise environment.
- 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 and resource replication mechanisms.
- The automated scaling listener determines when to redirect requests to cloud-based IT resources, and resource replication is used to maintain synchronicity between on-premise and cloud-based IT resources in relation to state information


2
Service A
Cloud Service A
automated scaling listener
1
Service Consumer A
Service Consumer B
Service Consumer C

Elastic Disk Provisioning Architecture

- Cloud consumers are commonly charged for cloud-based storage space based on fixed-disk storage allocation, meaning the charges are predetermined by disk capacity and not aligned with actual data storage consumption.
- The cloud consumer is billed for using 450 GB of storage space after installing the operating system, even though the operating system only requires 15 GB of storage space.

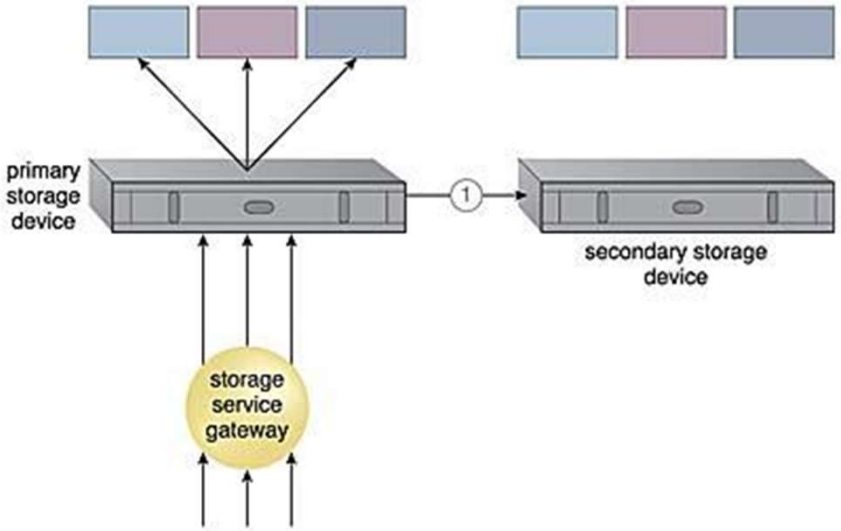

1
2
3
4
5
150 GB
150 GB
150 GB
provisioned: 450 GB
used: 0 GB
allocated: 450 GB

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

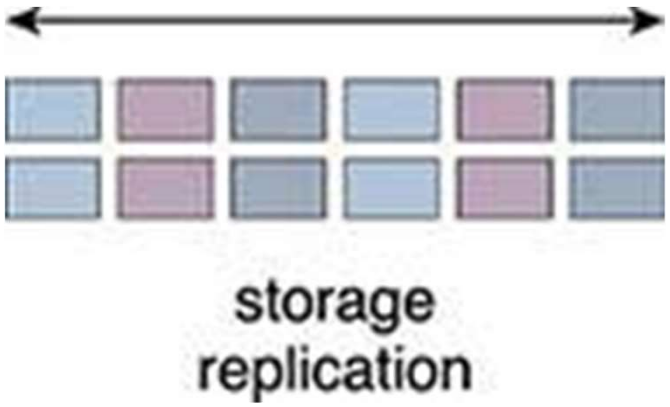
- *Cloud Usage Monitor* – Specialized cloud usage monitors can be used to track and log storage usage fluctuations.
- *Resource Replication* – Resource replication is part of an elastic disk provisioning system when conversion of dynamic thin-disk storage into static thick-disk storage is required.

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.



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



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.

Cloud providers may locate secondary cloud storage devices in a different geographical region than the primary cloud storage device, usually for economic reasons. However, this can introduce legal concerns for some types of data. The location of the secondary cloud storage devices can dictate the protocol and method used for synchronization, as some replication transport protocols have distance restrictions.

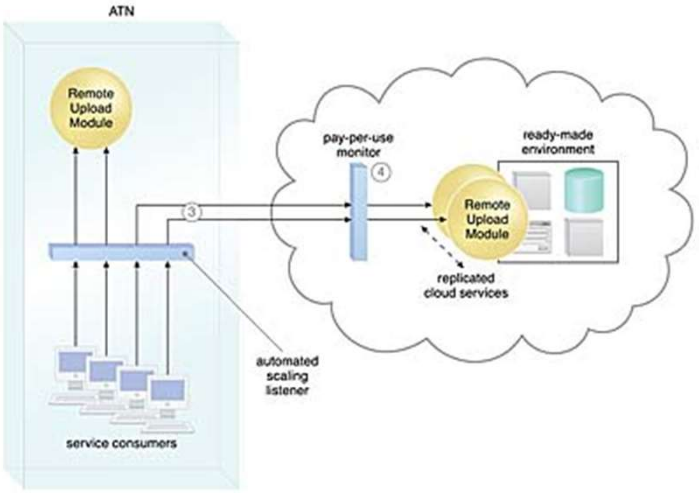
Some cloud providers use storage devices with dual array and storage controllers to improve device redundancy, and place secondary storage devices in a different physical location for cloud balancing and disaster recovery purposes. In this case, cloud providers may need to lease a network connection via a third-party cloud provider in order to establish the replication between the two devices.

Case Study Example

An in-house solution that ATN did not migrate to the cloud is the Remote Upload Module, a program that is used by their clients to upload accounting and legal documents to a central archive on a daily basis. Usage peaks occur without warning, since the quantity of documents received on a day-by-day basis is unpredictable.

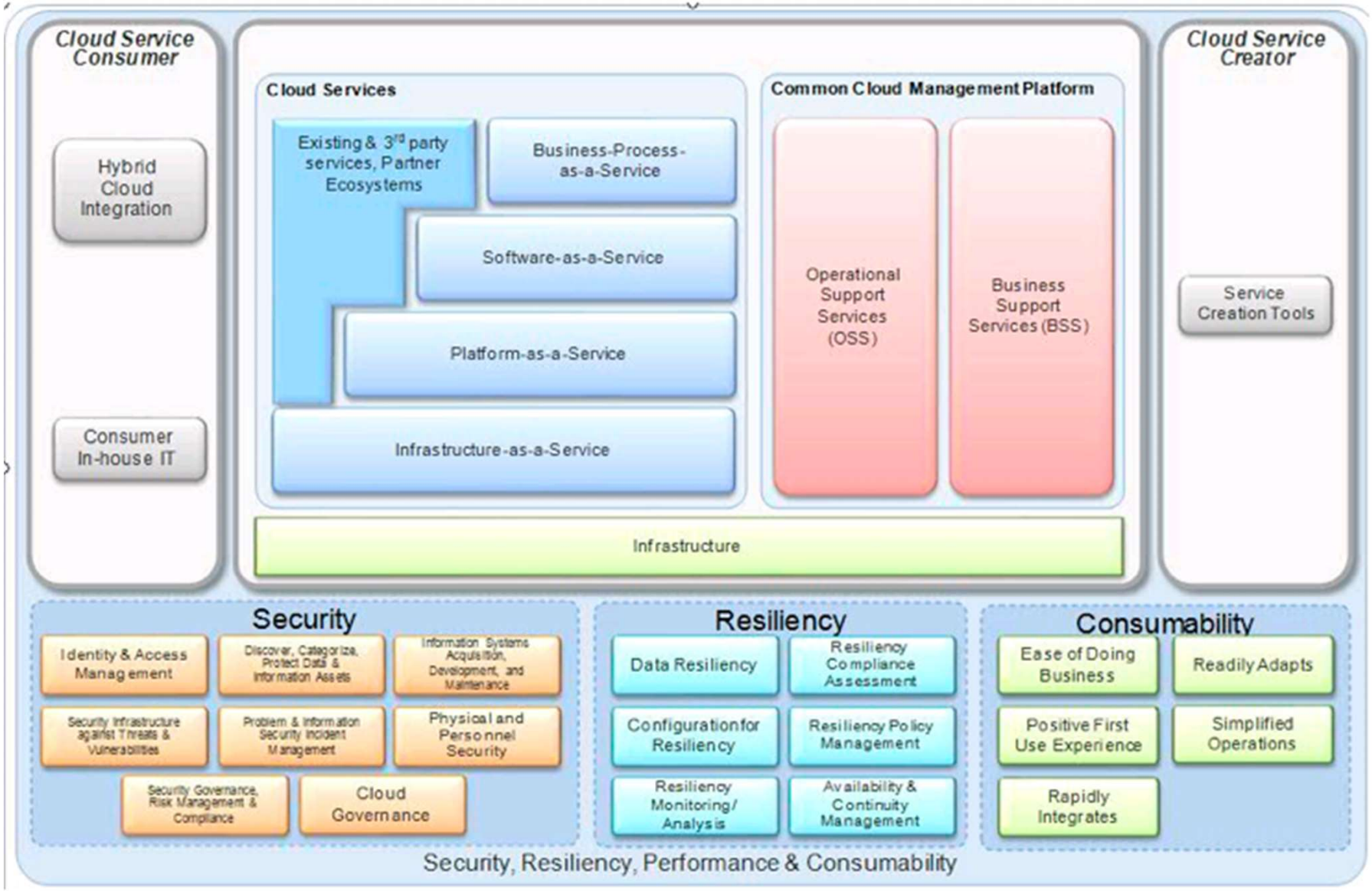
The Remote Upload Module currently rejects upload attempts when it is operating at capacity, which is problematic for users that need to archive certain documents before the end of a business day or prior to a deadline.

ATN decides to take advantage of its cloud-based environment by creating a cloud-bursting architecture around the on-premise Remote Upload Module service implementation. This enables it to burst out into the cloud whenever on-premise processing thresholds are exceeded.



Cloud Computing Reference Architecture (CCRA)

- IBM constantly refines the Cloud Computing Reference Architecture (CCRA) based on the changing regulatory and compliance needs (based on the solid security and privacy frameworks). The CCRA is intended to be used as a blueprint for architecting cloud implementations, driven by functional and non-functional requirements of the respective cloud implementation.
- The CCRA defines the basic building blocks—architectural elements and their relationships—which make up the cloud. IBM Federal Cloud Computing Reference Architecture considers key cloud requirements like FedRAMP and Sec-508 augmented by PaaS innovations. IBM products like Bluemix enable rapid application development in the cloud via DevOps. This improves the time to capability and reduces the overall IT costs associated with private, public and hybrid cloud models.


Cloud Service Consumer
Hybrid Cloud Integration
Consumer In-house IT
Cloud Services
Existing & 3rd party services, Partner Ecosystems
Business-Process-as-a-Service
Software-as-a-Service
Platform-as-a-Service
Infrastructure-as-a-Service
Common Cloud Management Platform
Operational Support Services (OSS)
Business Support Services (BSS)
Infrastructure
Cloud Service Creator
Service Creation Tools
Security
Identity & Access Management
Discover, Categorize, Protect Data & Information Assets
Information Systems Acquisition, Development, and Maintenance
Security Infrastructure against Threats & Vulnerabilities
Problem & Information Security Incident Management
Physical and Personnel Security
Security Governance, Risk Management & Compliance
Cloud Governance
Resiliency
Data Resiliency
Resiliency Compliance Assessment
Configuration for Resiliency
Resiliency Policy Management
Resiliency Monitoring/ Analysis
Availability & Continuity Management
Consumability
Ease of Doing Business
Readily Adapts
Positive First Use Experience
Simplified Operations
Rapidly Integrates
Security, Resiliency, Performance & Consumability

Benefits of CCRA

Using a reference architecture **allows the team to deliver a solution quicker, and with fewer errors**. Re-using an architecture provides advantages such as quicker delivery of a solution, reduced design effort, reduced costs, and increased quality.

Migration Risks and Mitigation

1. Intricate Architecture

Cloud migrations approaches often fail due to the complex architecture. Data-rich applications are also dependent on multiple elements and environments.

Cloud Migration Strategy: Manage only one complex architecture that is currently existing on-premise. Design architecture in such a way that it consumes data stored in the enterprise's IT environment.

2. Multiple Dependencies

Multiple dependencies with on-premises environments create problems during lift and shift.

Cloud Mitigation Strategy: The best bet is to consider solutions that test before migration. They can identify and remediate the differences in environments. Seek the services of a cloud services provider who offers services that are relevant to your needs.

3. Data Gravity

Data Gravity becomes difficult to test if an application and its data are not working as it should in the cloud. Most replication-based migration tools require data to be moved before the apps due to improper sequencing problems.

Cloud Migration Strategy: Use live cloud migration approaches and tools that stream the whole instance. Live migration eliminates the need for complex system synchronization and avoids consistency issues.

4. Management And Control Of Data Streams Within Heterogeneous Environments

Databases that require a consistent view create unpredictable issues. Also, transactional production servers that continuously generate data are hard to manage. After data migration, the system must track and synchronize new changes to the production application. Furthermore, there may be security concerns with storing production data in the public cloud. It leads to a lack of control over multiple data repositories across a hybrid IT landscape.

5. Cloud Gravity: IT teams require workload mobility for effective data and workload migration. They must ensure these factors do not affect the business or introduce hidden costs.

Cloud Migration Strategy: In the case of data-rich enterprise applications, evaluate migration solutions for speed and simplicity. Enable portability and interoperability of stateless applications in a multi-cloud strategy, using containers.

6. Latency

Applications can face latency issues when using cloud applications over the Internet.

Cloud Migration Strategy: Use optimization services from a cloud service provider to help tide over latency issues.

7. Architectural difference

It’s common to require modifications for your application design and architecture. But they may not be conducive for distributed cloud environments.

Cloud Migration Strategy: Implement a piece-by-piece evaluation and move only pertinent features.

Choosing a cloud service provider may seem like a tight-rope walk in the decision-making stage. Study your desired architecture thoroughly and choose a cloud resource that works for you. Does it scale for your enterprise, help manage fluctuations, and support the migration of critical applications? It’s essential to accomplish this without adding complexity or compromising data.

Assignment 5

- Evaluate Workload Distribution Architecture with suitable example.
- Evaluate Resource Pooling Architecture with suitable example.
- Evaluate Dynamic Scalability Architecture with suitable example.
- Evaluate Elastic Resource Capacity Architecture with suitable example.
- Evaluate Service Load Balancing Architecture with suitable example.
- Evaluate Cloud Bursting Architecture with suitable example.
- Evaluate Elastic Disk Provisioning Architecture with suitable example.
- Evaluate Redundant Storage Architecture with suitable example.
- Evaluate Cloud Computing Reference Architecture (CCRA) with suitable example.

Submission link: https://forms.gle/f8vGUapeDMowSL7n9

Presentation V (Choose any one topic)

- Case study on Cloud Architectures
- Case study on Cloud Computing Reference Architecture (CCRA)

Submission link: https://forms.gle/f8vGUapeDMowSL7n9

Lab 5

To install one node Hadoop cluster on local machine. .

Submission link: https://forms.gle/f8vGUapeDMowSL7n9