Chapter 13

Specialized Cloud Architectures

The architectural models that are covered in this chapter span a broad range of functional areas and topics to offer creative combinations of mechanisms and specialized components.

13.1 Direct I/O Access Architecture

Access to the physical I/O cards that are installed on a physical server is usually provided to hosted virtual servers via a hypervisor-based layer of processing called I/O virtualization. However, virtual servers sometimes need to connect to and use I/O cards without any hypervisor interaction or emulation.

With the *direct I/O access architecture*, virtual servers are allowed to circumvent the hypervisor and directly access the physical server's I/O card as an alternative to emulating a connection via the hypervisor (Figures 13.1 to 13.3).

To achieve this solution and access the physical I/O card without hypervisor interaction, the host CPU needs to support this type of access with the appropriate drivers installed on the virtual server. The virtual server can then recognize the I/O card as a hardware device after the drivers are installed.

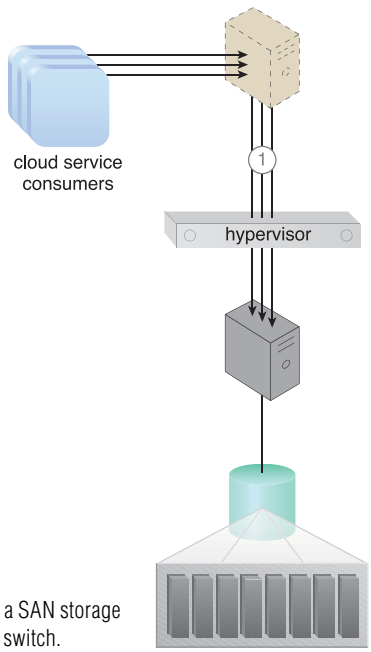


Figure 13.1
Cloud service consumers access a virtual server, which accesses a database on a SAN storage LUN (1). Connectivity from the virtual server to the database occurs via a virtual switch.

Figure 13.2
There is an increase in the amount of cloud service consumer requests (2), causing the bandwidth and performance of the virtual switch to become inadequate (3).

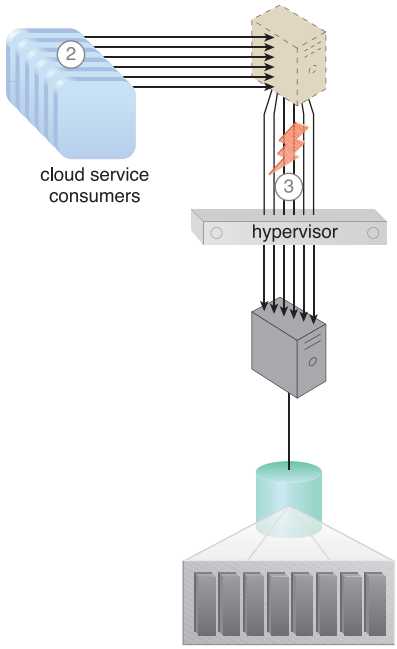
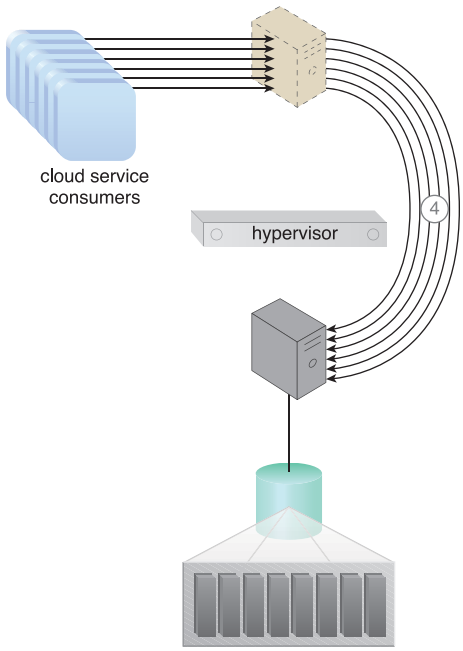


Figure 13.3
The virtual server bypasses the hypervisor to connect to the database server via a direct physical link to the physical server (4). The increased workload can now be properly handled.



Other mechanisms that can be involved in this architecture in addition to the virtual server and hypervisor include:

- *Cloud Usage Monitor* – The cloud service usage data that is collected by runtime monitors can include and separately classify direct I/O access.
- *Logical Network Perimeter* – The logical network perimeter ensures that the allocated physical I/O card does not allow cloud consumers to access other cloud consumers' IT resources.
- *Pay-Per-Use Monitor* – This monitor collects usage cost information for the allocated physical I/O card.
- *Resource Replication* – Replication technology is used to replace virtual I/O cards with physical I/O cards.

13.2 Direct LUN Access Architecture

Storage LUNs are typically mapped via a host bus adapter (HBA) on the hypervisor, with the storage space emulated as file-based storage to virtual servers (Figure 13.4). However, virtual servers sometimes need direct access to RAW block-based storage. For example, access via an emulated adapter is insufficient when a cluster is implemented and a LUN is used as the shared cluster storage device between two virtual servers.

The *direct LUN access architecture* provides virtual servers with LUN access via a physical HBA card, which is effective because virtual servers in the same cluster can use the LUN as a shared volume for clustered databases. After implementing this solution, the virtual servers' physical connectivity to the LUN and cloud storage device is enabled by the physical hosts.

The LUNs are created and configured on the cloud storage device for LUN presentation to the hypervisors. The cloud storage device needs to be configured using raw device mapping to make the LUNs visible to the virtual servers as a block-based RAW SAN LUN, which is unformatted, un-partitioned storage. The LUN needs to be represented with a unique LUN ID to be used by all of the virtual servers as shared storage. Figures 13.5 and 13.6 illustrate how virtual servers are given direct access to block-based storage LUNs.

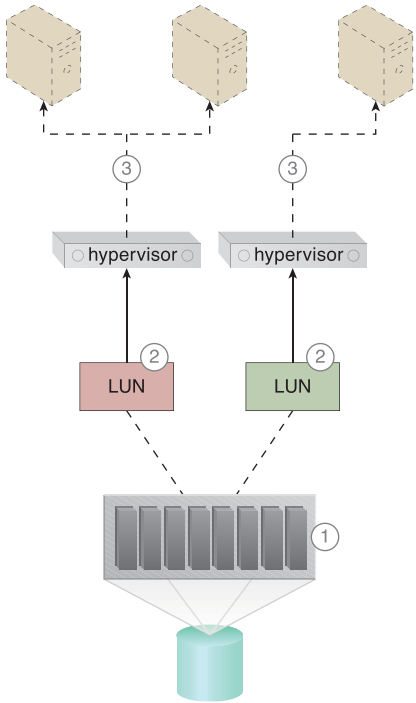


Figure 13.4
The cloud storage device is installed and configured (1). The LUN mapping is defined so that each hypervisor has access to its own LUN and can also see all of the mapped LUNs (2). The hypervisor shows the mapped LUNs to the virtual servers as normal file-based storage to be used as such (3).

Besides the virtual server, hypervisor, and cloud storage device, the following mechanisms can be incorporated into this architecture:

- *Cloud Usage Monitor* – This monitor tracks and collects storage usage information that pertains to the direct usage of LUNs.
- *Pay-Per-Use Monitor* – The pay-per-use monitor collects and separately classifies usage cost information for direct LUN access.
- *Resource Replication* – This mechanism relates to how virtual servers directly access block-based storage in replacement of file-based storage.

Figure 13.5

The cloud storage device is installed and configured (1). The required LUNs are created and presented to the hypervisors (2), which map the presented LUNs directly to the virtual servers (3). The virtual servers can see the LUNs as RAW block-based storage and can access them directly (4).

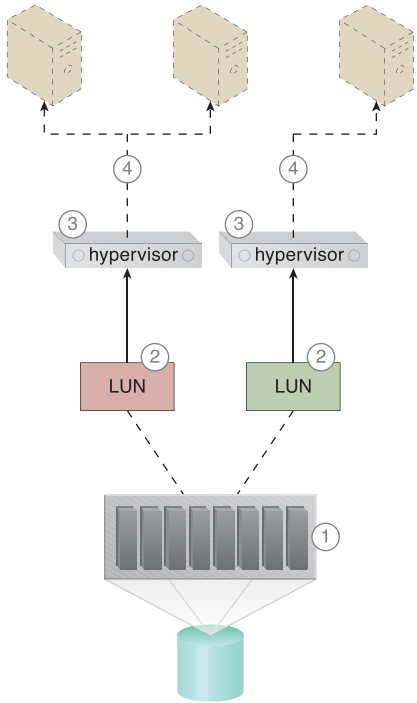
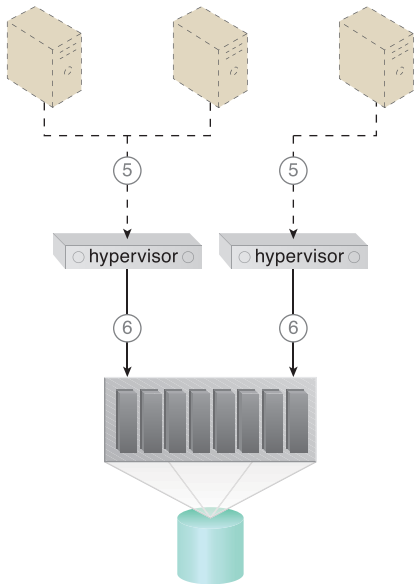


Figure 13.6

The virtual servers' storage commands are received by the hypervisors (5), which process and forward the requests to the storage processor (6).



13.3 Dynamic Data Normalization Architecture

Redundant data can cause a range of issues in cloud-based environments, such as:

- increased time required to store and catalog files
- increased required storage and backup space
- increased costs due to increased data volume
- increased time required for replication to secondary storage
- increased time required to backup data

For example, if a cloud consumer copies 100 MB of files onto a cloud storage device and the data is redundantly copied ten times, the consequences can be considerable:

- The cloud consumer will be charged for using 10 x 100 MB of storage space, even though only 100 MB of unique data was actually stored.
- The cloud provider needs to provide an unnecessary 900 MB of space in the online cloud storage device and any backup storage systems.
- Significantly more time is required to store and catalog data.
- Data replication duration and performance are unnecessarily taxed whenever the cloud provider performs a site recovery, since 1,000 MB need to be replicated instead of 100 MB.

These impacts can be significantly amplified in multitenant public clouds.

The *dynamic data normalization architecture* establishes a de-duplication system, which prevents cloud consumers from inadvertently saving redundant copies of data by detecting and eliminating redundant data on cloud storage devices. This system can be applied to both block and file-based storage devices, although it is most effective on the former. This de-duplication system checks each received block to determine whether it is redundant with a block that has already been received. Redundant blocks are replaced with pointers to the equivalent blocks that are already in storage (Figure 13.7).

The de-duplication system examines received data prior to passing it to storage controllers. As part of the examination process, a hash code is assigned to every piece of data that has been processed and stored. An index of hashes and pieces is also maintained. As a result, the generated hash of a newly received block of data is compared with the hashes in storage to determine whether it is a new or duplicate data block. New blocks

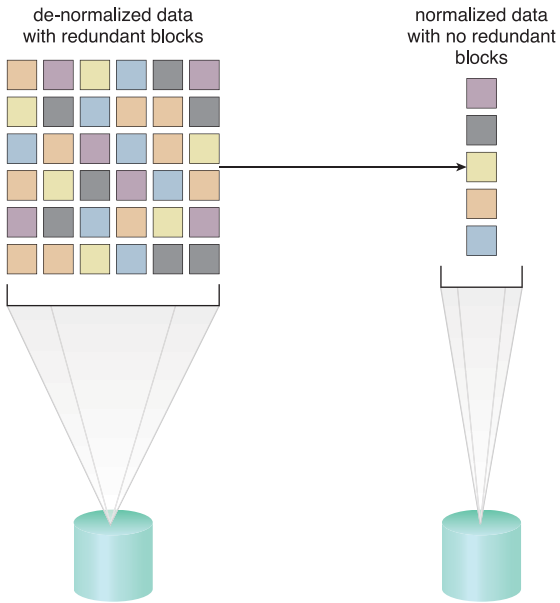


Figure 13.7
Data sets containing redundant data are unnecessarily bloating storage (left). The data de-duplication system normalizes the data, so that only unique data is stored (right).

are saved, while duplicate data is eliminated and a pointer to the original data block is created and saved instead.

This architectural model can be used for both disk storage and backup tape drives. One cloud provider can decide to prevent redundant data only on backup cloud storage devices, while another can more aggressively implement the data de-duplication system on all of its cloud storage devices. There are different methods and algorithms for comparing blocks of data to confirm their duplicity with other blocks.

13.4 Elastic Network Capacity Architecture

Even if IT resources are scaled on-demand by a cloud platform, performance and scalability can still be inhibited when remote access to the IT resources is impacted by network bandwidth limitations (Figure 13.8).

The *elastic network capacity architecture* establishes a system in which additional bandwidth is allocated dynamically to the network to avoid runtime bottlenecks. This

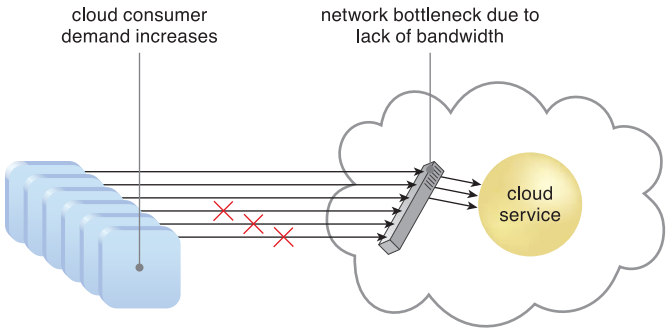


Figure 13.8
A lack of available bandwidth causes performance issues for cloud consumer requests.

system ensures that each cloud consumer is using a different set of network ports to isolate individual cloud consumer traffic flows.

The automated scaling listener and intelligent automation engine scripts are used to detect when traffic reaches the bandwidth threshold, and to dynamically allocate additional bandwidth and/or network ports when required.

The cloud architecture can be equipped with a network resource pool containing network ports that are made available for shared usage. The automated scaling listener monitors workload and network traffic, and signals the intelligent automation engine to modify the number of allocated network ports and/or bandwidth in response to usage fluctuations.

Note that when this architectural model is implemented at the virtual switch level, the intelligent automation engine may need to run a separate script that adds physical uplinks to the virtual switch specifically. Alternatively, the direct I/O access architecture can also be incorporated to increase network bandwidth that is allocated to the virtual server.

In addition to the automated scaling listener, the following mechanisms can be part of this architecture:

- *Cloud Usage Monitor* – These monitors are responsible for tracking elastic network capacity before, during, and after scaling.
- *Hypervisor* – The hypervisor provides virtual servers with access to the physical network, via virtual switches and physical uplinks.

- *Logical Network Perimeter* – This mechanism establishes the boundaries that are needed to provide individual cloud consumers with their allocated network capacity.
- *Pay-Per-Use Monitor* – This monitor keeps track of any billing-related data that pertains to dynamic network bandwidth consumption.
- *Resource Replication* – Resource replication is used to add network ports to physical and virtual servers, in response to workload demands.
- *Virtual Server* – Virtual servers host the IT resources and cloud services to which network resources are allocated and are themselves affected by the scaling of network capacity.

13.5 Cross-Storage Device Vertical Tiering Architecture

Cloud storage devices are sometimes unable to accommodate the performance requirements of cloud consumers, and have more data processing power or bandwidth added to increase IOPS. These conventional methods of vertical scaling are usually inefficient and time-consuming to implement, and can become wasteful when the increased capacity is no longer required.

The scenario in Figures 13.9 and 13.10 depicts an approach in which a number of requests for access to a LUN has increased, requiring its manual transfer to a high-performance cloud storage device.

The *cross-storage device vertical tiering architecture* establishes a system that survives bandwidth and data processing power constraints by vertically scaling between storage devices that have different capacities. LUNs can automatically scale up and down across multiple devices in this system so that requests can use the appropriate storage device level to perform cloud consumer tasks.

New cloud storage devices with increased capacity can also be made available, even if the automated tiering technology can move data to cloud storage devices with the same storage processing capacity. For example, solid-state drives (SSDs) can be suitable devices for data processing power upgrades.

Figure 13.9
A cloud provider installs and configures a cloud storage device (1) and creates LUNs that are made available to the cloud service consumers for usage (2). The cloud service consumers initiate data access requests to the cloud storage device (3), which forwards the requests to one of the LUNs (4).

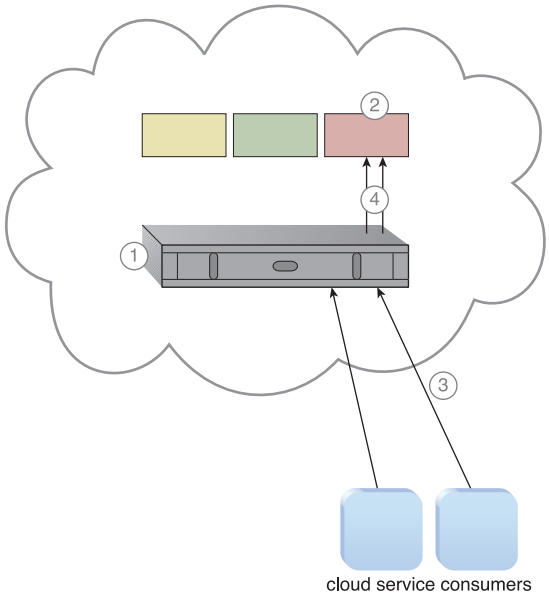
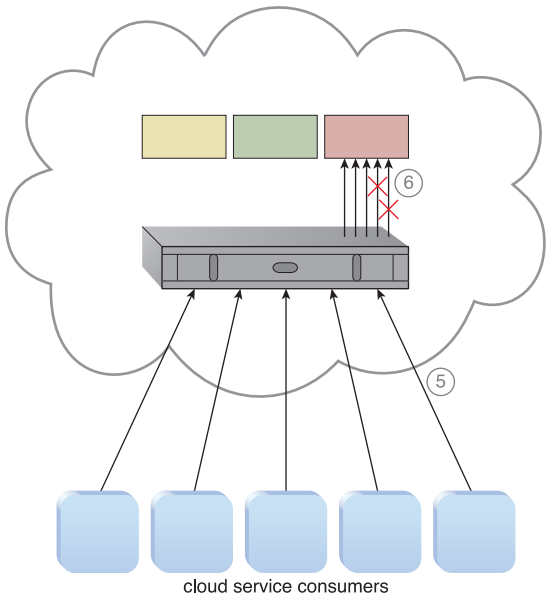


Figure 13.10
The number of requests increases, resulting in high storage bandwidth and performance demands (5). Some of the requests are rejected, or time out due to performance capacity limitations within the cloud storage device (6).



The automated scaling listener monitors the requests that are sent to specific LUNs, and signals the storage management program to move the LUN to a higher capacity device once it identifies a predefined threshold has been reached. Service interruption is prevented because there is never a disconnection during the transfer. The original device remains up and running, while the LUN data scales up to another device. Cloud consumer requests are automatically redirected to the new cloud storage device as soon as the scaling is completed (Figures 13.11 to 13.13).

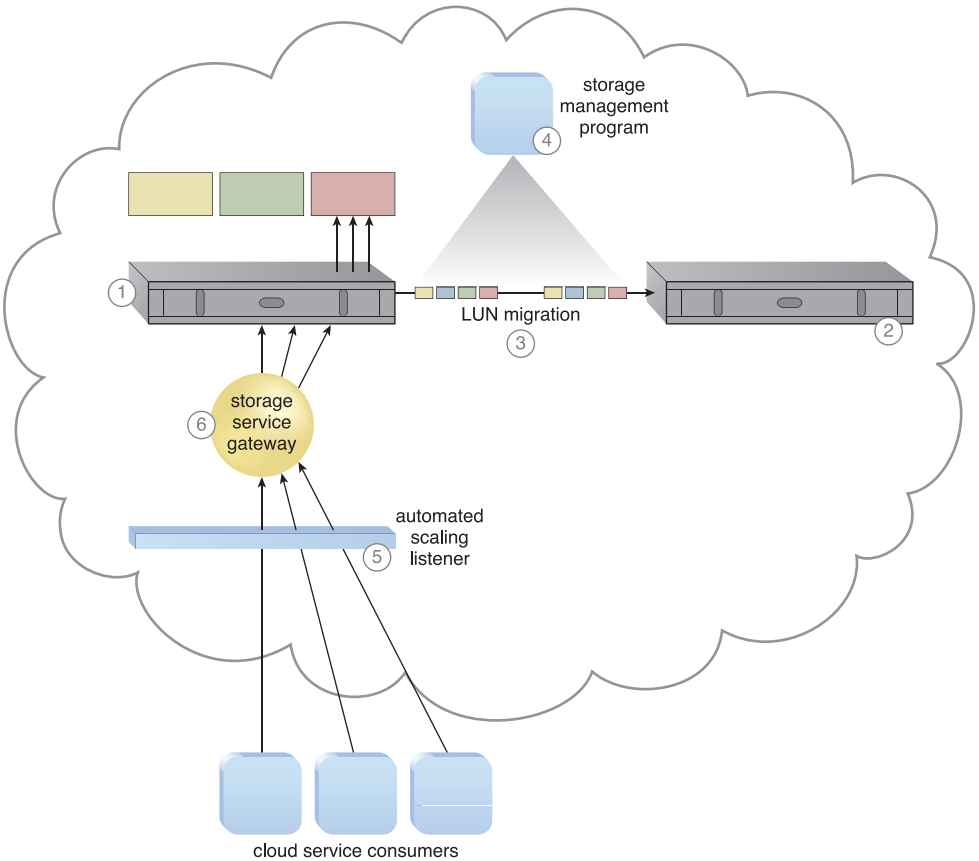


Figure 13.11

The lower capacity primary cloud storage device is responding to cloud service consumer storage requests (1). A secondary cloud storage device with higher capacity and performance is installed (2). The LUN migration (3) is configured via the storage management program that is configured to categorize the storage based on device performance (4). Thresholds are defined in the automated scaling listener that is monitoring the requests (5). Cloud service consumer requests are received by the storage service gateway and sent to the primary cloud storage device (6).

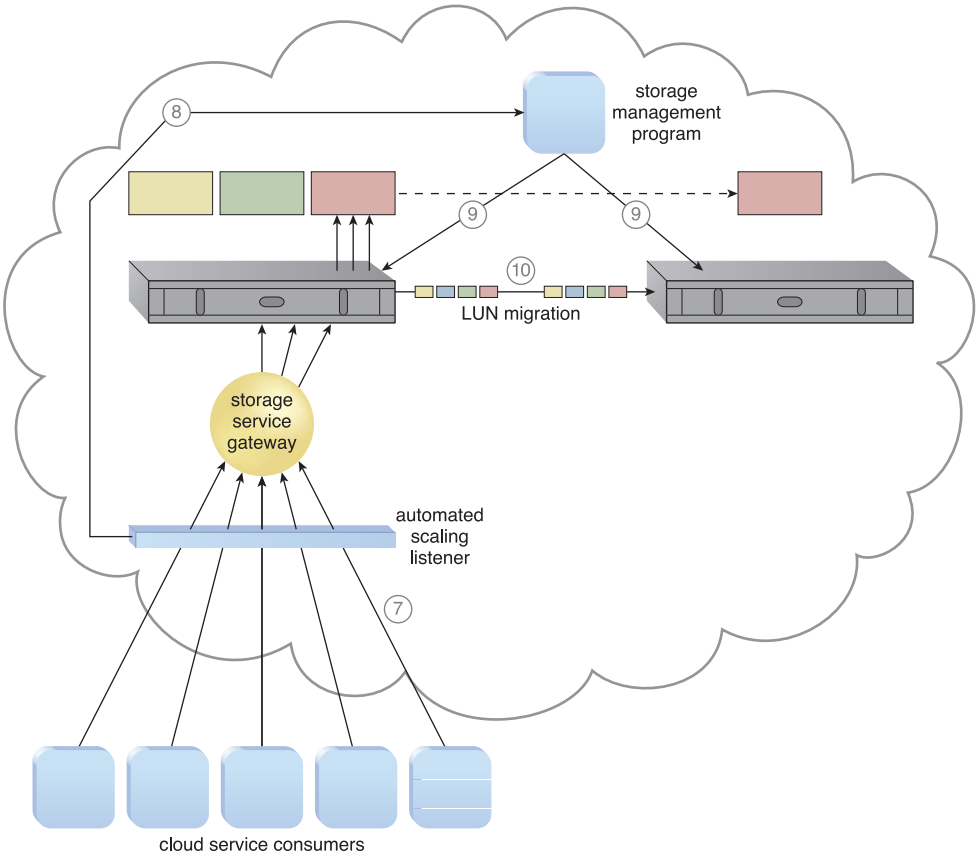


Figure 13.12

The number of cloud service consumer requests reaches the predefined threshold (7), and the automated scaling listener notifies the storage management program that scaling is required (8). The storage management program calls LUN migration to move the cloud consumer's LUN to the secondary, higher capacity storage device (9) and the LUN migration performs this move (10).

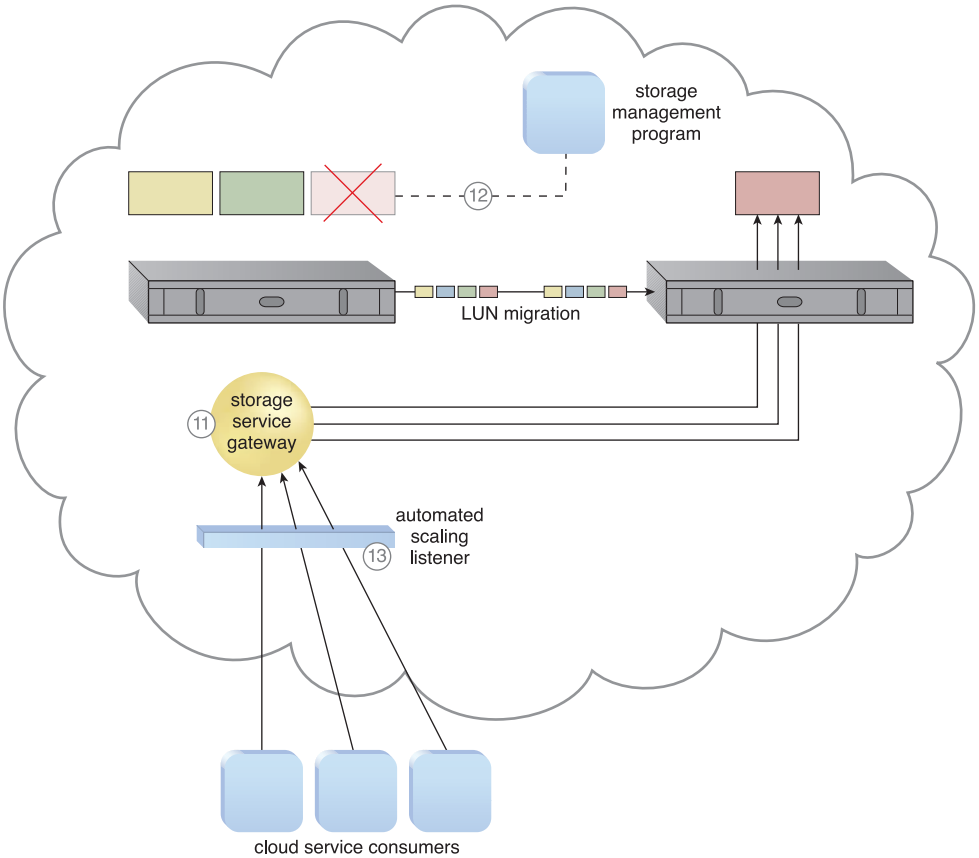


Figure 13.13

The storage service gateway forwards the cloud service consumer requests from the LUN to the new cloud storage device (11). The original LUN is deleted from the lower capacity device via the storage management program and LUN migration (12). The automated scaling listener monitors cloud service consumer requests to ensure that the request volume continues to require access to the higher capacity secondary storage for the migrated LUN (13).

In addition to the automated scaling listener and cloud storage device, the mechanisms that can be incorporated in this technology architecture include:

- *Audit Monitor* – The auditing performed by this monitor checks whether the relocation of cloud consumer data does not conflict with any legal or data privacy regulations or policies.

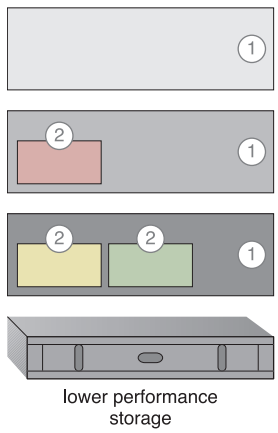
- *Cloud Usage Monitor* – This infrastructure mechanism represents various runtime monitoring requirements for tracking and recording data transfer and usage, at both source and destination storage locations.
- *Pay-Per-Use Monitor* – Within the context of this architecture, the pay-per-use monitor collects storage usage information on both source and destination locations, as well as IT resource usage information for carrying out cross-storage tiering functionality.

13.6 Intra-Storage Device Vertical Data Tiering Architecture

Some cloud consumers may have distinct data storage requirements that restrict the data's physical location to a single cloud storage device. Distribution across other cloud storage devices may be disallowed due to security, privacy, or various legal reasons. This type of limitation can impose severe scalability limitations upon the device's storage and performance capacity. These limitations can further cascade to any cloud services or applications that are dependent upon the use of the cloud storage device.

The *intra-storage device vertical data tiering architecture* establishes a system to support vertical scaling within a single cloud storage device. This intra-device scaling system optimizes the availability of different disk types with different capacities (Figure 13.14).

Figure 13.14
The cloud intra-storage device system vertically scales through disk types graded into different tiers (1). Each LUN is moved to a tier that corresponds to its processing and storage requirements (2).



This cloud storage architecture requires the use of a complex storage device that supports different types of hard disks, especially high-performance disks like SATAs, SASs, and SSDs. The disk types are organized into graded tiers so that LUN migration can vertically scale the device based on the allocation of disk types, which align with the processing and capacity requirements.

Data load conditions and definitions are set after disk categorization so that the LUNs can move to either a higher or lower grade, depending on which predefined conditions are met. These thresholds and conditions are used by the automated scaling listener when monitoring runtime data processing traffic (Figures 13.15 to 13.17).

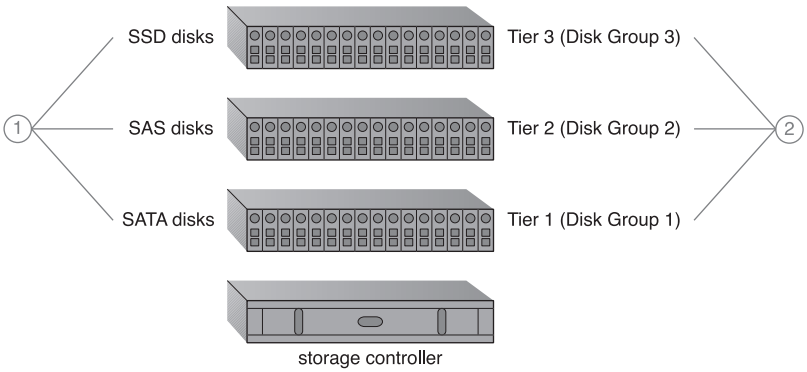


Figure 13.15

Different types of hard disks are installed in the enclosures of a cloud storage device (1). Similar disk types are grouped into tiers to create different grades of disk groups based on I/O performance (2).

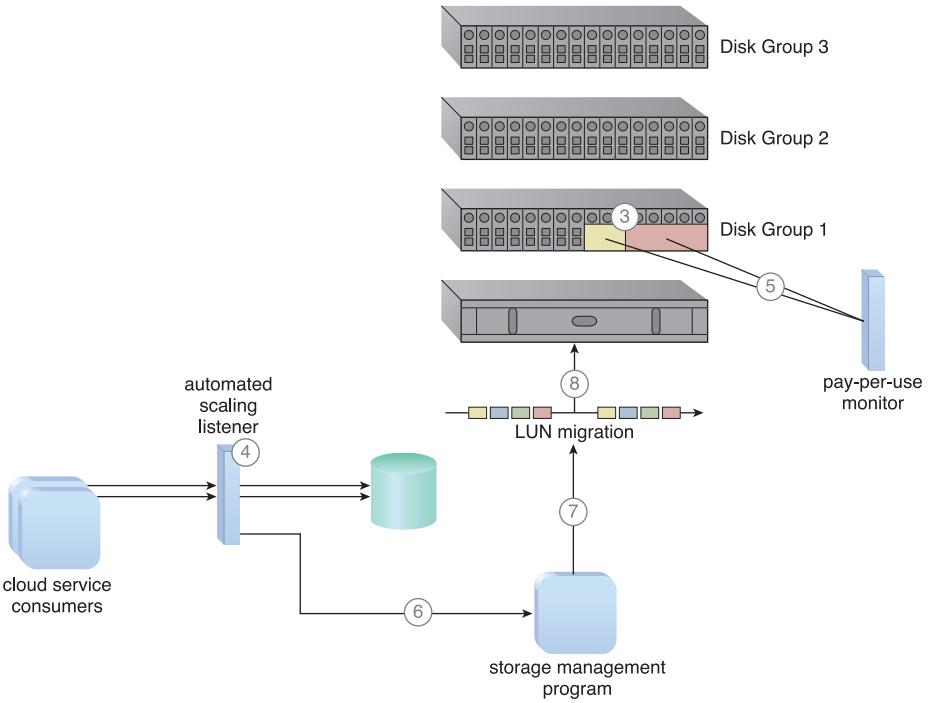
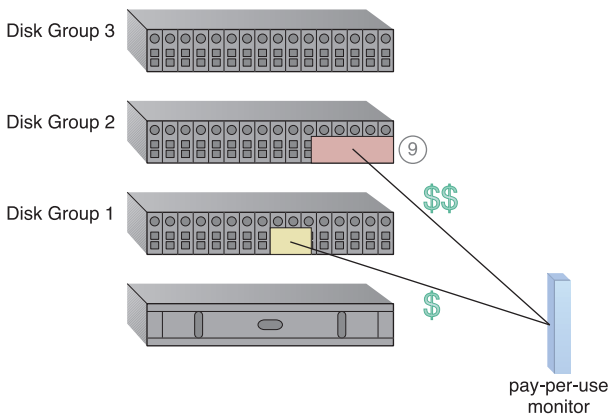


Figure 13.16

Two LUNs have been created on Disk Group 1 (3). The automated scaling listener monitors the requests in relation to pre-defined thresholds (4). The pay-per-use monitor tracks the actual amount of disk usage, based on free space and disk group performance (5). The automated scaling listener determines that the number of requests is reaching a threshold, and informs the storage management program that the LUN needs to be moved to a higher performance disk group (6). The storage management program signals the LUN migration program to perform the required move (7). The LUN migration program works with the storage controller to move the LUN to the higher capacity Disk Group 2 (8).

Figure 13.17

The usage price of the migrated LUN in Disk Group 2 is now higher than before, because a higher performance disk group is being used (9).



13.7 Load Balanced Virtual Switches Architecture

Virtual servers are connected to the outside world via virtual switches, which send and receive traffic with the same uplink. Bandwidth bottlenecks form when the network traffic on the uplink's port increases to a point that it causes transmission delays, performance issues, packet loss, and lag time (Figures 13.18 and 13.19).

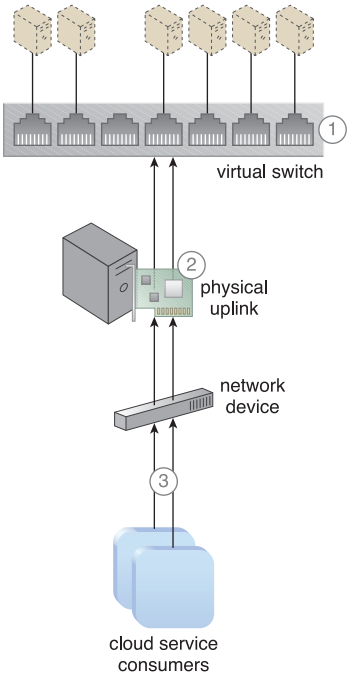


Figure 13.18
A virtual switch is interconnecting virtual servers (1). A physical network adapter has been attached to the virtual switch to be used as an uplink to the physical (external) network, connecting the virtual servers to cloud consumers (2). Cloud service consumers send requests via the physical uplink (3).

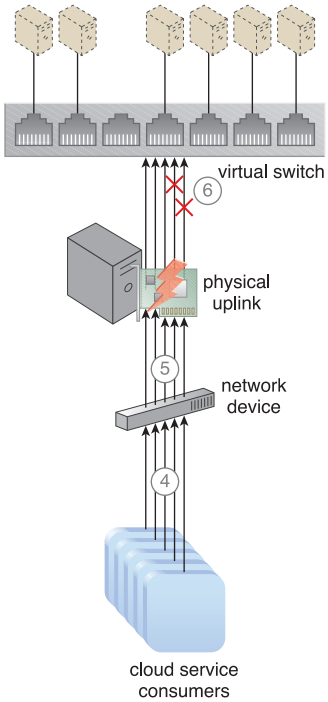


Figure 13.19
The amount of traffic passing through the physical uplink grows in parallel with the increasing number of requests. The number of packets that need to be processed and forwarded by the physical network adapter also increases (4). The physical adapter cannot handle the workload, now that the network traffic has exceeded its capacity (5). The network forms a bottleneck that results in performance degradation and the loss of delay-sensitive data packets (6).

The *load balanced virtual switches architecture* establishes a load balancing system where multiple uplinks are provided to balance network traffic workloads across multiple uplinks or redundant paths, which can help avoid slow transfers and data loss (Figure 13.20). Link aggregation can be executed to balance the traffic, which allows the workload to be distributed across multiple uplinks at the same time so that none of the network cards are overloaded.

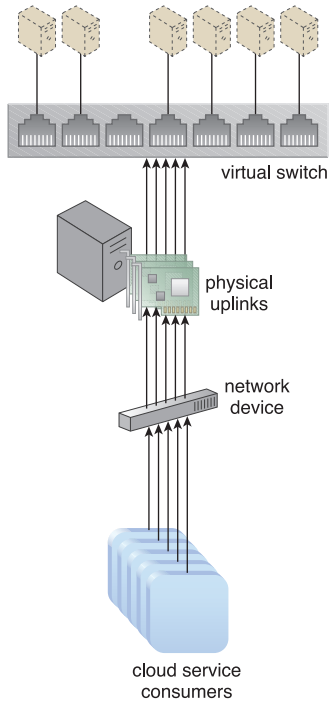


Figure 13.20
Additional physical uplinks are added to distribute and balance network traffic.

The virtual switch needs to be configured to support multiple physical uplinks, which are usually configured as an NIC team that has defined traffic-shaping policies.

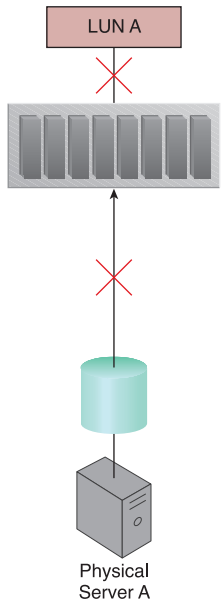
The following mechanisms can be incorporated into this architecture:

- *Cloud Usage Monitor* – Cloud usage monitors are used to monitor network traffic and bandwidth usage.
- *Hypervisor* – This mechanism hosts and provides the virtual servers with access to both the virtual switches and external network.
- *Load Balancer* – The load balancer distributes the network workload across the different uplinks.
- *Logical Network Perimeter* – The logical network perimeter creates boundaries that protect and limit the bandwidth usage for each cloud consumer.
- *Resource Replication* – This mechanism is used to generate additional uplinks to the virtual switch.
- *Virtual Server* – Virtual servers host the IT resources that benefit from the additional uplinks and bandwidth via virtual switches.

13.8 Multipath Resource Access Architecture

Certain IT resources can only be accessed using an assigned path (or hyperlink) that leads to their exact location. This path can be lost or incorrectly defined by the cloud consumer or changed by the cloud provider. An IT resource whose hyperlink is no longer in the possession of the cloud consumer becomes inaccessible and unavailable (Figure 13.21). Exception conditions that result from IT resource unavailability can compromise the stability of larger cloud solutions that depend on the IT resource.

Figure 13.21
Physical Server A is connected to LUN A via a single fiber channel, and uses the LUN to store different types of data. The fiber channel connection becomes unavailable due to a HBA card failure and invalidates the path used by Physical Server A, which has now lost access to LUN A and all of its stored data.



The *multipath resource access architecture* establishes a multipathing system with alternative paths to IT resources, so that cloud consumers have the means to programmatically or manually overcome path failures (Figure 13.22).

This technology architecture requires the use of a multipathing system and the creation of alternative physical or virtual hyperlinks that are assigned to specific IT resources. The multipathing system resides on the server or hypervisor, and ensures that each IT resource can be seen via each alternative path identically (Figure 13.23).

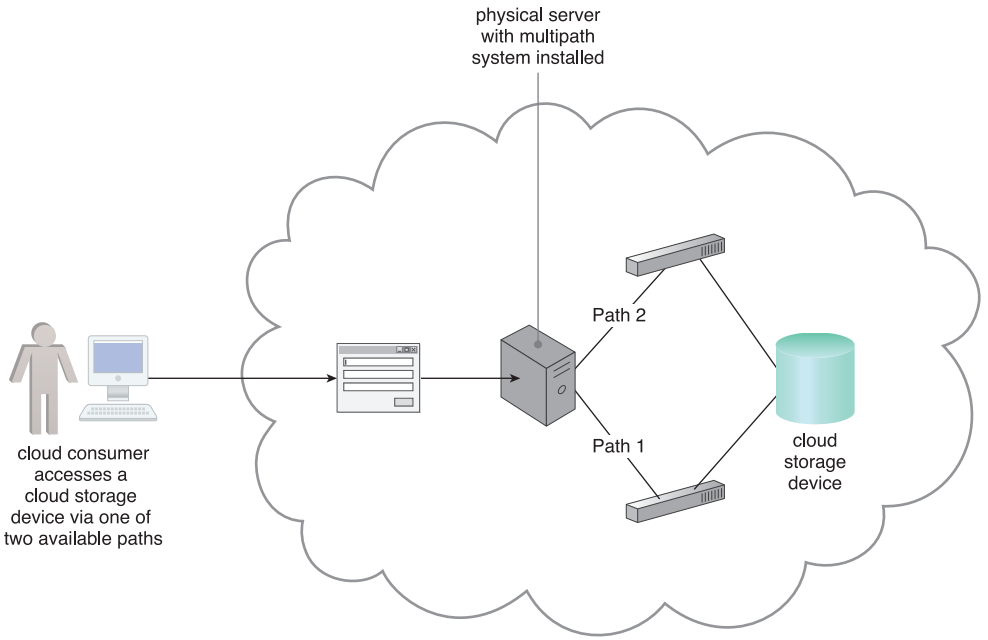


Figure 13.22
A multipathing system is providing alternative paths to a cloud storage device.

This architecture can involve the following mechanisms:

- *Cloud Storage Device* – The cloud storage device is a common IT resource that requires the creation of alternative paths in order to remain accessible to solutions that rely on data access.
- *Hypervisor* – Alternative paths to a hypervisor are required in order to have redundant links to the hosted virtual servers.
- *Logical Network Perimeter* – This mechanism guarantees the maintenance of cloud consumer privacy, even when multiple paths to the same IT resource are created.
- *Resource Replication* – The resource replication mechanism is required when a new instance of an IT resource needs to be created to generate the alternative path.
- *Virtual Server* – These servers host the IT resources that have multipath access via different links or virtual switches. Hypervisors can provide multipath access to the virtual servers.

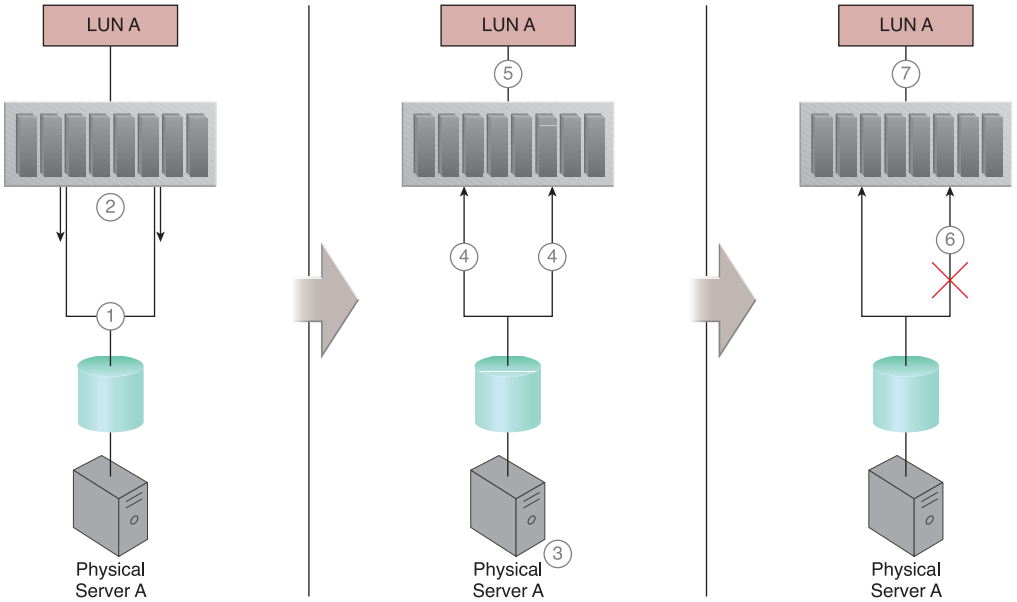


Figure 13.23

Physical Server A is connected to the LUN A cloud storage device via two different paths (1). LUN A is seen as different LUNs from each of the two paths (2). The multipathing system is configured (3). LUN A is seen as one identical LUN from both paths (4), and Physical Server A has access to LUN A from two different paths (5). A link failure occurs and one of the paths becomes unavailable (6). Physical Server A can still use LUN A because the other link remains active (7).

13.9 Persistent Virtual Network Configuration Architecture

Network configurations and port assignments for virtual servers are generated during the creation of the virtual switch on the host physical server and the hypervisor hosting the virtual server. These configurations and assignments reside in the virtual server's immediate hosting environment, meaning a virtual server that is moved or migrated to another host will lose network connectivity because destination hosting environments do not have the required port assignments and network configuration information (Figure 13.24).

In the *persistent virtual network configuration architecture*, network configuration information is stored in a centralized location and replicated to physical server hosts. This allows the destination host to access the configuration information when a virtual server is moved from one host to another.

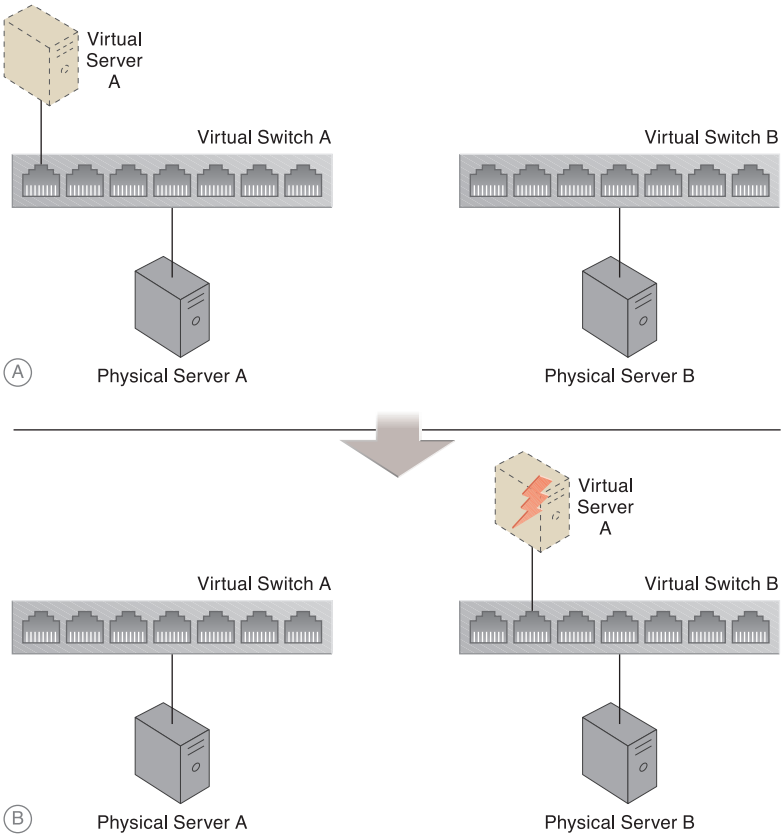


Figure 13.24

Part A shows Virtual Server A connected to the network through Virtual Switch A, which was created on Physical Server A. In Part B, Virtual Server A is connected to Virtual Switch B after being moved to Physical Server B. The virtual server cannot connect to the network because its configuration settings are missing.

The system established with this architecture includes a centralized virtual switch, VIM, and configuration replication technology. The centralized virtual switch is shared by physical servers and configured via the VIM, which initiates replication of the configuration settings to the physical servers (Figure 13.25).

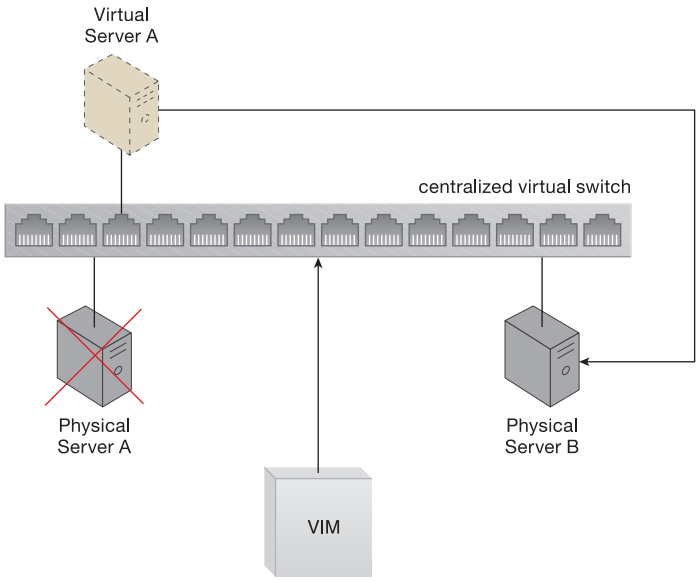


Figure 13.25

A virtual switch's configuration settings are maintained by the VIM, which ensures that these settings are replicated to other physical servers. The centralized virtual switch is published, and each host physical server is assigned some of its ports. Virtual Server A is moved to Physical Server B when Physical Server A fails. The virtual server's network settings are retrievable, since they are stored on a centralized virtual switch that is shared by both physical servers. Virtual Server A maintains network connectivity on its new host, Physical Server B.

In addition to the virtual server mechanism for which this architecture provides a migration system, the following mechanisms can be included:

- *Hypervisor* – The hypervisor hosts the virtual servers that require the configuration settings to be replicated across the physical hosts.
- *Logical Network Perimeter* – The logical network perimeter helps ensure that access to the virtual server and its IT resources is isolated to the rightful cloud consumer, before and after a virtual server is migrated.
- *Resource Replication* – The resource replication mechanism is used to replicate the virtual switch configurations and network capacity allocations across the hypervisors, via the centralized virtual switch.

13.10 Redundant Physical Connection for Virtual Servers Architecture

A virtual server is connected to an external network via a virtual switch uplink port, meaning the virtual server will become isolated and disconnected from the external network if the uplink fails (Figure 13.26).

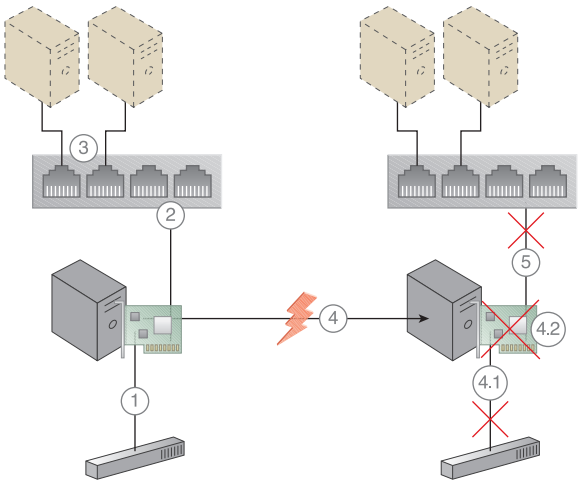
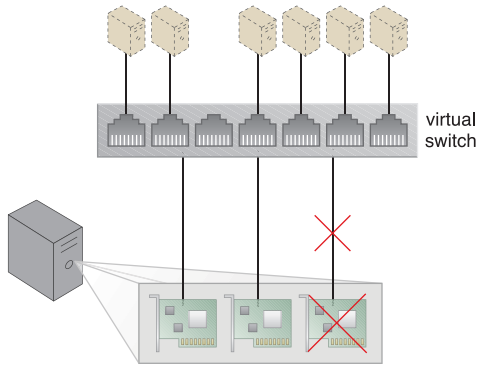


Figure 13.26

A physical network adapter installed on the host physical server is connected to the physical switch on the network (1). A virtual switch is created for use by two virtual servers. The physical network adapter is attached to the virtual switch to act as an uplink, since it requires access to the physical (external) network (2). The virtual servers communicate with the external network via the attached physical uplink network card (3). A connection failure occurs, either because of a physical link connectivity issue between the physical adapter and the physical switch (4.1), or because of a physical network card failure (4.2). The virtual servers lose access to the physical external network and are no longer accessible to their cloud consumers (5).

The *redundant physical connection for virtual servers architecture* establishes one or more redundant uplink connections and positions them in standby mode. This architecture ensures that a redundant uplink connection is available to connect the active uplink, whenever the primary uplink connection becomes unavailable (Figure 13.27).

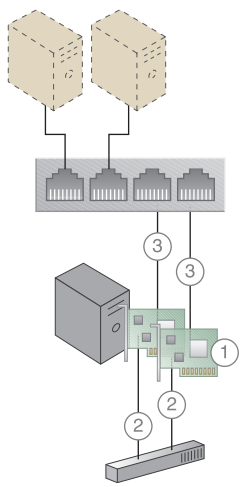
Figure 13.27
Redundant uplinks are installed on a physical server that is hosting several virtual servers. When an uplink fails, another uplink takes over to maintain the virtual servers' active network connections.



In a process that is transparent to both virtual servers and their users, a standby uplink automatically becomes the active uplink as soon as the main uplink fails, and the virtual servers use the newly active uplink to sends packets externally.

The second NIC does not forward any traffic while the primary uplink is alive, even though it receives the virtual server's packets. However, the secondary uplink will start forwarding packets immediately if the primary uplink were to fail (Figures 13.28 to 13.30). The failed uplink becomes the primary uplink again after it returns to operation, while the second NIC returns to standby mode.

Figure 13.28
A new network adapter is added to support a redundant uplink (1). Both network cards are connected to the physical external switch (2), and both physical network adapters are configured to be used as uplink adapters for the virtual switch (3).



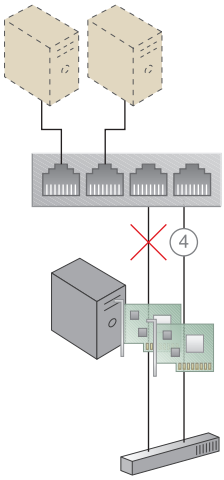


Figure 13.29
One physical network adapter is designated as the primary adapter (4), while the other is designated as the secondary adapter providing the standby uplink. The secondary adapter does not forward any packets.

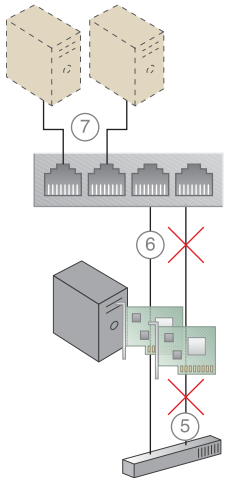


Figure 13.30
The primary uplink becomes unavailable (5). The secondary standby uplink automatically takes over and uses the virtual switch to forward the virtual servers' packets to the external network (6). The virtual servers do not experience interruptions and remain connected to the external network (7).

The following mechanisms are commonly part of this architecture, in addition to the virtual server:

- *Failover System* – The failover system performs the transition of unavailable uplinks to standby uplinks.
- *Hypervisor* – This mechanism hosts virtual servers and some virtual switches, and provides virtual networks and virtual switches with access to the virtual servers.
- *Logical Network Perimeter* – Logical network perimeters ensure that the virtual switches that are allocated or defined for each cloud consumer remain isolated.
- *Resource Replication* – Resource replication is used to replicate the current status of active uplinks to standby uplinks so as to maintain the network connection.

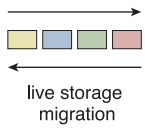
13.11 Storage Maintenance Window Architecture

Cloud storage devices that are subject to maintenance and administrative tasks sometimes need to be temporarily shut down, meaning cloud service consumers and IT resources consequently lose access to these devices and their stored data (Figure 13.31).

The data of a cloud storage device that is about to undergo a maintenance outage can be temporarily moved to a secondary duplicate cloud storage

LIVE STORAGE MIGRATION

The live storage migration program is a sophisticated system that utilizes the LUN migration component to reliably move LUNs by enabling the original copy to remain active until after the destination copy has been verified as being fully functional.



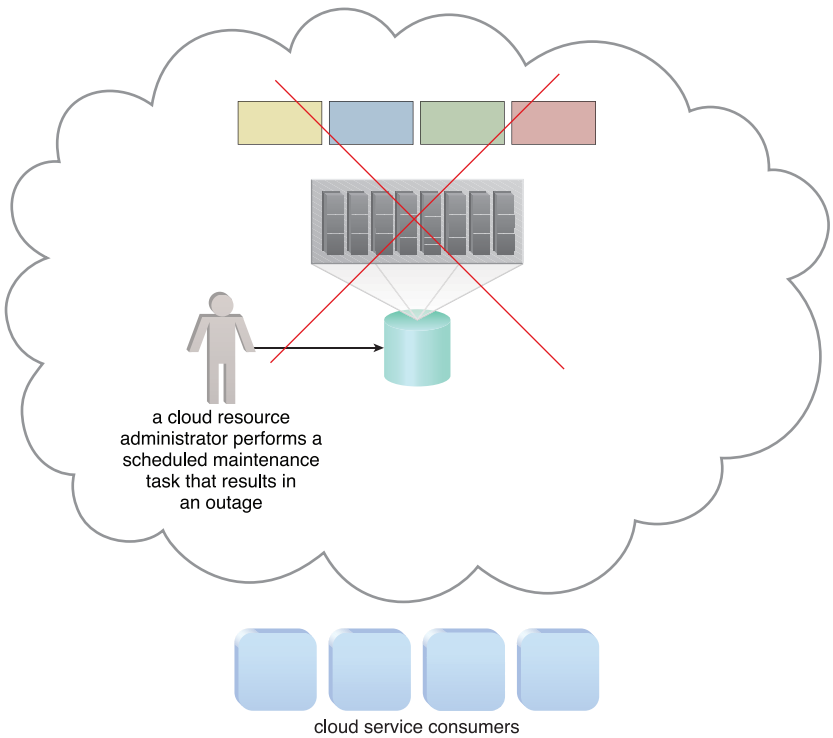


Figure 13.31

A pre-scheduled maintenance task carried out by a cloud resource administrator causes an outage for the cloud storage device, which becomes unavailable to cloud service consumers. Because cloud consumers were previously notified of the outage, cloud consumers do not attempt any data access.

device. The *storage maintenance window architecture* enables cloud service consumers to be automatically and transparently redirected to the secondary cloud storage device, without becoming aware that their primary storage device has been taken offline.

This architecture uses a live storage migration program, as demonstrated in Figures 13.32 to 13.37.

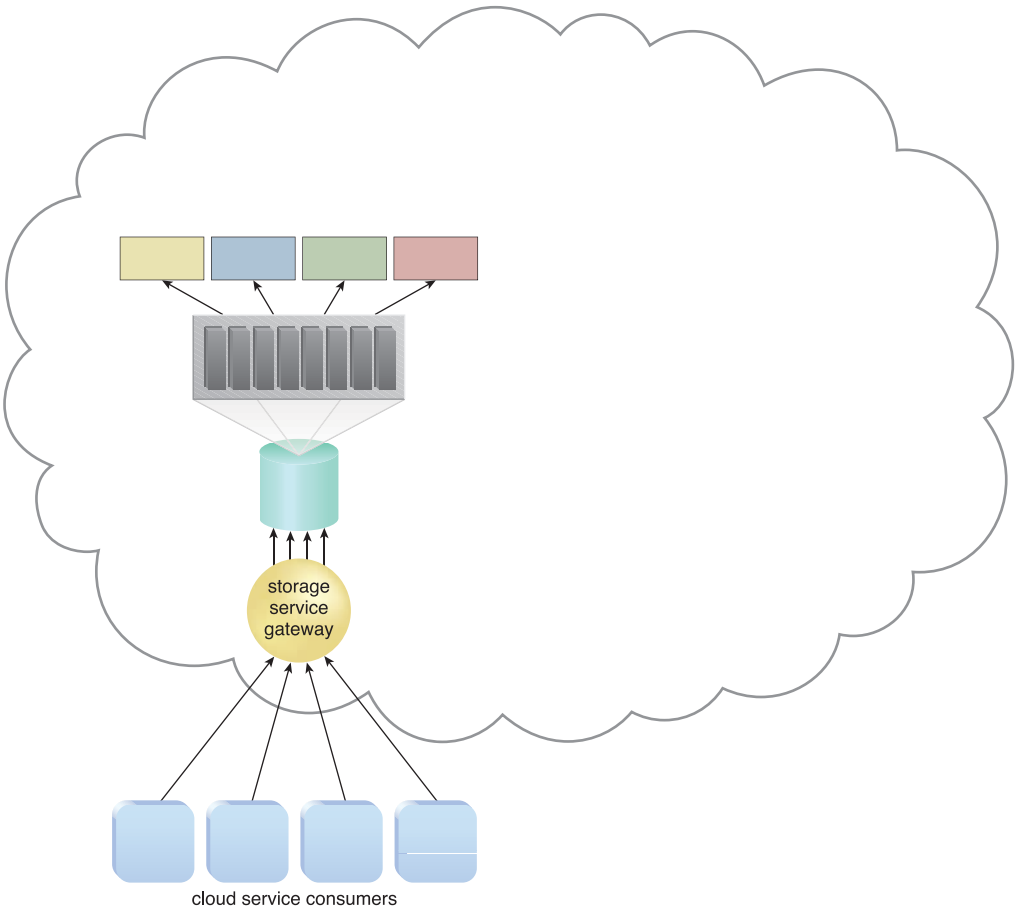


Figure 13.32
The cloud storage device is scheduled to undergo a maintenance outage, but unlike the scenario depicted in Figure 13.31, the cloud service consumers were not notified of the outage and continue to access the cloud storage device.

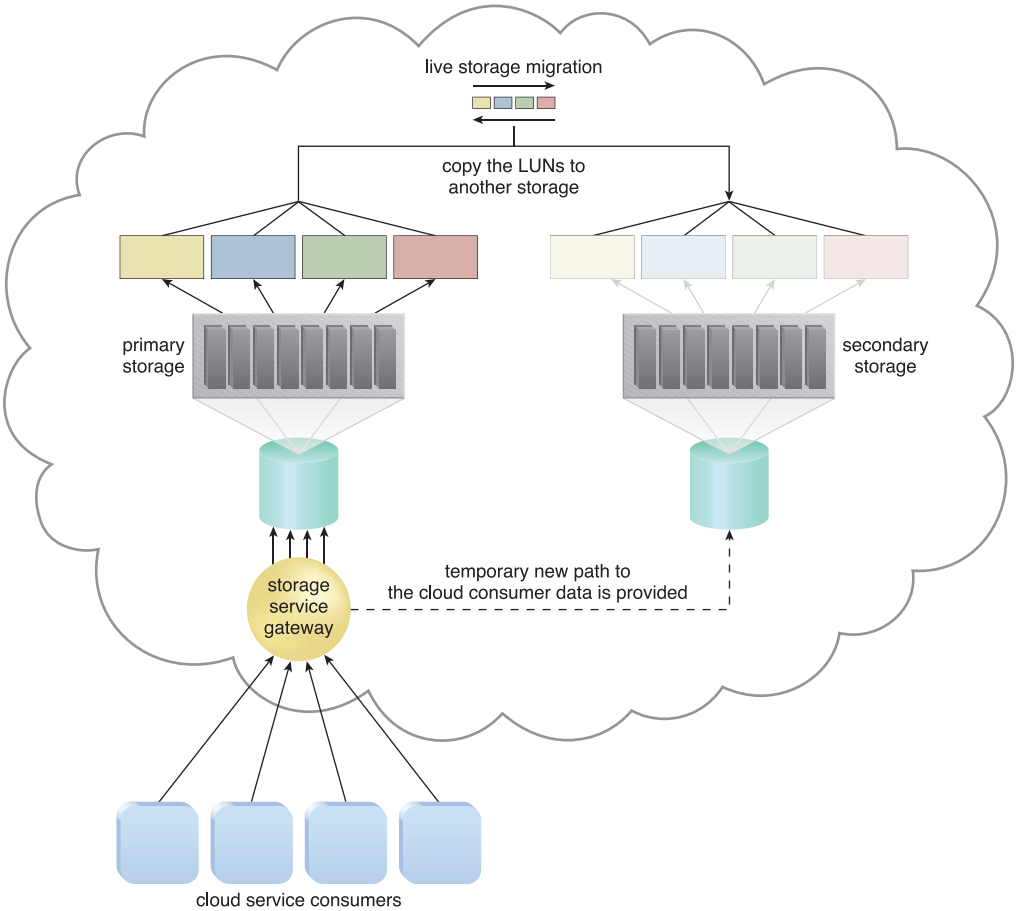


Figure 13.33
Live storage migration moves the LUNs from the primary storage device to a secondary storage device.

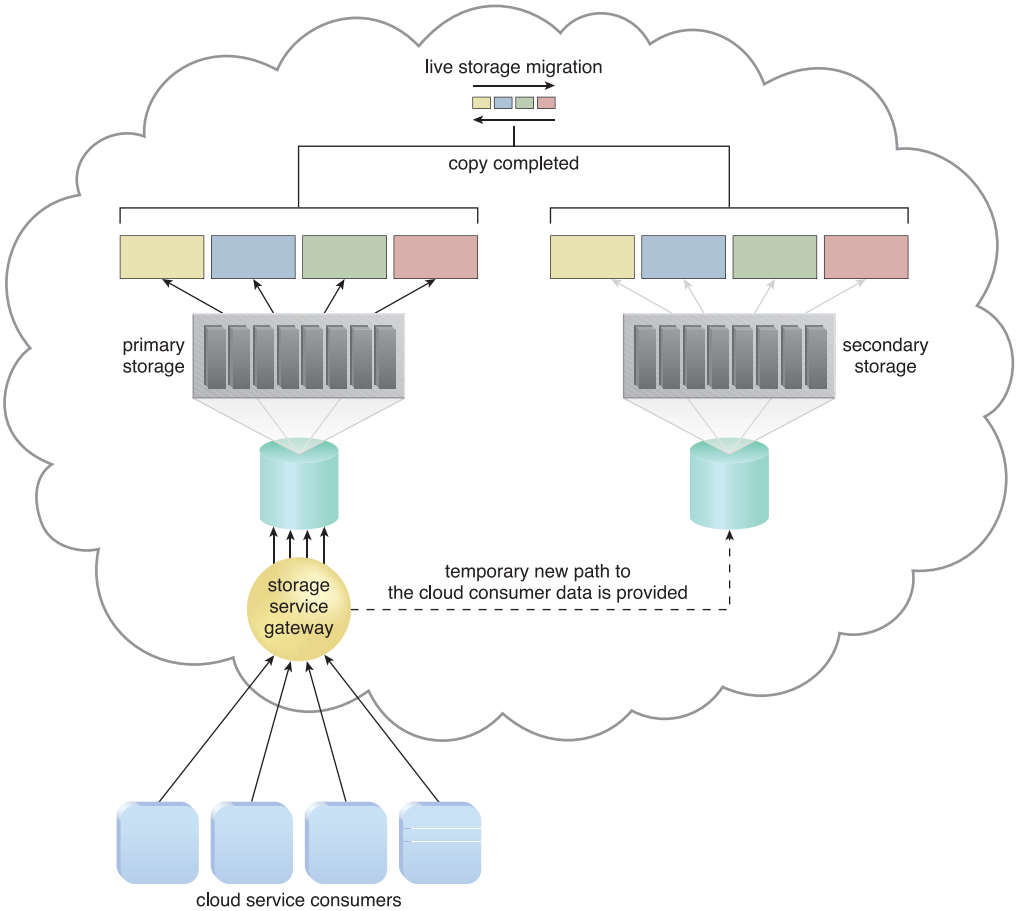


Figure 13.34

Requests for the data are forwarded to the duplicate LUNs on the secondary storage device, once the LUN's data has been migrated.

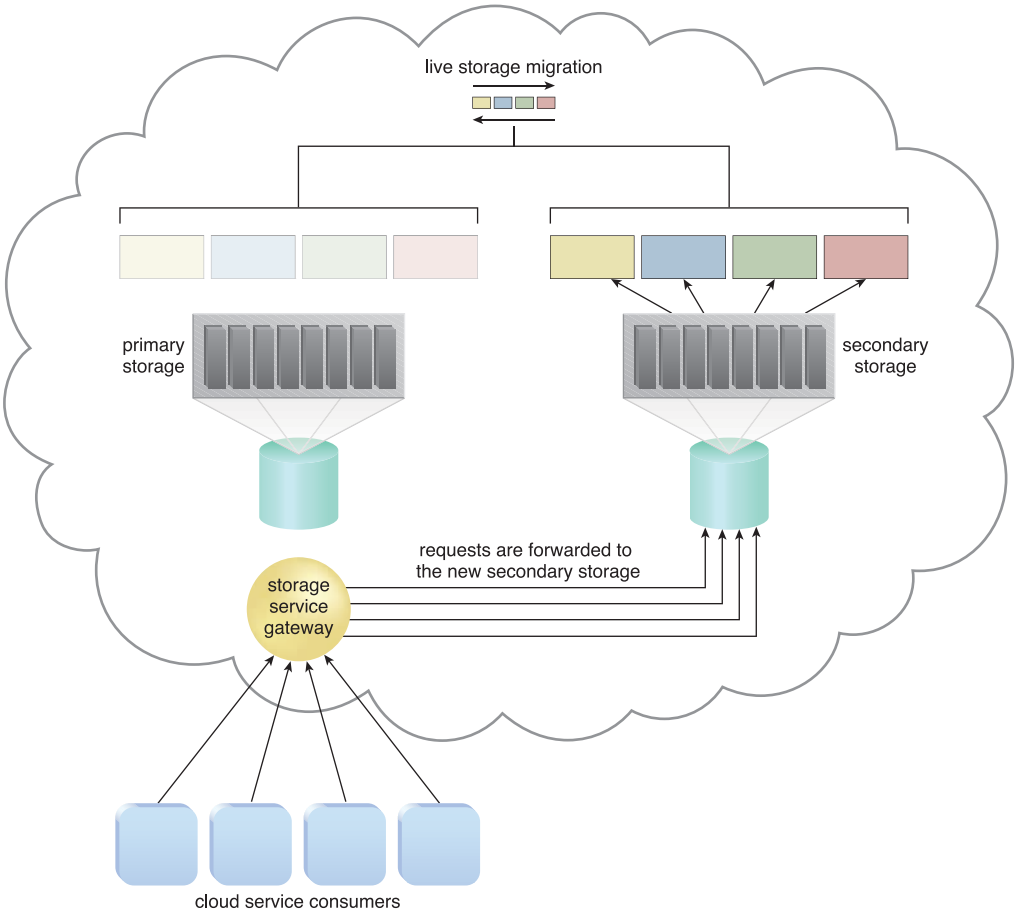


Figure 13.35
The primary storage is powered off for maintenance.

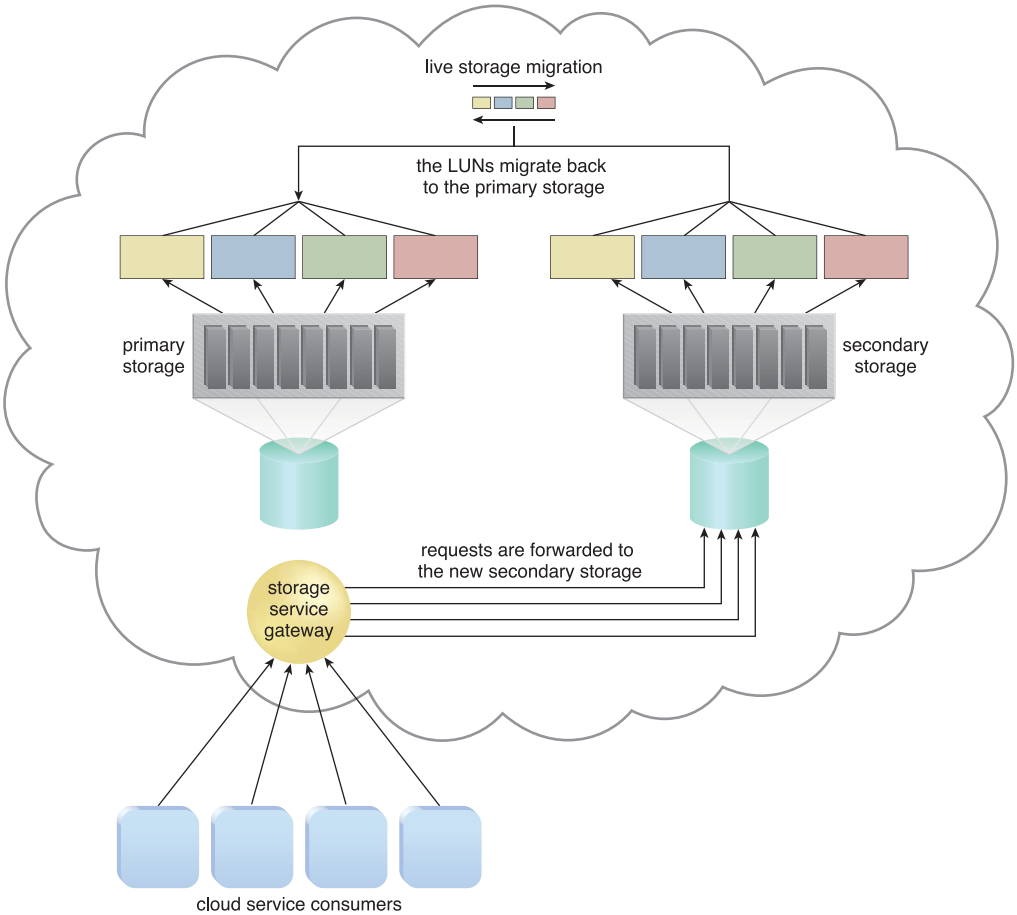


Figure 13.36

The primary storage is brought back online, after the maintenance task is finished. Live storage migration restores the LUN data from the secondary storage device to the primary storage device.

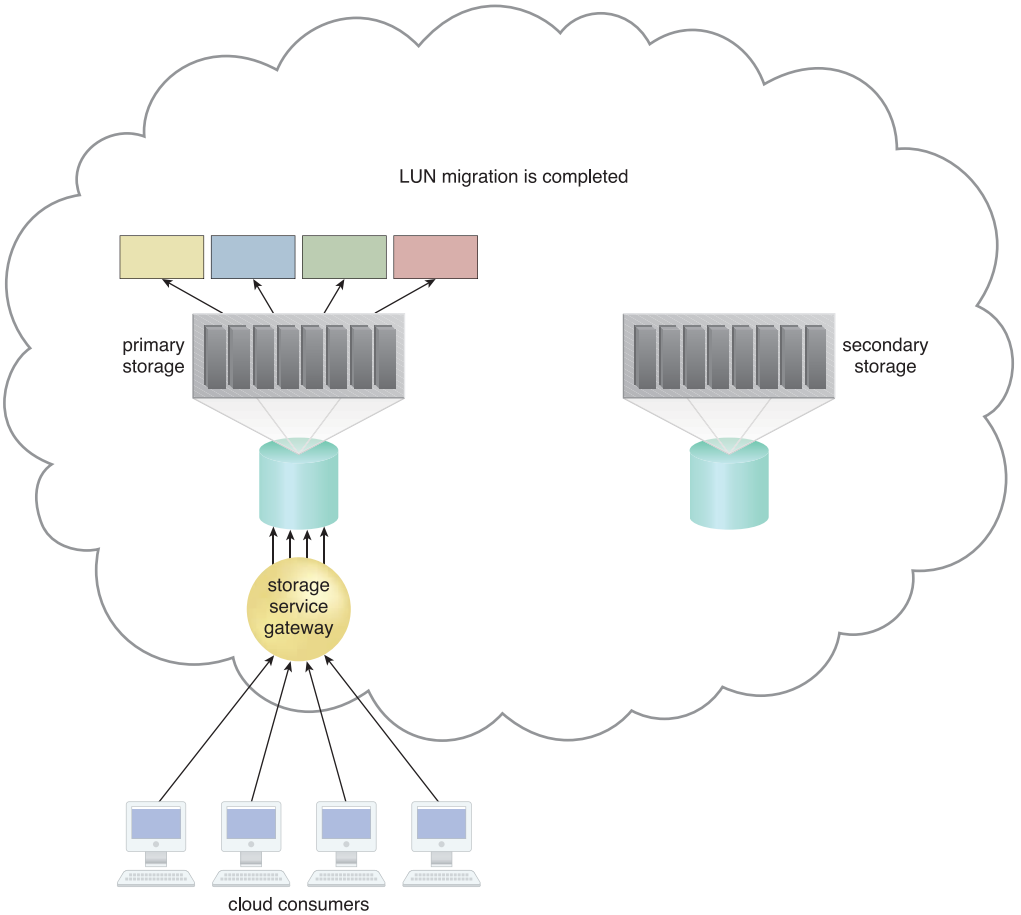


Figure 13.37
The live storage migration process is completed and all of the data access requests are forwarded back to the primary cloud storage device.

In addition to the cloud storage device mechanism that is principal to this architecture, the resource replication mechanism is used to keep the primary and secondary storage devices synchronized. Both manually and automatically initiated failover can also be incorporated into this cloud architecture via the failover system mechanism, even though the migration is often pre-scheduled.