CSE423 Virtualization and Cloud Computing

UNIT-III

Q1. Explain briefly how do you Understanding the cloud computing stack: SaaS, PaaS, and IaaS

i. SaaS applications are designed for end-users and delivered over the web.

ii. PaaS is the set of tools and services designed to make coding and deploying those applications quick and efficient.

iii. IaaS is the hardware and software that powers it all, including servers, storage, networks, and operating systems.

A. Software as a service

Software as a service (SaaS) is software deployed over the Internet. A provider licenses a SaaS app to customers as an on-demand service, either through a subscription or through a pay-as-you-go model.

Characteristics of SaaS

Following are some defining characteristics of SaaS:

i. Web access to commercial software

ii. Software managed from a central location

iii. Software delivered in a “one to many” model

iv. Users not required to handle software upgrades and patches

v. Application Programming Interfaces (APIs) makes integration between different pieces of software possible

When to use SaaS

The following are solutions that are prime candidates for an initial move to SaaS:

Standard offerings in which the solution is largely undifferentiated—for example, email. Competitors often use the same email software because this fundamental technology is a requirement for doing business but doesn’t award a competitive advantage.

Applications in which there is significant interplay between the organization and the outside world—for example, email newsletter campaign software.

Applications that have a significant need for web or mobile access—for example, mobile sales management software.

Collaboration software for a specific project.

Software for which demand spikes significantly—for example, tax or billing software used after a month.

When not to use SaaS

Although SaaS is a valuable tool, it’s not the best option for software delivery in certain situations. The following are examples where SaaS might not be appropriate:

i. Applications in which that need fast processing of real-time data.

ii. Applications which cannot host data externally because of legislation or other regulations.

B. Platform as a service

PaaS is similar to SaaS except that, rather than being software delivered over the web, it is a platform for the creation of software delivered over the web.

Characteristics of PaaS

Following are some basic characteristics of PaaS:

i. Services to develop, test, deploy, host, and support applications in the same integrated development environment—all the various services needed to fulfill the app development process.

ii. Web-based user interface (UI) creation tools used to create, change, test, and deploy different UI scenarios.
iii. Multi-tenant architecture in which many concurrent users use the same development app.
iv. Built-in scalability of deployed software, including load balancing and failover.
v. Integration with web services and databases via common standards.
vi. Support for development team collaboration. Some PaaS solutions include project planning and communication tools to support development team collaboration.

When to use PaaS

PaaS is especially useful when multiple developers are working on a development project or when other external parties need to interact with the development process. PaaS is invaluable for those who have an existing data source and want to create applications that leverage that data. PaaS is also useful when developers want to automate testing and deployment services.

When not to use PaaS

The ability to automate processes, use predefined components and building blocks, and deploy automatically to production creates value that makes PaaS highly attractive. However, PaaS might not be ideal in the following situations:

The app needs to be highly portable in terms of hosting.

Proprietary languages or approaches would impact the development process.

A proprietary language would hinder later moves to another provider.

Application performance requires customization of the underlying hardware and software.

A PaaS development environment enables quicker creation of apps. In some examples, in the absence of PaaS, the cost of developing the app would have been prohibitive.

C. Infrastructure as a service

Infrastructure as a service (IaaS) delivers cloud computing infrastructure—servers, storage, network, and operating systems as an on-demand service. Rather than purchasing servers, software, data-center space, or network equipment, clients instead buy those resources as a fully outsourced on-demand service.

You can obtain IaaS as public or private infrastructure, or a combination of the two.

i. *Public cloud* is infrastructure that consists of shared resources, deployed on a *self-service* basis over the Internet.
ii. *Private cloud* is infrastructure that emulates some cloud computing features, like virtualization, but does so on a private network.
iii. Some hosting providers offer a *hybrid cloud*. This cloud is a combination of traditional dedicated hosting alongside public cloud networks, private cloud networks, or both.

Characteristics of IaaS

The following core characteristics describe IaaS:

i. Resources distributed as a service
ii. Allows for dynamic scaling
iii. Has a variable cost, utility pricing model
iv. Allows for multiple users on a single piece of hardware

When to use IaaS

The following situations are particularly suitable for cloud infrastructure:

i. Any time there are significant spikes and troughs of demand on the infrastructure.
ii. For new organizations without the capital to invest in hardware.
iii. When the organization is growing rapidly, and scaling hardware would be problematic.
iv. When there is pressure on the organization to limit capital expenditure and to move to operating expenditure.

For a specific line of business, trial, or temporary infrastructure needs.

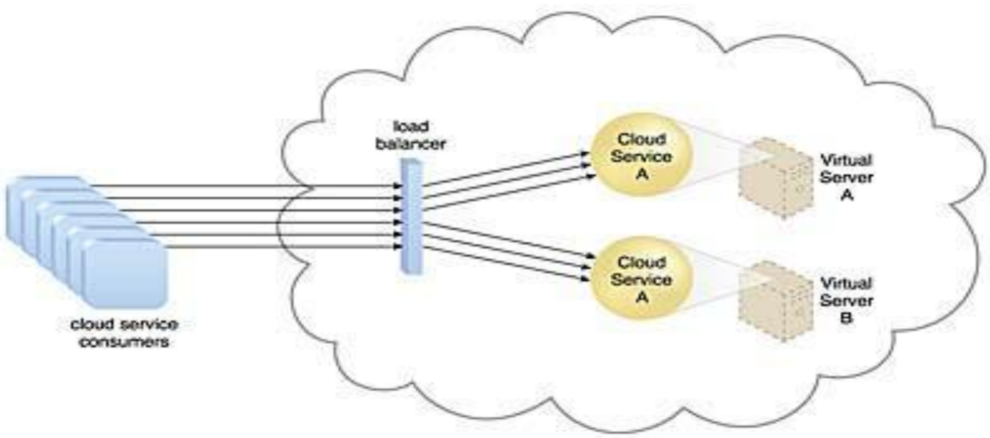
When not to use IaaS

i. Although IaaS has advantages when scalability and quick provisioning are beneficial, the following is a situation in which its limitations might be problematic:
ii. Regulatory compliance makes the offshoring or outsourcing of data storage and processing difficult.

Q2. Explain briefly the Workload Distribution Architecture with suitable diagram?

Workload distribution architecture uses IT resources that can be horizontally scaled with the use of one or more identical IT resources. This is accomplished through the use of a load balancer that provides runtime logic which distributes the workload among the available IT assets evenly. This model can be applied to any IT resource and is commonly used with; distributed virtual servers, cloud storage devices, and cloud services. In addition to a load balancer and the previously mentioned resources, the following mechanisms can also be a part of this model:

- Cloud Usage Monitor that can carry out run-time tracking and data processing.
- Audit Monitor used for monitoring the system as may be required to fulfill legal requirements.
- Hypervisor which is used to manage workloads and virtual hosts that require distribution.
- Logical network perimeter which isolates cloud consumer network boundaries.
- Resource clusters commonly used to support workload balancing between cluster nodes.
- Resource replication which generates new instances of virtualized resources under increased workloads.

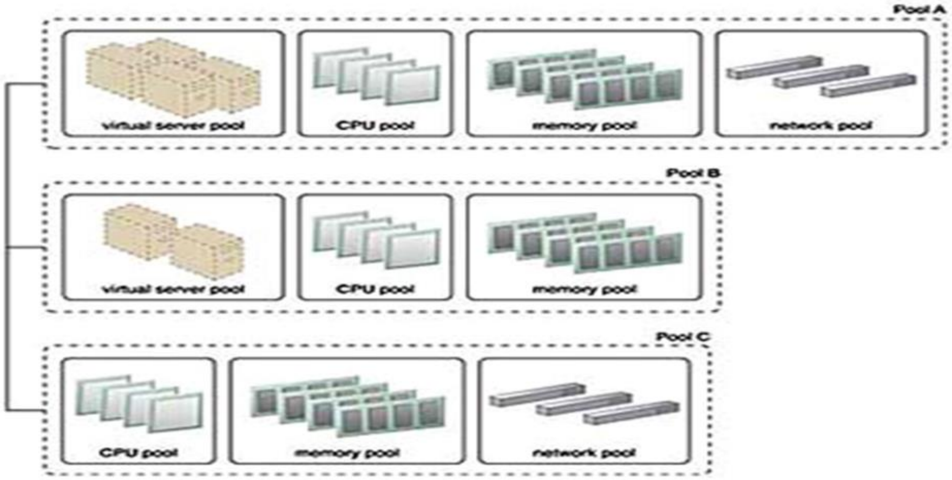


The workload architecture model basically functions as follows; Resource A and resource B are exact copies of the same resource. Inbound requests from consumers are handled by the load balancer which forwards the request to the appropriate resource dependent on workload being handled by each resource. In other words, if resource A is busier than resource B, it will forward the resource request to resource B. In this manner this model distributes the load among the available IT resources based on workload of each resource.

Q3. What does Resource Pooling Architecture is about?
Resource pooling architecture is based on pooling identical IT resources into groups. Pools can be physical and virtual resources. These identical resources are automatically grouped and maintained by the system which ensures they remain synchronized. Examples of resource pools are as follows:

- Physical server pools which consist of networked servers that already have operating systems and other required applications installed and are ready for immediate use.
- Virtual server pools usually configured from templates pre-chosen by the customer when they are provisioned.

- Storage pools consisting of file, or, block based storage containers.
- Network pools which consist of different pre-configured network devices. Example; virtual firewalls and switches used for redundant connections, load balancing, and link aggregation.
- CPU pools which allot CPU resources to virtual servers.
- Physical RAM pools that can be used to vertically scale newly provisioned physical servers.



Resource pool architecture uses many of the same type of mechanisms as workload distribution architectures as follows:

- Audit pools to monitor resource pool usage.
- Cloud Usage Monitor used at run-time to track and synchronize IT resources and underlying management systems.
- Hypervisor which manages virtual server pools as well as virtual server access to other resource pools.
- Logical Network Perimeter that logically organizes and isolates resource pools.
- Pay-Per-Use Monitor which collects usage and billing information based on how customers use cloud IT resources.
- Remote Administration System which is used to interface with back-end systems in order to administer resource pools.
- Resource Management System which provides customers with tools and permission management options used in administering resource pools.
- Resource Replication which generates new instances of resources for resource pools as demand dictates.

Q4. What are the important steps involved in capacity Planning?

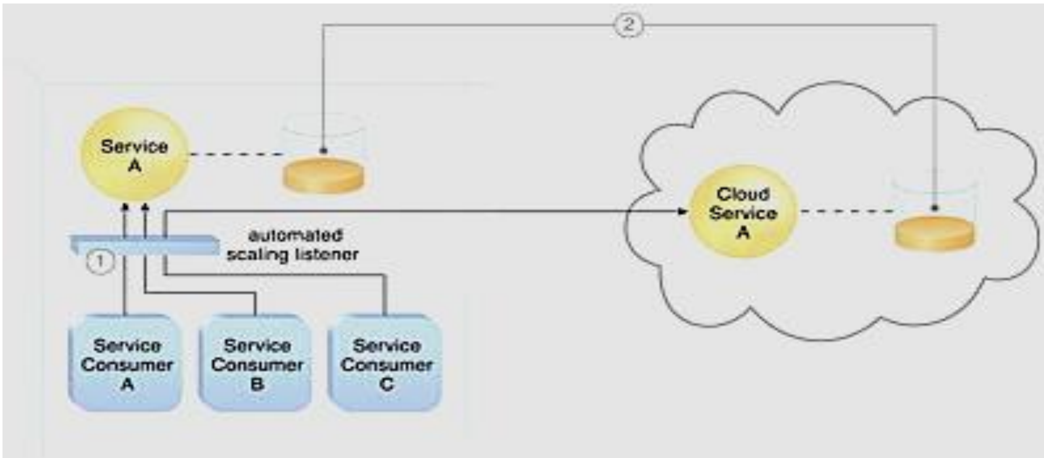
Ans:- Determine the distinctiveness of the present system.

i. Determine the working load for different resources in the system such as CPU, RAM, network, etc.
ii. Load the system until it gets overloaded; & state what's requiring to uphold acceptable performance.
iii. Predict the future based on older statistical reports & other factors.
iv. Deploy resources to meet the predictions & calculations.
v. Repeat step (i) through (v) as a loop.

Q5. What do you understand by Cloud Bursting Architecture?

Ans :- 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.



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 beenexceeded (1). A resource replication system is used to keep state management databases synchronized (2).

Q6. Explain the briefly the Elastic Disk provisioning Architecture?

1. Ans:- 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.
2. This system uses thin-provisioning technology for the dynamic allocation of storage space, and is further supported by runtime usage monitoring to collect accurate usage data for billing purposes.
3. 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

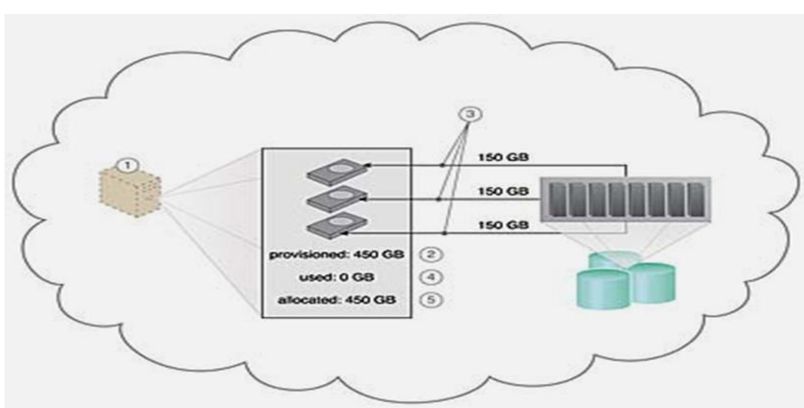


Figure 1. The Cloud consumers request a Virtual server with three hard disks, each with a capacity of 150 GB.

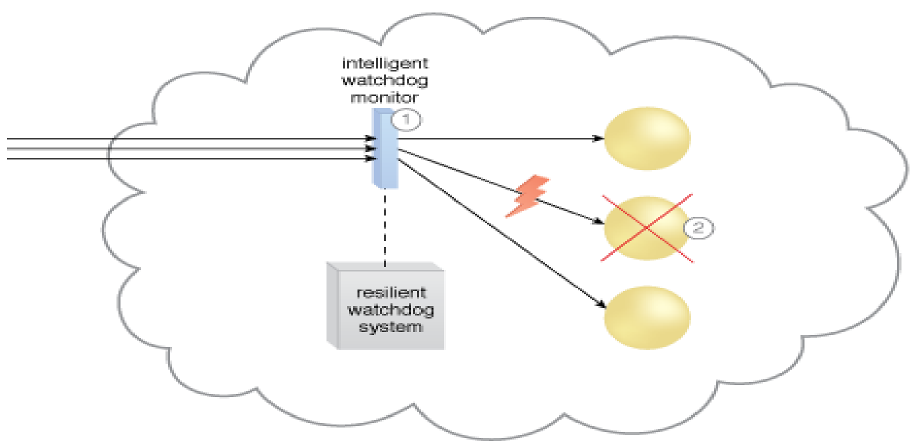
1. The Virtual server is provisioned according to the elastic disk provisioning architecture with a total of 450 GB
2. The 450 GB is allocated to the Virtual server by the cloud provider.
3. The cloud consumer has not installed any software yet, meaning the actual used space is 0 GB.
4. Because the 450 GB are already allocated and reserved for the cloud consumer, it will be charged for 450 GB of Disk usage as of the point of allocation

Q7. **Dynamic Failure Detection and Recovery Architecture?**

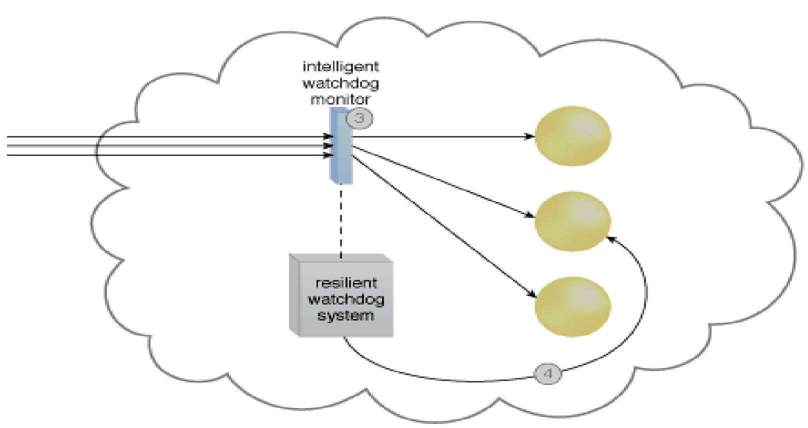
Ans:-i. IT resources can experience failure conditions that require more than manual intervention to resolve. Manually administering and solving IT resource failures is generally inefficient.

ii.The dynamic failure detection and recovery architecture establishes a resilient watchdog system to monitor and respond to a wide range of pre-defined failure scenarios.

iii. This system notifies and escalates the failure conditions that it cannot automatically resolve itself. It relies on specialized cloud storage usage monitor called the intelligent watchdog monitor to actively track IT resources and take pre-defined tasks and actions to predefined events.

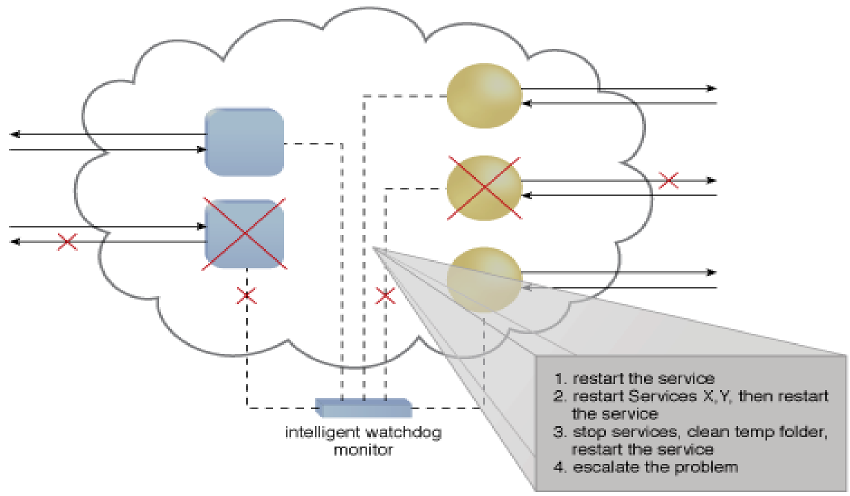


1)The intelligent watchdog monitor keeps track of cloud consumer requests, (2) and detects that a cloud service has failed.



The intelligent watchdog monitor notifies the resilient watchdog system (3), which restores the cloud service based on predefined policies (4).

In the event of any failures, the active monitor refers to its predefined policies to recover the service step by step, escalating the processes as the problem proves to be deeper than expected.

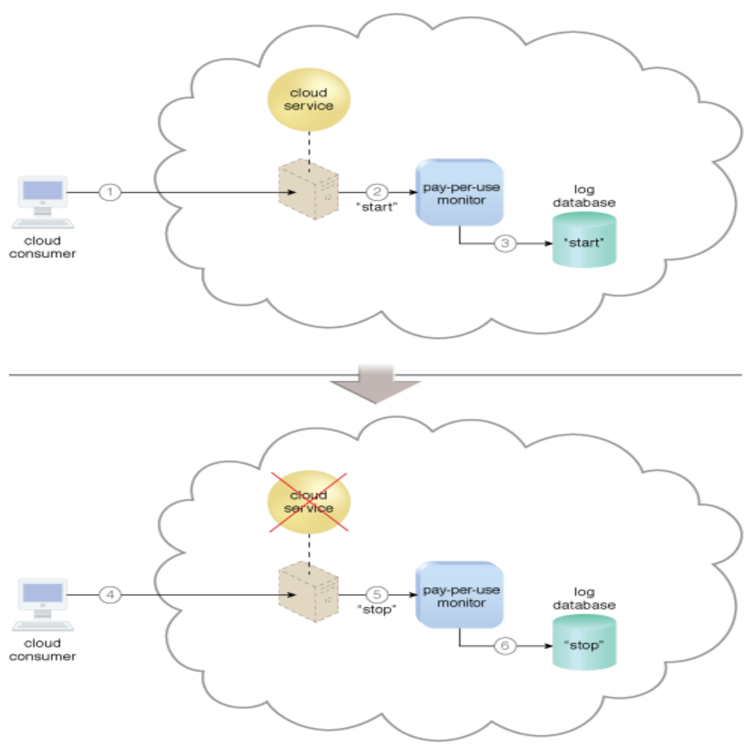


Q8. Explain the concept of **PAY-PER-USE MONITOR ?**

Ans:- The pay-per-use monitor mechanism measures cloud-based IT resource usage in accordance with predefined pricing parameters and generates usage logs for fee calculations and billing purposes.
Some typical monitoring variables are:-

i. request/response message quantity
ii. transmitted data volume
iii. bandwidth consumption

The data collected by the pay-per-use monitor is processed by a billing management system that calculates the payment fees.



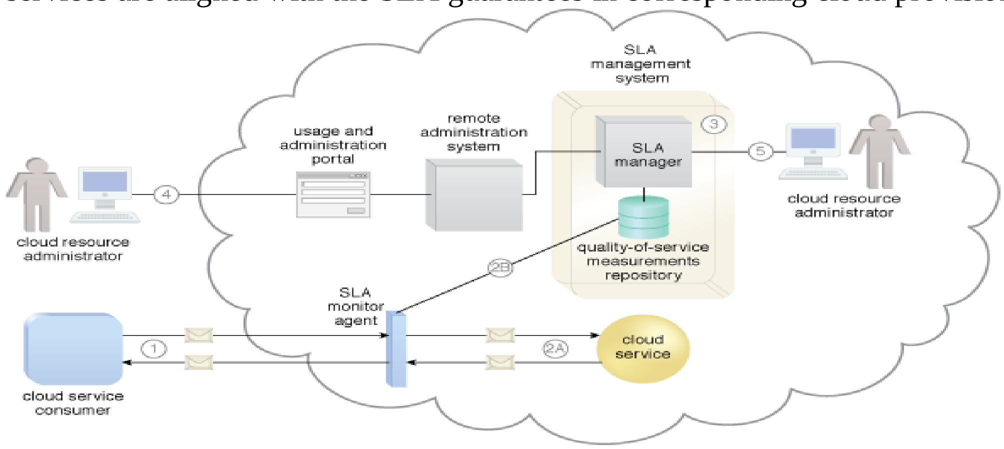
1) A cloud consumer requests the creation of a new instance of a cloud service
2) The IT resource is instantiated and they pay-per-use monitor mechanism receives a “start” event notification from the resource software
3) The pay-per-use monitor stores the value timestamp in the log database
4) The cloud consumer later requests that the cloud service instance be stopped

The pay-per-use monitor receives a “stop” event notification from the resource software and stores the value timestamp in the log database

Q9. How does the SLA Management System work in Cloud Computing System?

Ans :- The SLA management system mechanism represents a range of commercially available cloud management products that provide features pertaining to the administration, collection, storage, reporting, and runtime notification of SLA data.

An SLA management system deployment will generally include a repository used to store and retrieve collected SLA data based on pre-defined metrics and reporting parameters. The metrics monitored for individual cloud services are aligned with the SLA guarantees in corresponding cloud provisioning contracts.



A cloud service consumer interacts with a cloud service (1).

An SLA monitor intercepts the exchanged messages, evaluates the interaction, and collects relevant runtime data in relation to quality-of-service guarantees defined in the cloud service’s SLA (2A).

The data collected is stored in a repository (2B) that is part of the SLA management system (3).

Queries can be issued and reports can be generated for an external cloud resource administrator via a usage and administration portal (4) or for an internal cloud resource administrator via the SLA management system’s native user-interface (5).

The SLA monitor mechanism is used to specifically observe the runtime performance of cloud services to ensure that they are fulfilling the contractual QoS requirements published in SLAs (Figure 1). The data collected by the SLA monitor is processed by an SLA management system to be aggregated into SLA reporting metrics. This system can proactively repair or failover cloud services when exception conditions occur, such as when the SLA monitor reports a cloud service as “down.”