BITS Pilani

Cloud Computing

SEWP ZG527

Agenda

- Multi Tenency Recap
- SLA Management
- Introduction to SLA
- Types of SLA
- SLO & SLA
- SLA Life Cycle

innovate achieve lead

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Recap

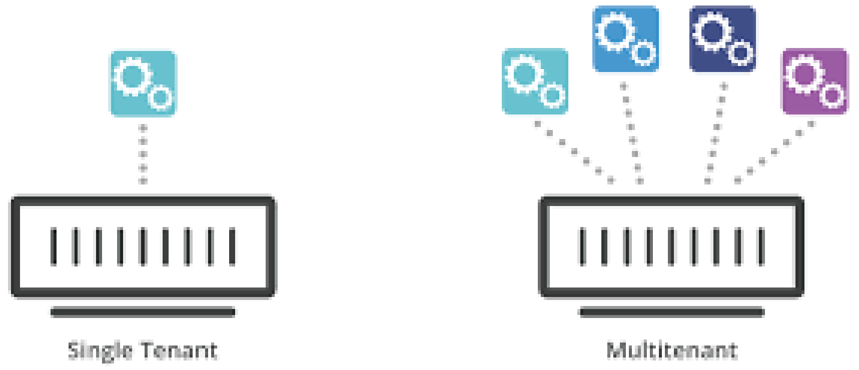
What is Multi Tenancy?

Multi-tenancy is an architectural pattern

A single instance of the software is run on the service provider's infrastructure

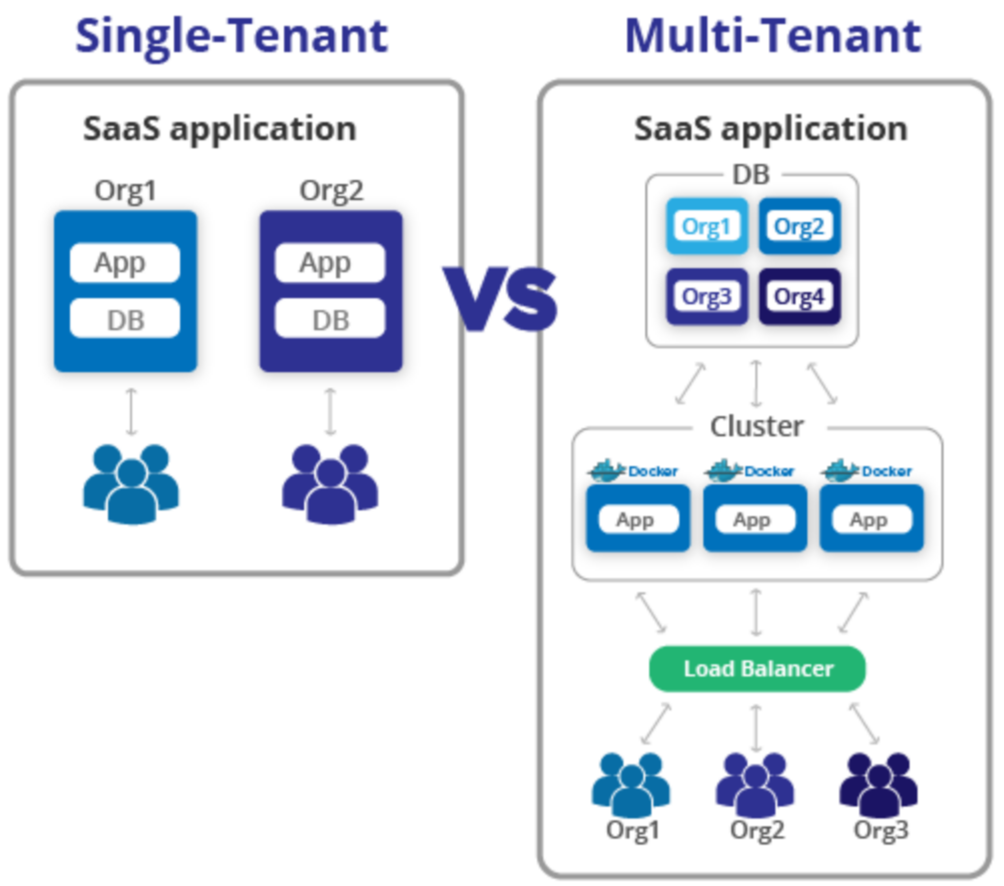
Multiple tenants access the same instance.

In contrast to the multi-user model, multi-tenancy requires customizing the single instance according to the multi-faceted requirements of many tenants.



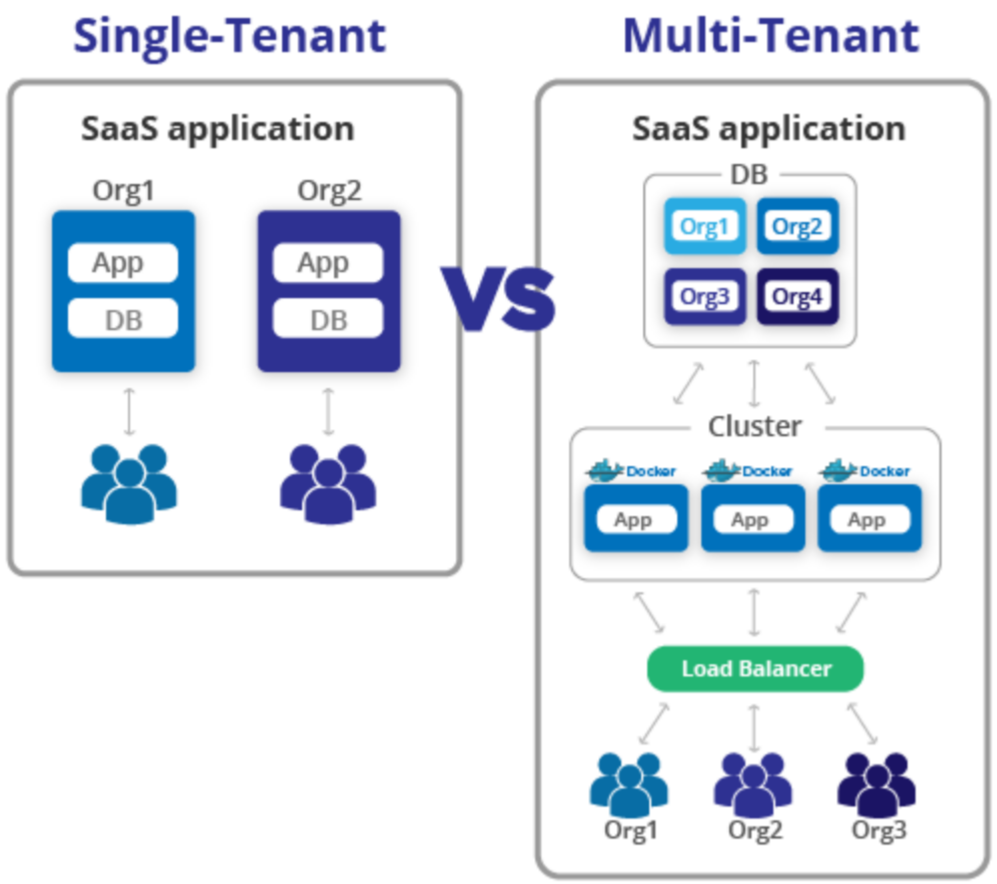
Some Facts

- In the multi tenant architecture, the application is redesigned to handle the resource sharing between the multiple tenants.
- For example, SalesForce.com (service provider) hosts the CRM application using their infrastructure.
- A company who wants to use this hosted CRM application for their business is the customer and the employees of the companies to whom the company provides privileges to access the CRM application are the actual users of the application

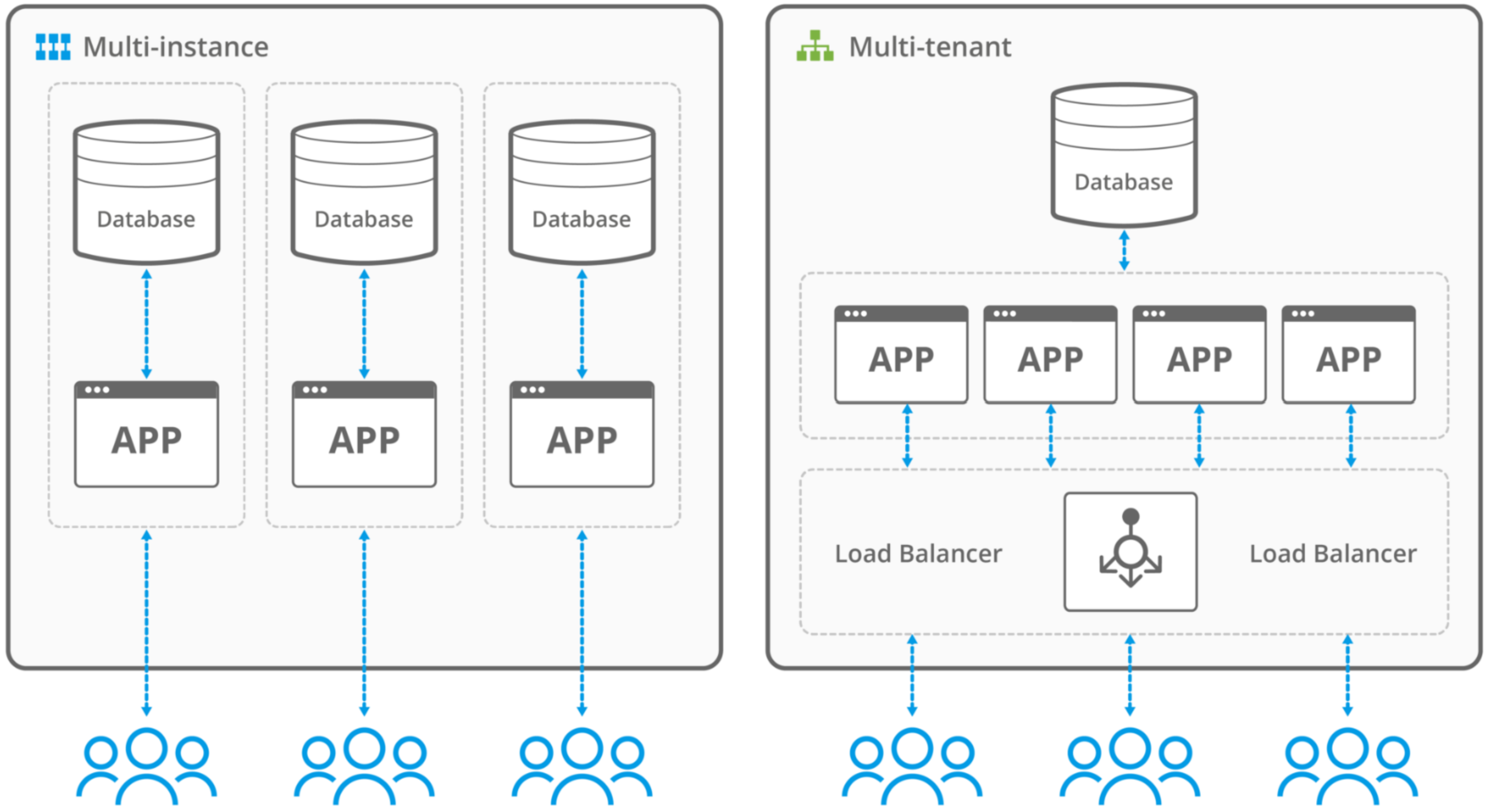


Some Facts

- With this **architecture**, data, configuration, user management, tenant specific functionality etc are shared between the tenants.
- MT contrasts with multi-instance architectures, where separate software instances operate on behalf of different tenants.
- In virtualization, the user is given the illusion that he owns the complete infrastructure on which application is running through concept of virtual machine.
- The hypervisor plays important role to achieve the separation between the multiple users.



Multi Tenant vs Multi Instance



MT- Different Levels

Custom instances

- Lowest level of MT
- Each customer has own custom version of application
- Different versions of application are running differently
- Extremely difficult to manage as needs dedicated support for each customer

Configurable instances

- Same version of application is shared between the customers with customizations specific to each customer
- Different instances of same application are running
- Supports customization like logos on the screen, tailor made workflows
- Managing application is better that custom instances approach as only one copy needs to be managed
- Upgrades are simple and seamless

MT- Different Levels

Configurable, multi-tenant efficient instances

- Same version of application is shared between the customers through a single instance of application
- More efficient usage of the resources
- Management is extremely simple

Scalable, configurable, multi tenant efficient resources

- All features of level 3 are supported.
- Application instances are installed on cluster of computers allowing it to scale as per demand.
- Maximum level of resource sharing is achieved.
- Example, Gmail

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

SLA Management

What is SLA or Service Level Agreement

- Describes a set of non functional requirements of the service.
- Example : RTO time – Return to Operation Time if case of failure

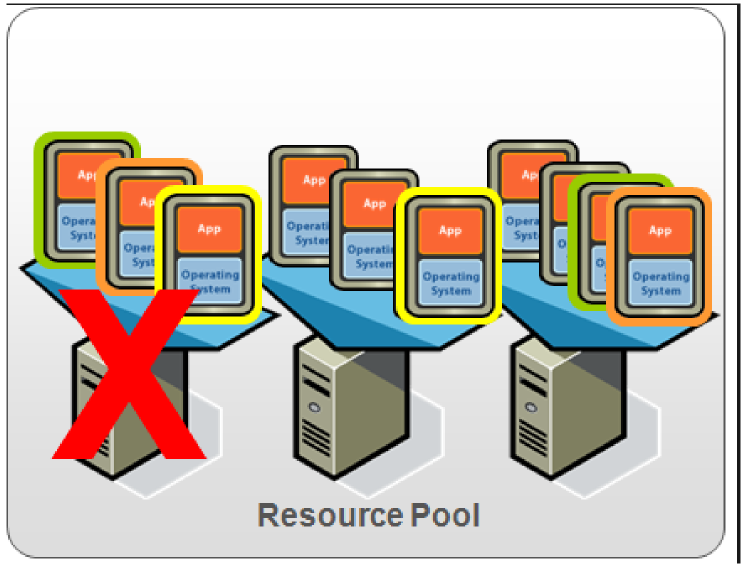
- SLO – Service Level Objective. That is, the objective to be achieved.
- KPI – Key Performance Indicators
- **Service Level Objective:**
- Objective of service quality that has to be achieved.
- Set of measurable KPIs with thresholds to decide if the objective is fulfilled or not.
  - Attainable
  - Repeatable
  - Measurable
  - Understandable
  - Meaningful
  - Controllable
  - Affordable
  - Mutually acceptable

SLA Role in High Availability

- **What is High Availability**
- Driven by SLA (Service Level Agreement)
- High Availability must conform to SLA. Goal here is to meet promised quality
- Examples: Service is available 99.95%.
  - Net Banking : Guarantees banking 24 x7 . There are published down times called " SYSTEM MAINTENANCE" window
  - Duronto Express : Promises point to point service with only Service halts

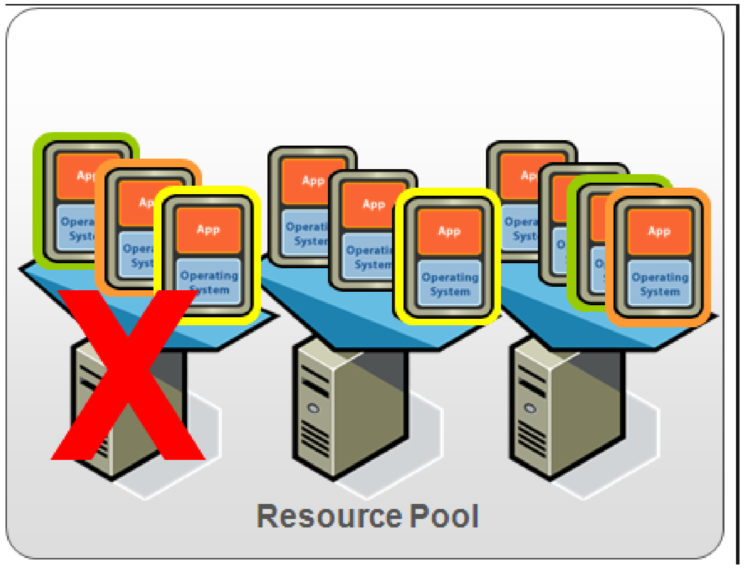
SLA Role in High Availability

- **What deters HA**
- Service outage which was unplanned because of Server or Human Error
- Service Disruption by Planned Maintenance windows (Downtime)
- Service Performance degradation due to lack of infrastructure or failure of critical components during peak load time
- Bottom Line : Need effective Capacity Planning to meet promised QoS or in other words maintain HA thus adhering to the SLA’s

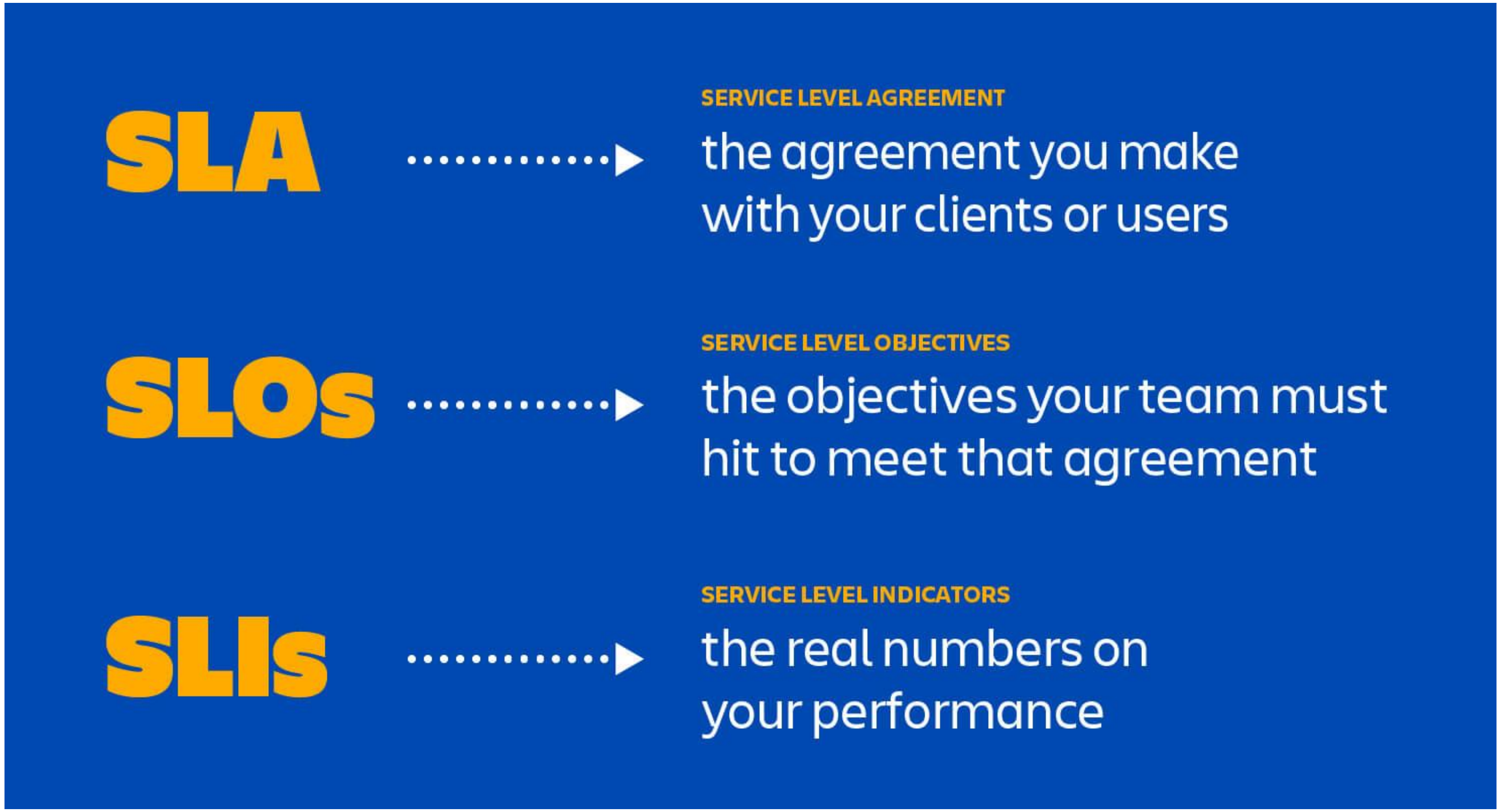


Steps for HA

- <u>**Steps to achieve high availability**</u>
- **Build for server failure**
  - Have redundant servers which can be made online
- **Build for zone failure**
  - Having DR sites in case of failure to switch over to backup site
- **Build for Cloud failure**
  - Plan for your Cloud setup to be robust and contained. Errors should not cascade, having fire door policy to contain threats or errors which affect the ecosystem
- **Automating and testing**
  - Test & Test again, low manual interference



The 3 Initialisms



SLA → SERVICE LEVEL AGREEMENT
the agreement you make with your clients or users

SLOs → SERVICE LEVEL OBJECTIVES
the objectives your team must hit to meet that agreement

SLIs → SERVICE LEVEL INDICATORS
the real numbers on your performance

SLA vs SLO

- The SLA is the entire agreement that specifies
- what service is to be provided,
- how it is supported,
- times,
- locations,
- costs,
- performance, and
- responsibilities of the parties involved.

SLOs are specific measurable characteristics of the SLA such as

availability,

throughput,

frequency,

response time, or quality.

SLIs are actual measure of the SLO and serves as a benchmark

to compare against the promised SLO

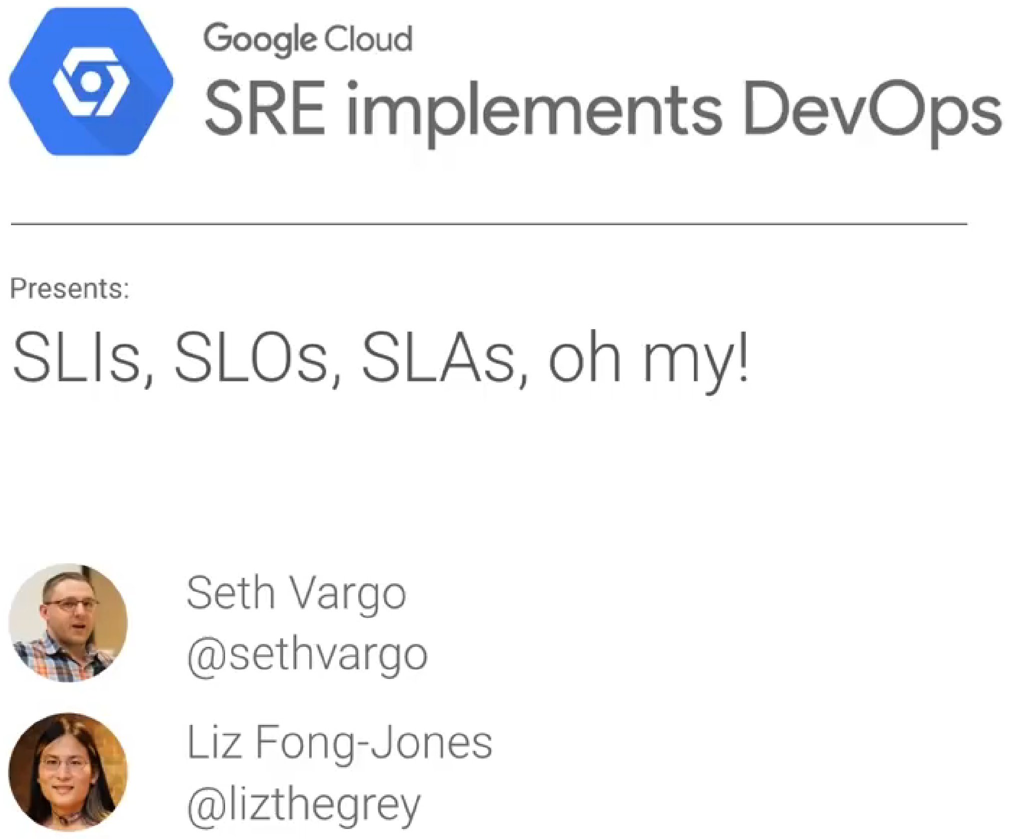
availability,

throughput,

frequency,

response time, or quality.

The 3 Initialisms



Google Cloud

SRE implements DevOps

Presents:

SLIs, SLOs, SLAs, oh my!

Seth Vargo
@sethvargo

Liz Fong-Jones
@lizthegrey

SLA
Promise
Promise
Promise
SLOs
Goal
Goal
Goal
SLIs
How did we do?
How did we do?
How did we do?

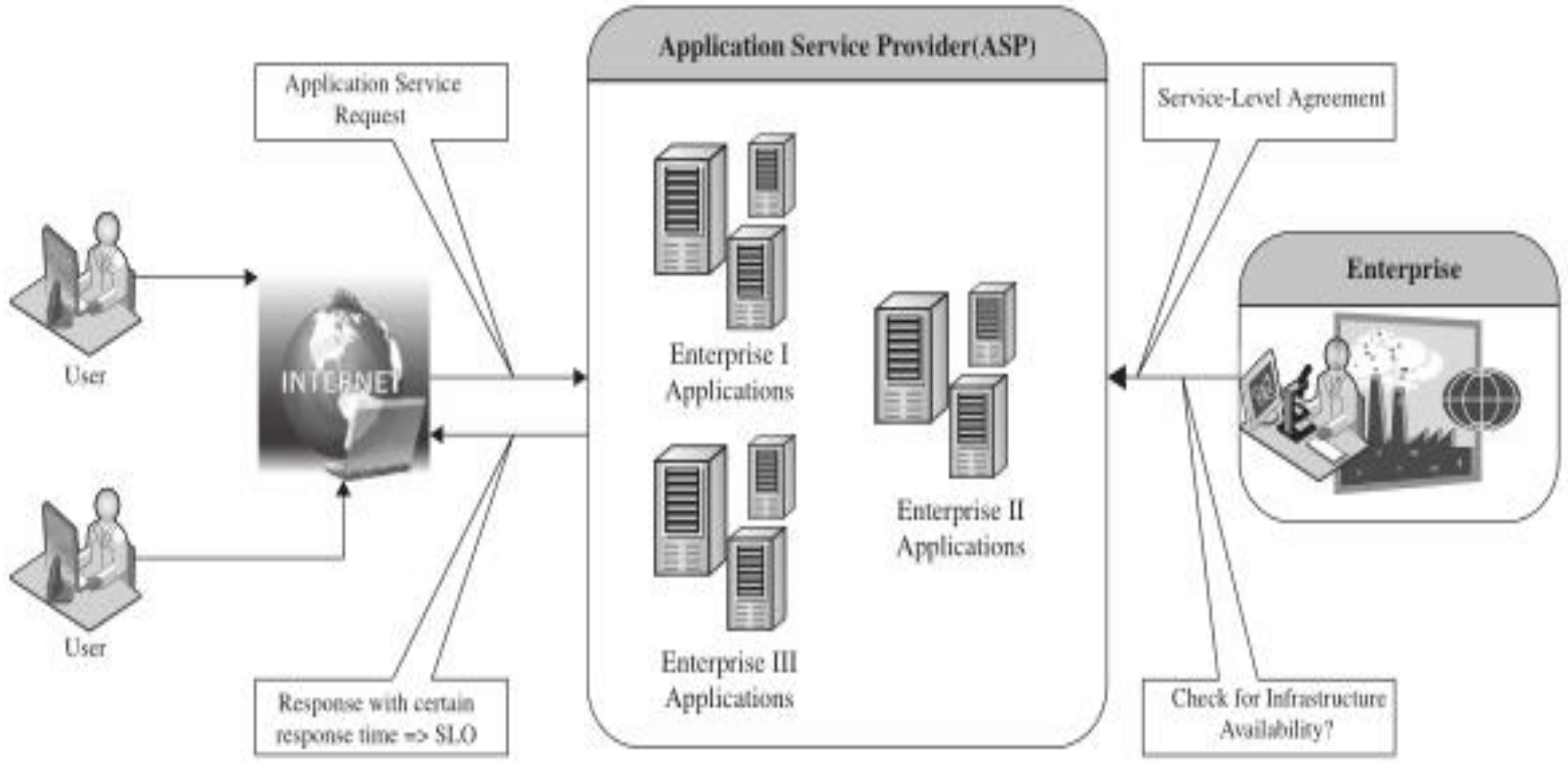
SLA

- In the early days of web-application deployment, **performance of the application** at peak load was a single important criterion for provisioning server resources.
- **Provisioning** in those days involved deciding hardware configuration, determining the number of physical machines, and acquiring them upfront so that the overall business objectives could be achieved.
- The web applications were **hosted** on these dedicated individual servers within enterprises' own server rooms. These web applications were used to provide different kinds of e-services to various clients.

SLA

- Due to the increasing **complexity of managing the huge Data centres**, enterprises started outsourcing the application hosting to the infrastructure providers. They would procure the hardware and make it available for application hosting.
- It necessitated the enterprises to enter into a **legal agreement with the infrastructure service providers** to guarantee a minimum quality of service (QoS).
- Typically, the **QoS parameters** are related to the availability of the system CPU, data storage, and network for efficient execution of the application at peak loads.
- This legal agreement is known as the service-level agreement (SLA)

SLA


Application Service Provider(ASP)
Application Service Request
Service-Level Agreement
User
INTERNET
Enterprise I Applications
Enterprise II Applications
Enterprise III Applications
Enterprise
User
Response with certain response time => SLO
Check for Infrastructure Availability?

Co-Hosting Application

The dedicated hosting practice resulted in massive redundancies within the ASP's data centers due to the underutilization of many of their servers.

This is because the applications were not fully utilizing their servers' capacity at nonpeak loads.

To reduce the redundancies and increase the server utilization in data centers, ASPs started co-hosting applications with complementary work load patterns.

Co-hosting of applications means deploying more than one application on a single server. This led to further cost advantage for both the ASPs and enterprises.

Co-Hosting Application: Issues

However, newer challenges such as **application performance isolation** and **security guarantees have emerged** and needed to be addressed.

Performance isolation implies that one application should not steal the resources being utilized by other co-located applications.

For example, assume that application A is required to use more quantity of a resource than originally allocated to it for duration of time t. For that duration the amount of the same resource available to application B is decreased. This could adversely affect the performance of application B.

Security guaranty: Similarly, one application should not access and destroy the data and other information of co-located applications. Hence, appropriate measures are needed to guarantee security and performance isolation.

Co-Hosting Application: Solution

These challenges prevented ASPs from fully realizing the benefits of co-hosting. Virtualization technologies have been proposed to overcome the above challenges. The ASPs could exploit the containerization features of virtualization technologies to provide performance isolation and guarantee data security to different co-hosted applications.

The applications, instead of being hosted on the physical machines, can be encapsulated using virtual machines.

System resource allocation to these virtual machines can be made in two modes:

(1) Conserving

(2) Nonconserving.

In the **conserving mode**, a virtual machine demanding more system resources (CPU and memory) than the specified quota cannot be allocated the spare resources that are remain un-utilized by the other co-hosted virtual machines.

In the **non-conserving mode** the spare resources that are not utilized by the co-hosted virtual machines can be used by the virtual machine needing the extra amount of resource. If the resource requirements of a virtual machine cannot be fulfilled from the current physical host, then the virtual machine can be migrated to another physical machine capable of fulfilling the additional resource requirements.

Traditional SLO Management Approaches

Load balancing – is to distribute the incoming request onto a set of physical machines, each hosting a replica of an application, so that the load on the machines is equally distributed. Front end node (node facing the client) receives the incoming requests and distributes these requests to different physical machines for further execution. Back end nodes serves the incoming requests.

Class agnostic load balancing – the front end machine are agnostic to nature of request. This means that the front end machine is neither aware of the type of client from which the request originates nor aware of the request category.

Class aware load balancing – The front end node additionally inspect the type of client making the request and/or the type of service requested before deciding which back end node should service the request.

Traditional SLO Management Approaches

Admission Control – These algorithms plays important role in deciding the set of requests that should be admitted into the application server when the server experiences very heavy loads. The objective of admission control mechanisms is to police the incoming requests and identify when the system faces overload situations.

Request based algorithms – reject new requests if the servers are running to their capacity. The disadvantage is that client's session may consist of multiple requests that are not necessarily unrelated. Some requests are rejected even if there are others that are honored.

Session based algorithms – ensure that longer sessions are completed and any new sessions are rejected. Once a session is admitted into the server, all future requests belonging to that session are admitted as well, even though new sessions are rejected by system.

SLA Types

Infrastructure SLA – Infrastructure provider manages and offers guarantees on availability of the infrastructure, namely server machine, power, network connectivity and so on. Enterprises manage their applications that are deployed on these server machines. The machines are leased to customers and are isolated from machines of other customers.

For example, SLAs can be

Hardware availability – 99% uptime in a calendar month

Power availability – 99.99% of the time in calendar month

Data center network availability – 99.99% of the time in calendar month

Application SLA – In application co-hosting model, the server capacity is available to the applications based solely on their resource demands. Hence the service providers are flexible in allocating and de-allocating computing resources among the co-located applications. Therefore the service providers are also responsible for ensuring to meet their customers application SLOs.

For example, SLAs can be

Web site response time (max of 3.5 sec per user request), Latency of web server (max of 0.2 sec per request), Latency of DB (max of 0.5 sec per query)

Infrastructure SLA vs Capacity Management

If temporal behavior of services with respect to resource demands is highly predictable, then capacity can be efficiently scheduled using reservations.

For **less predictable elastic workloads**., **exact scheduling of capacity may not be possible**.

Rather than that, **capacity planning and optimizations are required**.

- IaaS providers perform two complementary management tasks:

  (1) **Capacity planning** to make sure that SLA obligations are met as contracted with the service providers and;

  (2) **Continuous optimization** of resource utilization in specific workload to make the most efficient use of the existing capacity.

Infrastructure SLA

Thus, to deploy a service on a cloud, **a service provider orders suitable virtual hardware** and installs its application software on it.

From the IaaS provider, a given service configuration is a virtual resource array of black box resources, which correspond to the number of instances of resource type.

For example, a typical three-tier application may contain **ten general-purpose small instances** to run Web front-ends, **three large instances** to run an application server cluster with load balancing and redundancy, and **two large instances** to run a replicated database.

A risk mitigation mechanism to protect user experience in the IaaS model is offered by infrastructure SLAs (i.e., the SLAs formalizing capacity availability) signed between service provider and IaaS provider.

Infrastructure SLA

There is **no universal approach to infrastructure SLAs**. As the IaaS field matures and more experience is being gained, some methodologies may become more popular than others. Also some methods may be more suitable for specific workloads than other. There are three main approaches as follows.

No SLAs. This approach is based on two premises:

(a) Cloud always has spare capacity to provide on demand, and

(b) services are not QoS sensitive and can withstand moderate performance degradation. This methodology is best suited for the best effort workloads.

Probabilistic SLAs. (Epistemic) These SLAs allow **us to trade capacity availability for cost of consumption**. Probabilistic SLAs specify clauses that determine availability percentile for contracted resources computed over the SLA evaluation period. **The lower the availability percentile, the cheaper the cost of resource consumption.** This type of SLA is **suitable for small and medium businesses** and **for many enterprise grade applications**.

Deterministic SLAs. (Ontic) These are, in fact, probabilistic SLAs where **resource availability percentile is 100%**. These SLAs are most stringent and difficult to guarantee. From the provider's point of view, they do not admit capacity multiplexing. Therefore this is the most costly option for service providers, which may be applied for critical services.

SLA Life Cycle

Each SLA goes through a sequence of steps starting from identification of terms and conditions, activation and monitoring of the stated terms and conditions, and eventual termination of contract once the hosting relationship ceases to exist. Such a sequence of steps is called SLA life cycle and consists of the following five phases:

1. Contract definition
2. Publishing and discovery
3. Negotiation
4. Operationalization
5. De-commissioning

SLA Life Cycle

Contract Definition: Generally, service providers define a set of service offerings and corresponding SLAs using standard templates. These service offerings form a catalog. Individual SLAs for enterprises can be derived by customizing these base SLA templates.

Publication and Discovery. Service provider advertises these base service offerings through standard publication media, and the customers should be able to locate the service provider by searching the catalog. The customers can search different competitive offerings and shortlist a few that fulfill their requirements for further negotiation.

Negotiation: Once the customer has discovered a service provider who can meet their application hosting need, the SLA terms and conditions needs to be mutually agreed upon before signing the agreement for hosting the application. For a standard packaged application which is offered as service, this phase could be automated.

For customized applications that are hosted on cloud platforms, this phase is manual. The service provider needs to analyze the application's behavior with respect to scalability and performance before agreeing on the specification of SLA. At the end of this phase, the SLA is mutually agreed by both customer and provider and is eventually signed off. SLA negotiation can utilize the WS-negotiation specification

SLA Life Cycle

Operational: SLA operation consists of SLA monitoring, SLA accounting, and SLA enforcement.

SLA monitoring involves measuring parameter values and calculating the metrics defined as a part of SLA and determining the deviations.

On identifying the deviations, the concerned parties are notified. SLA Accounting involves capturing and archiving the SLA adherence for compliance.

As part of accounting, the application’s actual performance and the performance guaranteed as a part of SLA is reported.

De-commissioning : SLA decommissioning involves termination of all activities performed under a particular SLA when the hosting relationship between the service provider and the service consumer has ended.

SLA specifies the terms and conditions of contract termination and specifies situations under which the relationship between a service provider and a service consumer can be considered to be legally ended

SLA Life Cycle

- From the SLA perspective there are multiple challenges for provisioning the infrastructure on demand. These includes -
- The application if a black box to CSP (Cloud Service Provider) and the CSP have virtually no knowledge about the application runtime characteristics. CSP needs to determine the right amount of computing resources required for different components of application at various workloads.
- CSP needs to understand the performance bottlenecks and the scalability of the application.
- CSP analyses the application before it goes live. However, subsequent operations by customers to their applications or auto updates beside others can impact performance of the applications, thereby making the applications SLA at risk.
- The risk of capacity planning is with service provider instead of customer.

SLA LC Management – Cloud Applications

SLA management of applications hosted on cloud platforms involves five phases.

1. Feasibility
2. On-boarding
3. Pre-production
4. Production
5. Termination

SLA Cloud Application LC Management

Feasibility Analysis

Managed Service Providers(MSP) conducts the feasibility study of hosting an application on their cloud platforms.

This study involves three kinds of feasibility:

(a) **Technical feasibility**,

(b) **Infrastructure feasibility**

(c) **Financial feasibility**.

The technical feasibility of an application implies determining the following:

1. Ability of an application to scale out.
2. Compatibility of the application with the cloud platform being used within the MSP's data center.
3. The need and availability of a specific hardware and software required for hosting and running of the application.
4. Preliminary information about the application performance and whether they can be met by the MSP.

SLA Cloud Application LC Management

Performing the infrastructure feasibility involves determining the availability of infrastructural resources in sufficient quantity so that the projected demands of the application can be met.

The financial feasibility study involves determining the approximate cost to be incurred by the MSP and the price the MSP charges the customer so that the hosting activity is profitable to both of them.

A feasibility report consists of the results of the above three feasibility studies. The report forms the basis for further communication with the customer. Once the provider and customer agree upon the findings of the report, the outsourcing of the application hosting activity proceeds to the next phase, called “on boarding” of application.

Only the basic feasibility of hosting an application has been carried in this phase. However, the detailed runtime characteristics of the application are studied as part of the on-boarding activity

SLA Cloud Application LC Management

On-boarding of Application:

Once the customer and the MSP agree in principle to host the application based on the findings of the feasibility study, the application is moved from the customer servers to the hosting platform.

Moving an application to the MSP's hosting platform is called on-boarding.

As part of the on-boarding activity, the MSP understands the application runtime characteristics using runtime profilers*.

*** Profiling is a method of gathering performance data in any development or deployment scenario. This is for developers and system administrators who want to gather information about application performance**

SLA Cloud Application LC Management

On-boarding :

Packing of the application for deploying on physical or virtual environments. Application packaging is the process of creating deployable components on the hosting platform (could be physical or virtual).Open Virtualization Format (OVF) standard is used for packaging the application for cloud platform.

The packaged application is executed **directly on the physical servers** to capture and analyze the application performance characteristics. It allows **the functional validation of customer's application**. Besides, it provides a **baseline performance value** for the application in **non virtual environment**.

This can be used as **one of the data points for customer's performance expectation** and for **application SLA**. Additionally, it helps to identify the nature of application—that is, whether it is CPU-intensive or I/O intensive or network-intensive and the potential performance bottlenecks.

SLA Cloud Application LC Management

On-boarding :

The application is **executed on a virtualized platform** and the application performance characteristics are **noted again**. Important performance characteristics like the application's **ability to scale (out and up) and performance bounds** (minimum and maximum performance) are noted.

Based on the measured performance characteristics, **different possible SLAs are identified**. The **resources required** and the costs involved for each SLA are also computed.

Once the customer agrees to the set of SLAs and the cost, the MSP starts creating different policies required by the data center for automated management of the application. This implies that the management system should automatically infer the amount of system resources that should be allocated/de-allocated to/from appropriate components of the application when the load on the system increases/decreases.

SLA Cloud Application LC Management

Pre-Production

Once the determination of policies is completed as discussed in previous phase, the application is hosted in a simulated production environment.

It facilitates the customer to verify and validate the MSP’s findings on application’s runtime characteristics and agree on the defined SLA. Once both parties agree on the cost and the terms and conditions of the SLA, the customer sign-off is obtained. On successful completion of this phase the MSP allows the application to go on-live.

SLA Cloud Application LC Management

Production

In this phase, the application is made accessible to its end users under the agreed SLA.

However, there could be situations when the managed application tends to behave differently in a production environment compared to the preproduction environment.

This in turn may cause sustained breach of the terms and conditions mentioned in the SLA. Additionally, customer may request the MSP for inclusion of new terms and conditions in the SLA.

If the application SLA is breached frequently or if the customer requests for a new non-agreed SLA, the on-boarding process is performed again. In the case of the former, on-boarding activity is repeated to analyze the application and its policies with respect to SLA fulfillment. In case of the latter, a new set of policies are formulated to meet the fresh terms and conditions of the SLA.

SLA Cloud Application LC Management

Termination

When the customer wishes to withdraw the hosted application and does not wish to continue to avail the services of the MSP for managing the hosting of its application, the termination activity is initiated.

On initiation of termination, all data related to the application are transferred to the customer and only the essential information is retained for legal compliance. This ends the hosting relationship between the two parties for that application, and the customer sign-off is obtained.

Q & A……..

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Credits

•Hwang, Kai; Dongarra, Jack; Fox, Geoffrey C.. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things (Kindle Locations 3532-3533). Elsevier Science. Kindle Edition.