

Horizontal Scaling vs Vertical Scaling & Multi-Tenancy

In this section we will learn about various Cloud Concepts:- a) Scaling on the cloud b) Degradation of Services c) Availability Vs Durability on the Cloud d) Single & Multi-tenancy Applications e) Types of Cloud Deployments

In this section we will learn various Cloud Concepts.

1. Scaling on the cloud
2. Degradation of Services
3. Availability Vs Durability on the Cloud
4. Single & Multi-tenancy Applications
5. Types of Cloud Deployments

Scaling Out / Horizontal Scaling vs Scaling up/ Vertical Scaling

Horizontal scaling means that you scale by adding more machines into your pool of resources. Eg. As traffic goes up you add more web servers to take on the traffic.

Vertical scaling means that you scale by adding more power (CPU, RAM) to an existing machine.

On the Cloud you Scale horizontally !

Graceful Degradation of Services

Applications need to have a way to handle excessive load and handle failure gracefully no matter if it's the application or a dependent service under load.

The Site Reliability Engineering handbook describes graceful degradation in applications as offering “responses that are not as accurate as or that contain less data than normal responses, but that are easier to compute” when under excessive load.

Modern applications are build with breakers that do not overload other microservices if they malfunction. They also need to have sufficient telemetry that helps in troubleshooting and help in identifying what is causing a

problem.

If you take a moment to think of all the apps on your phone that you use when was the last time it had an outage ? The probability of an outage is so less likely because the system design and architecture of modern applications take into account every single point of failure from an application, database, Load balancer and even the infrastructure.

Graceful failure is about making sure that at the event of failure at any level that there is a backup that takes the responsibility of the component that failed and at the end there is no perceived customer impact. Note the word 'perceived', Lets take for example that a large ecommerce / retail website calculates the tax for each shipment before you place the order. If the tax microservice goes down then maybe you do not charge the customer tax for that one transaction the cost will be covered by the organization.

Availability Vs Durability on the Cloud

Availability

Availability and durability are two very different aspects of data accessibility. Most cloud services offer a 99.9999% availability. This varies by provider and by each service that is offered by the provider.

Availability refers to system uptime, i.e. the storage system is operational and can deliver data upon request. Historically, this has been achieved through hardware redundancy so that if any component fails, access to data will prevail.

Durability

Durability, on the other hand, refers to long-term data protection, i.e. the stored data does not suffer from bit rot, degradation or other corruption. Rather than focusing on hardware redundancy, it is concerned with data redundancy so that data is never lost or compromised.

Availability and durability serve different objectives. For data centers, availability/uptime is a key metric for operations as any minute of downtime is costly. The measurement focuses on storage system availability. But what happens when a component, system or even the data center goes down? Will your data be intact when the fault is corrected?

This illustrates the equal importance of data durability. When an availability fault is corrected, it is essential that access to uncorrupted data is restored. With the explosion of data created, the potential of mining, and growing needs for longer retention rates (for everything) you can imagine how this is paramount for business success.

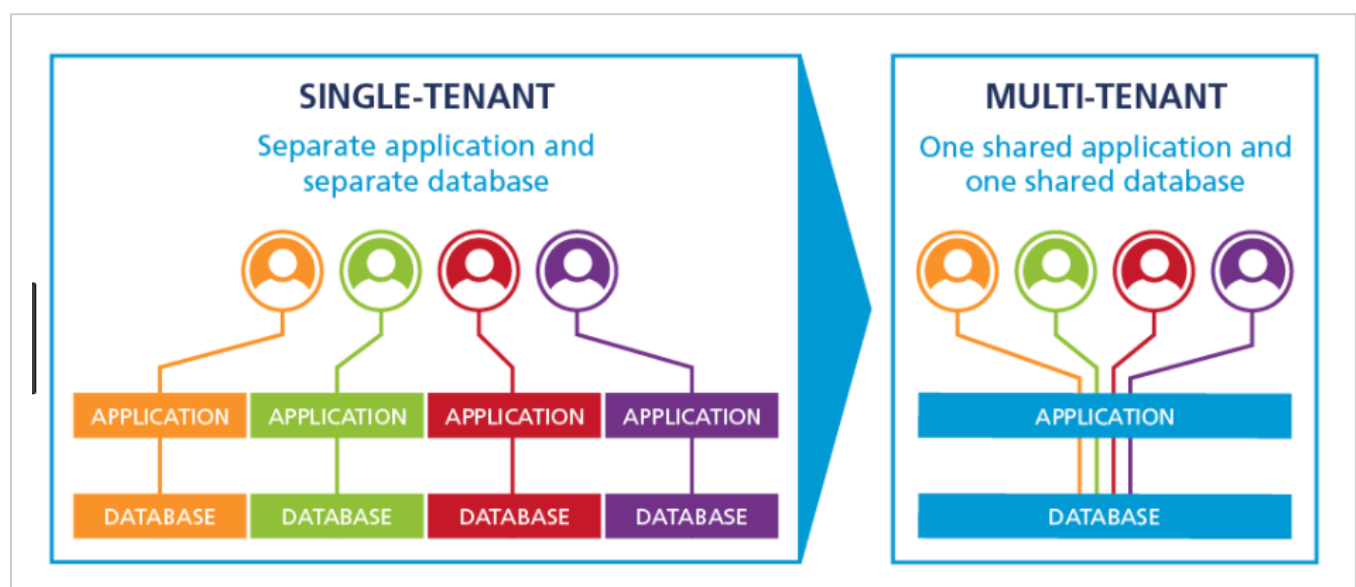
Consider the potential competitive, financial or even legal impact of not being able to retrieve the archived master/reference copy of data. Hence, both data availability and data durability are essential for short- and long-term business success.

Single vs Multi Tenancy Cloud Deployment

Single tenancy or multi-tenancy is in context with SaaS applications. A SaaS application can either choose to be one to the other. Most modern applications choose to be multi-tenant applications.

Multi-tenancy

Multi-tenancy means that a single instance of the software and all of the supporting infrastructure serve multiple end-customers. Each customer shares the software application and also shares a single database. The data is tagged in the database as belonging to one customer or another, and the software is smart enough to know who the data belongs to.



Single tenancy

In a Single tenancy approach a single instance of the software and all of the

supporting infrastructure serves a single customer. With single tenancy, each customer has their own independent database and instance of the application software as well. With this option, there's essentially no sharing going on. Everyone has their own, separate from everyone else.

Think about a SaaS application that is serving customers like coke and pepsi, sometimes these companies would care that they are not sharing the database tier in particular so that there is no possibility for information to leak. If this was a concern then they would go for a Single tenancy rather than a multi tenancy approach.

With single-tenant architectures, the marginal cost of adding new customers never goes down. Maintenance With single-tenant, maintenance is incremental. If you botch it, you typically only take down a single team. Instead of deploying a single app update, you're deploying N app updates. Instead of migrating one database, you're migrating N databases. On the surface level it sounds like this would create more work for you, but since the systems are isolated and identical most of that work can be automated. All you need is a bit of tooling to orchestrate the updates.

Resilience

Single-tenant creates resilience at the system level. Outside of DDOS attacks at the DNS level there are very few ways to take down the entire system. A team may experience problems, but it's unlikely those problems will extend beyond that single instance.

Types of Cloud Deployments

Hybrid cloud is an environment that uses a mix of on-premises, private cloud and /or third-party, public cloud services with orchestration between the two platforms.

Multi-cloud

A multi-cloud strategy is the use of two or more cloud services. While a multi-cloud deployment can refer to any implementation of multiple software as a service (SaaS) or platform as a service (PaaS) cloud offerings, today, it generally refers to a mix of public infrastructure as a service (IaaS) environments, such as Amazon Web Services, Microsoft Azure, Oracle Cloud

Infrastructure.

Multi-cloud was, and still is, seen as a way to prevent data loss or downtime due to a localized component failure in the cloud. The ability to avoid vendor lock-in was also an early driver of multi-cloud adoption. A multi-cloud strategy also offers the ability to select different cloud services or features from different providers.

The pros and cons of a multi-cloud strategy

There are several commonly cited advantages to multi-cloud computing, such as the ability to avoid vendor lock-in, the ability to find the optimal cloud service for a particular business or technical need and increased redundancy.

However, there are some potential drawbacks. For example, most public cloud providers offer volume discounts, where prices are reduced as customers buy more of a particular service. It becomes more difficult for an organization to qualify for those discounts when it doesn't concentrate its business with a single cloud provider.

In addition, a multi-cloud deployment requires an IT staff to have multiple kinds of cloud platform or provider expertise. Workload or application management in multi-cloud computing can also be a challenge, as information moves from one cloud platform to another.

Multi-cloud computing vs. Hybrid cloud computing

Multi-cloud and hybrid cloud computing are similar, but different IT infrastructure models. In general, hybrid cloud refers to a cloud computing environment that uses a mix of an on-premises, private cloud and a third-party, public cloud, with orchestration between the two. An enterprise often adopts hybrid cloud to achieve a specific task, such as the ability to run workloads in house, and then burst into the public cloud when compute demands spike.

However, multi-cloud doesn't preclude hybrid cloud, and a hybrid cloud could be part of a multi-cloud deployment. The two models are not an either/or situation; it simply depends on what a business hopes to achieve.

Cloud computing deployment models

Private	Hybrid	Public
<p>A cloud computing model in which an enterprise uses a proprietary architecture and runs cloud servers within its own data center.</p> <p>CHARACTERISTICS:</p> <ul style="list-style-type: none">Single-tenant architectureOn-premises hardwareDirect control of underlying cloud infrastructure <p>TOP VENDORS:</p> <ul style="list-style-type: none">HPE, VMware, Dell EMC, IBM, Red Hat, Microsoft, OpenStack	<p>A cloud computing model that includes a mix of on-premises, private cloud and third-party public cloud services with orchestration between the two platforms.</p> <p>CHARACTERISTICS:</p> <ul style="list-style-type: none">Cloud bursting capabilitiesBenefits of both public and private environments <p>TOP VENDORS:</p> <ul style="list-style-type: none">A combination of both public and private cloud providers	<p>A cloud computing model in which a third-party provider makes compute resources available to the general public over the internet. With public cloud, enterprises do not have to set up and maintain their own cloud servers in house.</p> <p>CHARACTERISTICS:</p> <ul style="list-style-type: none">Multi-tenant architecturePay-as-you-go pricing model <p>TOP VENDORS:</p> <ul style="list-style-type: none">AWS, Microsoft Azure, Google Cloud Platform

The basic benefits of multi-tenant are:

Resilience

Both architectures offer their own form of resilience. Multi-tenant creates resilience at the team level. Each team is serviced by multiple instances, spread across multiple regions/zones, and hosted behind load balancers. A team is unlikely to experience issues unless there is a system-wide outage.

Maintenance

With multi-tenant, deploys are typically all or nothing. Maintenance on multi-tenant systems can be scary. You push out a single update, and every customer is immediately on the new system. If you botch it you take down the entire system.

Easy Upgrades

Instead of a vendor needing to update every instance of their software across a large number of servers, they are able to update a single, central application or codebase and have the changes instantly available to all users. With a multi-tenant application the process for spinning up a new cloud and application on behalf of a new customer is incredibly easy and can be done very quickly.

With multi-tenancy based applications you can provide an additional layer to allow for customizations while still maintaining an underlying codebase which remains constant for all users, including of course, all new customers.

Ongoing Cost Savings

Multi-tenancy speeds up upgrades, saves time (and also cost) but in addition the server / cloud requirements for a multi-tenancy application cost much less. The opportunity to save money takes many forms and becomes greater as the application scales up. This reduces the cost of doing business for the vendor and savings can be passed onto customers.