Cloud Infrastructure(M-3)

Aradhana Behura

Communication & Computing Group

Department of CSE

Email: aradhana.behurafcs@kiit.ac.in,

Scalability and Elasticity in Cloud Computing

- **Scalability** is the ability of the system to accommodate larger loads just by adding resources either making hardware stronger (**scale up**) or adding additional nodes (**scale out**).
- **Elasticity** is the ability to fit the resources needed to cope with loads dynamically usually in relation to scale out

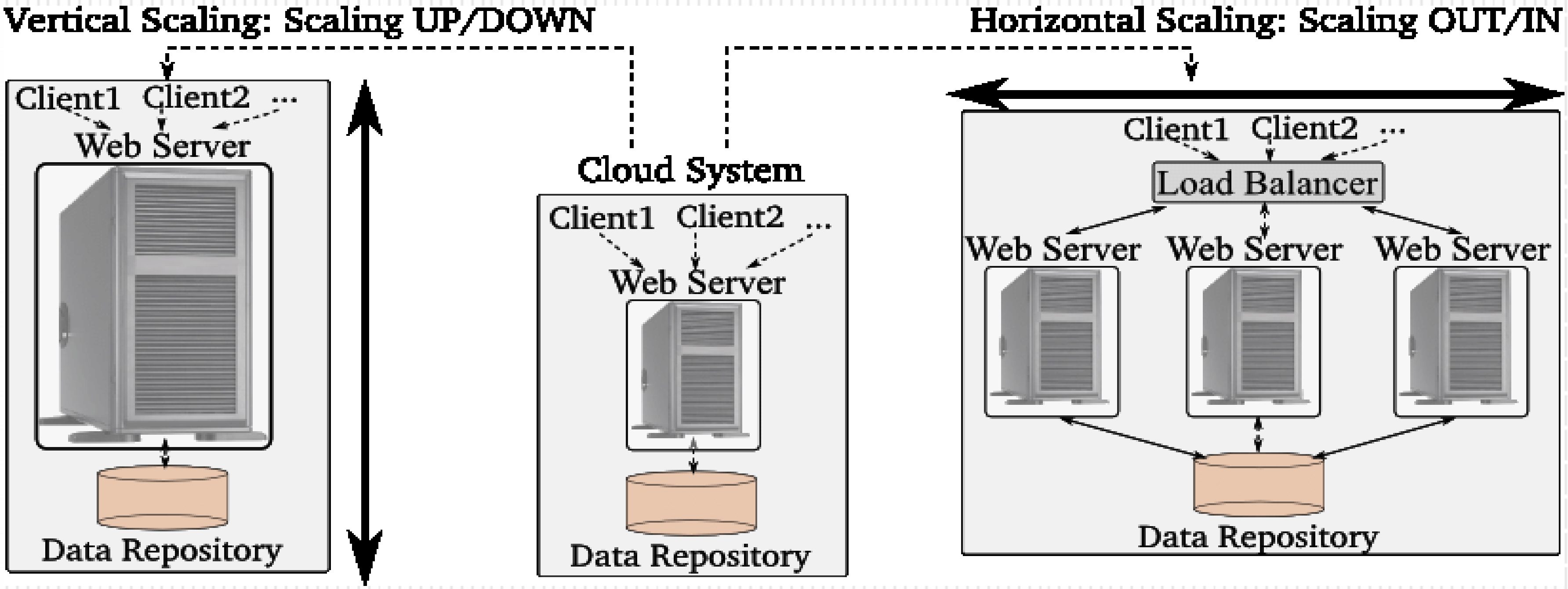
Scalability in cloud computing

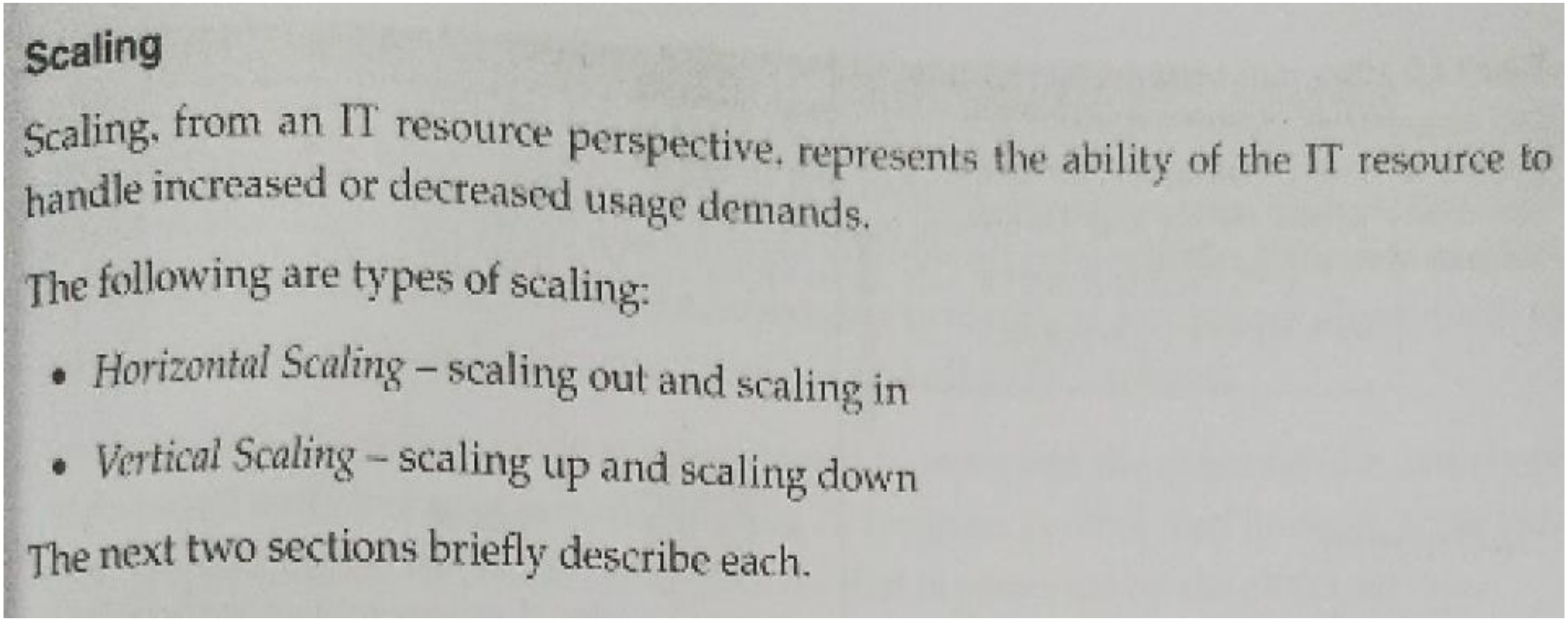
- **Scalability is** the property of a system to handle a growing amount of work by adding resources to the system.
- **It** is the ability to quickly and easily increase or decrease the size or power of an IT solution.
- A **scalable cloud** is why you can sign up and use most **cloud** solutions in just a few minutes – if not seconds. It's why you can add resources like storage to an existing account just as quickly.

Characteristics of a truly scalable application:

- Increasing resources results in a proportional increase in performance
- A scalable service is capable of handling heterogeneity
- A scalable service is operationally efficient
- A scalable service is resilient
- A scalable service should become more cost effective when it grows (Cost per unit reduces as the number of units increases)

Three types of scalability : Vertical, Horizontal and Diagonal





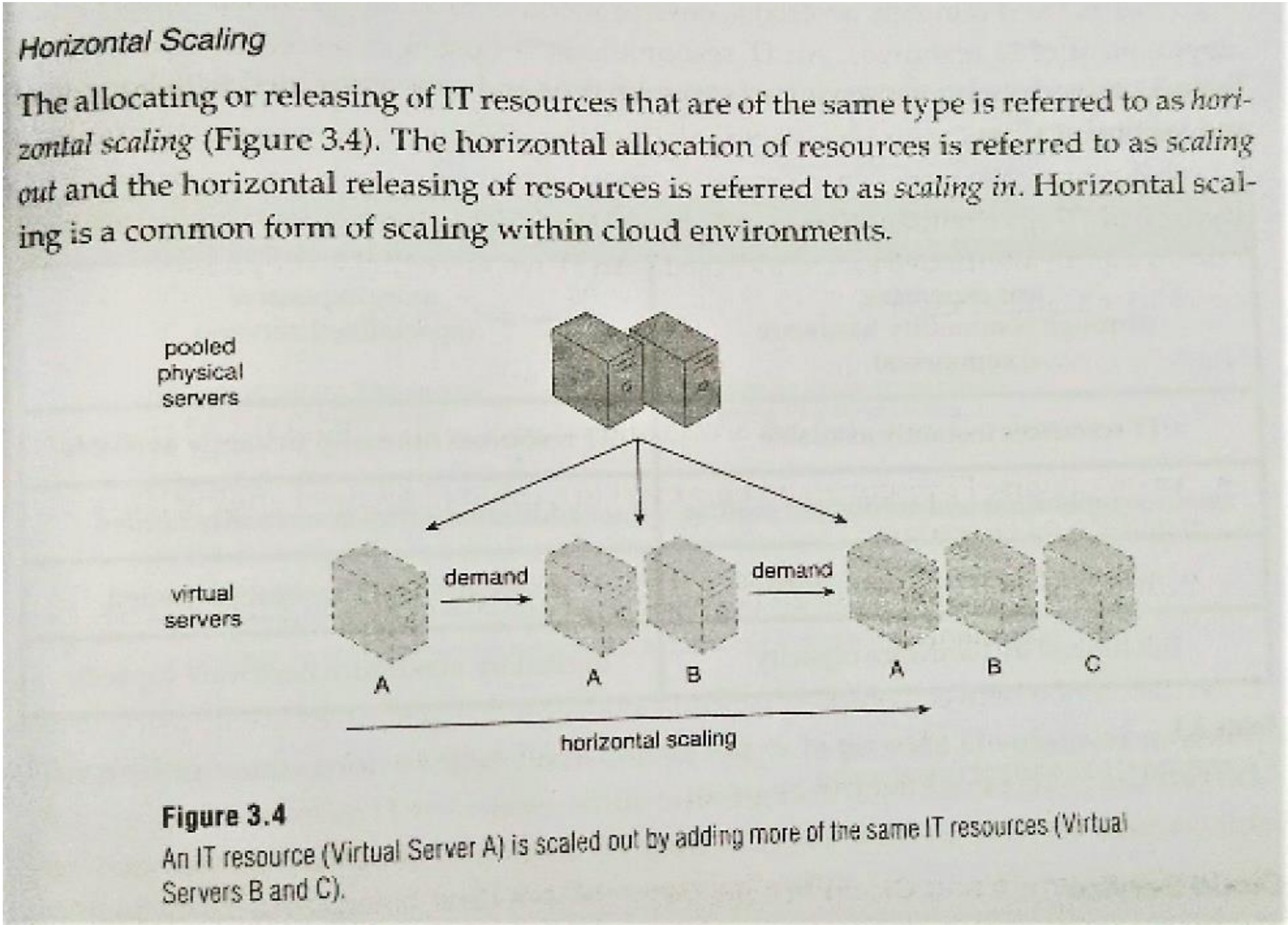
Scaling

Scaling, from an IT resource perspective, represents the ability of the IT resource to handle increased or decreased usage demands.

The following are types of scaling:

- *Horizontal Scaling* – scaling out and scaling in
- *Vertical Scaling* – scaling up and scaling down

The next two sections briefly describe each.



Horizontal Scaling

The allocating or releasing of IT resources that are of the same type is referred to as *horizontal scaling* (Figure 3.4). The horizontal allocation of resources is referred to as *scaling out* and the horizontal releasing of resources is referred to as *scaling in*. Horizontal scaling is a common form of scaling within cloud environments.

pooled
physical
servers
virtual
servers
demand
demand
A
A
B
A
B
C
horizontal scaling

Figure 3.4
An IT resource (Virtual Server A) is scaled out by adding more of the same IT resources (Virtual Servers B and C).

Contd.

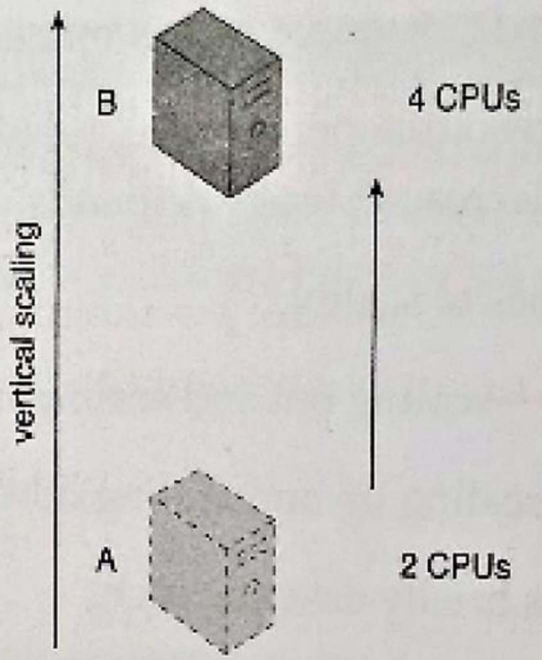
- The allocating or releasing of IT resources that are of the same type is referred to as horizontal scaling (fig 3.4).
- The horizontal allocation of resources is referred to as scaling out & the horizontal releasing of resources is referred to as scaling in.

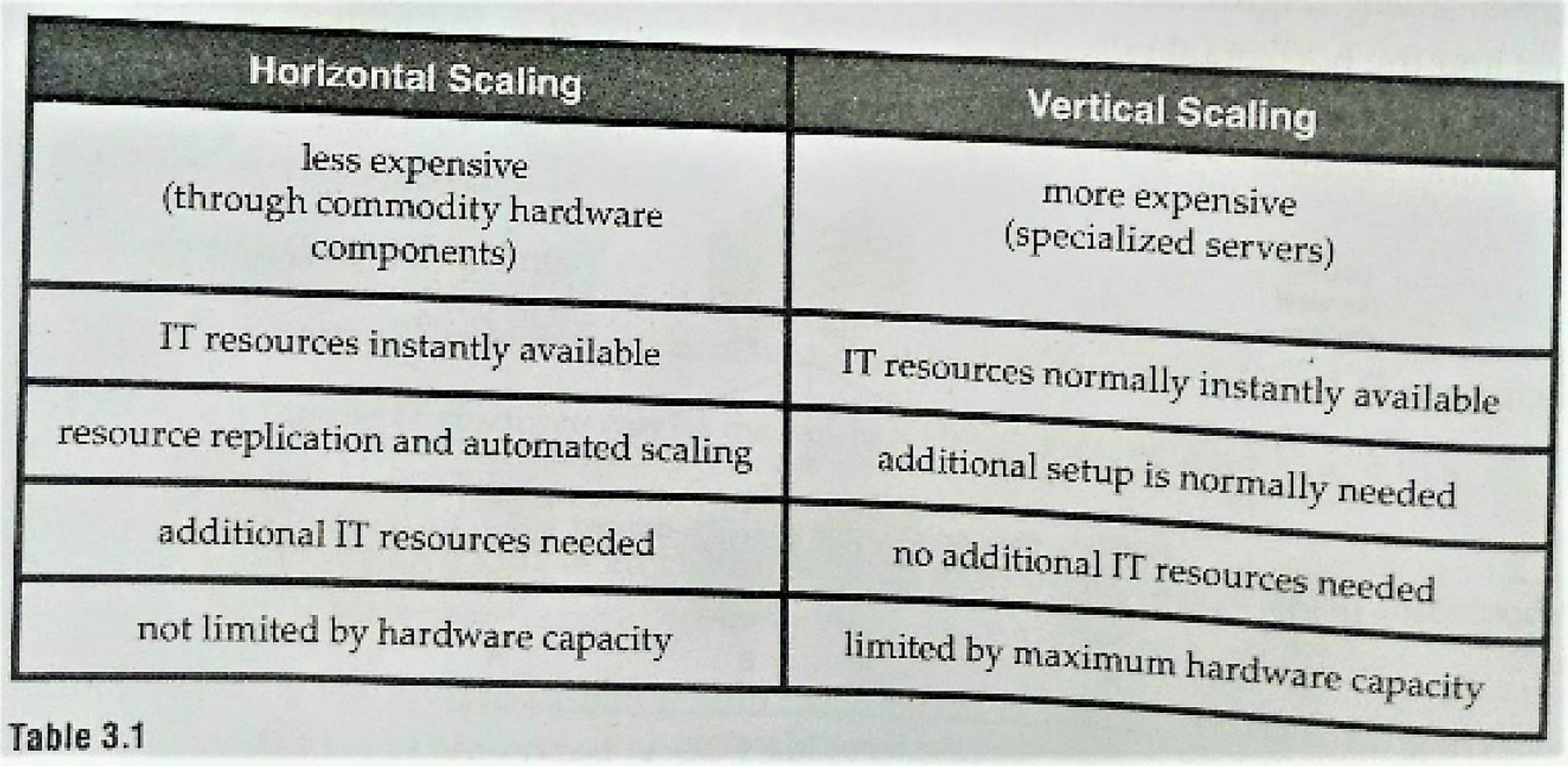
Vertical Scaling

- When an existing IT resource is replaced by another with higher or lower capacity, vertical scaling is considered to have occurred (Fig.3.5).
- Specifically, the replacing of an IT resource with another that has a higher capacity is referred to as scaling up & the replacing an IT resource with another that has a lower capacity is considered scaling down.
- Vertical scaling is less common in cloud environments due to the downtime required while the replacement is taking place.

Figure 3.5

An IT resource (a virtual server with two CPUs) is scaled up by replacing it with a more powerful IT resource with increased capacity for data storage (a physical server with four CPUs).





Horizontal Scaling	Vertical Scaling
less expensive (through commodity hardware components)	more expensive (specialized servers)
IT resources instantly available	IT resources normally instantly available
resource replication and automated scaling	additional setup is normally needed
additional IT resources needed	no additional IT resources needed
not limited by hardware capacity	limited by maximum hardware capacity

Table 3.1

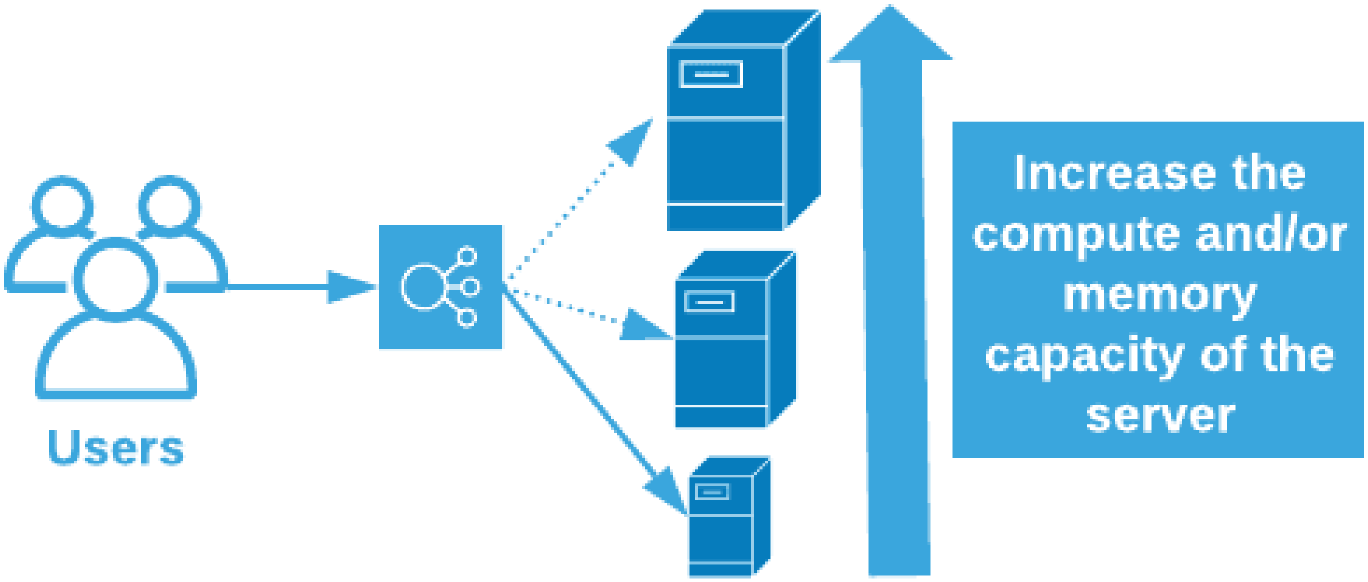
Three types of scalability – Vertical, Horizontal and Diagonal

Scale Vertically - Scale Up:

- Vertical Scaling or Scaling up is easy, it can be done by moving the application to bigger virtual machines deployed in the cloud or you can scale up by adding expansion units as well with your current infrastructure.
- This ability to add resources to accommodate increasing workload volumes is **vertical scaling**. It can resize your server with no change in your code.
- The downside to scaling up is that it increases storage capacity but the performance is reduced because the compute capacity remains the same. Workloads requiring higher throughput demand reduced latency and this can only by fulfilled by Horizontal Scaling / Scaling out.

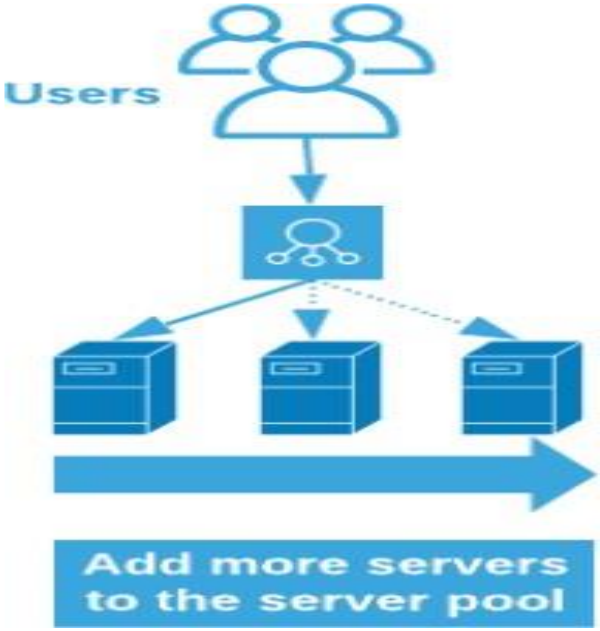
Vertical scale or scale-up

- Adding more compute or memory resources for your applications increases the maximum capacity of the server. When demand spikes, there should not be any noticeable change to your applications.
- **Example:** An example of this would be increasing the number of CPUs or increase the memory of a database server.

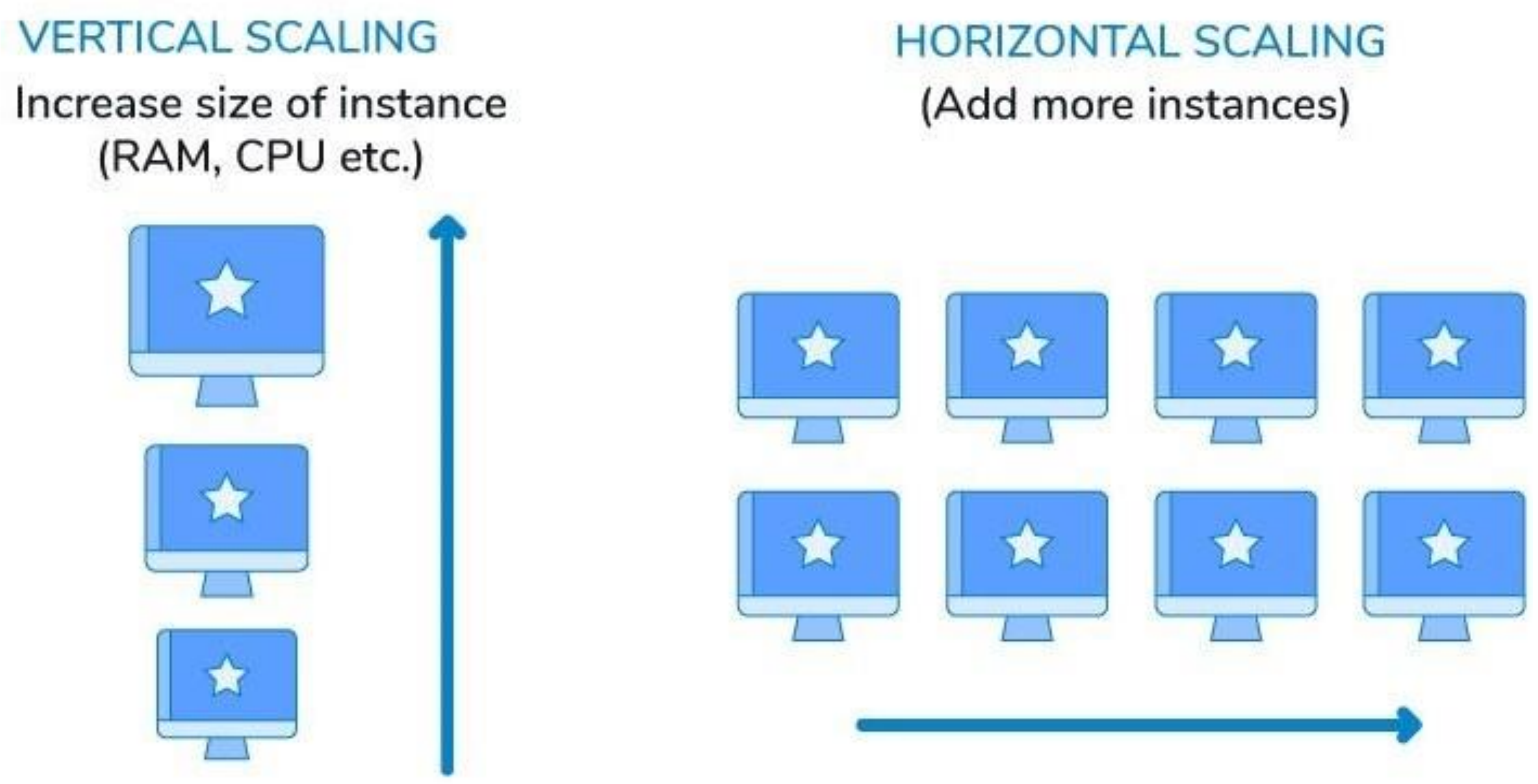


Horizontal scale or scale-out

- **Horizontal scale or scale-out:** Adding more individual servers into a resource pool where your applications run.
- **Example:** An example of this would be adding more web servers to your system to handle an increase in traffic.



Vertical vs Horizontal scaling



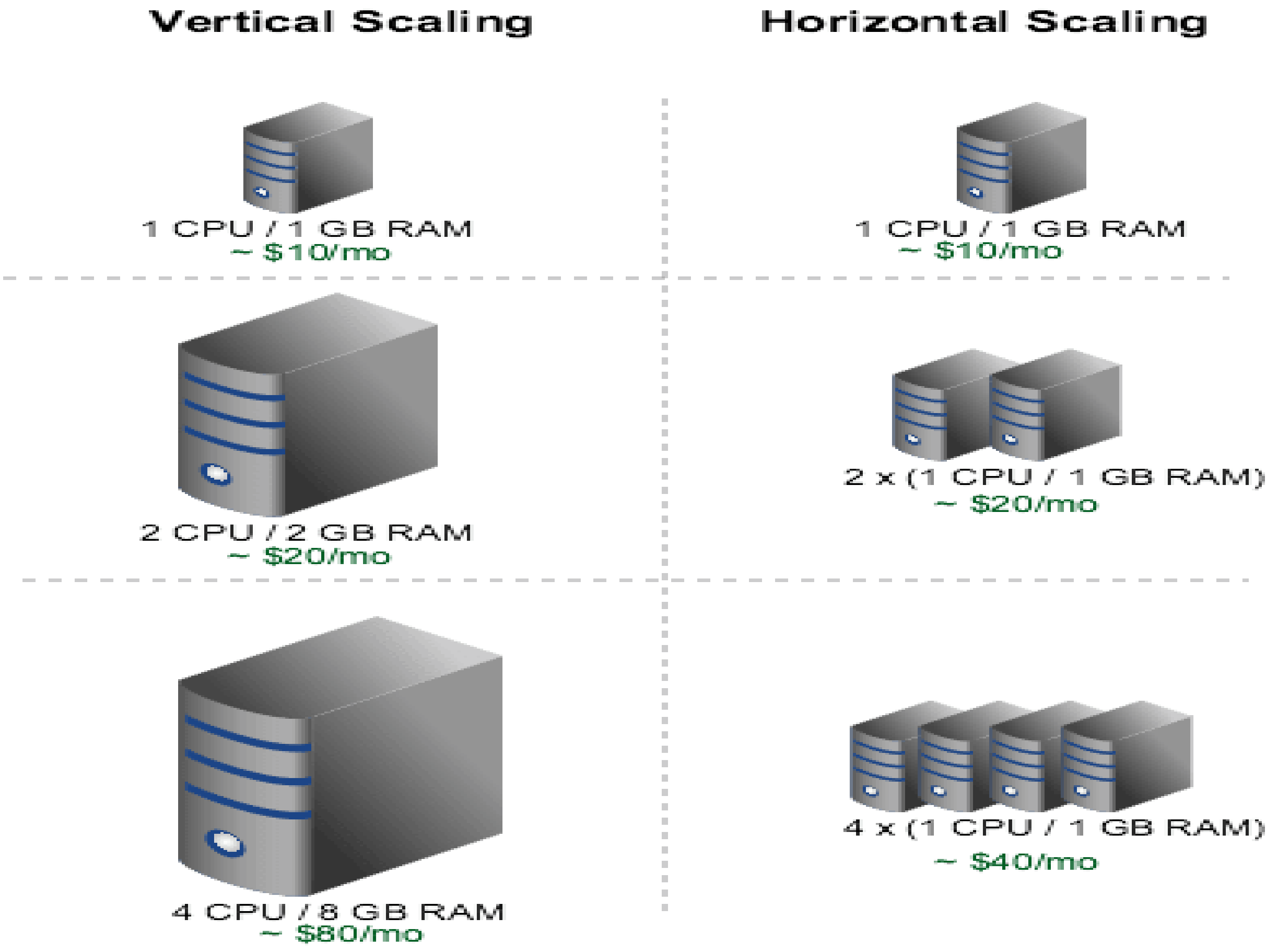
VERTICAL SCALING

Increase size of instance
(RAM, CPU etc.)

HORIZONTAL SCALING

(Add more instances)

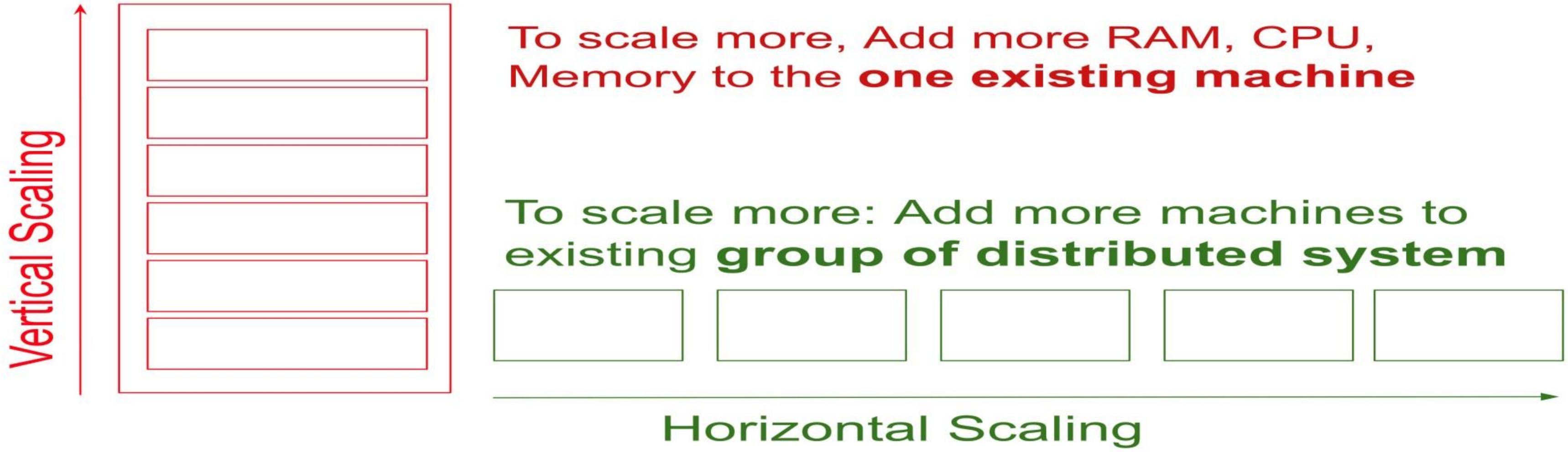
Vertical vs Horizontal scaling



Three types of scalability – Vertical, Horizontal and Diagonal

Scale Horizontally - Scale out:

- Horizontal Scaling or Scaling out is the addition of nodes to the existing infrastructure to accommodate additional workload volumes. Contrary to Vertical Scaling, Horizontal Scaling also delivers performance along with storage capacity.
- The total workload volume is aggregated over the total number of nodes and latency is effectively reduced. This scaling is ideal for workloads that require **reduced latency** and **optimized throughput.**

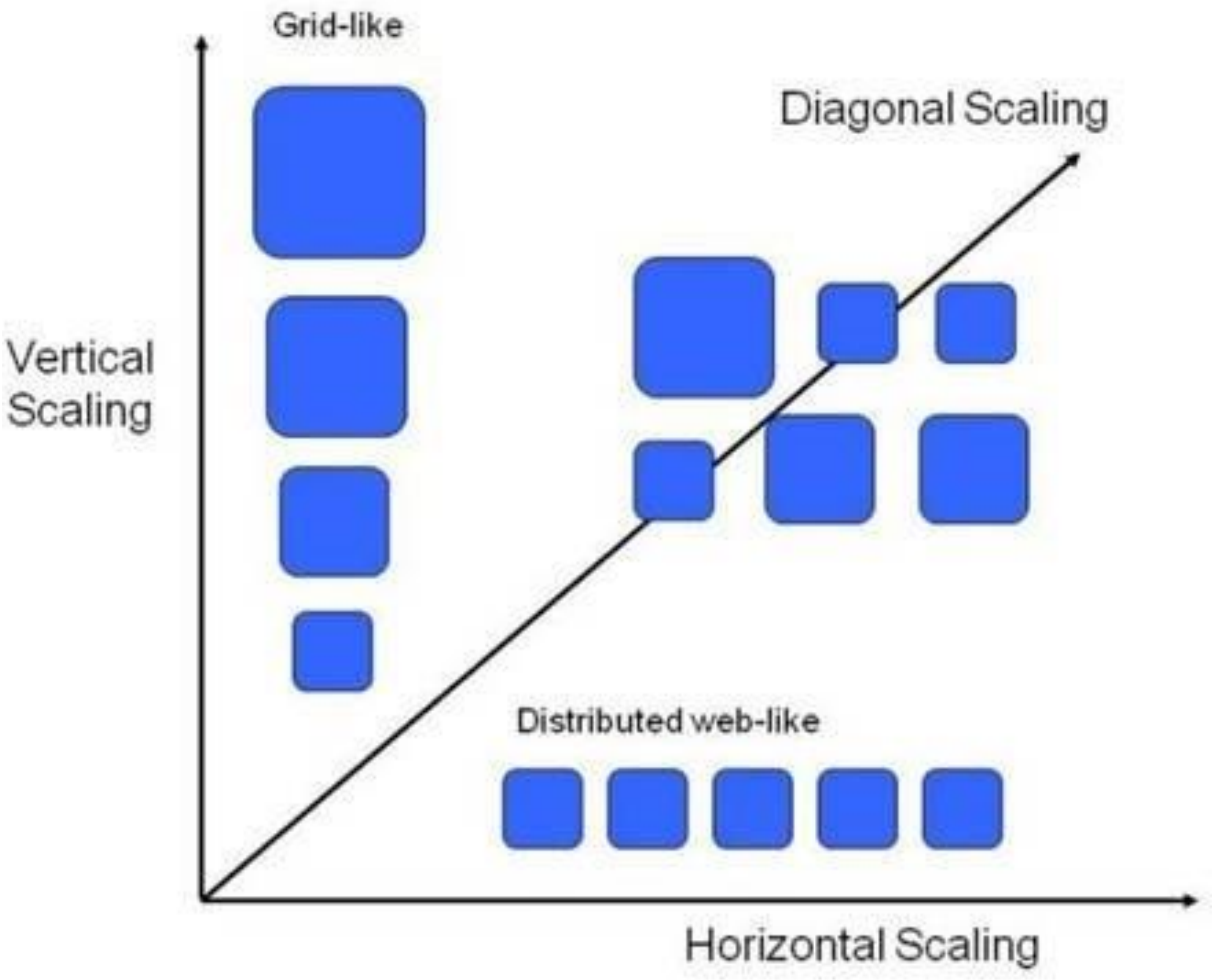

Vertical Scaling
To scale more, Add more RAM, CPU, Memory to the one existing machine
To scale more: Add more machines to existing group of distributed system
Horizontal Scaling

Three types of scalability – Vertical, Horizontal and Diagonal

Scale Diagonally:

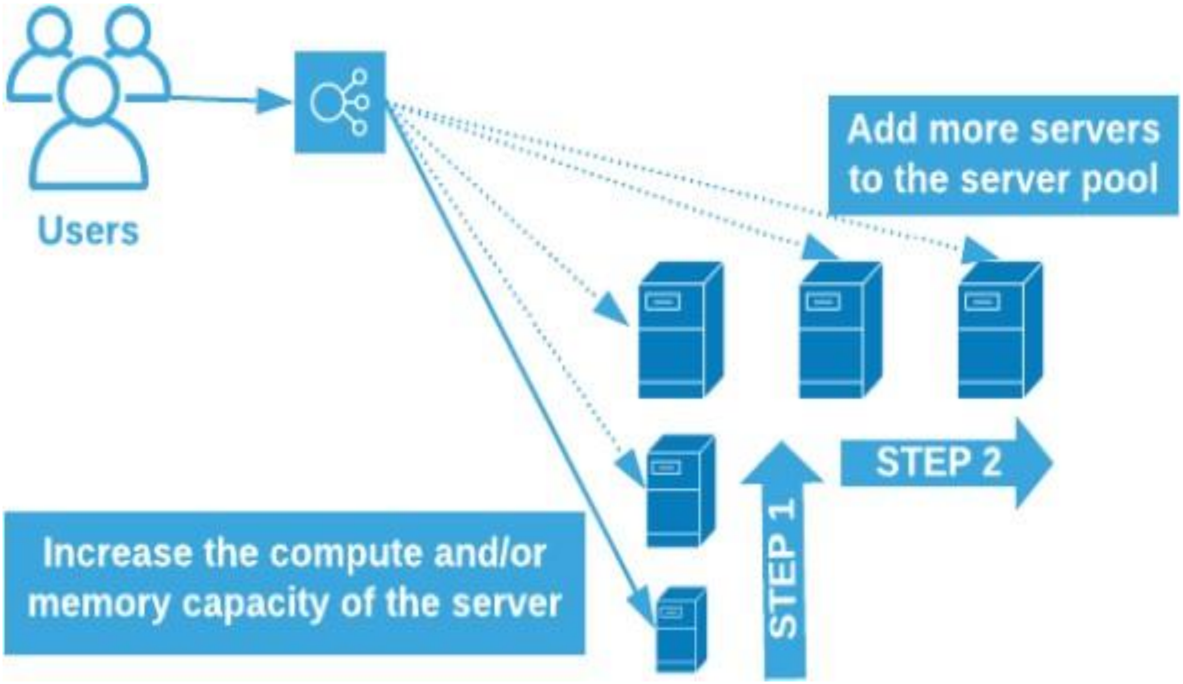
- Diagonal scaling helps you combine the scaling up and scaling down. As the term suggests, scaling down is the removal of storage resources as requirements decrease.
- Diagonal scaling delivers flexibility for workload that require additional storage resources for specific instances of time.
- For instance, a website sets up diagonal scaling; as the traffic increases, the compute requirements are accommodated. As the traffic decreases, the computation capacity is restored to its original size.
- This type of scaling introduces enhanced budgeting and cost effectiveness for environments and businesses dealing with variable workload volumes.

Diagonal Scaling



Diagonal scaling:

- **Diagonal scaling:** Essentially a combination of vertical and horizontal scaling, this setup will scale vertically first until you reach a preset limit and then scale the system horizontally.


Users
Add more servers to the server pool
STEP 2
STEP 1
Increase the compute and/or memory capacity of the server

Scalability function

1. Units of work (requests).
2. The rate of requests over time (arrival rate).
3. The number of units of work in a system at a time (concurrency).
4. The number of customers, users, or driver processes sending requests.

- Each of these can play sensible roles in the scalability function
- Scalability is the property of a system to handle a growing amount of work by adding resources to the system.
- Scalability can be defined as a mathematical function, a relationship between independent and dependent variables (input and output).

Best practices for using scalability in cloud computing:

- **Leverage auto-scaling capability with supervision:** Most cloud providers provide auto-scaling options. This allows us to manage the required resources appropriately as needed.
- **Architect your solution for scalability:** Not all applications can work as expected when scaled. This requires well-defined architecture and design patterns, such as distributed queues, Statelessness, Scalable storage needs etc. that makes the applications work well when scaled.
- **Use load balancers:** It is important to have load balancers in the front which will receive the incoming traffic and manage the distribution of load across all the servers as you **scale up** or **scale down**.
- **Have a robust testing strategy:** Ensure you can test the scalability of your applications and the configurations put in place. Real business transactions contributing to revenue stream is not the place to test this.

Business whose resource demands are increasing slowly and predictably.

Example: [Call center]

- The typical call center is continuously growing. New employees come in to handle an increasing number of customer requests gradually, and new features are introduced to the system (like sentiment analysis, embedded analytics, etc.).
- In this case, cloud scalability is used to keep the system’s performance as consistent and efficient as possible over an extended time and growth.

Example: [Call center]

- In natural language processing model training and optimization for chat-bots. The system starts off on a certain scale and requires room for gradual improvement as it is being used. The database expands and the operating inventory becomes much more intricate.

Scalable Cloud Based Services:

- **Infrastructure-as-a-Service (IaaS)** - like Amazon EC2 or Google Compute Engine;
- **Platform-as-a-Service (PaaS)** - like Magento Commerce Cloud or AWS Elastic Beanstalk;
- **Storage-as-a-Service (STaaS)** - Google Drive, Microsoft OneDrive, and the likes
- .
- **Data-as-a-Service (DaaS)** - customer relationship platforms like Salesforce and Hubspot, ERP applications;
- **Database-as-a-Service (DBaaS)** - AWS SimpleDB, Rackspace, Oracle, MongoDB;

Benefits of cloud scalability

Performance:

- One core benefit of scalability in the cloud is that it facilitates performance. Scalable architecture has the ability to handle the bursts of traffic and heavy workloads that will come with business growth.

Cost-efficient:

- You can allow your business to grow without making any expensive changes in the current setup. This reduces the cost implications of storage growth making scalability in the cloud very cost effective.

Easy and Quick:

- Scaling up or scaling out in the cloud is simpler; you can commission additional VMs with a few clicks and after the payment is processed, the additional resources are available without any delay.

Benefits of cloud scalability

Capacity:

- Scalability ensures that with the continuous growth of your business the storage space in cloud grows as well. Scalable cloud computing systems accommodate your data growth requirements. With scalability, you don't have to worry about additional capacity needs.

Scalability admonition:

- Scalability also has some limitations. If you want a fully scalable system then you have a large task to handle. It requires planning, testing and again testing for your data storage. If you have the applications already then splitting up the system will require code changes, updates

Monitoring:

- You have to be well prepared for the digital transformation of your infrastructure.

Elasticity

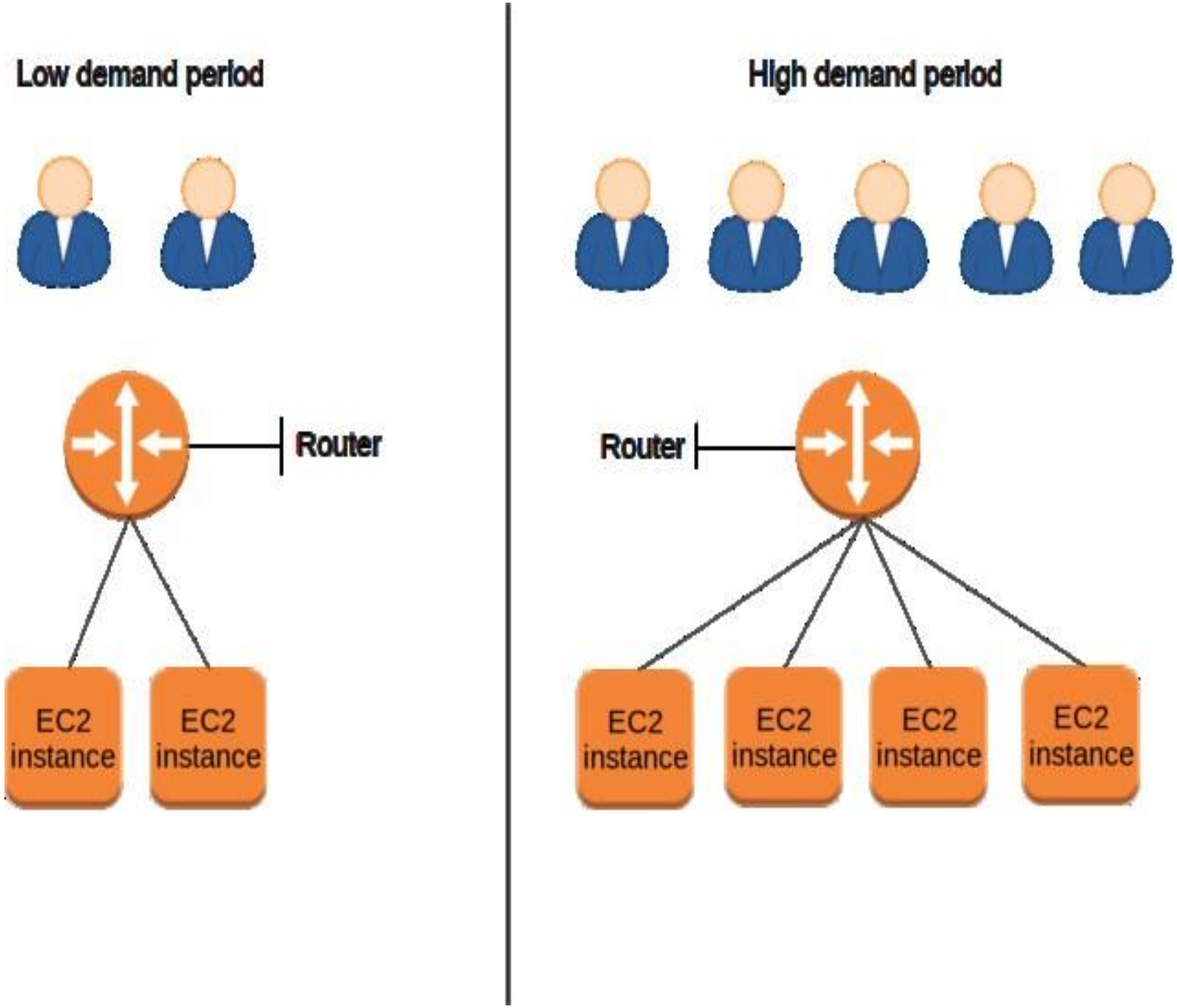
Elastic computing is the ability to quickly expand or decrease computer processing, memory and storage resources to meet changing demands without worrying about capacity planning and engineering for peak usage.

Elasticity – generally refers to increasing or decreasing cloud resources.

What is Cloud Elasticity?

- Cloud elasticity is a system’s ability to manage available resources according to the current workload requirements dynamically.
- Elasticity is a vital feature of cloud infrastructure. It comes in handy when the system is expected to experience sudden spikes of user activity and, as a result, a drastic increase in workload demand.
- Because of the pay-per-use pricing model of modern cloud platforms, cloud elasticity is a cost-effective solution for a business with a dynamic workload.
- Businesses with dynamic resource demands like streaming services or e-commerce marketplaces changes dynamically with various seasonal events. These volatile ebbs and flows of workload require flexible resource management to handle the operation consistently.

AWS and Elasticity:



Elasticity is Cloud Infrastructure

- *Elasticity is one of the fundamental properties of the cloud.*
- Elasticity is the power to scale computing resources up and down easily and with minimal friction. It is important to understand that elasticity will ultimately drive most of the benefits of the cloud.
- As a cloud architect, you need to internalize this concept and work it into your application architecture in order to take maximum benefit of the cloud.
- The on-demand and elastic nature of *the cloud approach* (Automated Elasticity), however, enables the infrastructure to be closely aligned (as it expands and contracts) with the actual demand, thereby increasing overall utilization and reducing cost.

Scale-up approach and scale-out approach

- **Scale-up approach and The traditional scale-out approach, b**oth approaches have initial start-up costs and both approaches are reactive in nature.
- Traditional infrastructure generally necessitates predicting the amount of computing resources your application will use over a period of several years. If you under-estimate, your applications will not have the horsepower to handle unexpected traffic, potentially resulting in customer dissatisfaction. If you over-estimate, you’re wasting money with superfluous resources

Example: Dynamic resource demands

- **Cloud elasticity** is a cost-effective solution for a business with a dynamic workload.

Example: Streaming Services.

Netflix is dropping a new season of Mindhunter. The notification triggers a significant number of users to get on the service and watch or upload the episodes. Resource-wise, it is an activity spike that requires swift resource allocation.

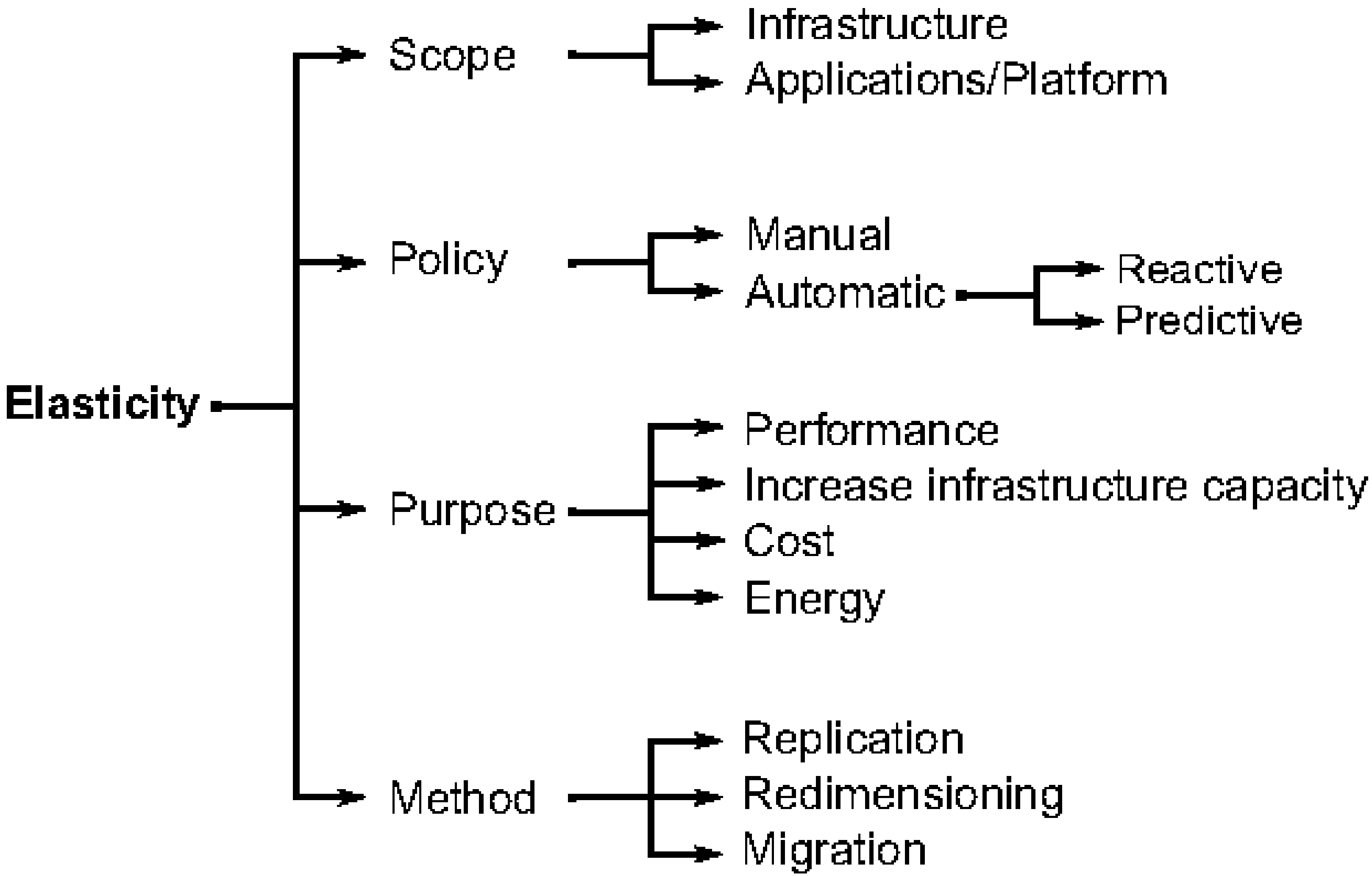
Example: E-commerce.

- Amazon has a Prime Day event with many special offers, sell-offs, promotions, and discounts. It attracts an immense amount of customers on the service who are doing different activities. Actions include searching for products, bidding, buying stuff, writing reviews, rating products. This diverse activity requires a very flexible system that can allocate resources to one sector without dragging down others.

Design Challenge

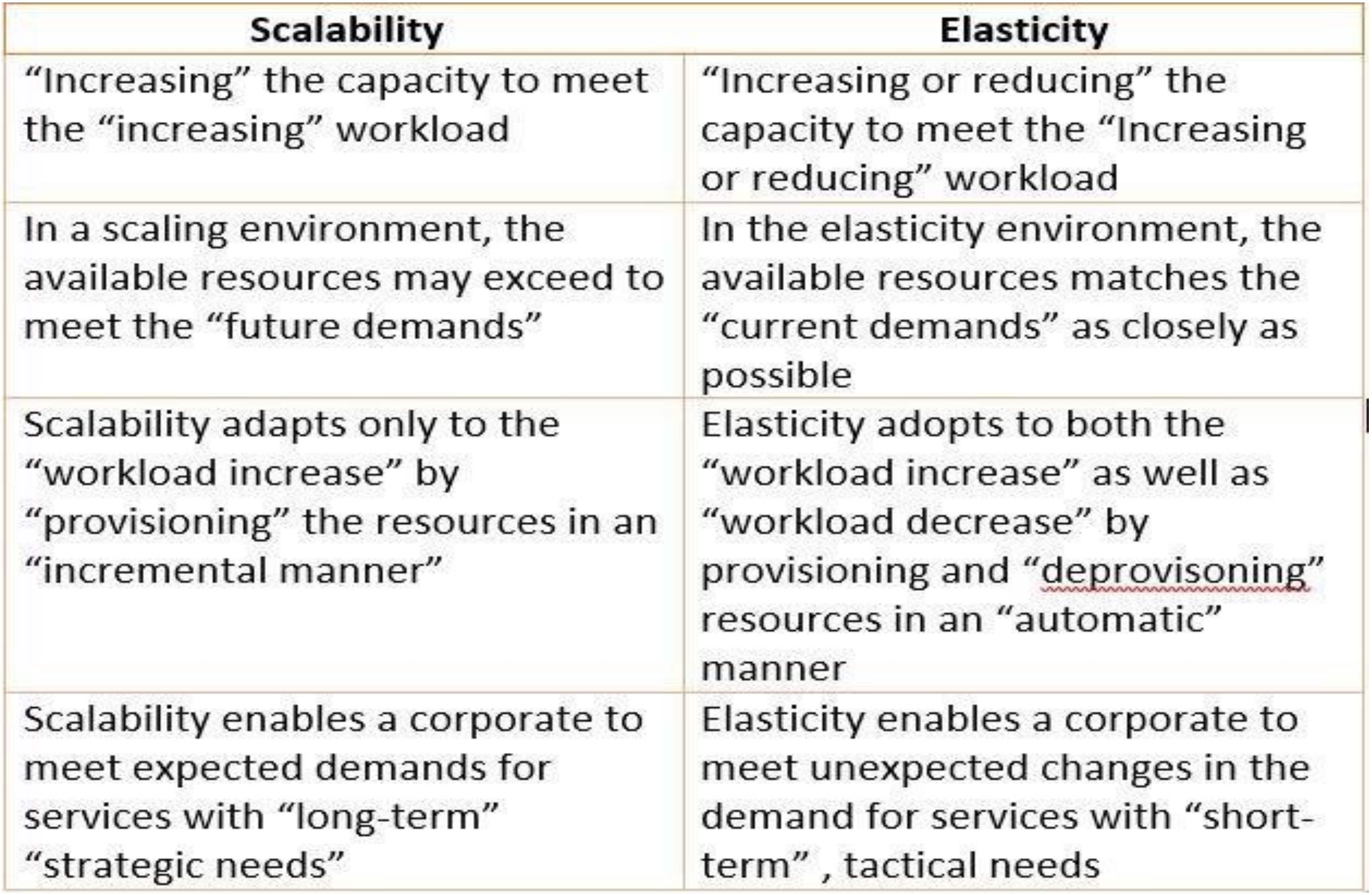
- Designing intelligent elastic cloud architectures, so that infrastructure runs only when you need it, is an art in itself.
- Elasticity should be one of the architectural design requirements or a system property.
- What components or layers in your application architecture can become elastic? What will it take to make that component *elastic*? What will be the impact of implementing elasticity to my overall system architecture?

NOTE: To effectively leverage the cloud benefits, it is important to architect with this mindset.



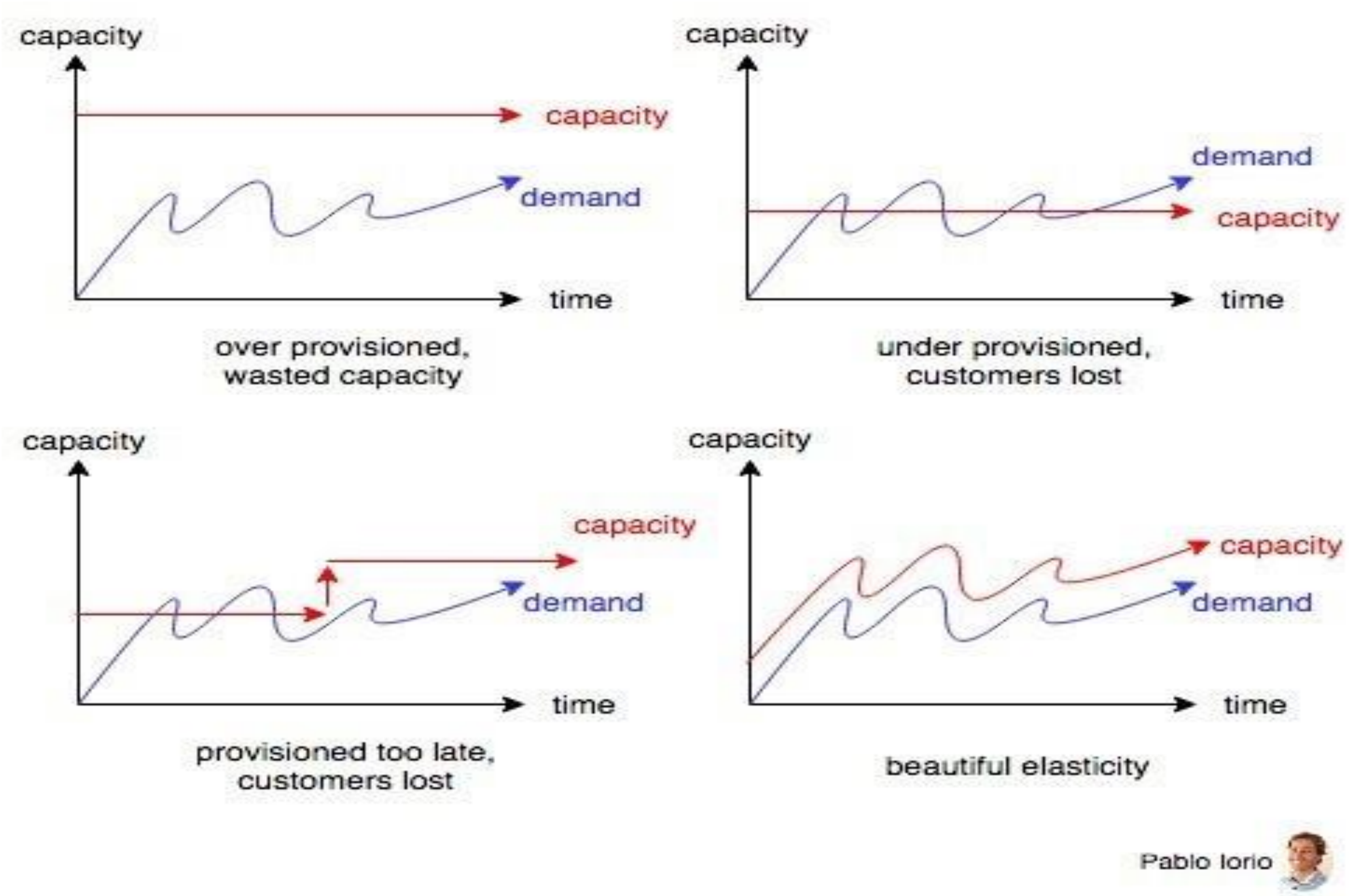
- Elasticity
  - Scope
    - Infrastructure
    - Applications/Platform
  - Policy
    - Manual
    - Automatic
      - Reactive
      - Predictive
  - Purpose
    - Performance
    - Increase infrastructure capacity
    - Cost
    - Energy
  - Method
    - Replication
    - Redimensioning
    - Migration

Scalability vs Elasticity



Scalability	Elasticity
“Increasing” the capacity to meet the “increasing” workload	“Increasing or reducing” the capacity to meet the “Increasing or reducing” workload
In a scaling environment, the available resources may exceed to meet the “future demands”	In the elasticity environment, the available resources matches the “current demands” as closely as possible
Scalability adapts only to the “workload increase” by “provisioning” the resources in an “incremental manner”	Elasticity adopts to both the “workload increase” as well as “workload decrease” by provisioning and “deprovisoning” resources in an “automatic” manner
Scalability enables a corporate to meet expected demands for services with “long-term” “strategic needs”	Elasticity enables a corporate to meet unexpected changes in the demand for services with “short-term” , tactical needs

Under and over provisioning vs perfect elasticity of resources



over provisioned, wasted capacity

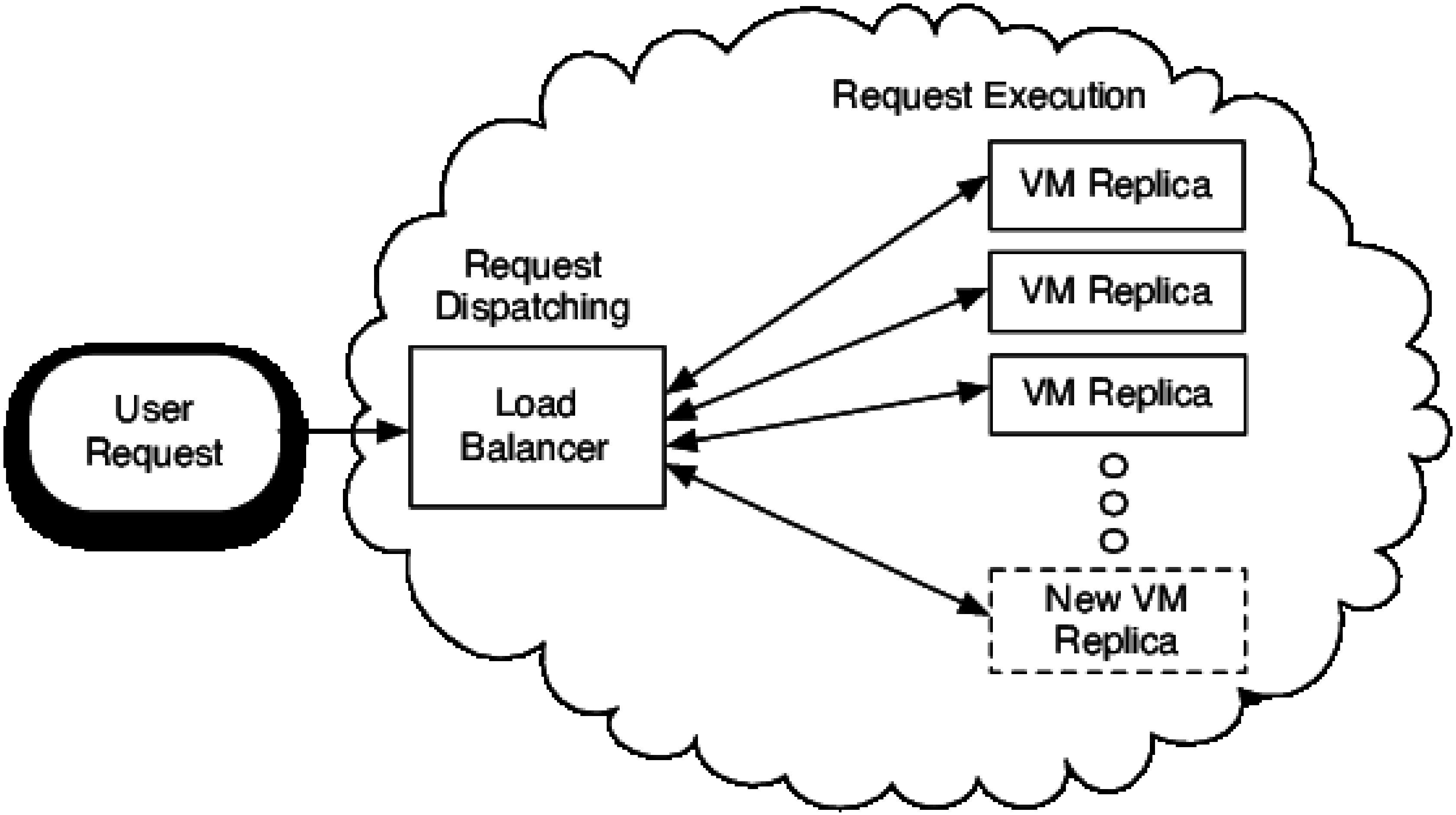
under provisioned, customers lost

provisioned too late, customers lost

beautiful elasticity

Pablo Iorio

VM replication on demand



Conclusion: cloud scalability and cloud elasticity

- Modern business operations live on consistent performance and instant service availability. Cloud scalability and cloud elasticity handle these two business aspects in equal measure.
- Cloud scalability is an effective solution for businesses whose workload requirements are increasing slowly and predictably.
- Cloud elasticity is a cost-effective solution for the business with dynamic and unpredictable resource demands.
- These features make both scalability and elasticity a viable instrument for the company to hold its ground, grow steadily, and gain a competitive advantage.

Exercises

1. What is cloud computing? What are the benefits of cloud computing?
2. What is a cloud? What are the different data types used in cloud computing? Which are the different layers that define cloud architecture?
3. List the top use Cases for Cloud Computing ?
4. What are the different layers in cloud computing? Explain working of them with appropriate use cases.
5. What is on-demand functionality? How is it provided in cloud computing?
6. What are the different models for deployment in cloud computing? Explain them with example?
7. What is the difference between scalability and elasticity?
8. What are the open source cloud computing platform databases? Give some example of large cloud provider and databases?
9. What is the difference between cloud and traditional datacenters?
10. What are the different datacenters in cloud computing?
11. What are the most essential things that must be followed before going for cloud computing platform?