**Scaling**

A business will always scale and serve more requests than current number in future. The system must be ready to serve the huge incoming requests.

To handle this, we can have:

1. Vertical Scaling

Here, we **increase** the serving **capacity** of **existing resources** ↑

1. Horizontal Scaling

Here, we **increase** the **number of resources** →.

While scaling, the **performance** of system should **not decrease** or the **cost** of system must **not increase** a lot and **managing systems** must **not become complex**.

You need to fulfill all 3 conditions to ensure that your system is really scalable.

Real life example:

Suppose you own a restaurant –

1. **Handle increased load**

You can handle more customers by putting more tables and chairs.

1. **Not complex**

For managing multiple customers, you added more waiters and managers.

1. **Performance must not decrease**

But, your kitchen capacity is still small. You have limited number of stoves and ovens.

In this case, restaurant will not scale as one of the 3 conditions has failed.

When to choose horizontal or vertical?

When the business is not huge and you can attain enough scalability just by increasing capacity of existing resources and thereby easy to manage - choose vertical scaling.

When the business is huge, has large user base, we need to increase instances of applications or server to handle multiple requests. Keep all instances behind load-balancer to manage the system – use horizontal scaling.

Note: this scaling is not limited to application, it’s also applicable to Cache instances, DB instances, Message Queues, Servers, etc.