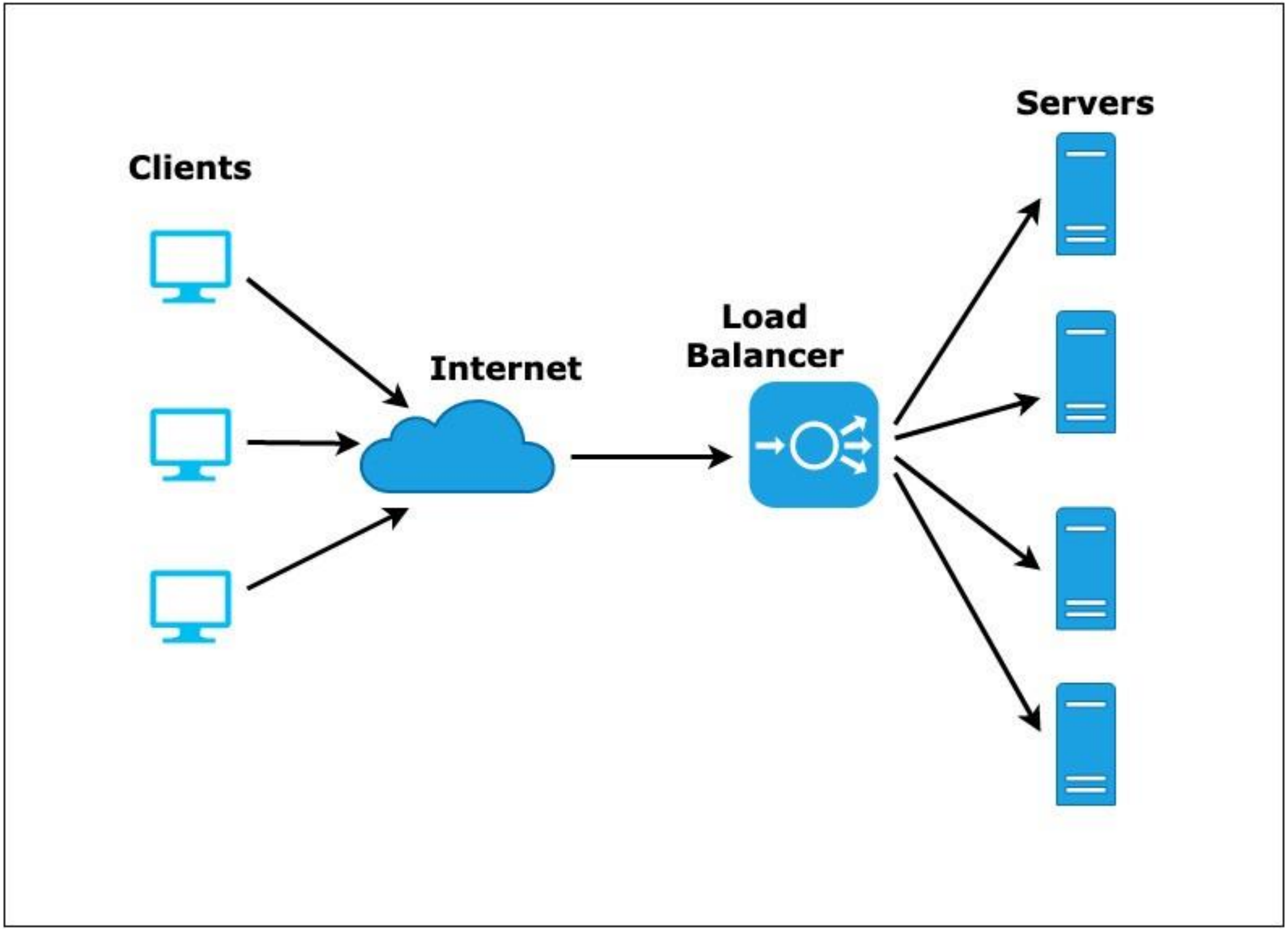
Load Balancing

- **Load Balancing**
  - All inbound traffic to a web role passes through a stateless load balancer, which distributes client requests among the role instances.
  - Individual role instances do not have public IP addresses, and are not directly addressable from the Internet. (This is why you can’t ‘Ping’ the IP addresses).
  - Web roles are stateless, so that any client request can be routed to any role instance.
  - A Status-check event is raised every 15 seconds. This can be used to indicate if the role is ready to receive traffic, or is busy and should be taken out of the load balancer rotation.

Load Balancing



Elasticity

- **Elasticity – What is it?**
  - *the degree to which a system is able to adapt to workload changes by provisioning and deprovisioning resources in an automated manner, such that at each point in time the available resources match the current demand as closely as possible*.

- **Elasticity is automated scalability.**
  - Scalability provides the ability to increase (or decrease) the amount of resources in scaling up (more powerful instances) or out (additional instances), which is usually done through manual intervention.

  - Elasticity does the same but in an automated manner, independent from human interaction.

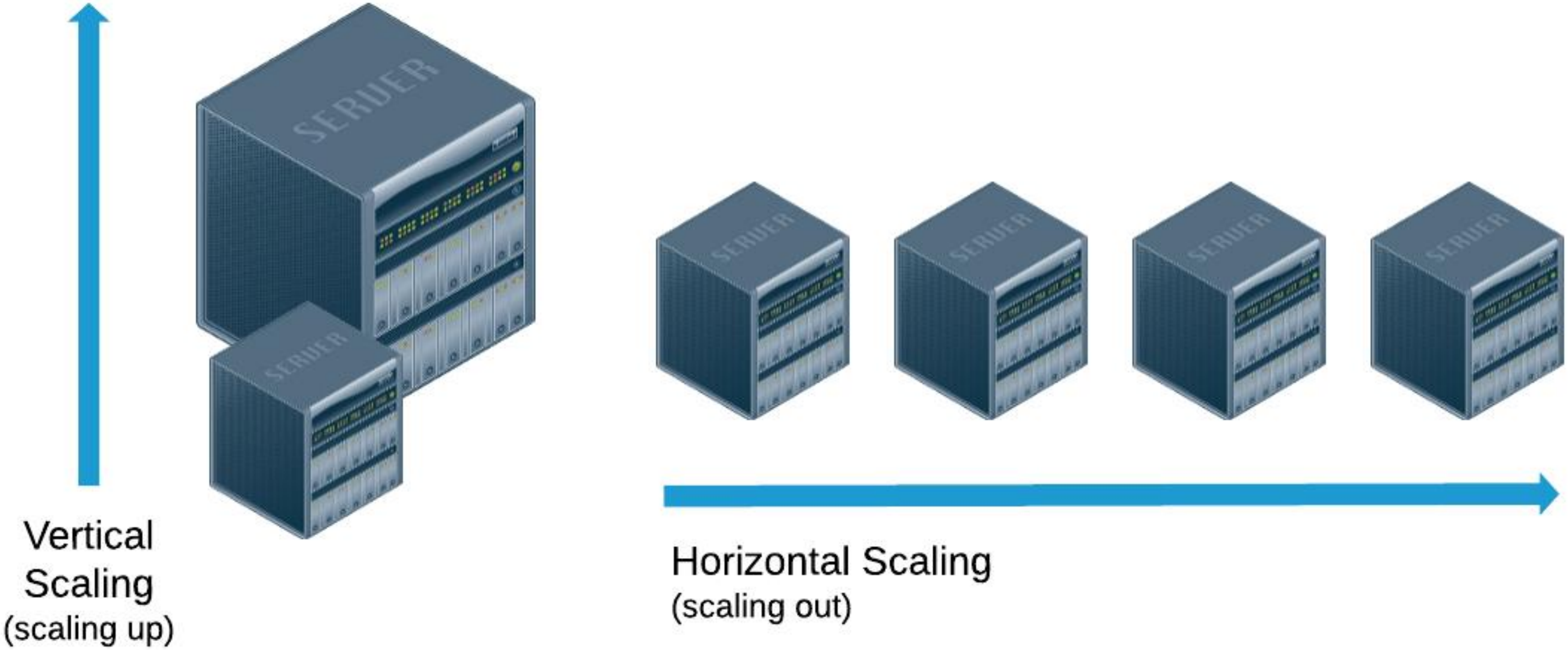
Auto Scaling

- **Elasticity**
  - Elasticity is implemented via Auto Scaling.

- **What is Auto Scaling**
  - Auto Scaling is the process of dynamically allocating the resources required by an application to match performance requirements and satisfy service-level agreements (SLAs), while minimizing runtime costs.

  - Auto Scaling applies to all of the resources used by an application, not just the compute resources. For example, if your system uses message queues to send and receive information, it could create additional queues as it scales.

Types of Auto Scaling



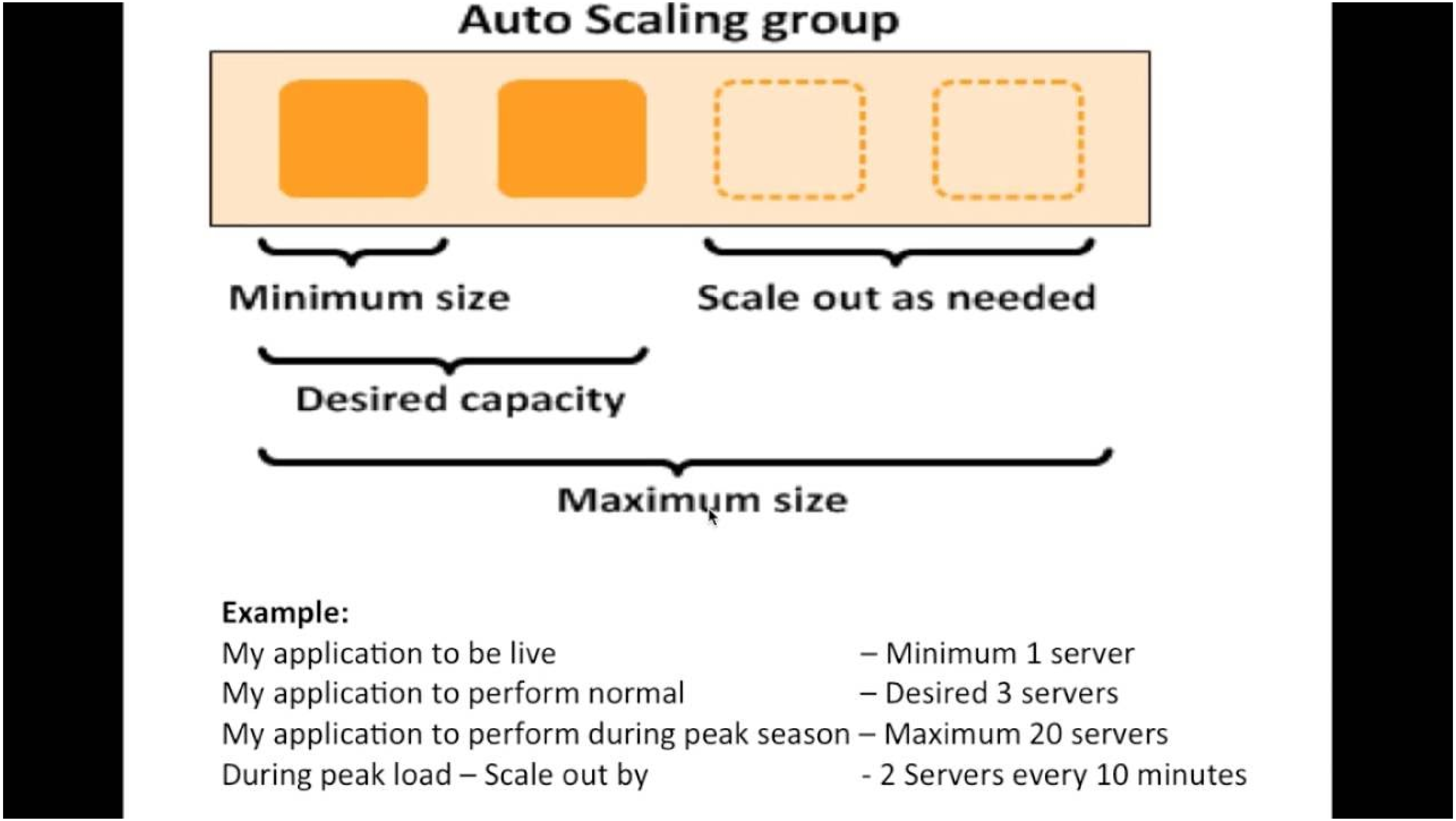
Types of Auto Scaling

- **Vertical Auto Scaling**
  - Often referred to as scaling-up or scaling down.
  - requires that you modify the hardware (expand or reduce its capacity and performance).
  - or redeploy the solution using alternative hardware that has the appropriate capacity and performance.
  - Vertical scaling is often a disruptive process that requires making the system temporarily unavailable while it is being redeployed .
  - **Not a common approach in the Cloud Industry!**

Types of Auto Scaling (ii)

- **Horizontal Auto Scaling**
  - Often referred to as scaling-in or scaling-out.
  - requires deploying the solution on additional or fewer resources, which are typically commodity resources rather than high-powered systems.
  - The solution can continue running without interruption while these resources are provisioned.
  - When the provisioning process is complete, copies of the elements that comprise the solution can be deployed on these additional resources and made available.
  - If demand drops, the additional resources can be reclaimed after the elements using them have been shut down cleanly.
  - **The most common approach to implementing Elasticity in the Cloud.**

AWS Auto Scaling



Example:

My application to be live – Minimum 1 server

My application to perform normal – Desired 3 servers

My application to perform during peak season – Maximum 20 servers

During peak load – Scale out by - 2 Servers every 10 minutes