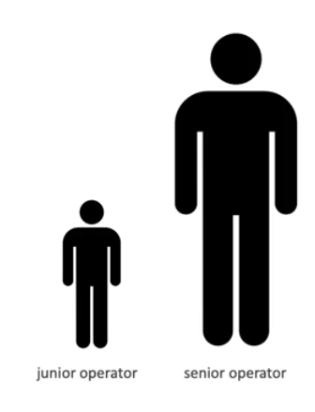
**Scalability and High Availability**

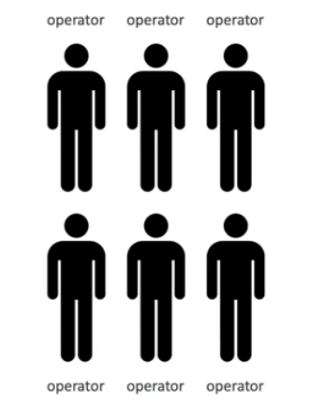
* Scalability means an application /system can handle greater loads by adapting
* There are two types of scalability in:
  + - Vertical Scalability
    - Horizontal Scalability (=Elasticity)
* Scalability is linked but different to High Availability
* **Vertical Scalability:**

1. Vertical Scalability means increasing the size if the instance.   
    Ex: t2.micro -> t2.large
2. Vertical Scalability is very common for non-distributed systems, such as Database
3. There is actually a limit to how much you can vertically scale, that is the limit of the hardware

****

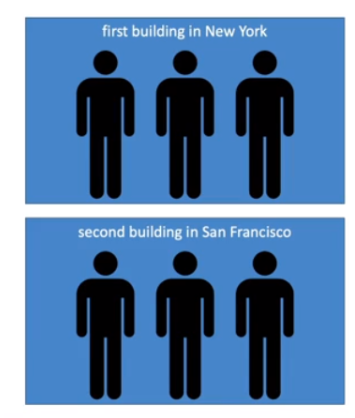
* **Horizontal Scalability:**

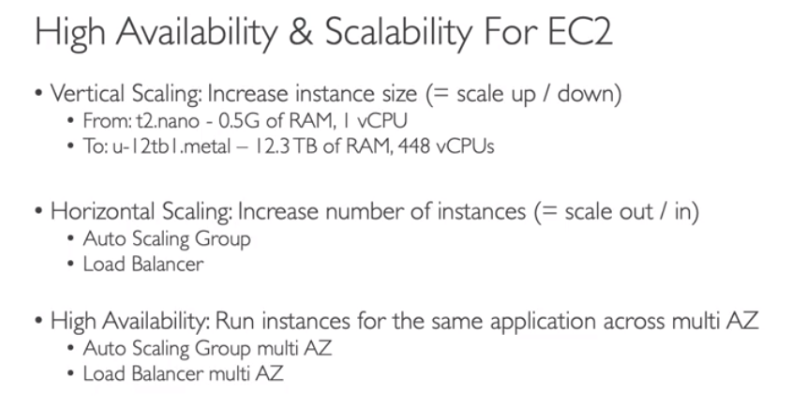
1. Horizontal Scalability means increasing number of instances/systems for your application
2. HS implies distributed systems
3. This is very common for web application/modern application
4. It’s easy to horizontally scale, by using EC2 instances

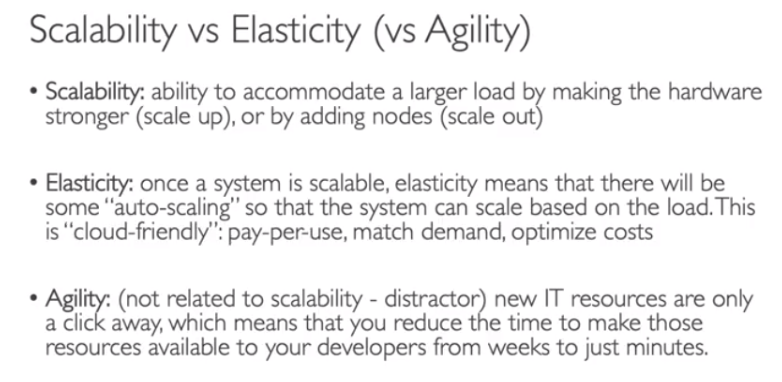
****

* **High Availability**

1. High Availability goes hand in hand with horizontal scaling
2. HA means you’re running your application/system in at least 2 AZs
3. The goal of HA is to avoid Data centre loss (Disaster Recovery)

****

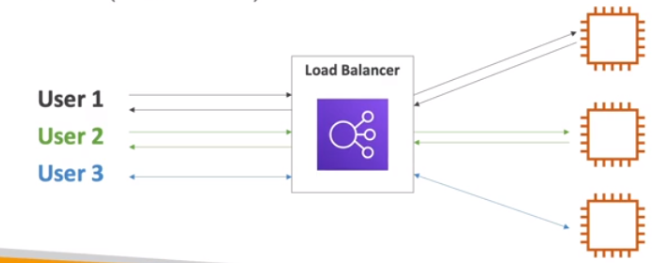
****

****

**ELASTIC LOAD BALANCING**

**Load Balancer:**

1. Load balancers are servers that forward internet traffic to multiple servers (EC2 Instances) downstream
2. Elastic Load Balancer (ELB) is managed by AWS

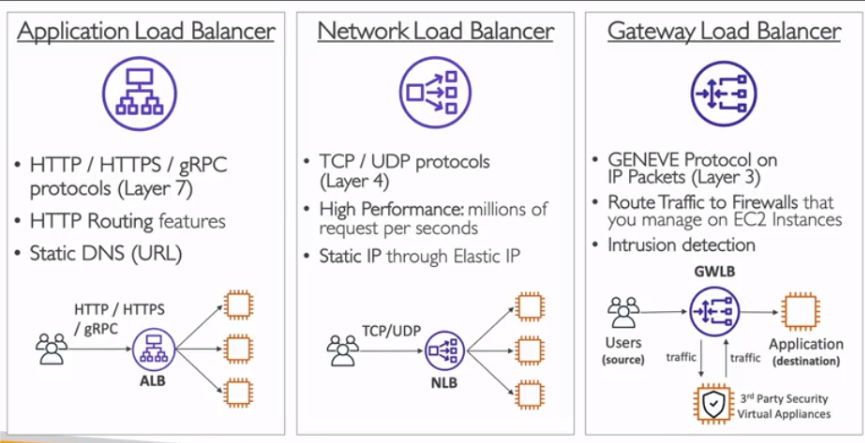


**Why use a Load balancer:**

1. Spread load across multiple downstream instances
2. Expose a single point of access (DNS) to your application
3. Seamlessly handles failures of downstream instances
4. Do regular health checks to your instances
5. Provide SSL termination (HTTPS) for your websites
6. High availability across zones

**Why use an Elastic Load Balancer:**

1. An ELB is a managed load balancer – we don’t need to be provisioning servers, AWS will do it for us.
2. AWS guarantees that it will be working
3. AWS takes care of upgrades, maintenance, HA
4. AWS provides only a few configuration knobs
5. It costs less to setup your own load balancer but it will be a lot more effort on your end (maintenance, integrations)
6. There are 4 kinds of Load Balancers offered by AWS:
7. Application Load Balancer **(ALB):** HTTP/HTTPS only - Layer 7. Handles traffic that are coming from HTTP, HTTPS only
8. Network Load Balancer (**NLB):** Ultra High performance, handles traffic coming from TCP, UDP protocols – layer 4. It provides a static IP (not a static URL)
9. Gateway Load Balancer (**GwLB):** Layer 3
10. Classic Load Balancer (**CLB):** Retired in 2023, layer – 4 and layer 7 – (Not required for the exam)



**ALB – HANDS-ON**

Load balancers can be used when there is some traffic handled by some VMs (EC2s here). So first, we need to launch an EC2.

To create a Load Balancer, under EC2s, scroll down till Load Balancer section and select and create the apt load balancer. Here we can configure the settings such as which AZs we need to deploy the ALBs, SGs etc.

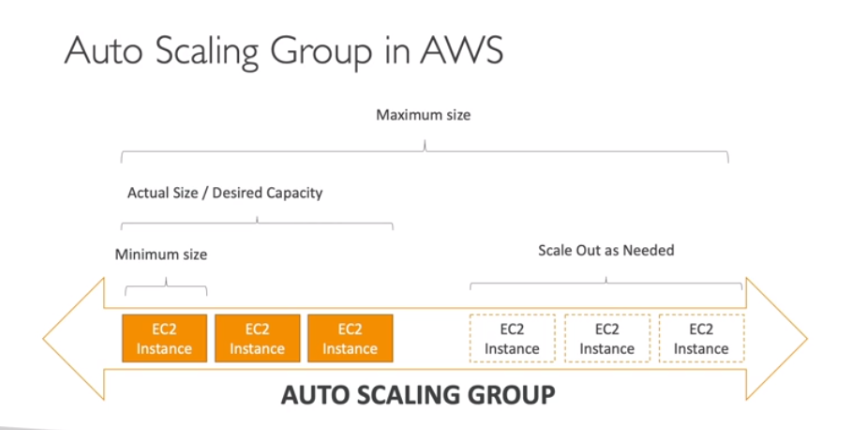
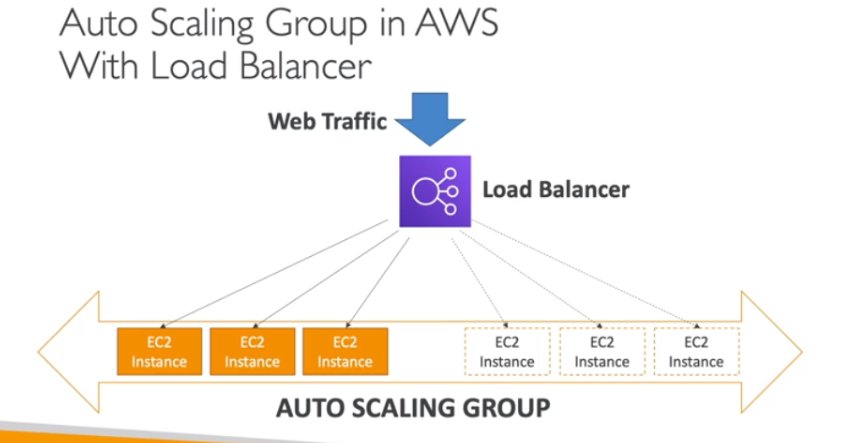
**AUTO SCALING GROUP**

So, we have an application which is load balanced by a load balancer. ASG is used to create service at the backend.

* In real life, the load on your websites and applications can change
* In the cloud, you can create and get rid of servers very quickly.
* The goal of ASG is to:

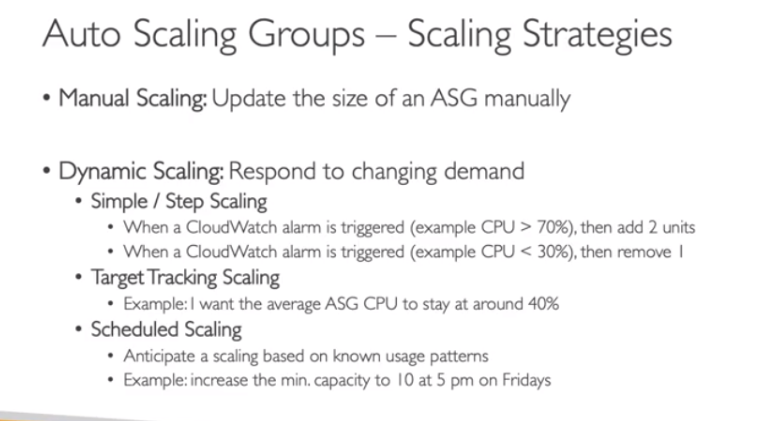
1. Scale-out (add EC2 instances) to match an increased load
2. Scale-in (remove EC2 instances) to match a decreased load
3. We have a minimum and a maximum of number of machines running
4. Once ASG create or remove EC2 instances, these instances are registered or deregistered to our load balancer
5. ASGs replace unhealthy instances by a healthy one

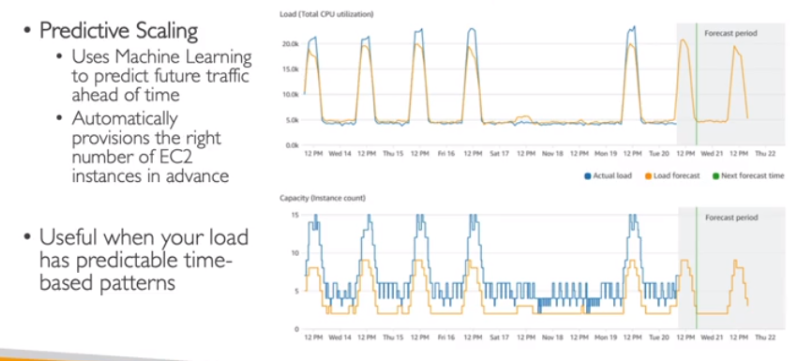
* Cost savings – only run at an optimal capacity



**ASG HAND-ON**

ASG can be created from the option given in the bottom left





**NOTE:** Target Groups don’t cost you any money.