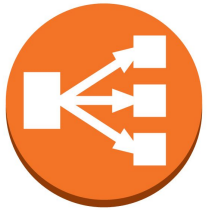


Load Balancer & Autoscaling

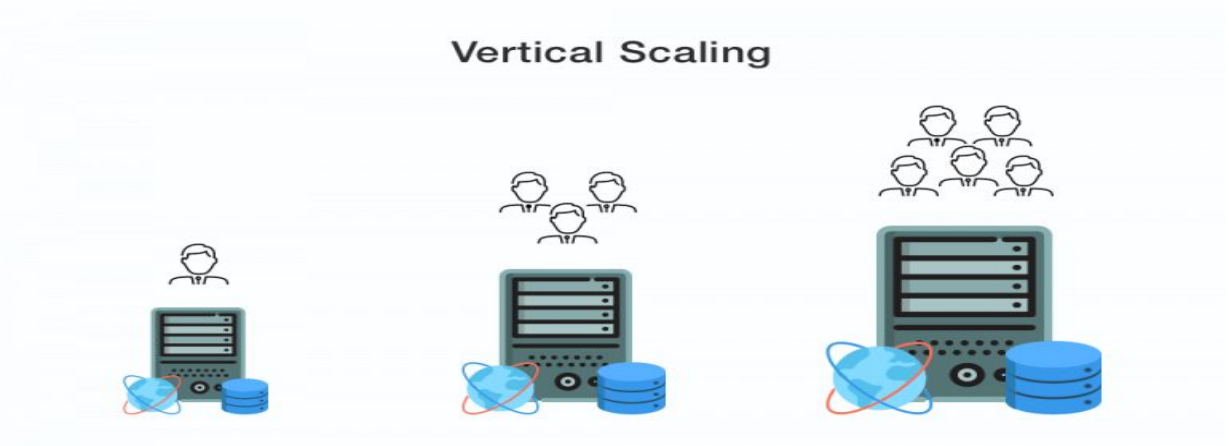
Scalability & High Availability

- Scalability means that an application / system can handle greater loads by adapting.

- There are 2 types of Scalability
  - Vertical Scalability.
  - Horizontal Scalability

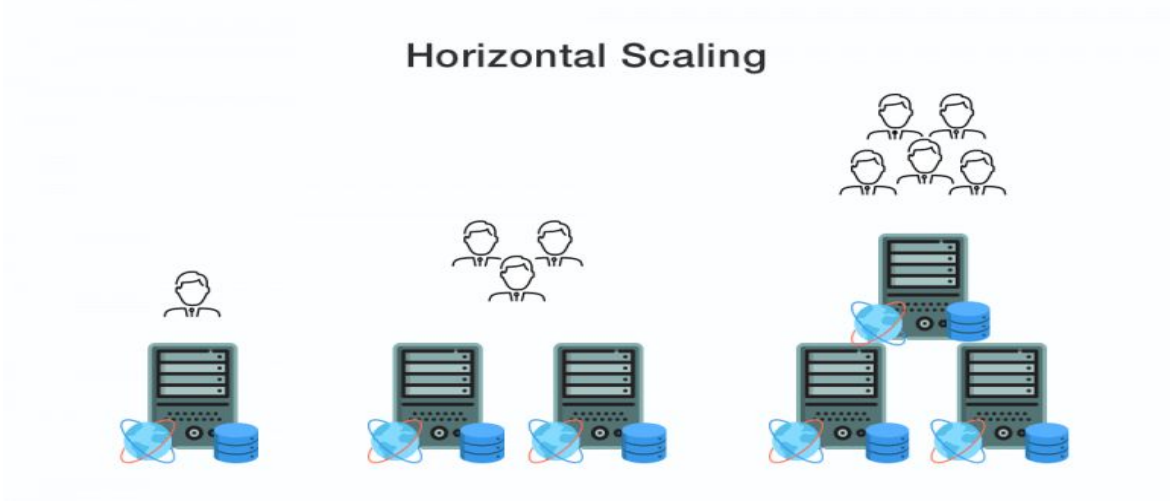
Vertical Scalability

- This means increasing the size of the instance.
- E.g scaling from t2.micro to t2.large.
- Commonly used to non-distributed systems such as db.
- RDS, ElastiCache also can use vertical scaling.



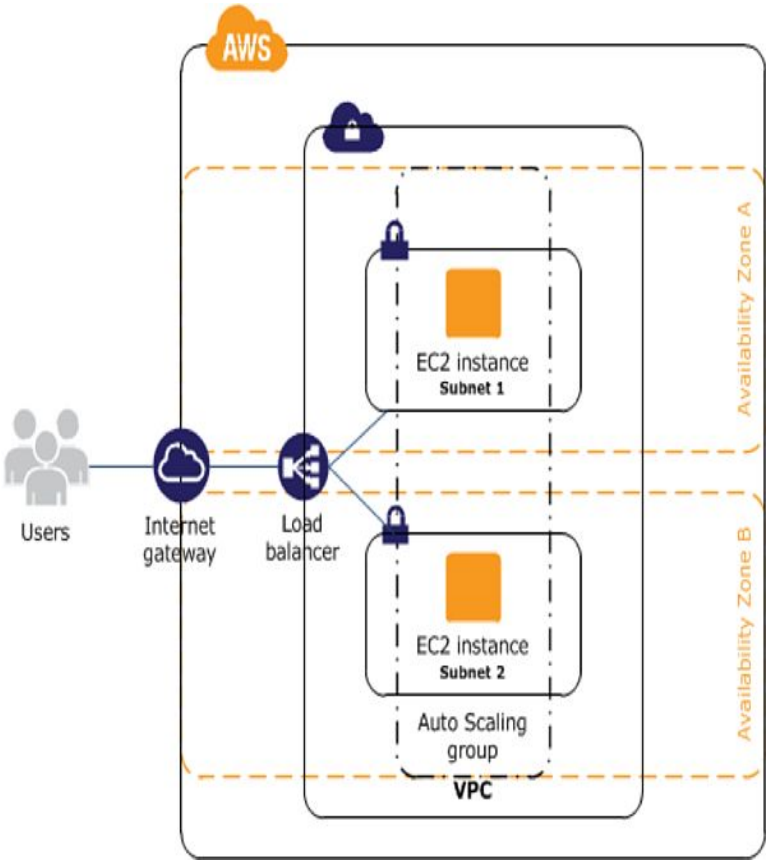
Horizontal Scalability

- It means Increasing the number of instance/systems for app.
- It means the system would be distributed.
- Very Common for web apps/ containerized applications.
- Easy to scale horizontally .



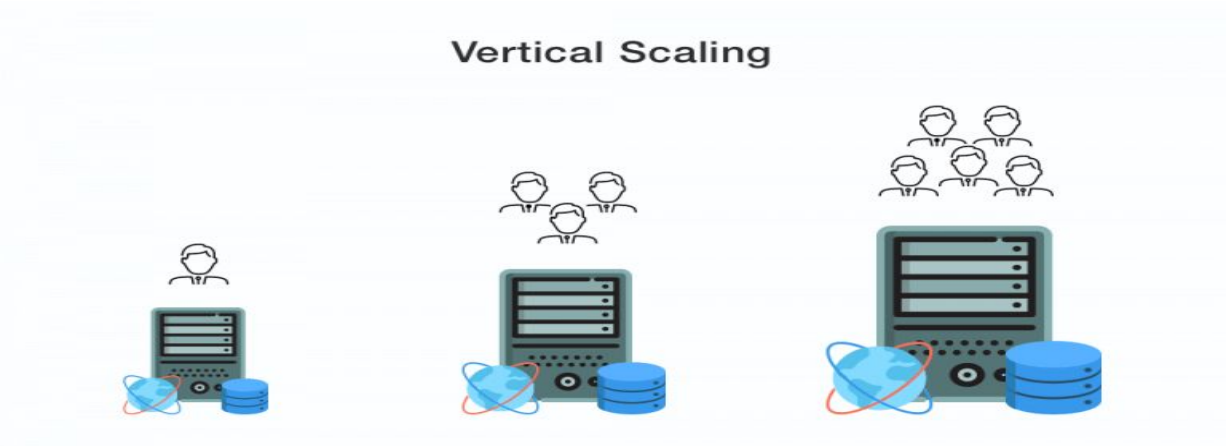
High Availability

- It means running your application /system in at least 2 availability zone (AZ)
- If the application in one AZ is down we still have it on one AZ.
- This one goes hand in hand with horizontal scaling



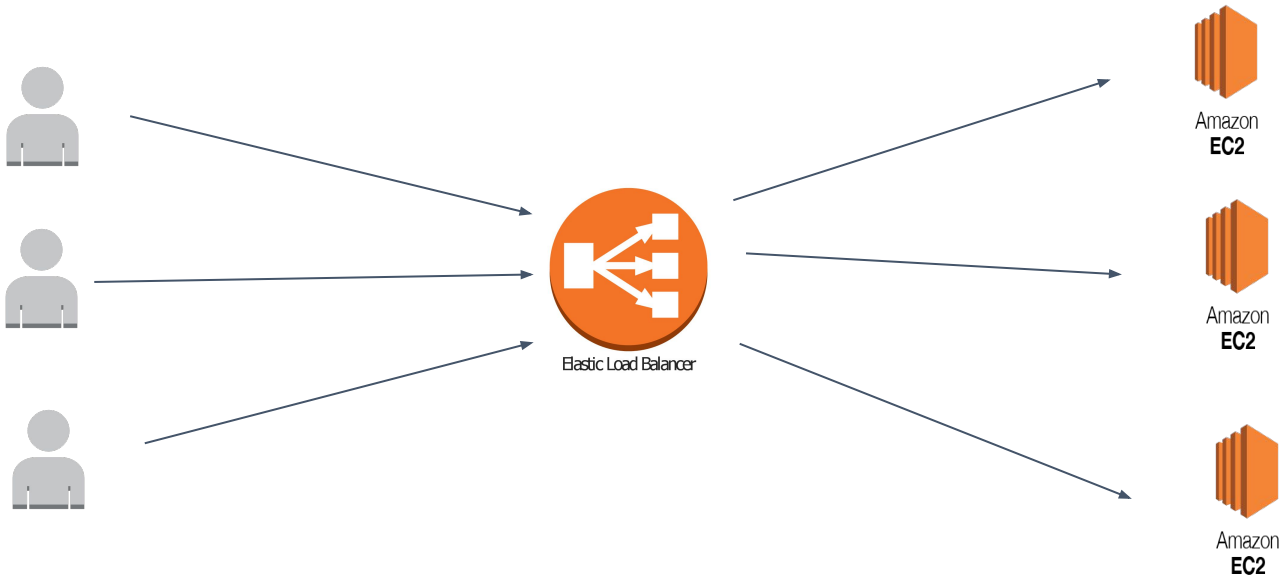
Vertical Scalability

- This means increasing the size of the instance.
- E.g scaling from t2.micro to t2.large.
- Commonly used to non-distributed systems such as db.
- RDS, ElastiCache also can use vertical scaling.



What is load balancing?

- Load balancers in simple words are devices/servers which will forward the traffic to multiple servers attached to it.



Why to use load balancer?

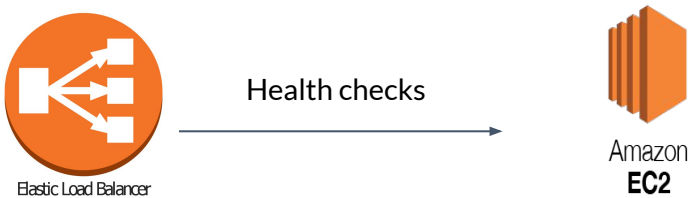
- To distribute the load across multiple servers/instance attached to it .
- You can expose your application to a single point of access (DNS).
- It handles the failures of the servers/instance attached to it (won't send traffic on those servers).
- It does a regular health checks to your servers/instance (we have to provide the inputs).
- Enforces session stickiness with cookies.
- High availability across multiple availability zones.

Why use an EC2 Load balancer ?

- An ELB (Ec2 Load Balancer) is an AWS managed load balancer. We get is as a service.
    - AWS assures us the uptime of the Load balancer, we will not have to worry about that.
    - AWS takes care of it's availability, upgrades & maintenance, we do not have to worry about that.
- It is integrated to your AWS account as a service.

Health Checks

- Health checks are very import part of a Load balancer.
- With health checks the LB knows if the server attached to the load balancer are available or are in healthy state.
- Health check is done on a port and a specific route or path.
- If response if not 200(OK), then the instance is unhealthy.



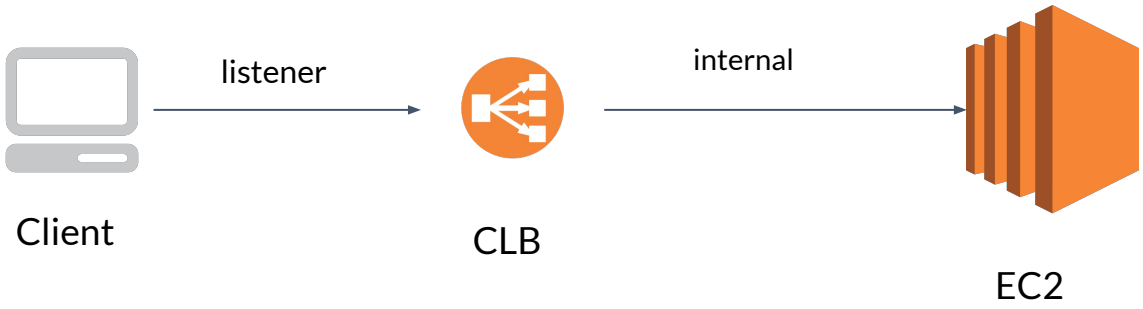
Types of Load Balancers on AWS

- Currently AWS has 3 types of Load balancer those are managed by them.
  - Classic Load balancer (v1 Old generation) - 2009
    - HTTPS, HTTP, TCP
  - Application Load balancer (v2 new generation) - 2016
    - HTTPS, HTTP, WebSocket
  - Network Load balancer (v2 new generation) - 2017.
    - TCP, TLS(secured TCP) & UDP

When creating a load balancer you can select if the load balancer is internal facing or public facing

Classic Load balancer

- Works on layer 4(TCP) & layer 7(HTTP & HTTPS)
- Health check are TCP & HTTP based.
- They have fixed hostname
  - E.g - mylb.region.elb.amazonaws.com



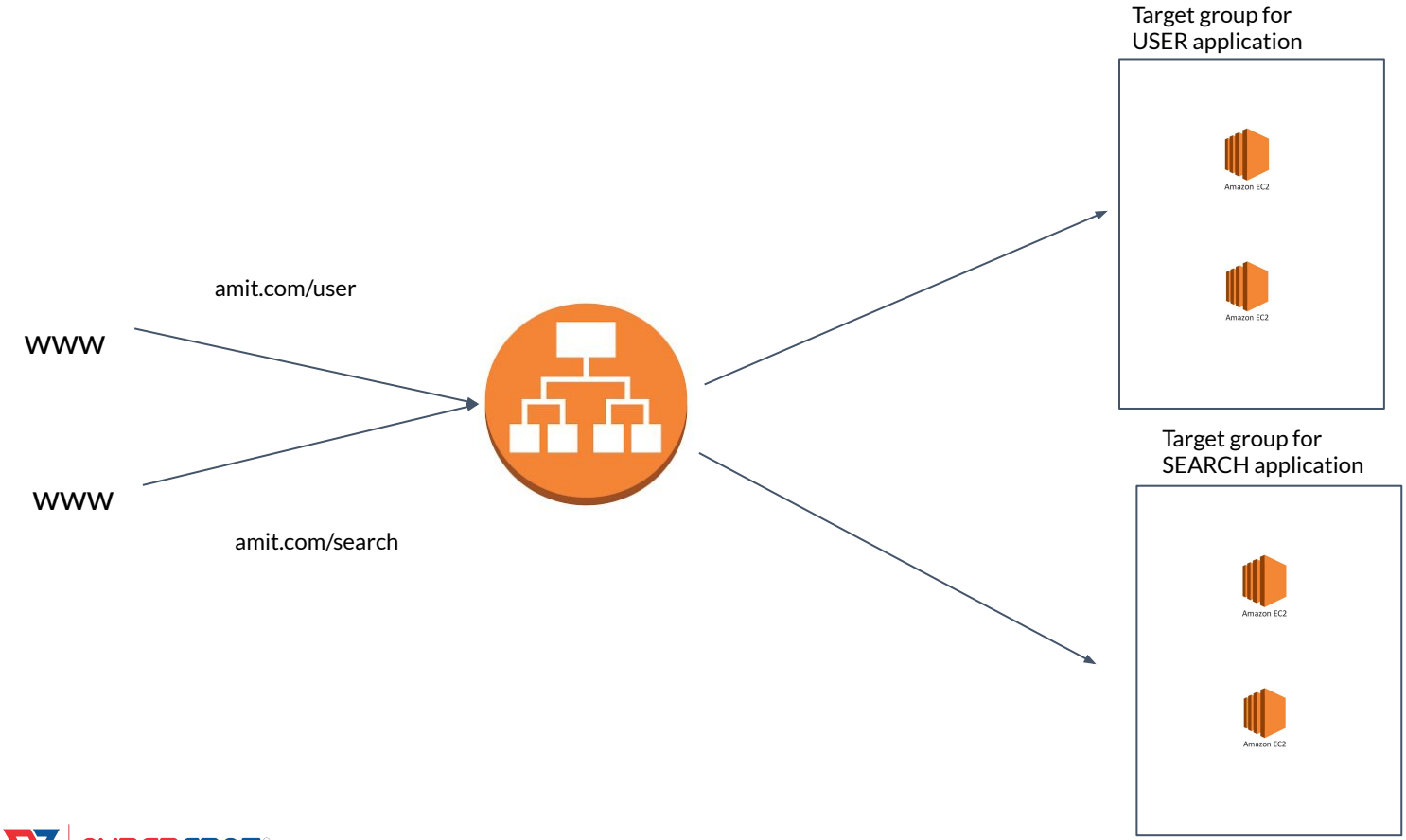
Application Load balancer

- It is a layer 7 load balancer (HTTP).
- Load balancing on multiple HTTP based application on multiple servers/instances (Target group).
- It can also load balance to multiple application on same instance/server. (e.g Containers).
- It also supports redirection (HTTP to HTTPS),

Application Load balancer

- It routes traffic to multiple target groups:
  - path based routing from URL (amit.com/users)
  - hostname based routing (one.amit.com , app.amit.com)
  - Query string, headers based routing (amit.com/user?id=123&order=false)
- ALB is a good fit for microservers based application (docker or ECS).
- It can also load balance to multiple application on same instance/server. (e.g Containers).
- port mapping feature to redirect to dynamic ports in ECS

Application Load balancer



Network Load balancer

- Layer 7 load balancer.
  - It forwards UDP & TCP traffic to the instances
  - Less latency ~100ms (400ms for ALB)
  - NLB is having 1 static IP per AZ & we can assign ElasticIP.
  - NLB are used for extreme performance, TCP or UDP traffic

Load Balancer Stickiness

- If Stickiness is enabled the same client's request will be always redirected to same instance behind load balancer.
- This option is available for only classic & Application Load balancer.
- We can choose the cookie expiry time.
- Useful when we have make sure the user will not lose his session.

SSL/TLS - Basics

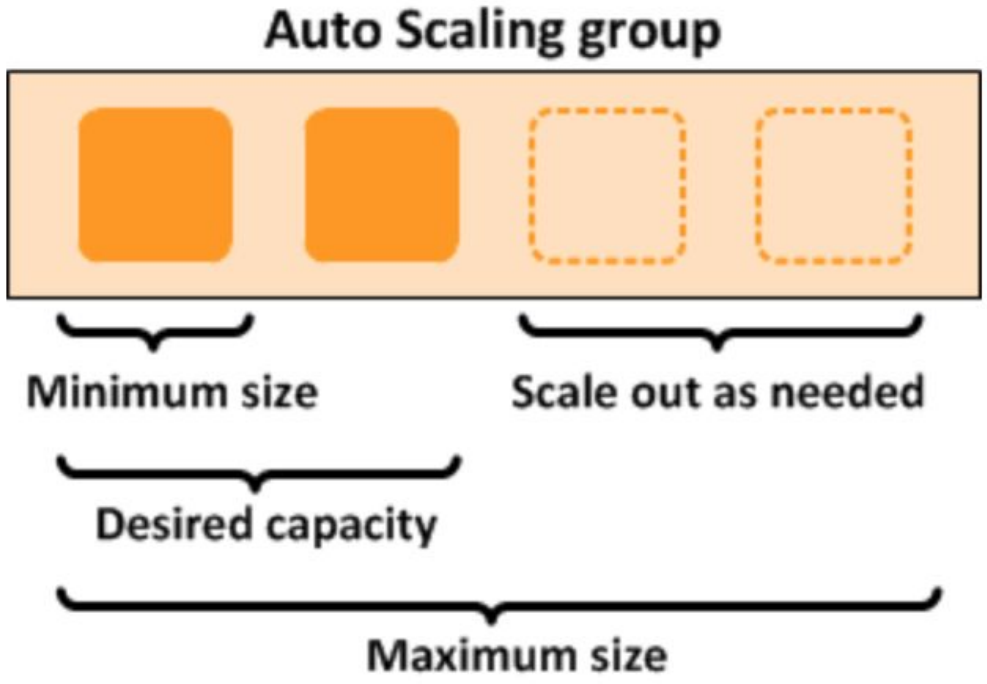
- With SSL certificate we can have the traffic between the clients & load balancer to be encrypted in transit (in-flight).
    - SSL - Secure Socket Layer
    - TLS - Transport Layer Security, Newer version than SSL
- These SSL Certificates are issued by CA (Certificate Authorities)
- e.g GoDaddy, Digicert, Symantec etc..
- You have to renew the SSL certificates as they have expiry dates.
- LB uses X.509 certificates (SSL/TLS server certificates)
- You can manage all the certificates in Amazon's certificate manager called as AWS Certificate Manager.
- We can also create our own certificate (self-signed) ls upload to ACM.

What's an Auto Scaling Group?

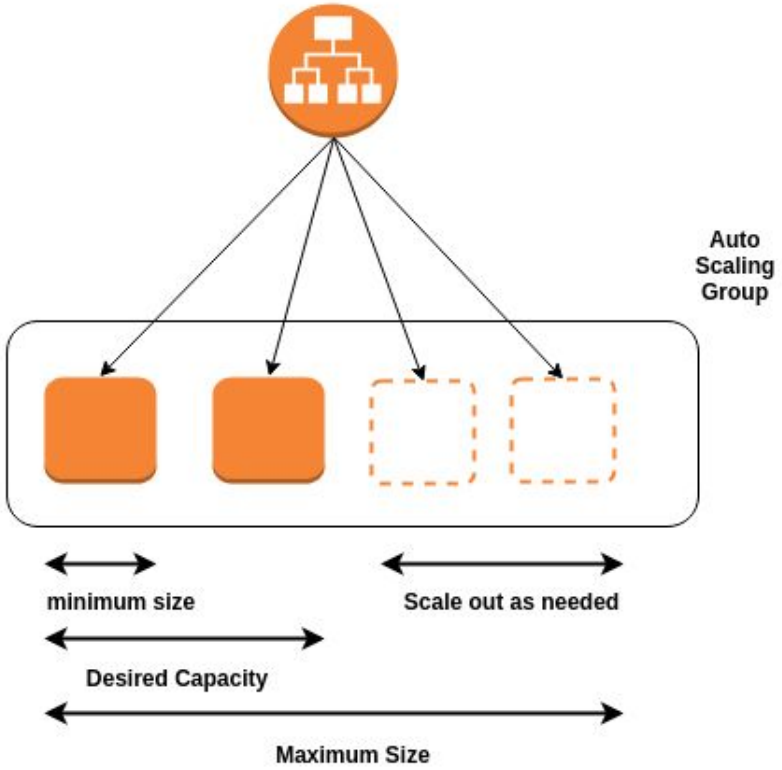
- In real time environment the traffic to your website or Application can change at any point of time.

- Why use the Auto Scaling group (ASG) ?
  - Scale out (add EC2) to match the high load/request.
  - Scale in (remove Ec2) to match the low load/request
  - Making sure we have a minimum & maximum number of machines running.
  - It can automatically register new instances to a load balancer.

Auto Scaling Group in AWS ?



Auto Scaling with AWS Load Balancer ?



ASGs have the following attributes:

- A Launch configuration.
  - AMI + instance type
  - Ec2 User data
  - EBs volumes
  - Security group
  - SSH key pair
- Min Size / Max size / Initial capacity
- Network + subnet Information
- Load balancer information
- Scaling Policies

Auto scaling Alarms.

- Possible to scale based on Cloudwatch alarms.
- An Alarm monitors metrics likes (cpu, network, etc).
- Metrics are calculated for over all ASG instances.
- Using these alarms we can:
  - create scale out policy (adding instances)
  - create scale in policy (removing instances)

Auto scaling Custom metric.

- We can auto scale based on custom metric
- We have to send custom metric from application on EC2 to AWS Cloudwatch
- Using cloudwatch Alarms to react to low/high values
- Using the created alarm as scaling policy for ASG.