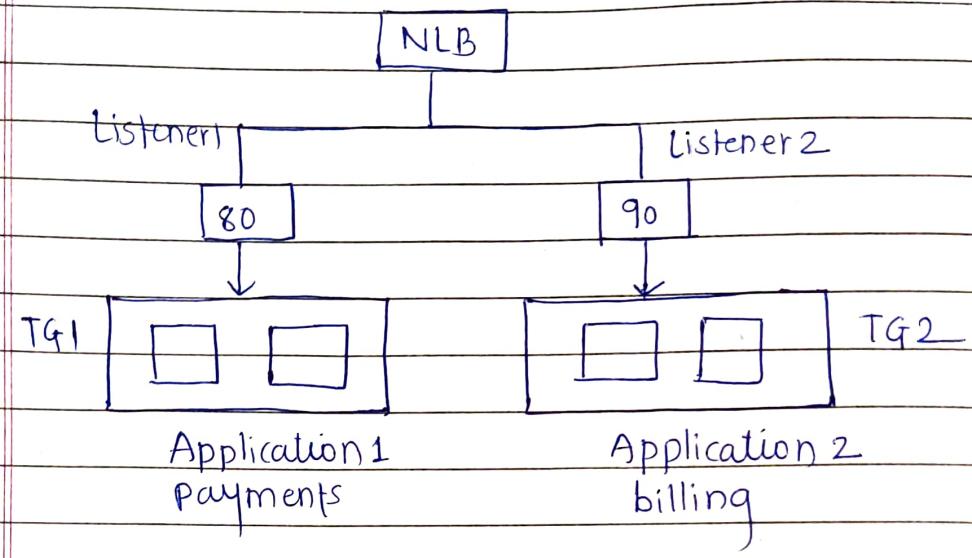
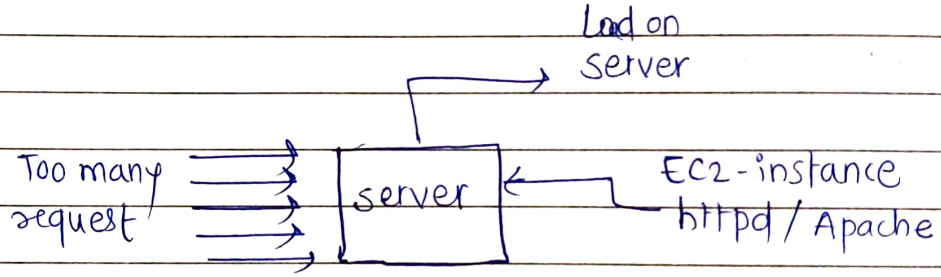
Friday
25/02/2022

Auto-Scaling

Assign :- Create a NLB → Listener 1 – port 80
→ Listener 2 – port 90

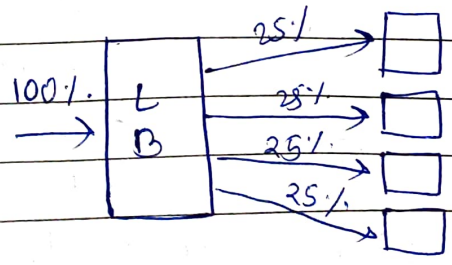




If load on server increases, CPU utilization of EC2-instance on which server is installed, will increase.

If CPU utilization → 100 %.
Server will crash
m/c will be stopped

To distribute load



If load = 500 %
then each m/c will get 125 %
but capacity of m/c 100 %.
∴ whole infrastructure will be down

If load is 500%
If we add 10 m/c → $\frac{500}{10}$ = 50% (each m/c will get 50% load)

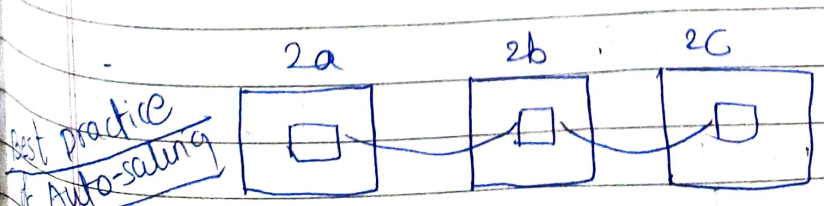
If load decreases to 50% → $\frac{50}{10}$ = 5% (for each m/c)
It is not feasible so we remove m/cs ↑ money will be wasted

So, as per our requirement, we are increasing or decreasing the capacity of resources manually ie. manual scaling.
But, it is not feasible everytime. So we go for Auto-scaling

→ **Auto-Scaling** :- As per CPU utilization, or as per load resources will be increased or decreased automatically.

Adv:-
1. Scalability
2. High availability
3. Fault Tolerance

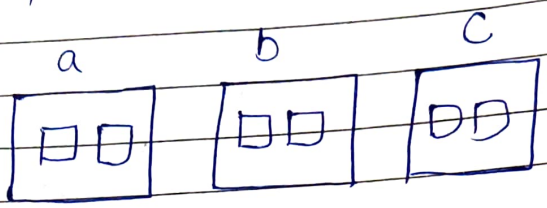
- It is Region specific (cannot do auto-scaling between 2 Regions)
- No extra cost is needed for Autoscaling
- Autoscaling can be between zones (It is not zone specific)



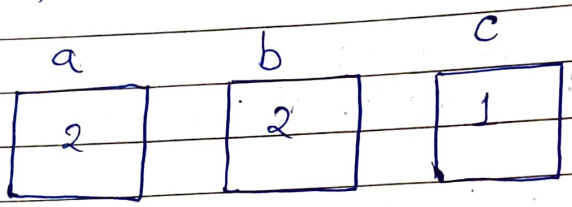
resources should be divided equally in each zone (Balanced)

Best practice of Auto-scaling
- During Autoscaling make sure that equal amount of EC2-instances/resources should be in each zone.

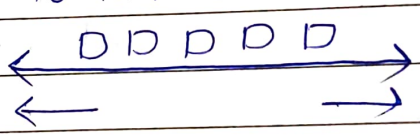
eg. If we have 6 m/c



eg. If we have 5 m/cs → it shouldn't be 3,1,1



* Scale up/out → To increase

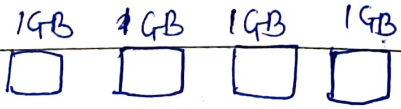


* Scale In/Down → To decrease

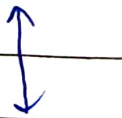
* Types of Scaling :-

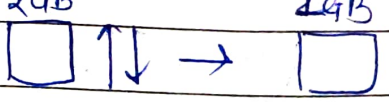
① Horizontal Scaling

- same types of m/c/resources are launching



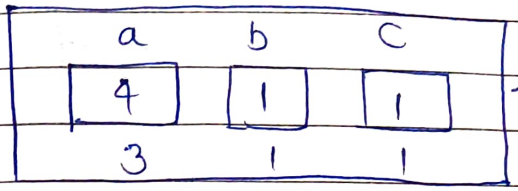
② Vertical Scaling





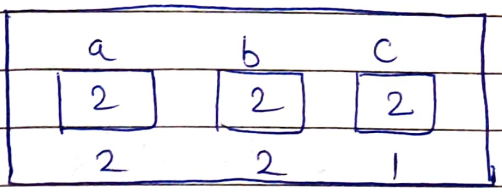
or

Same instance is configured differently



Not a best practice as it is not balanced / not equally distributed

Rebalancing



→ Balanced (~~Done automatically~~)

minimum → min. instances when scaling ~~out~~ down
maximum → max. instances when scaling up.
Desired → how much instances running in current state / now

If load increases, ~~instances~~ will increase above desired value & reach upto max. value.

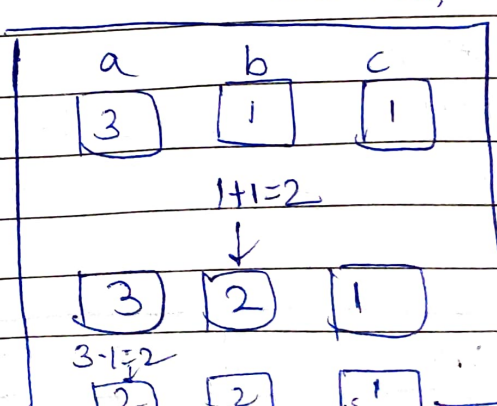
If load decreases, instances will decrease automatically below the desired value & reach upto min. value

In Rebalancing, first addition & then subtraction

↓

→ 10% of max value or 1 instance

If max = 10, it will allow 1 instance
max = 5, it will allow 1
max = 20, it will allow 2 instances to add / max value for a time being

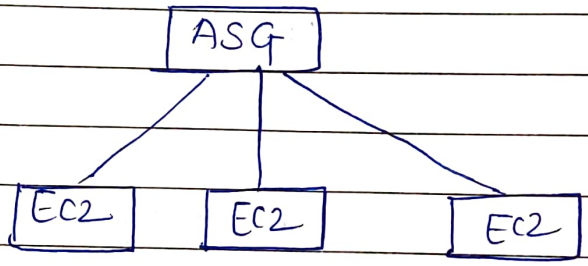


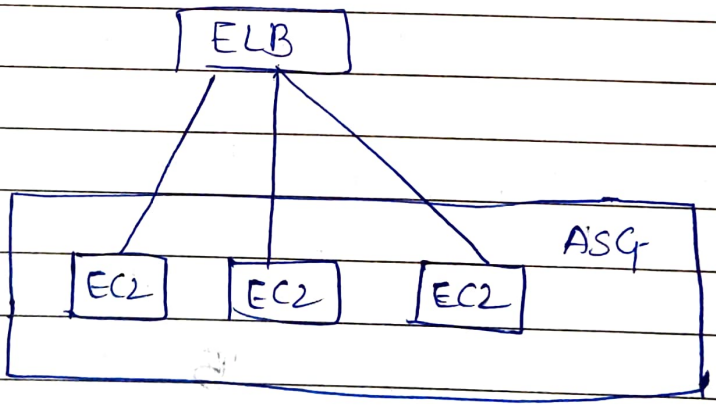
→ Re-balancing is needed in case of

1) failure of instances
2) manual addition of instances

To add an instance, it should be running & should be in same AZ ~~as~~ i as to ASG.

→ Health check is imp in ASG (Automatic Scaling Group)





If index.html corrupts / httpd fails ELB will not give 200 ok! but ASG shows all the targets are healthy.

∴ We need to do health check in both ELB & ASG

Health check grace period = 300 sec (default) → waiting period till 1st health check from launch time

In Health check, if unhealthy, it removes the instance first and then adds another instance.

RB	+	-	Rebalancing
HC	-	+	Heath check

+, - is of instances

- If EIP / EBS volume (Additional) gets detached from terminated instances, you have to attach it manually.

- In case of ASG, if patching is being applied on a particular instance, ASG will not terminate it, it will keep that instance on stand by mode. and after patching is done, ASG will attach it.

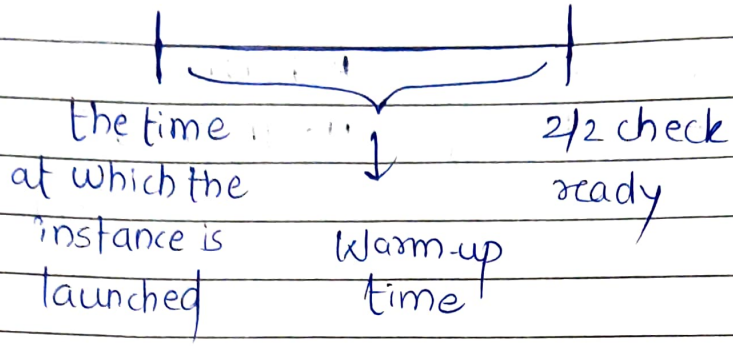
Patching / update is done when the load is low.

* **Auto-Scaling components :-**

1) **Launch configuration** → We can select AMI, SG, keypair, type (t2.micro etc)
   - New instances will be launched according to LC
   - We cannot edit LC, Once created We can only delete or copy

2) **Auto-Scaling Group** → We can select Group Name, Group Size (max, min, desired) & HC

* Warm-up time of an instance :-



- It defines the no. of seconds it takes for a newly launched instance to warm-up or to be in ready state.

* Cool-down period :-

example

↑ ASG (If load increases, ASG will do scaling) ⟶
- Scaling (1st) ⟶ cool-down period (In this, all alarms are ignored)
- Scaling (2nd) ⟶ or next scaling ⟶ cool-down period [no addition/decrease of instances during this period]
- next scaling

- It is a period of time after each scaling action is complete. During the cooldown period, scaling actions triggered by alarms will be ignored/denied.