**Interview Question AWS**

**EC2 ( Elastic cloud comput**e) – used to create instance

When try to launch instance with required requirement it will ask to select existing pair of key or create new (select based on requirement) , If you create new key it will be in pemp format

Then follow below steps to connect server.

* **Unix servers :**

**Login through GitBash -**

Select instance -> connect in AWS -> copy ssh link -> open git bash & paste the link and hit

**Or**

**Login through putty**

Convert pepm key file to ppk file putty zen app -> Putty enter server name -> browse ppk file to AUT in putty -> connect

* **Windows server :** Select instance -> connect in AWS-> generate key using pemp file -> RDP to connect server

**Action ->Instance -> change termination policy :** If you enable this user cant terminate the server.

**EBS ( Elastic block storage) –**  Used to backup Volume

**Volume:** Used to create new volume to existing instance

Create volume -> action -> attach to instance

**Note** : Region of the volume should be same as instance volume else we have to copy volume to instance region later we can attach the volume.

**Snapshot :** it capture the data written in volume and this can be used to create other instance.   
while creating snapshot we can also increase volume or can attach the volume.  
 when ever same volume is required in other instance we can use this concept .

* We can create snapshot of default volume or elastic volume
* We can copy data of same region or different region.

Select Volume -> Action -> create snapshot : This will create snapshot

Go to Snapshots -> select required Sanpshot -> Action -> Create Volume -> attach to required instance

**IMAGES :   
AMIs (Amazon Machine Images) :**  
when we required instance with same configuration which we have already then we can create an image of instance and with that image we can deploy any number of instance.

Select instance -> Action -> create image : this will create an image

While creating instance we can use the image which we create will be having same configuration and data

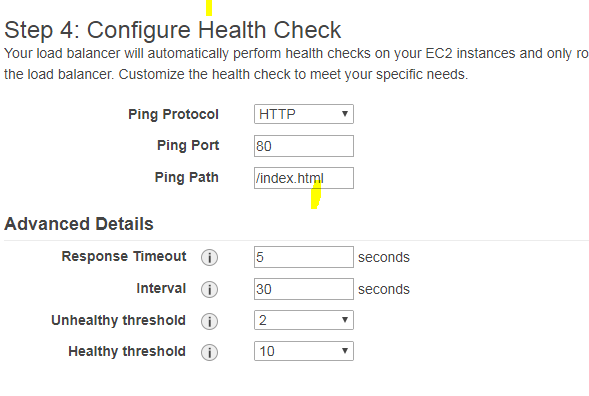
**Load balancing :** used to balance the traffic which flow into the application.  
we can create load balance for internet access or internally

**ELB(Elastic Load balance) or Load balancer :**

There are 3 types of load balancer :

* [**Application Load Balancers**](https://docs.aws.amazon.com/elasticloadbalancing/latest/application/): this will distribute the traffic based the request is received.   
  eg : It should balance based on request for image / request for blog
* **Classic Load Balancer** : This will equally distribute the traffic among the number of server.

Load Balance -> Create load balancer -> Classic Load Balancer -> add protocol if any ->  Configure Health Check -> Add EC2 Instances -> create

**Response Timeout :** time to wait when receiving a response from the health check

**HealthCheck Interval :** time between health checks of an individual instance

**Unhealthy Threshold:** number of consecutive failed health checks that must occur before declaring an EC2 instance unhealthy

**Healthy Threshold :** number of consecutive successful health checks that must occur before declaring an EC2 instance healthy.

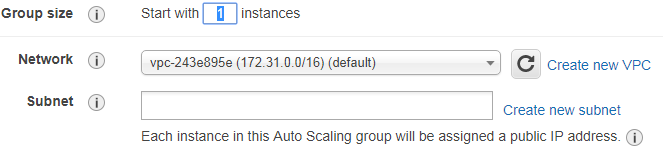
**Auto Scaling :** Adding or removing the instance based on the traffic flow into the application .

We have 2 steps here

**Launch Configuration :** This is similar to instance creation. We need to provide configuration details for the instance.

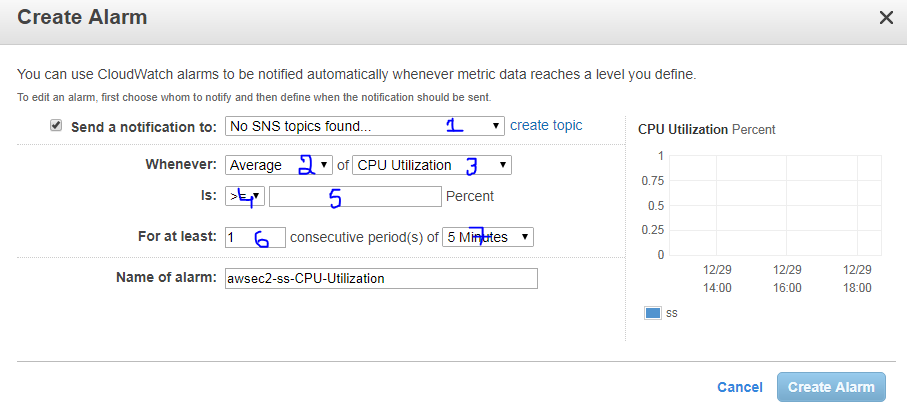
After this need create auto scaling group using create launch configuration

**Create auto scaling group :** Configuration set up is below

**group :** initial size of the group

**Subnet :** Select the subnet or region where instance need to get created (atleast 2 instance need to select). Configure Auto Scaling group details

In **Configure scaling policies** : Here we need to select min and max number of instance required in auto scaling   
  
And we have to provide condition when scale up or scle down should happen and on what basis it should happen.

Here provide recipients for email notification(1) , when instance need to be added (2), based on what criteria(3) , condition(4) , percentage(5) , how much instance needs to be add(6), with in what time instance need to be added(7). **Note :** Auto scaling will add or remove the instance but required ELB to balance the load between the instance

**VPS (Virtual Private Cloud) :** Used to create private virtual network on cloud in AWS and defines IP address in one range.

**Required Networking concept to create VPC :**

**Subnet :** Is a smaller part of a network and all the subnets in the networks will be having IP address with same prefix.

**SubNetMask** : Determine the size of the IP address of the subnet

**Internet gateway :** Is a public network routing If the subnet as internet gateway then it will public subnet else private subnet

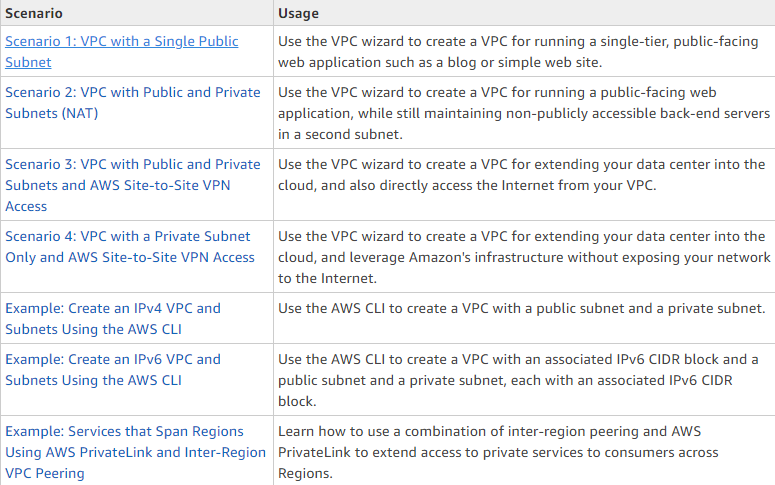
**Route table :** Determine how traffic will flow in internet gateway or Ethernet

**NAT :** It enable traffic from private subnet to internet or other **AWS** services , but prevent the internet from initiating a connection with those instance.(Charges will be applied to create NAT in AWS)

**VPC peering :** Used to connect VPC with another VPC via network route using private **IP** address.

**Benefits and difference between default/Non default VPC :** Mobile screenshot

**VPC wizard scenarios :**

****

Based on the above VPC wizard we will create VPC , Then this VPC is used to launch instance but public IP wont be available for instance.

But we can create allocate new public IP to VPC using Elastic IP option .

**IAM(Identity access management) :**

Below are the components of IAM :

1. **Users :** Used to create and manage AWS users , provide required permission to the user.

Programatic access : ? will get access key n secrete access key

AWS Management console access : ?

1. **Groups :** we can create group like user . when we provide permission to group then same permission will be apply to the user.
2. **Roles :** It is associated with the application not with user or group.

Use case : When EC2 instance need access to S3

Roles can be assigned to any type of AWS service (Select EC2) and select policy (access to S3) -> create role - ? then attach role to Ec2 instance  
here don’t want to specify any access or create access key for communication.

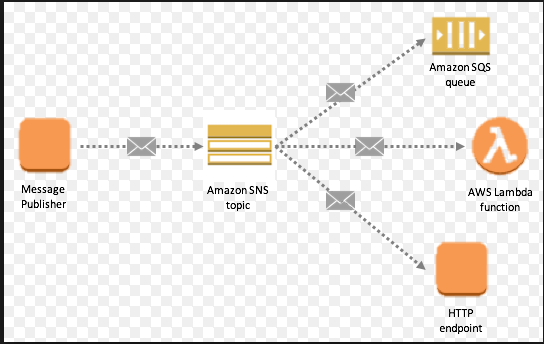
1. **Policy :**  Is nothing but permissions , To assign any permission to user/group/role we need to create policy

We can create policy using default policy/ using policy generator/ writing own policy

**Multi file authentication :** its just list OTP authentication normally will use.  
We can enable MFA authentication to users(enable MFA in security credential )

**SNS (Simple notification service) :**

This is push based service means this will push message to recipients. This is used to send messages from application.  
  
**Architecture :** publisher will forward the message to SNS topic , Then topic will send notifications line SMS/Email/Http/https etc



For SNS feature to be added in your application first create Topic and subscribe the topic -> Then subscriber need to subscribe the notification

**AWS cloud watch :**

Is a matrix based monitoring tool, we have to select a matrix for monitoring and need to provide threshold for selected matrix . when matrix crossed the threshold then it will trigger an alarm.

When alarm is triggered below 3 actions are possible :

1. SNS email notification
2. Auto scaling
3. EC2 Auto recovery/monitor 3/Monitor ELB/DB in RBS (recover/Stop/Terminate)

There are 2 types of monitoring in cloud watch:

1. **Basic monitoring :** Its Free and its monitor the system every 5 mins
2. **Detailed monitoring :** Its charged and its monitor the system every 5 mins.

**Events in Cloud watch :**  when ever status changed to any of the services in AWS this will create an event. Based on the event triggered we can send notification any functions (lamda/SNS/SES….))

**Dashboard :**

**S3 (simple storage service) :**

Used to store any amount of data anytime anywhere on the cloud. S3 is bucket that stores the data inthe form of object. Each file or anything we uoloaded into to bucket will store as object.  
 For every object inside the bucket will be having unique key that is used to identify object.   
 Max size of the each object in bucket 5TB any bucket can have any number of objects.

Each object can be stored in 3 class :

**Standard :** object that needs to be download/ access very frequently stored here.  
Eg : static file, Css file, jsp file etc.

**Standard-Infrequent Access :** object that needs not to be download/ access very frequently stored here. Eg : backupfile

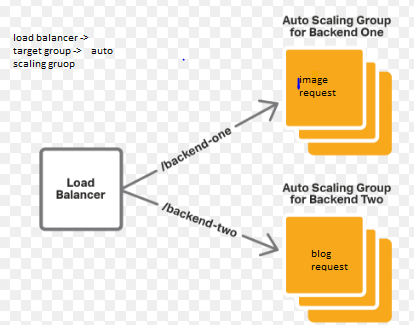
**Reduce redundency :** When we create a bucket in S3 same bucket will replicated to other 3 region but if we use reduce redundency S3 will replicate to only 2 region

**Note :** When you create a bucket or an object, Amazon S3 creates a default ACL that grants the resource owner full control over the resource.

**(Get KT)v ????????????????**

**Use case for Auto Scaling group and load balancer :**

* Request for image : traffic should move to one instance
* Request for blog : traffic should move to other instance
* Use auto scaling for traffic flow



Create 2 instance with required configuration and software   
create images of the above instance   
with the above instances launch a configuration for auto scaling group (provide policy details)  
create a application load balancer and 2 target groups (bez we have 2 auto scaling as a target)

Go to autoscaling group attach each one to target group which we have created in load balancer.  
Go to load balancer and provide condition when traffic should move to different task group in monitor tab.

**Here loadbalancer will connect to task group which interns connect to autoscaling group**

**Doubt :**

**S3 buckets   
dashboard in cloud watch   
IAM – types of access AWS manag. Access / programmatic access**